PROBABILISTIC MODELS, METHODS, AND SOFTWARE FOR EVALUATING RISK TO CIVIL INFRASTRUCTURE

by

Mojtaba Mahsuli

B.Sc., Sharif University of Technology, 2004

M.Sc., Sharif University of Technology, 2006

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

(Civil Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2012

ABSTRACT

The fundamental objective in this thesis is to advance the state-of-the-art in the field of infrastructure risk analysis. To meet this objective, probabilistic models, methods, and software are developed and applied. The work is conducted within a new reliability-based approach, in which reliability methods are employed to compute risk. Risk, in this context, means the probability of exceeding monetary loss. Evaluating such probabilities requires probabilistic models for hazards, response, damage, and loss. This motivates the contributions in this thesis, which are summarized in the next paragraphs.

First, a new computer program, called Rt, is developed. It is tailored to conduct reliability analysis with many probabilistic models. It orchestrates the interaction of models by means of a new object-oriented software design. Each model and analysis algorithm is represented by an object. As a result, new models and algorithms are easily implemented without modifying existing code. Another novelty is the parameterization of uncertainties, decisions, and model responses. This has several implications; one being that, in each step of an analysis, only the models affected by new parameter realizations are evaluated. Another novelty is the computation of “direct differentiation” response sensitivities in a multi-model analysis.

Second, a library of new probabilistic models are developed and implemented in Rt. The models are intended for use in regional seismic risk analysis. The library includes new models for location and magnitude of earthquakes, and response, damage, and loss of building. The library also features damage and loss models for entire regions.

Third, the models are applied in a risk analysis for the Vancouver metropolitan region in Canada. The primary results are “loss curves” and “hazard curves,” which show the probability of exceeding loss and spectral acceleration, respectively. As another example of results, it is found that Richmond is the most vulnerable municipality.

Finally, new sensitivity measures are developed to prioritize the allocation of resources to mitigate risk and to reduce model uncertainty. In particular, these measures identify the buildings whose retrofit yields the most reduction in regional risk. They also identify the models whose improvement yields the most reduction in uncertainty.

PREFACE

Chapters 2 to 5 of this thesis are based on versions of four manuscripts that have been accepted or submitted for publication in peer-reviewed journals:

- Chapter 2: Mahsuli, M. and Haukaas, T. “A Computer Program for Multi-model Reliability and Optimization Analysis.” ASCE Journal of Computing in Civil Engineering. Accepted for publication in February, 2012.
- Chapter 3: Mahsuli, M. and Haukaas, T. “Seismic Risk Analysis with Reliability Methods, Part I: Models.” Submitted.
- Chapter 4: Mahsuli, M. and Haukaas, T. “Seismic Risk Analysis with Reliability Methods, Part II: Analysis.” Submitted.
- Chapter 5: Mahsuli, M. and Haukaas, T. “Sensitivity Measures for Optimal Mitigation of Risk and Reduction of Model Uncertainty.” Submitted.

The author of this thesis is responsible for reviewing the literature, deriving equations, developing models, computer programming, data processing, conducting analyses, and interpreting the results. The manuscripts were drafted by the author of this thesis and finalized in an iterative process with the thesis advisor, Dr. Terje Haukaas. The author of this thesis is responsible for preparing the tables and figures.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Terje Haukaas, for his constant support, generosity with his time and his vast knowledge, and insightful advises throughout my doctoral studies. I would like to thank the members of my PhD supervisory committee: Dr. Ricardo Foschi for insightful presentations and discussions in seminars, Dr. Kenneth Elwood for teaching me many things about earthquakes, and Dr. Tuna Onur for fruitful comments and discussions.

I gratefully acknowledge the Izaak Walton Killam Memorial Doctoral Fellowship from the Killam Trusts and the financial support from the Natural Science and Engineering Research Council of Canada (NSERC). I also thank Dr. Carlos Ventura for providing the building inventory database of Vancouver, and Dr. Gail Atkinson for providing the shear wave velocity data of the Fraser Valley region. I am thankful to Mr. Aprizal Kristijanto for surveying buildings at the campus of the University of British Columbia in Vancouver. I appreciatively thank the faculty and staff of the Department of Civil Engineering for making up such a great environment for me to learn and grow.

I would also like to gratefully recognize my friends and fellow students at the Department of Civil Engineering for their helpful discussions and joyful company. Special thanks to Smitha, Armin, Abbas, Alireza Forghani, Majid, Ehsan, Alireza Ahmadnia, Alejandro, Laura, Navid, Mehdi, Soheil, and Jose. I am truly thankful to my dear friends, especially Mohammad Sadegh, Omid, Mahdi, Ramin, Hamid, Mohammad, and Hayder, who have been my family here in Vancouver.

I am utterly thankful to my wife, Fatemeh, for her loyal and continual support and inspiration throughout the long years of my studies. Words cannot express my appreciation to her. Finally, my heart is filled with deep respect and gratitude to my mother, my father, my sister and brother-in-law, my brother, and my wife's family for their patience, unconditional support, and encouragement, which never diminished by the long distance between us.

Dedication

To my wife, Fatemeh

And to my parents

Chapter 1. INTRODUCTION

1.1. OBJECTIVES AND SCOPE

This thesis targets the area of risk analysis in structural and earthquake engineering. Risk in this context denotes the probability of exceeding a measure of utility, such as economic loss. A new approach for risk analysis is employed, in which reliability methods compute the exceedance probabilities. Reliability methods have been developed over the last three decades and include first- and second-order reliability methods (FORM and SORM) and a variety of sampling schemes. Reliability methods are suited for risk analysis, because they are tailored to compute the probability of rare events. Such events are particularly important in risk analysis applications, especially in seismic risk, because they typically have dramatic impacts, *e.g.*, high monetary loss.

In a reliability analysis, random variables describe the uncertainty and a limit-state function defines the event for which the probability is sought. In classical structural reliability, the limit-state function is defined in terms of the demand and capacity of the structure. To extend the usage of reliability methods to risk analysis, the present study expresses the limit-state function in terms of the utility, here, seismic loss. In this case, the limit-state function defines the event that the loss exceeds a prescribed threshold. The loss depends on the damage, which in turn depends on the structural response, earthquake intensity, location, and magnitude. Each of these phenomena is represented by a model in this approach. These models are probabilistic and they discretize the uncertainty in terms of random variables. In the course of a reliability analysis, the limit-state function and possibly its gradient are repeatedly evaluated. In each evaluation, the models receive the trial realization of the random variables as input, and output physical responses, such as earthquake intensity or repair cost. These responses enter another model "downstream" as input, or directly enter the limit-state function. In summary, risk analysis with reliability methods requires a host of probabilistic models.

Orchestrating a reliability analysis with multiple models requires software tools. The first objective in this study is to develop a computer program for this purpose that addresses the challenge of coordinating many models. The program should have an object-oriented design to readily facilitate the implementation of new models and analysis algorithms. To efficiently evaluate the gradient of the limit-state function, the program should be capable of computing direct differentiation response sensitivities and communicating them between multiple models. Finally, to compute the risk when several hazards are present, multi-hazard analysis methods should be implemented in the program with hazard combination capabilities.

The second objective is to develop a library of probabilistic models. The use of reliability methods requires the models to meet a number of conditions. These conditions are explained in detail later in this thesis. An important condition is that the uncertainty in the models should be described by random variables, and the model should output a physical measureable quantity, not a probability. Therefore, conditional probability models, such as fragility models, are not suited for use in reliability analysis. The new library of models is intended for use in regional seismic risk analysis. The scope of these models is limited to earthquakes and buildings. In particular, the objective is to develop the following models:

1) Earthquake location model for arbitrary-shaped area sources, which produces realizations of the earthquake location
2) Earthquake magnitude model, which accounts for the uncertainty in the seismic parameters of the earthquake source, such as the maximum magnitude that the source can generate

3) Building response models, which output the peak drift and acceleration response given the earthquake intensity and building characteristics, such as height, age, material, and load bearing system
4) Building damage models, which outputs the structural and non-structural damage given the responses and building characteristics, including the irregularities of the structure
5) Building loss models, which output the total repair cost given the damage and building characteristics, including the occupancy type and area of construction
6) Regional damage models, which output the seismic damage to an entire region given the earthquake intensity and the type of land use
7) Regional loss model, which outputs the total repair cost of the buildings in a region given the damage and the area of construction

Items 1) and 2) aim at explicit modeling of the location and magnitude of earthquakes. This contrasts the traditional risk analysis approaches, where the uncertainties in the location and magnitude are implicit in a “hazard curve.” An important goal in items 3), 4), and 5) is to develop building models that take surveyed data as input. Such data are gathered by visual inspection of buildings, and also through satellite imagery and municipal databases. Finally, the goal in items 6) and 7) is to develop models that predict the damage and loss when data about the individual buildings in a region are unavailable.

The third objective is to apply the library of models and the computer program in risk analysis of a specific region that is subjected to seismicity from several sources. The primary results are “hazard curves,” which show the probability of exceeding values of a spectral ordinate, and “loss curves,” which show the probability of exceeding loss values. It is acknowledged that the building stock in the region may undergo some changes over time, *e.g.*, due to new construction, population shift, and changes in land use. However, the scope of this analysis does not include these changes. The results should therefore be interpreted in the context of uncertainties that are modeled, *i.e.*, uncertainties in earthquakes and in the current state of buildings.

The fourth and final objective addresses the risk once it is evaluated, *i.e.*, once the first three objectives are accomplished. This objective is twofold: 1) To improve the risk estimates by reducing model uncertainty; 2) To mitigate the risk by taking retrofit actions. In fact, both objectives address the problem of allocating limited resources, one for model improvement, and the other for retrofit. First, in order to improve the risk estimates, it is essential to refine some of the models in the analysis with more detailed modeling, or to improve the models in the library upon availability of new observations. This motivates the development of sensitivity measures to prioritize the models for refinement and improvement. Second, in order to mitigate the risk, it is essential to retrofit the infrastructure components over time. This motivates this study to develop a sensitivity measure to prioritize infrastructure for retrofit actions. It is noted that in this thesis, the phrase infrastructure includes buildings. In fact, buildings are the primary focus in the analyses that are carried out.

1.2. BACKGROUND

This section presents an overview of the literature that is pertinent to the overall research thrust of this thesis. The literature that pertains to the specific subject of each chapter is reviewed in the introduction of the respective chapter.

The first efforts to carry out probabilistic hazard analysis date back to the late 1960's. The seminal paper by Cornell (1968) pioneered the field of probabilistic seismic hazard analysis. The results of this work were in terms of a ground motion parameter, such as the peak ground acceleration, versus the mean return period. This is a form of conditional probability and is commonly referred to as hazard curve. Since then, many researchers have contributed to this field. The early studies in this field were focused on developing hazard curves for peak ground motion parameters, such as peak ground acceleration and velocity. These parameters purely depend on the hazard characteristics and are independent of structural properties. In the 1970’s, ground motion equations and seismic hazard curves were developed directly for spectral ordinates (McGuire 1974; McGuire and Barnhard 1977). This development introduced an elastic measure of structural response, *e.g.*, the spectral acceleration, in the risk analysis. McGuire (2008) presents an overview of the evolution of probabilistic seismic hazard analysis. Kramer (1996) and Thenhaus and Campbell (2003) provide a description of probabilistic seismic hazard analysis together with many relevant references. The models developed in this field were in the form of conditional probabilities, *e.g.*, they produced the probability of exceeding a spectral acceleration given a magnitude and a distance. Probabilistic hazard analysis was the first step towards the development of risk analysis methods. In passing, it is emphasized that the words hazard and risk are given the following definitions in this thesis: Hazard describes the probability of exceeding a measure of ground motion intensity, such as spectral acceleration. Conversely, risk is the probability of exceeding monetary loss.

In the 1980’s, researchers aimed at going beyond structural response in risk analysis. Steinbrugge (1982) and ATC-13 (ATC 1985) proposed methodologies to compute damage in the structure. Both methods employed empirical relationships to compute damage conditioned upon the modified Mercalli intensity (MMI) scale. The result of the Steinbrugge method was a probable maximum loss, which is a loss that is not exceeded by 9 out of 10 structures. In other words, the computed loss is at 90% confidence interval, and is *not* the mean loss. The result of the ATC-13 approach was a mean damage factor. ATC-13 defined this damage factor as the mean cost of repairing the structure divided by the replacement cost. According to the definition adopted in this thesis, the replacement cost is what it costs to replace a building with a similar type of construction. In contrast, the repair cost is what it costs to restore the building to its undamaged condition. For this purpose, ATC-13 categorized buildings based on their structural system and developed a “damage-probability matrix” for each category based on expert opinion. Each element of a damage-probability matrix is the probability of falling within a “damage state” given an MMI. Using the theorem of total probability, the seismic hazard could be translated into seismic risk by integrating the conditional probabilities of damage over the probability distribution of MMI.

In the late 1990’s, analytical models for seismic damage estimation became more common. A new loss estimation methodology was developed by the U.S. Federal Emergency Management Agency and the U.S. National Institute of Building Sciences (FEMA-NIBS 2003). The methodology was implemented in the HAZUS® computer program. This approach employed the capacity spectrum method (Freeman *et al.* 1975; Mahaney *et al.* 1993) to compute the drift and acceleration response of the building. Damage was estimated using fragility curves. Fragility curves in this methodology provide the probability of exceeding a damage state given the structural response. In turn, each damage state is associated with a damage factor. The methodology calculates an expected damage factor and thus, an expected loss, using the fragility conditional probabilities. The notion of fragility has become popular within the earthquake engineering community. The FEMA-NIBS fragility curves were developed for the global response of the structure. However, many studies have focused on developing fragility curves for individual components of a building, such as columns (Choe *et al.* 2008).

The Pacific Earthquake Engineering Research (PEER) Center put forward a methodology for seismic risk assessment. This methodology was originally proposed by Cornell and Krawinkler (2000). Later, it was presented in more detail by Moehle and Deierlein (2004). The result of this approach is the probability distribution of the repair cost. The models in this methodology have the format of conditional probability. At the core of this approach, the theorem of total probability is employed in the form of a triple integral, known as PEER framing equation. The probability distributions within the integral include: a) distribution of earthquake intensity, b) distribution of structural response given the earthquake intensity, c) distribution of damage given the structural response, and d) distribution of impacts given damage. This integral form was initially formulated to coordinate the research efforts into hazard modeling, structural modeling, damage modeling, and impact modeling, which address the items a) to d), respectively. Nearly a decade later, Yang *et al.* (2009) proposed a sampling-based approach to evaluate the integral. Several other research institutions have developed similar formulations, such as the Mid-America Earthquake Center (Ellingwood and Wen 2005) among others. In contrast to these approaches, the risk analysis approach in this thesis employs reliability methods in conjunction with many probabilistic models. The Introduction of Chapter 2 presents a detailed comparison of the two approaches.

The central theme in the aforesaid studies is the formulation of probabilistic models in the form of conditional probabilities. In parallel with these developments, structural reliability and safety has been a significant field of research. Structural reliability focuses on the development and application of reliability methods. These methods provide a means of evaluating the probability that an event of interest occurs. In fact, these methods are capable of accurately computing the probability of rare events, which are remarkably important in seismic risk analysis. This makes these methods applicable in risk analysis. Starting from the 1920’s, several researchers pioneered the field of structural safety, such as Meyer (1926), Prot (1936), Weibull (1938; 1939a; 1939b), Kjellman and Wästlund (1940), and Johnson (1953).

In the late 1960’s, the first formulation of the reliability index was put forward by Cornell (1967). Cornell formulated the reliability index as the ratio of the mean to the standard deviation of the limit-state function. This reliability method is known as the mean-value first-order second-moment method. In the 1970’s, the so-called “invariance problem” that is associated with this method was addressed by developing FORM (Ditlevsen 1973; Hasofer and Lind 1974). In the 1980’s, SORM was developed that increases the accuracy of the computed failure probability (Breitung 1984; Hohenbichler and Rackwitz 1988). In addition, many researchers have worked on developing various sampling schemes to compute event probabilities. Ditlevsen and Madsen (1996) and Der Kiureghian (2005) provide a comprehensive description of reliability methods.

The research on structural reliability in the early stages was mainly concentrated on developing reliability algorithms. Simple probabilistic models were often employed to demonstrate these algorithms. A coupling of the reliability methods with finite-element analysis has been the first step towards reliability-based risk analysis. Worth mentioning is the pioneering work of Der Kiureghian and Taylor (1983). Since then, a number of advances have been reported, including those by Der Kiureghian and Zhang (1999), Sudret and Der Kiureghian (2002), Imai and Frangopol (2000), Haldar and Mahadevan (2000), and Haukaas and Der Kiureghian (2007) among others. The probability that the structural response, *e.g.*, displacement, exceeds a certain threshold was the main focus in most studies.

In the late 2000’s, Haukaas (2008) proposed the “unified reliability analysis,” as an alternative to risk analysis based on conditional probabilities. In this approach, a “unified” limit-state function on the monetary loss is defined. This limit-state function is employed in a reliability analysis, in which individual probabilistic models estimate the ground motion intensities, structural responses, damage, and loss. Koduru and Haukaas (2010) presented an application of this approach by computing the seismic loss probabilities for a high-rise building in Vancouver, Canada.

1.3. OVERVIEW OF CONTRIBUTIONS

This section briefly describes how each chapter contributes to the research objectives, and portrays the organization of the thesis.

1.3.1. Development of Software

A new computer program is developed for reliability analysis with interconnected probabilistic models. The program is called Rt and is presented in Chapter 2. Rt is freely available online and it contains a library of versatile models. The user can also implement new models—without any recompilation of the program—by several means, including a scripting option. In the context of classical structural reliability software, the new multi-model computer program is the first of its kind. Rt orchestrates the communication of models by a new object-oriented software architecture, which is presented in Chapter 2. Parameters, models, and analysis algorithms are represented by objects. As a result, Rt is readily extended by implementing new model objects and new analysis objects, without the need to modify the existing code. Rt is parameterized with specific-purpose objects, which include: 1) Random variable objects for use in reliability analysis; 2) Decision variable objects for use in sensitivity and optimization analysis; 3) Response objects that are outputs of models; 4) Time objects for modeling time-varying phenomena. Rt is capable of computing direct differentiation response sensitivities and communicating them between multiple models. Two complementary analysis options for multi-hazard reliability analysis are implemented. One employs sampling and accommodates the inclusion of time-varying phenomena. The other employs FORM, SORM, and importance sampling and is computationally efficient. The program is illustrated by a multi-model example of a structure that is modeled in an external finite element program and subjected to multiple hazards, damage, and long-term deterioration. Appendix A provides guidelines for developers to add features to Rt, such as new models and new analysis algorithms.

1.3.2. Development of Models

A library of new probabilistic models is developed for prediction of seismic risk. The models are presented in Chapter 3. They are specifically intended for use with reliability methods to compute event probabilities, such as seismic loss probabilities. Models are proposed for earthquake location and magnitude, regional loss, building response, building damage, and building loss. They address the specific objectives that were enumerated for each model earlier in this section. Each model is presented with an explanation of its development and a discussion of its predictions. In addition, ground shaking intensity models from the literature are “smoothed” to make them amenable to reliability analysis. The models are implemented in Rt and employed in a regional risk analysis in Chapter 4. The statistics of the model parameters for regional and building models is provided in Appendix B.

1.3.3. Application to Regional Risk

Reliability methods are employed in Chapter 4 to analyze the seismic risk to the Vancouver metropolitan region in Canada. This region is also known as the Greater Vancouver Regional District. The use of reliability methods contrasts with several contemporary approaches for risk analysis. The library of models is employed in a comprehensive application that comprises 559 random variables and 3,227 model objects. The primary results are hazard curves and loss curves. Loss curves expose the seismic loss probabilities and serve as a basis for risk mitigation decisions in Chapter 5. It is found that the probability of loss in excess of $100 billion in a 50-year period is 5.6%. Findings of the analysis provide further insight about the seismic risk in the region. A summary of these findings is:

- Richmond is the most vulnerable city in the region
- Crustal earthquakes have the most influence on the loss exceedance probabilities
- Non-structural components contribute more to the losses than structural components
- Unreinforced masonry buildings are amongst the most vulnerable structural systems in the considered region

1.3.4. Development of Sensitivity Measures

A new set of reliability sensitivity measures are presented in Chapter 5. They identify the optimal manner in which to mitigate risk to civil infrastructure, and reduce model uncertainty in order to improve risk estimates. Three measures are presented. One identifies the infrastructure components that should be prioritized for retrofit. Another measure identifies the infrastructure that should be prioritized for more refined modeling. The third measure identifies the models that should be prioritized in research to improve models, for example by gathering new data. The developments are presented in the context of the campus of the University of British Columbia in Vancouver with 622 buildings. The buildings are subjected to seismicity from several sources. A comprehensive seismic risk analysis of this region is conducted, with over 300 random variables, 30 model types, and 4,000 model instances. The probabilistic models from Chapter 3 are employed and emphasis is placed on explicit characterization of epistemic uncertainty. For the considered region, the buildings that should first be retrofitted are found to be pre-code unreinforced masonry buildings. Conversely, concrete shear wall buildings rank highest on the list of buildings that should be subjected to more detailed modeling. The ground shaking intensity model for shallow crustal earthquakes and the concrete shear wall structural response model rank highest on the list of models that should be prioritized for further research.

Chapter 2. A COMPUTER PROGRAM FOR MULTI-MODEL RELIABILITY ANALYSIS

2.1. INTRODUCTION

This chapter presents the design and implementation of a new computer program. The objective of the program is to perform reliability and optimization analyses with multiple interacting probabilistic models. This is the first targeted effort at building an object-oriented software framework that orchestrates this kind of multi-model analysis. The program is general-purpose and a wide range of applications is envisioned. However, the primary motivation thus far is the simulation of events that may occur in the lifespan of civil infrastructure, such as extreme loading, deterioration, seismic damage, losses, and emissions to the environment. From this viewpoint, the new program is a tool for simulating and predicting actual infrastructure performance, including indirect impacts away from the building site. This information is unavailable from today's code-oriented engineering practice although it is essential for truly risk-based design decisions.

Prediction of possible future events is rife with uncertainty. Therefore, a fundamental premise for the new computer program is the utilization of probabilistic models. This leads to several specific objectives that are addressed in this study: a) Formulation of a consistent probabilistic model format; b) Implementation of a flexible library of steadily improving models; c) Comprehensive characterization of uncertainties by random variables; d) Propagation of uncertainties from one model to another; and e) Implementation of algorithms and a user interface that support analysis with many probabilistic models.

These objectives require a powerful and flexible probabilistic methodology. In this thesis, it is decided to adopt classical structural reliability methods in conjunction with models that characterize all uncertainties by random variables. This type of analysis repeatedly determines trial realizations of the variables, followed by evaluation of one or more limit-state functions. Figure 2-1 illustrates the overall methodology in terms of communication between the orchestrating analysis algorithm (left) and the function evaluation (right). For illustration purposes, the function in Figure 2-1 is denoted by g, and the responses from Models 1 and 2 are collected in vectors denoted by $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively. The type of function(s) that are involved depends upon the analysis type; in reliability analysis the function is a limit-state function; in an optimization analysis, the functions are constraint functions and an objective function. All functions are defined in terms of random variables, denoted by the vector $\mathbf{x}^{rv}$ in Figure 2-1, and decision variables, denoted by the vector $\mathbf{x}^{dv}$ in Figure 2-1. In the new software, the functions also take Time and Location parameters, as well as the response from other models, as described later. Particular emphasis in this chapter is placed on communication between the models, which simulate hazards, responses, damage, and impacts in the lifetime of civil infrastructure.

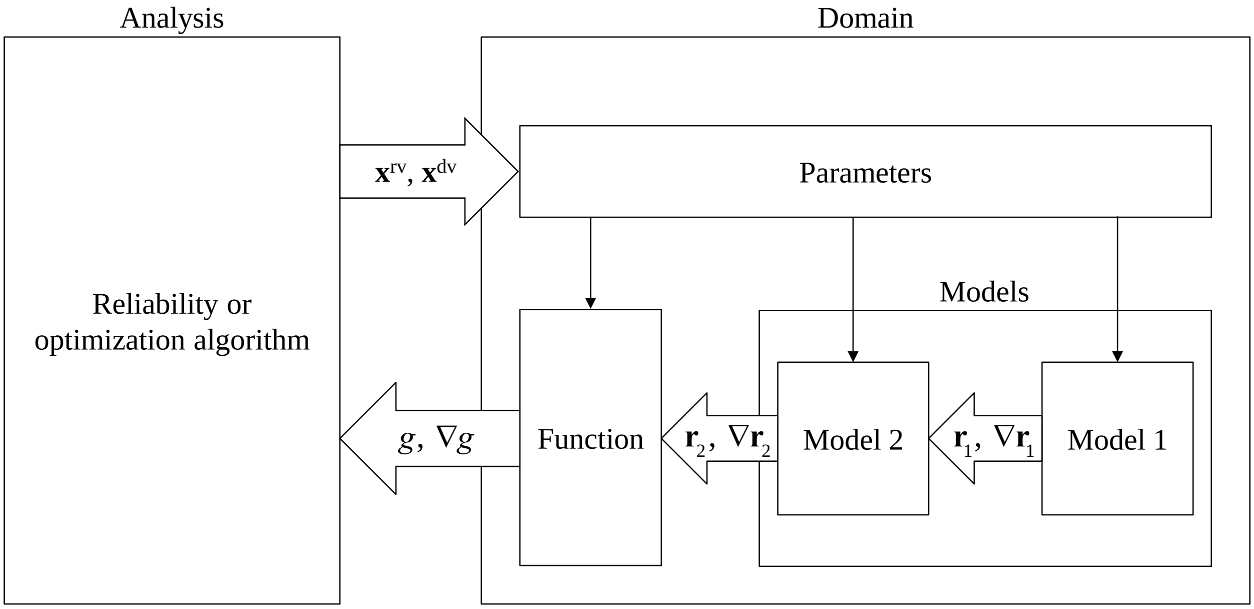

Analysis
Domain
$\mathbf{x}^{rv}$, $\mathbf{x}^{dv}$
Parameters
Models
Reliability or optimization algorithm
g, ∇g
Function
$\mathbf{r}_2$, $\nabla\mathbf{r}_2$
Model 2
$\mathbf{r}_1$, $\nabla\mathbf{r}_1$
Model 1

Figure 2-1: Overview of the analysis methodology.

The new computer program is called Rt, which is shorthand for “risk tools.” The name also alludes to the extensive use of Qt (Blanchette and Summerfield 2006), which is described later. Rt is freely available at www.inrisk.ubc.ca together with other engineering “tools.” Guidelines for developers for extending Rt with new models and analysis tools is provided in Appendix A. Rt represents a unique framework for multi-model probabilistic analysis. However, several other computer programs for reliability analysis already exist. Programs that were originally developed in the 1980’s, *e.g.*, CalREL (Liu *et al.* 1989) and RELAN (Foschi *et al.* 2000) evaluate the limit-state function by calls to a user-written FORTRAN subroutine. Although this allows any FORTRAN code to serve as a limit-state function, the sought multi-hazard analysis with many flexible interacting models for hazards, infrastructure, and impacts is impractical. Also, CalREL and RELAN are programmed with the procedural programming language FORTRAN, while Rt is implemented in an object-oriented architecture. This greatly assists the steady expansion of analysis options, and the crucial updating of the model library as new information becomes available.

Another software project that employs the object-oriented approach is OpenSees (McKenna *et al.* 2010). OpenSees is a finite element program that was amended with reliability and sensitivity analysis by Haukaas and Der Kiureghian (2007). Later, Liang *et al.* (2007) added optimization analysis options. However, OpenSees is primarily intended for analyses with one finite element model, while the primary focus in Rt is reliability and optimization analysis with many models. Although the Tcl interface of OpenSees is a scripting language that makes it technically possible to run multi-model analysis, it would be inconvenient compared with Rt. Furthermore, Rt contains a library of reusable general-purpose and special-purpose models, one of them being an interface to OpenSees, which facilitates finite element models as part of a series of models. Rt's parameterization, multi-model sensitivity analysis, and multi-hazard analysis options also contrast with the implementations in OpenSees.

An overview of computer programs for reliability analysis is provided in a special issue of the journal of Structural Safety (Schueller 2006). The software project OpenTURNS (Andrianov *et al.* 2007) represents a more recent initiative in the development of engineering uncertainty software, but the core capability remains focused on mainstream reliability methods. In summary, none of the available reliability analysis programs are tailored to work with a library of probabilistic models. In contrast, this is one of the novelties of Rt. Other contributions include: 1) Communication between many probabilistic models with emphasis on flexibility and efficiency; 2) Parameterization to facilitate the model communication, with classes that represent uncertainties, decisions, model responses, locations, and time; 3) Computation of direct differentiation sensitivities in multi-model analysis; 4) Development of a dedicated Time parameter to model time-varying phenomena; 5) Implementation of two state-of-the-art multi-hazard reliability analysis options; 6) Design of a user interface that makes multi-model reliability analysis available to a broad user group. The user interface also facilitates the modification of analysis settings during the analysis, and readily accommodates new analysis features. This chapter also presents a unique multi-model example that is analyzed in Rt, featuring risk analysis of a building subjected to multiple interacting hazards.

2.2. SOFTWARE ARCHITECTURE

Rt is implemented with the object-oriented programming language C++ (Stroustrup 2000). This means that a number of classes are developed, from which objects are instantiated at run-time. This chapter assumes that the reader has elementary knowledge of object-oriented programming, such as the presence of data members and member functions, called methods, in the classes. It is also noted that the previously presented Figure 2-1 provides an overview of the software architecture in Rt. Figure 2-1 shows how the reliability and optimization algorithms are separated from the domain, which contains the functions, models, and parameters. This chapter devotes particular attention to the domain and the interaction between models during a reliability or optimization analysis. The details are provided in the following, starting with the overview of the three groups of C++ classes that Rt consists of, hereafter called modules:

- The user interface module contains the classes that define the user interface and is described in the next section.
- The domain module contains the functions, models, and parameters. The design, implementation, and application of the domain module are the main contributions in this chapter.
- The analysis module contains algorithms for reliability and optimization analysis. At this time, it includes algorithms for FORM, SORM, and sampling reliability analysis, two options for multi-hazard load-combination analysis, and basic optimization algorithms. The object-oriented principles of inheritance, composition, and polymorphism are utilized to create a framework of analysis tools that is easily amended and interchanged. A description is omitted in this thesis because this is done in the same way as in OpenSees (Haukaas and Der Kiureghian 2007; McKenna *et al.* 2010). Some of the capabilities of the analysis module are demonstrated later in this chapter.

In its current state, Rt includes 121 classes and over 65,000 lines of code. 76 of these classes belong to the domain module, 26 classes to the analysis module, and 19 classes to the user interface module. Several programming tools are employed to shape the software architecture in Rt. The GNU Scientific Library (Galassi *et al.* 2006) is extensively utilized to carry out vector and matrix operations, as well as other mathematical computations. The cross-platform application and user interface framework Qt (Blanchette and Summerfield 2006) is also extensively employed. Qt is a powerful tool for developing user interfaces that run well on different operating systems, including Microsoft Windows®, Mac OS X®, and Linux®. Qt also provides a host of features and design patterns that are convenient for the multi-model implementations in Rt. For this reason, services provided by Qt are included in almost every class in Rt, not only in the user interface module.

The name of all classes in Qt starts with the letter Q, and for uniformity, classes in Rt are given names that start with R. The most fundamental class in Rt is RObject, which is a subclass of the fundamental class in Qt called QObject. All classes in the analysis and domain modules of Rt are subclasses of RObject. This inheritance structure offers several important advantages. Obviously, it allows once-and-for-all implementations in the base class, RObject, of features that are common to all subclasses, which is the basic premise of class inheritance. More importantly, it facilitates the introduction of parent-child relationships between objects, which is an important concept in Qt. For example, all functions, models, and parameters are made children of an RDomain object. This has the useful implication that the RDomain object is able to provide a host of information about its children. For instance, by means of the `findChildren` method, it produces *ad hoc* lists of parameters of various types. This circumvents the hard-coding of, *e.g.*, parameter lists, and greatly assists with the organization and communication between functions, models, and parameters in Rt.

In programming parlance, the RDomain object "has" the functions, models, and parameters in a relationship called dynamic composition. In contrast with the somewhat more general concepts of association and aggregation, composition means that all the objects that the

RDomain object has are deleted once the RDomain object is deleted. The word dynamic means that the relationship is established at run-time rather than at compilation-time. This prevents hard coding of lists, and allows re-parenting of children at run-time, for example to generate special-purpose domains with certain data.

The software architecture in Rt also benefits from two additional implications of the inheritance from QObject. First, it enables the definition of “meta-properties” of a class. Essentially, meta-properties serve the same purpose as ordinary data members of a class, but with useful additional features. In particular, the user interface is designed to automatically display the meta-properties of objects, such as functions, models, and parameters. For this reason, all properties that require user input are defined as meta-properties, as described shortly. The inheritance from QObject also makes it possible to communicate between objects by means of “signals and slots;” this Qt-feature is utilized in the user interface to interpret the actions taken by the user, such as button clicks in dialog boxes.

2.3. USER INTERFACE

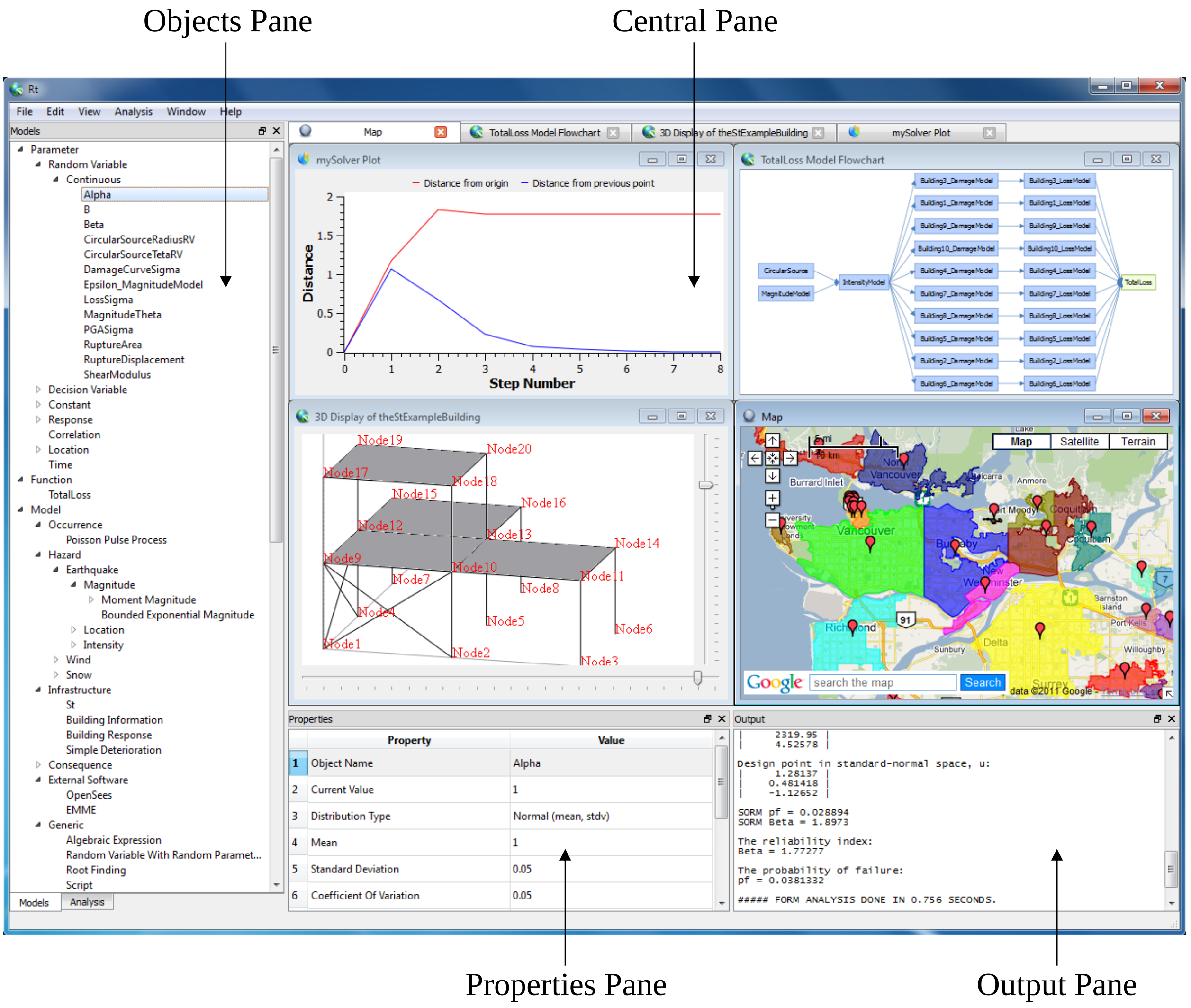
An important objective in this chapter is to offer a user interface that makes multi-model reliability and optimization analyses accessible and understandable to a broad engineering audience. It is also important that the user interface offers the flexibility and simplicity to accommodate future developments, such as new models and new analysis options without the need to change the user experience or existing code. One testament to the success of Rt in meeting these objectives is the linear structural analysis program St, which is also freely available at www.inrisk.ubc.ca. St was developed with little effort by adopting the user interface of Rt, followed by the implementation of structural analysis options.

The user interface of Rt reflects the object-oriented architecture of Rt in two ways. First, it serves as a playground in which the user creates, views, modifies, and deletes objects. These objects are functions, models, parameters, and analysis tools. Second, the elements of the user interface, *i.e.*, the different areas of the playground, are also objects. These two ways of looking at the user interface are described in the following.

Figure 2-2 shows a screenshot of the user interface of Rt. The picture is taken on a computer running the Windows operating system. The appearance is similar on other platforms, like Mac OS X, but is native to the operating system. The interface has four "panes" that are identified in the figure. The shown arrangement of the panes is the default view, but it can be customized by the user and stored for subsequent sessions. The following features of each pane are emphasized:

- The Objects Pane shows a tree-view of the objects that the user has created. The objects are created either by loading an input file or by right-clicking in this pane. Right-clicking also provides features such as plotting probability distributions and creating location objects from Google Maps®. Within the Objects Pane, two object trees are viewable: one with analysis tools and the other with functions, models, and parameters. Each tree has several branches and sub-branches that categorizes and sorts the objects.
- The Properties Pane displays the properties of the object that is selected by the user in the Objects Pane. In this pane, the user can modify any property of the object, such as the mean of a random variable. It is even possible to modify analysis parameters at run-time, which immediately affects the ongoing analysis.
- The Output Pane displays messages from the analysis algorithms and from the models as they are being evaluated.

- The Central Pane is usually the largest area because it visualizes a variety of analysis-dependent results, implemented as described in the following.



Properties

	Property	Value
1	Object Name	Alpha
2	Current Value	1
3	Distribution Type	Normal (mean, stdv)
4	Mean	1
5	Standard Deviation	0.05
6	Coefficient Of Variation	0.05

Output

```
|      2319.95 |
|      4.52578 |

Design point in standard-normal space, u:
|      1.28137 |
|     0.481418 |
|     -1.12652 |

SORM pf = 0.028894
SORM Beta = 1.8973

The reliability index:
Beta = 1.77277

The probability of failure:
pf = 0.0381332

##### FORM ANALYSIS DONE IN 0.756 SECONDS.
```

Figure 2-2: A screenshot of the user interface of Rt.

The implementation of the user interface of Rt consists of several classes. The orchestrating class is RMainWindow, which is instantiated in the `main()` method, *i.e.*, once the user starts the program. Figure 2-3 shows the composition of classes that are managed by RMainWindow. Dashed lines in the figure identify the following three groups of classes, which are ordered from left to right in the figure:

- The four panes of the user interface, including the QMdiArea that displays visualizations in the Central Pane, which in turn contains classes that display plots, maps, *etc*.
- The RDomain, which, as mentioned earlier, contains the functions, models, and parameters. Although it is not shown in Figure 2-3, the RDomain also manages the analysis tools. This composition structure allows any model and analysis tool to display results in the Central Pane via communication with the RMainWindow.
- The dialog boxes that trigger various analyses, such as FORM reliability analysis.

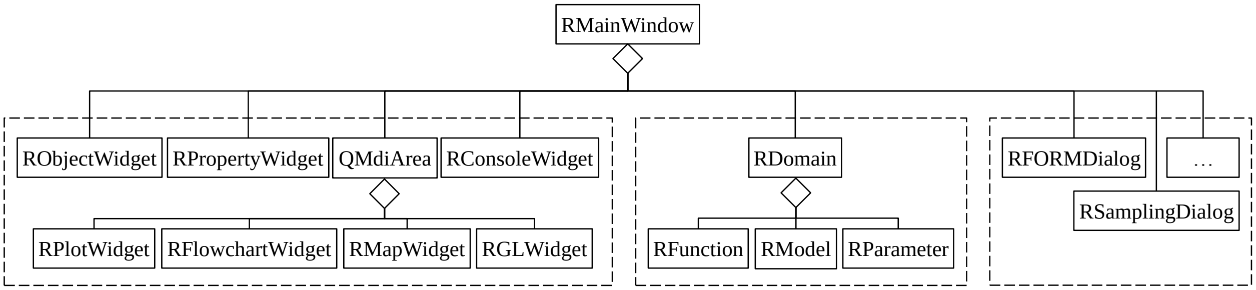


Figure 2-3: The composition of classes that are managed by RMainWindow.

The first group of classes, shown at the left in Figure 2-3 and called widgets in accordance with the naming convention in Qt, provide the functionality of the user interface by a variety of techniques. For example, the class RObjectWidget populates the tree of objects by asking the RDomain for the list of object names. When the user clicks on an object in the tree, the RObjectWidget emits a signal that contains a pointer to that object. The RPropertiesWidget receives the signal and automatically displays the meta-properties of the selected object, regardless of the object type. It is reemphasized that the meta-properties are editable at any time. This provides the user the unique opportunity to change analysis and model parameters during the analysis. For example, the user can modify the step size in an optimization search at run-time, which offers unique and powerful opportunities for experimentation. The RConsoleWidget class, which displays the messages and errors in the Output Pane, is essentially a text-viewer class. To enable all classes throughout Rt to write to the Output Pane, a message handler method is implemented that redirects all the standard output and standard error channels to the RConsoleWidget.

As shown in Figure 2-3, the QMdiArea class, which manages the Central Pane, displays multiple windows that simultaneously appear in this pane. Examples of what is visualized include diagrams that are updated as the analysis progresses. The Qwt library (Rathmann and Wilgen 2011) is employed for plotting by means of the RPlotWidget class, which has a number of functionalities, such as exporting the diagrams to pixel and vector graphics formats. Other visualization options that are implemented in Rt are Google Maps® and OpenGL (Shreiner 2010). In addition, the Qt 2D Graphics library is utilized to develop the RFlowchartWidget class that renders a flowchart of all the models in a multi-model analysis. Finally, the user interface of Rt saves and loads input files similar to most professional computer programs. The file format is standard ASCII text, which means that experienced users can easily work with the input file instead of the graphical user interface. The process of opening and saving input files in Rt is optimized for large files to facilitate large-scale multi-model analyses, which permits input files that define tens of thousands of objects.

2.4. PARAMETERIZATION

Reliability and optimization analyses act on vectors of parameters. As mentioned earlier, the parameters are either random variables or decision variables. Depending on the problem at hand, some analyses have few parameters, while others have thousands. In the type of analyses that are contemplated in this thesis, trial realizations of the parameters are repeatedly passed—as illustrated in Figure 2-1—to a domain that returns the value of function(s) and gradients. For this reason, the use of parameterization is more suitable in reliability and optimization analyses than, say, deterministic finite element analysis. Therefore, while parameterization is associated with a small added computational cost, it is worth the cost because of a drastically improved software architecture and data safety. Moreover, parameterization carries particular importance in Rt because it facilitates the communication between models that is described in the next section.

In this thesis, parameterization means that each parameter is an object, which is instantiated from a special-purpose parameter class in Rt. Conversely, in non-parameterized programs, each parameter is a basic integer or real number that is a data member of the class that needs its value. In such programs, the value is typically set by calls that go through classes that are higher in the composition hierarchy. The strategy is therefore referred to as a "top-down" approach. In contrast, the parameterization in Rt is called "bottom-up" because the class that has the parameter looks up its value and subsequently passes the computed responses up to the orchestrating algorithm. An example of the parameterization in Rt is that each continuous random variable is an instantiation of the class RContinuousRandomVariable. This class has the distribution type and distribution parameters as meta-properties. Figure 2-4 provides the inheritance tree for the parameter classes in Rt. In addition to the ones with obvious names, the RResponse is unique to Rt. This is the parameter that carries responses, as well as response sensitivities, from one model to another, or to a function.

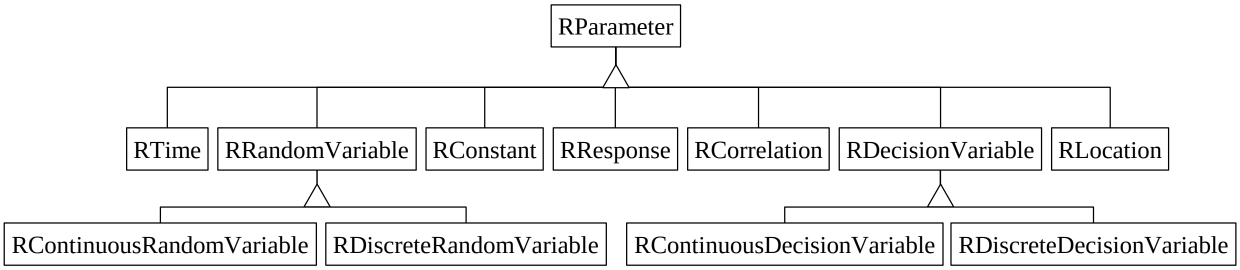


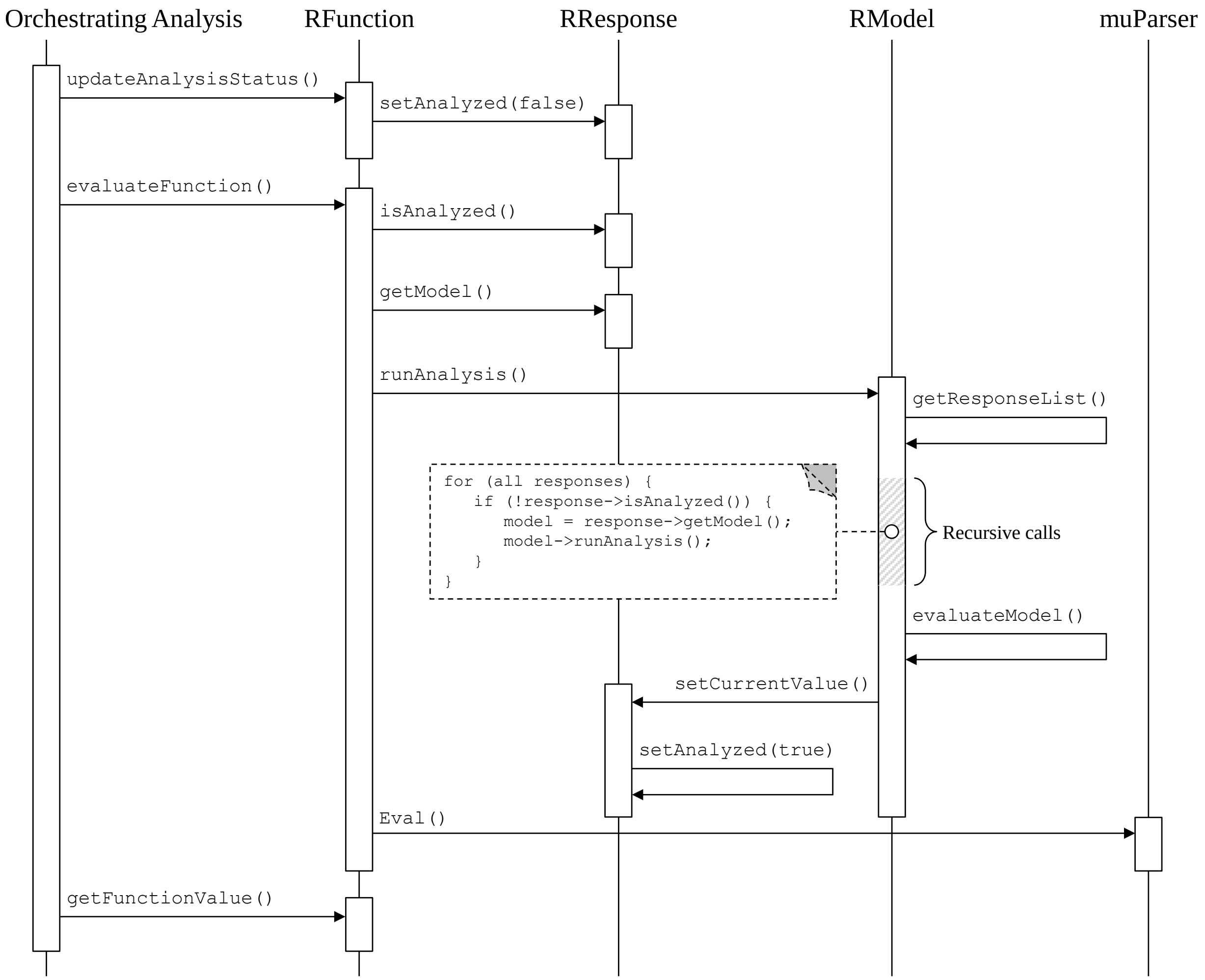
Figure 2-4: The inheritance tree for parameter classes in Rt.

The first base class in Figure 2-4, *i.e.*, RParameter, serves a particularly important role in Rt. Every model takes RParameter objects as input, rather than specific types like an RContinuousRandomVariable. This means that the models do not know whether a particular input parameter is a random variable, a decision variable, or even a response from another model. In other words, the user is free to exchange an RContinuousRandomVariable for an RResponse as input to a model. This parameterization is the first cornerstone in the functionality of the multi-model framework in Rt.

Other parameters also play significant roles, such as the unique RTime parameter, which is vital in the modeling of deterioration, discounting, and hazard occurrences. This is described in a later section. The class RLocation also serves an important purpose by providing information like the location of earthquake events and building to certain models.

2.5. MODEL COMMUNICATION

It is understood from the previous section that the class RResponse, which is a subclass of RParameter, facilitates the communication between models in Rt. To further examine this communication, this section describes the steps that are taken to evaluate a function. In other words, this section describes the workings of the right-most box in Figure 2-1. As a starting point, suppose an optimization analysis has arrived at a point where the value of some function is required. In accordance with the graphics jargon in Gamma *et al.* (Gamma *et al.* 1994), Figure 2-5 shows an interaction diagram that illustrates the sequence of calls during the evaluation of a function in the multi-model analysis in Rt. Specifically, the orchestrating analysis obtains the function value by successively calling the following three methods of the RFunction object:



```
for (all responses) {
   if (!response->isAnalyzed()) {
      model = response->getModel();
      model->runAnalysis();
   }
}
```

Figure 2-5: An interaction diagram that shows the sequence of calls during the evaluation of a function that requires multi-model analysis. Time runs downwards. The class names at the top identify the type of object that is invoking/receiving the call. Vertical rectangles indicate the time that the object is active.

- `RFunction::updateAnalysisStatus`, which tells the relevant RResponse objects that their respective models need to be re-analyzed because new parameter realizations are available. As shown in Figure 2-5, this is accomplished by setting the Boolean `isAnalyzed` flag of the RResponse objects to false. This flag is set only in those RResponse objects that are affected by the new parameter realizations.
- `RFunction::evaluateFunction`, which calls the relevant models to compute the function value, as described shortly.

- `RFunction::getFunctionValue`, which returns the function value to the reliability or optimization algorithm.

The call to `evaluateFunction` in the second item above triggers the multi-model interaction that is at the heart of the functionality of Rt. Specifically, a loop iterates through all the RResponse objects that appear in the function and makes the following calls, shown in Figure 2-5:

1) `RResponse::isAnalyzed`, which checks that the model associated with this response indeed needs to be analyzed. This may not be the case if the function evaluation is part of a finite difference sensitivity analysis, which is described in the next section.
2) `RResponse::getModel`, which provides a pointer to the model that is associated with this response. This call is made only if the associated model needs to be reanalyzed according to the previous item. Notice that this prevents one model to be analyzed several times because the `isAnalyzed` flag is set to true for all responses of a model once the model is analyzed.
3) `RModel::runAnalysis`, which tells the model from Item 2 to carry out the analysis that computes its responses. However, before executing this task, which is identified as Item b) below, a crucial step is made within `runAnalysis`, identified in Item a) below.
    a) `RModel::getResponseList` generates a list of the responses from upstream models that are input to this model. Thereafter, a loop iterates through these RResponse objects and makes the calls in Items 1, 2, and 3 in this list. This leads to a nested recursion over all models, which provides the sought linkage with the other models. The pseudo-code is shown in a dashed box in Figure 2-5. This code ensures that upstream models compute responses before downstream computations are carried out.

b) `RModel::evaluateModel` evaluates the expressions and algorithms that the model developer has implemented. All the responses are set by calls to the method `RResponse::setCurrentValue` and the `isAnalyzed` flag is set to true. In addition, this method is where the model developer implements equations related to direct differentiation sensitivity analysis, which is described in the next section.

Upon executing this nested algorithm, all responses are up-to-date and the function expression can be evaluated. As shown in Figure 2-5, this is also done within the `evaluateFunction` method by means of a call to `muParser::Eval`, where MuParser (Berg 2011) is a freeware library that is utilized in Rt to tokenize and parse mathematical expressions.

2.6. RESPONSE SENSITIVITIES

Several effective reliability and optimization methods require the derivative of the governing functions with respect to the intervening parameters. For example, several reliability methods require the derivative of the limit-state function with respect to the random variables. This information is collected in the gradient vector of a function, which essentially contains response sensitivities, *i.e.*, derivatives. There are two strategies for obtaining response sensitivities; both are outlined here and implemented in Rt.

The finite difference scheme is conceptually straightforward but associated with high computational cost and unpredictable inaccuracy. It entails repeated computation of the response with slightly perturbed values of the intervening parameters. In Rt, the finite difference computations take place in the RFunction class, in a method called `evaluateGradient`. In a loop that iterates over all parameters that affect the function, it perturbs the value of each parameter and sets the `isAnalyzed` flag to false only for those models that are affected by that parameter. This list of models, one for each parameter, is established the first time the gradient is evaluated. The information is stored in a QMap object that links each parameter to the list of models it affects. The information is used in subsequent calls to evaluate the gradient. This implementation of finite difference computations for multi-model problems in Rt is novel and particularly efficient in the presence of computationally costly models because only the required models are re-evaluated.

The other strategy for computing response sensitivities is implementing equations and algorithms that are analytical derivatives of the original response computations. This is referred to as the direct differentiation method (DDM) (Kleiber *et al.* 1997). This approach yields accurate and efficiently computed response sensitivities at the one-time cost of differentiating and implementing DDM equations. This can be done for every model in Rt, and such equations are already implemented for most models in Rt.

As a demonstration, Figure 2-6 illustrates the DDM-computations for a response, r_3, with respect to four parameters, x_1, x_2, x_3, and x_4. These parameters can be of any type, such as random variables and decision variables. Consider the response sensitivity computations in the model that is identified by a thick solid line in Figure 2-6. To illustrate a general situation, x_1 and x_2 enter into an upstream model, x_2 and x_3 enter into another upstream model, while x_3 and x_4 enter directly into the considered model. The two upstream models produce responses r_1 and r_2 that enter the computation of r_3. Mathematically, the response r_3 is written

$$r_3 = r_3\left(r_1(x_1, x_2),\ r_2(x_2, x_3),\ x_3,\ x_4\right) \tag{2-1}$$

Eq. (2-1) highlights the implicit dependence on x_1, x_2, and x_3 via the upstream models, as well as the explicit dependence on x_3 and x_4. As shown in Figure 2-6, the two upstream models also produce the response sensitivities $\partial r_1/\partial x_1$, $\partial r_1/\partial x_2$, $\partial r_2/\partial x_2$, and $\partial r_2/\partial x_3$, which are stored as data members of the RResponse objects. It is understood from the previous section that the developer implements the computation of r_3 in the method called `evaluateModel`. This is a pure virtual method of RModel, which means that it must be implemented in every subclass of RModel, *i.e.*, in every model. As illustrated in Figure 2-6, the model developer also implements DDM equations in the `evaluateModel` method. Specifically, given the expression or algorithm for the response r_3, the developer carries out the differentiation that yields $\partial r_3/\partial r_1$, $\partial r_3/\partial r_2$, $\partial r_3/\partial x_3$, $\partial r_3/\partial x_4$. The results are stored in one QMap object for every response. These maps essentially link each RParameter object with one response sensitivity value. Subsequently, the `runAnalysis` method of the RModel base class, which is shown in Figure 2-6 but hidden to the model developer, automatically computes the following derivatives that are sought by the reliability or optimization algorithm:

$$\begin{aligned}
\frac{\partial r_3}{\partial x_1} &= \frac{\partial r_3}{\partial r_1}\cdot\frac{\partial r_1}{\partial x_1} \\
\frac{\partial r_3}{\partial x_2} &= \frac{\partial r_3}{\partial r_1}\cdot\frac{\partial r_1}{\partial x_2}+\frac{\partial r_3}{\partial r_2}\cdot\frac{\partial r_2}{\partial x_2} \\
\frac{\partial r_3}{\partial x_3} &= \frac{\partial r_3}{\partial r_2}\cdot\frac{\partial r_2}{\partial x_3}+\frac{\partial r_3}{\partial x_3} \\
\frac{\partial r_3}{\partial x_4} &= \frac{\partial r_3}{\partial x_4}
\end{aligned} \tag{2-2}$$

These values are stored in a new QMap object and are accessible to downstream models because the new QMap object is a data member of the RResponse object. These implementations are believed to be the first general-purpose implementation of the DDM for this type of multi-model probabilistic analysis.

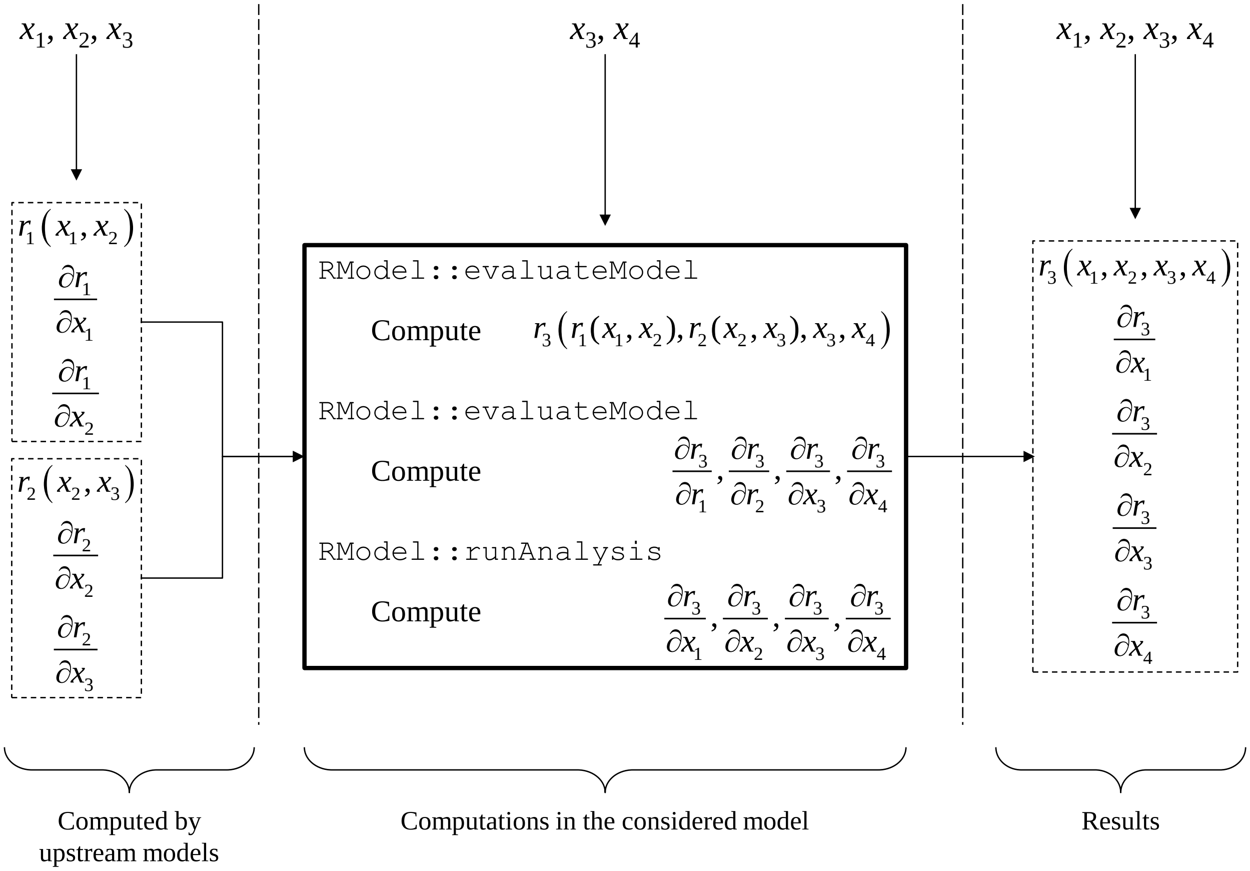

x_1, x_2, x_3
x_3, x_4
x_1, x_2, x_3, x_4
$r_1(x_1, x_2)$
$\frac{\partial r_1}{\partial x_1}$
$\frac{\partial r_1}{\partial x_2}$
$r_2(x_2, x_3)$
$\frac{\partial r_2}{\partial x_2}$
$\frac{\partial r_2}{\partial x_3}$
RModel::evaluateModel
Compute $r_3(r_1(x_1, x_2), r_2(x_2, x_3), x_3, x_4)$
RModel::evaluateModel
Compute $\frac{\partial r_3}{\partial r_1}, \frac{\partial r_3}{\partial r_2}, \frac{\partial r_3}{\partial x_3}, \frac{\partial r_3}{\partial x_4}$
RModel::runAnalysis
Compute $\frac{\partial r_3}{\partial x_1}, \frac{\partial r_3}{\partial x_2}, \frac{\partial r_3}{\partial x_3}, \frac{\partial r_3}{\partial x_4}$
$r_3(x_1, x_2, x_3, x_4)$
$\frac{\partial r_3}{\partial x_1}$
$\frac{\partial r_3}{\partial x_2}$
$\frac{\partial r_3}{\partial x_3}$
$\frac{\partial r_3}{\partial x_4}$
Computed by upstream models
Computations in the considered model
Results

Figure 2-6: Implementation of the direct differentiation method in Rt.

2.7. Time Dependent Multi-hazard Analysis

Because a stated objective in this thesis is to simulate events in the lifetime of civil infrastructure, Rt addresses the modeling of time dependent phenomena. Examples include structural deterioration, monetary discounting, and the uncertain occurrence time of many hazards. It is the RTime class that facilitates this pioneering feature. This class is utilized in three manners in Rt:

- RTime objects are set by hazard occurrence models, which provide the occurrence times of events, such as earthquakes.
- RTime objects are input to models that require knowledge of time. For example, a model that

simulates structural deterioration requires knowledge of time to predict the present amount of corrosion. Similarly, the model that discounts monetary losses to present value requires knowledge of the time at which the loss was incurred and the time to which the loss should be discounted.

- RTime objects are set by orchestrating algorithms to specify the time at which the functions are evaluated. For example, in an analysis type that is described shortly, the evaluation of the models is triggered when, say, an earthquake occurs.

In the context of reliability analysis, time dependent analysis must be carefully defined. In the literature, the phrase time-variant reliability identifies problems in which the time of failure is uncertain. *I.e.*, the time of violation of a prescribed limit-state is uncertain. A significant body of literature exists, with a variety of time-variant reliability methods. One fundamental approach is to sample the governing functions at selected time instants. Obviously, it is computationally prohibitive to carry out an analysis at every minute or hour in a lifetime analysis to check the status of the models. Instead, an innovative version of the basic sampling approach is implemented in Rt. It is called “scenario sampling” and accommodates the aforementioned applications of RTime objects. The scenario sampling algorithm takes any number of designated occurrence models as input. Occurrence models are the most upstream models and they trigger events during the user-given time period. As an illustration, consider the analysis example that is provided later in this chapter, where occurrence models are provided for earthquakes and windstorms. The scenario-sampling analysis simulates the occurrence of these hazards, as well as downstream damage and impacts, for the prescribed 50-year time-period. The number of samples, *i.e.*, the number of times that the time-period is analyzed, determines the accuracy of the probability results.

The evaluation of functions, *i.e.*, the evaluation of the models, is triggered whenever a significant event occurs. The occurrence model determines the duration of each hazard occurrence, such as the duration of the wind load. The functions can record the responses cumulatively, or keep only the final value at the end of the time period. It is also possible to record the minimum or the maximum response over the time-period.

In this type of analysis, the realization of some random variables, such as the strength and stiffness of a structural component and a model uncertainty term, is re-sampled for every time-period, but remains constant within each time-period. Other random variables that characterize the severity and location of hazard occurrences, such as earthquake magnitude, take on new realizations during the time-period.

Several other analysis options are implemented in Rt, including another multi-hazard analysis option that is based on the load coincidence method (Wen 1990). That option is more computationally efficient than scenario sampling. However, it limits the inclusion of time-dependent phenomena. In this analysis option, a separate reliability analysis is carried out for each individual hazard as well as each possible combination of hazards. The occurrence time of hazards is not modeled in these analyses and therefore, they do not make use of the RTime object. This is why this analysis restricts the inclusion of time-dependent phenomena. For instance, while this analysis takes into account the occurrence and coincidence of multiple hazards, it is incapable of accounting for log-term deterioration and discounting.

2.8. Analysis Example

The following example is intended to demonstrate multi-model analysis in Rt. It is also intended to expose some of the models that are already available in the growing library of models in Rt. To this end, suppose an engineer is asked to estimate the probability of events that may occur in the 50-year lifespan of a new reinforced concrete structure in Whistler, Canada. The structural model is shown in Figure 2-7. Three hazards are considered in this illustration: snowstorms, windstorms, and earthquakes from the subduction source outside the coast of British Columbia. Figure 2-8 presents an overview of the models that are utilized to solve the problem. The figure identifies each model by a number and key facts about each model are provided in the following. To complement these explanations, Table 2-1 provides details about the input parameters.

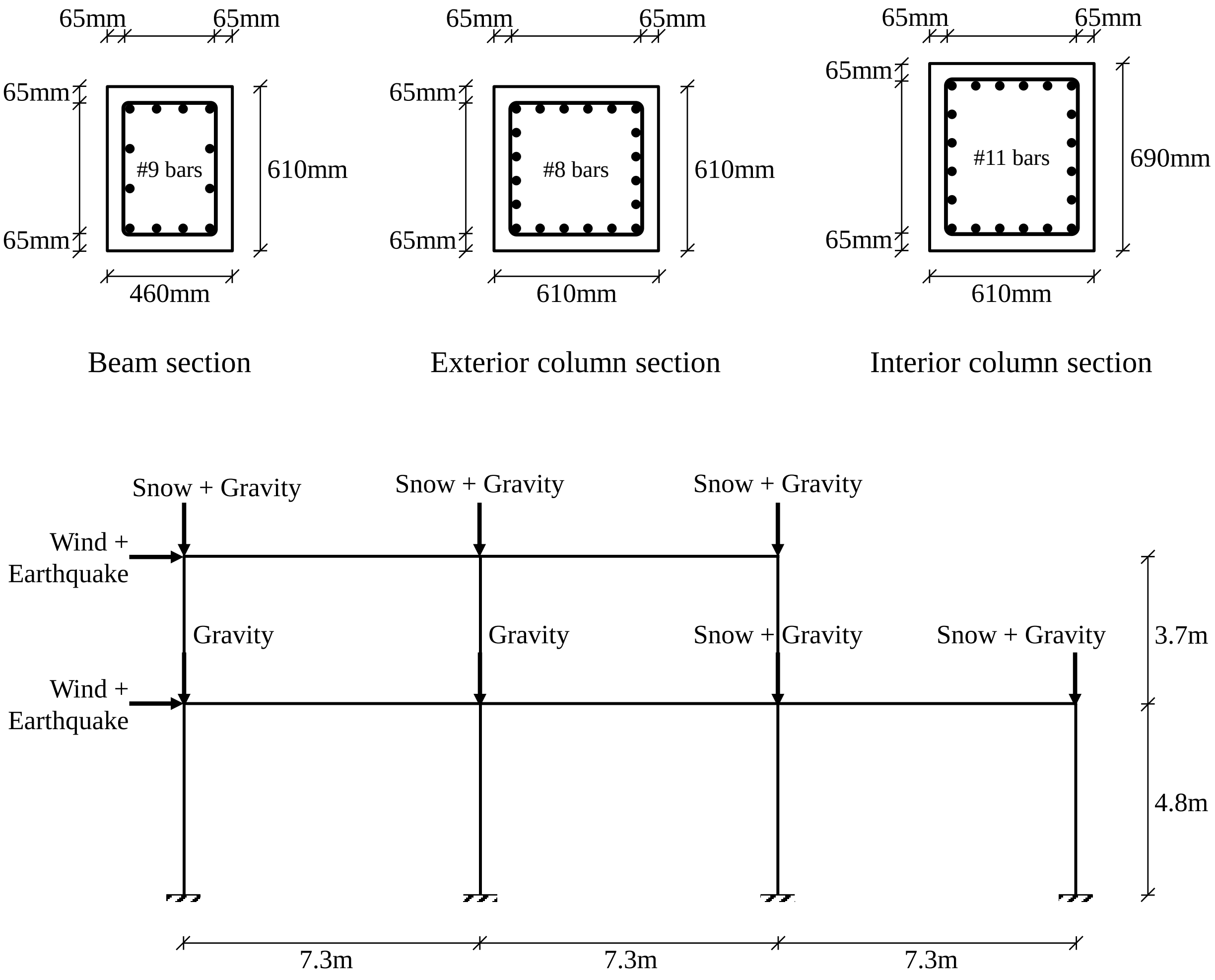


Figure 2-7: Structural model.

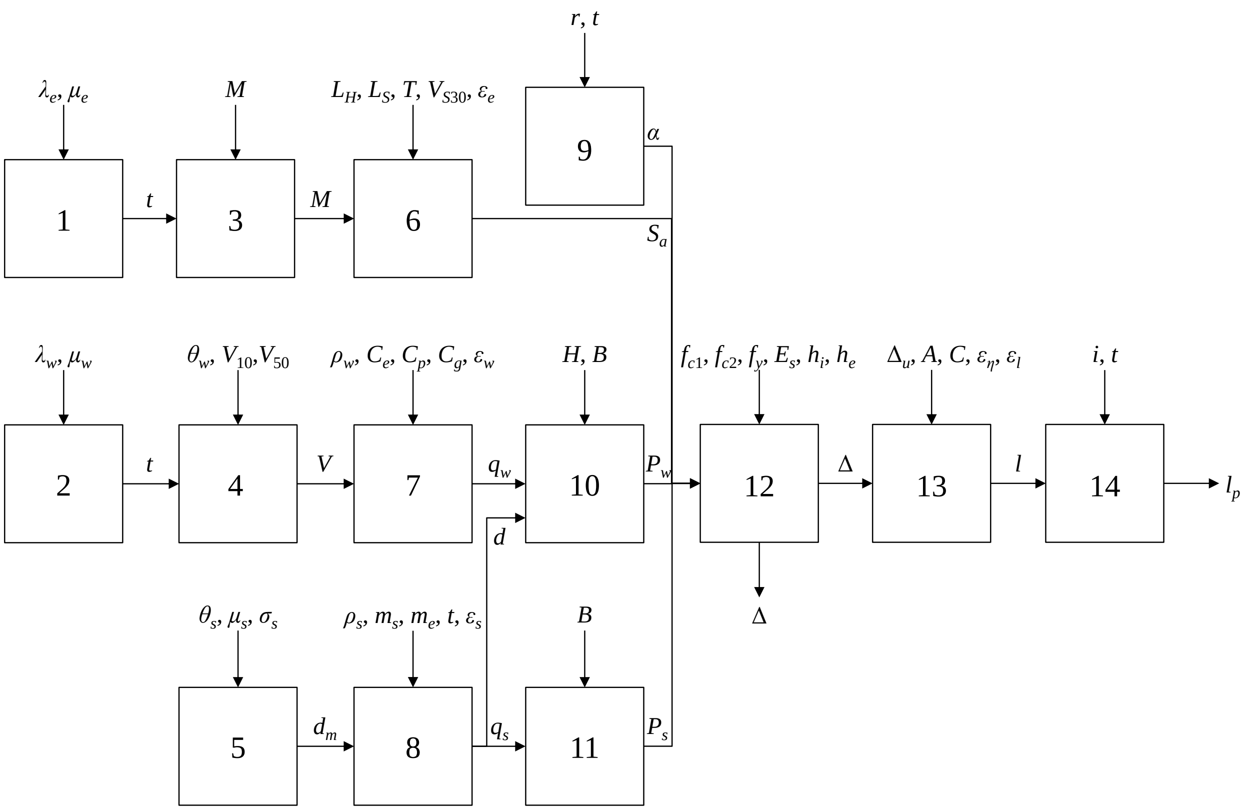

r, t
λ_e, μ_e
M
$L_H, L_S, T, V_{S30}, \varepsilon_e$
9
α
1
t
3
M
6
S_a
λ_w, μ_w
θ_w, V_{10}, V_{50}
$\rho_w, C_e, C_p, C_g, \varepsilon_w$
H, B
$f_{c1}, f_{c2}, f_y, E_s, h_i, h_e$
$\Delta_u, A, C, \varepsilon_\eta, \varepsilon_l$
i, t
2
t
4
V
7
q_w
10
P_w
12
Δ
13
l
14
l_p
d
Δ
$\theta_s, \mu_s, \sigma_s$
$\rho_s, m_s, m_e, t, \varepsilon_s$
B
5
d_m
8
q_s
11
P_s

Figure 2-8: Overview of the models in the analysis example. The symbols are explained in Table 2-1.

Table 2-1: Parameters in the analysis example. $N(\mu, \sigma)$ and $LN(\mu, \sigma)$ denote the normal and lognormal distributions, respectively, with mean μ and standard deviation σ.

Symbol	Description	Class/Type	Characteristics/Value	Unit
A	Total building area	RConstant	800	m^2
B	Bay width	RConstant	7.3	m
C	Replacement cost per unit area	RRandomVariable	*LN*(2200, 220)	$\$/m^2$
C_e	Wind exposure factor	RRandomVariable	*LN*(0.93, 0.12)	-
C_q	Wind gust factor	RRandomVariable	*LN*(0.97, 0.09)	-
C_p	Wind external pressure factor	RRandomVariable	*LN*(0.88, 0.06)	-
d	Snow depth	RResponse	-	m
d_m	Maximum snow depth	RResponse	-	m
E_s	Steel Young’s modulus	RRandomVariable	*LN*(200000,8000)	N/mm^2
f_{c1}	Cover concrete strength	RRandomVariable	*LN*(16, 0.64)	N/mm^2
f_{c2}	Core concrete strength	RRandomVariable	*LN*(22, 0.88)	N/mm^2
f_y	Steel yield stress	RRandomVariable	*LN*(300, 12)	N/mm^2
H	Building height	RConstant	8.3	m
h_e	Exterior columns width	RDecisionVariable	610	mm
h_i	Interior columns width	RDecisionVariable	690	mm
i	Effective interest rate	RRandomVariable	*N*(0.03, 0.003)	1/year
l	Loss	RResponse	-	$
L_H	Earthquake source location*	RLocation	48° 11′ 13″ N, 124° 19′ 51″ W	°
l_p	Loss discounted to present value	RResponse	-	$
L_S	Building location	RLocation	50° 7′ 0″ N, 122° 57′ 40″ W	°
M	Earthquake magnitude	RRandomVariable	User-defined distribution	-
m_e	End month of the snow season	Real number	5	-
m_s	Start month of the snow season	Real number	10	-
P_s	Snow load	RResponse	-	N
P_w	Wind load	RResponse	-	N
q_s	Distributed snow load	RResponse	-	N/m^2
q_w	Wind pressure	RResponse	-	N/m^2
r	Deterioration rate	RRandomVariable	*LN*(0.014, 0.0007)	1/year
S_a	Spectral acceleration	RResponse	-	g
T	Natural period of the first mode	RConstant	0.59	s
t	Time	RTime	-	day
V	Wind velocity	RResponse	-	m/s
V_{10}	10-year wind velocity	RConstant	24.25	m/s
V_{50}	50-year wind velocity	RConstant	29.43	m/s
V_{S30}	Shear wave velocity	RRandomVariable	*LN*(900, 90)	m/s
α	Deterioration factor	RResponse	-	-
Δ	Roof horizontal displacement	RResponse	-	mm
Δ_u	Ultimate displacement capacity	RRandomVariable	*LN*(332, 33.2)	mm
ε_e	Attenuation relationship model error	RRandomVariable	*N*(0, 1)	-
ε_l	Loss model error	RRandomVariable	*N* (1, 0.1)	-
ε_s	Snow load model error	RRandomVariable	*N*(1, 0.1)	-
ε_w	Wind pressure model error	RRandomVariable	*N*(1, 0.1)	-
ε_η	Damage model error	RRandomVariable	*N*(1, 0.1)	-
θ_s	Maximum snow depth uncertainty	RRandomVariable	*N*(0, 1)	-
θ_w	Wind velocity uncertainty	RRandomVariable	*N*(0, 1)	-
λ_e	Earthquake occurrence rate	Real number	0.0037	1/year
λ_w	Windstorm occurrence rate	Real number	1	1/year
μ_e	Earthquake mean duration	Real number	180	s
μ_s	Maximum snow depth mean	RRandomVariable	*LN*(3.5, 0.7)	m
μ_w	Windstorm mean duration	Real number	7200	s
ρ_s	Snow density	RRandomVariable	*LN*(360, 30)	kg/m^3
ρ_w	Air density	RRandomVariable	*LN*(1.293, 0.065)	kg/m^3
σ_s	Max. snow depth standard deviation	RRandomVariable	*LN*(0.7, 0.14)	m

* A refined model for subduction earthquakes is presented in Chapter 4, where different location models are employed for different magnitude ranges.

Model (12) is described first, because it is the structural model. It is an ROpenSeesModel, which is an interface to the external software OpenSees mentioned earlier. The structure is a reinforced concrete moment frame shown in Figure 2-7, subjected to nonlinear static analysis. The cross-sections, which are shown at the top of Figure 2-7, are fiber-discretized with two fibers in the cover concrete and 12 and 14 fibers in the core concrete for the beams and columns, respectively. The loads are applied as lumped nodal loads, as shown in Figure 2-7. Earthquake forces are applied as equivalent static forces, computed based on the shape of the first natural mode of vibration. As illustrated in Figure 2-8, the structural model takes input from several upstream models and produces the displacement of the building at the roof level. This is done by running an OpenSees finite element analysis, executed by Tcl commands issued from Rt. It is also noted that the analysis accounts for hazard-interaction effects, including the weight of snow on the roof if an earthquake occurs, which amplifies the P-delta effect, and the increase in the surface that the wind acts over when snow is present. The latter effect occurs when the snow accumulates on the roof and effectively makes the building taller, which increases the area that the wind acts on. These effects are included to emphasize the flexibility of the multi-model framework in Rt.

Models (1) and (2) in Figure 2-8 are RPoissonPulseProcessModels, which simulate the occurrence of earthquakes and windstorms, respectively, and produce realization of time instants. Model (3) is an RAlgebraicExpressionModel that models the magnitude of the earthquakes. Although the magnitude is essentially modeled by a random variable with probability distribution provided by Petersen *et al.* (2008), it is given as an algebraic expression model because scenario sampling requires hazard severities to be given as a model rather than as a random variable. Model (4) is an RSimpleWindVelocityModel, which produces a velocity for each windstorm according to the Gumbel distribution. The distribution parameters are based on characteristic wind velocities for Whistler. Model (5) is an RRandomVariableWithRandomParametersModel, which produces realizations of the maximum seasonal snow depth based on the lognormal distribution with random mean and standard deviation, which both have the lognormal distribution. Model (6) is an RAtkinsonBoore2003IntensityModel, which produces the site-specific spectral acceleration at the natural period of the building by means of the attenuation equation provided by Atkinson and Boore (2003). Model (7) is an RSimpleWindPressureModel, which takes the wind velocity as input and computes the wind pressure on the building. Model (8) is an RSimpleSnowLoadModel that takes the time-of-year and the maximum seasonal snow depth as input and produces the snow load intensity per unit area as output. Model (9) is an RSimpleDeteriorationModel, which produces a time-dependent corrosion factor that reduces the reinforcing bar area in the structural model. Model (10) is an RAlgebraicExpressionModel, which takes the structural geometry, including the snow depth, and produces the total wind load on the building. Model (11) is an RAlgebraicExpressionModel, which takes the structural geometry as input and produces the total snow load. Model (13) is an RScriptModel, which produces the damage ratio and the cost of repair of damage. The RScriptModel is versatile because it facilitates the implementation of complex models by writing JavaScript code without the need to recompile Rt. All standard constructs of structured programming, including if, for, and while blocks, are available and a host of debugging tools are included. Model (14) is an RDiscountingModel, which takes future losses, as well as the effective interest rate and an RTime object to produce loss values that are discounted to present time.

Reliability analysis is explored in this example, and two responses are considered: Δ=horizontal displacement at the roof and l_p=present value of the total losses incurred during the lifespan of the building. For illustration purposes, consider first the occurrence of an earthquake when no other hazard acts and no deterioration is present. In that situation, the first-order second-moment (FOSM) analysis option in Rt provides the following means and standard deviations: μ_Δ=25.6mm, σ_Δ=22.7mm, μ_{lp}=\$25,680, and σ_{lp}=\$46,110. The FOSM analysis option in Rt can also be employed to carry out a FOSM reliability analysis (Melchers 1999), which produces a reliability index. It is interesting to compare this with Rt's Histogram Sampling option for this case, which yields the results in Figure 2-9 for the response Δ. It is observed that the first-order approximation of the mean is slightly higher than the peak of the normalized histogram shown by a thin black line. However, the more accurate mean computed from the histogram is higher, namely 34.5mm, because of the long probability tail. The standard deviation is also higher than the first-order approximation; the sampling yields 34.1mm. However, it is noticed that Figure 2-9 is far more informative than only second-moment information. The bump in the tail of the thin black line in Figure 2-9 is caused by a magnitude saturation term in the attenuation relationship in Model 6, which effectively lumps probability density at this location.

FORM analysis is then carried out to investigate the probability that the roof displacement exceeds 83mm, which corresponds to a global drift ratio of 1%. The result is 11.5%, which implies a reliability index equal to 1.20 associated with this limit-state. Interestingly, the FORM analysis also reveals that the model error in the attenuation relationship in Model (6), ε_e, is the most important random variable, rather than the magnitude of the earthquake. Rt makes it straightforward to carry out importance sampling around the "design point" from the FORM analysis. The result is a probability equal to 8.9%, corresponding to a reliability index equal to 1.35, which exposes a mild nonlinearity in the limit-state function. It is possible in Rt to visualize how the limit-state function varies due to variation in the value of the random variables. The resulting plots are helpful to identify nonlinearity, and examples are provided later in the thesis.

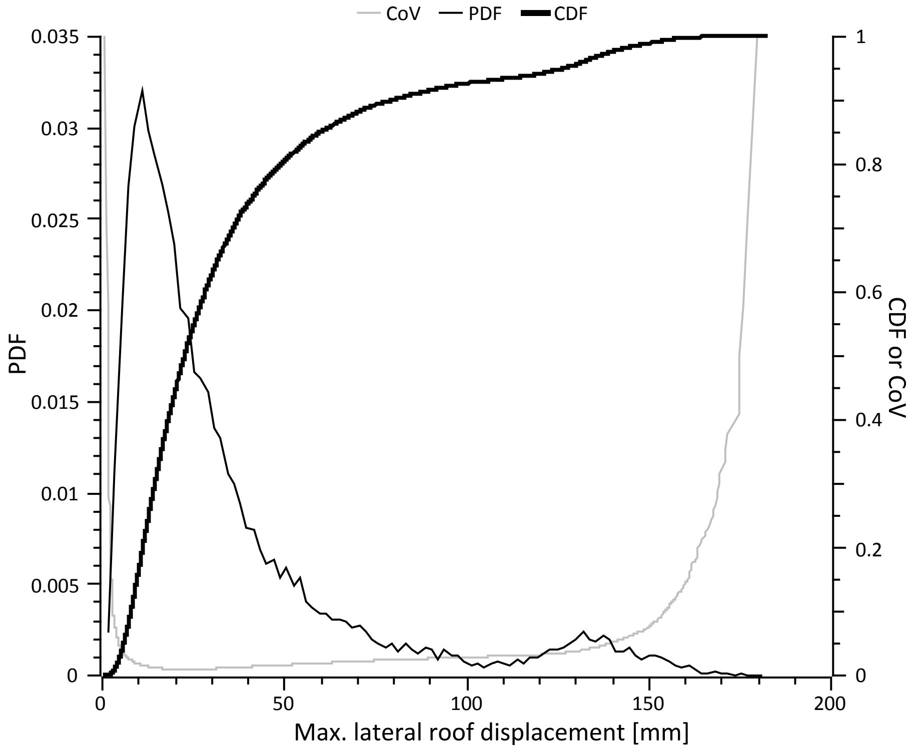


Figure 2-9: Histogram from Rt for an earthquake occurrence. The function value is the roof horizontal displacement in mm. The thin black line is comparable to a probability density function (PDF). The thick black line is comparable to a cumulative distribution function (CDF). The gray line shows the coefficient of variation (CoV) of the probability displayed by the CDF.

The previous paragraphs provide results given the occurrence of an earthquake. Now, consider a 50-year time period starting on January 1, 2012. The number of occurrences of earthquakes and other hazards in this time period is uncertain. This type of situation is addressed by scenario sampling analysis. Each sample in this analysis is a 50-year scenario. The losses are accumulated at the end of each sample. The resulting loss histogram is shown in Figure 2-10. The mean loss, μ_{lp}, is \$61,555 and the standard deviation, σ_{lp}=\$71,818. *I.e.*, the loss is highly uncertain. This is reflected in the coefficient of variation of the loss, σ_{lp}/μ_{lp}, which is greater than 100%. To further demonstrate what can be learned from the results, it is observed in Figure 2-10 that most of the probability density is concentrated below $200,000. In fact, the histogram in Figure 2-10 that Rt produces in this case shows that there is only a 1.6% probability that the loss exceeds $300,000. This result was obtained by 15,000 samples, *i.e.*, 15,000 50-year scenarios. The gray line in Figure 2-10 shows that the probability that the loss exceeds $300,000 is associated with a 6.3% coefficient of variation.

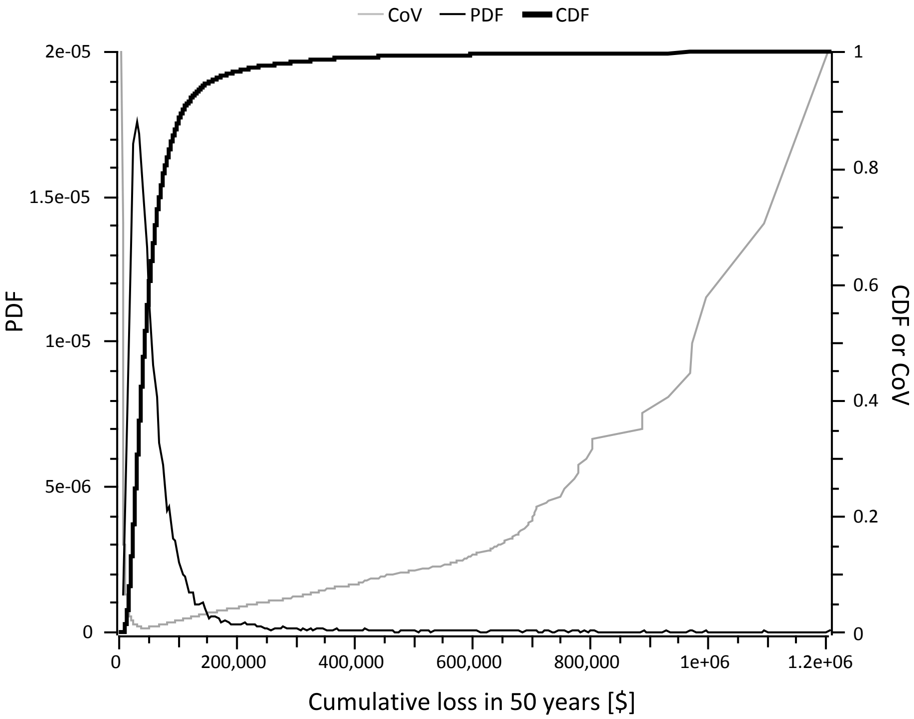


Figure 2-10: Loss histogram from Rt for a time period of 50 years.

Two decision variables were defined for this analysis example: the width of the cross-section of the interior columns, and that of the exterior columns. It was observed that the reliability results are most sensitive to the width of the interior columns. In other words, a decision to modify the dimension of the interior columns is most effective in reducing the probability of failure. Chapter 5 will present an in-depth study on decision sensitivity analysis together with a comprehensive example application.

2.9. CONCLUSIONS

The computer program Rt is a new type of software for reliability and optimization analysis. By its focus on multi-model analysis, it addresses the need for tools to carry out modern reliability and optimization analyses. This type of analyses entails the repeated evaluation of functions that predict quantities, such as structural responses, damage, and even direct and indirect monetary losses. A variety of probabilistic models are required to make the predictions, particularly in situations with multiple hazards. Rt addresses this challenge by coordinating a library of probabilistic models. The models in the library range from algebraic expressions to complex numerical simulation algorithms. A particular objective in this chapter is to expose the unique object-oriented software architecture, which includes: a comprehensive parameterization; effective and flexible communication within the framework of models; computation of response sensitivities by the direct differentiation method; and implementation of a Time parameter that facilitates the modeling of time-varying phenomena. The user is provided with exceptional flexibility in implementing *ad hoc* models, including the possibility of creating models in third-party software that are connected to Rt by means of novel interface models, which was demonstrated in the analysis example. In conclusion, Rt is envisioned as a tool to provide probabilistic predictions of the actual performance of civil infrastructure during their lifespan, including a broad range of impacts.

Chapter 3. SEISMIC RISK ANALYSIS WITH RELIABILITY METHODS: MODELS

3.1. INTRODUCTION

The overarching objective in this chapter is to improve the prediction of seismic risk for civil infrastructure. Risk in this context refers to loss probabilities, where the losses are due to repair of damaged structural and non-structural components. Loss probabilities are often presented in the form of a "loss curve," which displays the probability of exceeding any loss value. Loss curves are commonly employed on a regional scale in the insurance industry (Grossi and Patel 2005) and for individual buildings in modern performance-based engineering (Yang *et al.* 2009). However, the definition of risk in terms of loss curves contrasts the classical definition from the field of structural reliability. Risk is there defined as expected loss, *i.e.*, the product of probability and cost of a predefined failure event. In classical reliability analysis, the failure event is specified by a limit-state function, the uncertainty is characterized by random variables, and reliability methods are employed to estimate the failure probability (Ditlevsen and Madsen 1996). This approach has mostly been applied to capacity-demand-type limit-state functions. However, two observations are made: 1) Reliability methods are tailored to estimate the probability of rare events, *i.e.*, small probabilities; 2) The tail of the loss curve, *i.e.*, where the probability is low but the loss is high, is particularly important for seismic mitigation decisions. These observations motivate the present effort to explore the use of reliability methods in risk analysis and loss estimation, with emphasis on accuracy in the tail of the loss curve.

New probabilistic models are presented in this chapter, while analysis methods, loss curves, and other results for the Vancouver metropolitan region in Canada are presented in the next chapter. Although the details of the analysis are presented in the next chapter, it is immediately clear that the use of reliability methods influences the form of the sought models. To appreciate this fact, consider the key steps in a reliability analysis: Trial realizations of the random variables are repeatedly generated and each realization is employed to compute the corresponding loss, which is returned to the reliability algorithm. After several trial realizations—the exact number depends on the efficiency of the reliability method—an estimate of the loss probability is obtained. In short, reliability methods require models that take realizations of random variables as input and return the value of one or more responses, which are not probabilities. This contrasts with contemporary approaches that employ conditional probability models. To this end, the following "rules" are identified for each probabilistic model intended for use in reliability analysis:

Rule 1: It is an equation or algorithm

Rule 2: It simulates possible scenarios of physical phenomena without conservative bias

Rule 3: It discretizes all uncertainty in terms of random variables

Rule 4: It takes as input the realization of continuous random variables

Rule 5: Ideally, it has random model parameters whose probability distribution is updated when new data emerge; *i.e.*, it includes epistemic uncertainty

Rule 6: It may take constants and decision variables as input; decision variables are not uncertain, rather, their value is at the discretion of the engineer

Rule 7: It returns one or more physical responses, not probabilities

Rule 8: It returns a unique response for each unique realization of the random variables

Rule 9: It returns a response that is "continuously differentiable," *i.e.*, smooth, with respect to the random variables; this is necessary for gradient-based reliability methods, such as FORM

Rule 10: It permits the simulation of all possible realizations of the outcome space

Rule 11: It is modularized in the sense that it can take input from "upstream" models and/or return output to "downstream" models

Some models in the literature conform to these definitions and are directly amenable to reliability analysis. One example is the following ground shaking intensity model, which predicts the value of the site-specific acceleration response spectrum, *Sa*:

$$\ln(Sa) = h(m, R, V_{S30}, T_n) + \varepsilon \tag{3-1}$$

where *h*=function or algorithm, *m*=earthquake moment magnitude, *R*=distance, V_{S30}=average shear-wave velocity of the top 30m of the ground at the site, T_n=first natural period of vibration of the structure, and ε=model uncertainty. Intensity models of the format in Eq. (3-1) include those presented by Atkinson and Boore (2003) and Boore and Atkinson (2008), which are employed later in this study. Notably, these intensity models yield a unique realization of *Sa* for given realizations of *m*, *R*, V_{S30}, and T_n. In a reliability analysis, the parameters *m*, *R*, and V_{S30} are typically random variables. Thus, this model satisfies all 11 rules, except Rule 5 if the regression parameters in the attenuation relationship are constants instead of random variables.

A contrasting example of models that are intractable in reliability analysis is the damage fragility model for a concrete shear wall high-rise building in Figure 3-1 presented by FEMA-NIBS (2003). The solid thick line in Figure 3-1 is the probability that the damage to the building exceeds Damage State 3 for a given value of the maximum lateral displacement, Δ. The probability that the building is in Damage State 3 is the distance between the solid and dashed thick curves. This model is here referred to as a conditional probability model because it has a probability as output. Thus, it violates Rule 7 and cannot be directly employed in a reliability analysis. However, this model is employed later to calibrate models that obey this rule. As it stands, the model in Figure 3-1 limits the analysis formats to analytical integration by the total probability theorem or sampling. It is also noted that with models like the one in Figure 3-1, each building configuration requires a unique model. In contrast, this study advocates the development of probabilistic models that take an array of material, geometry, and model parameters as input, thus covering a range of configurations.

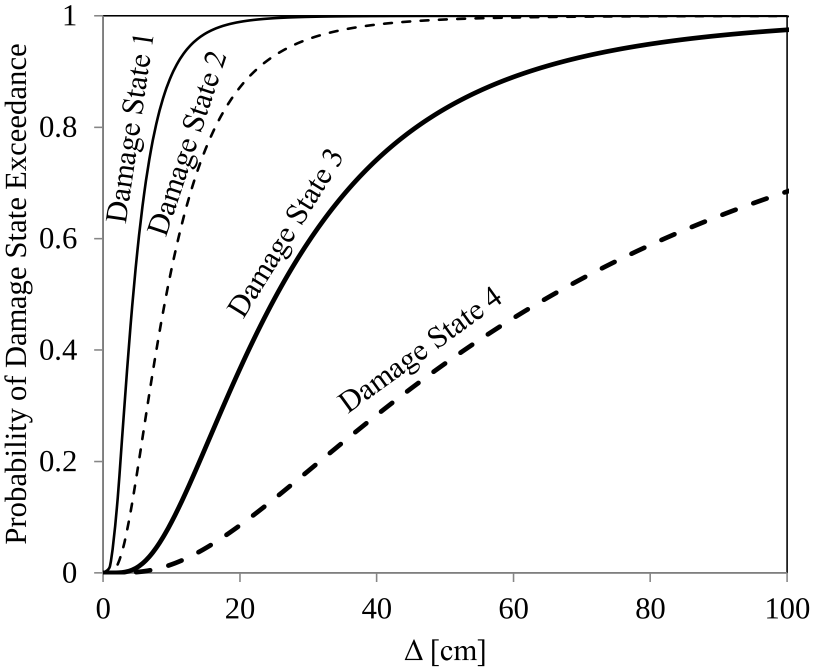


Figure 3-1: Damage fragility model for a concrete shear wall high-rise building from FEMA-NIBS (2003).

Several modeling techniques are available to develop probabilistic models suitable for reliability analysis. One approach is the development of linear models by Bayesian inference, as described by Box and Tiao (1992). Gardoni *et al.* (2002) applied this approach to reinforced concrete members in the context of seismic risk. This approach is appealing because the resulting models conform to all the abovementioned rules. In fact, model uncertainty is explicitly included by means of random model parameters in accordance with Rule 5. The extension to nonlinear models, *i.e.*, models that are nonlinear in terms of the model parameters, is described by Seber and Wild (2003).

The models presented in this chapter are implemented in the computer program Rt. The software architecture to support this type of analysis was presented in the previous chapter, but the first comprehensive utilization of Rt is presented here.

An overview of the models addressed in this chapter is provided in Figure 3-2. Each of the following sections presents one of the models, or collection of models, identified in Figure 3-2. The exception is the occurrence model, which is not described because the well-known Poisson point process is employed without modifications. Conversely, a model that discounts future losses to present value is not shown in Figure 3-2, but it is briefly described in a section below. It is also emphasized that several other models are available in Rt, while the scope of Figure 3-2 and this chapter is limited to models pertaining to regional seismic risk. Figure 3-2 emphasizes that the buildings are modeled at three different levels in Rt, *i.e.*, region, building, and component levels. A regional model covers many buildings and provides rough estimates compared with building models, which utilize characteristics such as building height and structural system to estimate response, damage, and loss. Detailed analysis at the component level, *i.e.*, finite element analysis with element models for each structural and non-structural member is not considered in this region-oriented study.

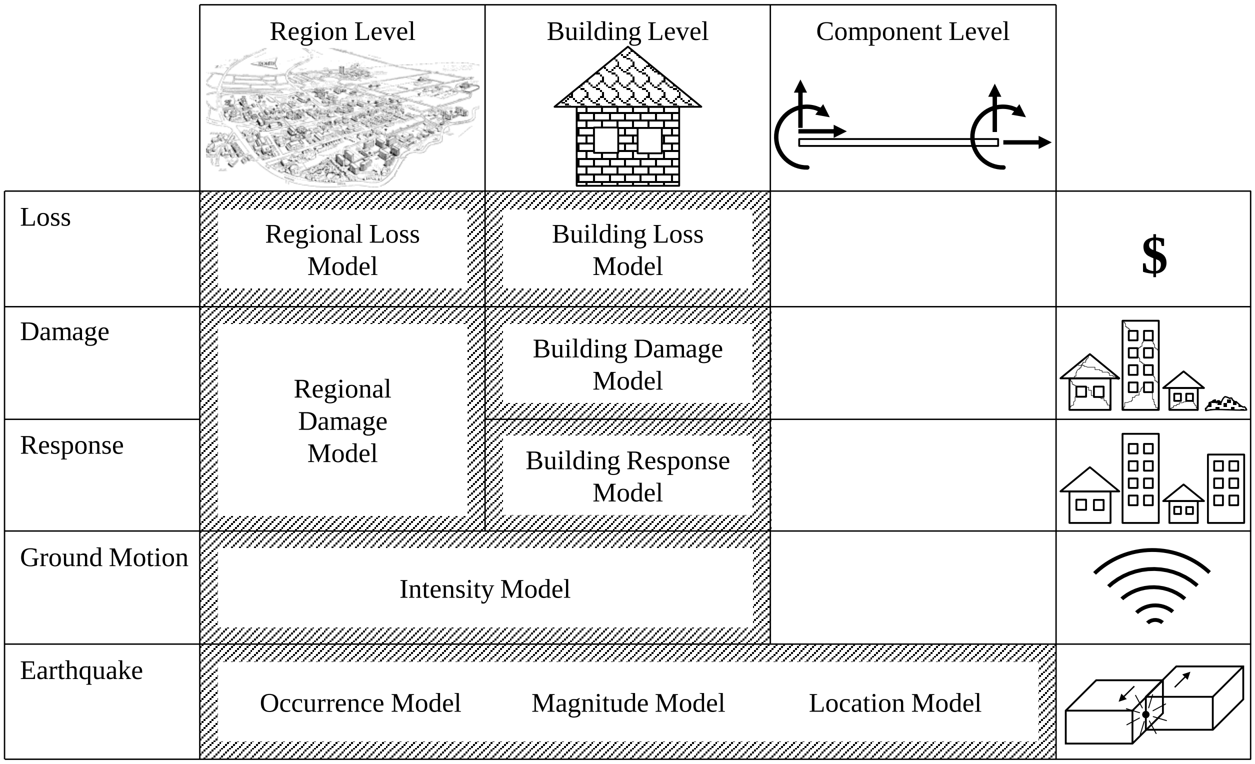


Figure 3-2: Overview of models at the region, building, and component levels.

3.2. Location Models

Contemporary risk analysis approaches often take a “hazard curve” as a starting point; see, *e.g.*, McGuire (2004). In the context of seismic risk, a hazard curve displays the probability of exceeding values of a site-specific ground shaking intensity, such as *Sa*. This study aims at circumventing the hazard curve to expose the underlying models. This is necessary for the comprehensive and continuously improving modeling of uncertainties in seismic risk analysis. Behind each hazard curve are models for the location and magnitude of earthquakes, and for the propagation of rupture energy. This section addresses the location models, which aim at predicting the location of future earthquakes, with due consideration of uncertainties.

The literature provides probability distributions for the distance, *R*, for various source geometries (Der Kiureghian and Ang 1977). Technically, these probability distributions can be employed directly in the reliability analysis with the premise that the distance is input as a random variable. However, this approach has several disadvantages in the present regional analysis. To address all buildings in a region, different random variables with different distributions would be necessary for each site. This is cumbersome and computationally inefficient, and the stipulation of correlation between the ground motions at different sites poses an additional challenge. Therefore, an alternative is proposed here, whereby location models are developed that take random variables as input and give earthquake location as output. Each realization of the random variable(s) is associated with one earthquake location and once the coordinates are known, the distance to any and many building sites is readily computed. The proposed approach is also advantageous if aftershocks are modeled, because their locations are dependent on the location of the main shock. The dedicated location models presented here also make it easier to model non-uniform probability of occurrence within a source, although this is not relevant in this study.

Models are developed here for area sources and line sources. The line source model is straightforward. It takes two random variables as input, together with the longitude and latitude of the end points, as well as the longitude and latitude of potential intermediate points for multi-segment lines. One of the random variables represents the surface location along the line, which is usually a uniform random variable. The other represents the depth.

The formulation of the area source model is new in this chapter. It models arbitrarily shaped area sources and takes three random variables, x_1, x_2, and x_3, as input. The output is the hypocenter depth, d, and the longitude, L_o, and latitude, L_a, of the surface location. The random variable x_3 directly represents the depth, while x_1 and x_2 are uniformly distributed random variables that indirectly represent L_o and L_a. The key objective of the model is to transform realizations of x_1 and x_2 into realizations of L_o and L_a so that the probability for the surface location is uniformly distributed within the arbitrary area.

To understand how the objective is accomplished, consider Figure 3-3. The shape identified by thick solid lines is an arbitrary polygon that represents an earthquake source. The user specifies the area by defining any number of longitude-latitude pairs. This is conveniently done in Rt's map interface. Next, it is recognized that if L_o and L_a have the probability density functions (PDFs) shown in Figure 3-3 and denoted by $f(L_o)$ and $f(L_a)$, then the sought model is obtained. Specifically, if $f(L_o)$ is uniform and $f(L_a)$ is proportional to the width of the area at that latitude, then the joint PDF $f(L_o,L_a)=f(L_o)\cdot f(L_a)$ is uniform over the area. Therefore, the model transforms the uniform distributions $f(x_1)$ and $f(x_2)$ to the distributions $f(L_o)$ and $f(L_a)$, where $f(L_a)$ is automatically computed based on the width of the area source at every latitude L_a.

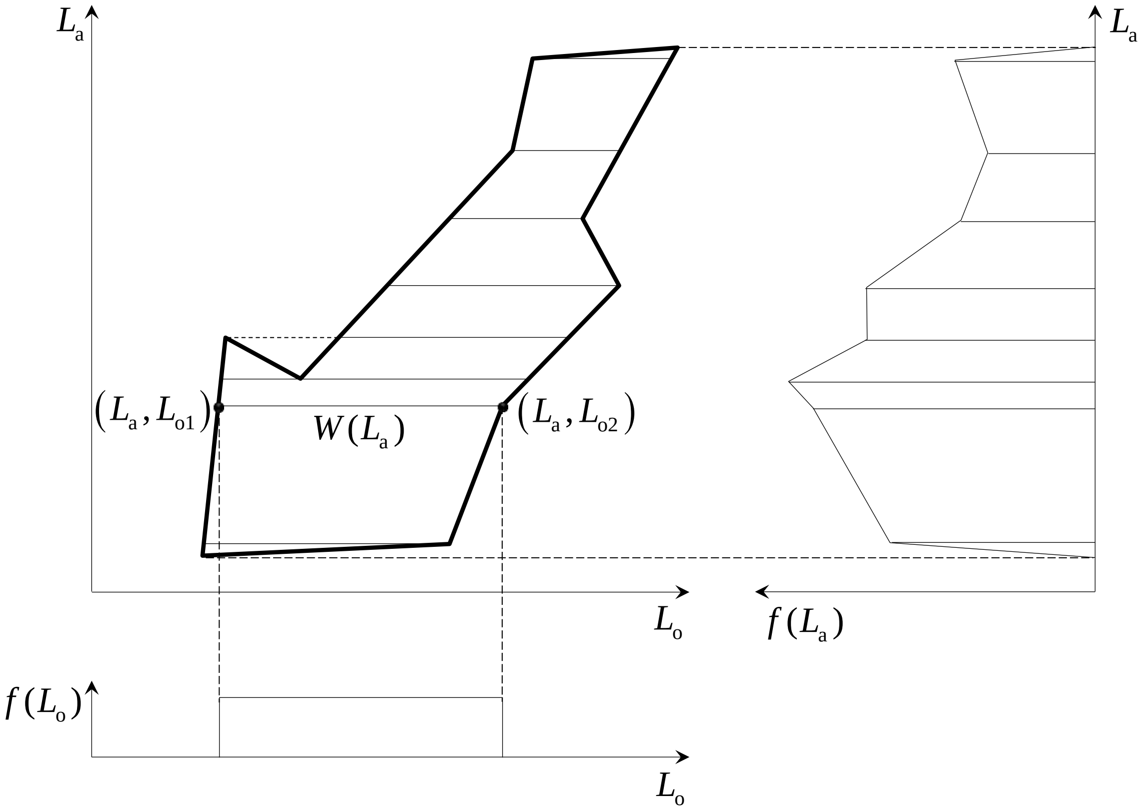


Figure 3-3: Earthquake location model for an arbitrary area source.

The width of the area source at each latitude is denoted by $W(L_a)$, as shown in Figure 3-3. Several geometrical considerations are made in the computation of $W(L_a)$. First, note that L_o and L_a are measured in radians, and that L_a is zero at the Equator. Second, let the radius of the Earth be denoted by r, under the approximation that the Earth is spherical. The radius of a circle around the Earth at a specific latitude is $r \cdot \cos(L_a)$. As a result, the length of a segment on this circle, delimited by the longitudes L_{o1} and L_{o2}, is

$$W(L_a) = \left|L_{o2} - L_{o1}\right| \cdot r \cdot \cos(L_a) \qquad (3\text{-}2)$$

Furthermore, the "great circle effect" is included when determining longitude and latitude values on the straight lines between the user-given corner points, shown by thick solid lines in Figure 3-3. This means that the slight curvature of these lines, due to the curvature of the Earth, is accounted for.

To demonstrate the effectiveness of the area source model, Figure 3-4 shows 2,000 locations generated by random samples of x_1 and x_2. An arbitrary area in the northern hemisphere is selected for this demonstration because it highlights the ability of the model to produce uniformly distributed earthquake locations. It is observed in Figure 3-4 that the locations correctly appear uniformly distributed in Google Earth®, which gives a correct spherical view of the Earth, while they appear non-uniform in Google Maps®, which gives an incorrect flattened view of the Earth. Figure 3-4 also shows that it is possible to model concave polygons with this new probabilistic model. All the location models presented here are available online in Rt. In addition, Rt includes area source models for rectangular, quadrilateral, and circular sources.

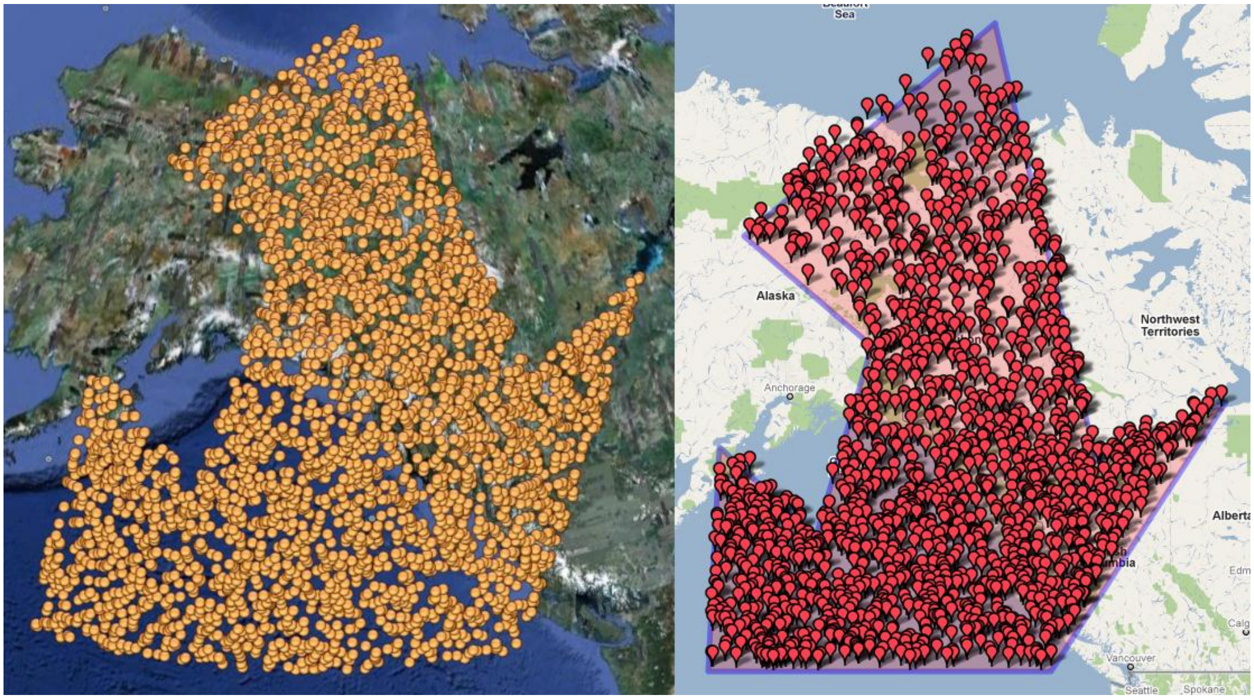


Figure 3-4: Uniformly distributed realizations for the surface location shown in the spherical Google Earth (left) and the flattened Google Maps (right). The maps in the background are from Google (© 2012 Google, © 2012 Cnes/Spot Image, Image © 2012 TerraMetrics, Image U.S. Geological Surveys).

3.3. MAGNITUDE MODEL

The magnitude of earthquakes is commonly represented by a bounded exponential random variable (McGuire 2004). The probability distribution of this random variable is based on the Gutenberg-Richter law (Gutenberg and Richter 1944) and the PDF is

$$f(m) = \frac{b' \cdot \exp\left[-b' \cdot \left(m - M_{\min}\right)\right]}{1 - \exp\left[-b' \cdot \left(M_{\max} - M_{\min}\right)\right]} \quad \text{for} \quad M_{\min} \le m \le M_{\max} \tag{3-3}$$

where m=moment magnitude, b'=parameter that depends on the relative occurrence of different magnitudes, $M_{\min}$=magnitude lower bound, and $M_{\max}$=magnitude upper bound. This study addresses the uncertainty in b' and $M_{\max}$ by modeling them as lognormal random variables. Furthermore, because Eq. (3-3) produces a probability density, thus violating Rule 7, it is necessary to transform it into a model that yields a magnitude realization as output. For this purpose, a standard normal random variable, x, is introduced as an input variable. Its cumulative distribution function (CDF) is denoted by $\Phi(x)$ and it is employed as a surrogate measure of the earthquake magnitude. Specifically, a relationship between m and x is introduced by the well-known probability-preserving transformation (Ang and Tang 2007)

$$\Phi(x) = F(m) \tag{3-4}$$

where F is the CDF that corresponds to the PDF in Eq. (3-3). As a result, magnitude realizations are produced for given realizations of x by the formula:

$$m = F^{-1}\left(\Phi(x)\right) \tag{3-5}$$

where F^{-1} is the inverse CDF of x. In summary, the magnitude model takes the realization of b', $M_{\max}$, and x as input and produces the corresponding realization of the magnitude, m, as output. This model is also available online in Rt.

3.4. GROUND SHAKING INTENSITY MODELS

A ground shaking intensity model employs characteristics of the earthquake and the path of shock wave propagation, and perhaps features of the structure, to predict characteristics of the site-specific ground shaking. A variety of models and intensity measures exist in the literature. While time-series of the ground acceleration are sought in the component-level approach in Figure 3-2, the region- and building-level analysis in this study requires models that produce a scalar intensity measure. Specifically, models are sought that take earthquake location and magnitude as input and return the elastic 5%-damped Sa for given periods T_n at specific sites. The models employed in this study are of the generic form of Eq. (3-1) and proposed by Atkinson and Boore (2003) and Boore and Atkinson (2008), with ε as a random variable in the reliability analysis. In this study, a particular effort is made to model V_{S30}. The author was given access to a comprehensive database of V_{S30}-measurements for the Vancouver metropolitan region in Canada. As a result, V_{S30} is modeled as a lognormal random variable with probability distribution inferred from the data. This will be described in more detail in Chapter 4.

The contribution to intensity modeling in this chapter is "smoothing" to satisfy Rule 9. In particular, it is required that the model response, *Sa*, is continuously differentiable with respect to *m*, *R*, and V_{S30}. The models presented by Atkinson and Boore (2003) and Boore and Atkinson (2008) violate Rule 9 due to the presence of discrete variables that depend on V_{S30} and if-statements that introduce different model forms for different ranges of V_{S30}, *m*, T_n, and depth. To make the intensity models continuously differentiable, the variation of *Sa* with respect to its parameters is given a third-order polynomial shape in the vicinity of the parameter values where the model changes form. Figure 3-5 and Figure 3-6 illustrate the effect of smoothing for the two utilized intensity models. It is arbitrarily selected to plot the model response against magnitude and shear wave velocity. As a result, the solid lines in Figure 3-5 and Figure 3-6 demonstrate that kinks and discontinuities in the original model are replaced by smooth transitions. This makes the models amenable to gradient-based reliability analysis, and the smoothed models are available online in Rt.

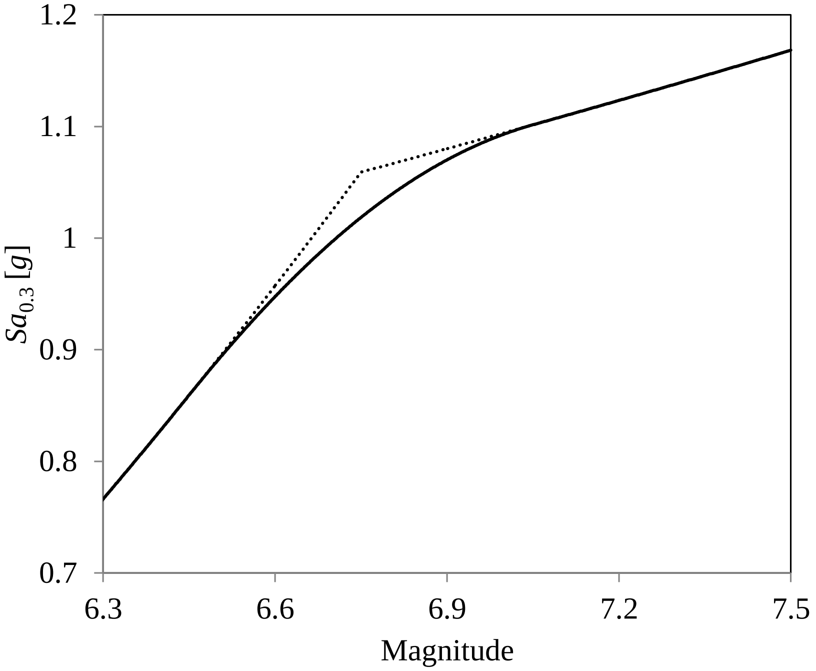


Figure 3-5: Smoothed ground motion relationship by Boore and Atkinson (2008): Spectral acceleration at 0.3 sec. plotted against the magnitude.

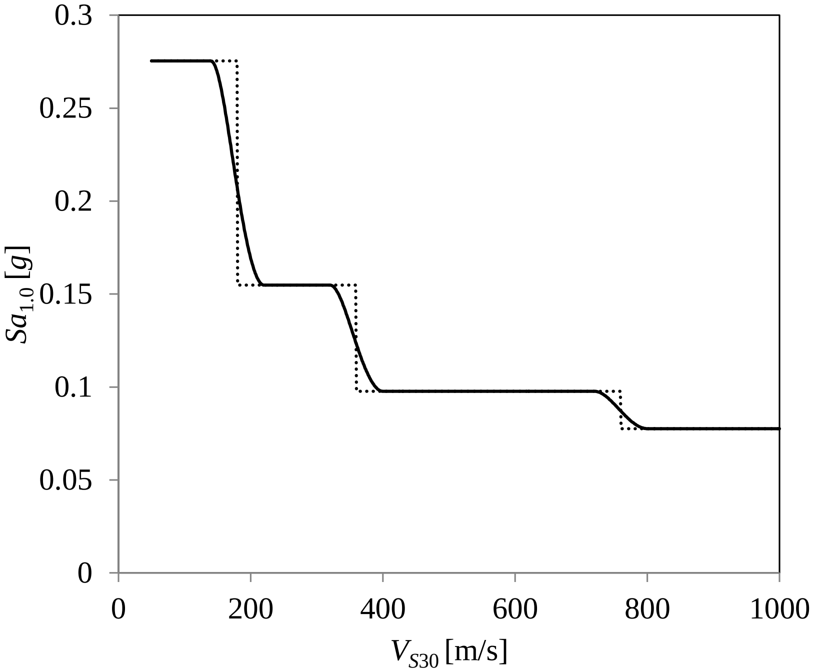


Figure 3-6: Smoothed ground motion relationship by Atkinson and Boore (2003): Spectral acceleration at 1.0 sec. plotted against the shear wave velocity.

3.5. REGIONAL DAMAGE MODEL

Three levels of modeling were identified in Figure 3-2. Modeling at the regional scale, which is described in this section, requires less data and less computational resources compared to the building-by-building modeling that is described later. As the name suggests, a regional model predicts damage and loss within an entire region. Naturally, the savings in modeling and computer efforts are counteracted by high model uncertainty. The characterization of this uncertainty is the primary objective in this section.

In general, a region encompasses different land uses, here categorized as "zones." These zones are analogous to typical city zoning and—in the implementation in Rt—each region has K zones. Damage to the building stock within a zone is measured by a damage ratio, η_k, (k=1,2,…,K) where η_k=0 indicates no damage and η_k=1 indicates complete damage. Complete damage requires replacement of the entire building stock. In this study K=5 because five zone types are considered: single-residential (detached family housing); multi-residential (apartments); commercial; industrial; and comprehensive development (high-rise construction).

In the following, regression models for η_k for the different zone types are developed. For this purpose, it is necessary to have observed earthquake damage in specific regions. Ideally, data from past earthquakes are utilized, such as insurance claims data. In lieu of such information, detailed analysis of a large building inventory is conducted. The author was given access to a database for 8,330 buildings in Vancouver, for which key characteristics are known. One of these characteristics is the occupancy type, described shortly, which is employed to categorize the buildings into different zones.

To generate the data, first a range of earthquake magnitudes and distances are randomly generated. For each magnitude and distance pair, Sa at T_n=0.3sec and T_n=1.0sec, denoted by $Sa_{0.3}$

and $Sa_{1.0}$, at the location of each building is computed. $Sa_{0.3}$ and $Sa_{1.0}$ are not independently generated because that would miss the correlation between these two quantities. Rather, this correlation is preserved by using the same realization of magnitude and distance to generate each pair of $Sa_{0.3}$ and $Sa_{1.0}$. *Sa* at these two periods is commonly employed in the literature to construct a complete response spectrum, see, *e.g.*, NEHRP (Building Seismic Safety Council 1997) and FEMA-NIBS (2003). $Sa_{0.3}$ and $Sa_{1.0}$ for each building are input to the capacity spectrum method (Mahaney *et al.* 1993), which is implemented in Rt as a separate model. Furthermore, the peak displacement and acceleration response from the capacity spectrum method are input to the HAZUS fragility and loss functions (FEMA-NIBS 2003) to compute the loss for each of the 8,330 buildings. Thereafter, the damage ratio, η_k, for each zone is computed by dividing the total loss of the buildings in that zone by their total value. The latter is obtained by imposing full damage in the HAZUS fragility and loss functions. Finally, η_k is recorded together with $Sa_{0.3}$ and $Sa_{1.0}$ at the centroid of the region. Although the correlation coefficient between $Sa_{0.3}$ and $Sa_{1.0}$ is substantial at 0.85, both $Sa_{0.3}$ and $Sa_{1.0}$ were considered as regressors in the model development.

A number of model forms for η_k as a function of $Sa_{0.3}$ and $Sa_{1.0}$ are explored. To model the transition from no damage (η_k=0) for low values of $Sa_{0.3}$ and $Sa_{1.0}$ to complete damage (η_k=1) for high values of $Sa_{0.3}$ and $Sa_{1.0}$, several functions are explored. Rather than the linear and the sine function, the most successful turned out to be the standard normal CDF, Φ(.), which transitions smoothly from zero to unity as the argument increases. It is first attempted to let the argument of Φ be a linear function of $Sa_{0.3}$ and $Sa_{1.0}$, *i.e.*, with the following model form for each zone type:

$$\eta = \Phi\left(\theta_1 + \theta_2 \cdot Sa_{0.3} + \theta_3 \cdot Sa_{1.0}\right) + \varepsilon \tag{3-6}$$

where θ_i=model parameters and ε=model error. However, Eq. (3-6) turns out to provide a poor fit to the data; it tends to over-predict low damage and under-predict high damage. As a result, the following nonlinear form of the argument is introduced:

$$\ln(\eta) = \ln\left[\theta_1 \cdot \Phi\left(\theta_2 + \theta_3 \cdot \left(Sa_{0.3}\right)^{\theta_5} + \theta_4 \cdot \left(Sa_{1.0}\right)^{\theta_5}\right)\right] + \varepsilon \tag{3-7}$$

where it is noted that the powers of $Sa_{0.3}$ and $Sa_{1.0}$ are equal, and that the logarithmic transformation is introduced to obtain a homoscedastic model. It is reiterated that $Sa_{0.3}$ and $Sa_{1.0}$ are evaluated at the centroid of the region.

To determine the mean, standard deviation, and correlation for the model parameters θ_i in Eq. (3-7), the nonlinear regression described by Seber and Wild (2003) is employed. The regression analysis is carried out in MATLAB® with the Levenberg–Marquardt algorithm (Marquardt 1963) for solving the nonlinear least squares problem. According to Seber and Wild (2003), the model parameters, θ_i, are jointly t-distributed, and as the number of data increases, the t-distribution tends to a normal distribution. As an approximation, in the application of Eq. (3-7) for regional risk analysis, the model parameters are assumed to have the normal distribution. Table 3-1 shows the mean and coefficient of variation that is obtained for θ_i and ε for the five zone-types. The table shows that for the single-residential, the mean of θ_3 is several times larger than the mean of θ_4. This correctly suggests that $Sa_{0.3}$ is a more important regressor than $Sa_{1.0}$ for single-residential buildings, which are mostly low-rise and thus have shorter periods. Conversely, the Table 3-1 shows that $Sa_{1.0}$ is more influential in a comprehensive development zone, which mostly consists of high-rise buildings with long periods. The complete second-moment information of the model parameters for different zone types is provided in Appendix B.

To demonstrate the predictions made by the regional damage model in Eq. (3-7), Figure 3-7 shows the median damage plus/minus one standard deviation for a single-residential zone. The plot displays damage for $Sa_{1.0} \leq Sa_{0.3} \leq 3 \cdot Sa_{1.0}$ because points outside this range are uncommon. The desired smooth S-shaped increase in damage for increasing ground shaking intensity is observed. It is also observed in Figure 3-7 that the uncertainty in the damage is significant. This is reasonable because it covers a range of possible building types within a zone. In fact, it is stressed that the uncertainty in Eq. (3-7) has four contributions: 1) Variability in the ground shaking intensity, which is included in $Sa_{0.3}$ and $Sa_{1.0}$; 2) Variability in the spatial distribution of ground shaking, which is included in the model parameters because each of the 8,330 buildings have a different intensity according to their exact location; 3) Variability in the amount of damage to each building, which is included in the FEMA-NIBS fragility curves that are employed; and 4) Variability in the building characteristics within a zone.

Table 3-1: Second moments of model parameters for regional damage models. CoV=coefficient of variation and σ_ε=standard deviation of the model error, ε.

		Single-residential	Multi-residential	Commercial	Industrial	Comprehensive Development
θ_1	Mean	0.779	1.415	0.580	0.304	0.663
	CoV	0.021	0.101	0.020	0.013	0.030
θ_2	Mean	−3.981	−11.397	−6.333	−5.677	−9.461
	CoV	0.016	0.109	0.030	0.034	0.049
θ_3	Mean	3.126	3.667	2.282	3.401	1.215
	CoV	0.018	0.119	0.039	0.035	0.082
θ_4	Mean	0.698	6.641	4.126	2.916	7.919
	CoV	0.051	0.115	0.030	0.036	0.049
θ_5	Mean	0.436	0.081	0.209	0.249	0.117
	CoV	0.028	0.145	0.042	0.046	0.065
σ_ε	Mean	0.117	0.095	0.095	0.100	0.086
	CoV	0.023	0.023	0.023	0.023	0.023

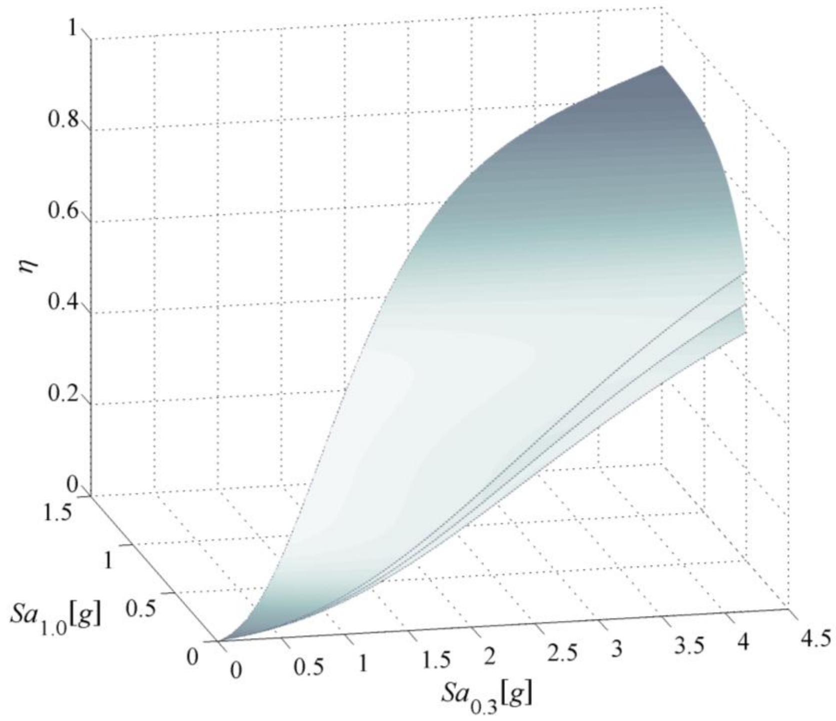


Figure 3-7: Damage ratio predicted by the regional damage model versus $Sa_{0.3}$ and $Sa_{1.0}$.

3.6. REGIONAL LOSS MODEL

This chapter addresses seismic loss due to the cost of repair of damaged buildings. Once the zone-specific damage ratio, η_k, is computed by Eq. (3-7), the total regional loss is

$$l = \sum_{k=1}^{K} \left(\eta_k \cdot A_k \cdot C_k \right) \tag{3-8}$$

where A_k=area of zone k and C_k=replacement cost of the buildings in zone k measured per unit area. Two interpretations are possible for A_k and C_k. One is that A_k is the floor area of construction, so that C_k is the replacement cost per unit floor area. Another interpretation is that A_k is the area of the entire zone, so that C_k is the replacement cost per unit area of land. In the analysis presented in the next chapter, the former interpretation is adopted, but in Rt the analyst is free to adopt either one. In Rt, the analyst defines any number of corner points of the region by clicking on an interactive map. Thereafter, zone percentages within the region are specified. For example, of the total area of a region, 10% may be commercial and 40% residential. Because of the different possible interpretations of A_k, the zone percentages need not add up to 100%. A_k is automatically computed in Rt once the analyst has specified the zone-percentages, and the analyst inputs the replacement cost, C_k, as a parameter, which may be given as a constant or a random variable. Values for C_k for different zones are provided in Chapter 4.

3.7. BUILDING RESPONSE MODELS

With reference to Figure 3-2, it is straightforward in Rt to replace regional models with many models for individual buildings. This refinement leads to more accurate results at the cost of more modeling and computer efforts. The characteristics of the building models are obtained by walk-down surveys, satellite imagery, and municipal databases. In this and the next sections, the following information is employed: 1) Load bearing system and construction material, which places the building in one of 13 “prototype” categories, listed in Table 3-2; 2) Occupancy type, which places the building in one of, coincidentally, 13 occupancy classes; 3) State of seismic retrofit and time of construction, which determines the parameter α for construction quality in accordance with Table 3-3; this parameter is analogues to the building code level in FEMA-NIBS (2003); 4) Number of stories, N, which determines the building height, H=N·3m; 5) Footprint area, A; 5) Plan irregularity, I_{PI}, which is unity if the building has a non-rectangular plan and zero otherwise; 6) Vertical irregularity, I_{VI}, which is unity if there is a change in the building plan along its height and zero otherwise; 7) Soft story, I_{SS}, which is a specific form of vertical irregularity, and is unity if the building has a story with significantly less stiffness than other stories, *e.g.*, parking in the first floor, and zero otherwise; 8) Short column, I_{SC}, which is unity if there are columns in the building with high ratio of width to height and zero otherwise;

and 9) Pounding, I_P, which is unity if there is insufficient separation with adjacent buildings and zero otherwise.

Table 3-2: Building prototypes.

Material	Load Bearing System				
	Shear Wall	Moment Frame	Braced Frame	Light Frame	Frame/Masonry Wall
Reinforced Concrete	Concrete Shear Wall	Concrete Moment Frame	-	-	Concrete Frame with Masonry Infill Wall
Precast Concrete	Precast Frame with Concrete Shear Wall	-	-	-	-
Steel	Steel Frame with Concrete Shear Wall	Steel Moment Frame	Steel Braced Frame	Steel Light Frame	Steel Frame with Masonry Infill Wall
Wood	-	Wood Large Frame	-	Wood Light Frame	-
Reinforced Masonry	-	-	-	-	Reinforced Masonry Bearing Wall
Unreinforced Masonry	-	-	-	-	Unreinforced Masonry Bearing Wall

Table 3-3: Building code levels, *i.e.*, construction quality.

Time of Construction	Unretrofitted	Retrofitted
Before 1940*	α=1: Pre-code	α=3: Moderate-code
From 1940 to 1975	α=2: Low-code	α=3: Moderate-code
After 1975**	α=3: Moderate-code	α=4: High-code

* The first introduction of seismic provisions in the Canadian building code (Mitchell *et al.* 2010).

** Upgrade of the Canadian building code with seismic detailing provisions (Uzumeri *et al.* 1978).

In this section, a new building response model is developed for each of the 13 prototypes. The responses are peak inter-story drift ratio, δ_p, and peak acceleration response, A_p. These two responses are selected because, according to the definition that is adopted in this study, they are directly related to the structural and non-structural damage. In fact, corresponding damage models are developed in the next section, while a later section presents loss models for the 13 occupancy classes. The motivation for developing new models instead of employing those implemented in HAZUS is threefold: 1) To explicitly describe the model uncertainty by random variables; 2) To have continuously differentiable models, which is not the case for, *e.g.*, the response model in HAZUS due to presence of "kinks" in the demand curve; and 3) To create models that are updated in a Bayesian fashion once new observations become available.

Seismograph readings of δ_p and A_p from past earthquakes are limited. Therefore, the capacity spectrum method (Mahaney *et al.* 1993) described in FEMA-NIBS (2003) is employed to generate data for the regression analysis. In particular, δ_p and A_p are computed and H, α, and Sa at T_n are recorded as regressors. In the initial modeling efforts, these regressors were directly employed in a variety of model forms that unfortunately were incapable of providing good predictions of δ_p and A_p. Therefore, a stronger emphasis on the mechanics of the problem was introduced. As a result, several "explanatory functions" that define the structural dynamics of the building are considered as regressors. These parameters include T_n, as well as the strength-to-weight ratio, V, yield drift ratio, δ_y, ductility, μ, ultimate drift ratio, δ_u, and degradation factor, κ. Sub-models are established for these parameters, in which T_n is modeled as a function of the building height:

$$T_n = \theta_1 \cdot H^{\theta_2} \tag{3-9}$$

Eq. (3-9) is a well-known equation for estimating the period. V and μ are modeled as functions of code level and height:

$$V = \theta_1 \cdot \exp\left(-\theta_2 \cdot H\right) \cdot \frac{2+\left(\alpha-2\right)\cdot\left(\alpha-1\right)}{8} \tag{3-10}$$

$$\mu = \theta_1 \cdot H^{-\theta_2} \cdot \frac{10+\left(\alpha-2\right)\cdot\left(\alpha-1\right)}{16} \tag{3-11}$$

because the data indicate that these two parameters typically increase with code level and decrease with height. For instance, the α-dependent factor on the right-hand side of Eq. (3-10) equals 1.0 for the high code level, while it reduces V by factors of 0.5 and 0.25 for moderate and low code levels, respectively. It is noted that *V* represents a means by which damage can be introduced into the model, for example in aftershock predictions. *I.e.*, in a damaged building, the strength is reduced to the residual capacity of the structure, as pointed out by Luco *et al.* (2004). Next, the yield drift ratio equals the strength, $V \cdot g$, where *g* is the acceleration of gravity, divided by the elastic stiffness, $4\pi^2/T_n^2$, and normalized by *H*:

$$\delta_y = \frac{T_n^2}{4\pi^2} \cdot \frac{V \cdot g}{H} \tag{3-12}$$

The ultimate drift ratio is

$$\delta_u = \mu \cdot \delta_y \tag{3-13}$$

Finally, the degradation factor is modeled as a function of the spectral acceleration demand and the building code level:

$$\kappa = \exp\left(-\theta_1 \cdot Sa \cdot \alpha^{-\theta_2}\right) \tag{3-14}$$

where the negative sign in the exponent appears because the structure degrades more as the demand increases. When *Sa* increases, then κ tends to zero, *i.e.*, full degradation. Conversely, κ=1 when *Sa*=0, *i.e.*, no degradation.

Eqs. (3-9) to (3-14) are established separately for each of the 13 prototypes. In turn, the six parameters T_n, V, δ_y, μ, δ_u, and κ are utilized as regressors to model δ_p and A_p for all prototypes. A number of different model forms are tried. Each model is assessed by plotting the model predictions against the data and the model residuals against the regressors. In this process, some model forms exhibited inadequate predictions and some suffered from heteroscedasticity. In conclusion, the models that best predict δ_p and A_p are:

$$\ln(\delta_p) = \theta_1 + \theta_2 \cdot \ln(\delta_y) + \theta_3 \cdot \ln(\delta_u) - \theta_4 \cdot \ln(V) - \theta_5 \cdot \ln(\kappa) + \theta_6 \cdot \ln(Sa) + \theta_7 \cdot Sa + \varepsilon \tag{3-15}$$

$$\ln(A_p) = \theta_1 - \theta_2 \cdot \ln(\delta_y) + \theta_3 \cdot \ln(V) - \theta_4 \cdot \ln(\mu) + \theta_5 \cdot \ln(\kappa) + \theta_6 \cdot \ln(Sa) + \varepsilon \qquad (3\text{-}16)$$

where θ_i=model parameters and ε=model errors. The second-moment information for the model parameters is obtained from a linear regression analysis. The procedure to carry out the regression analysis is described on Page 117 in Box and Tiao (1992) and implemented as a "model inference" analysis option in Rt. These second-moments are presented in Appendix B and implemented in Rt. It is observed that the model form in Eqs. (3-15) and (3-16) is essentially multiplicative due to the natural logarithm on the left-hand side. A model without the natural logarithm on the left-hand side suffers from heteroscedasticity and non-normality of the residuals. The latter is illustrated in Figure 3-8, where the residual quantiles for such a model are plotted against normal theoretical quantiles. The points in this plot significantly deviate from the 45° line, which indicates the non-normality of errors. This is remedied by the natural logarithm on the left-hand side. The natural logarithms on the right-hand side improve the model prediction and homoscedasticity of the model.

Figure 3-9 shows the median model predictions against the data for the drift model in Eq. (3-15). The data points are relatively close to the solid line, which is one indication that the model provides reasonable predictions. The models in Eqs. (3-15) and (3-16) originally included damping and overstrength as regressors, but these were omitted in a stepwise modeling process, similar to the one described in Gardoni *et al.* (2002). In this process, the regressor whose θ_i has the largest coefficient of variation is the least informative and may be omitted. This was the case for damping and overstrength. By removing these terms, the standard deviation of the model error increased only slightly, which indicates that the reduced model is acceptable. In conclusion, T_n, V, δ_y, μ, δ_u, and κ appear to be the most important building characteristics.

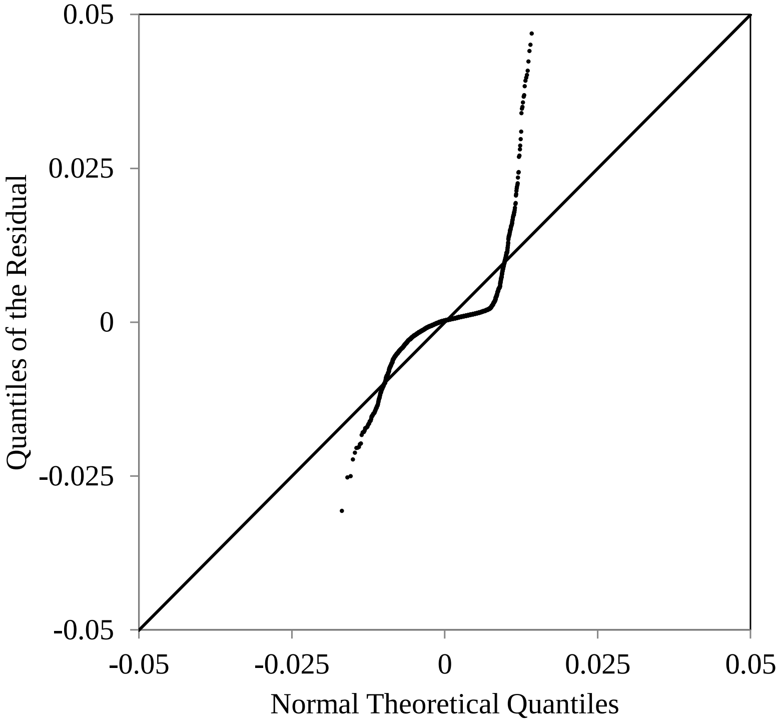


Figure 3-8: Quantile-quantile plot to assess the normality of residuals of a drift response model without the natural logarithm on the left-hand side.

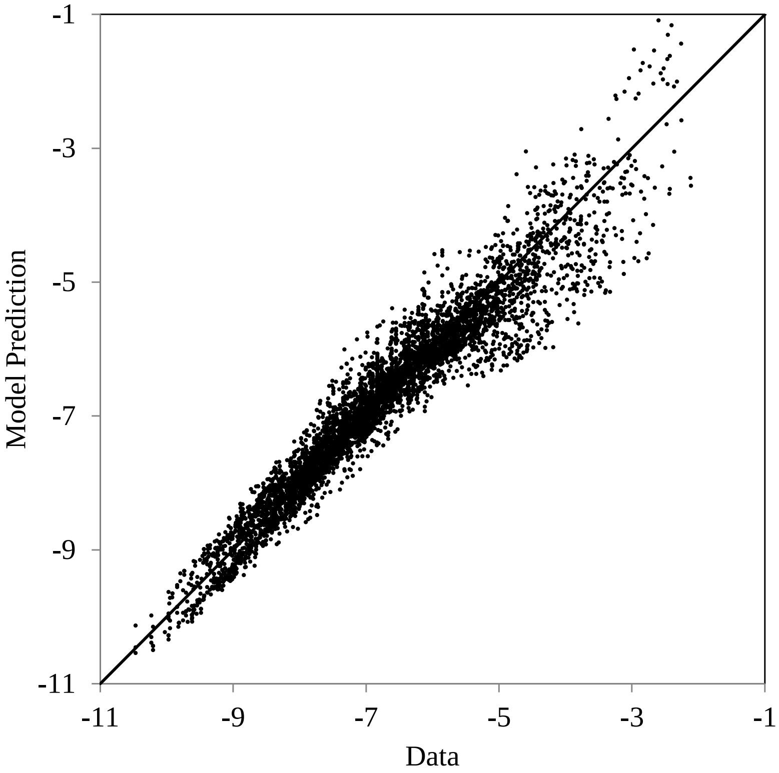


Figure 3-9: Predictions of the drift model in Eq. (3-15) versus data.

3.8. BUILDING DAMAGE MODELS

Given the responses δ_p and A_p, this section addresses the ensuing damage. Damage is here expressed as the ratio of the repair cost to the replacement cost of the building. Four damage ratios are developed: 1) Structural damage, η_S; 2) Non-structural drift-sensitive damage, η_{ND}; 3) Non-structural acceleration-sensitive damage, η_{NA}; and 4) Content damage, η_C. The first two factors depend on δ_p, while the last two depend on A_p.

Data for the regression analysis is generated by substitution of realizations of δ_p and A_p into the fragility curves from FEMA-NIBS (2003). To account for the increased damage due to building irregularities, the information from the building scoring system in ATC-21 (ATC 1988) is employed. ATC-21 assigns a base score to each building, which is the negative $\log_{10}$ of the probability of exceeding a 60% damage ratio. In addition, different irregularities modify the base score. For example, ATC-21 proposes a decrease of 0.5 to the base score for a wood frame building if it has a vertical irregularity. Consequently, to generate data for such a building, the probability of the highest damage state is increased by $10^{0.5}$=3.2. A similar method is employed by MacCormack and Rad (1997) to incorporate irregularities in the risk assessment of a city. Using the updated damage state probabilities and the damage ratios associated with each damage state, the expected damage ratio is computed, with δ_p, A_p, H, α, and irregularity indices, I_i, recorded as regressors.

For all models, a smooth increase in damage from 0 to 1 due to increasing building responses is sought. As in the regional modeling, polynomial, trigonometric, and even logit functions were tested, but the standard normal CDF, Φ, again turned out to provide the best fit. The building irregularities are included in the structural damage model by means of the exponential function. This function produces a factor to increase the damage if irregularities exist. This yields the following model for structural damage:

$$\eta_S = \begin{pmatrix} \Phi\left(\theta_1 + \theta_2 \cdot \ln(\delta_p) + \theta_3 \cdot \ln(H) - \theta_4 \cdot \alpha\right) \\ \cdot \exp\left(\theta_5 \cdot I_{VI} + \theta_6 \cdot I_{PI} + \theta_7 \cdot I_{SS} + \theta_8 \cdot I_{SC} + \theta_9 \cdot I_P\right) \end{pmatrix} + \varepsilon \tag{3-17}$$

The second moment information for the model parameters θ_i for each of the 13 building prototypes is presented in Appendix B and implemented in Rt. The other damage models are considered independent from the building prototype and building irregularities:

$$\eta_{ND} = \Phi\left(\theta_1 + \theta_2 \cdot \ln(\delta_p)\right) + \varepsilon \tag{3-18}$$

$$\eta_{NA} = \Phi\left(\theta_1 \cdot \ln(A_p) - \theta_2 \cdot \alpha\right) + \varepsilon \tag{3-19}$$

$$\eta_C = \Phi\left(\theta_1 \cdot \ln(A_p) - \theta_2 \cdot \alpha\right) + \varepsilon \tag{3-20}$$

Figure 3-10 shows median predictions of the structural damage according to Eq. (3-17) for four common prototypes. The figure shows that at the same drift ratio, unreinforced masonry buildings incur the most damage, while steel frame and concrete shear wall buildings experience the least damage. Light wood frame buildings fall in between. The negative sign of the θ_i parameters associated with α in Eqs. (3-17), (3-19), and (3-20) correctly indicates that the damage decreases as the quality of construction increases. It is also noted that the structural damage model in Eq. (3-17) suggests that taller buildings incur more damage at the same level of drift ratio. Furthermore, amongst the θ_i parameters that correspond to irregularities (θ_5 to θ_9) in Eq. (3-17), regression yields the highest mean for θ_7 for most prototypes. This implies that soft-story irregularity is the most detrimental type of irregularity. Conversely, θ_9 has the lowest mean for most prototypes, which indicates that pounding imposes least damage compared with other irregularities.

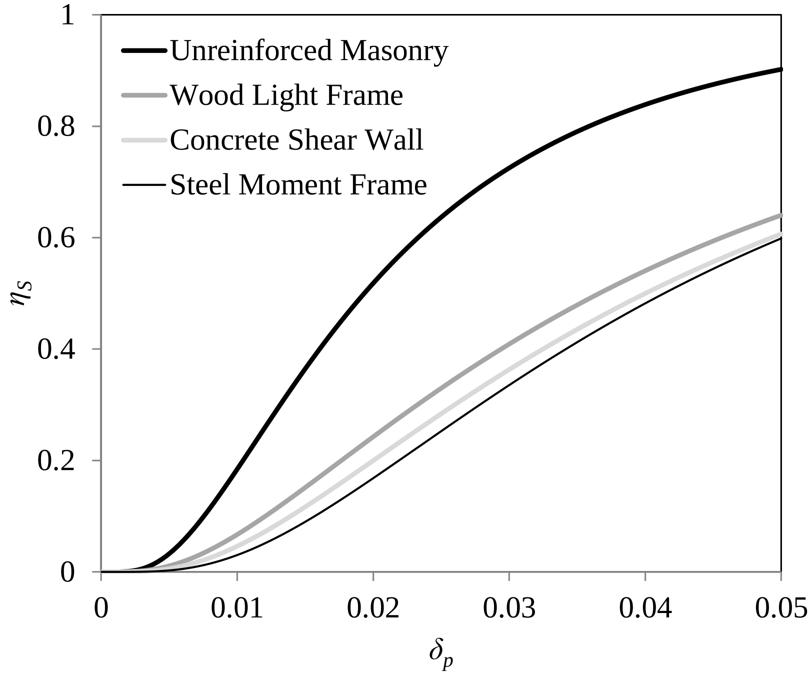


Figure 3-10: Structural damage ratio versus drift ratio.

3.9. Building Loss Model

Provided a damage ratio, the associated repair cost is computed by multiplying it with the building replacement cost per unit floor area and the building floor area. Summation over structural, non-structural, and content yields:

$$l = \left(\eta_S \cdot C_S + \eta_{ND} \cdot C_{ND} + \eta_{NA} \cdot C_{NA} + \eta_C \cdot C_C\right) \cdot A \cdot \varepsilon \tag{3-21}$$

where η_i=damage ratios from the previous section, C_i=corresponding replacement costs per unit floor area, A=total floor area, and ε=model error variable that is a normal random variable with unit mean and 10% coefficient of variation. Specific information for different occupancy classes are provided in Chapter 4.

3.10. Discounting Model

Future seismic losses must be discounted to present value to facilitate comparison of different risk mitigation actions. The discount model in Rt employs continuous discounting to discount the future loss, l, to present value, l_p. As a result, losses far into the future have less present value than a loss at present time. The discount model in Rt takes a dedicated Time parameter, t, as input, which identifies the time that the loss occurs. The model reads

$$l_p = l \cdot \exp\left(-\rho \cdot t\right) \qquad (3\text{-}22)$$

where ρ=effective interest rate. In the reliability analyses conducted in this study, ρ is a normal random variable with mean equal to 3% and 10% coefficient of variation.

3.11. Conclusions

This study proposes a reliability-based approach for risk analysis. The approach employs a collection of many interacting probabilistic models, and new models are presented in this chapter. The models are generic, and they are applied in the next chapter to risk analysis for the Vancouver metropolitan region in Canada. Comprehensive modeling of uncertainty and updating of the models as new information becomes available are key driving forces behind this study. In fact, an important vision behind this study is to promote candid modeling of epistemic uncertainty, *i.e.*, reducible uncertainty, which is subsequently reduced by targeted efforts as more observations and better mechanical understanding become available. The models are implemented in Rt, which was presented in Chapter 2. An important objective in Rt is to make reliability methods and a library of predictive models available to a broad engineering audience. This is intended to advance the use of probabilistic models and reliability methods in a variety of applications, here with focus on seismic risk.

Chapter 4. SEISMIC RISK ANALYSIS WITH RELIABILITY METHODS: ANALYSIS

4.1. INTRODUCTION

Two objectives are pursued in this chapter. One is to put forward a new risk analysis methodology that is based on reliability methods. The other is to provide seismic loss probabilities and additional insight for the Vancouver metropolitan region in Canada. The use of reliability methods to assess seismic risk is emphasized. This approach is suggested here as an alternative to other risk analysis strategies; one existing approach employs the total probability theorem and conditional probability models (Cornell and Krawinkler 2000); another is based on modified Mercalli intensity (MMI) and damage probability matrices (ATC 1985). The intentions in this study is to leverage the power of classical reliability methods, make them available through the new computer program, Rt, and put forward a library of models at the region, building, and component level, as suggested in Figure 3-2.

The previous chapter described the development of models for use with reliability methods. A collection of models for seismic hazard, response, damage, and loss is proposed and implemented in Rt. The software architecture of Rt is specifically designed to accommodate reliability and optimization with many interacting probabilistic models. Another important purpose of Rt is to make reliability methods accessible to a broad engineering audience. In the past, reliability methods, such as FORM and SORM, have primarily been employed by reliability experts in special applications and code calibration. Risk analysis, in contrast, is often conducted with other analysis techniques. The ATC-13 approach (ATC 1985), proposed by the Applied Technology Council, was the first comprehensive framework for regional evaluation of damage due to earthquakes. ATC employed expert opinion to develop estimates of seismic damage and loss for different MMI-values. Rojahn *et al.* (1986) presented an overview of the methodology, including a list of building categories and results. Two decades after the introduction of ATC-13, a handful of strong earthquakes prompted the development of a new loss estimation methodology by the U.S. Federal Emergency Management Agency and the U.S. National Institute of Building Sciences (FEMA-NIBS 2003). The methodology was implemented in the computer program HAZUS®. Unlike ATC-13, the FEMA-NIBS approach employs the capacity spectrum method (Mahaney *et al.* 1993), which introduces mechanical considerations into the assessment of damage. The result of these approaches is an expected loss value. Alternatives to ATC-13 and FEMA-NIBS are explored in this chapter, with predictive probabilistic models to generate the probability distribution of loss.

Recently, the use of conditional probabilities and total probability integration has become popular in seismic loss assessment. One example is the framing equation of the Pacific Earthquake Engineering Center (Cornell and Krawinkler 2000). Wen and Ellingwood (2005) also proposed a framework based on total probability integration for fragility-based loss estimation of buildings. Ellingwood *et al.* (2007) assessed seismic fragilities for typical building types in the central United States. The work presented in this thesis can be regarded as a reliability-based alternative to total probability integration over conditional probabilities. The models presented in this thesis simulate a physical phenomenon and produce measurable responses, with uncertainty described by random variables.

Another set of studies on regional infrastructure focused on spatial distribution of the building performance over a region, and spatial correlation of the ensuing damage and loss. Wesson and Perkins (2001) introduced a method to calculate the spatial correlation of earthquake ground motions and losses. Bielak *et al.* (2005) provided an overview of the methods developed for large-scale regional simulation of seismic performance in a project on Seismic Performance of Urban Regions. Lee and Kiremidjian (2007) presented a framework for treating the correlation in network risk analysis. Goda and Hong (2008) investigated the effect of spatial correlation of seismic demand on the aggregate seismic loss of spatially distributed structures in a region. In contrast, the present study addresses this correlation by modeling the underlying phenomena, *i.e.*, the location and magnitude of the earthquake that causes the ground shaking.

While most regional studies have considered a portfolio of hypothetical structures, some applications have established databases of actual buildings. In accordance with the ATC-13 (ATC 1985) approach, Ventura *et al.* (2005) developed the BC-31 classification of buildings in British Columbia, Canada, and generated a comprehensive database. The data were collected using rapid screening and visual inspection and covered a considerable part of the buildings in the cities of Vancouver, New Westminster, and Victoria in British Columbia. The first two cities are located within the Vancouver metropolitan region that is considered in this study. Onur *et al.* (2005) applied the ATC-13/BC-31 approach to the collected data to assess the regional seismic risk. One result was that the mean loss for a part of the City of Vancouver that was subject to study due to an earthquake with an MMI-intensity equal to eight is $3.5 billion.

The Vancouver metropolitan region covers an area of 2,887 km^2 and consists of 23 municipalities. For readers unfamiliar with the region, it is shown in Figure 4-1. The names in the figure identify cities and townships that are addressed in this study. This region is located in southwestern Canada, and is exposed to a significant seismic hazard. This chapter presents a comprehensive seismic risk analysis for the region conducted with Rt while the models are presented in the previous chapter. In the following, the two analysis approaches are first presented, followed by a description of the Vancouver-specific modeling assumptions. Thereafter, the analysis results are presented and discussed.

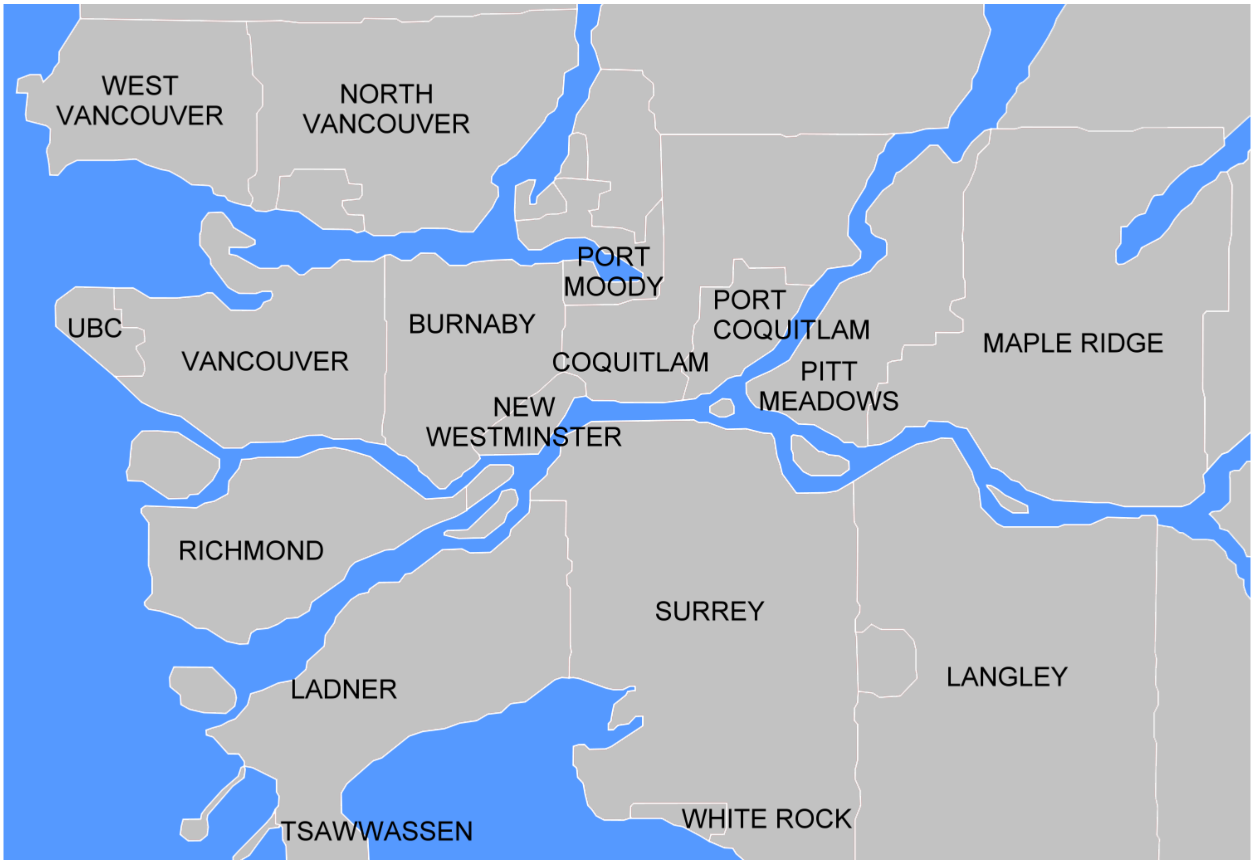


Figure 4-1: Vancouver metropolitan region.

4.2. FORM, SORM, and Hazard Combination

The essence of a reliability analysis is random variables, collected in the vector **x**, and limit-state functions, $g_i(\mathbf{x})$. Both physical variables, *e.g.*, magnitude, and model variables, *e.g.*, model error, are included in **x**. The primary objective of a reliability analysis with one limit-state function is to determine the probability that the limit-state function will take on negative outcomes. This probability is denoted by $p_i=P[g_i(\mathbf{x})\leq 0]$. In other words, the limit-state function identifies the event for which the probability is sought. The limit-state function

$$g_i(\mathbf{x}) = l_o - l_i(\mathbf{x}) \tag{4-1}$$

is central in this study because it yields the probability that the loss, $l(\mathbf{x})$, is greater than the threshold, l_o. It is emphasized that the evaluation of $l(\mathbf{x})$ requires a host of probabilistic models of the type developed in the previous chapter.

Any reliability method evaluates g and perhaps the gradient vector $\partial g/\partial \mathbf{x}$ several times, for different realizations of $\mathbf{x}$, to obtain an estimate of p_i. The FORM analysis is an appealing method because it requires only a handful of evaluations of g and $\partial g/\partial \mathbf{x}$ to produce an estimate. FORM also provides valuable insight into the relative importance of each random variable. As described by Der Kiureghian (2005), FORM includes a search for the "design point," which is the most likely realization of $\mathbf{x}$ associated with g_i=0 in the space of standard normal variables. The result of the search is the reliability index β_i, which is related to the sought probability by the equation

$$p_i = \Phi(-\beta_i) \tag{4-2}$$

FORM produces a good estimate of p_i depending on the topology of the limit-state surface near the design point. This result may be inaccurate if the limit-state function is strongly nonlinear in the space of standard normal variables. Under such circumstances, the SORM and importance sampling are utilized to improve the FORM result; see details in, *e.g.*, Der Kiureghian (2005) and Ditlevsen and Madsen (1996).

The problem under consideration has several limit-state functions of the form in Eq. (4-1) because several sources of seismic hazard are present. Specifically, the region is subjected to shallow crustal earthquakes, deep subcrustal earthquakes, and megathrust subduction earthquakes. Because each of these sources is associated with different location and magnitude models, they are modeled as different hazards with different occurrence rates. In fact, the aforementioned sources are further subdivided in this study, and N denotes the total number of hazards. As a result, multi-hazard analysis is necessary when analyzing the seismic risk in Vancouver. Several multi-hazard analysis methods are available in the literature. One option is the load coincidence method described by Wen (1990), which employs the Poisson pulse process. However, matters simplify in this study because the probability of the coincidence of two earthquakes is negligible. This implies that the Poisson point process is employed to model each hazard, each with rate of occurrence denoted by λ_i, i=1,2,3,…,N.

Suppose a reliability analysis is carried out for each hazard, so that β_i and p_i are known for all hazards. It follows that the rate of exceeding the loss threshold l_o is $\lambda_i \cdot p_i$ for each hazard. The combined rate that includes all hazards is the sum of the individual rates, and the well-known Poisson distribution provides the probability of exceedance within a time period, T:

$$p = 1 - \exp\left(-T \cdot \sum_{i=1}^{N} \lambda_i \cdot p_i \right) \tag{4-3}$$

where p is the probability that the total loss, $l(\mathbf{x})$, exceeds the threshold, l_o, when all hazards are considered. It is noted that this analysis approach is associated with inaccuracy when the probability of more than one earthquake in T is substantial. This is because of correlation between the loss events when there is more than one earthquake in T. For example, the realization of an uncertain building characteristic must be the same in the first and second earthquake. In contrast, Eq. (4-3) assumes statistically independent events during T. As shown later in this chapter, this leads to overestimation of p for small loss values, while p is accurately estimated by Eq. (4-3) for high loss thresholds, *i.e.*, rare events. The potential inaccuracy in Eq. (4-3) is also remedied in the analysis approach presented next.

4.3. Scenario Sampling Analysis

The FORM-based analysis is an effective approach that is successfully employed in the seismic risk analysis presented later. However, it does not readily include time-varying phenomena, such as long-term structural deterioration and discounting of future losses to present value. Furthermore, the approach presented above produces only one point on the loss curve. Therefore, an approach based on Monte Carlo sampling is explored to complement the FORM-based approach. This approach is called scenario sampling and employs mean-centered Monte Carlo sampling, where realizations of hazards, structures, and losses during T are repeatedly generated. The result is a rough estimate of the entire loss curve. For example, suppose the probability of seismic losses within the 50-year period from 2012 to 2062 is sought. The scenario sampling analysis repeatedly generates realizations, *i.e.*, scenarios, of this 50-year period. Some scenarios include several earthquakes at different times from different sources, while other scenarios do not see any earthquakes. The implementation in Rt is optimized so that evaluation of the response, damage, and loss models is only triggered when a significant event occurs. The total loss is accumulated for each scenario, with the same models as in the FORM-based analysis.

4.4. Modeling of Seismic Hazards in the Vancouver Region

While generic models are presented in the previous chapter, specific model input and modeling assumptions for the Vancouver metropolitan region are presented here.

4.4.1. Locations

The geometry of seismic sources from Adams and Halchuk (2003) is adopted for the modeling of the crustal, subcrustal, and subduction earthquakes in the Vancouver region. Specifically, the coordinates provided by Adams and Halchuk (2003) are used as input to the line and area source location models. Two alternative models are proposed by Adams and Halchuk (2003) for crustal earthquakes in western Canada, and two others are proposed for subcrustal earthquakes. For both sources, a “historical model” and a “regional model” are put forward. The latter is utilized here because it includes both historical data and geological considerations. In accordance with Adams and Halchuk (2003), the crustal source is divided into three area sources: CASR (Cascade Mountains, regional), JDFF (Juan de Fuca plate bending, offshore), and JDFN (Juan de Fuca plate bending, onshore). These sources are identified in Figure 4-2, which is produced by the map interface in Rt. This figure also shows the subcrustal source, which in accordance with Adams and Halchuk (2003) is modeled by one area source named GSP (Georgia Strait/Puget Sound). In passing, it is noted that the GSP subcrustal source overlaps with the CASR crustal source. The subduction source, which has the potential to generate high-magnitude earthquakes, is described as line source by Adams and Halchuk (2003) and shown by a thick line in Figure 4-2. The Poisson occurrence rate for all these sources is provided in Table 4-1. It is seen in this table that the CASR source is associated with the highest occurrence rate.

Table 4-1: Occurrence rate of earthquakes from different sources. Area source rates are from Adams and Halchuk (2003). The rate of subduction earthquakes is from Petersen *et al.* (2008).

Source	Annual Occurrence Rate
CASR	0.188
JDFF	0.0079
JDFN	0.0035
GSP	0.0932
Subduction	0.0037

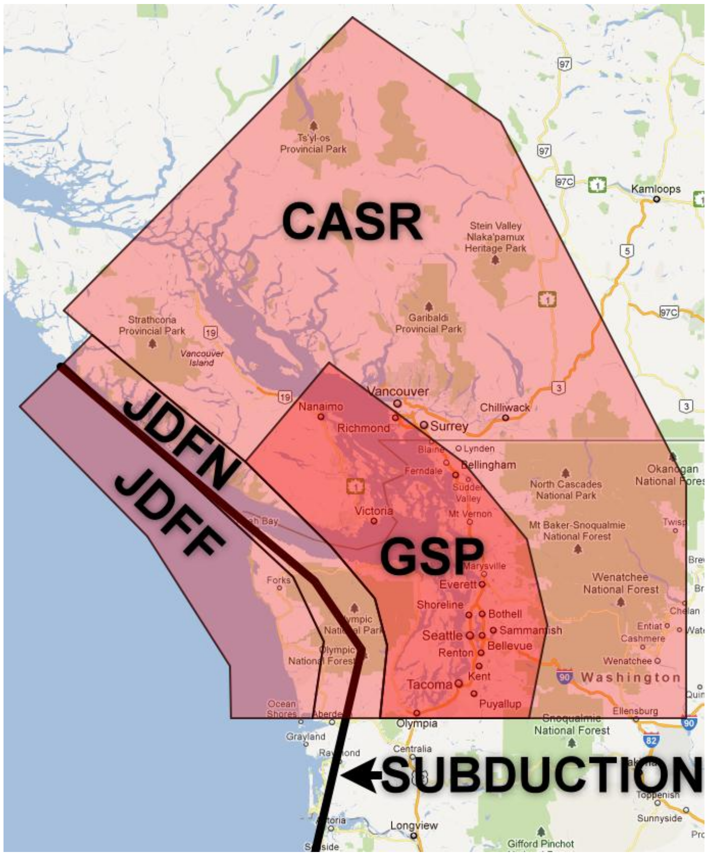


Figure 4-2: Seismic area sources in the Vancouver metropolitan region, visualized in Rt's Google Maps® interface. The map in the background is from Google (© 2012 Google).

4.4.2. Magnitudes

The magnitude of the crustal and subcrustal sources is modeled by a bounded exponential random variable with random parameters. In accordance with Eq. (3-3) in the previous chapter, this model takes two lognormal random variables, *i.e.*, $M_{\max}$ and b'. Their mean and standard deviation are computed by using the lower, best, and upper estimates for $M_{\max}$ and b' given by Adams and Halchuk (2003) together with their weights. The results are provided in Table 4-2. The deterministic parameter $M_{\min}$ in the magnitude model, *i.e.*, the smallest considered magnitude, is set equal to 5.0.

Table 4-2: Magnitude model parameters.

Source	b'		$M_{\max}$	
	Mean	Std. Dev.	Mean	Std. Dev.
CASR	1.01	0.38	7.7	N/A*
JDFF	1.87	0.52	6.94	0.19
JDFN	2.07	0.66	7.00	0.22
GSP	1.13	0.11	7.06	0.16

* A constant value is used as the maximum magnitude of the CASR source (Adams and Halchuk 2003).

Although the location of subduction earthquakes is modeled according to Adams and Halchuk (2003), who use a deterministic magnitude, the magnitude of these earthquakes is here modeled in accordance with Petersen *et al.* (2008). This strategy is adopted in this study because the geometry of the source in both studies is consistent. Specifically, the geometry provided by Adams and Halchuk (2003) is based on a 2D "dislocation model" by Hyndman and Wang (1995), while the geometry provided by Petersen *et al.* (2008) is based on a similar 3D model by Fluck, Hyndman, and Wang (Fluck *et al.* 1997). In accordance with Fluck *et al.* (1997), the difference between the results using these two models is small. The "recurrence curve" given by Petersen *et al.* (2008), which shows the annual occurrence rate of different magnitudes, is reproduced in Figure 4-3. In this study, different location models are associated with different magnitude ranges. For this purpose, it is noted that two scenarios were considered by Petersen *et al.* (2008) to construct the curve in Figure 4-3: Events with magnitudes 8.0 to 8.8 were associated with limited rupture zone, while events with magnitude 8.8 to 9.2 were associated with rupture of the entire subduction zone. As indicated in Figure 4-3, magnitudes below 8.8 are modeled by a location model called Subduction Source 1, while those above 8.8 are modeled by a location model called Subduction Source 2. Subduction Source 1 is a line source model, because the rupture location is uncertain. In contrast, Subduction Source 2 is modeled as the point closest to Vancouver on the subduction line, because the entire zone ruptures in such events and the location is therefore not uncertain.

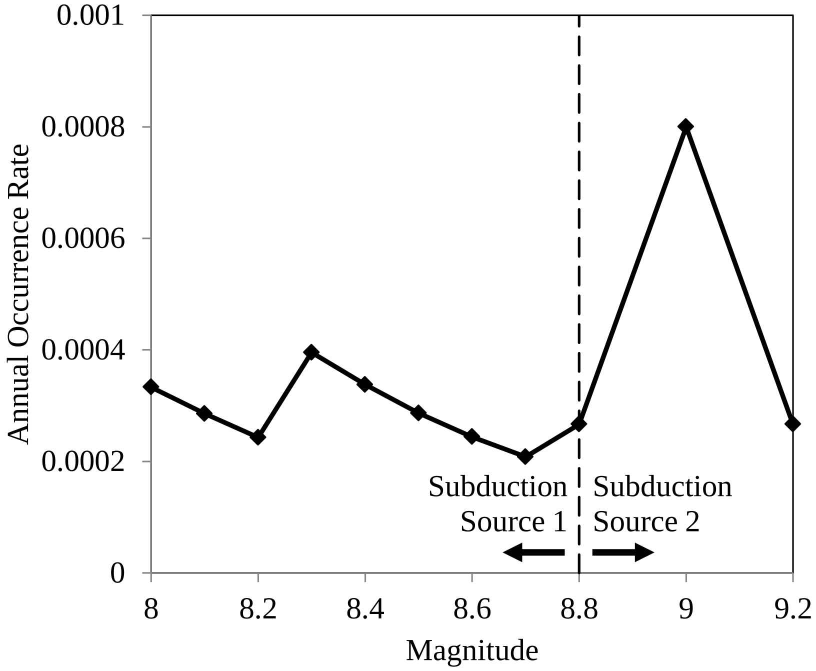


Figure 4-3: Recurrence relationship for the magnitude of subduction earthquakes.

Since two location models are utilized for subduction earthquakes, the occurrence rate in Row 5 of Table 4-1 must be split into two rates. These two rates are computed by integrating the curve in Figure 4-3 over the two respective magnitude ranges. In addition, the rate associated with Subduction Source 1 is multiplied by 500/1,150=0.43 because only 500km of the entire 1,150km actual line is included in the analysis and in Figure 4-2. This simplification is made because earthquakes beyond the 500km line in Figure 4-2 will have negligible effect on the Vancouver region. Finally, the magnitude for each of the two subduction sources is modeled by one random variable. The PDF for magnitudes in the range 8.0 to 8.8 is the normalized version of the recurrence curve on the left-hand side of the dashed line in Figure 4-3. Similarly, the PDF for magnitudes in the range 8.8 to 9.2 has the shape of the curve on the right-hand side of the dashed line in Figure 4-3.

4.4.3. Ground Shaking Intensities

For the CASR, JDFN, and JDFF crustal sources, the ground shaking intensity model proposed by Boore and Atkinson (2008) for shallow earthquakes is utilized. This is one of the most recently developed intensity models. The fact that it has a simpler form and fewer input parameters compared with other recent models makes it suitable for this region-oriented study. For the GSP subcrustal source, the intra-slab equation from Atkinson and Boore (2003) is employed. For subduction earthquakes, the relationship for interface events proposed by Atkinson and Boore (2003) is adopted. The Atkinson and Boore (2003) relationship is also one of the most recent models for subcrustal and subduction sources that is developed using a large database of ground motions. All these models were subjected to smoothing in the previous chapter.

As mentioned in the previous chapter, the shear wave velocity, V_{S30}, is a random variable that is input to the intensity models, with probability distribution inferred from observed data. Each region, described shortly, is associated with one V_{S30}-value. The author was given access to a database of 371 in-situ measurements of V_{S30} that covers nearly a third of the Vancouver metropolitan region. These measurements are employed to compute the mean and standard deviation of V_{S30} within each region. For regions with no information, nominal V_{S30}-values for the soil condition of the region are used as the mean, with a 20% coefficient of variation. It is emphasized that the lognormal probability distribution for V_{S30} is unbounded. This may prompt questions about its suitability because there is obviously some unknown value above which realizations of V_{S30} will not be observed. The justification for the unbounded distribution is threefold: 1) It is unclear which upper bound to apply; 2) It is inappropriate to neglect the remote possibility of high V_{S30}-values; and 3) The probability distribution decays exponentially, which appropriately describes the diminishing probability of high V_{S30}-values.

In the present analyses, the parameter ε in the intensity models is considered as a random variable that represents model uncertainty. As a result, the same realization of ε is employed throughout a region. In contrast with intensity models for peak ground acceleration, this is an approximation in the application of the intensity models for *Sa*. Ideally, different but correlated ε-variables should be utilized at different locations. This is readily possible in Rt, but circumvented in this study with the use of one ε-variable for each intensity model.

4.5. Modeling of Buildings in the Vancouver Region

Generic models that were presented in the previous chapter are employed here to model the building stock of all urban areas within the Vancouver metropolitan region. One of these areas is the campus of the University of British Columbia (UBC) in Vancouver. 622 buildings at the UBC campus are surveyed and modeled individually by the building response, damage, and loss models. The rest of the region, where the building inventory is incomplete or unavailable, is modeled by the regional damage and loss models.

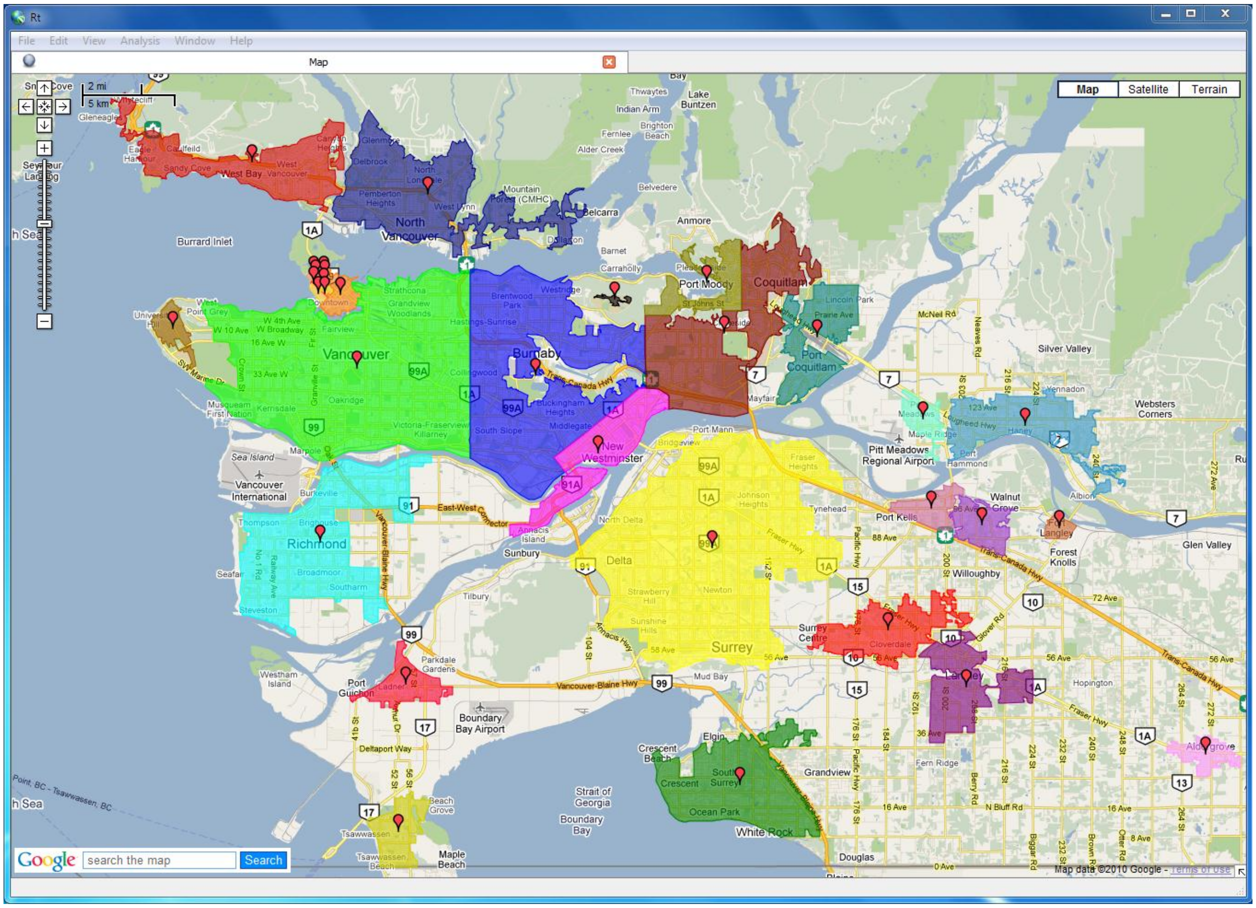


Figure 4-4: Regions modeled by the regional damage and loss models, visualized in Rt's Google Maps® interface. The colors in the map correspond to the colors for different regions in Table 4-3. The map in the background is from Google (© 2012 Google).

In total, 31 regional models are used to model all municipalities that are shown in Figure 4-1. For some municipalities, there is more than one regional model, because they comprise more than one city or township. The City of Vancouver is a special case with 10 regional models, nine of which belong to downtown Vancouver in order to refine the modeling of this important area. Figure 4-4 displays all 31 regional models in the map interface in Rt, and Figure 4-5 shows the nine regional models for downtown Vancouver. The markers in these figures identify the centroid of the regions. The 31 regional models are listed in Table 4-3. This table presents for each region the total value of the building stock, the total land area, and the percentage of different zones as the ratio of the floor area of construction to the land area. Each percentage in this table is the mean of a lognormal random variable with a 20% coefficient of variation. According to Table 4-3, the City of Vancouver has the largest value of the building stock and Surrey has the largest land area. Table 4-4 presents the replacement cost per unit floor area of construction for each zone type. These costs are modeled as lognormal random variables for each region, with means and standard deviations that are shown in Table 4-4.

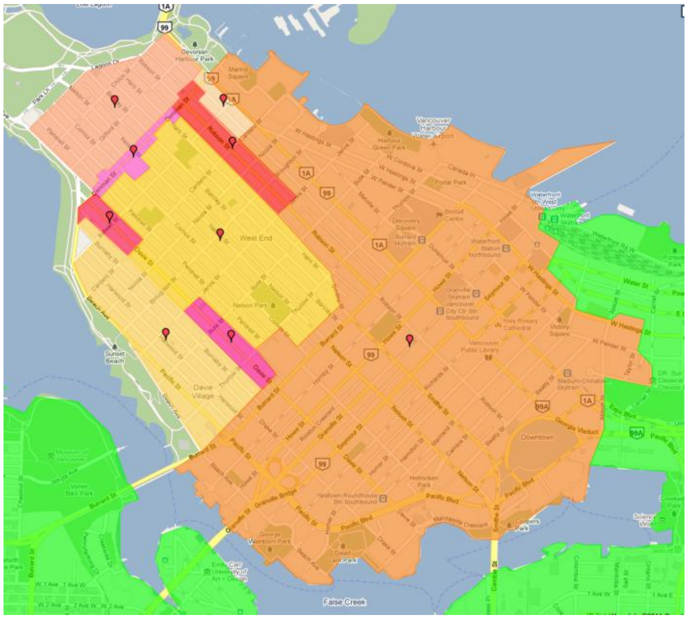


Figure 4-5: Zoomed view of regions modeled by the regional damage and loss models in downtown Vancouver. The colors in the map correspond to the colors for different regions in Table 4-3.

At the UBC campus, each building is placed in one of 13 occupancy classes using the survey data. In accordance with the previous chapter, each occupancy class is associated with a set of structural and non-structural costs per unit floor area, which are input to the building loss model. Table 4-5 presents these costs for all 13 occupancy classes. The table shows that medical buildings have the highest total unit cost, because of a very high cost of non-structural acceleration sensitive components. These components are mainly costly medical equipment. Structural and non-structural loss probabilities are discussed in detail later in this chapter.

Table 4-3: Regions modeled with the regional damage and loss models. The colors in the first column correspond to the colors for different regions in Figure 4-3 and Figure 4-4.

	Region	Total Value (Million $)	Area (m^2)	Zone Area Percentage (%)				
				Single-res.	Multi-res.	Commercial	Industrial	Comp. Devel.
🟦	Burnaby Region 1	50,123	$6.8 \cdot 10^7$	14	12	8	4	2
⬛	Burnaby Region 2	505	$4.2 \cdot 10^5$	0	0	60	0	0
🟫	Coquitlam	25,023	$3.9 \cdot 10^7$	16	11	5	2	1
🟥	Ladner	3,830	$7.1 \cdot 10^6$	17	8	5	2	0
🟪	Langley Region 1	9,163	$1.7 \cdot 10^7$	14	9	5	2	1
🟪	Langley Region 2	2388	$5.1 \cdot 10^6$	0	0	0	30	0
🟪	Langley Region 3	3556	$6.5 \cdot 10^6$	15	9	5	1	0
🟧	Langley Region 4	719	$1.6 \cdot 10^6$	14	6	4	1	0
🟪	Langley Region 5	2402	$4.4 \cdot 10^6$	15	8	5	2	0
🟦	Maple Ridge	12,757	$2.0 \cdot 10^7$	16	11	5	2	1
🟪	New Westminster	9,771	$1.8 \cdot 10^7$	14	9	5	2	1
🟦	North Vancouver	29955	$4.0 \cdot 10^7$	14	14	8	2	2
🟩	Pitt Meadows	2182	$4.0 \cdot 10^6$	15	9	5	2	0
🟩	Port Coquitlam	8617	$1.6 \cdot 10^7$	14	8	5	3	1
🟫	Port Moody	4737	$8.6 \cdot 10^6$	14	9	5	2	1
🟦	Richmond	30,298	$4.7 \cdot 10^7$	16	11	5	2	1
🟨	Surrey Region 1	75,114	$1.2 \cdot 10^8$	16	11	5	2	1
🟥	Surrey Region 2	10,395	$1.6 \cdot 10^7$	18	11	4	2	1
🟨	Tsawwassen	4,115	$7.6 \cdot 10^6$	17	8	5	2	0
🟥	Vancouver Region 1	118	$5.9 \cdot 10^4$	0	0	100	0	0
🟥	Vancouver Region 2	188	$9.4 \cdot 10^4$	0	0	100	0	0
🟪	Vancouver Region 3	80	$4.0 \cdot 10^4$	0	0	100	0	0
🟥	Vancouver Region 4	79	$3.9 \cdot 10^4$	0	0	100	0	0
🟨	Vancouver Region 5	1,283	$6.5 \cdot 10^5$	0	100	0	0	0
🟨	Vancouver Region 6	684	$3.4 \cdot 10^5$	0	100	0	0	0
🟧	Vancouver Region 7	75	$3.8 \cdot 10^4$	0	100	0	0	0
🟧	Vancouver Region 8	557	$2.8 \cdot 10^5$	0	100	0	0	0
🟧	Vancouver Region 9	4,625	$3.2 \cdot 10^6$	0	0	0	0	70
🟩	Vancouver Region 10	102,922	$1.1 \cdot 10^8$	13	23	10	2	4
🟥	West Vancouver	15,602	$2.4 \cdot 10^7$	14	12	5	2	2
🟩	White Rock	17,632	$3.2 \cdot 10^7$	15	9	4	2	1

Table 4-4: Replacement cost per unit floor area of construction for different zone types.

Zone Type	Mean ($/m^2)	St Dev. ($/m^2)
Single-residential	1,681	549
Multi-residential	1,986	363
Commercial	1,998	597
Industrial	1,557	409
Comprehensive Development	2,079	487

Table 4-5: Replacement cost per unit floor area for different buildings occupancy classes.

Occupancy Class	Total ($/m^2)	Structural ($/m^2)	Non-structural Drift ($/m^2)	Non-structural Acceleration ($/m^2)
Single Residence	1681	402	738	541
Multi Residence	1943	266	833	844
Institutional Residence	2104	391	850	863
Trade	1398	432	377	589
Services	2297	548	625	1124
Medical	2499	355	863	1281
Recreation	2143	238	758	1148
Parking	810	493	141	176
Industry	1506	236	178	1092
Agriculture	1215	561	94	560
Government	2194	364	735	1095
School	1908	361	929	618
University	2111	232	1267	612

4.6. HAZARD RESULTS

A seismic hazard analysis for the Vancouver metropolitan region is carried out, with a threefold objective: To illustrate the seismicity in the region; to compare the analysis results with seismic codes; and to validate the hazard modeling approach in this study. In accordance with the hazard models described earlier, there are six seismic sources in this analysis: three crustal, one subcrustal, and two subduction sources. One approach to obtain hazard curves, *i.e.*, the probability of exceeding spectral acceleration at different natural periods, is scenario sampling in Rt. This approach is different from traditional seismic hazard analysis (McGuire 2004), where conditional probabilities are used. A timespan of *T*=50 years is considered for this analysis, which makes the results directly comparable with seismic codes. Figure 4-6 shows the exceedance probability curves for spectral acceleration at natural periods of 0, 0.2, 0.5, 1.0 and 2.0 sec. These results are presented for a site in downtown Vancouver. The spectral acceleration at period equal to zero in Figure 4-6 is the peak ground acceleration, denoted by PGA in the figure.

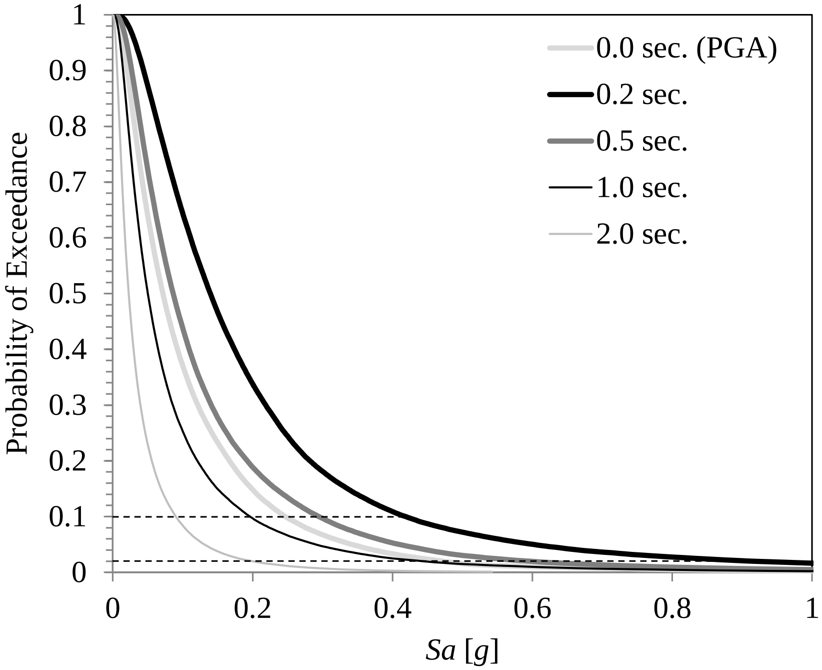


Figure 4-6: Probability of exceedance of spectral acceleration, *Sa*, at selected periods obtained by scenario sampling analysis in Rt. Dashed lines show the probability levels of 2% and 10% in 50 years.

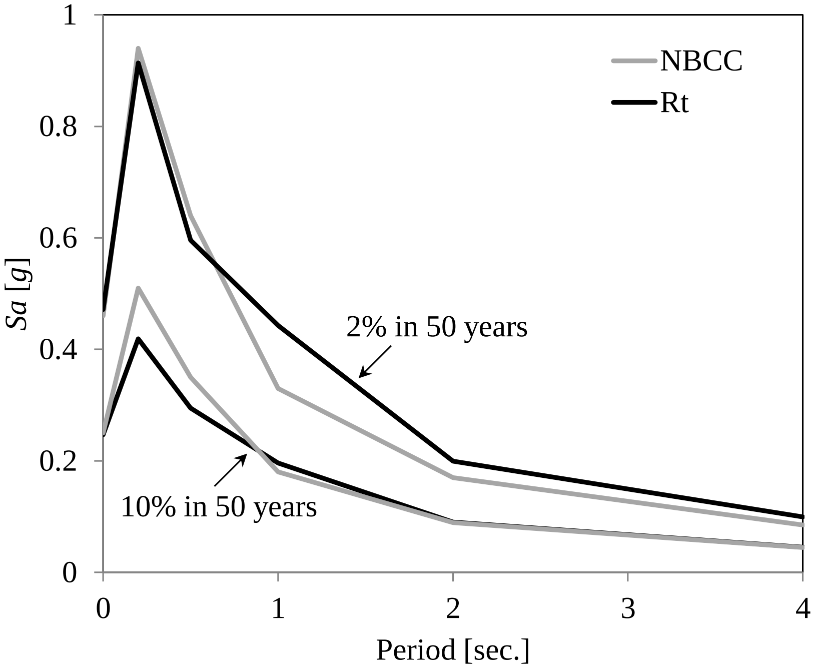


Figure 4-7: Comparison of the hazard spectra from NBCC (Adams and Halchuk 2003) and Rt.

The two dashed lines in Figure 4-6 indicate the 2% and 10% probabilities of exceedance in 50 years, obtained by Rt. By plotting spectral accelerations at these probability levels against the natural period, a hazard spectrum is constructed in Figure 4-7. For comparison, Figure 4-7 also shows the hazard spectrum for Vancouver from the National Building Code of Canada (NBCC) (Adams and Halchuk 2003). There is a reasonable agreement between the hazard spectrum from Rt and NBCC. The differences have several causes. First, each spectrum is computed using a different intensity model. The intensity models used in this study were presented earlier. In contrast, NBCC employs Boore *et al.* (1997) for crustal earthquakes and Youngs *et al.* (1997) for subcrustal and subduction earthquakes. Second, NBCC uses a deterministic magnitude of 8.2 for the subduction source, while this study models the subduction magnitude by random variables with realizations in the range of 8.0 to 9.2. Finally, in the NBCC approach, the probabilities 2% and 10% in 50 years describe only the uncertainty in the occurrence of different magnitudes. NBCC addresses other uncertainties, such as the model error in the intensity model, separately by computing the median and the standard deviation of the spectral acceleration. In contrast, this study also incorporates uncertainty in earthquake magnitude, location, and intensity models. *I.e.*, the uncertainty in the shear wave velocity, maximum magnitude, ground shaking model error, *etc.*, all contribute to the shown probabilities from Rt.

4.7. LOSS RESULTS

This section presents the results of the seismic loss analysis. First, the loss probabilities for the entire Vancouver metropolitan region are computed by the two analysis options described earlier and compared thereafter. Next, the influence of different hazard sources and the influence of the analysis timespan on loss probabilities are assessed. Thereafter, the vulnerability of different municipalities within the considered region is compared. Finally, the losses of buildings on the UBC campus are studied in detail.

4.7.1. Scenario Sampling

Potential seismic losses in the Vancouver metropolitan region are first sought by a scenario sampling analysis. Figure 4-8 shows the relative frequency diagram (comparable to a PDF) for the total loss in the Vancouver metropolitan region in a 50-year period. The uncertainty in the occurrence of earthquakes is here included. The PDF in Figure 4-8 is the result of 100,000 samples; *i.e.*, 100,000 scenarios are generated for the 50-year period. It is observed that the PDF has a skewed shape and decays exponentially because more dramatic earthquakes are also more infrequent. From scenario sampling, the mean loss over the 50-year period is $24.1 billion with a standard deviation equal to $81.0 billion. The mean loss is higher than what appears from the PDF in Figure 4-8, because the upper tail of the PDF has a significant probability density, as explained shortly. These values are in present-value (year 2012) Canadian dollars. The solid line in Figure 4-9 shows the same result as in Figure 4-8, in the form of the complementary CDF, which is essentially the loss curve. It identifies the probability of exceeding any loss value. For example, it is read from the solid curve in Figure 4-9 that there is a 5.6% chance of exceeding $100 billion in losses during the 50-year period. This emphasizes that the upper tail of the relative frequency diagram in Figure 4-8 has a significant probability density.

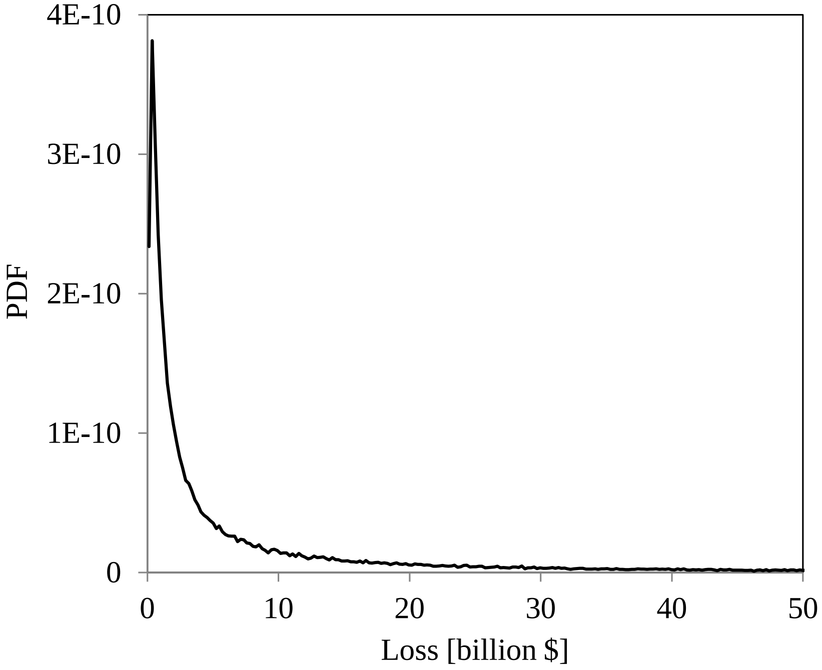


Figure 4-8: Relative frequency diagram for losses, computed by scenario sampling.

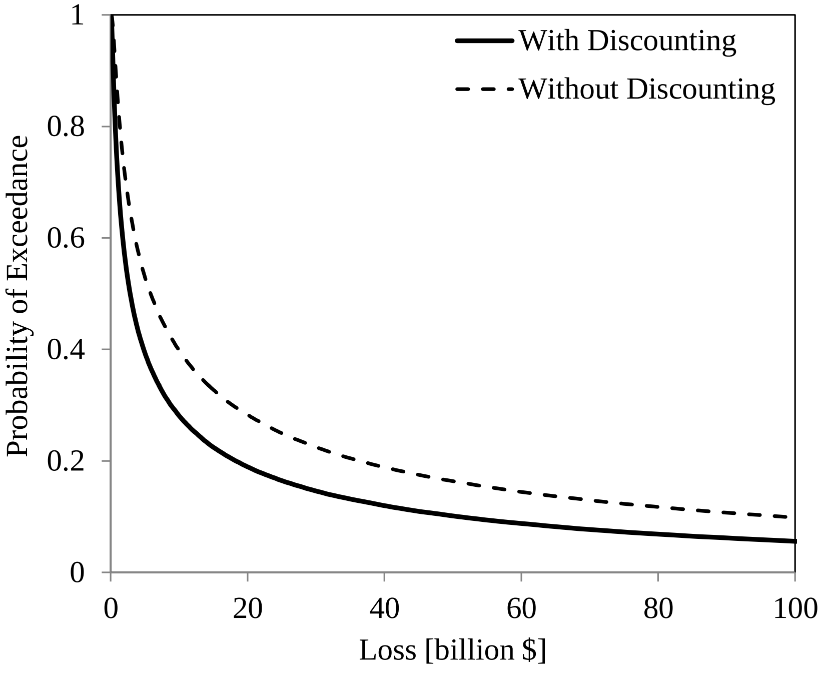


Figure 4-9: Loss curve computed by scenario sampling.

Figure 4-9 is also employed to show the loss curve without discounting, displayed as a dashed curve. Clearly, discounting future losses to present value has a significant impact on the loss estimates, even with a real interest rate as low as 3%. For example, the probability of exceeding $100 billion in losses in the 50-year period is 9.8% when discounting is omitted. By comparison, the mean value of the total value at stake in the Vancouver metropolitan region is nearly $434 billion. This is a random variable due to uncertainties in construction areas and costs.

4.7.2. FORM

This section employs the FORM-based analysis option to compute loss probabilities. It is selected to compute the probability that the seismic loss in the Vancouver metropolitan region exceeds $100 billion in a 50-year period. The limit-state function for this analysis reads

$$g = 100,000,000,000 - \left(\sum_{i=1}^{31} l_i + \sum_{i=1}^{622} l_i \right) \tag{4-4}$$

The first term in parentheses sums up the loss of the 31 regional loss models, while the second term sums up the loss of the 622 buildings on the UBC campus. The result of the reliability analysis with Eq. (4-4) is one point on the loss curve. A FORM analysis is first carried out individually for each hazard source with the limit-state function in Eq. (4-4). Below, the results of FORM are verified by importance sampling. FORM provides reasonably accurate probability estimates for all sources except the CASR source. To understand the reason for the discrepancy, the limit-state function in Eq. (4-4) is plotted in Figure 4-10 against the two surface location random variables of the CASR location model. The plot is made in the space of standard normal variables. Figure 4-11 displays the contours of g in the same space, where the contour that corresponds to g=0, *i.e.*, the limit-state surface, is highlighted as a curved thick line. The thick straight line in both figures represents the linear approximation in FORM. Significant nonlinearity in the limit-state function is observed in these figures. The reason is understood by first noting that the CASR source encompasses the Vancouver region. In turn, this means that the CASR source can produce earthquake locations very close to Vancouver. As a result, earthquake locations from the CASR source are associated with very different values of spectral acceleration, and thus loss. Hence, large nonlinearity is observed in Figure 4-11. This problem is addressed by SORM in the following.

4.7.3. SORM and Importance Sampling

SORM analysis is carried out to improve the FORM estimates, which is particularly important for the CASR source as noted above. In this study, only the first principal curvature of the limit-state surface is employed in SORM, because it is readily available from the FORM analysis (Der Kiureghian 2005). This provided good results for all cases considered here. In fact, SORM provides a second-order approximation that closely matches the curved thick line in Figure 4-11 for the CASR source. For this source, SORM yields a 0.35% chance of exceeding a $100 billion loss. This is reasonably close to the 0.57% result from the more accurate importance sampling. In contrast, FORM resulted in the rather inaccurate value 6.6%.

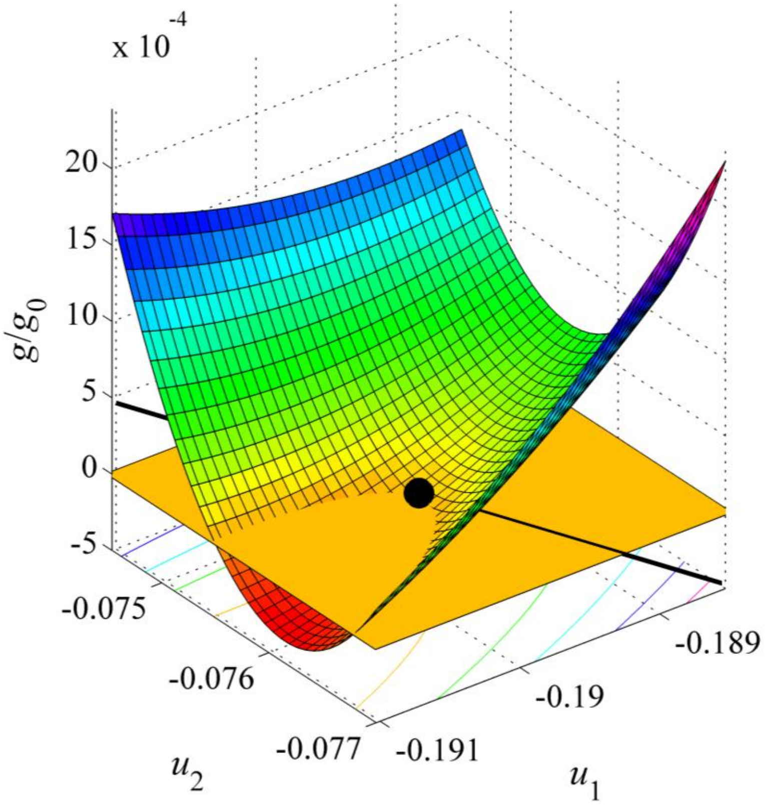


Figure 4-10: Visualization of the limit-state function in Eq. (4-4), normalized by a scaling factor g_0, near the design point (marked with dot). u_1 and u_2 represent the two surface location variables for the CASR source.

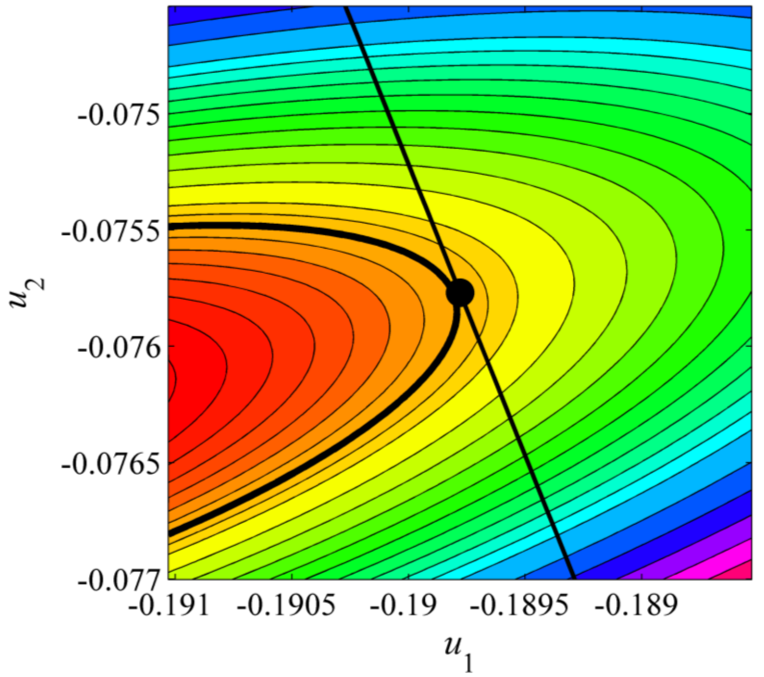


Figure 4-11: Contours of the limit-state function in Eq. (4-4) near the design point (marked with dot). The thick curved line identifies the limit-state surface g=0, while the straight line is the FORM approximation.

Importance sampling is another relatively efficient approach for improving the FORM results. Although it increases the computational cost, it is far more efficient than mean-centered Monte Carlo sampling. Table 4-6 quantifies the improvement in the FORM results due to importance sampling with a coefficient variation of 2%. Although the FORM results are quite accurate, except for the CASR source as noted above, the importance sampling results are employed in the following. Table 4-6 also shows that SORM analysis based on the first principal curvature yields results that are close to the sampling results, which are employed as the benchmark solution. With the SORM results from Table 4-6 in Eq. (4-3), there is 8.3% chance of exceeding a $100 billion loss in the 50-year period. When the more accurate importance sampling results are utilized, the probability changes slightly, to 8.6%. In the following, this is referred to as the “FORM-based” result, because both SORM and importance sampling require prior FORM analysis.

Table 4-6: Probability of exceeding a $100 billion loss for each earthquake source.

Source	FORM	SORM	Importance Sampling
CASR	0.0667	0.0035	0.0057
JDFF	0.0014	0.0010	0.0011
JDFN	0.0022	0.0026	0.0017
GSP	0.0111	0.0091	0.0053
Subduction Source 1	0.0803	0.0719	0.0660
Subduction Source 2	0.0923	0.1026	0.1128

4.7.4. Comparison of Analysis Options

The results from FORM-based analysis are fairly close to those obtained by scenario sampling for the $100 billion loss threshold. However, when making this comparison, it is noted that scenario sampling addresses the accumulated discounted losses over 50 years, which may include more than one seismic event. To make the two analysis approaches comparable, the scenario sampling is modified to sample the maximum loss, without discounting. Then, the scenario sampling result changes from 5.6% to 4.4% chance of exceeding a $100 billion loss. This is the result that should be compared with the results from FORM-based analysis, which is 8.6%. These two probabilities, *i.e.*, 8.6% from FORM-based analysis and 4.4% from scenario sampling analysis, are identified by solid circles in Figure 4-12, which displays the loss curves obtained by the two analysis approaches. The FORM-based curve is obtained by reliability analysis with different thresholds in the limit-state function in Eq. (4-4). Each threshold yields a point on the loss curve, shown by hollow circles in Figure 4-12. The figure shows that the accuracy of the FORM-based results increases as the threshold increases, as explained after Eq. (4-3).

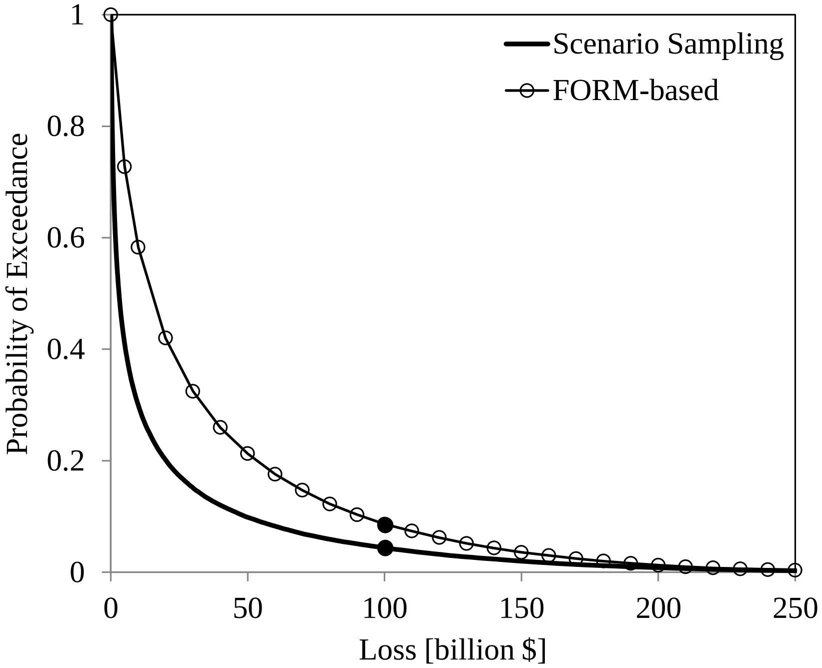


Figure 4-12: Comparison of loss curves computed by scenario sampling and FORM-based analyses.

To compare the computational cost of scenario sampling and FORM-based analysis, the limit-state function in Eq. (4-4) is reconsidered. The CPU time to obtain FORM/SORM/importance sampling results for this threshold is 38 minutes. In contrast, scenario sampling to achieve a 2% coefficient of variation of the probability results requires 2.7 hours, *i.e.*, more than four times the cost of FORM-based analysis. These numbers vary with the threshold in the limit-state function; the computational cost of scenario sampling increases drastically when the probability of higher losses is sought. For example, the CPU time of scenario sampling to compute the probability of exceeding a $250 billion loss is 46 hours. The computational cost of scenario sampling also increases with the time period under consideration. For instance, the CPU time of scenario sampling with the limit-state function in Eq. (4-4) increases by more than 50% if a time period of 200 years is considered instead of 50 years. Conversely, the CPU time of the FORM-based analysis is independent of the threshold in Eq. (4-4) and the time period under consideration. In conclusion, the scenario sampling approach is flexible, powerful, and conceptually straightforward. However, it is computationally costly, especially in the tail of the loss curve. In contrast, the FORM-based approach is computationally efficient and provides accurate results in the tail, but does not allow time-varying phenomena, such as long-term deterioration.

4.7.5. Ranking of Hazards

With FORM-based analysis, it is possible to identify the contribution from each hazard to the overall loss probabilities. For this purpose, Table 4-6 provides the disaggregation of the probability of exceeding $100 billion for earthquakes from the six sources identified earlier. At first, Table 4-6 gives the impression that subduction earthquakes are the dominant source, because it is associated with a high loss probability. However, Table 4-6 does not contain information about the rate of occurrence of each earthquake. To extract actual contribution of each source, the probabilities in Table 4-6 are multiplied by the corresponding occurrence rates. The results are shown in Table 4-7, which shows that the CASR crustal source governs the probability of exceeding a $100 billion loss, followed by the GSP subcrustal and the subduction sources.

Table 4-7: Annual rate of exceedance due to each earthquake source.

Source	Annual Rate of Exceedance
CASR	0.0010736
GSP	0.0004981
Subduction	0.0002181
JDFF	0.0000090
JDFN	0.0000059

4.7.6. Influence of Timespan

The results that are presented up to this point are for a time period of *T*=50 years. To illustrate the influence of *T* on the results, Figure 4-13 compares the loss curves computed for *T*=1, 5, 10, 20, 50, 100, and 200 years. The curve associated with *T*=1 shows the annual loss probabilities. Naturally, the loss probabilities increase when a longer time period is considered, because the number of earthquake events increase. However, it is observed in Figure 4-13 that the difference between the loss curves decreases as *T* increases, *e.g.*, the loss curves for *T*=100 and *T*=200 years are close. This is because the present values of loss decay exponentially with time due to discounting. Hence, the losses that are incurred in far future have small present values.

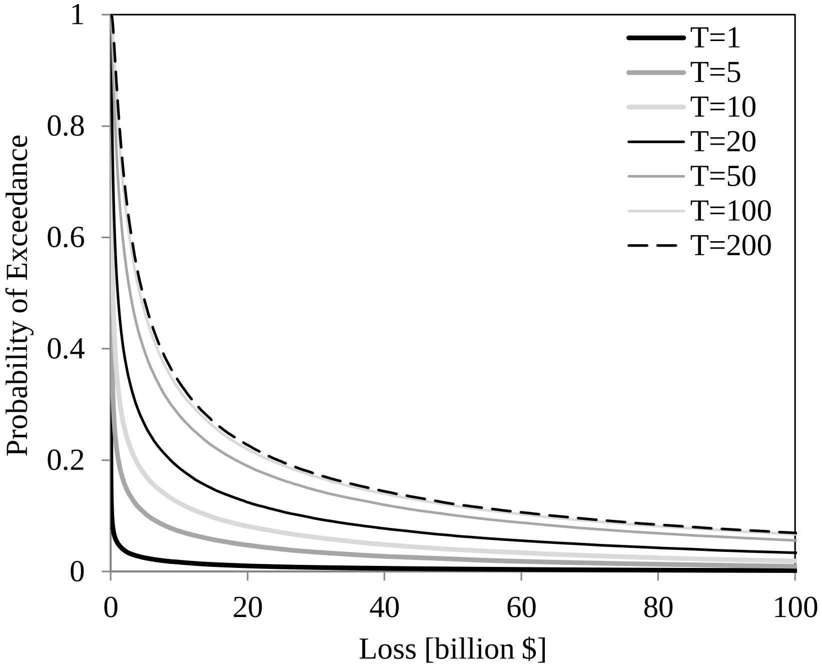


Figure 4-13: Comparison of loss curves for time periods of 1, 5, 10, 20, 50, 100, and 200 years.

4.7.7. Vulnerability of Different Municipalities

To determine the relative vulnerability of the municipalities in the Vancouver metropolitan region, a municipality damage ratio, η_M, is defined. For each municipality, it equals the accumulated loss during *T*, divided by the mean total value at stake in that municipality. For instance, for the City of Vancouver, the sum of the losses of its 10 regional models over 50 years is divided by $110.6 billion. With this definition, a scenario sampling is carried out and η_M-values for individual municipalities are recorded. The resulting probability distributions for η_M are shown in Figure 4-14 for selected municipalities. Each point on these curves represents the probability of exceeding the corresponding η_M-value. The probability axis is shown in logarithmic scale to highlight the tail probabilities. Note that η_M may take a value beyond 1.0 because it is the sum of losses over time. From Figure 4-14, it is seen that Richmond is the most vulnerable municipality, followed by Vancouver. In contrast, Port Moody and Langley are least vulnerable in this analysis. The results are reasonable because the five most vulnerable municipalities, *i.e.*, Richmond, Vancouver, Tsawwassen, Ladner, and White Rock coincide with the GSP subcrustal source, contrary to other municipalities. In addition, the cities of Richmond, Ladner, and Tsawwassen are located on soft soil, *i.e.*, soil with low V_{S30}, which contributes to high loss probabilities. This amplifies the spectral acceleration as understood from Figure 3-6. In turn, this increases the damage. Conversely, the least vulnerable municipalities, *i.e.*, Port Moody, Langley, and Maple Ridge, are relatively far from most of the earthquake sources.

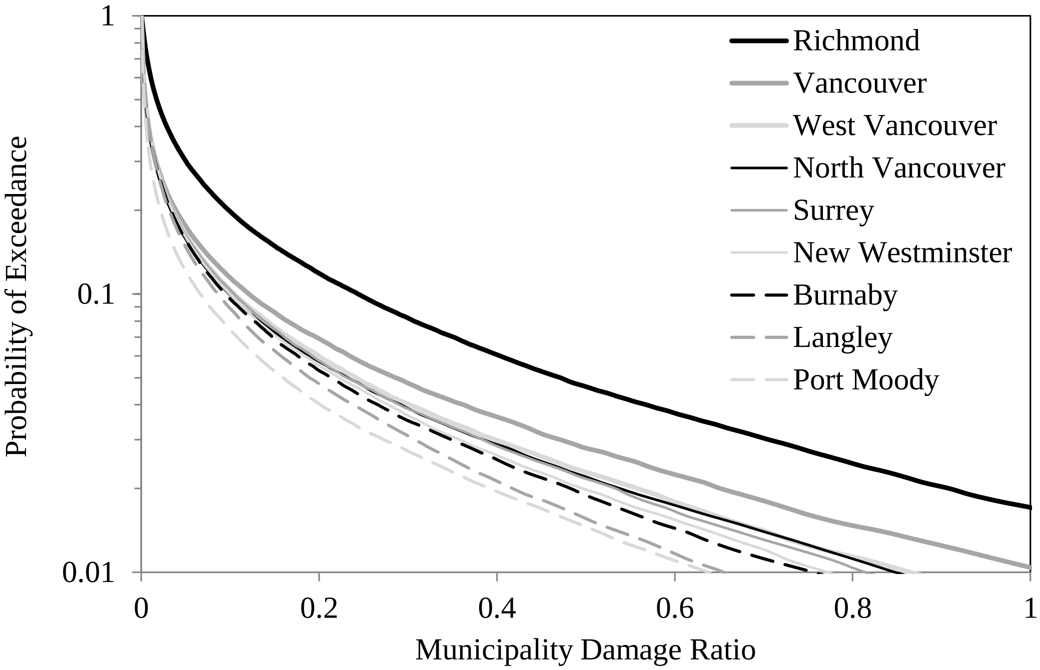


Figure 4-14: Regional damage ratio probabilities for selected municipalities.

4.7.8. Annual Losses

The phrase “return period loss” is often used, particularly in the insurance industry. Return period loss is essentially an alternative way of presenting the loss curve. It is here demonstrated how return period losses can be extracted from the previously presented results. The annual loss curve was shown in Figure 4-13, identified by the label T=1. From that curve, the results in Table

4-8 are extracted. Specifically, the return period losses associated with 50, 100, 200, 250, 500, and 1,000 years are presented. These are obtained by extracting the loss associated with an annual probability of exceedance equal to the inverse of the return period. For instance, the annual loss for the return period of 1,000 years is the loss with an exceedance probability of 1/1,000 on the curve for *T*=1 in Figure 4-13. Table 4-8 also shows the annual loss for individual municipalities in the region. It is observed that the City of Vancouver has the largest contribution to the annual loss of the region. This is reasonable because this municipality has the largest value of the building stock according to Table 4-3, and is the second most vulnerable municipality in the region according to Figure 4-14.

Table 4-8: Annual loss in million dollars for different return periods.

	Annual Loss (million $)					
Return Period (Years)	50	100	200	250	500	1,000
Entire Region	7,322	20,176	45,710	56,624	91,945	132,232
Burnaby	640	1,843	4,295	5,280	9,470	13,804
Coquitlam	377	1,079	2,412	2,966	5,036	7,380
Ladner	69	190	418	502	841	1,198
Langley	209	624	1,376	1,667	2,760	4,126
Maple Ridge	146	445	1,004	1,257	2,224	3,281
New Westminster	128	384	866	1,070	1,813	2,630
North Vancouver	400	1,204	2,877	3,616	5,981	8,927
Pitt Meadows	32	92	196	244	398	593
Port Coquitlam	118	342	731	905	1,587	2,269
Port Moody	38	123	303	386	666	1,060
Richmond	1,188	2,969	5,751	6,659	9,934	13,721
Surrey	1,158	3,510	7,961	9,730	16,946	23,821
Tsawwassen	76	217	456	575	902	1,287
Vancouver	1,818	5,632	13,145	16,497	29,120	41,994
West Vancouver	213	638	1,495	1,954	3,405	4,929
White Rock	256	771	1,710	2,121	3,663	5,285

4.7.9. Detailed Study of the UBC Campus

A detailed study of the losses at the UBC campus is possible because each building in this region is modeled individually. First, it is of interest to compare the results from these building models with those obtained by regional models. Figure 4-15 compares three loss curves: 1) One that models the entire UBC campus with regional damage and loss models; 2) One that models each of the 622 buildings at UBC by building response, damage, and loss models; and 3) One that analyzes each building by the capacity spectrum method and the damage fragility and loss functions in FEMA-NIBS (2003), which are implemented in Rt. The hazard models are the same in all three cases. Figure 4-15 shows that the three loss curves are quite close, particularly at high loss thresholds.

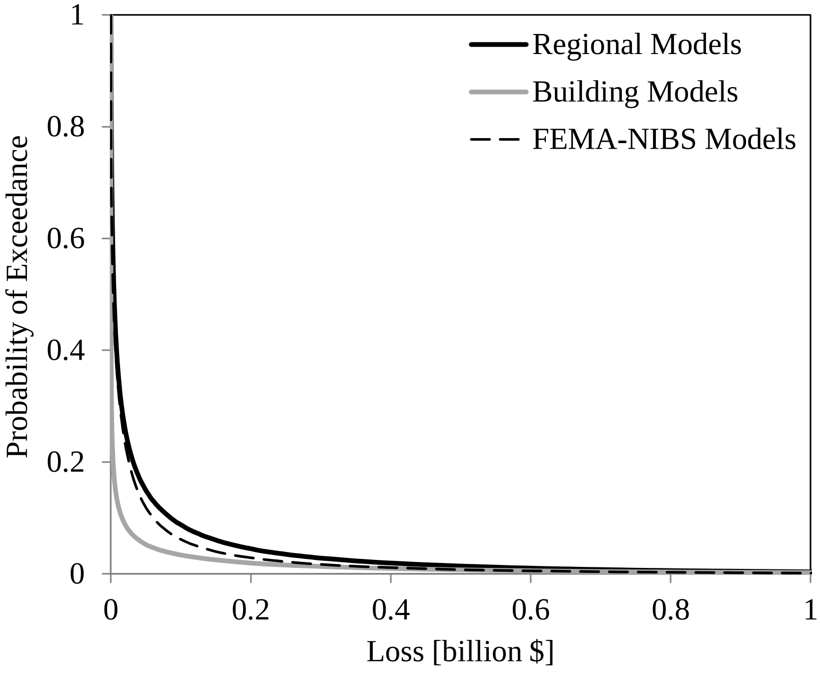


Figure 4-15: Loss curves for UBC that are computed using regional models, building models, and the FEMA-NIBS method.

The contribution of structural and non-structural components of the UBC buildings to overall losses is illustrated in Figure 4-16. A logarithmic scale is employed to highlight the tail probabilities. Figure 4-16 shows that the probability of exceedance for non-structural losses, *i.e.*, the sum of acceleration- and drift-sensitive losses, is higher than that of structural losses. This has been observed in several studies, see, *e.g.*, FEMA-NIBS (2003). In turn, non-structural drift-sensitive components contribute more to the loss probabilities than acceleration-sensitive components.

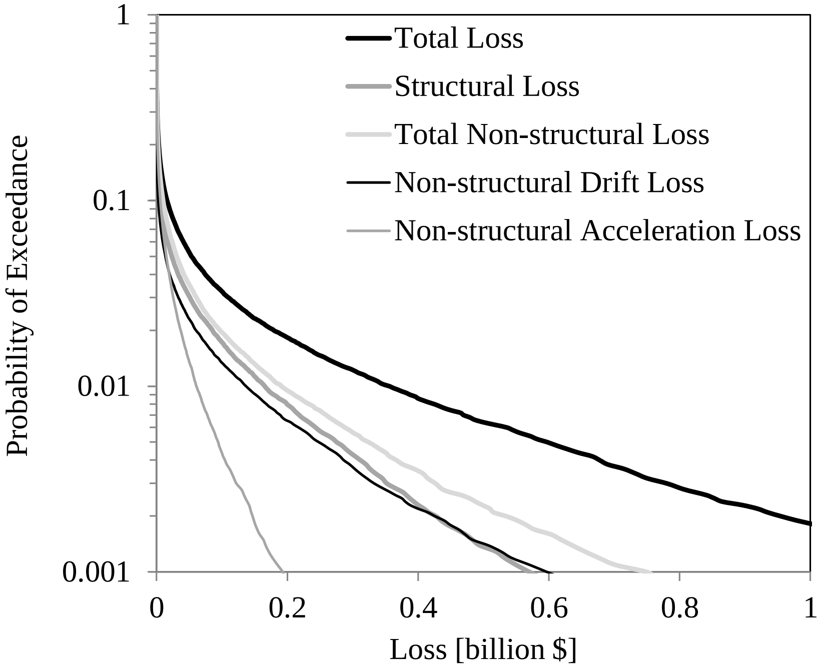


Figure 4-16: Structural and non-structural loss probabilities.

The performance of different structural systems and construction materials is assessed to determine the relative vulnerability of different building prototypes. For this purpose, a prototype damage ratio is defined. For each prototype, it equals the total loss of all buildings with that prototype divided by their total value. Figure 4-17 presents the probability of exceeding damage ratios for *T*=50 years for different prototypes. This figure indicates that unreinforced masonry buildings are the most vulnerable, while concrete shear wall building are the least. Note that this ranking is specific to the composition of buildings at UBC. For example, it is observed in Figure

4-17 that concrete moment frame is ranked amongst the vulnerable prototypes. This is because the buildings of this type at the UBC campus are mostly non-ductile. In fact, more than 80% of these concrete moment frame buildings are built before 1975, which places them in low-code and pre-code categories. Conversely, most of the modern concrete buildings at UBC are shear wall buildings, which exhibit the best performance according to Figure 4-17. Furthermore, Figure 4-17 indicates that reinforced masonry buildings at the UBC campus exhibit a good performance, in fact almost as good as concrete shear wall buildings. The reason for this surprising finding is that 95% of the buildings of this type on campus belong to the moderate code level, *i.e.*, they are built in accordance with modern seismic standards.

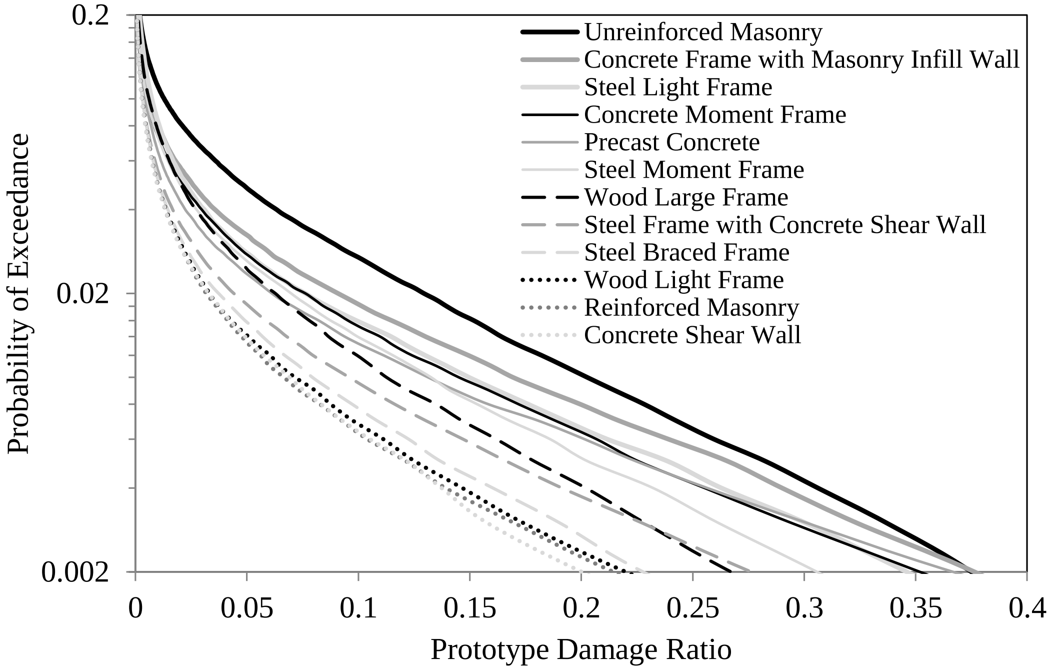


Figure 4-17: Damage ratio probabilities for different building prototypes.

4.8. CONCLUSIONS

This chapter presents a comprehensive seismic risk analysis of the Vancouver metropolitan region in Canada. Generic models for earthquakes and buildings were presented in Chapter 3, and these models are employed in this chapter to predict the seismic losses in the region. The analyses are carried out by Rt. Two analysis approaches are employed; scenario sampling is conceptually straightforward and flexible, but computationally costly. The alternative approach employs FORM, SORM, and importance sampling, which is computationally efficient but limits the inclusion of time-varying phenomena in the multi-hazard analysis presented here. The computed loss probabilities, computational cost values, and error estimates suggest that both analysis approaches are viable. The risk analysis reveals that Richmond is the most vulnerable city in the region. It is also found that crustal earthquakes have the most influence on the loss. Furthermore, the analysis suggests that non-structural components contribute more to the losses than structural components, and that unreinforced masonry buildings are amongst the most vulnerable structural systems in the considered region.

Chapter 5. SENSITIVITY MEASURES FOR OPTIMAL MITIGATION OF RISK AND REDUCTION OF MODEL UNCERTAINTY

5.1. INTRODUCTION

The primary objective in this chapter is to guide the allocation of resources for civil infrastructure subjected to multiple hazards. This addresses one of the great challenges faced by the modern society: Limited resources must be prioritized to mitigate risk and/or improve the understanding of risk. To this end, three different but complementary questions are asked and addressed in this chapter: 1) Which infrastructure components should be prioritized for seismic retrofit to mitigate seismic risk? 2) Which infrastructure components should be subjected to detailed modeling to reduce the epistemic uncertainty in order to improve the quality of the risk analysis? 3) Which models should be prioritized in research to reduce the epistemic uncertainty to improve the quality of future risk analyses? All three questions deal with the general problem of distributing limited resources in an optimal manner under conditions of uncertainty. Therefore, although the application in this chapter is specific to structural earthquake engineering, the developments are intended to be broadly applicable.

In this study, risk is quantified by cost probabilities. The cost has two causes: repair of damage after an earthquake and cost of *a priori* retrofit actions. Thus, the primary result from the risk analysis is the cost exceedance probability curve, or selected points on it. A version of this curve that omits retrofit cost is sometimes referred to as a “loss curve,” which forms an important basis for decision-making in the insurance industry (Grossi and Patel 2005). Loss curves are also central in contemporary performance-based earthquake engineering (Yang *et al.* 2009).

The vehicle for the developments in this study is the computation of cost exceedance probability curves by reliability analysis. This requires the specification of a limit-state function, g, because reliability methods are designed to estimate the probability that $g \leq 0$. Therefore, to compute the probability that the total cost, c, exceeds a threshold, c_t, the following limit-state function is specified:

$$g = g\left(\boldsymbol{\theta}, \mathbf{x}, \mathbf{v}\right) = c_t - c\left(\boldsymbol{\theta}, \mathbf{x}, \mathbf{v}\right) \tag{5-1}$$

where $\boldsymbol{\theta}$=vector of “epistemic random variables,” $\mathbf{x}$=vector of “aleatory random variables,” and $\mathbf{v}$=vector of decision variables that are at the discretion of the decision maker. The phrase epistemic is employed to identify random variables that have a probability distribution that changes when new data or new research emerge. In contrast, aleatory random variables are intended to represent irreducible uncertainty. A central vision behind this thesis is to establish a framework of probabilistic models that explicitly include the characterization of epistemic uncertainty. This permits targeted efforts to reduce that uncertainty. In fact, it provides a rational basis for allocating resources to gather data and build better models, which ultimately fosters improved decisions to mitigate risk.

In this study, many probabilistic models are utilized to evaluate the total cost in Eq. (5-1). The models represent earthquake characteristics, building response, building damage, and repair costs. These models were presented in Chapter 3. The risk analysis and the sensitivity analysis are carried out by Rt, which was presented in Chapter 2.

This chapter starts with a brief overview of the models that are utilized, and the analysis approach. Next, three sensitivity measures are presented, which address the three questions posed earlier. Each sensitivity measure is utilized in a comprehensive analysis of the 622 buildings on the campus of UBC in Vancouver, which is subjected to several sources of seismicity.

5.2. MODELS

The approach adopted in this thesis has two components: probabilistic models and reliability methods. In contrast with many contemporary seismic risk analysis approaches, the present approach circumvents conditional probability models in favour of simulation-type models that produce scalars or vectors of physical responses. This is necessary in order to evaluate Eq. (5-1). In particular, the models in this study employ random variables to discretize the uncertainty. A simple but instructive model is the linear regression model

$$y = \theta_1 + \theta_2 \cdot h_2(\mathbf{x}) + \theta_3 \cdot h_3(\mathbf{x}) + \cdots + \varepsilon \tag{5-2}$$

where y=model response, θ_i=model parameters, $h_i(\mathbf{x})$=explanatory functions, and ε=zero-mean normally distributed model error. In the Bayesian approach to linear regression, the parameters θ_i, as well as the standard deviation of ε, denoted by σ_ε, are random variables. This approach is adopted here, and the model parameters are categorized as epistemic random variables, *i.e.*, $\boldsymbol{\theta}=\{\theta_1, \theta_2, \ldots, \varepsilon, \sigma_\varepsilon\}$. Importantly, their probability distribution is affected by model improvement efforts, typically data-gathering. Box and Tiao (1992), Gardoni *et al.* (2002), and others describe the statistical inference to obtain the probability distribution for $\boldsymbol{\theta}$. The Bayesian philosophy, in which the model uncertainty is explicitly included, is employed throughout this study, regardless of model form.

Table 5-1 provides an overview of the regression models that are utilized in the numerical example. They predict the intensity of ground shaking and the response and damage of buildings. Some of the models are from the literature, while others are created in-house. All these models were described in detail in Chapter 3. The two first columns in Table 5-1 provide the name and formulation of each model. The third column shows the number of instances of each model in the numerical example. The last two columns in Table 5-1 show the number of input variables that each model takes. Specifically, the penultimate column displays the number of physical variables, comprising **x** and **v**, and the last column displays the number of model variables, $\boldsymbol{\theta}$.

Table 5-1: Overview of the regression models employed in the numerical example.

Model Name	Formulation	Instances	Physical Variables	Model Variables
Concrete Frame with Masonry Infill Wall Structural Damage	Nonlinear Regression	10	7	9
Concrete Frame with Masonry Infill Wall Structural Response	Linear Regression	10	4	25
Concrete Moment Frame Structural Damage	Nonlinear Regression	22	8	10
Concrete Moment Frame Structural Response	Linear Regression	22	4	25
Concrete Shear Wall Structural Damage	Nonlinear Regression	134	7	9
Concrete Shear Wall Structural Response	Linear Regression	134	4	25
Crustal Ground Shaking Intensity	Algorithm based on Regression	1	4	1
Non-Structural Acceleration Damage	Nonlinear Regression	622	2	3
Non-Structural Drift Damage	Nonlinear Regression	622	2	3
Precast Concrete Structural Damage	Nonlinear Regression	11	8	10
Precast Concrete Structural Response	Linear Regression	11	4	25
Reinforced Masonry Structural Damage	Nonlinear Regression	58	8	10
Reinforced Masonry Structural Response	Linear Regression	58	4	25
Steel Braced Frame Structural Damage	Nonlinear Regression	5	7	9
Steel Braced Frame Structural Response	Linear Regression	5	4	25
Steel Frame with Concrete Shear Wall Structural Damage	Nonlinear Regression	6	6	8
Steel Frame with Concrete Shear Wall Structural Response	Linear Regression	6	4	25
Steel Frame with Masonry Infill Wall Structural Damage	Nonlinear Regression	2	6	8
Steel Frame with Masonry Infill Wall Structural Response	Linear Regression	2	4	25
Steel Light Frame Structural Damage	Nonlinear Regression	22	6	8
Steel Light Frame Structural Response	Linear Regression	22	4	25
Steel Moment Frame Structural Damage	Nonlinear Regression	4	7	9
Steel Moment Frame Structural Response	Linear Regression	4	4	25
Subcrustal Ground Shaking Intensity	Algorithm based on Regression	1	4	1
Subduction Ground Shaking Intensity	Algorithm based on Regression	1	4	1
Unreinforced Masonry Structural Damage	Nonlinear Regression	14	6	8
Unreinforced Masonry Structural Response	Linear Regression	14	4	25
Wood Large Frame Structural Damage	Nonlinear Regression	128	5	7
Wood Large Frame Structural Response	Linear Regression	128	4	25
Wood Light Frame Structural Damage	Nonlinear Regression	206	5	7
Wood Light Frame Structural Response	Linear Regression	206	4	25

The models in Table 5-1 are employed in reliability analysis to compute cost exceedance probabilities for the UBC campus. Figure 5-1 pinpoints this region in the Google Maps® interface in Rt. The markers in the zoomed map of the UBC campus identify the 622 considered buildings. 26 of the buildings are numbered in Figure 5-1 because they appear prominently in the rankings that are presented later. To provide an outline of the information that is available for each building, Table 5-2 displays selected information for the 26 buildings that are numbered in Figure 5-1.

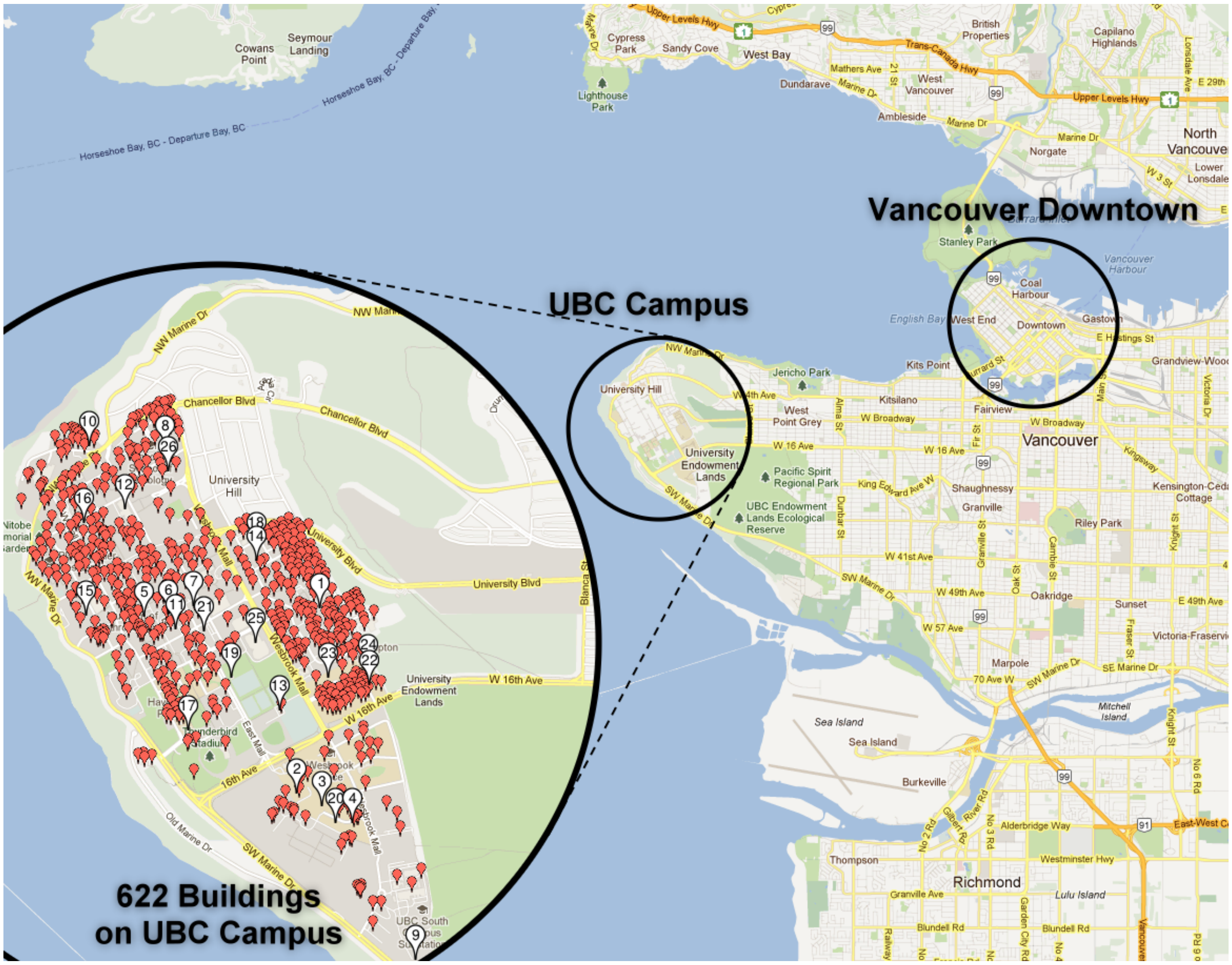


Figure 5-1: Map of the UBC campus and the 622 buildings that are modeled in this study. Buildings in Table 5-2 are identified by their number on the zoomed map. The map in the background is from Google (© 2012 Google).

Table 5-2: Information for selected buildings among the 622 buildings on the UBC campus.

No.	Building Name	Footprint Area (m^2)	Number of Stories	Year Erected	Mean Total Value ($)	Code Level	Longitude	Latitude
1	Acadia Park Highrise	1,204	15	1967	30,358,860	Moderate	−123.2368	49.2624
2	Animal Care Rodent Breeding	538	1	1981	653,136	Moderate	−123.2387	49.2525
3	Animal Science Aquaculture Centre	1,746	1	1969	4,010,562	Low	−123.2366	49.2519
4	Animal Science Main Sheep Unit	552	1	1976	671,109	Moderate	−123.2341	49.2509
5	Barn Coffee Shop	267	1	1925	572,448	Pre	−123.2511	49.2619
6	Cheeze Factory Undergrad Society	228	1	1930	488,832	Pre	−123.2492	49.2620
7	ChemBio Engineering	4,244	6	2004	58,490,808	Moderate	−123.2471	49.2625
8	Corus	747	13	2004	18,874,990	Moderate	−123.2494	49.2708
9	Environmental Services Facility	103	1	1969	236,591	Low	−123.2290	49.2437
10	Green College Kitchen	54	1	1992	115,776	Moderate	−123.2556	49.2710
11	ICICS Main	1,550	6	2003	21,362,100	Moderate	−123.2486	49.2613
12	Irving Barber Library	6,723	5	2008	77,208,257	Moderate	−123.2527	49.2677
13	John Owen Pavilion	856	1	1967	1,834,949	Low	−123.2401	49.2570
14	Mackenzie House	988	13	1990	24,962,965	Moderate	−123.2419	49.2649
15	Marine Drive Tower 1	1,148	18	2005	40,144,586	Moderate	−123.2559	49.2620
16	Old Auditorium	1,075	1	1925	2,304,800	Pre	−123.2561	49.2669
17	Promontory	860	16	2004	26,735,680	Moderate	−123.2475	49.2559
18	Ross House	853	13	1990	21,545,927	Moderate	−123.2419	49.2656
19	Rugby Pavillion	435	1	1963	933,139	Low	−123.2440	49.2587
20	Sherwood Building	251	1	1970	576,547	Low	−123.2349	49.2509
21	Technology Enterprise Facility	1,880	6	2001	25,910,160	Moderate	−123.2463	49.2611
22	The Balmoral	944	16	1989	29,361,057	Moderate	−123.2328	49.2585
23	The Chatham	2,682	19	1989	99,006,814	Moderate	−123.2362	49.2587
24	The Regency	951	17	1989	31,425,237	Moderate	−123.2329	49.2591
25	Thunderbird Sports Centre	10,618	1	1968	22,764,992	Low	−123.2421	49.2605
26	Walter Gage East Tower	643	18	1972	22,488,282	Low	−123.2491	49.2696
…	…	…	…	…	…	…	…	…

The UBC campus is subjected to three sources of seismicity: Shallow crustal earthquakes, deep subcrustal earthquakes, and megathrust subduction earthquakes. The first two are modeled as area sources, while subduction earthquakes originate from a line source (Adams and Halchuk 2003). Figure 5-2 shows the location of the earthquake sources relative to the UBC campus. In this figure, area sources are divided into several sub-areas. Specifically, the crustal earthquake source is divided into six area sources, while the subcrustal area source is divided into three area sources. This is done because FORM analysis is employed, as described shortly, which requires the limit-state function in Eq. (5-1) to be continuously differentiable and relatively linear in the space of random variables. This is achieved by the subdivision of the area sources in Figure 5-2. It is also noted that the subduction source is divided into two sources for reasons that were discussed in detail in Chapter 4. In brief, a certain range of magnitudes of subduction earthquakes is associated with rupture of the entire subduction zone (Petersen *et al.* 2008), in which case the location is not uncertain. In contrast, subduction earthquakes of lower magnitudes are associated with a limited rupture zone with unknown location, modeled by a line source.

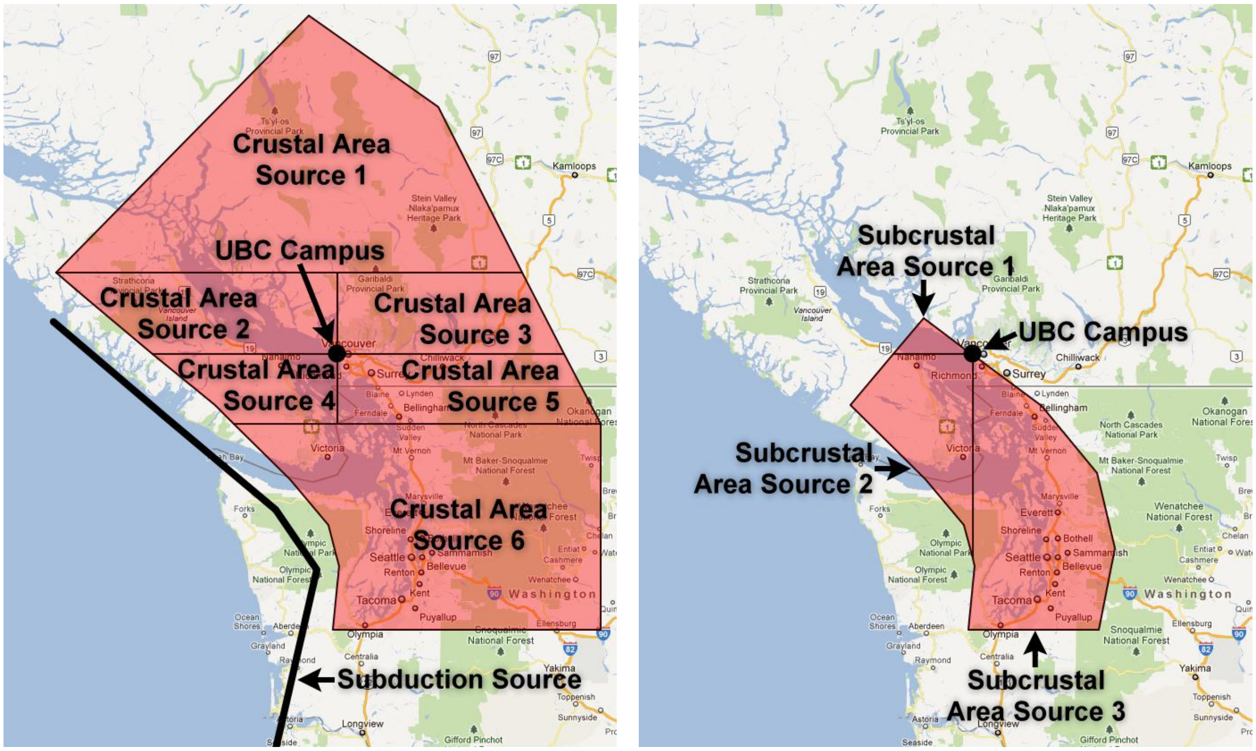


Figure 5-2: Earthquake sources that affect the UBC campus. The map in the background is from Google (© 2012 Google).

Figure 5-3 is a diagram that shows the flow of information in Rt during each evaluation of the limit-state function in Eq. (5-1). It is observed that a host of parameters are utilized to convey information from one model to the next. Each parameter in Figure 5-3 is briefly described in Table 5-3. The figure provides a rough idea of the hazard modeling, which includes the occurrence time, magnitude, and location of earthquakes, as well as ground shaking intensity models that produce site-specific spectral acceleration values. The right-most half of Figure 5-3 outlines the building modeling for two arbitrarily selected buildings. Each building is associated with response, damage, and cost models. Ultimately, the cost of repair and the cost of construction for retrofit enter into the total cost in the limit-state function.

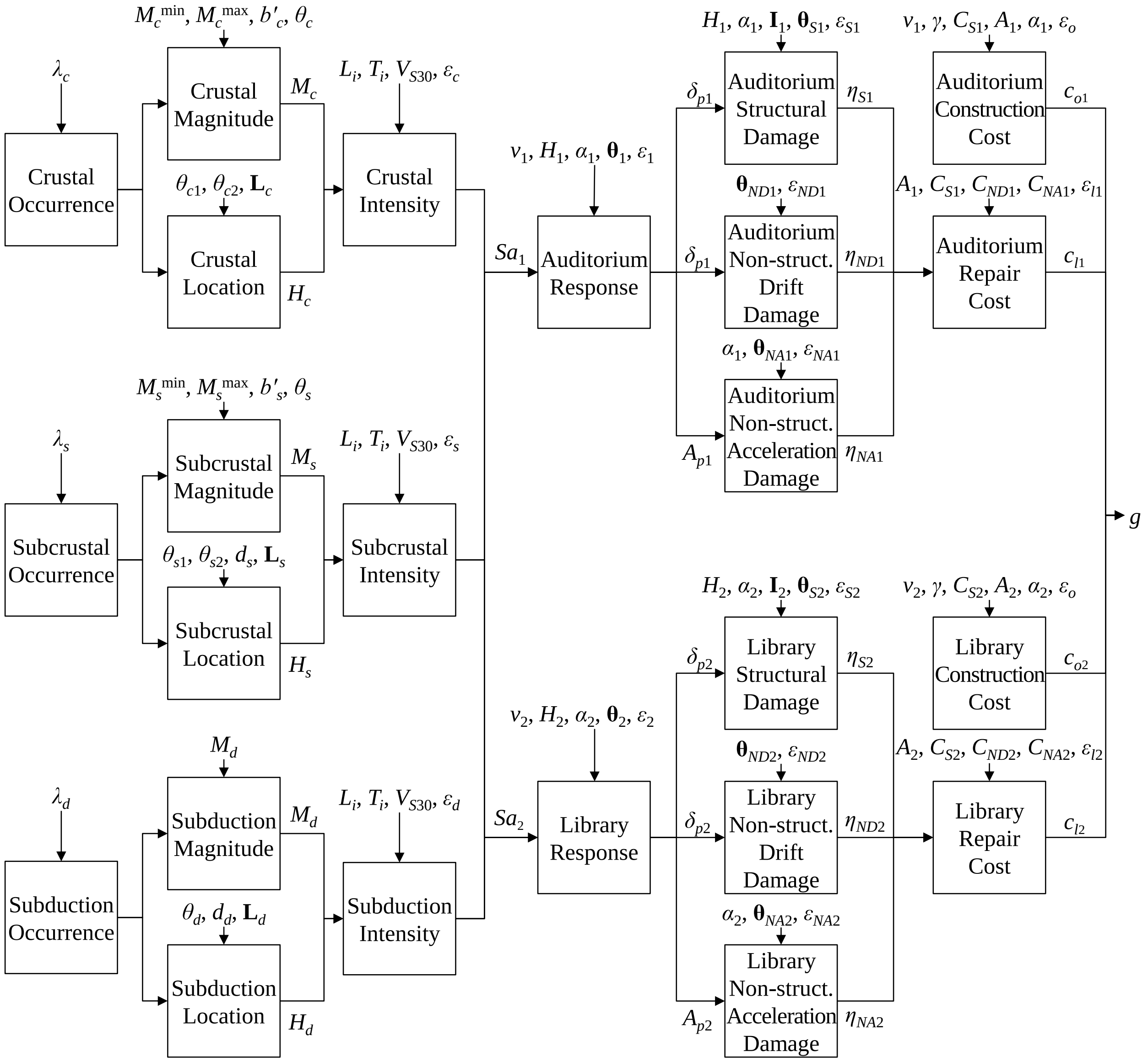

$M_c^{\min}, M_c^{\max}, b'_c, \theta_c$
λ_c
Crustal Occurrence
Crustal Magnitude
M_c
$\theta_{c1}, \theta_{c2}, \mathbf{L}_c$
Crustal Location
H_c
$L_i, T_i, V_{S30}, \varepsilon_c$
Crustal Intensity
$M_s^{\min}, M_s^{\max}, b'_s, \theta_s$
λ_s
Subcrustal Occurrence
Subcrustal Magnitude
M_s
$\theta_{s1}, \theta_{s2}, d_s, \mathbf{L}_s$
Subcrustal Location
H_s
$L_i, T_i, V_{S30}, \varepsilon_s$
Subcrustal Intensity
M_d
λ_d
Subduction Occurrence
Subduction Magnitude
M_d
$\theta_d, d_d, \mathbf{L}_d$
Subduction Location
H_d
$L_i, T_i, V_{S30}, \varepsilon_d$
Subduction Intensity
Sa_1
$v_1, H_1, \alpha_1, \boldsymbol{\theta}_1, \varepsilon_1$
Auditorium Response
Sa_2
$v_2, H_2, \alpha_2, \boldsymbol{\theta}_2, \varepsilon_2$
Library Response
$H_1, \alpha_1, \mathbf{I}_1, \boldsymbol{\theta}_{S1}, \varepsilon_{S1}$
δ_{p1}
Auditorium Structural Damage
η_{S1}
$\boldsymbol{\theta}_{ND1}, \varepsilon_{ND1}$
δ_{p1}
Auditorium Non-struct. Drift Damage
η_{ND1}
$\alpha_1, \boldsymbol{\theta}_{NA1}, \varepsilon_{NA1}$
A_{p1}
Auditorium Non-struct. Acceleration Damage
η_{NA1}
$v_1, \gamma, C_{S1}, A_1, \alpha_1, \varepsilon_o$
Auditorium Construction Cost
c_{o1}
$A_1, C_{S1}, C_{ND1}, C_{NA1}, \varepsilon_{l1}$
Auditorium Repair Cost
c_{l1}
g
$H_2, \alpha_2, \mathbf{I}_2, \boldsymbol{\theta}_{S2}, \varepsilon_{S2}$
δ_{p2}
Library Structural Damage
η_{S2}
$\boldsymbol{\theta}_{ND2}, \varepsilon_{ND2}$
δ_{p2}
Library Non-struct. Drift Damage
η_{ND2}
$\alpha_2, \boldsymbol{\theta}_{NA2}, \varepsilon_{NA2}$
A_{p2}
Library Non-struct. Acceleration Damage
η_{NA2}
$v_2, \gamma, C_{S2}, A_2, \alpha_2, \varepsilon_o$
Library Construction Cost
c_{o2}
$A_2, C_{S2}, C_{ND2}, C_{NA2}, \varepsilon_{l2}$
Library Repair Cost
c_{l2}

Figure 5-3: Overview of the models in this analysis. The symbols are explained in Table 5-3.

Table 5-3: Parameters in the numerical example. $N(\mu, \sigma)$ and $LN(\mu, \sigma)$ denote the normal and lognormal distributions, respectively, with mean μ and standard deviation σ. $U(a, b)$ denotes uniform distribution between a and b.

Symbol	Description	Parameter Type	Characteristics
A_i	Building i floor area	Constant	Varies for each building
A_{pi}	Building i peak acceleration response	Model Response	-
b'_c	Crustal source magnitude distribution parameter	Random Variable	$LN(1.015, 0.378)$
b'_s	Subcrustal source magnitude distribution parameter	Random Variable	$LN(1.128, 0.113)$
c_{li}	Building i repair cost	Model Response	-
C_{NAi}	Building i non-structural acceleration cost per unit area	Constant	Varies for each building
C_{NDi}	Building i non-structural drift cost per unit area	Constant	Varies for each building
c_{oi}	Building i construction cost	Model Response	-
C_{Si}	Building i structural cost per unit area	Constant	Varies for each building
d_d	Subduction source depth uncertainty	Constant	25 km
d_s	Subcrustal source depth uncertainty	Random Variable	$N(50, 5)$ in km
H_c	Crustal earthquake location	Model Response	-
H_d	Subduction earthquake location	Model Response	-
H_i	Building i height	Constant	Varies for each building
H_s	Subcrustal earthquake location	Model Response	-
$\mathbf{I}_i$	Vector of building i irregularity indices	Constant	5 indices for each building
$\mathbf{L}_c$	Vector of crustal area source corner coordinates	Location	18 latitude-longitudes
$\mathbf{L}_d$	Vector of subduction line source coordinates	Location	5 latitude-longitudes
L_i	Building i location	Location	1 latitude-longitude for each building
$\mathbf{L}_s$	Vector of subcrustal area source corner coordinates	Location	14 latitude-longitudes
M_c	Crustal earthquake magnitude	Model Response	-
$M_c^{\max}$	Crustal source maximum magnitude	Constant	7.7
$M_c^{\min}$	Crustal source minimum magnitude	Constant	5.0
M_d	Subduction earthquake magnitude	Random Variable	User-defined distribution
M_s	Subcrustal earthquake magnitude	Model Response	-
$M_s^{\max}$	Subcrustal source maximum magnitude	Random Variable	$LN(7.06, 0.157)$
$M_s^{\min}$	Subcrustal source minimum magnitude	Constant	5.0
Sa_i	Spectral acceleration at building i location	Model Response	-
T_i	Building i natural period of vibration	Constant	Varies for each building
v_i	Building i decision variable	Decision Variable	0
V_{S30}	Shear wave velocity	Random Variable	$LN(900, 90)$ in m/s
α_i	Building i code level	Constant	Varies for each building
γ	Ratio of lateral load system cost to structural cost	Constant	0.2
δ_{pi}	Building i peak drift ratio response	Model Response	-
ε_c	Crustal ground shaking intensity model error	Random Variable	$N(0, 1)$
ε_d	Subduction ground shaking intensity model error	Random Variable	$N(0, 1)$
ε_i	Building i response model error	Random Variable	$N(0, \sigma)$, σ varies with building type
ε_{li}	Building i repair cost model error	Random Variable	$LN(1, 0.1)$
ε_{NAi}	Building i non-str. acceleration damage model error	Random Variable	$N(0, \sigma)$, σ varies with building type
ε_{NDi}	Building i non-str. drift damage model error	Random Variable	$N(0, \sigma)$, σ varies with building type
ε_o	Construction cost model error	Random Variable	$LN(1, 0.1)$
ε_s	Subcrustal ground shaking intensity model error	Random Variable	$N(0, 1)$
ε_{Si}	Building i structural damage model error	Random Variable	$N(0, \sigma)$, σ varies with building type
η_{NAi}	Building i non-structural acceleration damage	Model Response	-
η_{NDi}	Building i non-structural drift damage	Model Response	-
η_{Si}	Building i structural damage	Model Response	-
θ_c	Crustal source magnitude uncertainty	Random Variable	$N(0, 1)$
θ_{c1}	Crustal source latitude uncertainty	Random Variable	$U(0, 1)$
θ_{c2}	Crustal source longitude uncertainty	Random Variable	$U(0, 1)$
θ_d	Subduction source surface location uncertainty	Random Variable	$U(0, 1)$
$\boldsymbol{\theta}_i$	Vector of building i response model parameters	Random Variable	21 normal random variables
$\boldsymbol{\theta}_{NAi}$	Vector of building i non-str. accel. damage model parameters	Random Variable	2 normal random variables
$\boldsymbol{\theta}_{NDi}$	Vector of building i non-str. drift damage model parameters	Random Variable	2 normal random variables
θ_s	Subcrustal source magnitude uncertainty	Random Variable	$N(0, 1)$
θ_{s1}	Subcrustal source latitude uncertainty	Random Variable	$U(0, 1)$
θ_{s2}	Subcrustal source longitude uncertainty	Random Variable	$U(0, 1)$
$\boldsymbol{\theta}_{Si}$	Vector of building i structural damage model parameters	Random Variable	9 normal random variables
λ_c	Crustal source occurrence rate	Constant	0.188/year
λ_d	Subduction source occurrence rate	Constant	0.0037/year
λ_s	Subcrustal source occurrence rate	Constant	0.0932/year

Two models are described in greater detail in the following because they appear prominently in the subsequent sensitivity analysis. One is the building response model; the other is the construction cost model. Both models take the decision variable *v* as input, where *v* is a dimensionless measure of the seismic strength of the building. An increase in *v* is achieved by adding shear walls to a structure, or similar retrofit measures.

The model that predicts the peak drift ratio of the building, δ_p, was introduced in Eq. (3-15) in Chapter 3. In this chapter, the model is augmented to have the decision variable *v* as input:

$$\ln(\delta_p) = \theta_1 + \theta_2 \ln(\delta_y) + \theta_3 \ln(\delta_u) - \theta_4 \ln\left((1+v)V\right) - \theta_5 \ln(\kappa) + \theta_6 \ln(Sa) + \theta_7 Sa + \varepsilon \qquad (5\text{-}3)$$

where δ_y=yield drift ratio, δ_u=ultimate drift ratio, *V*=lateral strength-to-weight ratio, κ=degradation factor, and *Sa*=5%-damped spectral acceleration at the building period computed at the building location. The second-moment statistics for the model parameters is presented in Appendix B and available in Rt. It is observed that *v* serves to modify the building strength, *V*. In turn, this reduces the peak drift ratio, δ_p, which implies less damage and eventually less repair cost. In passing, it is noted that an increase in strength is usually accompanied by an increase in stiffness. However, under the assumption that they increase proportionally, the parameters δ_y and δ_u remain unchanged, because they depend on the ratio of the strength and stiffness.

The model that predicts incremental construction cost associated with retrofit is

$$c_o = v \cdot \left(\gamma \cdot C_S \cdot A\right) \cdot \left(\frac{7-\alpha}{4}\right) \cdot \varepsilon_o \qquad (5\text{-}4)$$

where γ=ratio of the cost of the lateral force resisting system to the total structural cost, C_S=structural cost per unit floor area, *A*=total floor area, α=code-level factor that expresses the strength of the building prior to retrofit, and ε_o=model error factor. In accordance with Chapter 3,

α=1 for buildings that are built prior to seismic standards, α=2 for low-code, α=3 for moderate code, and α=4 for building built to high seismic standards. In Eq. (5-4), the term (7–α)/4 implies that buildings built to high standards cost 25% less to retrofit than the moderate code level, while pre-code buildings cost 50% more to retrofit than the moderate code level.

5.3. ANALYSIS

The premise for the proposed sensitivity measures is reliability analysis with FORM. In addition, it is necessary to utilize a multi-hazard technique to address the presence of multiple earthquake sources identified earlier in Figure 5-2. In fact, most practical problems related to infrastructure risk involve multiple hazards. Several multi-hazard analysis options are available. One is the load coincidence method (Wen 1990), which employs the Poisson pulse process for occurrences. The subsequent developments are valid for that approach, but matters simplify in the following because the short duration of earthquakes yields a negligible probability of coincidence. Thus, the Poisson point process is employed to model earthquake occurrences.

Let N denote the number of hazards, which includes possible hazard combinations for circumstances when the Poisson pulse process is employed. Furthermore, let β_i for i=1,2,…,N denote the reliability index obtained by FORM analysis with the limit-state function in Eq. (5-1) for each hazard or hazard combination. The corresponding cost exceedance probability is

$$p_i = \Phi(-\beta_i) \tag{5-5}$$

where Φ=standard normal CDF. Provided the occurrence of each hazard or hazard combination is governed by the rate λ_i, the rate of cost exceedance associated with each hazard is $\lambda_i \cdot p_i$. The combined rate, including all hazards, is the sum of the individual rates. As a result, the Poisson distribution provides the probability of cost exceedance within the time period T:

$$p = 1 - \exp\left(-T \cdot \sum_{i=1}^{N} \lambda_i \cdot p_i \right) \tag{5-6}$$

This formula remains valid in the load coincidence method, *i.e.*, when coincidence of hazards is included (Wen 1990). The probability p is one point on the cost exceedance probability curve. The use of Eq. (5-6) requires the assumption that all random variables take on new realization at each earthquake occurrence. This results in an overestimation of p for small cost values, as explained in Chapter 4. In contrast, estimations of p become more accurate for high cost thresholds, *i.e.*, rare events, which are in focus here. In addition, Eq. (5-6) assumes that the code level of a repaired building matches its pre-damage code level, which introduces some approximation in the results. This limitation stems from the limitation of the load coincidence method to include time-varying phenomena, which provides a direction for further research.

In the context of FORM analysis, it is common to employ the generalized reliability index, β, as a surrogate measure for p:

$$\beta = -\Phi^{-1}(p) \tag{5-7}$$

where Φ^{-1}=inverse standard normal CDF.

Given the sub-division of seismic sources in Figure 5-3, there are six crustal, three subcrustal, and two subduction hazards, *i.e.*, N=11. In the subsequent analyses, the probability that the total cost exceeds \$100 million in T=50 years is sought, *i.e.*, c_t=\$100 million in Eq. (5-1). FORM analysis for the 11 individual hazards yield the reliability indices shown in Table 5-4. It is observed that subduction earthquakes are associated with the lowest reliability indices, *i.e.*, the highest cost exceedance probabilities. However, it is also observed in Table 5-4 that these earthquakes are associated with low occurrence rates, which mitigate their overall influence on the risk estimates. Substitution of the results in Table 5-4 into Eq. (5-6) yields p=0.076, *i.e.*, a

7.6% chance that the total cost exceeds \$100 million when all earthquake sources are considered. The remainder of this chapter is devoted to the investigation of actions that may affect this estimate of seismic risk.

Table 5-4: Occurrence rates and reliability indices for individual hazards.

Source	Occurrence Rate	Reliability Index
Crustal Area Source 1	0.062	3.5
Crustal Area Source 2	0.019	2.6
Crustal Area Source 3	0.017	2.5
Crustal Area Source 4	0.010	2.4
Crustal Area Source 5	0.017	2.6
Crustal Area Source 6	0.063	3.5
Subcrustal Area Source 1	0.0029	1.6
Subcrustal Area Source 2	0.027	2.0
Subcrustal Area Source 3	0.063	2.7
Subduction Source 1	0.0010	1.3
Subduction Source 2	0.0013	1.2

5.4. RETROFIT DECISIONS

Suppose the manager of a building portfolio seeks to allocate limited resources in an optimal manner to retrofit selected buildings. One approach is to establish a decision tree that compares the cost of retrofit with the expected cost of damage for each building. However, the risk analysis presented earlier yields the entire distribution of cost; hence, an approach that goes beyond expected cost is desirable. To this end, the sensitivity measure $\partial\beta/\partial c_o$ is proposed and discussed in the following, where c_o is cost spent on retrofit. This sensitivity measure reveals the change in reliability index, *i.e.*, total cost exceedance probability, per dollar spent on retrofit. Riederer and Haukaas (2007) proposed a precursory measure, in which classical reliability sensitivity vectors are multiplied by the incremental cost of changing, say, the mean of a random variable. However, while the previous study defined failure by means of force or displacement thresholds, the limit-state function in Eq. (5-1) contains cost. In fact, it contains the total cost, which includes both the construction cost and the repair cost. This fundamentally changes the evaluation of $\partial\beta/\partial c_o$, as demonstrated in the following. The present study also pioneers the use of $\partial\beta/\partial c_o$ under circumstances when a portfolio of buildings subjected to multiple hazards is considered.

5.4.1. Derivation

The fundamental idea behind $\partial\beta/\partial c_o$ is that buildings that yield the largest increase in the reliability index, *i.e.*, the largest reduction in cost exceedance probability, per dollar spent on retrofit should be prioritized. To evaluate $\partial\beta/\partial c_o$, it is necessary to recognize that β depends on c_o through p, p_i, β_i, and v, which means the chain rule of differentiation yields

$$\frac{\partial\beta}{\partial c_o}=\frac{\partial\beta}{\partial p}\cdot\sum_{i=1}^{N}\left(\frac{\partial p}{\partial p_i}\cdot\frac{\partial p_i}{\partial\beta_i}\cdot\frac{\partial\beta_i}{\partial v}\cdot\frac{\partial v}{\partial c_o}\right) \tag{5-8}$$

For convenience of subsequent derivations, the three first derivatives in the right-hand side are merged and evaluated by differentiation of Eqs. (5-5), (5-6), and (5-7), which yields

$$\frac{\partial\beta}{\partial\beta_i}=\frac{\partial\beta}{\partial p}\cdot\frac{\partial p}{\partial p_i}\cdot\frac{\partial p_i}{\partial\beta_i}=\frac{1}{\varphi(\beta)}\cdot T\cdot\lambda_i\cdot\exp\left(-T\cdot\sum_{i=1}^{N}\lambda_i\cdot p_i\right)\cdot\varphi(\beta_i) \tag{5-9}$$

where φ=standard normal PDF. According to Eq. (5-9), the value of $\partial\beta/\partial\beta_i$ increases with the occurrence rate, λ_i, and decreases with the reliability index, β_i. In other words, frequent and/or damaging hazards contribute more to the sensitivity in Eq. (5-8). The derivative $\partial\beta_i/\partial v$ in the right-hand side of Eq. (5-8) is obtained by differentiating the reliability index from FORM (Der Kiureghian 2005):

$$\frac{\partial\beta_i}{\partial v}=\frac{1}{\|\nabla G\|}\cdot\left.\frac{\partial g}{\partial v}\right|_{\mathbf{x}^*,\boldsymbol{\theta}^*} \tag{5-10}$$

where *G*=limit-state function in the space of standard normal variables, and the asterisk denotes the most likely realization of random variables on the limit-state surface. Eq. (5-10) requires the derivative of *g* in Eq. (5-1), which in turn requires the derivative of the models that enter into the evaluation of total cost. These derivatives are available in Rt because each model computes response sensitivities by the direct differentiation method (Kleiber *et al.* 1997). The last derivative in the right-hand side of Eq. (5-8) represents the marginal cost of retrofit; differentiation of Eq. (5-4) yields

$$\frac{\partial c_o}{\partial v} = \left(\gamma \cdot C_S \cdot A\right) \cdot \left(\frac{7-\alpha}{4}\right) \cdot \varepsilon_o \tag{5-11}$$

5.4.2. Application

The sensitivity measure $\partial\beta/\partial c_o$ is evaluated for all 622 buildings on the UBC campus at their "present state," *i.e.*, v=0. Table 5-5 displays the value of $\partial\beta/\partial c_o$ for the 10 highest ranked buildings. It is observed that most of the highest ranked buildings are unreinforced masonry buildings. This is not surprising, because unreinforced masonry buildings tend to sustain significant damage in earthquakes. It is also observed in Table 5-5 that most of the highest ranked buildings belong to pre-code and low-code levels. In other words, they are either built before seismic codes appeared around 1940 or before the seismic codes were upgraded around 1975. For example, the Old Auditorium, which ranks at the top in Table 5-5, is a large unreinforced masonry structure, built in 1925. The value of $\partial\beta/\partial c_o$ for this building implies that spending $154,000 on retrofit for this building changes the reliability index for the entire campus by 0.01. This is more than any of the other buildings on the UBC campus.

Figure 5-4 disaggregates $\partial\beta/\partial c_o$ to expose the contributions from crustal, subcrustal, and subduction earthquakes. The figure indicates that the highest contribution to the cost sensitivities is from subcrustal earthquakes. This is reasonable because subcrustal earthquakes are associated with relatively high frequency and high probabilities of cost exceedance, as shown earlier in Table 5-4.

Table 5-5: Top 10 buildings according to retrofit priority.

Building Name	Prototype	Code Level	$\partial\beta/\partial c_o$
Old Auditorium	Unreinforced Masonry	Pre	$6.5\cdot10^{-8}$
Rugby Pavillion	Unreinforced Masonry	Low	$3.7\cdot10^{-8}$
John Owen Pavilion	Unreinforced Masonry	Low	$3.7\cdot10^{-8}$
Thunderbird Sports Centre	Unreinforced Masonry	Low	$3.7\cdot10^{-8}$
Animal Science Aquaculture Centre	Unreinforced Masonry	Low	$2.7\cdot10^{-8}$
Environmental Services Facility	Unreinforced Masonry	Low	$1.9\cdot10^{-8}$
Sherwood Building	Unreinforced Masonry	Low	$1.8\cdot10^{-8}$
Green College Kitchen	Wood Large Frame	Moderate	$1.8\cdot10^{-8}$
Barn Coffee Shop	Wood Large Frame	Pre	$1.6\cdot10^{-8}$
Cheeze Factory Undergrad Society	Wood Large Frame	Pre	$1.6\cdot10^{-8}$

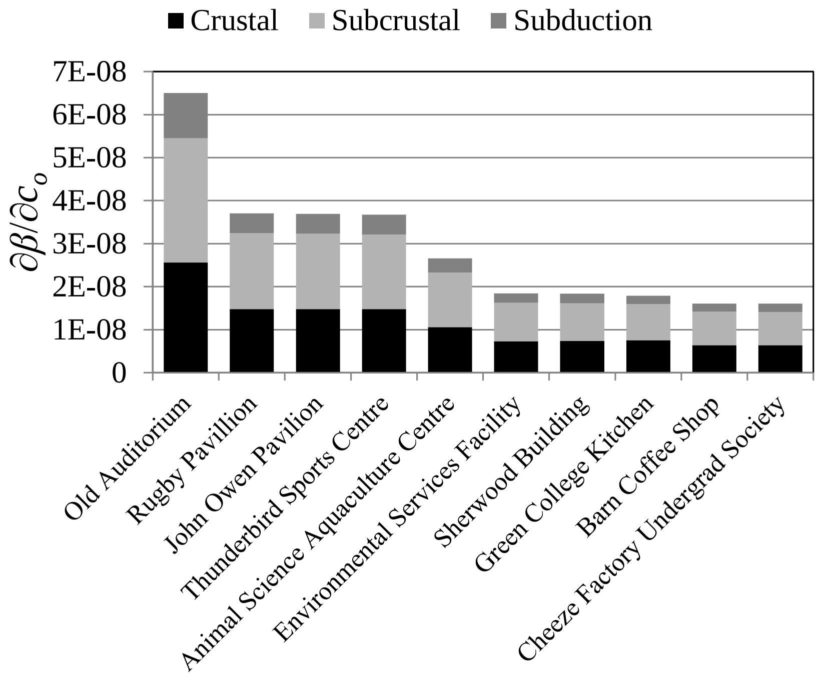


Figure 5-4: Top 10 buildings according to retrofit priority.

The positive value of $\partial\beta/\partial c_o$ in Table 5-5 suggests that it is worthwhile to allocate resources to retrofit these buildings. In fact, a positive value for $\partial\beta/\partial c_o$ is observed for 393 of the buildings on campus, which is about 63% of the building stock. Conversely, $\partial\beta/\partial c_o$ takes on negative values for the other 229 buildings. The negative sign indicates that it is not worthwhile to retrofit these 229 buildings because the construction cost surpasses the gain from decreased damage. Table 5-6 shows the negative values of $\partial\beta/\partial c_o$ for the 10 lowest ranked buildings. These buildings yield the largest increase in cost exceedance probability per dollar spent on retrofit. It is observed that Table 5-6 contains mostly concrete shear wall buildings. This is not surprising, because the lateral force resisting system of these buildings is specifically designed to carry high seismic forces. In fact, the buildings in Table 5-6 belong to the moderate-code level, which means that they are built after 1975 or retrofitted to contemporary codes. It is therefore reasonable that these buildings are not prioritized for seismic retrofit according to the measure $\partial\beta/\partial c_o$.

Table 5-6: Bottom 10 buildings according to retrofit priority.

Building Name	Prototype	Code Level	$\partial\beta/\partial c_o$
Animal Care Rodent Breeding	Concrete Shear Wall	Moderate	$-2.1\cdot10^{-9}$
Animal Science Main Sheep Unit	Concrete Moment Frame	Moderate	$-1.8\cdot10^{-9}$
Acadia Park Highrise	Concrete Shear Wall	Moderate	$-1.7\cdot10^{-9}$
The Regency	Concrete Shear Wall	Moderate	$-1.6\cdot10^{-9}$
ChemBio Engineering	Concrete Shear Wall	Moderate	$-1.6\cdot10^{-9}$
The Chatham	Concrete Shear Wall	Moderate	$-1.6\cdot10^{-9}$
Irving Barber Library	Concrete Shear Wall	Moderate	$-1.6\cdot10^{-9}$
Technology Enterprise Facility	Concrete Shear Wall	Moderate	$-1.6\cdot10^{-9}$
ICICS Main	Concrete Shear Wall	Moderate	$-1.6\cdot10^{-9}$
The Balmoral	Concrete Shear Wall	Moderate	$-1.6\cdot10^{-9}$

It was observed that unreinforced masonry buildings are the top candidates for retrofitting, while those that are not worthwhile to retrofit are mostly concrete shear wall buildings. This reaffirms the findings in Figure 4-17 of the previous chapter. This figure shows that concrete shear wall buildings at UBC exhibit the best seismic performance, while unreinforced masonry buildings exhibit the poorest.

5.5. Model Refinement Decisions

Suppose an analyst is carrying out a regional risk analysis and contemplates the refinement of selected models to improve the results. In particular, the analyst considers replacing selected simple building response models of the type in Eq. (5-3) with detailed structural analysis models. Clearly, only the most important buildings can be addressed due to the time it takes to establish a detailed model and the added computational cost. This section provides guidance for the analyst to prioritize between buildings.

5.5.1. Derivation

First, it is recognized that the objective is to reduce epistemic uncertainty. Provided that the epistemic uncertainty is properly included in the models, a more detailed model will produce results with less uncertainty. Second, it is understood that it is the effect on the overall probability of exceedance that must guide the decision to replace a model. In other words, the model that has the largest influence on β as defined in Eq. (5-7) should be replaced with a better model. For example, for a model with only one epistemic random variable, it is the sensitivity $\partial\beta/\partial\sigma$, where σ is the standard deviation of that epistemic random variable, that should guide the prioritization. In general, each model has several epistemic random variables. To this end, a sensitivity measure that represents the derivative of β with respect to the standard deviation of all the epistemic random variables in the model is sought.

Suppose a model is generically written as $y=y(\boldsymbol{\theta},\mathbf{x})$, where $\boldsymbol{\theta}$ and $\mathbf{x}$ remain the vectors of epistemic and aleatory random variables, respectively. Furthermore, let K denote the number of epistemic random variables in the model. The value of K for a variety of models employed in this study is shown in the last column of Table 5-1. Next, consider the variance of the response from this model with respect to the epistemic random variables. If the model is linear, then an exact expression exists, otherwise the mean-centered approximation based on the first-order Taylor approximation of the model is employed. Both read

$$\tilde{\sigma}^2 = \nabla_{\boldsymbol{\theta}} y^T \cdot \boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}} \cdot \nabla_{\boldsymbol{\theta}} y = \sum_{i=1}^{K} \sum_{j=1}^{K} \left(\frac{\partial y}{\partial \theta_i} \cdot \frac{\partial y}{\partial \theta_j} \cdot \rho_{ij} \cdot \sigma_i \cdot \sigma_j \right) \tag{5-12}$$

where $\nabla_{\boldsymbol{\theta}} y$=derivatives of y with respect to $\boldsymbol{\theta}$, $\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}}$=covariance matrix of $\boldsymbol{\theta}$, ρ_{ij}=correlation coefficient between the components of $\boldsymbol{\theta}$, and σ_i=standard deviation of the components of $\boldsymbol{\theta}$. In order to study the influence of epistemic uncertainty on β, the derivative $\partial\beta/\partial\tilde{\sigma}$ is sought. The chain rule of differentiation yields

$$\frac{\partial \beta}{\partial \tilde{\sigma}} = \sum_{i=1}^{N} \left(\frac{\partial \beta}{\partial \beta_i} \cdot \sum_{j=1}^{K} \left(\frac{\partial \beta_i}{\partial \sigma_j} \cdot \frac{\partial \sigma_j}{\partial \tilde{\sigma}} \right) \right) \tag{5-13}$$

In the following, this equation is evaluated separately for each instance of a model. The first derivative in the right-hand side of Eq. (5-13) is provided earlier in Eq. (5-9). The second derivative quantifies the change in the individual reliability indices due to changes in the standard deviation of some of the intervening random variables, which is a well-known reliability sensitivity measure (Der Kiureghian 2005). The inverse of the last derivative in the right-hand side of Eq. (5-13) is obtained by differentiation of Eq. (5-12), which yields

$$\frac{\partial \tilde{\sigma}}{\partial \sigma_j} = \frac{1}{\tilde{\sigma}} \cdot \frac{\partial y}{\partial \theta_j} \cdot \sum_{i=1}^{K} \left(\frac{\partial y}{\partial \theta_i} \cdot \rho_{ij} \cdot \sigma_i \right) \Bigg|_{\boldsymbol{\theta}^*} \tag{5-14}$$

5.5.2. Application

In the following, $\partial\beta/\partial\tilde{\sigma}$ is evaluated for several model types. First, consider the model that predicts the drift response for each of the 622 buildings on the UBC campus. Each of these models has epistemic random variables, and $\tilde{\sigma}$ is the amalgamated standard deviation that includes all of them, in accordance with Eq. (5-12). Table 5-7 shows the values of $\partial\beta/\partial\tilde{\sigma}$ for the 10 highest ranked buildings. These are buildings for which a reduction in epistemic uncertainty would have the greatest impact on the risk. As expected, a reduction in epistemic uncertainty, *i.e.*, a reduction in $\tilde{\sigma}$, decreases the probability of exceeding $100 million. This implies an increase in the reliability index, which is correctly captured by the negative values of $\partial\beta/\partial\tilde{\sigma}$ in Table 5-7. It is interesting to note that the highest ranked buildings in Table 5-7 are concrete shear wall buildings. This suggests that detailed structural modeling of this type of buildings has the greatest beneficial impact on the risk estimates. This point is further examined in the next section.

Table 5-7: Top 10 building models according to epistemic uncertainty.

Building Name	Total Value ($)	Building Type	$\partial\beta/\partial\tilde{\sigma}$
The Chatam	85,809,083	Concrete Shear Wall	−16,558
Marine Drive Tower 1	35,309,484	Concrete Shear Wall	−14,077
The Regency	29,438,934	Concrete Shear Wall	−13,353
The Balmoral	25,864,220	Concrete Shear Wall	−11,742
Promontory	24,481,800	Concrete Shear Wall	−11,073
Acadia Park Highrise	26,717,850	Concrete Shear Wall	−10,268
Corus	17,573,677	Concrete Shear Wall	−7,576
Ross House	19,660,023	Concrete Shear Wall	−7,540
Mackenzie House	22,764,992	Concrete Shear Wall	−7,516
Walter Gage East Tower	20,034,434	Concrete Shear Wall	−6,463

The question might be asked why the sensitivity with respect to the standard deviation of loss is not used for this ranking. The standard deviation of loss may vary by orders of magnitude for different buildings. It can be millions of dollars for a large expensive building while just a few thousand dollars for a small one. Therefore, the sensitivity of β per dollar change in loss standard deviation would undermine the influence of the building value on the ranking. In contrast, the standard deviation of the drift ratio is comparable for different buildings. Thus, expensive buildings would rank high according to the sensitivity with respect to drift ratio. In fact, the buildings in Table 5-7 are amongst the most expensive buildings on the UBC campus.

Next, $\partial\beta/\partial\tilde{\sigma}$ is evaluated for crustal and subcrustal magnitude models and presented in Table 5-8. These models are based on a bounded exponential probability distribution (McGuire 2004), and the distribution parameters are here modeled as epistemic random variables. The subduction magnitude model is excluded here because it is a single random variable with probability distribution from Petersen *et al.* (2008), which was presented in Chapter 4. Table 5-8 reveals that the magnitude model for subcrustal Area Source 2 in Figure 5-2 is the model for which a reduction in the epistemic uncertainty would have the greatest impact on the cost probability. This observation is reasonable in light of the frequent and damaging earthquakes that this source generates. In fact, Table 5-4 shows that for this source, the reliability index is low, while the occurrence rate is high. This yields high $\partial\beta/\partial\beta_i$-values for this source, as shown in Table 5-8, which is an important factor in Eq. (5-13).

Table 5-8: Ranking of magnitude models according to epistemic uncertainty.

Model	$\partial\beta/\partial\tilde{\sigma}$	$\partial\beta/\partial\beta_i$
Subcrustal Area Source 2 Magnitude	−0.097	0.44
Subcrustal Area Source 3 Magnitude	−0.062	0.20
Subcrustal Area Source 1 Magnitude	−0.023	0.098
Crustal Area Source 2 Magnitude	−0.019	0.092
Crustal Area Source 3 Magnitude	−0.019	0.087
Crustal Area Source 5 Magnitude	−0.018	0.085
Crustal Area Source 4 Magnitude	−0.017	0.077
Crustal Area Source 6 Magnitude	−0.0039	0.018
Crustal Area Source 1 Magnitude	−0.0034	0.016

$\partial\beta/\partial\tilde{\sigma}$ is also evaluated for the ground shaking intensity models, which produce the spectral acceleration at each building site. Figure 5-5 shows the value of $\partial\beta/\partial\tilde{\sigma}$ for three arbitrarily selected building sites. The figure suggests that efforts to reduce the epistemic uncertainty in the crustal intensity model will have the greatest impact on the risk estimate. In fact, this holds true for all 622 buildings, although values are only shown for three buildings in Figure 5-5.

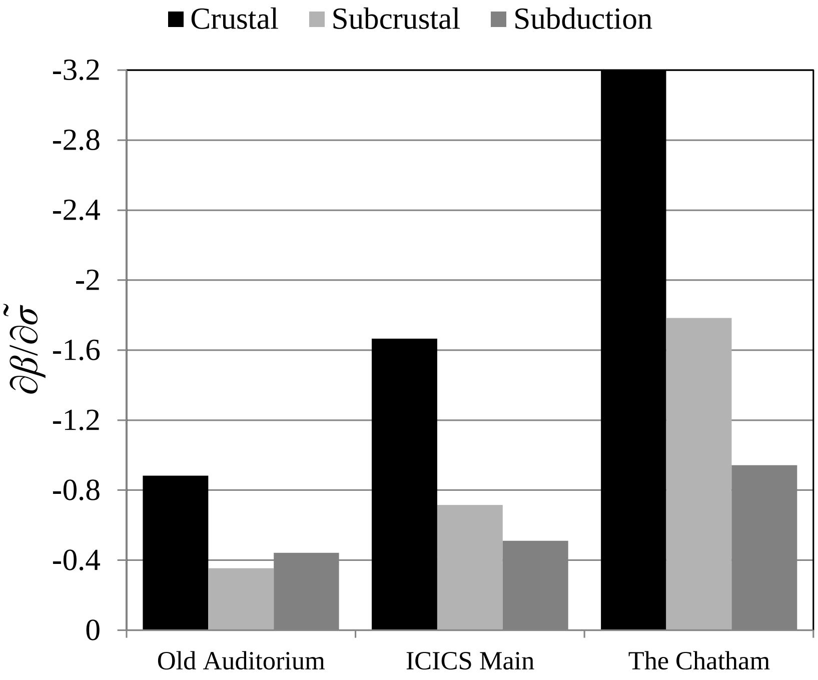


Figure 5-5: Ranking of ground shaking intensity models according to epistemic uncertainty.

Finally, the relative influence of hazard modeling and modeling of building response, damage, and repair cost is compared. In this case, $\partial\beta/\partial\tilde{\sigma}$ is evaluated with all epistemic random variables included in the evaluation of $\tilde{\sigma}$. The expression for $\partial\beta/\partial\tilde{\sigma}$ then disaggregates into the two parts

$$\frac{\partial\beta}{\partial\tilde{\sigma}}=\sum_{i=1}^{15}\frac{\partial\beta}{\partial\sigma_i}\cdot\frac{\partial\sigma_i}{\partial\tilde{\sigma}}+\sum_{i=16}^{271}\frac{\partial\beta}{\partial\sigma_i}\cdot\frac{\partial\sigma_i}{\partial\tilde{\sigma}} \tag{5-15}$$

where the first sum is taken over 15 epistemic random variables associated with the hazard models, while the second sum is taken over 256 epistemic random variables associated with the building models. The analysis reveals that the first sum equals $-8.54\cdot10^{-9}$ and the second equals $-7.20\cdot10^{-7}$. This shows that, for the epistemic uncertainty that is modeled in this study, it is far more effective to reduce the epistemic uncertainty in the building models rather than the hazard models. However, it is stressed that the aleatory uncertainty in the occurrence of hazards far exceeds the aleatory uncertainty in the building-related models. In short, it is more cost effective to reduce the uncertainty in the building modeling, while overall the greatest uncertainty is associated with earthquake occurrences, much of which is irreducible.

5.6. Model Improvement Decisions

Instead of replacing a model with a more detailed one, as contemplated above, risk estimates may be improved by long-term efforts to improve existing models. Researchers carry out such model improvement efforts by gathering new data, *etc*. This section develops a new sensitivity measure to prioritize between models, to allocate limited resources in an optimal manner.

5.6.1. Derivation

In order to examine the cost-effectiveness of improving models, it is necessary to quantify the cost of such efforts. This cost is here denoted by $\tilde{c}$. As a result, the sensitivity measure $\partial\beta/\partial\tilde{c}$ quantifies the effect of allocating resources for model improvement on the risk. To derive this measure, it is first recognized that model improvement is measured by changes in σ_j, *i.e.*, changes in the standard deviation of the model parameters in $\boldsymbol{\theta}$. Moreover, the primary means of model improvement is data-gathering, which is measured by the number, *n*, of new observations. To this end, the chain rule of differentiation yields

$$\frac{\partial\beta}{\partial\tilde{c}} = \sum_{i=1}^{N}\left(\frac{\partial\beta}{\partial\beta_i}\cdot\sum_{j=1}^{K}\left(\frac{\partial\beta_i}{\partial\sigma_j}\cdot\frac{\partial\sigma_j}{\partial n}\cdot\frac{\partial n}{\partial\tilde{c}}\right)\right) \tag{5-16}$$

where the first derivative in the right-hand side is provided in Eq. (5-9) and the second derivative is a well-known reliability sensitivity measure, as stated in the previous section. The third derivative, $\partial\sigma_j/\partial n$, represents the change in the standard deviation of a model parameter due to a change in the number of observations. This derivative is expressed below for three cases: Linear regression models, nonlinear regression models, and individual random variables. The last derivative in the right-hand side of Eq. (5-16) is the inverse of the cost of obtaining one data point, which is discussed shortly.

To derive $\partial\sigma_j/\partial n$ for linear regression models, the starting point is the variance of the model parameters, provided by Box and Tiao (1992) as

$$\sigma_j^2 = \frac{1}{n-k}\left(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\theta}}\right)^T\left(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\theta}}\right)\left(\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\right)_{jj} \tag{5-17}$$

where k=number of model parameters, $\mathbf{y}$=vector of observed results, $\mathbf{X}$=matrix of observations, $\hat{\boldsymbol{\theta}}$=mean vector of model parameter, and $(\ )_{jj}$ identifies the jth diagonal component. The derivative of Eq. (5-17) with respect to n is

$$\frac{\partial\sigma_j}{\partial n} = \frac{-\sigma_j}{2\left(n-k\right)} \tag{5-18}$$

Box and Tiao (1992) also provides the variance of the model error as

$$\sigma_\varepsilon^2 = \frac{1}{n-k}\left(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\theta}}\right)^T\left(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\theta}}\right) \tag{5-19}$$

The derivative of Eq. (5-19) with respect to n is

$$\frac{\partial\sigma_\varepsilon}{\partial n} = \frac{-\sigma_\varepsilon}{2\left(n-k\right)} \tag{5-20}$$

where Eqs. (5-18) and (5-20) are implemented in Rt and applied in the following.

For nonlinear regression models, Seber and Wild (2003) provides the variance of the model parameters as

$$\sigma_j^2 = \frac{1}{n-k}\left(\mathbf{y}-\hat{\mathbf{y}}(\hat{\boldsymbol{\theta}},\mathbf{X})\right)^T\left(\mathbf{y}-\hat{\mathbf{y}}(\hat{\boldsymbol{\theta}},\mathbf{X})\right)\left(\left(\mathbf{J}^T\mathbf{J}\right)^{-1}\right)_{jj} \tag{5-21}$$

where $\hat{\mathbf{y}}(\hat{\boldsymbol{\theta}},\mathbf{X})$=vector of model predictions, and $\mathbf{J}=\partial\hat{\mathbf{y}}/\partial\boldsymbol{\theta}$=Jacobian matrix of $\hat{\mathbf{y}}$ with respect to $\boldsymbol{\theta}$. They also provide the variance of the model error as

$$\sigma_\varepsilon^2 = \frac{1}{n-k}\left(\mathbf{y}-\hat{\mathbf{y}}(\hat{\boldsymbol{\theta}},\mathbf{X})\right)^T\left(\mathbf{y}-\hat{\mathbf{y}}(\hat{\boldsymbol{\theta}},\mathbf{X})\right) \tag{5-22}$$

The derivatives of these nonlinear regression equations are equal to Eqs. (5-18) and (5-20), except **X** is replaced by **J**. In fact, the **J** linearizes the nonlinear model $\hat{\mathbf{y}}$ at $\hat{\theta}$ in order to compute the variance.

It is also educational to examine the reduction in epistemic uncertainty for ordinary random variables. This is instructive because of the simplicity of the equations, and useful because it effectively addresses generic models, for which random variables serve as input. From elementary statistics, the variance of a sample of *n* observations is

$$\sigma_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i-\mu_j\right)^2 \tag{5-23}$$

where x_i=i^{th} observation and μ_j=sample mean. The derivative is

$$\frac{\partial\sigma_j}{\partial n} = \frac{-\sigma_j}{2\left(n-1\right)} \tag{5-24}$$

It is emphasized that this equation and the other expressions for $\partial\sigma_j/\partial n$ are derived under the assumption that the mean model is unaffected by added observations. In other words, it is required that the variance diminishes when new data points are added. In practice, this is often an approximation, because it requires the fundamental model form to be correct. However, it is argued here that the approximation is useful, for two reasons: Over time, new data will generally tend to reduce the epistemic uncertainty, and the approximation facilitates a quantitative measure to guide the data-gathering efforts.

In passing, it is observed that the minus sign in the expressions for $\partial\sigma_j/\partial n$ correctly implies that that the standard deviation of the model parameter decreases when n increases. Moreover, it is observed that the reduction in the standard deviation is smaller when n is large than when n is small. *I.e.*, a model that is based on a large number of observations will benefit less from a few more observations. The reduction in the standard deviation is also greater when k is large than when k is small. That is, a complex model with many model parameters will benefit more from new observations.

The last derivative in the right-hand side of Eq. (5-16) requires the estimation of the cost of obtaining one data point. This is feasible in specific circumstances, while it is more challenging to give general values. Some observations are readily obtained, while others come at a significant cost. Examples of observations that are relevant to the example in this study and counted by n include: 1) Testing of structural components or entire building models on a shaking table, which is utilized to calibrate the building response, damage, and repair cost models; 2) Analysis of a highly refined numerical models, which is utilized to calibrate building response models; 3) Reconnaissance of damaged buildings after an earthquake, which is utilized to calibrate damage models; and 4) Claims reports from insurance companies, which are utilized to calibrate damage and repair cost models. Obviously, the cost of obtaining such data varies and rough estimates are provided here for the numerical example. In particular, it is assumed that each observation takes a day or two of paid work and costs around \$500, *i.e.*, $\partial\tilde{c}/\partial n=500$.

5.6.2. Application

The sensitivity measure $\partial\beta/\partial\tilde{c}$ is evaluated for the model types that are listed in Table 5-1 and employed in the regional risk analysis for the 622 buildings on the UBC campus. As shown in Table 5-1, the number of instances varies for each model. For instance, there are 134 structural response models for concrete shear wall buildings, while the equivalent number of models for concrete moment frames is only 22. Therefore, to make the sensitivities comparable, the value of $\partial\beta/\partial\tilde{c}$ for each model type is normalized by the number of instances of that model.

Table 5-9 ranks the models according to the normalized $\partial\beta/\partial\tilde{c}$-values. The crustal ground shaking intensity model and the concrete shear wall structural response model rank highest. This suggests that allocating resources to improve these models is the most cost-effective way of changing the risk estimate by model improvement efforts. Table 5-9 also reveals that the structural response models tend to rank highest, while structural damage models rank lowest. Thus, it appears worthwhile to gather data to reduce the epistemic uncertainty in the models for the displacement and acceleration response of the buildings, rather than the damage models.

The positive sign of the $\partial\beta/\partial\tilde{c}$-values in Table 5-9 indicates that the reliability index increases when resources are allocated to gather new data. This is reasonable, because the model improvement reduces the uncertainty, which in turn reduces the probability of exceeding the \$100 million total cost threshold.

An additional remark about the results in Table 5-9 is made in regards to the results in the previous section. There it was noted that several of the concrete shear wall buildings should be prioritized for more refined structural modeling to reduce epistemic uncertainty. It was also noted that the crustal intensity model is the primary candidate for more refined hazard analysis. It is reassuring that these two models also rank high in this section, where the improvement of models by data-gathering is contemplated.

Table 5-9: Ranking of 31 model types according to cost of model improvement.

Model Type	$\partial\beta/\partial\tilde{c}$
Crustal Ground Shaking Intensity	$3.7\cdot10^{-7}$
Concrete Shear Wall Structural Response	$3.5\cdot10^{-7}$
Subcrustal Ground Shaking Intensity	$2.7\cdot10^{-7}$
Subduction Ground Shaking Intensity	$2.4\cdot10^{-7}$
Concrete Frame with Masonry Infill Wall Structural Response	$5.2\cdot10^{-8}$
Precast Concrete Structural Response	$4.0\cdot10^{-8}$
Wood Large Frame Structural Response	$3.5\cdot10^{-8}$
Steel Frame with Masonry Infill Wall Structural Response	$2.7\cdot10^{-8}$
Concrete Moment Frame Structural Response	$2.3\cdot10^{-8}$
Steel Moment Frame Structural Response	$1.4\cdot10^{-8}$
Steel Braced Frame Structural Response	$1.2\cdot10^{-8}$
Steel Frame with Concrete Shear Wall Structural Response	$1.1\cdot10^{-8}$
Unreinforced Masonry Structural Response	$7.2\cdot10^{-9}$
Reinforced Masonry Structural Response	$5.7\cdot10^{-9}$
Wood Light Frame Structural Response	$4.9\cdot10^{-9}$
Steel Light Frame Structural Response	$2.5\cdot10^{-9}$
Non-structural Drift Damage	$2.2\cdot10^{-12}$
Concrete Shear Wall Structural Damage	$1.6\cdot10^{-12}$
Unreinforced Masonry Structural Damage	$1.2\cdot10^{-12}$
Concrete Moment Frame Structural Damage	$1.2\cdot10^{-12}$
Concrete Frame with Masonry Infill Wall Structural Damage	$1.0\cdot10^{-12}$
Steel Frame with Concrete Shear Wall Structural Damage	$2.1\cdot10^{-13}$
Precast Concrete Structural Damage	$1.7\cdot10^{-13}$
Wood Large Frame Structural Damage	$1.2\cdot10^{-13}$
Non-structural Acceleration Damage	$2.9\cdot10^{-14}$
Reinforced Masonry Structural Damage	$2.1\cdot10^{-14}$
Steel Braced Frame Structural Damage	$1.1\cdot10^{-14}$
Steel Frame with Masonry Infill Wall Structural Damage	$7.8\cdot10^{-15}$
Steel Moment Frame Structural Damage	$6.7\cdot10^{-15}$
Steel Light Frame Structural Damage	$3.9\cdot10^{-15}$
Wood Light Frame Structural Damage	$1.0\cdot10^{-15}$

5.7. CONCLUSIONS

The developments in this chapter are based on the use of reliability methods to assess risk. This approach has proved accurate and effective for computation of cost exceedance probabilities. On this basis, the overarching objective in this chapter is twofold. First, it is sought to identify and characterize epistemic uncertainty in a comprehensive manner. This is important because epistemic uncertainty, such as model uncertainty, is reducible and has significant influence on risk estimates. This goal is achieved by utilizing a library of probabilistic models that contain random variables that represent both aleatory and epistemic uncertainty. Second, it is sought to allocate resources in an optimal manner to address risk, either by retrofitting efforts or efforts to reduce the epistemic uncertainty. This is achieved by the development of three new sensitivity measures: One prioritizes infrastructure components for retrofit; one prioritizes physical phenomena for detailed modeling; and one prioritizes models for model improvement efforts. The three sensitivity measures are evaluated and discussed for a comprehensive risk analysis of 622 buildings on the UBC campus in Vancouver, Canada. The results show that the pre-code unreinforced masonry buildings on campus are most cost-effective to retrofit. The results also show that the epistemic uncertainty in the ground shaking intensity model for shallow crustal earthquakes and the response model for concrete shear wall buildings are the most cost-effective to improve for the analysis of this region.

Chapter 6. Conclusions and Future Work

A summary of the contributions in this thesis is provided in the Introduction. This chapter provides observations on the research approach from a broader perspective. Several directions and tasks for future research are also suggested.

6.1. Summary of the Research Approach

This thesis revisits risk analysis in structural and earthquake engineering. The methodology for risk analysis in the present study contrasts the approaches in the literature in two aspects: treatment of uncertainties, and computation of risk. In the existing approaches, the uncertainty is described by conditional probability models, *e.g.*, fragility curves that output a probability conditioned upon the value of input(s). Thereafter, risk is typically computed in either of two ways: 1) Computing the expectation of the loss; 2) Integrating the conditional probabilities by the theorem of total probability to compute the probability that the loss exceeds a threshold. In contrast, the uncertainty in the present approach is discretized in terms of random variables that are input to models. Given the realization of these random variables, the model produces a physical measureable output, such as spectral acceleration and seismic loss, not a probability. In turn, risk, *i.e.*, the probability of exceeding a loss threshold, is computed by reliability methods. This reliability-based approach offers the following advantages:

1) Uncertainties are explicitly characterized by random variables. In contrast, uncertainties are implicit in conditional probability models.
2) It is possible to distinguish between aleatory and epistemic uncertainties, which permits targeted efforts to reduce the epistemic uncertainty, *e.g.*, by the methods in Chapter 5.

3) Many of the existing engineering models that receive a number of inputs and produce physical output(s) can be employed in the analysis, provided that the uncertainty in the model is properly characterized by random variables. For instance, finite element models can be directly incorporated in the analysis, as shown by the application in Chapter 2.
4) It is possible to compute the risk when multiple hazards are present and account for possible coincidence of those hazards. An example is provided in the application in Chapter 2.
5) It is possible to include time-varying phenomena, such as long-term deterioration and discounting in the analysis. Examples are provided in Chapter 2 and Chapter 3.
6) The models can take limitless number of inputs and provide limitless number of outputs to many downstream models. The limit is determined by the memory of the computer. In contrast, the conditional probability models are typically limited to one or two inputs and provide a single output.
7) Portfolio level analyses are readily carried out by modifying the limit-state function to include the sum of the losses of all infrastructures, *e.g.*, all buildings in a region. Chapter 4 presents an example of such analyses.
8) The probabilities are computed by efficient reliability methods, rather than a multifold integral, which may be cumbersome to evaluate.
9) By-products of the reliability analysis, *i.e.*, importance and sensitivity measures, provide further insight about the computed risk. Examples are the sensitivity measures for risk mitigation and model uncertainty reduction in Chapter 5.

6.2. FUTURE RESEARCH DIRECTIONS

A number of topics are identified for further research in the course of this study. These topics are presented in three categories: probabilistic models, probabilistic analysis methods, and software.

First, it is suggested that further research efforts focus on extending the library of probabilistic models and improving the existing models. In particular, the following developments are envisioned for future research:

- It is suggested that the regional and building models, presented in Chapter 3, be improved by data-gathering efforts. The sensitivity measures in Chapter 5 will guide the allocation of resources for such efforts. The extensive seismic damage and loss databases that are in possession of insurance companies can greatly enhance the predictions of these models.
- It is mentioned earlier in this thesis that buildings are modeled at three levels. In this thesis, models are provided only at the regional and building levels. Future work should implement models for detailed analysis of buildings at the component level. This entails both synthetic ground motion models and probabilistic models for prediction of damage and loss of structural and non-structural components.
- Further research is suggested to develop probabilistic models for bridges, lifelines, and other types of infrastructure with the model format that is described in Chapter 3. By employing such models, it will be possible to carry out comprehensive regional risk analyses, taking into account economic, environmental, and socioeconomic impacts. Models for the seismic damage of bridges are proposed by FEMA-NIBS (2003). However, these models are in the form of conditional probability.
- Inclusion of models for windstorm, snowstorm, and other hazards is suggested. Rt is capable of employing these models in a multi-hazard risk analysis that considers the possible coincidence and interaction of the hazards.
- Smoothing and implementation of more ground shaking intensity models from the literature is suggested. This is necessary in order to use multiple intensity models for each earthquake

source in a risk analysis, which is an accepted practice in contemporary hazard and risk estimations. Rt allows for including an arbitrary number of intensity models for each earthquake source.

Second, the following two topics for research on probabilistic analysis methods are proposed:

- A computationally-efficient but limited multi-hazard reliability analysis option was presented in Chapter 4. This analysis option employs efficient reliability methods, such as FORM, SORM, and importance sampling. The limitation of the method lies in its lack of ability to include time-varying phenomena, such as long-term deterioration and discounting. Further research is suggested to address this limitation. This will significantly reduce the computational cost compared to the scenario sampling analysis, which is currently the only method to include time-dependency. As a first step, it is suggested that the time between events be modeled with exponential random variables. However, this poses several challenges that require further research, such as presence of a varying number of random variables in each step of FORM analysis.
- It is suggested that the sensitivity analysis in Chapter 5 for model retrofit decisions be extended to optimization analysis. While the sensitivity measure in Chapter 5 prioritizes buildings for retrofit, an optimization analysis will determine the optimal amount of retrofit for each building. That is, it determines the optimal amount of resources that should be allocated for each building in order to minimize the risk. The objective in this optimization analysis should be to minimize the tail probabilities of cost exceedance, where the cost includes the cost of *a priori* retrofit actions plus the cost of repair. This contrasts the current optimization analysis schemes, which aim at minimizing the expected cost.

Finally, the following suggestions are made for further development of the computer program Rt:

- Implementation of parallel processing especially for sampling analysis
- Implementation of more interfaces with external software, such as MATLAB, ANSYS, Abaqus
- Computation of second-order direct differentiation response sensitivities and communication of these sensitivities between multiple models; these sensitivities will be employed in SORM analysis
- Computation of sensitivity results from sampling, *i.e.*, the sensitivity of exceedance probabilities with respect to random variable parameters and decision variables; currently, these sensitivities are only computed in a FORM analysis

BIBLIOGRAPHY

Adams, J., and Halchuk, S. (2003). “Fourth generation seismic hazard maps of Canada: values for over 650 Canadian localities for the 2005 National Building Code of Canada.” *Rep. No. Open File 4459,* Geological Survey of Canada, Ottawa, ON, Canada.

Andrianov, G., Burriel, S., Cambier, S., Dutfoy, A., Dutka-Malen, I., De Rocquigny, E., Sudret, B., Benjamin, P., Lebrun, R., Mangeant, F., Conty, R., Pendola, M., and Suau, F. (2007). “Open TURNS, an open source initiative to treat uncertainties, risks'N statistics in a structured industrial approach.” *European Safety and Reliability Conference,* Taylor and Francis/Balkema, Stavanger, Norway, 1935-1942.

Ang, A. H., and Tang, W. H. (2007). *Probability concepts in engineering.* Wiley, New York.

ATC. (1988). “Rapid visual screening of buildings for potential seismic hazards: A handbook.” *Rep. No. ATC-21,* Applied Technology Council, Redwood City, California.

ATC. (1985). “Earthquake damage evaluation for California.” *Rep. No. ATC-13,* Applied Technology Council, Redwood City, California.

Atkinson, G. M., and Boore, D. M. (2003). “Empirical ground-motion relations for subduction-zone earthquakes and their application to Cascadia and other regions.” *Bulletin of the Seismological Society of America,* 93(4), 1703-1729.

I. Berg. (2011). “muParser - A fast math parser library.” <http://muparser.sourceforge.net> (July 15, 2011).

Bielak, J., Askan, A., Fernández, A., Fenves, G. L., Stojadinovic, B., Park, J., Petropoulos, G., Haupt, T., King, R., and Meyer, J. (2005). “Simulation for determining the seismic performance of urban regions.” *6th European Conference on Structural Dynamics.*

Blanchette, J., and Summerfield, M. (2006). *C++ GUI Programming with Qt 4.* Prentice Hall, New Jersey.

Boore, D. M., Joyner, W. B., and Fumal, T. E. (1997). “Equations for estimating horizontal response spectra and peak acceleration from western North American earthquakes: a summary of recent work.” *Seismol.Res.Lett.,* 68(1), 128-153.

Boore, D. M., and Atkinson, G. M. (2008). “Ground-motion prediction equations for the average horizontal component of PGA, PGV, and 5%-damped PSA at spectral periods between 0.01 s and 10.0 s.” *Earthquake Spectra,* 24(1), 99-138.

Box, G. E. P., and Tiao, G. C. (1992). *Bayesian Inference in Statistical Analysis.* Wiley-Interscience, New Jersey.

Breitung, K. (1984). "Asymptotic Approximations for Multinormal Integrals." *J.Eng.Mech.,* 110(3), 357-366.

Building Seismic Safety Council. (1997). "NEHRP Guidelines for the Seismic Rehabilitation of Buildings." *Rep. No. FEMA-273,* Federal Emergency Management Agency, Washington, DC.

Choe, D., Gardoni, P., Rosowsky, D., and Haukaas, T. (2008). "Probabilistic capacity models and seismic fragility estimates for RC columns subject to corrosion." *Reliability Engineering and System Safety,* 93(3), 383-393.

Cornell, C. A. (1968). "Engineering seismic risk analysis." *Bulletin of the Seismological Society of America,* 58(5), 1583-1606.

Cornell, C. A. (1967). "Bounds on reliability of structural systems." *American Society of Civil Engineers Proceedings, Journal of the Structural Division,* 93(1), 171-200.

Cornell, C. A., and Krawinkler, H. (2000). "Progress and challenges in seismic performance assessment." *PEER Center News.*

Deitel, H. M., and Deitel, P. J. (2000). *C++ How to program.* Prentice Hall, New Jersey.

Der Kiureghian, A. (2005). "First-and Second-order Reliability Methods." *Engineering Design Reliability Handbook,* E. Nikolaidis, D. M. Ghiocel, and S. Singhal, eds., CRC Press, Boca Raton, FL.

Der Kiureghian, A., and Taylor, R. L. (1983). "Numerical methods in structural reliability." *Proceedings of the ICASP4.*

Der Kiureghian, A., and Ang, A. H. (1977). "A fault-rupture model for seismic risk analysis." *Bulletin of the Seismological Society of America,* 67(4), 1173-1194.

Ditlevsen, O. (1973). *Structural reliability and the invariance problem.* University of Waterloo, Solid Mechanics Division.

Ditlevsen, O., and Madsen, H. O. (1996). *Structural reliability methods.* John Wiley & Sons, Chichester, UK.

Ellingwood, B. R., Celik, O. C., and Kinali, K. (2007). "Fragility assessment of building structural systems in Mid-America." *Earthq. Eng. Struct. Dyn.,* 36(13), 1935-1952.

Ellingwood, B. R., and Wen, Y. (2005). "Risk-benefit-based design decisions for low-probability/high consequence earthquake events in Mid-America." *Progress in Structural Engineering and Materials,* 7(2), 56-70.

Ezust, A., and Ezust, P. (2006). *An Introduction to Design Patterns in C++ with Qt 4.* Prentice Hall, New Jersey.

FEMA-NIBS. (2003). "Earthquake loss estimation methodology - HAZUS Technical Manual." Federal Emergency Management Agency and National Institute of Building Sciences, Washington, D.C.

Fluck, P., Hyndman, R. D., and Wang, K. (1997). "Three-dimensional dislocation model for great earthquakes of the Cascadia subduction zone." *Journal of Geophysical Research,* 102 B9 20539-20550.

Foschi, R. O., Li, H., Zhang, J., and Yao, F. (2000). "RELAN: a general software package for reliability analysis." Department of Civil Engineering, The University of British Columbia, Vancouver, BC, Canada.

Freeman, S. A., Nicoletti, J. P., and Tyrrell, J. B. (1975). "Evaluations of existing buildings for seismic risk – a case study of Puget Sound naval shipyard, Bremerton, Washington." *Proceeding of the U.S. National Conference on Earthquake Engineering,* EERI, 113-122.

Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Booth, M., and Rossi, F. (2006). "GNU Scientific Library Reference Manual - Revised Second Edition." Network Theory, Ltd., Bristol, UK.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. M. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley Professional, New Jersey.

Gardoni, P., Der Kiureghian, A., and Mosalam, K. M. (2002). "Probabilistic capacity models and fragility estimates for reinforced concrete columns based on experimental observations." *J.Eng.Mech.,* 128(10), 1024-1038.

Goda, K., and Hong, H. P. (2008). "Estimation of seismic loss for spatially distributed buildings." *Earthquake Spectra,* 24(4), 889-910.

Grossi, P., and Patel, C. C. (2005). *Catastrophe modeling: A new approach to managing risk.* Springer Verlag, Heidelberg, Germany.

Gutenberg, B., and Richter, C. F. (1944). "Frequency of earthquakes in California." *Bulletin of the Seismological Society of America,* 34(4), 185-188.

Haldar, A., and Mahadevan, S. (2000). *Reliability assessment using stochastic finite element analysis.* John Wiley and Son, New York.

Hasofer, A. M., and Lind, N. C. (1974). “Exact and invariant second-moment code format.” *Journal of the Engineering Mechanics Division,* 100(1), 111-121.

Haukaas, T. (2008). “Unified reliability and design optimization for earthquake engineering.” *Prob.Eng.Mech.,* 23(4), 471-481.

Haukaas, T., and Der Kiureghian, A. (2007). “Methods and object-oriented software for FE reliability and sensitivity analysis with application to a bridge structure.” *J.Comput.Civ.Eng.,* 21(3), 151-163.

Hohenbichler, M., and Rackwitz, R. (1988). “Improvement Of Second-Order Reliability Estimates by Importance Sampling.” *J.Eng.Mech.,* 114(12), 2195-2199.

Hyndman, R. D., and Wang, K. (1995). “The rupture zone of Cascadia great earthquakes from current deformation and the thermal regime.” *Journal of Geophysical Research,* 100 22,133-22,154.

Imai, K., and Frangopol, D. M. (2000). “Geometrically nonlinear finite element reliability analysis of structural systems. I: theory.” *Computers and Structures,* 77(6), 677-691.

Johnson, A. I. (1953). “Strength, safety and economical dimensions of structures.” *Bulletin of the Division of Building Statics and Structural Engineering at the Royal Institute of Technology,* 22 68.

Kiureghian, A. D., and Zhang, Y. (1999). “Space-variant finite element reliability analysis.” *Comput.Methods Appl.Mech.Eng.,* 168(1-4), 173-183.

Kjellman, W., and Wästlund, G. (1940). *Säkerhetsproblemet i byggnadskonsten.* Generalstabens litografiska anstalts förlag.

Kleiber, M., Hien, T. D., Antúnez, H., and Kowalczyk, P. (1997). *Parameter sensitivity in nonlinear mechanics: theory and finite element computations.* Wiley, West Sussex, UK.

Koduru, S. D., and Haukaas, T. (2010). “Probabilistic seismic loss assessment of a Vancouver high-rise building.” *J.Struct.Eng.,* 136(3), 235-245.

Kramer, S. L. (1996). *Geotechnical earthquake engineering.* Prentice Hall.

Lee, R., and Kiremidjian, A. S. (2007). “Uncertainty and correlation for loss assessment of spatially distributed systems.” *Earthquake Spectra,* 23(4), 753-770.

Liang, H., Haukaas, T., and Royset, J. O. (2007). “Reliability-based optimal design software for earthquake engineering applications.” *Can. J. Civ. Eng.,* 34(7), 856-869.

Liu, P. L., Lin, H. Z., and Der Kiureghian, A. (1989). “CalREL User Manual.” *Rep. No. Report No. UCB/SEMM-89/18,* Department of Civil Engineering, University of California, Berkeley.

Luco, N., Bazzurro, P., and Cornell, C. A. (2004). “Dynamic versus static computation of the residual capacity of a mainshock-damaged building to withstand an aftershock.” *Proceedings of the 13th World Conference on Earthquake Engineering.*

Mahaney, J. A., Paret, T. F., Kehoe, B. E., and Freeman, S. A. (1993). “The capacity spectrum method for evaluating structural response during the Loma Prieta earthquake.” *Proceedings of the 1993 United States National Earthquake Conference,* US Central United States Earthquake Consortium (CUSEC), 501-510.

Marquardt, D. W. (1963). “An algorithm for least-squares estimation of nonlinear parameters.” *Journal of the Society for Industrial and Applied Mathematics,* 11(2), 431-441.

Mayer, M. (1926). *Die Sicherheit der Bauwerke und ihre Berechnung nach Grenzkräften anstatt nach zulässigen Spannungen.* J. Springer.

McCormack, T. C., and Rad, F. N. (1997). “An earthquake loss estimation methodology for buildings based on ATC-13 and ATC-21.” *Earthquake Spectra,* 13(4), 605-621.

McGuire, R. K. (2004). *Seismic hazard and risk analysis.* Earthquake Engineering Research Institute, Berkeley, CA.

McGuire, R. K. (1974). “Seismic structural response risk analysis incorporating peak response regressions on earthquake magnitude and distance.” *Rep. No. Publ. 399,* Department of Civil Engineering, Massachusetts Institute of Technology, Boston, MA.

McGuire, R. K. (2008). “Probabilistic seismic hazard analysis: Early history.” *Earthquake Engineering and Structural Dynamics,* 37(3), 329-338.

McGuire, R. K., and Barnhard, J. A. (1977). “Magnitude, Distance, and Intensity Data for C. I. T. Strong Motion Records.” *Journal of Research of the United States Geological Survey,* 5(4), 437-443.

McKenna, F., Scott, M. H., and Fenves, G. L. (2010). “Nonlinear finite-element analysis software architecture using object composition.” *J.Comput.Civ.Eng.,* 24(1), 95-107.

Mitchell, D., Paultre, P., Tinawi, R., Saatcioglu, M., Tremblay, R., Elwood, K., Adams, J., and DeVall, R. (2010). “Evolution of seismic design provisions in the National building code of Canada.” *Canadian Journal of Civil Engineering*, 37(9), 1157-1170.

Moehle, J., and Deierlein, G. G. (2004). “A framework methodology for performance-based earthquake engineering.” *Proceedings of the 13th World Conference on Earthquake Engineering,* 3812-3814.

Onur, T., Ventura, C. E., and Finn, W. D. L. (2005). “Regional seismic risk in British Columbia - Damage and loss distribution in Victoria and Vancouver.” *Can. J. Civ. Eng.*, 32(2), 361-371.

Petersen, M. D., Frankel, A. D., Harmsen, S. C., Mueller, C. S., Haller, K. M., Wheeler, R. L., Wesson, R. L., Zeng, Y., Boyd, O. S., and Perkins, D. M. (2008). “Documentation for the 2008 update of the United States National Seismic Hazard Maps.” *Rep. No. Open File 2008-1128,* US Geological Survey, Reston, VA.

Prot, M. (1936). “Note sur la notion de coefficient de sécurité.” *Annales des Ponts et Chaussées.*

U. Rathmann, and Wilgen, J. (2011). “Qwt user's guide.” <http://qwt.sourceforge.net> (July 15, 2011).

Riederer, K. A., and Haukaas, T. (2007). “Cost-benefit importance vectors for performance-based structural engineering.” *J.Struct.Eng.*, 133(7), 907-915.

Rojahn, C., Sharpe, R. L., Scholl, R. E., Kiremidjian, A. S., Nutt, R. V., and Wilson, R. R. (1986). “Earthquake damage and loss evaluation for California.” *Earthquake Spectra,* 2(4), 767-782.

Schueller, G. I. (2006). “Structural reliability software.” *Struct.Saf.,* 28(1-2).

Seber, G. A. F., and Wild, C. J. (2003). *Nonlinear regression.* Wiley-IEEE, New York.

Shreiner, D. (2010). *The OpenGL Reference Manual-The Bluebook.* Addison Wesley Professional, Boston, MA.

Steinbrugge, K. V. (1982). “Earthquakes, volcanoes and tsunamis: an anatomy of hazards.” Skandia America Group, New York.

Stroustrup, B. (2000). *The C++ programming language.* Addison-Wesley Professional, Massachusetts.

Sudret, B., and Der Kiureghian, A. (2002). “Comparison of finite element reliability methods.” *Prob.Eng.Mech.,* 17(4), 337-348.

Thenhaus, P. C., and Campbell, K. W. (2003). “Seismic hazard analysis.” *Earthquake Engineering Handbook,* 8-1.

Uzumeri, S. M., Otani, S., and Collins, M. P. (1978). “Overview of Canadian Code Requirements for Earthquake Resistant Concrete Buildings.” *Canadian Journal of Civil Engineering,* 5(3), 427-441.

Ventura, C. E., Finn, W. D. L., Onur, T., Blanquera, A., and Rezai, M. (2005). “Regional seismic risk in British Columbia - Classification of buildings and development of damage probability functions.” *Can. J. Civ. Eng.,* 32(2), 372-387.

Weibull, W. (1939a). *The phenomenon of rupture in solids.* Generalstabens Litografiska Anst.

Weibull, W. (1939b). “A statistical theory of the strength of materials.” *IVA Handlingar (Proceedings of Royal Swedish Academy of Engineering Sciences),* 151.

Weibull, W. (1938). *Investigations into strength properties of brittle materials.* Generalstabens Litografiska Anst.

Wen, Y. K. (1990). *Structural load modeling and combination for performance and safety evaluation.* Elsevier, Amsterdam, Netherlands.

Wen, Y. K., and Ellingwood, B. R. (2005). “The role of fragility assessment in consequence-based engineering.” *Earthquake Spectra,* 21(3), 861-877.

Wesson, R. L., and Perkins, D. M. (2001). “Spatial correlation of probabilistic earthquake ground motion and loss.” *Bulletin of the Seismological Society of America,* 91(6), 1498-1515.

Yang, T. Y., Moehle, J., Stojadinovic, B., and Der Kiureghian, A. (2009). “Seismic performance evaluation of facilities: Methodology and implementation.” *J.Struct.Eng.,* 135(10), 1146-1154.

Youngs, R. R., Chiou, S., Silva, W. J., and Humphrey, J. R. (1997). “Strong ground motion attenuation relationships for subduction zone earthquakes.” *Seismol.Res.Lett.,* 68(1), 58-73.

Appendix A. A DEVELOPER’S GUIDE TO RT

Rt is a growing collection of probabilistic models and analysis algorithms. The objective in this appendix is to guide the developers through the implementation of new models and new analysis algorithms in Rt. This information is not intended for the end-users of the program. This appendix builds upon the description of the software architecture in Chapter 2.

This appendix starts with an overview of classes in Rt. The next sections explain how to implement a new analysis algorithm and a new model in Rt. The last section provides further details about the inner workings of Rt. A step-by-step guide on how to start developing Rt is provided online at www.inrisk.ubc.ca. The latter guide explains how to install the different libraries that are required to develop Rt, and how to compile the Rt’s source code.

It is assumed that the reader is familiar with object-oriented programming in C++ (Stroustrup 2000). Deitel and Deitel (2000) is a comprehensive textbook on how to program in C++. In addition, Rt makes extensive use of the application development framework Qt. Various references are made to features and classes of Qt throughout this appendix. Each reference to Qt is followed by citing a specific page of either of the following two books: Blanchette and Summerfield (2006), which is the official Qt programming book, and Ezust and Ezust (2006), which focuses on advanced topics like design patterns in Qt. The Qt Assistant, which is installed with the Qt package, is also an abundant source of information on Qt.

A.1. OVERVIEW OF CLASSES

It was explained in Chapter 2 that Rt consists of three group of classes, *i.e.*, modules:

- Graphical user interface (GUI) module, which contains the classes that define the user interface

- Domain module, which contains function, model, and parameter classes; implementation of new models is one of the main subjects of this appendix
- Analysis module, which contains classes of reliability and optimization algorithms; implementation of new analysis algorithms is another main subjects of this appendix

In the source code of Rt, three folders named RGUI, RDomain, and RAnalysis contain the header and source files of these three modules. Each new class must be added to the correct folder depending on the module it belongs to.

Chapter 2 presented the composition of classes in the GUI module. In this section, more details on the domain and analysis modules are provided. Figure A-1 shows the inheritance tree for the domain module. The triangles in this figure show the inheritance relationship: The class on the bottom of the triangle, called the subclass, inherits from the class on top, called the base class. RObject is the base class of all domain and analysis classes in Rt. RObject is in turn a subclass of QObject, which is the most fundamental class in Qt. QObject is explained in detail by Ezust and Ezust (2006, 191). In the domain module, RObject is inherited by RParameter, RModel, and RFunction. The RFunction class has no subclasses. RModel has many subclasses, each of which is a probabilistic model. Figure A-1 shows only a few subclasses of RModel. Currently, the probabilistic model library of Rt includes 35 models. The name of all model classes in Rt routinely ends with the word "Model."

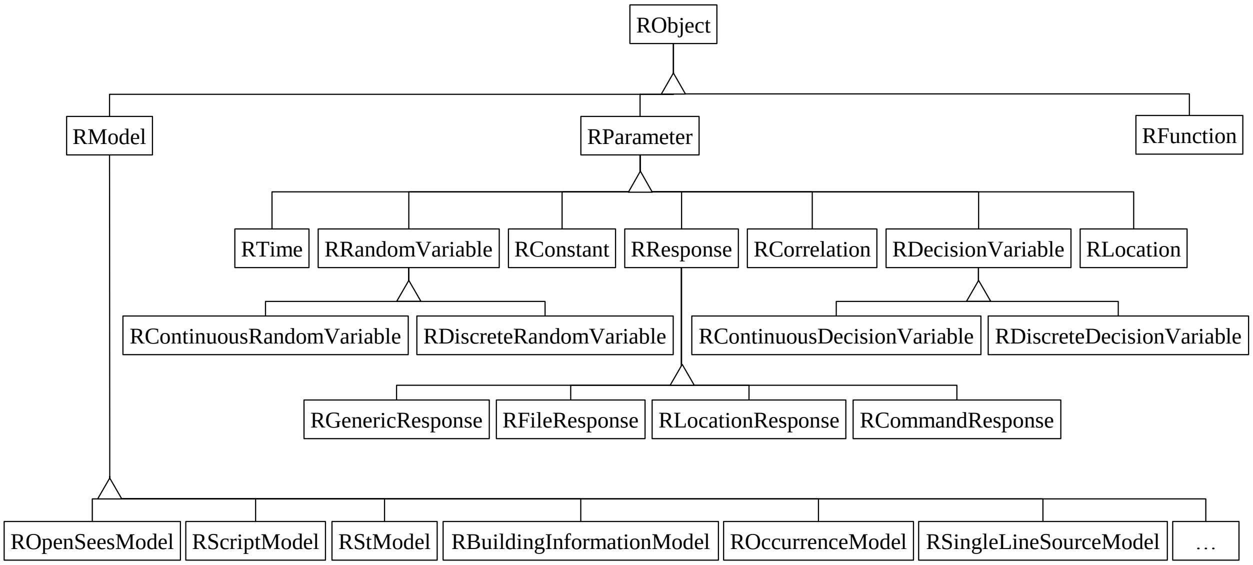


Figure A-1: The inheritance tree for the domain module in Rt.

Figure A-2 shows the inheritance tree for the analysis module. In the current state, the analysis module comprises seven types of analysis tools. Each type is represented by a base class and is inherited by subclasses that define a specific implementation of the base class. For example, RTransformer transforms random variable realizations and gradients of functions between the original and standard normal spaces. It has two specific implementations:

1) RNoCorrelationTransformer for the situation that uncorrelated random variables are transformed
2) RNatafTransformer for the situation that correlated random variables are transformed using the Nataf transformation

As seen in Figure A-2, the convention for naming the analysis tools in Rt is to include the base class name in the name of the subclass.

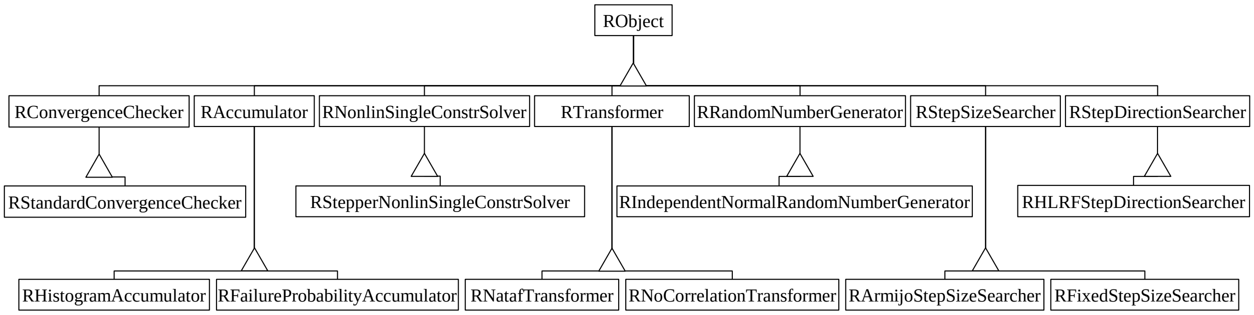

RObject
RConvergenceChecker
RAccumulator
RNonlinSingleConstrSolver
RTransformer
RRandomNumberGenerator
RStepSizeSearcher
RStepDirectionSearcher
RStandardConvergenceChecker
RStepperNonlinSingleConstrSolver
RIndependentNormalRandomNumberGenerator
RHLRFStepDirectionSearcher
RHistogramAccumulator
RFailureProbabilityAccumulator
RNatafTransformer
RNoCorrelationTransformer
RArmijoStepSizeSearcher
RFixedStepSizeSearcher

Figure A-2: The inheritance tree for the analysis module in Rt.

A.2. HOW TO IMPLEMENT A NEW ANALYSIS ALGORITHM

This section provides a step-by-step guideline for implementing a new analysis algorithm in Rt. A new class should be implemented for each new analysis algorithm. In this regard, three key questions are answered in this section:

- **Where to place the implementation of the algorithm:** The algorithm is implemented in a method of the class. The developer is free in choosing a name for that method. In the example in this section, this method is named `analyze`. Step 9 presents an example implementation for this method.
- **How to evaluate a function:** This is explained in the implementation of the `analyze` method. Specifically, items 15) to 18) in Step 9 explain how to evaluate a function.
- **How to make calls to another analysis tool:** If another analysis tool is needed in the algorithm, then that tool must first be provided as input to the model, *i.e.*, as a meta-property. In the example in this section, the algorithm requires an RTransformer object. Step 5 explains how to declare a meta-property for RTransformer. Thereafter, RTransformer is employed in in the `analyze` method to transform probabilities. Specifically, items 10) to 13) in Step 9 explain how the RTransformer object is used.

Step 1: Create the header and source files

Choose a name for the class. The name should start with the letter R. No spaces are allowed. Use uppercase for the first letter of each word. Here, the name RExampleAnalyzer is selected. Conventionally, agent-nouns are used to name analysis tools in Rt, such as "Checker," "Searcher," or in this case, "Analyzer." The name of the files should match the name of the class. Therefore, the name of the header file will be RExampleAnalyzer.h and the name of the source file will be RExampleAnalyzer.cpp. Place the files in the appropriate folder within the source code of Rt. For example, because this is an analysis tool, the files should be placed in the RAnalysis folder. In this example, a folder named RExampleAnalyzer is created within RAnalysis and the files are placed in the new folder.

Step 2: "Include" the files in the developer environment

In case a new folder is created for this class in the source code, the folder path should be included in the developer environment. In order to do this, open Rt.pro, which is located in the source code at Rt/Rt.pro. Under INCLUDEPATH, append ./RAnalysis/RExampleAnalyzer. If the developer environment is Visual Studio 2008, an additional step should be taken: Point to Project ➔ Properties ➔ Configuration Properties ➔ C/C++ ➔ General ➔ Additional Include Directories. Then, append .\RAnalysis\RExampleAnalyzer to the end of the list. Note that in Rt.pro, the character for separating the folders in a path is / (forward-slash), while in Visual Studio, it is \ (backslash).

To include the header and source files, open Rt.pri. Under HEADERS, append ./RAnalysis/RExampleAnalyzer/RExampleAnalyzer.h and under SOURCES, append ./RAnalysis/RExampleAnalyzer/RExampleAnalyzer.cpp. Again, an additional step is needed to include the files in Visual Studio: Right click on the files in the "Solution Explorer" and select "Include In Project".

Step 3: Define the class in the header file

First, define the directives at the beginning of the header file:

```
#ifndef RExampleAnalyzer_H
#define RExampleAnalyzer_H

#include "RObject.h"

class RTransformer;
class RFunction;
```

Define other necessary `#include` directives after the lines above. The last two lines in the code snippet above are necessary, because the RTransformer and RFunction classes will be used in this class, as described shortly. Next, define the RExampleAnalyzer class:

```
class RExampleAnalyzer : public RObject
{
      Q_OBJECT

public:

private:

};
```

`Q_OBJECT` is a macro from Qt that allows the developer to use Qt-specific features in the class (Ezust and Ezust 2006, 209), such as meta-properties, which are described shortly. The RExampleAnalyzer class inherits from RObject. This would not be the case if RExampleAnalyzer was a specific implementation of another class. For example, if RExampleAnalyzer was a new type of merit function checker, it would inherit from RMeritChecker.

Finally, the `#ifndef` directive in the beginning of the header file is completed at the end of the file by `#endif`:

```
#endif // RExampleAnalyzer_H
```

Step 4: Declare the constructor and destructor methods

These two methods are public and therefore, are declared under `public` in the header file. Classes in Rt have a single constructor with two arguments:

- The first argument is a pointer to the parent object. The type of the parent must be QObject. Rt will automatically set the parent of all domain and analysis objects to RDomain at run-time. The parent-child relationship is specific to Qt, and is briefly explained in Chapter 2. For further information, refer to Ezust and Ezust (2006, 192).
- The second argument is the object name. Object names are the identity of objects in Rt and are assigned by the user at run-time. The user will provide a unique object name for each object. The type of the object name variable is QString, which is a Qt-specific string type.

In this example, the constructor and destructor are written as:

```
RExampleAnalyzer(QObject *parent, QString name);
~RExampleAnalyzer();
```

Step 5: Declare meta-properties

Determine which inputs must be provided by the user at run-time for an object from this class. An example is the maximum number of iterations in an optimization class, which will be entered as an integer. Another example is a pointer to another analysis tool, say, an RTransformer that provides the RExampleAnalyzer class with probability transformation services. In Rt, these inputs are defined as meta-properties of the class in the header file. The meta-property system in Rt, which is adopted from Qt, is briefly explained in Chapter 2. Further information is available in Ezust and Ezust (2006, 344). These meta-properties are shown in the Properties Pane of Rt when the object is selected in the Objects Pane. Meta-properties are hereafter called properties for brevity. Declaring each property needs the following four items:

1) A private data member
2) A public method to get the data member, hereafter called the getter method
3) A public method to set the data member, hereafter called the setter method
4) A macro to declare the property name and type, and to associate it with the setter and getter methods

Each property must be assigned an alphanumeric name. The name should be descriptive of what the property is about. The name must start with an uppercase letter. No spaces are allowed. Also, the first letter of each word should be in uppercase. This way, Rt will automatically separate the words when showing the property name in the Properties Pane. For instance, Rt will show the name "SoilCondition" as "Soil Condition".

For demonstration purposes, it is assumed that the RExampleAnalyzer requires six user inputs, which are tabulated in Table A-1.

Table A-1: Characteristics of the properties for RExampleAnalyzer.

Description of the Property	Property Name	Property Type	Data Member Type
Double-precession real number	`DoubleInput`	`double`	`double`
Integer	`IntegerInput`	`int`	`int`
String	`StringInput`	`QString`	`QString`
Boolean value	`BooleanInput`	`bool`	`bool`
RTransformer	`Transformer`	`QObject *`	`QPointer<RTransformer>`

The first four properties in Table A-1 correspond to data members that are not pointers. As shown in Table A-1, the type of the data members for such properties is the same as the type of the property. Below, it is explained how to write the code for these four properties:

1) Data members are declared under `private`. As a naming convention in Rt, the name of such data members consists of "the + property name". The code is:

```
double theDoubleInput;
int theIntegerInput;
QString theStringInput;
bool theBooleanInput;
```

2) Getter methods are declared under `public`. As a naming convention in Rt, the name of such getter methods consists of "get + property name". Getter methods are declared as `const` for data safety. The code is:

```
double getDoubleInput() const;
int getIntegerInput() const;
QString getStringInput() const;
bool getBooleanInput() const;
```

3) Setter methods are also declared under `public`. As a naming convention, the name of setter methods consists of "set + property name". The code is:

```
void setDoubleInput(double value);
void setIntegerInput(int value);
void setStringInput(QString value);
void setBooleanInput(bool value);
```

4) The `Q_PROPERTY` macro is used to declare a property. This macro must be placed in the header file after `Q_OBJECT` and before `public`. The arguments of the macro are:

    a) Property type

    b) Property name

    c) `READ` followed by the name of the getter method

    d) `WRITE` followed by the name of the setter method

    The code is:

```
Q_PROPERTY(double DoubleInput READ getDoubleInput WRITE setDoubleInput)
Q_PROPERTY(int IntegerInput READ getIntegerInput WRITE setIntegerInput)
Q_PROPERTY(QString StringInput READ getStringInput WRITE setStringInput)
Q_PROPERTY(bool BooleanInput READ getBooleanInput WRITE setBooleanInput)
```

The last property in Table A-1 is a pointer to another object. In this example, this property is a pointer to an RTransformer object, which is another analysis tool. In other words, it is assumed here that the RExampleAnalyzer needs the services of another analysis tool to carry out its analysis. Using this RTransfomer pointer, the RExampleAnalyzer will be able to invoke the methods (member functions) of the RTransformer object and thus, make use of its services. The type of the data member that corresponds to this property is `QPointer<RTransformer>`. QPointer is a "guarded" pointer type that is adopted from Qt (Ezust and Ezust 2006, 458). In fact, `QPointer<RTransformer>` is a guarded version of `RTransformer*`. A QPointer has all the characteristics of a normal pointer, except that the QPointer is automatically set to 0 when the object it is pointing to is destroyed. This prevents the program from crashing when the object it is pointing to is deleted by the user. Therefore, in Rt, all pointer data members that belong to a property must be declared as protected pointers.

The pointer property in this example is declared as follows:

1) The data member:

```
QPointer<RTransformer> theTransformer;
```

2) The getter method that must return `QObject*`:

```
QObject *getTransformer() const;
```

3) The setter method that must have `QObject*` in its argument:

```
void setTransformer(QObject *value);
```

4) The macro:

```
Q_PROPERTY(QObject *Transformer READ getTransformer WRITE setTransformer)
```

Note that the data member can be a pointer to any subclass of RObject, *e.g.*, RTransfomer, RParameter, RModel, *etc*. However, regardless of the type of the data member, the type of a pointer property should always be `QObject*`.

Step 6: Declare the key method that will include the algorithm

The class has a key method that contains the implementation of the algorithm. There are *no* requirements or rules for naming such methods. However, in the orchestrating class of the existing analysis options in Rt, a method named `analyze` includes the algorithm. The method is declared under `public`:

```
int analyze(RFunction *theFunction);
```

To demonstrate how analysis algorithms act on the domain objects, it is arbitrarily chosen here to have an RFunction pointer in the argument of this method. It will be shown in the implementation of the method how to make calls to this function to get a list of RParameters that affect the function or how to evaluate the function.

Note that the return value of this method is an integer. This is the case for any method in Rt that performs an operation. The integer shows the status of the operation, and importantly, whether or not there has been any error during the operation. This integer is used for error-handling after the call to the method is complete.

Step 7: Define the constructor and destructor in the source file

First, define the `#include` directives at the beginning of the source file. In addition to RExampleAnalyzer.h, the files RTransformer.h, RFunction.h, RRandomVariable.h, and gsl_vector.h should be included, because they will be used in the class:

```
#include "RExampleAnalyzer.h"
#include "RTransformer.h"
#include "RFunction.h"
#include "RRandomVariable.h"
#include <gsl/gsl_vector.h>
```

Then, define the constructor and destructor methods:

```
RExampleAnalyzer::RExampleAnalyzer(QObject *parent, QString name)
      : RObject(parent, name)
{
      theDoubleInput = 1.0;
      theIntegerInput = 1;
      theBooleanInput = false;
      theTransformer = 0;
}

RExampleAnalyzer::~RExampleAnalyzer()
{

}
```

The data members that should have a default value are initialized in the constructor.

Step 8: Define the getter and setter methods

Defining the setter and getter methods for non-pointer properties is straightforward:

```
double RExampleAnalyzer::getDoubleInput() const
{
      return theDoubleInput;
}

void RExampleAnalyzer::setDoubleInput(double value)
{
      theDoubleInput = value;
}

int RExampleAnalyzer::getIntegerInput() const
{
      return theIntegerInput;
}

void RExampleAnalyzer::setIntegerInput(int value)
{
      theIntegerInput = value;
}

QString RExampleAnalyzer::getStringInput() const
{
      return theStringInput;
}

void RExampleAnalyzer::setStringInput(QString value)
{
      theStringInput = value;
}

bool RExampleAnalyzer::getBooleanInput() const
{
      return theBooleanInput;
}

void RExampleAnalyzer::setBooleanInput(bool value)
{
      theBooleanInput = value;
}
```

The getter and setter methods for the fifth property, which is a pointer property, are:

```
QObject * RExampleAnalyzer::getTransformer() const
{
      return theTransformer;
}

void RExampleAnalyzer::setTransformer(QObject *value)
{
      theTransformer = qobject_cast<RTransformer *>(value);
}
```

When assigning the `theTransformer` in the setter method, an RTransformer pointer is downcast to the passed pointer, `value`. For this purpose, the operator `qobject_cast` is employed, which is similar to the C++ `dynamic_cast` operator. For more information on `qobject_cast`, see Ezust and Ezust (2006, 345).

Step 9: Implement the algorithm

This is the methods that define the functionality of the class. For instance, the `analyze` method contains the algorithm that an object from this class will perform at run-time. A schematic implementation of this method is presented here with the following objectives:

1) To demonstrate how to employ other analysis
2) To demonstrate how to communicate with functions and parameters, *e.g.*, how to evaluate a function and how to set/get the current realization of parameters
3) To present an example of using GSL functions and vectors
4) To provide examples of error-handling
5) To show how to print messages to the Output Pane

```
int RExampleAnalyzer::analyze(RFunction *theFunction)
{
   // Checking if all pointer properties are given by the user
   int status = this->checkProperties();

   // Error handling
   if (status < 0) {

      // Printing an error message to the output pane
      qCritical() << "Error in" << this->objectName() << ":"
                  << "Could not find all the passed objects";

      // Returning the error code
      return -1;
   }

   // Parsing the function and pouplating its parameter lists
   status = theFunction->populateLists();
   if (status < 0) {
      qCritical() << "Error in" << this->objectName() << ":"
                  << "Could not parse the fuction"
                  << theFunction->objectName();
      return -1;
   }

   // Getting the list of random variables that affect the function
   QList<RRandomVariable *> *rvList =theFunction->getRandomVariableList();

   // Getting the number of random variables in rvList
   int numRVs = rvList->count();
   if (numRVs == 0) {
      qCritical() << "Error in" << this->objectName() << ":"
                  << "The function" << theFunction->objectName()
                  << "is not a function of any random variables";
      return -1;
   }

   // Allocating vector u with the size numRVs
   // u represents the random variable vector in the st. normal space
   gsl_vector *u = gsl_vector_calloc(numRVs);

   // Allocating vector x with the size numRVs
   // x represents the random variable vector in the original space
   gsl_vector *x = gsl_vector_calloc(numRVs);

   // Setting the random variable list in theTransformer
   // This also resets theTransfomer for a new analysis
   status = theTransformer->setRandomVariableList(rvList);
   if (status < 0) {
      qCritical() << "Error in" << this->objectName() << ":"
                  << "Could not set the random variable list"
                  << "in the transformer";
      return -1;
   }

   // Transforming vector u to vector x
   status = theTransformer->transformPointToOriginalSpace(u, x);
Line 58    if (status < 0) {
Line 59       qCritical() << "Error in" << this->objectName() << ":"
Line 60                   << "Could not transform the point to original space";
Line 61       return -1;
Line 62    }
Line 63 
Line 64    // Updating random variables parameters using vector x
Line 65    for (int i = 0; i < numRVs; i++) {
Line 66       double xi = gsl_vector_get(x, i);
Line 67       rvList->value(i)->setCurrentValue(xi);
Line 68    }
Line 69 
Line 70    // Updating the isAnalyzed flag of RResponses
Line 71    // due to the new realization of random variables
Line 72    theFunction->updateAnalysisStatus();
Line 73 
Line 74    // Evaluating the function
Line 75    status = theFunction->evaluateFunction(false, RObject::RandomVariable);
Line 76    if (status < 0) {
Line 77       qCritical() << "Error in" << this->objectName() << ":"
Line 78                   << "Could not evaluate the function"
Line 79                   << theFunction->objectName();
Line 80       return -1;
Line 81    }
Line 82 
Line 83    // Getting the function value
Line 84    double fn = theFunction->getFunctionValue();
Line 85 
Line 86    double result;
Line 87    if (theBooleanInput) {
Line 88       result = theDoubleInput * fn;
Line 89    } else {
Line 90       result = theIntegerInput * fn;
Line 91    }
Line 92 
Line 93    // Printing the result to the outpaut pane
Line 94    qDebug() << theStringInput << ":" << result;
Line 95 
Line 96    // Freeing vectors x and u
Line 97    gsl_vector_free(x);
Line 98    gsl_vector_free(u);
Line 99 
Line100    // Returning 0, meaning that the analysis is carried out successfully
Line101    return 0;
Line102 }
```

In brief, the objective in the `RExampleAnalyzer::analyze` method in this example is to evaluate `theFunction` by setting the current realization of random variable to the value at the origin of the standard normal space. To this end, this method carries out the following operations:

1) Line 4: The method `checkProperties` is invoked, which checks if all the pointer properties of this object are provided by the user. This method is inherited from RObject. If

`RObject::checkProperties` returns 0, it means that all pointer properties are provided. Otherwise, it returns a negative value, which means that one or more pointer properties are missing. For instance, if the user does not provide `theTransformer` to this object at run-time, this method reports an error.

2) Lines 7 to 15: Error handling for `RObject::checkProperties`.
    a) Line 7: The if-statement checks if `RObject::checkProperties` has returned a negative value.
    b) Line 10: An error message is printed to the Output Pane using the syntax `qCritical`. More details about printing to the Output Pane are provided shortly.
    c) Line 14: The `RExampleAnalyzer::analyze`, in turn, returns −1 to report the error. As mentioned previously, such methods that carry out an analysis conventionally return an integer in Rt. This integer reports the error status of the analysis.
3) Line 18: `theFunction` is parsed (tokenized) once and for all at the beginning of analysis. This operation must take place before any call is made to evaluate the function. For this purpose, `RFunction::populateLists` is invoked. This method first parses the function and then, populates the parameter lists. These lists are data members of the RFunction and contain the RParameter objects that affect the function, *i.e.*, the RFunction object is an implicit or explicit function of these parameters. One of these lists contains the RRandomVariable objects that affect the function.
4) Lines 19 to 24: Error handling for `RFunction::populateLists`.
5) Line 27: A pointer to the list of random variables in `theFunction` is obtained by calling `RFunction::getRandomVariableList`.
6) Line 30: The number of random variables is obtained by calling `QList::count`.

7) Lines 31 to 36: Error handling for the situation that the number of random variables is zero.
8) Line 40: A GSL vector to represent random variables in the standard normal space is declared and allocated by calling the GSL function `gsl_vector_calloc`. The vector is named `u` and its elements are equal to zero. This represents the origin in the standard normal space.
9) Line 44: A GSL vector to represent random variables in the original space is declared and allocated. The vector is named `x`.
10) Line 48: The list of random variables is passed to `theTransformer` by calling `RTransformer::setRandomVariableList`. This method also resets `theTransformer`.
11) Lines 49 to 54: Error handling for `RTransformer::setRandomVariableList`.
12) Line 57: The vector of standard normal random variables is transformed to the original space by calling `RTransformer::transformPointToOriginalSpace`. The resulting vector is stored in `x`.
13) Lines 58 to 62: Error handling for `RTransformer::transformPointToOriginalSpace`.
14) Lines 65 to 68: A for-loop iterates on the list of RRandomVariable objects and sets their current value to the corresponding element of vector `x`. For this purpose, first the `gsl_vector_get` method reads the value from the vector `x`. Then, the method `RParameter::setCurrentValue` assigns the value to the random variable. Now, the realization of all random variables is at the origin of the standard normal space.
15) Line 72: Before each call to evaluate the function, the `isAnalyzed` flag of RResponse objects must be updated. More information about this flag is provided in Chapter 2. This is done by calling `RFunction::updateAnalysisStatus`. If the argument of this method is left empty, which is the case in this example, the `isAnalyzed` flag of *all* RResponse objects is set to

`false`. In this situation, all models that affect the RFunction object will be evaluated. Conversely, if an RParameter or a list of RParameters is passed to the argument of this method, only the models that have those RParemeter(s) as input will be evaluated.

16) Line 75: The function is evaluated by calling `RFunction::evaluateFunction`. The first argument of this method is a Boolean flag that indicates if the DDM response sensitivities should also be evaluated. The second argument indicates the type of parameter for which the DDM sensitivities are required. In this example, the DDM sensitivities are not needed; thus, the first argument is `false`.
17) Line 76-81: Error-handling for `RFunction::evaluateFunction`.
18) Line 84: The value of the function is obtained by calling `RFunction::getFunctionValue`.
19) Lines 86 to 91: For demonstration purposes, these lines make use of the properties of RExampleAnalyzer. *E.g.*, the function value is multiplied by either `theDoubleInput` or `theIntegerInput` depending on whether `theBooleanInput` is `true` or `false`.
20) Line 94: The result from the previous item is printed to the Output Pane using the syntax `qDebug`. More details about printing to the Output Pane are provided shortly.
21) Lines 97 to 98: The GSL vectors `x` and `u` are freed. It is important to free the memory from GSL objects once they are no longer needed.
22) Line 101: The method `RExampleAnalyzer::analyze` returns 0, which indicates that the analysis is carried out successfully.

The implementation of the class is complete at this step.

It is possible to write messages to the Output Pane in Rt anywhere within the code. Two examples were provided in the implementation of the `RExampleAnalyzer::analyze` method:

- The one with the syntax `qCritical` is employed when an error occurs. Rt will print the messages that are handled by this syntax in red.
- The one with the syntax `qDebug` is employed to print standard messages. Rt will print the messages that are handled by this syntax in black.

Step 10: Add the class to the GUI

The RExampleAnalyzer is added to the GUI by writing a few lines of code in the RDomain class. This class is located in the source code of Rt in Rt/RDomain. The following items must be implemented in RDomain.cpp:

1) `#include` the header file of the RExampleAnalyzer in RDomain.cpp:

```
#include "RExampleAnalyzer.h"
```

2) Choose where in the tree of objects in the Objects Pane the RExampleCalss should appear. It is assumed in this example that this class will appear in the Analysis tree at the root. Next, choose an alphanumeric title, which will be shown in the objects tree. The tile that is chosen in this example is "Example Analyzer". Finally, in the `RDomain::getSubClasses` method and under the appropriate if-statement, append the selected title as a string:

```
else if (className == QString("Analysis").remove(' ')) {
      result << "Nonlinear Single Constraint Solver"
             << "Step Size Searcher" << "Step Direction Searcher"
             << "Merit Checker" << "Transformer" << "Convergence Checker"
             << "Random Number Generator" << "Accumulator"
             << "Example Analyzer";
}
```

In this code snippet, the new part of the code that is appended to the existing code is shown in boldface. This code adds one more item under the Analysis tree in objects pane, which reads "Example Analyzer".

3) In the `RDomain::createObject` method, add the code necessary to instantiate object from the RExampleAnalyzer class. For this purpose, add an "`else if`"-block to the implementation of the method in the following format:

```
else if ((className == QString("Example Analyzer").remove(' ')) ||
         (className.toLower() == QString("RExampleAnalyzer").toLower())) {

      theLastAddedObject = new RExampleAnalyzer(this, name);
}
```

   Note that the argument of the `QString` syntax in the first line of this code snippet should include the full path of the class in the objects tree. For instance, a "Step Size Searcher" of the type "Fixed" that appears under the branch "Step Size Searcher" in the objects tree would have `QString("Step Size Searcher Fixed")` in this part of the code.

4) In the `RDomain::getObjectList` method, add the code necessary to generate a list of objects in the domain that are instantiated from RExampleAnalyzer. For this purpose, add an "`else if`"-block to the implementation of the method in the following format:

```
else if (className == QString("Example Analyzer").remove(' ')) {
   QList<RExampleAnalyzer *> objectList = findChildren<RExampleAnalyzer *>();

   for (int i = 0; i < objectList.count(); i++) {
      result << objectList[i];
   }
}
```

   The argument of the `QString` syntax in the first line of this code snippet follows the rule that was stated in the item 3) above.

Now, the RExampleAnalyzer is added to the GUI of Rt and the user can instantiate objects from this class by pointing to the Analysis tree in the Objects Pane.

A.3. HOW TO IMPLEMENT A NEW MODEL

This section provides a step-by-step guideline for implementing a new model in Rt. Some of the steps are similar to those required to implement an analysis algorithm. Therefore, the reader is referred to the previous section in those steps for more details.

A new class should be implemented for each new model. In this regard, two key questions are answered in this section:

- **Where to place the implementation of the model:** The model algorithm is implemented in a method of the class that is named `evaluateModel`. This is a pure virtual method of the base class, RModel. Therefore, it must be implemented in every subclass of RModel, *i.e.*, every new model in Rt. Step 8 in this section presents an example implementation for this method.
- **Where to place the implementation of the DDM response sensitivities:** Implementing the DDM response sensitivities is optional. These sensitivities are implemented inside the `evaluateModel` method. Step 9 presents the implementation of the DDM sensitivities for the example model in this section.

Step 1: Create the header and source files

Choose a name for the model. The rules for naming a class were explained in the previous section. Conventionally, the name of the model classes in Rt ends with the word "Model". Here, the name RExampleModel is selected. Therefore, the name of the header file will be RExampleModel.h and the name of the source file will be RExampleModel.cpp. Place the files in the appropriate folder within the source code of Rt. The model classes are placed in RDomain/RModel. Select a folder depending on whether the model is a hazard, infrastructure, or other types of model. It is assumed here that RExampleModel is a generic model. Hence, the header and source files are placed in RDomain/RModel/Generic.

Step 2: "Include" the files in the developer environment

Open Rt.pri. Under HEADERS, append ./RDomain/RModel/Generic/RExampleModel.h and under SOURCES, append ./RDomain/RModel/Generic/RExampleModel.cpp. An additional step is needed to include the files in Visual Studio: Right click on the files in the "Solution Explorer" and select "Include In Project".

Step 3: Define the class in the header file

First, define the directives at the beginning of the header file, and then, define the class. All model classes inherit from RModel:

```
#ifndef RExampleModel_H
#define RExampleModel_H

#include "RModel.h"

class RExampleModel : public RModel
{
  Q_OBJECT

public:
  RExampleModel(QObject *parent, QString name);
  ~RExampleModel();

  int evaluateModel(bool evaluateDDM, RObject::RParameterType parameterType);

private:

};

#endif // RExampleModel_H
```

For details about the `Q_OBJECT` macro and the constructor and destructor methods refer to the previous section. As mentioned earlier, the `evaluateModel` method must be implemented in every model in Rt and it contains the model algorithm. As shown in the code snippet above, this method is declared under `public` in the header file of the RExampleModel. A schematic implementation of this method will be presented shortly.

Step 4: Declare properties

Determine which inputs must be provided to this model by the user at run-time. As explained in detail in the previous section, these inputs are defined as properties of the class in the header file. A model class usually has a number of RParameter objects as input, which form the input parameters of the model. Two types of input are considered here:

- For each prescribed input parameter, a single property is declared
- When the number of input parameters is unknown, a list property is declared

For demonstration purposes, it is assumed that the RExampleModel requires one prescribed parameter called `ParameterInput` and a list of unlimited number of parameters called `ParameterList`. The characteristics of these properties are presented in Table A-2.

Table A-2: Characteristics of the properties for RExampleModel.

Description of the Property	Property Name	Property Type	Data Member Type
Parameter	`ParameterInput`	`QObject *`	`QPointer<RParameter>`
List of parameters	`ParameterList`	`QString`	`QList<QPointer<RParameter> >`

The first property in Table A-2 is a pointer to an RParameter. The type of its data member is `QPointer<RParameter>`. More information on properties of pointer type is provided in the previous section. This property, its data member, and its setter and getter methods are declared as follows:

1) The data member:

```
QPointer<RParameter> theParameterInput;
```

2) The getter method:

```
QObject *getParameterInput() const;
```

3) The setter method:

```
void setParameterInput(QObject *value);
```

4) The macro:

```
Q_PROPERTY(QObject *ParameterInput READ getParameterInput WRITE setParameterInput)
```

Complete information on where to insert the above lines of code in the header file was provided in the previous section.

The second property in Table A-2 is unique, because it is a list of several pointers. This type of property is useful when the number of inputs to the model is unknown. An example is a ground motion model computes the earthquake intensity for many buildings in a region. The user must provide the natural period of all buildings as input to this model. Therefore, the model needs a property that can accept an unknown number of parameters as input. According to Table A-2, the data member that stores these parameters is of the type `QList<QPointer<RParameter>>`. It is essentially a QList of protected pointers. QList is a sequential container in Qt. It is similar to, but more powerful than the list object in C++ standard library. More information on QList is available in Blanchette and Summerfield (2006, 251) and Ezust and Ezust (2006, 96). The following rules are established in Rt for properties of this type:

- The name of such properties must contain the word "List"
- The type of such properties must be QString
- Properties of QString type should *not* have the word "List" in their name unless they are of this type (a list of parameters)

For this property, the user will enter several parameter names as a string in the Properties Pane. The names are separated by space, comma, or semicolon. The setter method will convert this string to a list of parameters and stores it in the data member. When the GUI needs to show the parameter names in the Properties Pane, the getter method creates a string by assembling the name of parameters in the list. The implementation of the setter and getter methods is described shortly. This property is declared in the heather file as follows:

1) The data member:

```
QList<QPointer<RParameter> > theParameterList;
```

2) The getter method that must return QString. Contrary to the getter method of other property types, this method cannot be `const`, because it returns a QString that is not `const`:

```
QString getParameterList();
```

3) The setter method that must have a QString in its argument:

```
void setParameterList(QString value);
```

4) The macro:

```
Q_PROPERTY(QString ParameterList READ getParameterList WRITE setParameterList)
```

Step 5: Declare the output responses

With the exception of a few models, the response objects of most models are generated automatically at run-time. For this purpose, the model responses are instantiated in the constructor of the model class. Let's assume that the RExampleModel has a response object named `theResponse`. This object is declared under `private` in the header file of the class as follows:

```
RResponse *theResponse;
```

This object will be instantiated in the constructor method in the next step.

Step 6: Define the constructor and destructor

First, define the `#include` directives at the beginning of the source file. In addition to RExampleModel.h, the RParameter.h and RResponse.h should be included, because they are used in the class:

```
#include "RExampleModel.h"
#include "RParameter.h"
#include "RResponse.h"
```

Then, define the constructor and destructor methods:

```
RExampleModel::RExampleModel(QObject *parent, QString name)
   : RModel(parent, name)
{
   theParameterInput = 0;

   theDomain->createObject(objectName() + "Response", "RGenericResponse");
   theResponse = qobject_cast<RResponse *>(theDomain->getLastAddedObject());
   theResponse->setModel(this);
}

RExampleModel::~RExampleModel()
{

}
```

The constructor must include the code to instantiate the response objects that are to be auto-generated at run-time. The method `RDomain::createObject` orders the domain to instantiate an object of the type `RGenericResponse`. Next, a pointer of the type `RResponse` is downcast to the instantiated response object and is stored in the data member `theResponse`. Finally, the `RResponse::setModel` method lets `theResponse` object know that it is an output for the RExampleModel. These lines of codes must be written separately for each response of the model. It is reemphasized that the data members with a default value must be initialized in the constructor.

Step 7: Define the getter and setter methods

The getter and setter methods for the first property, which is a pointer to an RParameter, are defined in the source file as follows:

```
QObject * RExampleModel::getParameterInput() const
{
      return theParameterInput;
}

void RExampleModel::setParameterInput(QObject *value)
{
      theParameterInput = qobject_cast<RParameter *>(value);
}
```

Details on implementing the setter and getter methods for a pointer property were provided in the previous section. The getter and setter methods for the second property, which is a list of pointers, are defined as follows:

```
QString RExampleModel::getParameterList()
{
      return parameterListToString(theParameterList);
}

void RExampleModel::setParameterList(QString value)
{
      theParameterList = stringToParameterList(value);
}
```

The `parameterListToString` method builds a string out of the name of the parameters in `theParameterList` data member. Conversely, the `stringToParameterList` method builds a QList of parameter pointers out of the string of parameter names that the user has entered. These two methods are implemented in the RObject class and therefore are inherited by every class in Rt.

Step 8: Implement the model algorithm

The model algorithm is implemented in the `evaluateModel` method. A schematic implementation is provided below:

```
int RExampleModel::evaluateModel(bool evaluateDDM, RObject::RParameterType
                                  parameterType)
{
   double result = 0;

   // Summing up the square of all parameters in theParameterList
   for (int i = 0; i < theParameterList.count(); i++) {
      double parameterValue = theParameterList[i]->getCurrentValue();
      result += parameterValue * parameterValue;
   }

   // Adding theParameterInput to the sum
   result += theParameterInput->getCurrentValue();

   // Updating the output response
   theResponse->setCurrentValue(result);

   // Returning 0, meaning that the model is evaluated successfully
   return 0;
}
```

The method `RParameter::getCurrentValue` returns the current realization of the parameter. The algorithm above first sums up the square of all parameters in `theParameterList` and next, adds the value of `theParameterInput` to the sum. Subsequently, this sum is set as the value of the output response by invoking `RResponse::setCurrentValue`. Finally, 0 is returned which indicates that the model is successfully evaluated. Returning a negative number will inform Rt that evaluating the model has encountered an error.

Step 9: Implement the DDM response sensitivities

As mentioned above, implementing the DDM response sensitivities is optional, and these sensitivities are implemented in the `evaluateModel` method. The Boolean flag `evaluateDDM` that is passed to this method indicates if the DDM sensitivities are needed in the analysis. The implementation for these sensitivities should be placed inside an if-block, which is conditioned upon this flag. This is done for efficiency reasons, *i.e.*, not all model evaluations require computing the sensitivities, *e.g.*, when carrying out a sampling analysis. Here, the schematic implementation of the `evaluateModel` is updated to include the implementation of DDM sensitivities:

```
int RExampleModel::evaluateModel(bool evaluateDDM, RObject::RParameterType
                                 parameterType)
{
   double result = 0;

   // Summing up the square of all parameters in theParameterList
   for (int i = 0; i < theParameterList.count(); i++) {
      double parameterValue = theParameterList[i]->getCurrentValue();
      result += parameterValue * parameterValue;
   }

   // Adding theParameterInput to the sum
   result += theParameterInput->getCurrentValue();

   // Updating the output response
   theResponse->setCurrentValue(result);


   // Evaluating DDM response sensitivities
   if (evaluateDDM) {

      // Declaring a QMap for the sensitivities with respect to theResponse
      QMap<RParameter *, double> theRGradientMap;

      // Differentiating theResponse with respect theParameterList members
      for (int i = 0; i < theParameterList.count(); i++) {
         double dR_dParameteri = 2 * theParameterList[i]->getCurrentValue();
         theRGradientMap.insertMulti(theParameterList[i], dR_dParameteri);
      }

      // Differentiating theResponse with respect to theParameterInput
      double dR_dParameterInput = 1.0;
      theRGradientMap.insertMulti(theParameterInput, dR_dParameterInput);

      // Adding theRGradientMap to theDDMMap
      theDDMMap.insertMulti(theResponse, theRGradientMap);
   }

   // Returning 1, meaning the model and its DDM are evaluated successfully
   return 1;
}
```

The code that is added or changed compared to the previous implementation is shown in boldface. The if-statement checks whether the DDM sensitivities should be evaluated. Inside the if-block, first a QMap object is declared for each output response of the model. A QMap is an associative container, which is a collection of (key, value) pairs. For each output response, a

QMap object associates the value of the sensitivity (with the type `double`) to the input parameter that the response is differentiated with respect to (with the type `RParameter*`). For more information on QMap objects, see Blanchette and Summerfield (2006, 260). In this example, the QMap object is named `theRGradientMap`. The for-loop in this example iterates on the input parameters in `theParameterList` and computes the derivative of the output response with respect to each parameter. This derivative is then stored in the `theRGradientMap` using the `QMap::insertMulti` method. The first argument in this method is a pointer to the input parameter that the response is being differentiated with respect to. The second argument is the sensitivity as a double. In the last line inside the if-block, `theRGradientMap` is stored in another QMap object named `theDDMMap`. This object is a protected data member of the RModel class and is inherited by all model classes as a private data member. In other words, there is no need to declare this data member. `theDDMMap` relates an output response to its QMap of sensitivities. In this example, the model has only one output response; hence, this operation is carried out once. The `QMap::insertMulti` method is invoked, where its first argument is a pointer to the output response, namely `theResponse`, and the second is the QMap of sensitivities, namely `theRGradientMap`.

Note that the return value of the `evaluateModel` method has changed here. Now, 1 is returned instead of 0. This will tell Rt that this model does contain the implementation of DDM sensitivities. As a rule, if a model contains DDM implementations, it must return 1 upon successful evaluation of the model and its sensitivities. Conversely, if a model does *not* contain DDM implementations, it must return 0 upon successful evaluation of the model. If this rule is not observed, Rt will disregard the DDM sensitivities.

For error handling, the `evaluateModel` method must return values between −1 to −9 if an error occurs in the evaluation of the model itself. Conversely, if an error occurs in the evaluation DDM sensitivities, the value of −10 is returned.

Step 10: Implement the method to reset the model

The `reset` method consists of the operations that should be carried out to reset the model before the start of an analysis, *e.g.*, a sampling analysis. Implementing this method is optional, and depends on the needs of the model developer. For example, the discounting model stores the current time from the passed Time object in the beginning of an analysis. This will serve as the present time, to which the future costs will be discounted. This method is a virtual method in the RModel class. RModel includes the default (blank) implementation of the method. This method is invoked for all models when the function is tokenized and is evaluated for the first time in an analysis.

Step 11: Implement the methods to reset and update the history variables

The `resetHistoryVariables` method consists of the operations that should be carried out to reset the history variables at the start of a new sample (scenario) in a scenario sampling analysis. Conversely, the `updateHistoryVariables` method consists of the operations that should be carried out to update the history variables at the end of each function evaluation within a scenario. For instance, this method is invoked by the orchestrating algorithm after evaluating the loss for each earthquake event within a scenario. Implementing these two methods is optional, and depends on the needs of the model developer. These are is virtual methods in the RModel class. RModel includes the default (blank) implementation of these methods.

Step 12: Observe the special considerations for hazard models

Two points are made regarding the implementation of hazard models. First, to be able to use a hazard magnitude (severity) or location model in scenario sampling analysis, a method called `getPhysicalParameterList` should be implemented. This method returns a list of parameters that must be sampled at each significant time instant during a scenario. For instance, in an earthquake location model, this method must return the list of parameters that determine the location of the earthquake. An example is the random variables x_1, x_2, and x_3 in the earthquake area source model in Chapter 3. For an earthquake magnitude model, this method must return a list of parameters that determine the magnitude of different events within a scenario, *e.g.*, random variable x in Eq. (3-5). Conversely, model parameters and model error terms should not usually be returned by this method, because the model error does not vary with time, *e.g.*, the error does not improve over time. The `getPhysicalParameterList` method is a virtual method in the RModel class and its default implementation returns an empty parameter list. Assuming that `theParameterInput` is a parameter in RExampleModel that should be re-sampled during a scenario, the `getPhysicalParameterList` method will be implemented as follows:

```
QList<RParameter *> RExampleModel::getPhysicalParameterList()
{
      QList<RParameter *> parameterList;
      parameterList << theParameterInput;
      return parameterList;
}
```

Second, to be able to use a hazard intensity model in any multi-hazard analysis in Rt, the model must return a zero intensity when the incoming magnitude (severity) is zero. For instance, the attenuation relationships that are implemented in Rt produce a zero spectral acceleration when the earthquake magnitude is zero.

Step 13: Add the model class to the GUI

The RExampleModel is added to the GUI by adding a few lines of code to the RDomain class. For this purpose, the following items must be implemented in RDomain.cpp:

1) `#include` the header file of the RExampleModel

```
#include "RExampleModel.h"
```

2) Choose where in the tree of objects in the Objects Pane the RExampleModel should appear. It is assumed in this example that this model is a generic model. Thus, in the `RDomain::getSubClasses` method and under the if-statement for generic models, append a title for this model as a string:

```
else if (className == QString("Model Generic").remove(' ')) {
      result << "Algebraic Expression"
             << "Random Variable With Random Parameters"
             << "Root Finding" << "Script"
             << "Example";
}
```

   In this code snippet, the new part of the code that is appended to the existing code is shown in boldface.

3) In the `RDomain::createObject` method, add the code necessary to instantiate object from RExampleModel. For this purpose, add an "`else if`"-block to the implementation of the method in the following format:

```
else if ((className == QString("Model Generic Example").remove(' ')) ||
         (className.toLower() == QString("RExampleModel").toLower())) {

      theLastAddedObject = new RExampleModel(this, name);
}
```

4) In the `RDomain::getObjectList` method, add the code necessary to generate a list of RExampleModel objects in the domain. For this purpose, add an "`else if`"-block to the implementation of the method in the following format:

```
else if (className == QString("Model Generic Example ").remove(' ')) {
      QList<RExampleModel *> objectList = findChildren<RExampleModel *>();

      for (int i = 0; i < objectList.count(); i++) {
            result << objectList[i];
      }
}
```

Now, RExampleModel is added to the GUI of Rt and the user can instantiate objects of this class by pointing to the Generic branch of the Domain tree in the Objects Pane.

A.4. DETAILS OF THE INNER WORKINGS OF RT

A.4.1. Accessing the Domain and the Main Window within the Class

It was explained in Chapter 2 that RDomain is the parent of all objects in Rt. It produces and returns lists of objects when the `RDomain::findChildren` method is invoked. Further information about the `findChildren` method is available in Ezust and Ezust (2006, 199). Suppose an analysis method is being implemented. The implementation is at a point that a list of all random variables in the domain is needed. To obtain the list, first, a pointer to is downcast to the domain object. Subsequently, the `findChildren` method of the domain object is invoked to obtain the list of random variables. A sample code is provided below:

```
RDomain *theDomain = qobject_cast<RDomain *>(this->parent());
QList<RRandomVariable *> list = theDomain->findChildren<RRandomVariable *>();
```

The first line above downcasts a pointer to the parent of the class that is being implemented, which is always the domain object. The second line invokes the `findChildren` method of the domain object. The argument of the method is `RRandomVariable*`, which indicates that a list of RRandomVariable objects is needed.

Accessing the Main Window object is necessary when the class that is being implemented requires access to the GUI, *e.g.*, for plotting. RMainWindow is the parent of

RDomain. Therefore, accessing the Main Window object is possible by downcasting a pointer to the parent of the domain object. A sample code is provided below:

```
RDomain *theDomain = qobject_cast<RDomain *>(this->parent());
RMainWindow *mainWindow = qobject_cast<RMainWindow *>(theDomain->parent());
```

The first line above downcasts a pointer to the domain object. The second line downcasts a pointer to the Main Window object, which is always the parent of the domain object. The pointer is stored in the `mainWindow` variable. Now, it is possible to call, say, the `RMainWindow::addSubWindow` method to add a new tabbed window to the Central Pane. For instance, to add a plotting window, the following code is implemented:

```
RPlotWidget *thePlot = new RPlotWidget(mainWindow);
mainWindow->addSubWindow(thePlot, "Plot");
```

A.4.2. Link with External Libraries

Rt employs several libraries and application programming interfaces (API), which include Google Maps API, Qwt, GSL, and muParser. In this section, directions are provided to maintain the link between Rt and these libraries.

Rt employs the Google Maps API to display geographical information and to geocode locations. The link between Rt and Google Maps is established by two files:

1) An HTML file named rtmap.html, which is located online at www.inrisk.ubc.ca/rt/rtmap.html. This file is loaded in the RMapWidget class in Rt to create a map. If the server that hosts the file changes, *i.e.*, if the file is moved to another URL, the old URL must be replaced with the new one in the source code of Rt. For this purpose, open the file RMapWidget.cpp, which is located in the source code of Rt at Rt/RGUI/RMapWidget. Then, in the constructor of the RMapWidget class, set the value of

the variable `theMapURL` to the new URL of the HTML file. This variable is a data member of the RMapWidget class with the type QString.

2) A JavaScript file named RGoogle.js, which is located in the source code of Rt in Rt/RGUI/RScript. This file contains the JavaScript code to carry out various operations in the map, including obtaining coordinates, drawing polygons and polylines, *etc*. Since this file is a part of the Rt's source code, there is no need to take any action if the URL of the HTML file changes.

The Qwt library is utilized in Rt for plotting. Rt is compatible with the version 5.2.1 of this library. Rt is statically linked to this library via ".lib" files in Visual Studio and ".a" files in Qt Creator. By following the online instructions at www.inrisk.ubc.ca to install Qwt, the link to these files is automatically established in the Visual Studio project Rt.vcproj or the Qt Creator project Rt.pro. No further action is required for this linkage.

Rt employs GSL for mathematical and matrix computations. Rt is dynamically linked to this library via ".lib" files in Visual Studio and ".a" files in Qt Creator. By following the online instructions at www.inrisk.ubc.ca to install GSL, the link to these files is automatically established. In Windows, Rt requires two additional files from GSL at run-time. These files are libgsl.dll and libgslcblas.dll for the "Release mode" and libgsl_d.dll and libgslcblas_d.dll for the "Debug mode." These files must be placed in the folder where the Rt.exe exists. Currently, these files exist in the Rt/Bin/Debug folder, which hosts the Debug executable file of Rt. These files have also been made a part of the installation-ready package of Rt for Windows, which hosts the Release executable file of Rt. Therefore, unless these files are deleted, no action is required to maintain this linkage.

The muParser library is used in Rt for evaluating mathematical expressions. This library is a part of the source code of Rt at Rt/RDomain/RFunction/MuParser and is compiled with Rt. Therefore, no installation or maintenance is necessary for this library.

Appendix B. Statistics of Model Parameters

This appendix presents the second-moment information for the parameters of the models that are developed in this thesis. This information contains the mean, coefficient of variation, and correlation coefficient between the parameters θ_i for the models in Eqs. (3-7) to (3-20). It also includes the mean and coefficient of variation for the standard deviation of the model error ε, denoted by σ_ε. Some of the correlation coefficients between the θ-parameters are fairly high. The high correlation indicates that the model has the potential to be reduced to a simpler form (Gardoni *et al.* 2002).

B.1. Regional Damage Model

Five set of model parameters are presented here for five zone types: single-residential, multi-residential, commercial, industrial, and comprehensive development. The parameters in Table B-1 to Table B-5 belong to the model that was presented in Eq. (3-7).

Table B-1: Second-moments for the regional damage model parameters for a single-residential zone.

Parameter	Mean	CoV	Correlation Coefficients				
			θ_1	θ_2	θ_3	θ_4	θ_5
θ_1	0.779	0.021	1				
θ_2	−3.981	0.016	−0.62	1			
θ_3	3.126	0.018	0.21	−0.74	1		
θ_4	0.698	0.051	0.12	−0.17	−0.46	1	
θ_5	0.436	0.028	−0.68	0.98	−0.64	−0.24	1
σ_ε	0.117	0.023					

Table B-2: Second-moments for the regional damage model parameters for a multi-residential zone.

Parameter	Mean	CoV	Correlation Coefficients				
			θ_1	θ_2	θ_3	θ_4	θ_5
θ_1	1.415	0.101	1				
θ_2	−11.397	0.109	−0.90	1			
θ_3	3.667	0.119	0.87	−0.98	1		
θ_4	6.641	0.115	0.89	−0.99	0.95	1	
θ_5	0.081	0.145	−0.92	1	−0.97	−0.99	1
σ_ε	0.095	0.023					

Table B-3: Second-moments for the regional damage model parameters for a commercial zone.

Parameter	Mean	CoV	Correlation Coefficients				
			θ_1	θ_2	θ_3	θ_4	θ_5
θ_1	0.580	0.020	1				
θ_2	−6.333	0.030	−0.70	1			
θ_3	2.282	0.039	0.42	−0.72	1		
θ_4	4.126	0.030	0.57	−0.87	0.29	1	
θ_5	0.209	0.042	−0.76	0.99	−0.67	−0.88	1
σ_ε	0.095	0.023					

Table B-4: Second-moments for the regional damage model parameters for an industrial zone.

Parameter	Mean	CoV	Correlation Coefficients				
			θ_1	θ_2	θ_3	θ_4	θ_5
θ_1	0.304	0.013	1				
θ_2	−5.677	0.034	−0.63	1			
θ_3	3.401	0.035	0.40	−0.79	1		
θ_4	2.916	0.036	0.43	−0.72	0.15	1	
θ_5	0.249	0.046	−0.69	0.99	−0.73	−0.75	1
σ_ε	0.100	0.023					

Table B-5: Second-moments for the regional damage model parameters for a comprehensive development zone.

Parameter	Mean	CoV	Correlation Coefficients				
			θ_1	θ_2	θ_3	θ_4	θ_5
θ_1	0.663	0.030	1				
θ_2	−9.461	0.049	−0.80	1			
θ_3	1.215	0.082	0.45	−0.61	1		
θ_4	7.919	0.049	0.77	−0.98	0.43	1	
θ_5	0.117	0.065	−0.83	1	−0.57	−0.98	1
σ_ε	0.086	0.023					

B.2. BUILDING RESPONSE MODELS

The model parameters for the sub-models of the building response model are first presented for 13 building prototypes. These prototypes were introduced in Table 3-2. These sub-models were presented in Eqs. (3-9), (3-10), (3-11) and (3-14). Subsequently, the parameters for the peak drift ratio model in Eq. (3-15) and the peak acceleration model in Eq. (3-16) are presented.

B.2.1. Natural Period Model

The parameters in Table B-6 to Table B-18 correspond to the model in Eq. (3-9), which outputs the natural period of a building. In order to use linear regression to obtain the second moments of the model parameters, natural logarithm is taken from both sides of Eq. (3-9). Therefore, linear regression results in the second moments for ln(θ_1) instead of θ_1. In the following, the second moments for parameters ln(θ_1), θ_2, and σ_ε are presented.

Table B-6: Second-moments for the natural period model parameters for a concrete moment frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			ln(θ_1)	θ_2
ln(θ_1)	−2.223	0.024	1	
θ_2	0.719	0.027	−0.97	1
σ_ε	0.024			

Table B-7: Second-moments for the natural period model parameters for a concrete shear wall building.

Parameter	Mean	CoV	Correlation Coefficients	
			ln(θ_1)	θ_2
ln(θ_1)	−2.229	0.090	1	
θ_2	0.633	0.113	−0.97	1
σ_ε	0.091			

Table B-8: Second-moments for the natural period model parameters for a concrete frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.229	0.090	1	
θ_2	0.633	0.113	−0.97	1
σ_ε	0.091			

Table B-9: Second-moments for the natural period model parameters for a steel moment frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.256	0.035	1	
θ_2	0.794	0.033	−0.97	1
σ_ε	0.035			

Table B-10: Second-moments for the natural period model parameters for a steel braced frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.482	0.028	1	
θ_2	0.794	0.029	−0.97	1
σ_ε	0.031			

Table B-11: Second-moments for the natural period model parameters for a steel light frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.482	0.028	1	
θ_2	0.794	0.029	−0.97	1
σ_ε	0.031			

Table B-12: Second-moments for the natural period model parameters for a steel frame building with concrete shear wall.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.470	0.023	1	
θ_2	0.709	0.027	−0.97	1
σ_ε	0.025			

Table B-13: Second-moments for the natural period model parameters for a steel frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.470	0.023	1	
θ_2	0.709	0.027	−0.97	1
σ_ε	0.025			

Table B-14: Second-moments for the natural period model parameters for a precast concrete building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.229	0.090	1	
θ_2	0.633	0.113	−0.97	1
σ_ε	0.091			

Table B-15: Second-moments for the natural period model parameters for a reinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.229	0.090	1	
θ_2	0.633	0.113	−0.97	1
σ_ε	0.091			

Table B-16: Second-moments for the natural period model parameters for an unreinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.679	0.019	1	
θ_2	0.418	0.039	−0.98	1
σ_ε	0.010			

Table B-17: Second-moments for the natural period model parameters for a wood light frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.766	0.096	1	
θ_2	0.462	0.162	−0.96	1
σ_ε	0.087			

Table B-18: Second-moments for the natural period model parameters for a wood large frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.766	0.096	1	
θ_2	0.462	0.162	−0.96	1
σ_ε	0.087			

B.2.2. Strength-to-weight Ratio Model

The parameters in Table B-19 to Table B-31 correspond to the model in Eq. (3-10), which outputs the strength-to-weight ratio of a building. In order to use linear regression to obtain the second moments of the model parameters, natural logarithm is taken from both sides of Eq. (3-10). Therefore, linear regression results in the second moments for $\ln(\theta_1)$ instead of θ_1. In the following, the second moments for parameters $\ln(\theta_1)$, θ_2, and σ_ε are presented.

Table B-19: Second-moments for the strength-to-weight ratio model parameters for a concrete moment frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.355	0.072	1	
θ_2	0.034	0.124	−0.83	1
σ_ε	0.093			

Table B-20: Second-moments for the strength-to-weight ratio model parameters for a concrete shear wall building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.086	0.0023	1	
θ_2	0.020	0.0055	−0.83	1
σ_ε	0.0024			

Table B-21: Second-moments for the strength-to-weight ratio model parameters for a concrete frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.086	0.0023	1	
θ_2	0.020	0.0055	−0.83	1
σ_ε	0.0024			

Table B-22: Second-moments for the strength-to-weight ratio model parameters for a steel moment frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.530	0.102	1	
θ_2	0.023	0.225	−0.82	1
σ_ε	0.154			

Table B-23: Second-moments for the strength-to-weight ratio model parameters for a steel braced frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.105	0.014	1	
θ_2	0.015	0.035	−0.82	1
σ_ε	0.015			

Table B-24: Second-moments for the strength-to-weight ratio model parameters for a steel light frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.105	0.014	1	
θ_2	0.015	0.035	−0.82	1
σ_ε	0.015			

Table B-25: Second-moments for the strength-to-weight ratio model parameters for a steel frame building with concrete shear wall.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.328	0.012	1	
θ_2	0.015	0.035	−0.82	1
σ_ε	0.015			

Table B-26: Second-moments for the strength-to-weight ratio model parameters for a steel frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.105	0.014	1	
θ_2	0.015	0.035	−0.82	1
σ_ε	0.015			

Table B-27: Second-moments for the strength-to-weight ratio model parameters for a precast concrete building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.229	0.090	1	
θ_2	0.633	0.113	−0.97	1
σ_ε	0.091			

Table B-28: Second-moments for the strength-to-weight ratio model parameters for a reinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−1.086	0.0023	1	
θ_2	0.020	0.0055	−0.83	1
σ_ε	0.0024			

Table B-29: Second-moments for the strength-to-weight ratio model parameters for an unreinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	−2.206	0.039	1	
θ_2	0.027	0.425	−0.94	1
σ_ε	0.051			

Table B-30: Second-moments for the strength-to-weight ratio model parameters for a wood light frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	1.139	0.023	1	
θ_2	0.011	0.171	−0.82	1
σ_ε	0.025			

Table B-31: Second-moments for the strength-to-weight ratio model parameters for a wood large frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	1.139	0.023	1	
θ_2	0.011	0.171	−0.82	1
σ_ε	0.025			

B.2.3. Ductility Model

The parameters in Table B-32 to Table B-44 correspond to the model in Eq. (3-11), which outputs the ductility of a building. In order to use linear regression to obtain the second moments of the model parameters, natural logarithm is taken from both sides of Eq. (3-11). Therefore, linear regression results in the second moments for $\ln(\theta_1)$ instead of θ_1. In the following, the second moments for parameters $\ln(\theta_1)$, θ_2, and σ_ε are presented.

Table B-32: Second-moments for the ductility model parameters for a concrete moment frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.760	0.037	1	
θ_2	0.387	0.095	−0.97	1
σ_ε	0.047			

Table B-33: Second-moments for the ductility model parameters for a concrete shear wall building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.760	0.037	1	
θ_2	0.387	0.095	−0.97	1
σ_ε	0.047			

Table B-34: Second-moments for the ductility model parameters for a concrete frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.760	0.037	1	
θ_2	0.387	0.095	−0.97	1
σ_ε	0.047			

Table B-35: Second-moments for the ductility model parameters for a steel moment frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.790	0.048	1	
θ_2	0.370	0.121	−0.97	1
σ_ε	0.060			

Table B-36: Second-moments for the ductility model parameters for a steel braced frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.790	0.048	1	
θ_2	0.370	0.121	−0.97	1
σ_ε	0.060			

Table B-37: Second-moments for the ductility model parameters for a steel light frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.790	0.048	1	
θ_2	0.370	0.121	−0.97	1
σ_ε	0.060			

Table B-38: Second-moments for the ductility model parameters for a steel frame building with concrete shear wall.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.790	0.048	1	
θ_2	0.370	0.121	−0.97	1
σ_ε	0.060			

Table B-39: Second-moments for the ductility model parameters for a steel frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.790	0.048	1	
θ_2	0.370	0.121	−0.97	1
σ_ε	0.060			

Table B-40: Second-moments for the ductility model parameters for a precast concrete building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.760	0.037	1	
θ_2	0.387	0.095	−0.97	1
σ_ε	0.047			

Table B-41: Second-moments for the ductility model parameters for a reinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.760	0.037	1	
θ_2	0.387	0.095	−0.97	1
σ_ε	0.047			

Table B-42: Second-moments for the ductility model parameters for an unreinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.288	0.091	1	
θ_2	0.470	0.230	−0.98	1
σ_ε	0.066			

Table B-43: Second-moments for the ductility model parameters for a wood light frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.536	0.086	1	
θ_2	0.275	0.349	−0.96	1
σ_ε	0.111			

Table B-44: Second-moments for the ductility model parameters for a wood large frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			$\ln(\theta_1)$	θ_2
$\ln(\theta_1)$	2.536	0.086	1	
θ_2	0.275	0.349	−0.96	1
σ_ε	0.111			

B.2.4. Degradation Factor Model

The parameters in Table B-45 to Table B-57 correspond to the model in Eq. (3-14), which outputs the degradation factor of a building.

Table B-45: Second-moments for the degradation factor model parameters for a concrete moment frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	8.720	0.169	1	
θ_2	1.537	0.108	0.89	1
σ_ε	0.071	0.224		

Table B-46: Second-moments for the degradation factor model parameters for a concrete shear wall building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	8.720	0.169	1	
θ_2	1.537	0.108	0.89	1
σ_ε	0.071	0.224		

Table B-47: Second-moments for the degradation factor model parameters for a concrete frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	7.635	0.318	1	
θ_2	0.609	0.528	0.85	1
σ_ε	0.128	0.224		

Table B-48: Second-moments for the degradation factor model parameters for a steel moment frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	8.720	0.169	1	
θ_2	1.537	0.108	0.89	1
σ_ε	0.071	0.224		

Table B-49: Second-moments for the degradation factor model parameters for a steel braced frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	8.600	0.275	1	
θ_2	1.298	0.204	0.88	1
σ_ε	0.114			

Table B-50: Second-moments for the degradation factor model parameters for a steel light frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	8.600	0.275	1	
θ_2	1.298	0.204	0.88	1
σ_ε	0.114			

Table B-51: Second-moments for the degradation factor model parameters for a steel frame building with concrete shear wall.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	8.600	0.275	1	
θ_2	1.298	0.204	0.88	1
σ_ε	0.114			

Table B-52: Second-moments for the degradation factor model parameters for a steel frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	7.635	0.318	1	
θ_2	0.609	0.528	0.85	1
σ_ε	0.128	0.224		

Table B-53: Second-moments for the degradation factor model parameters for a precast concrete building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	8.600	0.275	1	
θ_2	1.298	0.204	0.88	1
σ_ε	0.114			

Table B-54: Second-moments for the degradation factor model parameters for a reinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	8.720	0.169	1	
θ_2	1.537	0.108	0.89	1
σ_ε	0.071	0.224		

Table B-55: Second-moments for the degradation factor model parameters for an unreinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	7.635	0.318	1	
θ_2	0.609	0.528	0.85	1
σ_ε	0.128	0.224		

Table B-56: Second-moments for the degradation factor model parameters for a wood light frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	4.607	0.198	1	
θ_2	1.372	0.149	0.88	1
σ_ε	0.085	0.224		

Table B-57: Second-moments for the degradation factor model parameters for a wood large frame building.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	8.720	0.169	1	
θ_2	1.537	0.108	0.89	1
σ_ε	0.071	0.224		

B.2.5. Peak Drift Ratio Model

The parameters in Table B-58 correspond to the model in Eq. (3-15), which predicts the peak drift ratio of a building.

Table B-58: Second-moments for the peak drift ratio model parameters.

Parameter	Mean	CoV	Correlation Coefficients						
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
θ_1	−1.382	0.073	1						
θ_2	0.600	0.045	0.84	1					
θ_3	0.306	0.085	−0.46	−0.85	1				
θ_4	1.111	0.017	−0.08	0.29	−0.70	1			
θ_5	0.139	0.036	−0.26	−0.13	0.02	−0.05	1		
θ_6	1.029	0.007	0.42	0.22	−0.11	−0.12	−0.02	1	
θ_7	0.201	0.175	−0.34	−0.11	0.01	0.01	0.60	−0.62	1
σ_ε	0.390	0.0097							

B.2.6. Peak Acceleration Model

The parameters in Table B-59 correspond to the model in Eq. (3-16), which predicts the peak acceleration of a building.

Table B-59: Second-moments for the peak acceleration model parameters.

Parameter	Mean	CoV	Correlation Coefficients					
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
θ_1	−0.711	0.076	1					
θ_2	0.115	0.073	0.71	1				
θ_3	0.285	0.037	−0.08	−0.72	1			
θ_4	0.089	0.169	−0.48	0.24	−0.70	1		
θ_5	0.110	0.021	−0.07	−0.13	−0.07	0.01	1	
θ_6	0.798	0.004	0.28	0.13	−0.15	−0.12	0.57	1
σ_ε	0.224	0.0097						

B.3. Building Damage Models

The model parameters for the building structural damage model in Eq. (3-17) are first presented for 13 building prototypes. Subsequently, the parameters for the non-structural damage models in Eqs. (3-18), (3-19), and (3-20) are presented.

B.3.1. Structural Damage Model

The parameters in Table B-60 to Table B-72 correspond to the model in Eq. (3-17), which predicts the structural damage ratio of a building. The model for some of the building prototypes does not include the explanatory variables of one or two of the building irregularities. This is due to lack of data for those irregularities in those prototypes. In such cases, the θ-parameter of the missing irregularity does not appear in the table.

Table B-60: Second-moments for the structural damage model parameters for a concrete moment frame building.

Parameter	Mean	CoV	Correlation Coefficients								
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
θ_1	4.337	0.008	1								
θ_2	1.305	0.007	0.88	1							
θ_3	0.665	0.010	0.27	0.64	1						
θ_4	0.287	0.014	−0.49	−0.34	−0.24	1					
θ_5	0.392	0.048	0.09	0.14	0.03	−0.06	1				
θ_6	0.247	0.060	−0.12	−0.04	−0.01	0.04	0.16	1			
θ_7	0.924	0.026	0.20	0.33	0.27	−0.15	0.22	0.15	1		
θ_8	0.617	0.020	−0.09	0.02	0.03	0.06	0.23	0.22	0.24	1	
θ_9	0.303	0.034	−0.22	−0.18	−0.17	0.13	0.19	0.20	0.14	0.27	1
σ_ε	0.019	0.021									

Table B-61: Second-moments for the structural damage model parameters for a concrete shear wall building.

Parameter	Mean	CoV	Correlation Coefficients							
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8
θ_1	4.025	0.008	1							
θ_2	1.215	0.006	0.84	1						
θ_3	0.627	0.011	0.21	0.65	1					
θ_4	0.268	0.016	−0.43	−0.24	−0.21	1				
θ_5	0.114	0.172	0.00	0.03	0.01	−0.08	1			
θ_6	0.290	0.033	−0.25	−0.21	−0.16	0.06	0.15	1		
θ_7	0.839	0.023	0.18	0.35	0.26	0.01	0.14	0.13	1	
θ_8	0.388	0.039	0.00	0.16	0.16	0.02	0.15	0.18	0.27	1
σ_ε	0.015	0.022								

Table B-62: Second-moments for the structural damage model parameters for a concrete frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients							
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8
θ_1	5.050	0.011	1							
θ_2	1.359	0.010	0.87	1						
θ_3	0.692	0.015	0.36	0.72	1					
θ_4	0.370	0.032	−0.44	−0.13	−0.04	1				
θ_5	0.650	0.025	−0.08	0.10	0.11	0.16	1			
θ_6	0.293	0.084	−0.03	−0.01	−0.06	−0.03	0.21	1		
θ_7	0.471	0.054	0.05	0.18	0.09	0.16	0.31	0.15	1	
θ_8	0.539	0.041	0.14	0.14	−0.04	−0.09	0.30	0.19	0.24	1
σ_ε	0.020	0.032								

Table B-63: Second-moments for the structural damage model parameters for a steel moment frame building.

Parameter	Mean	CoV	Correlation Coefficients							
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_9
θ_1	4.348	0.008	1							
θ_2	1.322	0.006	0.85	1						
θ_3	0.626	0.011	0.18	0.62	1					
θ_4	0.275	0.014	−0.46	−0.35	−0.29	1				
θ_5	0.268	0.046	−0.12	−0.09	−0.06	−0.03	1			
θ_6	0.284	0.038	−0.24	−0.15	0.00	0.05	0.18	1		
θ_7	0.878	0.024	0.16	0.33	0.35	−0.22	0.14	0.15	1	
θ_9	0.275	0.038	−0.14	−0.15	−0.19	0.04	0.21	0.21	0.13	1
σ_ε	0.017	0.022								

Table B-64: Second-moments for the structural damage model parameters for a steel braced frame building.

Parameter	Mean	CoV	Correlation Coefficients							
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_9
θ_1	4.159	0.009	1							
θ_2	1.282	0.007	0.84	1						
θ_3	0.647	0.012	0.19	0.64	1					
θ_4	0.288	0.016	−0.42	−0.31	−0.29	1				
θ_5	0.171	0.122	−0.04	0.01	0.00	−0.06	1			
θ_6	0.316	0.034	−0.30	−0.26	−0.17	0.09	0.16	1		
θ_7	1.102	0.017	0.12	0.33	0.31	−0.09	0.18	0.16	1	
θ_9	0.224	0.058	−0.13	−0.13	−0.18	0.07	0.17	0.26	0.19	1
σ_ε	0.018	0.022								

Table B-65: Second-moments for the structural damage model parameters for a steel light frame building.

Parameter	Mean	CoV	Correlation Coefficients						
			θ_1	θ_2	θ_4	θ_5	θ_6	θ_7	θ_9
θ_1	5.489	0.011	1						
θ_2	1.238	0.010	0.97	1					
θ_4	0.278	0.022	−0.36	−0.16	1				
θ_5	0.210	0.118	0.03	0.09	−0.06	1			
θ_6	0.330	0.050	−0.19	−0.13	0.05	0.17	1		
θ_7	0.585	0.031	−0.06	0.00	−0.07	0.22	0.22	1	
θ_9	0.347	0.040	−0.27	−0.18	0.13	0.18	0.23	0.24	1
σ_ε	0.020	0.039							

Table B-66: Second-moments for the structural damage model parameters for a steel frame building with concrete shear wall.

Parameter	Mean	CoV	Correlation Coefficients							
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_9
θ_1	4.200	0.010	1							
θ_2	1.251	0.008	0.86	1						
θ_3	0.665	0.012	0.22	0.64	1					
θ_4	0.291	0.016	−0.43	−0.28	−0.23	1				
θ_5	0.407	0.052	0.04	0.12	0.06	−0.01	1			
θ_6	0.308	0.042	−0.19	−0.17	−0.13	0.04	0.14	1		
θ_7	0.997	0.021	0.21	0.34	0.18	0.00	0.23	0.14	1	
θ_9	0.230	0.055	−0.13	−0.12	−0.17	0.05	0.18	0.22	0.20	1
σ_ε	0.020	0.022								

Table B-67: Second-moments for the structural damage model parameters for a steel frame building with masonry infill wall.

Parameter	Mean	CoV	Correlation Coefficients						
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
θ_1	4.875	0.010	1						
θ_2	1.305	0.009	0.82	1					
θ_3	0.614	0.016	0.29	0.72	1				
θ_4	0.375	0.031	−0.45	−0.08	0.04	1			
θ_5	0.587	0.025	−0.05	0.16	0.18	0.18	1		
θ_6	0.266	0.085	0.00	0.02	−0.03	−0.04	0.19	1	
θ_7	0.435	0.053	0.06	0.22	0.15	0.19	0.31	0.14	1
σ_ε	0.018	0.034							

Table B-68: Second-moments for the structural damage model parameters for a precast concrete building.

Parameter	Mean	CoV	Correlation Coefficients								
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9
θ_1	4.229	0.007	1								
θ_2	1.235	0.006	0.87	1							
θ_3	0.651	0.009	0.25	0.64	1						
θ_4	0.254	0.015	−0.46	−0.29	−0.22	1					
θ_5	0.375	0.042	0.04	0.10	0.05	−0.04	1				
θ_6	0.339	0.047	0.03	0.05	0.00	−0.14	0.16	1			
θ_7	0.535	0.020	−0.12	−0.01	−0.01	0.10	0.20	0.18	1		
θ_8	0.372	0.038	0.04	0.07	−0.06	0.01	0.19	0.17	0.23	1	
θ_9	0.203	0.053	−0.13	−0.09	−0.10	0.00	0.18	0.18	0.24	0.21	1
σ_ε	0.016	0.021									

Table B-69: Second-moments for the structural damage model parameters for a reinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients						
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
θ_1	4.110	0.009	1						
θ_2	1.184	0.008	0.82	1					
θ_3	0.623	0.015	0.15	0.64	1				
θ_4	0.269	0.019	−0.40	−0.25	−0.24	1			
θ_5	0.102	0.216	0.01	0.04	0.01	−0.09	1		
θ_6	0.425	0.039	−0.04	0.08	0.11	−0.02	0.15	1	
θ_7	0.781	0.028	0.17	0.35	0.24	0.00	0.15	0.23	1
σ_ε	0.017	0.024							

Table B-70: Second-moments for the structural damage model parameters for an unreinforced masonry building.

Parameter	Mean	CoV	Correlation Coefficients						
			θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7
θ_1	5.297	0.013	1						
θ_2	1.363	0.010	0.81	1					
θ_3	0.619	0.027	−0.14	0.41	1				
θ_4	0.300	0.046	−0.57	−0.27	0.08	1			
θ_5	0.714	0.030	0.28	0.13	−0.22	−0.38	1		
θ_6	0.583	0.034	0.10	0.21	0.21	−0.21	0.25	1	
θ_7	0.573	0.043	0.16	0.20	0.11	−0.28	0.26	0.30	1
σ_ε	0.018	0.042							

Table B-71: Second-moments for the structural damage model parameters for a wood light frame building.

Parameter	Mean	CoV	Correlation Coefficients					
			θ_1	θ_2	θ_4	θ_5	θ_6	θ_7
θ_1	4.328	0.007	1					
θ_2	1.154	0.006	0.97	1				
θ_4	0.170	0.022	−0.24	−0.03	1			
θ_5	0.166	0.055	−0.18	−0.10	0.04	1		
θ_6	0.380	0.032	0.05	0.15	0.13	0.21	1	
θ_7	0.319	0.037	0.01	0.05	−0.12	0.21	0.20	1
σ_ε	0.008	0.018						

Table B-72: Second-moments for the structural damage model parameters for a wood large frame building.

Parameter	Mean	CoV	Correlation Coefficients					
			θ_1	θ_2	θ_4	θ_5	θ_6	θ_7
θ_1	4.524	0.008	1					
θ_2	1.206	0.007	0.96	1				
θ_4	0.178	0.028	−0.34	−0.12	1			
θ_5	0.333	0.030	−0.36	−0.28	0.11	1		
θ_6	0.510	0.029	−0.08	0.02	0.12	0.25	1	
θ_7	0.562	0.022	−0.14	−0.07	−0.08	0.30	0.26	1
σ_ε	0.014	0.018						

B.3.2. Non-structural Damage Models

The parameters in Table B-73, Table B-74, and Table B-75 correspond to the models in Eqs. (3-18), (3-19), and (3-20), respectively, which predict the non-structural damage ratios of a building.

Table B-73: Second-moments for the non-structural drift-sensitive damage model parameters.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	4.868	0.010	1	
θ_2	1.273	0.010	0.99	1
σ_ε	0.036	0.030		

Table B-74: Second-moments for the non-structural acceleration-sensitive damage model parameters.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	1.268	0.007	1	
θ_2	0.193	0.008	−0.66	1
σ_ε	0.024	0.030		

Table B-75: Second-moments for the building content damage model parameters.

Parameter	Mean	CoV	Correlation Coefficients	
			θ_1	θ_2
θ_1	1.300	0.008	1	
θ_2	0.128	0.015	−0.58	1
σ_ε	0.032	0.030		