# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA CIVIL – INGENIERÍA SÍSMICA

## CÁLCULO AUTOMATIZADO DE LA VULNERABILIDAD SÍSMICA EN EDIFICIOS DE CONCRETO REFORZADO

TESIS
QUE PARA OPTAR POR EL GRADO DE:
**DOCTOR EN INGENIERÍA**


PRESENTA:
**CARLO RUIZ CASTILLO**

TUTOR PRINCIPAL
**DR MARIO GUSTAVO ORDAZ SCHROEDER**
INSTITUTO DE INGENIERÍA UNAM

COMITÉ TUTOR

**DR LUIS ESTEVA MARABOTO**
INSTITUTO DE INGENIERÍA UNAM

**DR EDUARDO REINOSO ANGULO**
INSTITUTO DE INGENIERÍA UNAM

**DR JOSÉ ALBERTO ESCOBAR SÁNCHEZ**
INSTITUTO DE INGENIERÍA UNAM

**DR SERGIO MANUEL ALCOCER MARTÍNEZ DE CASTRO**
INSTITUTO DE INGENIERÍA UNAM


CIUDAD DE MÉXICO, NOVIEMBRE DE 2024

**JURADO ASIGNADO (PENDIENTE APROBACIÓN POR EL SACC):**

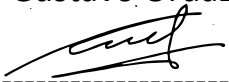
Presidente Dr Luis Esteva Maraboto
Secretario: Dr Eduardo Reinoso Angulo
Vocal: Dr Mario Gustavo Ordaz Schroeder
1er Suplente: Dr José Alberto Escobar Sánchez
2o Suplente Dr Sergio Manuel Alcocer Martínez de Castro

Lugar o lugares donde se realizó la tesis:

**Instituto de Ingeniería, UNAM**

**TUTOR DE TESIS:**

Dr. Mario Gustavo Ordaz Schroeder



---

**FIRMA**

# Automated seismic vulnerability assessment of reinforced concrete buildings

Carlo Ruiz

October 17, 2024

# Contents

# Abstract

We present a conceptual framework, and a computer tool, to achieve a more rational, risk-informed, and hopefully optimal design of structures. Within this framework, we present a novel, experimental approach towards structural design, which can

- Formalize the design process via algorithmic means
- Perform fast time-domain, nonlinear structural analysis of many design alternatives in succession
- Evaluate the consequences of each design alternative both in terms of initial cost as well as in terms of future costs due to earthquakes
- Provide the designer with means to rank design alternatives and choose the best

Following traditional risk-evaluation theory, the implementation shown consists of four independent modules: Design, Hazard, Structural Analysis, and Loss Calculation.

The Design Module takes as input a building specification and generates a detailed and realistic analytical structural model alongside a complete description of the assets contained therein.

The Hazard Module allows the user to specify hazards by providing an annual rate of exceedance of pseudo-acceleration curves and a set of representative ground motion records.

The Structural Analysis Module runs a set of Incremental Dynamic Analyses (IDA) using the predefined records and hazard levels on the designed building. This result set contains the complete time-history response of all degrees of freedom for any desired demand parameters *e.g.* peak drifts, reactions, floor accelerations, etc.

Finally, the Loss Module uses this information to estimate the distribution of losses for the structure's assets. The module outputs a set of loss measures, such as the annual rate of exceedance of loss and the expected annual loss, among others, along with the construction costs.

While this is a standard risk-evaluation procedure, the implementation presented allows for rapid design, analysis, and risk computation of many design

alternatives by automating and abstracting away the difficult and computationally expensive processes such as structural design, analysis, and asset loss estimation. This allows the analyst to effectively explore the space of structural designs to eventually find global optima for a given risk measure, or at least better solutions than those associated with coded design. We also envision that, as our society becomes more risk-averse and computational power cheapens, building codes should tend to increasingly adopt a version of criteria based on monetary —or other types of— risk and initial building cost as ingredients of its measure of good performance.

# Chapter 1

# Introduction

> Las tareas esenciales del ingeniero son inventar y decidir, debe idear opciones y elegir la mejor. Son un placer y obligación imaginar y optimar.
>
> *Emilio Rosenblueth*

Buildings in seismic zones might suffer severe damage and even collapse during high-intensity events, which cause human and monetary losses that impact negatively on society. For this reason, building codes enforce strict rules on designers, aimed at fulfilling limit states of life-safety and collapse-prevention for their corresponding medium and high return periods respectively. However, they consider monetary and human losses only as indirect proxies to good performance, and therefore it is unclear whether these losses are tolerable by the owners or by society. This is unsatisfactory.

Furthermore, risk analysis is rarely carried out in actual structural engineering projects. This is due, in part, to the absence of good tools to do so in a practical manner, because it is a data-and computation-intensive task, particularly for tall buildings or for multiple design proposals. To achieve this, it is necessary to evaluate the behavior of the building against events whose intensity is smaller, close to, and above the design capacity, using sophisticated models that are necessarily nonlinear. This procedure has to be repeated for multiple design options which, for now, is not feasible. However, the author believes it is the responsibility of the engineer to deliver risk measures to society, so they can take decisions accordingly [Hadjian, 2001].

The authors are unaware of a procedure that allows the analyst to effectively explore the design space in search of a global optimum for a given risk measure, or at least better solutions than those associated with codified design. If these tools existed, we would be in a position to understand the effect of structural design on risk, and we could start thinking seriously about moving toward risk-informed design procedures.

## 1.1 Optimum design of buildings

Optimum design has been a primary goal of earthquake engineers since the field's inception; however, it has been relatively unexplored. The first studies on optimum design by Esteva [Esteva, 1967] [Esteva, 1970], Rosenblueth [Rosenblueth, 1976b], Whitman and Cornell [Whitman and Cornell, 1976] made the following hypotheses:

1. that occurrence in time of future earthquakes follows a Poissonian process;
2. that the initial cost of the building and future losses depended on a single intensity measure, $c$, the base shear demand;
3. that the vulnerability model is of complete fragility, that is, when the demand $c$ exceeds the capacity $c_s$ then total loss takes place;
4. that the building is restituted immediately to the same strength and stiffness conditions as it was before it suffered damage.

Even for this simplified approach in which a building is characterized by a single design parameter, it was found that the optimum return period changes depending on the hazard and that this optimum always produces return periods that are shorter for high-seismicity zones and longer for low-seismicity zones [Esteva, 1967], [Rosenblueth, 1976a], [Ordaz et al., 2017]. The main insight derived from this observation was that it is wise to buy lateral strength when it is cheap and to avoid buying too much strength when it is expensive. This means that in areas of low seismicity, one should design for very long return periods, and conversely, one should design for short periods in areas of high seismicity. This interesting result implies that designing structures by specifying a constant return period for all zones is not optimum.

Intuitively, the cost function associated with a design decision parameter $c_s$ is the sum of the initial cost of the building today, $C_I$, plus the present value of the future losses due to earthquakes, $C_{FL}$, and so the *expected present value cost* associated with a design decision is [Ordaz et al., 2017]

$$C(c_s) = C_I(c_s) + C_{FL}(c_s) = C_I(c_s) + (1 + S_L)\mathrm{AAL}/\gamma \tag{1.1}$$

Variable $c_s$ might be interpreted as a measure of the lateral strength, such as a seismic design coefficient, or more broadly, as a building instance. In this equation, $\gamma$ is the discount rate that accounts for the value of money in the future. This factor can also be interpreted as our measure of the importance of failures that would happen far in the future. If it is very low, this implies that we care more about events in the distant future than in the near future. The numerical values of $\gamma$ have been relatively constant across the last 200 years and range from 0.03 to around 0.05 [Picketty, 2014].

The term $(1+S_L)\mathrm{AAL}/\gamma$ is exactly the expected present value of future losses regardless of the vulnerability model [Ordaz et al., 2017] (see their Appendix). In this equation, $\mathrm{AAL} = E(L)\nu_0$ is the average annual loss due to earthquake

perils, where $E(L)$ is the expected loss for the next random earthquake, and $\nu_0$ is the expected number of earthquakes per unit time.

Naive intuition would suggest, perhaps, that future losses are the product of AAL times the life span of the building, *e.g.* 50 times the AAL; however, this is in fact not the case, as future money is usually worth less than what it is worth today. Equation 1.1 shows that future losses are indeed proportional to the AAL, in fact, $1/\gamma$ years of average annual losses. For the standard values of of 0.03 or 0.05 this amounts to 33.3 and 20 years respectively, which is less than the usual lifecycle of a building of around 50 years.

The secondary losses are represented by $S_L$ (as fractions of the future losses $C_{FL}$), which can be caused by the complete or partial collapse of the building, and account for the fact that there might be other factors to protect besides building cost. The numerical value of $S_L$ ranges from 0, in the case where there is no collateral damage, and up to 12 or more, in the case where collapse causes severe collateral damage. In other formulations, this factor could account for the value of lives lost as a consequence of building damage and collapse. This leads naturally to the question of the equivalent monetary value for a human life which is a difficult topic [Salgado-Gálvez et al., 2015] that we will avoid in the present work.

According to game theory, a perfectly rational decision maker would minimize the expected value of the cost function [VonNeumann and Morgernstern, 1953] and not other statistical properties; therefore, optimum design means minimizing the expected value of 1.1 by finding a suitable building design instance. To design a building optimally in this context means that for a fixed $(\gamma, S_L)$ pair, one has to find the building $c_s$, such that $C$ in Eq.1.1 is minimum. The variable $C$ is, again, the expected cost of the decision to build of a design instance today.

Taking this reasoning to its logical conclusion implies that building codes could be drafted to include a single instruction, which would be to demand that the proposed designed building minimizes Eq.1.1 [Rosenblueth, 1976a]. This raises some interesting questions: are there additional restrictions we must place upon design besides optimizing present cost? What if collapses and human losses are too frequent? Maybe a minimum lateral strength should be specified or a minimum return period. Perhaps if the risk underlying the decisions were more easily known, society could have a more direct say in what level of risk is acceptable. And that is the purpose of this thesis.

## 1.2 Automated risk-based design

Efforts towards automating seismic design in steel buildings have been made; [Guan et al., 2020] present a framework called 'AutoSDA' within the performance-based framework, which attempts to significantly reduce the time and effort involved in creating structural designs and running nonlinear analyses. It intends to allow users to bridge the gap between design variables and seismic design and performance outcomes. Unfortunately, it is not clear how to actually go from a proposed design to a design that minimizes some risk measure.

Other authors solve this by formulating the design process as an optimization procedure; [Möller et al., 2015] minimize a similar equation as 1.1, where they take into account the social cost directly as en extra term in the algebraic sum.

They select a set of structural variables $X_i$ such as beam mass per unit length, beam and column section depth, rebar ratios, etc. Then, their numerical optimization algorithm searches the solution space modifying $X_i$ to minimize the objective function constrained to the different performance levels, using neural networks to speed up the dynamic response stage.

They show by sensitivity analyses that the optimal total cost is dependent on the assumed social costs, implying that this value must be considered in detail if we wish to have realistic results.

Rojas et.al [Rojas et al., 2011] formulate risk-based design as minimizing the initial investment in the structure plus the expected annual direct economic losses due to damage to structural and nonstructural components. They use an evolutionary algorithm that iteratively mutates the 'genetic code' of the structure each generation to search for the optimum in parameter space. Using 300 generations, they generated a Pareto front that helps assess the fitness of each candidate structure, and select the optimal frame design. It is important to mention that this procedure is computationally taxing, taking up to 48 hours per run. They show that their tool allows structural engineers with the possibility of quickly evaluate and compare alternative designs, which can provide stakeholders with decision-making tools based on risk measures.

As exposed, minimizing Eq.1.1 or equivalent cost functions is computationally hard and highly sensitive to the cost models. It is necessary to fully automate most if not all the processes in the design, analysis, and loss pipelines. Furthermore, many assumptions have to be made on all models; with regards to seismic hazard definition, record selection and scaling, fragility and vulnerability functions, direct and indirect loss computation, structural analyses implementations, structural model occupancy models, etc. Most of these assumptions have a direct impact on the computed risk, and therefore on the decision of the stakeholders. This thesis aims to explore these aspects in detail.

## 1.3 Objective

This thesis has the following objectives:

1. To develop a framework that allows the designer to achieve a more rational, risk-informed, and hopefully, optimal design of structures.

2. To present a computer tool alongside this framework, which among other things, allows for rapid structural design, analysis, and risk computation of many design alternatives, by way of automating and abstracting away the difficult and computationally expensive processes.

3. To use the presented framework and computer tool to study the influence of the design procedure on its seismic risk.

4. To present a case study analyzing and comparing the level of risk achieved by two different design procedures for the same building specification.

## 1.4 Scope

This study is so far restricted to:

1. Planar framed reinforced concrete buildings designed with the 2023 version of the Mexican code [Gob.CDMX, 2023c] [Gob.CDMX, 2023a] [Gob.CDMX, 2023b].
2. An occupation class and assets of public offices commonly found in Mexico City.
3. A hazard scenario that corresponds to the design period of the buildings and seismicity of Mexico City alongside representative strong motion records recorded in soft soils.
4. Estimation of monetary losses only, which means that no downtime or human losses are considered.

# Chapter 2

# State of the art on seismic risk computation for reinforced-concrete buildings

## 2.1 Background

The problem of building loss estimation due to seismic hazard has been approached at the micro or macro level, that is, individually or for multiple buildings at once.

Regional loss studies conducted in the 70s and 80s introduced concepts such as damage probability matrices [Whitman et al., 1973]. They relate the ground motion intensity expressed in the MMI scale to a level of building damage using a variant of the Delphi procedure, which takes expert opinion into account. These studies often attempt to classify or group buildings by "class" or "taxonomy" which is problematic [Brzev et al., 2013]. These taxonomies are defined in terms of rough structural attributes such as their topology, constituent materials, load-resisting systems, occupancy class, height, etc. Some researchers have suggested different indices to represent a given archetype [Silva et al., 2017].

To do this, one takes into consideration the intra and inter-building variability which is the result of the uncertainty on the building's properties. It is accepted that inter-building variability is the predominant uncertainty because one expects greater variability across different buildings of the same taxonomy than among plausible models of a given structure.

The former approach usually simplified the buildings as single-degree-of-freedom systems due to the computational challenges that are still present to this day. In this manner, devising a method to capture the inter-building variability leads naturally to the usage of one or two-dimensional simplified models,

unfortunately, this always introduces bias to our estimation and shifts the values towards under-conservative estimates of the response because of plan asymmetry and extra damage induced by torsional shears and moments. It is the opinion of the author that calibrating single-degree-of-freedom models is ultimately a dead-end approach. However, with the dawn of cheap computing power granted to us by Moore's law, building-specific loss estimation methods are beginning to surface as the trend forward, and are the main focus of this literature review and this research work.

## 2.2 Building-specific loss estimation

The first building-specific loss estimation procedures started off as deterministic, component-based methods for specifying damage to structural and non-structural elements, as well as to the inventory contents for a given hazard scenario. They employed elastic analyses in combination with response spectra and succeeded in establishing relationships between a *level of damage* and the *intensity measure* such as a spectral ordinate. Their results were expressed as damage factors which is the ratio between repair costs induced by an earthquake of a given intensity and the total replacement value of the building [Ramirez, 2009]. This is termed *scenario-based assessment*. One of the difficulties of this approach which prevails to this day is the lack of empirical data. This difficulty is accentuated for components and elements that are rare or region-specific.

## 2.3 Scenario based assessments

Gunturi [Gunturi, 1992] estimated the monetary loss of a building using non-linear time history analysis. The response of the building was considered deterministic, that is, the properties of the structural design and materials were fixed for a given set of seismic records. The damage suffered by the assets was estimated by mapping the structural response parameters at each story with discrete damage states via simple rules, such as drift thresholds. They scaled the peak ground acceleration linearly in fractions of $g$ and related this value to damage levels for each component at each story.

They employed Park and Ang's energy-based damage index [Park and Ang, 1985] to obtain the corresponding damage level for the structural elements. For non-structural elements and inventory contents, they used simple relationships between peak interstorey drift and peak floor accelerations respectively. An empirical mapping that related damage levels to monetary losses was used, which was mostly based on expert opinion.

Some researchers have proposed simplified models for approximating lateral drift given an intensity measure that is useful when data about the building is scarce or non-existent. This is justified in the consensus that lateral drifts are the main source of the loss suffered by structural elements in multi-storey buildings [Miranda and Reyes, 2002]. Miranda [Miranda, 1999] presents an equiva-

lent continuum structure with non-uniform stiffness composed of a combination of a flexural cantilever beam alongside a shear cantilever beam. Using closed-form expressions and analytical solutions to the equilibrium differential equations of said configuration, simplified expressions were derived for estimating the maximum lateral drift as a function of the spectral acceleration of the first mode and specified level of critical damping:

$$\gamma_i = \frac{\left(\eta N^{\rho}\right)^2 \beta_1 \beta_2 \beta_3 \beta_4}{4\pi^2 N h} A(T_1, \xi)$$

Wherein all the terms except the pseudo-acceleration $A$ are constants that depend upon the dynamical and material properties of the structure in both its elastic and inelastic stages of behavior. This equation is useful for rapid assessment of the overall behavior of a portfolio of buildings. However, these simplified methods are often insufficient because the collapse scenario accounts for a large portion of the losses, even if those intense events are extremely infrequent. This raises the need for more sophisticated numerical models and methods.

Among these advanced numerical methods, the so-called incremental dynamic analysis or IDA for short [Vamvatsikos and Cornell, 2002] [Vamvatsikos, 2004] has gained popularity among researchers due to its intuitive approach and interpretability. To perform an IDA, a refined non-linear model of the structure is subjected to a set of records scaled at varying degrees of intensity, and the corresponding structural response is logged, obtaining thus a "full characterization" of the structural response (including the variability), ranging from elastic to inelastic and finally to the onset of collapse.

In an IDA curve 2.1, intensity usually measured as the first-mode spectral pseudo-acceleration at 5% critical damping $S_a(T_1, \xi = 0.05)$ is plotted against a response, usually the maximum inter storey drift attained during the complete non-linear time history analysis run. In this framework, collapse due to dynamic instability is defined as a disproportionate response to a small increment in the intensity, which mathematically corresponds to the derivative $d\gamma/dS_a$ vanishing, this can be identified easily as a flatline.

## 2.4 Distribution of losses in asset classes

Aslani and Miranda [Aslani and Miranda, 2005] identified the contributions of different asset classes to the overall loss via a disaggregation technique (see 2.2). One notices that drifts cause around 50% of the total loss at different levels of intensity, while only around 30% of loss comes from acceleration-sensitive assets. This justifies developing simplified models for estimating drifts and peak ground accelerations.

In their work, they argued that previous investigations have ignored the relationship between partition-based components, which are highly correlated structural and non-structural elements that need to be replaced whenever there is sufficient damage in a part of the partition.

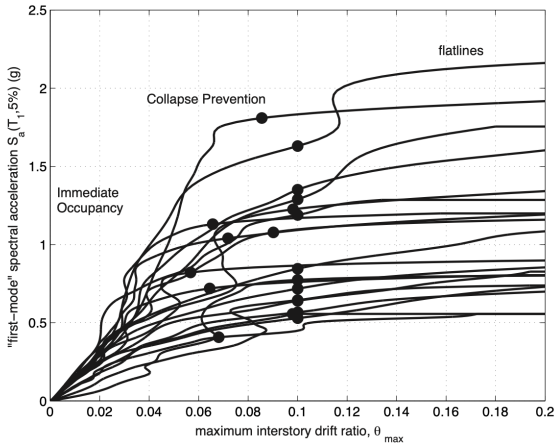



Figure 2.1: A set of IDA curves for a 9 storey steel building. Plotting the maximum inter storey drift ratio attained at any time in the structural analysis against the first mode spectral acceleration at a 5% critical damping level. Each line corresponds to a given record at increasing intensities. The black dots indicate a 80% decrease in "stiffness" and correspond to a definition of the collapse-prevention limit state which is at the onset of dynamic instability.

### 2.4.1 Trends in modeling damage to the exposure dataset

The term "damage" is used to denote an undesirable state of an asset that causes loss to the stakeholders. This is because it forces a repair or replacement action. Damage is usually represented via states or indices, these indices try to capture very complex phenomena such as fracturing, cracking, bending, sliding, and turning over, etc.

The formulation of these indices obeys the following boundary conditions: the first index always corresponds to no damage, and the last one is complete destruction of the asset. One of the first such index scales was implemented by Whitman [Whitman et al., 1973] as a Damage-Probability Matrix (see Table 2.3). This matrix expresses the statistical distribution of having damage state D given a macro-seismic intensity I in the Modified Mercalli Scale and depends on the building of type T (not shown explicitly).

This approach gained traction and multiple entities started defining their indices (see Table 2.4). Scales like HAZUS classify damage into grades; from slight to moderate, heavy, very heavy and finally collapse. Other more elaborate scales utilize a functional scale that includes impacts to nonstructural elements as well [Rojahn et al., 1985], [Hill and Rossetto, 2008].

### 2.4.2 Damage indices for reinforced concrete structures

They mention that these indices are used in practice effectively and work well for assessing damage induced by seismic events. Damage indices have been

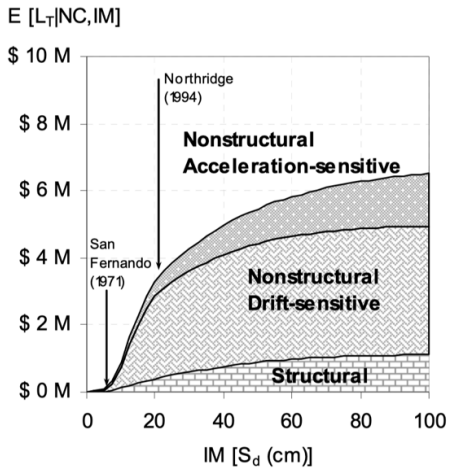



Figure 2.2: Disaggregation of the expected loss at different levels of intensity into losses estimated from structural components, drift-sensitive non-structural components, and acceleration-sensitive non-structural components.

used effectively in the past to predict the damaged state of reinforced concrete buildings and members [Jiang et al., 2011]. There are two main methodologies: those based on structural properties and those on structural dynamics. The first measures the change in structural properties after being subjected to a loading pattern. Many response parameters can act as good proxies for the degree of structural damage: drift, rotation, displacement, strain, dissipated energy, or combinations of these.

Structural property indices have been divided into two kinds: local and global, these refer to the members or to the structure itself. Local damage is also subdivided into cumulative and non-cumulative indices, which allude to the loading cycles being considered in its computation. The second methodology relies on measuring the dynamic response change in the structure, such as the fundamental frequencies, shapes, and damping ratios. This means that damage can be assessed via modal analyses.

Makhloof et al. [D. A. Makhloof, 2021] present an excellent literature review paper on these indices, which have been diagrammed and summarized in (see Fig 2.5)

It has been experimentally observed that the degree of damage for a structural element depends both on the maximum displacement recorded under the earthquake, and on the number of load cycles and hysteretic energy absorbed during the motion. Therefore, most of these indices depend on the number and shape of loading and unloading cycles, as well as the maximum displacement recorded. This empirical finding implies that attempting to quantify the damage state of a reinforced concrete structure using a single parameter such as the

| Damage state | | Earthquake intensity | | | | |
|---|---|---|---|---|---|---|
| | | VI | VII | VIII | IX | X |
| 1 | None | 95 | 49 | 30 | 14 | 3 |
| 2 | Slight | 3 | 38 | 40 | 30 | 10 |
| 3 | Light | 1.5 | 8 | 16 | 24 | 30 |
| 4 | Moderate | 0.4 | 2 | 8 | 15 | 26 |
| 5 | Heavy | 0.1 | 1.5 | 3.0 | 10 | 18 |
| 6 | Major | | 1 | 2 | 4 | 10 |
| 7 | Destroyed | | 0.5 | 1 | 2 | 3 |

Figure 2.3: Damage probability matrix for earthquake intensities VI through X (as shown in ATC-13 [Rojahn et al., 1985]).

| HRC | Hazus | Vision2000 | EMS-98 | ACT-13 | ATC-21 |
|---|---|---|---|---|---|
| none | no damage | fully operational | no damage | none | green tag |
| slight | slight | | slight | slight | |
| light | | operational | light | | |
| moderate | moderate | life safety | moderate | moderate | yellow tag |
| extensive | extensive | near-collapse | heavy | heavy | |
| partial collapse | | | very heavy | major | red tag |
| collapse | collapse | collapse | collapse | destroyed | |

Figure 2.4: Comparison of damage state classifications according to different governing entities [Giovinazzi et al., 2015].

maximum absolute displacement is a rather crude approximation.

structures.

The most used local cumulative index is the Park & Ang expression [Park and Ang, 1985]. In its basic form, it expresses fractional damage as a linear combination of the strain and the normalized hysteretic energic dissipation:

$$DI = \delta_m/\delta_u + \frac{\beta_{PA}}{F_y \delta_u} \int dE \tag{2.1}$$

Where $\delta_m$ is the maximum absolute displacement under the seismic effect, and $\int dE$ refers to the cumulative hysteretic energy dissipated by the member under the loading pattern. Member yield ductility and strength are $\delta_u$ and $F_y$ respectively. The factor that considers the member dissipation capacity is captured by $\beta_{PA}$, which is determined experimentally.

This index is very frequently used in practice and is highly reliable, if care is taken to adapt it and calibrate it correctly to structures in which the shear deformation dominates the flexural failure modes such as in walls or shells.

For reinforced concrete members that have flexure-dominant behavior, $\beta$ can

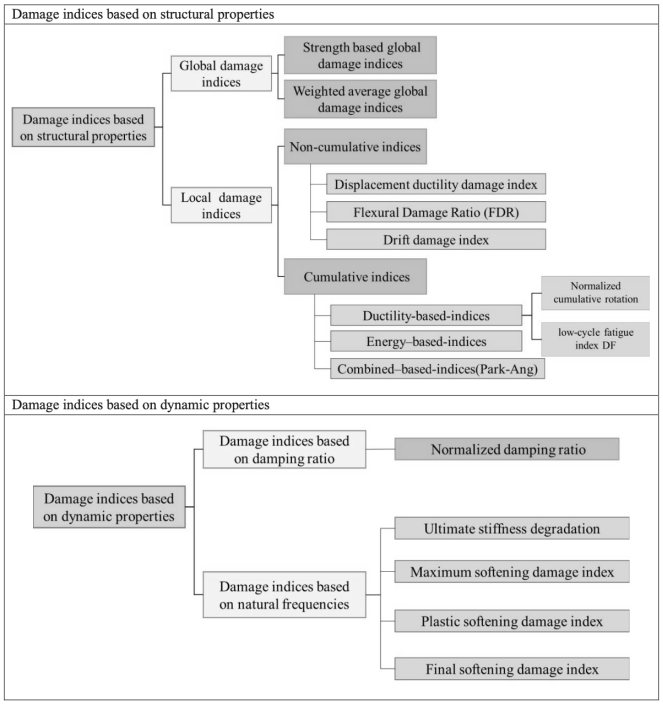



Figure 2.5: Compilation of different damage indices for reinforced concrete structures and elements [D. A. Makhloof, 2021]. These are divided into two main groups: those based on structural properties and those on dynamical properties. The former is subdivided into global/local while the former is subdivided into damping-ratio and those based on natural frequencies or shapes.

be estimated as

$$\beta_{PA} = \rho_w(-0.447 + 0.073\lambda + 0.24n_0 + 0.314\rho_t)$$

Where

- $\lambda$ is $\max(1.7, \lambda_0)$ where $\lambda_0$ is the shear span ratio .
- $n_0$ is the axial compression ratio or 0.2 if $n_0 \leq 0.2$.
- $\rho_t$ is the percentage of longitudinal reinforcement or 0.75% if $\rho_t \leq 0.75\%$.
- $\rho_w$ is the transverse reinforcement ratio.

In equation 2.1, DI ranges from 0 to $\infty$, and can be mapped to a properly defined discrete damage state according to Table 2.6

| $D_{PA}$ | Damage state | Comment |
|---|---|---|
| 0 | No damage | – |
| 0∼0.2 | Minor damage (MID) | Repairable |
| 0.2∼0.4 | Moderate damage (MOD) | – |
| 0.4∼1.0 | Strong damage (SD) | Almost unrepairable (repair cost is very high) |
| >1.0 | Collapse damage (CD) | Total loss of the structure |

Figure 2.6: Discrete damage state based on Park-Ang indices. Note that value at or above 1.0 always corresponds to collapse [D. A. Makhloof, 2021].

### 2.4.3 PBEE probabilistic approach

The concept of discrete damage states was absorbed by the PBEE framework and was generalized into "fragility functions", which contain the probability of attaining or surpassing a specific damage state given structural response (see Figure 2.7).

In component-based methodologies, one uses a fragility function for each component type to get a very accurate estimate of the damage distribution. This estimation consists of selecting a damage state given a response variable via probabilistic simulations so that for a given structural analysis result, each component may end up in a different damage stage which may not be the same if repeated. These densities are commonly assumed to have lognormal distributions, which is a reasonable approximation to empirical evidence.

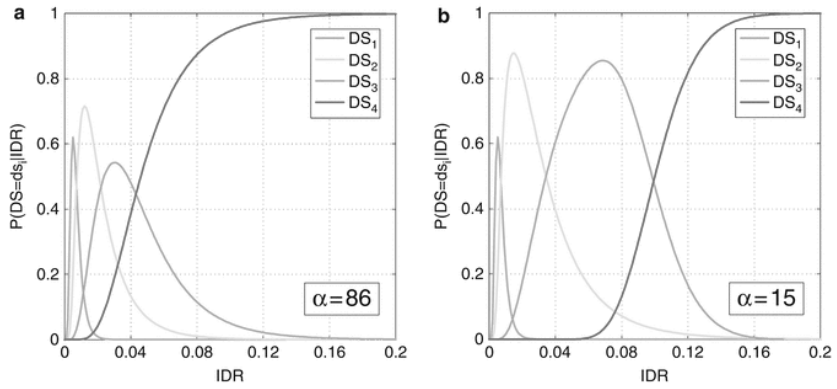



Figure 2.7: Fragility functions corresponding to four damage states of a non-ductile reinforced-concrete column, $\alpha$ measures the level of transverse reinforcement.

We note that the term "fragility" is overloaded in the literature, some researchers tend to associate fragility with the probability of damage given a level of earthquake intensity while others associate with to a monetary loss. We

avoid this ambiguity by always referring to the named conditional probability distribution.

## 2.5 The effect and importance of collapse

It has been widely understood that the phenomenon of building collapse has a significant influence on the overall losses (see Fig 2.8), thus, studies from the 90's onward attempt to use more elaborate damage scales and models where partial collapse and total collapse were also considered. Due to the complexity of the collapse phenomenon, engineers are sometimes led to perform "outside checks" and other ad-hoc procedures to compensate for the model's inability to simulate structural collapse.

Since then, two trends regarding collapse estimation have surfaced: semi-empirical methods based on correlating local parameters *e.g.* column joint rotation to known fragility functions [Elwood, 2004] or numerical methods using sophisticated structural models that can estimate some forms of collapse, such as side-sway due to dynamical instability [López, 2015] [Vamvatsikos, 2004].

Gradually, researchers acknowledged that different collapse mechanisms change the outcome of a loss computation scenario. An example of this phenomenon is the scenario of complete versus partial collapse, which implies complete loss for every asset as well as a high fatality rate for the former, and only a limited amount of non-structural asset loss and perhaps no human fatalities for the latter. There are non-collapse damage states that might cause the building to be demolished completely or partially. This could happen because of excessive foundation settlements or high residual drift. In these cases, the structural loss is high, while non-structural elements, inventory contents or human losses would be negligible. This highlights the fact that it is paramount to distinguish the different collapse mechanisms and types.

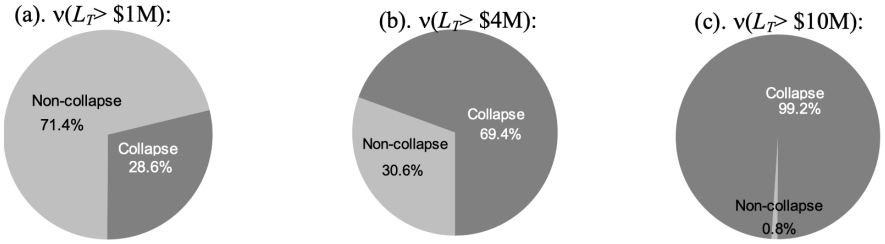



Figure 2.8: Disaggregation of the mean annual frequency of exceedance of the building at three levels to non-collapse and collapse cases. Note that collapse contributes significantly at a medium completely above a certain rate of exceedance level [Aslani and Miranda, 2005]

## 2.6 Loss and its measures

Loss is what stakeholders and society suffer as the result of a seismic event. Classically, the 'triple D' approach to loss modeling has been popular; which means estimating dollars, deaths, and business downtime. Each requires separate treatment and, henceforth, we will focus on dollars only, and refer to ''monetary loss" simply as *loss*. In component-based methodologies, it is assumed for simplicity that the total loss of the building is the sum of the individual component losses. This approach seems intuitive but overlooks some key facts:

- that once assets are in a given damage state, a repair strategy must take action to restore them back to their original state, this strategy might not be the same for two similar components or happen at the same instant.
- that the contracting work done to restore the building to a serviceable status is subjected to economic fluctuations which are highly uncertain.
- that the strategy to repair or replace assets is usually not performed strictly component-wise, this is because they are physically interconnected.

The last bullet point implies that most assets are not statistically independent. The correlation of losses between components depends highly upon the partition-like nature of the layout and their covariance matrices are difficult to estimate [Ramirez and Miranda, 2009]. Figure 2.9 Examples of the correlation between losses in components when different damage states have ocurred between them.

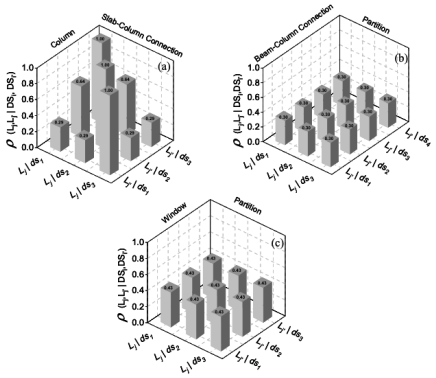



Figure 2.9: Correlation values for partition-like assets; (a) column and slab-column connection, (b) beam-column connection (c) a window and a partition [Aslani and Miranda, 2005]

Stakeholders and society usually care about different measures of loss or scores [McCormack and Rad, 1997], due to this, many such measures have been proposed and used in the past:

- Average annual loss
- Average loss given an earthquake scenario with a fixed intensity
- Average loss given a set of earthquake scenarios with varying intensity
- Maximum probable loss for a given hazard level
- Probability of exceeding a loss value given a hazard level
- Mean annual frequency of exceeding a loss value
- Expected loss for the next event
- Variance of loss for the next event

The chosen measure is usually normalized with respect to the initial construction cost of the component. In this vein, the normalized total loss of the building or a component is sometimes referred to as the "replacement ratio" and is denoted with the literal $\beta$. Since loss depends on the seismic hazard scenario among the other intrinsic uncertainties, it is a random variable, and according to past experiences its density is usually considered lognormal [Ramirez, 2009].

Once the damage state of each asset is established, a correspondence with a decision variable such as loss must be drawn. The most commonly used procedure to estimate the replacement ratio for a given damage state is via a vulnerability function (see figure 2.10). In this procedure, once an asset has achieved a discrete damage state, the probability of having a loss less than or equal to a specified loss level is determined by numerical simulation. Note that counter-intuitively it is possible for $\beta > 1$, as it can sometimes cost more to replace an asset than getting it in the first place, for example, retrofitting partially collapse columns is harder and costs more than building the initial column.

Unfortunately, non-generic vulnerability functions are hard to come by or to create, especially for assets which are region-specific such as a masonry or rare inventory contents such as a new medical devices. This is mainly because little or no empirical data is available. This is an ongoing area of research [Silva et al., 2019].

## 2.7 Software patterns for structural design, structural analysis, loss and risk assessment

Software is concerned with expressing abstract conceptual structures of great complexity. This complexity can be divided into *essential* and *incidental*, where the former refers to an inherent property of the problem to be solved and the latter appears during the solution procedure, usually through the design decisions and the technologies chosen. Details are part of the essential complexity and are very hard to specify because there are "unknown unknowns" and our idea of a goal may change over time.

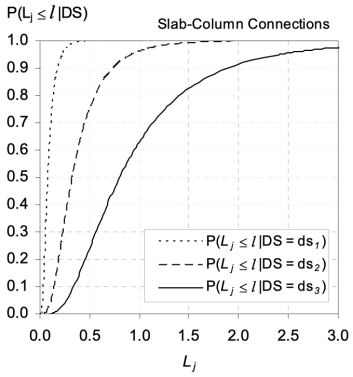



Figure 2.10: Conditional probability of attaining a level of loss given that the component is in damage state $i$, curves developed for the first three damage states of slab-column connections [Aslani and Miranda, 2005]

This means the hard part of software is the specification, design, and testing of this conceptual structure, this is called the essential complexity, and not the labor of representing it and testing the fidelity of the representation, this is incidental complexity [Brooks, 1982].

This statement has two profound implications; first, it states that the essential complexity is far bigger than the incidental complexity by at least an order of magnitude. Which means that even if we reduce software's incidental complexity there is still an underlying essential complexity in the problem to be solved. The statement also implies that a conceptual error is bigger than a syntax error, so we should spend most of our time solving the conceptual real-world problem, and not focus too much on the technology itself. In summary: *there is no silver bullet for software*. This is party because descriptions that try to simplify inherent complexity take away the essence of software. An example of this is trying to turn software into diagrams; software is invisible and not embedded in any space, therefore diagramming software will mutilate its essence. One notices that curiously, diagrams are always incomplete and out-of-date.

Within the essence of software is specification and representation, we show methods and tools to make specification and representation easier. First, we lay out some useful ideas for writing correct and maintainable software. Afterward, we present techniques and abstractions that help reduce incidental complexity.

## Object and types

A breakthrough in the expressiveness and power of programming languages came due to the idea of creating custom or user-defined types. These are usually representations of real-life objects for example; 'FiniteElementModel, Building, Column', or reifications that are inherent to the problem or language themselves, such as 'Viewers, Serializers, Factories, Handlers' , etc [Stroustrup, 2013].

The contemporary notion of object-oriented programming, which deviated from the original conception of Alan Kay [Kay, 1996], tries to create modularity and encapsulation while promoting decoupling and establishing system invariants i.e sources of truth by providing generic interfaces. This is done so that objects can message and work with each other without having to know the specifics of their implementations. It is a common misconception that object-oriented programming is all about classes, inheritance, polymorphism, and hierarchies. Experience has shown that deep inheritance hierarchies that try to model 'is-a' relationships create tightly coupled code that is both difficult to reason about and highly fragile. Most programmers agree that composition (which represents a 'has-a' relationship ) creates looser coupling, therefore one should favor composition over inheritance [Gamma et al., 1995].

## Principles and design patterns for object-oriented programs

There are two flavors of code styles:

1. Imperative, which instructs the computer *how* to perform an action. This style uses assignments and statements heavily, variable definitions, and control flow such as 'if', 'for', 'switch' etc.

2. Declarative, which tells the computer *what* to do. This style is characterized by heavy use of expressions such as function calls or operators that often read like prose.

The author has found the latter to be easier to write, debug, maintain, and to reason about.

It is a strong belief of the author that a program's performance and speed is not paramount. Rather, one must focus on the capacity to extend and maintain a codebase, and to focus on the simplicity and ease of the representation of real-world concepts, this means good abstractions are trademarks of great software.

Code must be written first and foremost for a human to read, and only later for a computer to interpret, compile and execute. This is because code is read by humans much more often than it is written. Therefore, we must write code that is easy to reason about and follow. This is accomplished mainly by reducing lines of code to a minimum. This also combats bugs, as they appear to grow with the square of the number of lines of code. A corollary of this fact is that the best code one can write is no code at all.

Great design is about preventing bugs from being birthed. It is also accomplished by reducing the cognitive load, by keeping logic and sources of truth in a single place. Good code is simple and not clever. Good design is modular, declarative and functional. Good design consists of separating a system into composable units. Great software is *composed* of independent working units. Each unit does one job and does it well.

Do not plan everything ahead, this is because part of the specification is essential complexity and impossible to correctly get fully in advance. Projects that start small and grow are much more successful than ones that try to solve

the most general case from scratch. Every big complex system that ever worked started off from a small simple system that worked well. Incidental coupling arises from broad design, this causes unnecessary complexity. Pick strict constraints and get stuff done.

The software engineering community has produced a set of acronyms that convey wisdom toward writing good object-oriented programs:

1. KISS: Keep It Super Simple, this is another instance of Occam's razor. One shouldn't multiply entities unnecessarily and try to write first the simplest thing that works.

2. DRY: Don't Repeat Yourself, try to avoid code duplication, non-dry solutions are called WET solutions as in "we enjoy typing" or "write everything thrice".

3. AHA and YAGNI: Avoid Hasty Abstractions and You Ain't Gonna Need It, they advocate passing up on early opportunities for abstraction and skipping classes or methods that aren't completely required for the functionality needed *right now*.

4. SOLID:

    - Single responsibility: a class should do one and only one job. Ask, whose responsibility is this? Avoid the so-called 'god-classes' that try to do everything, rather, separate responsibilities into different classes that each do a single job well and compose them to solve the problem.
    - Open for extension, closed for modification; this means that one shouldn't modify the parent class to suit a child's implementation. It encourages the use of ABC (abstract base classes) or interfaces.
    - Liskov substitution principle: posits that pointers from parent classes should be interchangeable with children classes without breaking the application. This emphasizes that children should be supersets of parents in terms of behavior.
    - Interface segregation: states that no class should depend on methods that it doesn't use, it promotes shrinking of interfaces and elimination of dependencies between classes.
    - Dependency inversion: abstractions do not depend on concretions, use ABCs or interfaces for connecting disparate modules.

Some previous points mention interfaces or abstract base classes (ABCs), also known as virtual classes. These are not directly instantiated but rather inherited, and all children must implement their interface. This pattern has served the author well because it forces the programmer to find the right abstraction to avoid essential complexity.

This design pattern works well in practice, as it follows the principle that great software is composed of a set of independent working units. Each unit

does one job and does it well. Software design is best done at the level of interfaces, everything else is an implementation detail. If one thinks of a program as sets entities neatly fitted together, then the interfaces allow us to decouple and write each part independently of others. Writing objects as interfaces decouples specification from implementation, therefore enforcing decoupling and promoting composition. In chapter 4, we expound some interfaces we used such as **FiniteElementModel** or **DesignCriterion**.

## Testing software

Bugs in software are errors in the implementation or flaws in the architecture of the code or the infrastructure running it. They are notorious for causing catastrophic losses multi-million dollar losses [Board and Eddington, 2013]. Software tests are a tool to prevent bugs and assert correct behavior in a reduced set of scenarios.

There are two flavors of tests: verification and validation. Unfortunately, there is no consensus within the software engineering community to what these really mean.

A simple classification is that any implementation must be *verified*, so that it does what it was supposed to.

Therefore verification answers the question: "are we implementing the right equations and procedures?"

A model must be *validated*, which means its results should be compared against reality. Therefore validation answers the question "are we modeling the physical phenomenon correctly?"

## Verification

Verification is the process of checking that we are implementing the right relationships, equations, or logic that we wanted our program to have. Software verification answers the following question: "are we implementing the equations and procedures correctly?" And more simply put; "is the software doing what we want it to do?"

This is accomplished by writing tests that directly run our code and assert that given a scenario, the program outputs or does what we expect. These are usually simple cases. By corollary, software without tests is incomplete. Tests also have the added benefit that they assert behavior invariance across code refactors, this gives the programmer the freedom to refactor and reimplement with confidence.

1. Typings: is for checking sets of values for inputs and outputs. One states that the function accepts only some class of values and returns a class of values. Static type checkers will prevent the code from compiling or being interpreted if these signatures are violated.

2. Property-based testing: is used for checking conditions for a given set of values. It is like typing but goes further and checks exact values from the

class. For example; given the property: "sum of 2 positive numbers is also positive", these tests will take random positive numbers, call the function and check that the result is also positive. It is much more restrictive than plain typing.

3. Contracts: are a powerful mix of typing and property-based testing:

    - like type annotations, contracts are part of the function signature, and can be checked statically.
    - like properties, contracts allow you to specify any conditions, and the framework will take care of choosing exact values, generating test cases, and checking their call results.

There are three types of tests; unit, integration, and end-to-end tests. The author has found unit-tests them to yield the best results-to-effort ratio. Unit tests are code written to test a single function or method from a class. It is conjectured that, if all parts of a whole work correctly, then the union of these parts will do so as well. However, if our input set is infinite, it is not possible all cases with this technique.

Integration testing, as its name suggests, attempts to test the integration between modules, these are hard to mock and do not offer that much benefit for their trouble. Finally, "end-to-end" tests perform complete runs of the scenarios from start to finish mocking real user-flow through a program. This is usually very difficult to accomplish robustly. The line between a unit and an interface test and an interface with an end-to-end test is blurry. The author does not recommend writing interface or end-to-end tests.

There is a recent trend that highly emphasizes tests called "test-driven" development. Its mantra is: red-green-refactor. They advocate writing failing tests first (the red part) that capture what we want, then writing code to make the tests pass (the green part). Finally, they recommend refactoring the code to meet other standards, such as style or coding principles. The author has found it difficult to say whether one should write tests first, and whether unit tests are enough to be certain about the correctness of an implementation. To conclude, software can never be verified completely, so no software is completely "correct" or "free of bugs", however, by verifying we can assert that it is no yet wrong *yet*.

## Validation

Software validation answers the following question: "are we modeling the physical phenomenoma correctly?". In other words, is our implementation valid? This is a deep scientific question. For our purposes, a necessary yet not sufficient condition is that the results of the program match what has been measured by experiment. This implies that we are at least not violating physical law. The tolerance of this match is up to the engineer, and for structural and geotechnical engineering we are usually content with a 10% error margin. Numerical model

validation can be done by comparing our results to known valid experimental data and theoretical results. Experimental results can also be used to calibrate our model. Theoretical results are useful here, and can be implemented easily with unit tests. For example, when some parameters tend to asymptotic values such as $n \to \infty$ or $\theta \to 0$ can be easily checked with the proper tests. This will increase confidence in our model.

# Chapter 3

# Description of the conceptual framework and program implementation

To achieve the objective of this research, the following steps are required:

1. Formalize structural design via algorithmic means
2. Automate the placement of assets according to an occupancy class
3. Specify the corresponding seismic hazard and representative ground motion records
4. Run IDA analysis on the designed building
5. Compute losses and risk measures using the results from the IDA and the hazard and assets

Following traditional risk-evaluation theory, and identifying the independent statements in these steps, the implementation developed consists of four independent modules, each with a single responsibility (see Fig.3.1): Design, Hazard, Structural Analysis, and Loss Calculation. Each of these modules attempts to abstract away the complex processes, such as structural design or analysis, and minimize the possible human errors;

- The goal of the design module is to take in a minimal building specification and produce a complete finite element model, along with a detailed description of the building assets.
- The goal of the hazard module is to generate a hazard object that contains the annual rate of exceedance of intensity and to store its corresponding representative records.

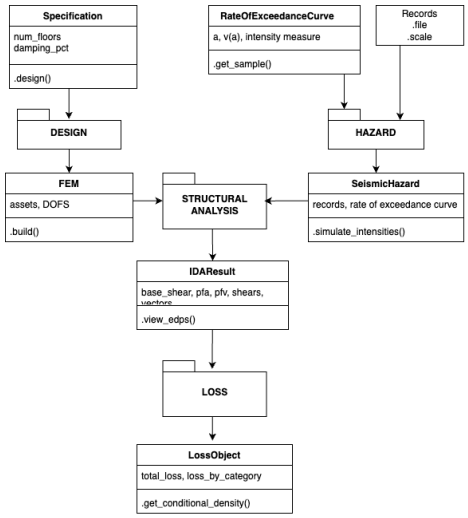



Figure 3.1: Flow chart for the program, data flows from top to bottom. Each module is specified by a folder icon and each object by a rectangle. The output is a loss object containing the conditional probability density of losses given an intensity measure, as well as a full breakdown of all the individual asset losses.

- The goal of the structural analysis module is to perform a fast time-domain, nonlinear structural analysis of many design alternatives under many strong-motion recordings rapidly, producing an exhaustive set of results for all degrees of freedom.
- The goal of the loss module is to evaluate the consequences of each design alternative both in terms of initial cost as well as in terms of future costs due to earthquakes. The program must therefore provide the designer with a means to rank design alternatives and choose the best

While this is a standard risk-evaluation procedure, the implementation presented allows for quick design, analysis, and risk computation of many design alternatives by automating and abstracting away the difficult and computationally expensive processes such as structural design, analysis, and asset loss estimation. This allows one to effectively explore the space of structural designs to eventually find global optima for a given risk measure, or at least better solutions than those associated with coded design. The conceptual framework aims to achieve a more rational, risk-informed, and hopefully optimal design of structures. We also envision that, as our society becomes more risk-averse and computational power cheapens, building codes will increasingly adopt a version of criteria based on monetary —or other types of— risk and initial building cost as ingredients of its measure of good performance.

## 3.1 Automated structural design module

This module must produce a finite element model capable of being implemented and executed by a nonlinear structural analysis software. Furthermore, it must create a complete description of the assets of a building, information that is required to perform a loss computation after the structural analysis results are obtained. A designed building is the realization of a process that starts with a specification. In turn, a design process is an algorithm that outputs the stiffness and strengths, no matter how complicated, of the structural elements, as well as the parameters required to implement a model into a structural analysis program. For example, one needs to specify masses and critical damping percentages to perform linear response history analyses, while member strengths are required to perform pushover analysis. The novel approach presented in this work is to implement this procedure incrementally and iteratively (see Fig.3.2), starting perhaps with a simple rule that assigns masses according to some codified procedure, and then assigns stiffnesses to achieve realistic natural vibration periods and shapes. The algorithm continues in this manner, refining the model until it achieves the properties we need in the final building, such as a desired level of strength and ductility. However, this description of the design process is incomplete because a finished building also contains assets. Therefore, we must broaden our definition to include the automatic generation of a complete description of its assets. This information is required for risk and loss estimation. The definition and location of the assets are obtained by establishing which occupancy class this building belongs to. For example, a hospital holds distinct assets compared to an office. It is also noted that given the same specification, two engineers will almost surely produce different solutions or "designs", either because they used different methods or criteria as to what is an acceptable result. Therefore, it is evident that a clear distinction between a specification and a designed instance exists and as such must be made in a computer program. This means that that a specification with different occupancy classes and design criteria will produce different building instances and therefore incur different losses for the same seismic events. Thus, a building specification is a high-level container of information needed to create a design instance. The authors have found that by reducing the specification to a given building archetype, a shorter and simplified list of parameters is obtained. This allows for a declarative way to talk about buildings of a given archetype.

## 3.2 Hazard module

Seismic hazard expresses the relationship between a measure of seismic excitation and its annual rate of exceedance at a specific location. This relationship is usually given by a hazard curve, of a chosen intensity measure, such as first-mode spectral pseudo-acceleration. Risk-based assessment needs to bind structural responses to rates of exceedance using one or more intensity measures such as spectral displacements or pseudo-accelerations. The goal of intensity-based

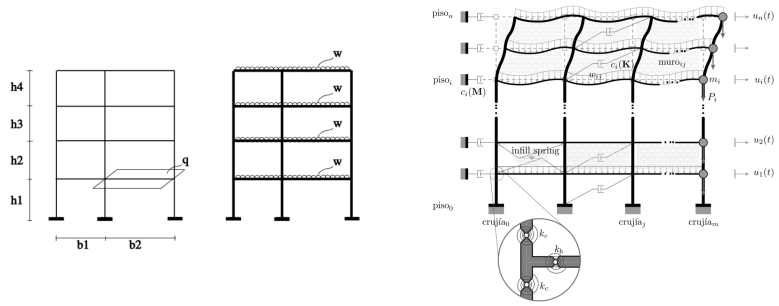



Figure 3.2: Iterative design process, from specification to codified mass assignment, to stiffness assignment, and finally to strength assignment and a refinement of the model ultimately expressed as a finite element program.

scaling methods is to provide scaling factors for records so that nonlinear analyses of the structure for these scaled records give an accurate estimate of the probability distribution of the engineering demand parameters (EDPs). We require that this procedure is also efficient, i.e., it minimizes the record-to-record variability in the structural response [Kalkan and Chopra, 2010]. The responsibility of this module is to generate an object that represents the seismic hazard and to store its representative ground motion records. The user can specify the rate of exceedance curve, using a fixed number of points or via an analytic expression such as a Pareto-type curve. This module is decoupled from the rest of the program, which allows the user to generate sets of "hazards" to be used in comparative or parametric studies. In general, the hazard module would contain an exceedance rate curve of the selected intensity measure plus a set of accelerograms equally likely representatives of the ground motion associated with a certain exceedance rate.

## 3.3 Structural analysis module

This module uses the records defined in the previous module to run Incremental Dynamic Analyses (IDA) [Vamvatsikos and Cornell, 2002] [Vamvatsikos, 2004] for a designed building, producing an exhaustive set of results that contain, among other things, the response for all degrees of freedom for any demand parameter such as reactions, accelerations, velocities, displacements, etc. The fundamental goal of an IDA is to capture the variability of the structural response against a set of records with distinct frequency contents and increasing intensity, to gain insight into the dynamical behavior of a structure. To do this, one employs a suite of selected ground motions. For each ground motion, the structure is subjected to a monotonically increasing intensity (in pseudo-acceleration), obtaining, for each record and each intensity, the structural response in terms of EDPs, e.g. peak inter-story drift. In this manner it is conjectured that the

response of the building across these intensity levels and with distinct frequency content could offer insight into its dynamical capacity and performance, starting from the elastic stage up until the onset of collapse.

The set of IDA curves plotted in a response-vs-intensity manner ($(\theta, S_a(5\%, T_1))$) is called an IDA-plot (see Fig.3.3). Using these results, one can statistically summarize the structural response as a conditional probability distribution of the EDP given an intensity. Moreover, one can estimate the probability of collapse given a level of intensity, which is a useful quantity in damage analysis and risk analysis in general. For example, one can compute the median intensity at which collapse caused by dynamic instability is achieved. In this procedure, collapse can be identified as a softening in each curve that can be identified visually as a flat line. This softening corresponds to numerical instability, and if the model satisfies some basic robustness properties, then this corresponds to dynamical instability [López, 2015].

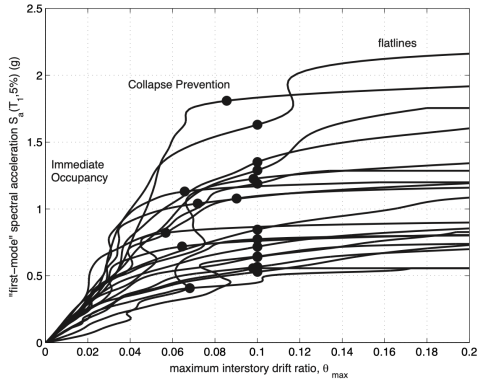



Figure 3.3: IDA curves for a 15-story steel building, black lines correspond to a record/intensity combination, black dots correspond to the onset of collapse e.g. collapse prevention damage state defined as 20% of the elastic stiffness [Vamvatsikos and Cornell, 2002].

## Loss Module

The responsibility of this module is to estimate the conditional probability of loss given an intensity, for all the assets of a building. For this purpose, it uses both the results from the IDA, alongside the hazard curve, and the specification of the assets generated by the design module. Assets can be classified into three categories; structural, non-structural, and contents. The fundamental difference between the non-structural elements and contents is that the former refers to fixed elements, while the latter is usually mobile and is typically inventoried (see Fig.3.4) In the literature, human health and business uptime are sometimes also considered assets. In general, they require separate treatment and will not

be considered as such nor discussed further herein. Henceforth, all assets are physical objects located somewhere in or on the outside of a building.


Figure 3.4: Examples of assets from the three different categories; a. structural elements (beams, columns, and walls), b. non-structural elements such as marble tiling, c. contents (plotters, printers, chairs, tables).

In this spirit, we say an asset is sensitive to a demand parameter such as acceleration, displacement (drift), force, or velocity when this is the primary cause of loss. For example, a flower vase (Fig 4.5 a.) is acceleration sensitive, while most lateral force-resisting structural elements (Fig 4.5 b.) are drift sensitive. Conversely, we say an asset is rugged (Fig 4.5 c.) when it is unaffected by any local action, and therefore the only cause for monetary loss would be total or partial building collapse. The terminology comes from a rug which is the representative element of this class. Other examples of this class would be sofas and clothing or shoes.

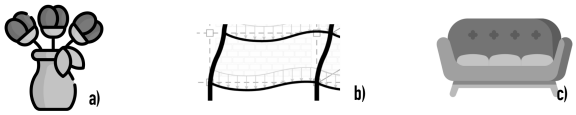



Figure 3.5: Example representation of; a. acceleration sensitive asset (flower vase), b. drift sensitive asset (beams and columns), c. rugged asset (sofa).

The main contribution of this module is to show how to compute the direct loss for structural elements based on physical principles. We propose a method for reinforced concrete beams and columns based on a modification of the energy dissipation capacity index proposed by Park and Ang [Park and Ang, 1985]. We postulate that any member possesses a reference inherent hysteretic energy dissipation capacity that is independent of loading history and that the piece can dissipate a certain amount of half cycles worth of energy before it collapses from degradation. Figure 3.6 shows an example of damage progression for a reinforced concrete column joint, from linear behavior to yielding, and finally to structural failure; observe that the piece failed before reaching its maximum ductile capacity. This shows that while cyclic degradation is important, a more sound approach is to consider a linear combination of cyclic capacity and monotonic ductile capacity [Kunnath et al., 1997]. This new index ranges from 0 to 1, and consists of two terms; the first term is related to the ratio of the maximum ductility demand to the ductile capacity achieved by the section during the loading cycles, while the second has to do with the hysteretic energy dissipated during the loading cycles.

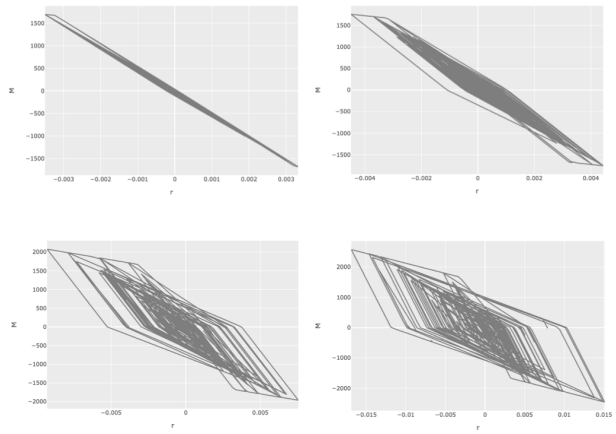


Figure 3.6: Damage progression in a reinforced concrete column joint (chord rotation vs moment), loss is computed directly from the physical hysteresis observed, that is, a combination of the energy dissipated during the loading cycles and the ductility demand.

Therefore, the repair costs estimated by this module are proportional to the damage index value computed for each structural element and subelement for a given run. The scenario of total building collapse is considered with the criterion expounded in the IDAs section, in which case all assets will incur in total loss.

The loss measure we are most interested in is the annual-rate-of exceedance-of-loss curve. This curve tells us how frequently a given level of loss is exceeded, and contains all the information to compute other loss measures, such as the average annual loss (AAL) or the expected loss. This curve is computed as follows; fix a level of loss, say $L$, then, for all records and all intensities, sum the corresponding frequency of events that caused losses greater or equal to $L$; this is the rate of exceedance of that level of loss. Repeat this procedure for all loss levels from 0 to the total building cost. We are also very interested in the AAL, since it is necessary to minimize the expected present value cost of future losses. This measure can be computed as the product of the expected loss for each event of the set and its corresponding annual frequency.

# Chapter 4

# Design module

## 4.1 Introduction

Design is one of the main activities of the structural engineer. To perform the commonly used force-based design specified in Building Codes, one must estimate the design forces for the model and members, but to carry out this analysis one needs a structural model. This is inherently a cyclical problem. Displacement-based design also faces this problem but in the opposite direction, starting from desired displacements. It is therefore a chicken-versus-egg situation of what came first, which has been sidestepped by postulating an initial design based mostly on experience and empirical success.

It is evident therefore, that the designer necessarily needs structural analysis. The latter is straightforward and algorithmic, as such, it proceeds by taking a given building with well-defined properties and solving the equations of motions to obtain its response under pre-defined actions.

However, the problem of design is much harder. Historically, simplified methods such equivalent structure methods [Shibata and Sozen, 1976] and direct-design [Priestley et al., 2007] attempt to avoid the aforementioned cyclical issue by reducing the complexity of the model or the analyses performed. Naturally, this yields designs that are (by definition) approximate and therefore will never be optimal. Are we satisfied with claiming they pass a checklist specified by a building code? If we seek optimality, we must rethink how we approach the design process and realize that we must necessarily include loss and risk analysis.

In this chapter, we show the concepts related to automated design procedures such as building specification versus instances, asset definition, asset classification, asset distribution, occupancy classes, asset collocation algorithms, and asset behavior. Then, using those concepts we show how to develop and implement automated structural design of buildings. Finally, we show how to procedurally generate computer models of buildings whose structural, non-structural, and content response resembles their true behavior.

## 4.2 What is a design process?

A designed building is the realization of a process that starts with a specification. A design process is an *algorithm* that outputs the stiffness and strengths, no matter how complicated, of the structural elements. This procedure must output the parameters required to implement a model into a program that can run advanced non-linear dynamic analysis. For example, one needs to specify masses and critical damping percentage to perform linear response history analyses, while member strengths are required to perform pushover analysis.

### 4.2.1 Building specification vs. designed instances

This description of the design process is incomplete because a finished building also contains assets. Therefore, we must broaden our definition to include the automatic generation of the complete description of its assets. This information is required for risk and loss estimation. The definition and location of the assets are primarily dictated by the building occupancy class. For example, a hospital holds different assets compared to an office. It is also noted that given the same specification, two engineers will almost surely produce different solutions or 'designs', either because they used distinct methods or criteria as to what is an acceptable result. It follows, that a specification with different occupancy classes or design criteria will produce distinct building instances and therefore incur distinct losses for the same seismic events.

Therefore, it is evident that a clear distinction between a specification and a designed instance exists and as such must be made in a computer program.

#### Building Specification

A building specification is a high-level container of information needed to produce a design instance. The author has found that by reducing the specification to a fixed building archetype, a shorter and simpler list of parameters is obtained. This allows for a declarative way to talk about buildings of a given archetype. This way one also avoids the trap of attempting to describe all possible building types at once.

In short, **Spec** is an interface that allows but is not limited to the following possible implementations:

1. Reinforced concrete special moment-resisting frames (RC SMRF)
2. Steel buckling-restrained braced frames (Steel BRBF)
3. Steel special concentrically-braced frames (Steel SCBF)
4. Special reinforced concrete shear walls (Special RCSW)

In this work, only RC SMRF was implemented, although once this is done the others are straightforward. This building type is essentially drift-controlled

to around 1-2% with a yield drift of around 0.5% and collapse capacity as low as 2g.

For this implementation, some of the parameters are:

1. storeys, specified as a list of heights in meters e.g. '3.5, 3.0, 3.0, 2.5'
2. bays, specified as a list of widths in meters e.g. '5.0, 7.0, 5.0'
3. percentage of viscous critical damping
4. masses per level in tons e.g. '50, 50, 50, 30'
5. number of identical frames perpendicular to the plane

**Designed instance or Finite Element Model (FEM)**

This object implements our concept of a finished building. It is the realization of a design process.

Ideally, a **FEM** must also contain sufficient information for the structural plans to be drawn and the construction to be executed. In practice, this class holds enormous amounts of data, therefore it is complex for a human to parse and impossible to specify manually. However, by automating the design procedure, we overcome the most difficulties associated with risk-based design.

This information includes but is not limited to

- nodes
- masses
- elements such as columns and beams (assets)
- geometric aspects such as length, height, width, etc.
- damping specification
- elastic periods and frequencies
- elastic eigenvectors
- nonstructural elements (assets)
- contents (assets)
- summary of net worth, broken down by concept
- 'OccupancyClass'

As can be seen, a **FEM** holds the information required to describe the assets that are not structural, we will defer talking about their implementation until the loss module, as it is more appropriate to understand loss computation.

Mathematically, a **FEM** is composed of nodes and elements, in our case those elements are beams and columns (and in other implementations possibly

slabs, walls, and even foundations). Finite elements are also assets, this means they must implement an asset interface. A critical point of any **FEM** implementation is that it must be linked to a structural analysis software. That is, at the very least, it must provide a way to generate valid models for a program to consume and a way to process the results. In our implementation, we have chosen OpenSees [McKenna et al., 2004] as our analysis program or 'kernel' in short. We show how to construct and produce 'input files' of our designed building and desired analyses. This is expounded on the structural analysis chapter.

### 4.2.2 Automated design procedure

When structural engineers design a building, they are developing a series of incrementally more realistic design instances. As such, they start with a specification of what needs to be built and usually begin their procedure with a very simplified model of reality *e.g.* a one-degree-of-freedom structure. This model is called a 'pre-design' and the procedure a 'pre-design' procedure. Afterwards, they perform a series of iterative and incremental updates to the model until it satisfies some criteria, usually what is specified in a Building Code, which are the minimum requirements acceptable.

Assuredly, we are primarily interested in the 'final' design instance and tend to ignore that there were a series of intermediate designs and an original 'pre-design' or 'seed' that allowed them to bootstrap the process.

Where does this 'seed' come from? The existence of this 'seed' is proved by looking and measuring the actual strengths and stiffness and dynamic properties of structures that are already built and have behaved well.

With these concepts in mind, a design algorithm is a mapping from the space of specifications to the space of finite element models.

$$\texttt{DesignProcedure: Spec} \rightarrow \texttt{FEM}$$

When an engineer executes this algorithm, they have a criterion that the result must satisfy. They checked sequentially at each 'design step' whether their model was satisfactory, and proceeded with the next step. Therefore a more accurate description of this procedure would be

$$\texttt{DesignProcedure: Spec} \mapsto \texttt{FEM\_0} \mapsto \texttt{FEM\_1} \ldots \mapsto \texttt{FEM\_n}$$

Each mapping constitutes a single design criterion. It is recommended that these steps are simple and deterministic, although it is not strictly necessary.

This approach is agnostic to the design philosophy employed. In this manner, force-based design, displacement-based design, and others can be implemented easily as pipelines *i.e.* as compositions of simple criteria. For example, one might imagine a criterion that makes the building comply with Code-specified *e.g.* [ASCE and SEI, 2006], and another, that complies with Code-specified stiffness and so on. One can imagine a criterion that implements a complex displacement-based design procedure as well [López, 2015]. In this manner, we can envision implementing design pipelines that can capture arbitrarily complexity and which can get computationally expensive.

The author believes that the optimal model can be obtained as the limiting result of this process. In other words, our measure can be approximated to any degree of accuracy as we let the composition of procedures grow and is exactly optimal when it tends to infinity.

It is interesting that, historically, as engineers realized that some analysis procedures were not algorithmic *e.g.* the flexibility method [Felippa, 2001], so too must we realize that in engineering practice we used procedures which were ultimately heuristics and as such impossible to implement in a computer. Invariably, we must turn away from such heuristics and start looking for design criteria that yield efficient and correct algorithms which produce optimal designs.

### 4.2.3 Design criteria

As discussed previously, a design procedure implements a sequence of design criteria that map a specification to a finite element model.

Therefore design criterion is a function from the sum type **Spec** | **FEM** to the type **FEM**. That is, given a specification **OR** an existing instance, we produce a new instance which is usually more complex than the previous one if present, as such

```
DesignCriterion: Spec | FEM → FEM
```

A criterion must always return a **FEM**.

Those criteria that take in a **Spec** are what we commonly call 'pre-design' criteria, hence, they tend to output simple models which then are sent as inputs to more sophisticated criteria. As an example of a simple criterion, consider the pre-design criterion that assigns loads and masses to the structure based on a Building code, let's assign the name 'CodeMassesPre' to such criterion.

#### CodeMassesPre

This class requires a notion of uniform load per unit area $q$. It also requires a notion of how much of the slab's mass goes to the beam at each bay. Suppose this criterion specified a uniform load of $q = 10$ kPa and that the area of the slab is exactly $A = 1 \cdot \ell^2$ where $\ell$ is the length of the beam. This is represented in Fig 4.1

The mapping $q \mapsto w$ produces uniform beam loads with the mass required by a chosen building code. Notice this simplified pre-design criterion does not produce inertias or member strengths, and as such must be used as input by other criteria.

A more elaborate example is a force-based design procedure whose solution is shown in the next section.

### 4.2.4 Automated force-based design

In this family of design procedures, the fundamental unknowns are the forces to be used for design, which are a function of the designed members themselves and are related by the building's dynamic response to seismic events.

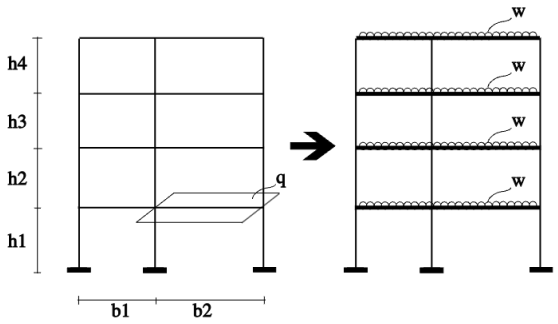



Figure 4.1: Mapping of $q$ in kPa to $w$ in kN/m, notice this mapping does not set member stiffnesses, only loads.

The pipeline to estimate the design forces consists of the following steps; to

1. Set code-compliant masses and loads.
2. Set some initial realistic stiffness and member dimensions.
3. Modify initial stiffness iteratively to achieve a fundamental period close to an empirical period.
4. Perform Response Spectruam Analysis (RSA) to determine the design axial forces, shears, and moments.

This process is summarized by the following mapping

```
ForceBasedDesign: CodeMassesPre ↦ CodeStiffnessPre ..
  .. ↦ ShearStiffnessIterative ↦ CodeRSA
```

We explore those criteria in detail.

**CodeStiffnessPre**

The objective of this class is to initialize the starting stiffnesses and member dimensions of the building. An example of a simple rule is to set the area of the cross-section such that the axial stress is a percentage of the concrete compressive strength *e.g.* $0.1f'_c$ for working gravity conditions.

Another rule might be to set the member dimensions such that the beams/-columns avoid being slender or comply with the minimum allowed dimensions specified by a given Code.

**ShearStiffnessIterative**

Suppose we wanted our building to have dynamic properties similar to real buildings.

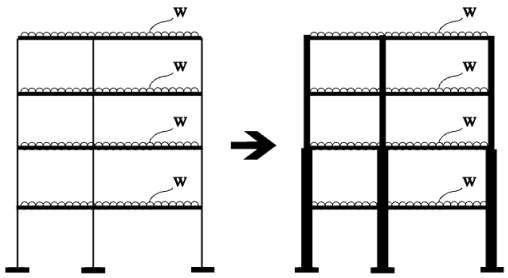



Figure 4.2: Mapping of a building with no stiffnesses to some member dimensions and second moments of area $I_{yy}$, notice this mapping requires loads/masses to run.

We, therefore, seek an empirical period formula, possibly based on the height or the number of storeys. This criterion attempts to make our building have a fundamental period similar to what has been measured for existing buildings. In this spirit, [Chopra and Goel, 2000] present a set of formulas that are a function of the height $H$ of the steel or concrete building, with their corresponding standard deviations $\pm 1\sigma$. On the other hand, a simpler period estimation formula in seconds such as $T_D = N_{\text{storeys}}/8$ suffices.

The objective of this criterion is to attempt to attain a fundamental period close to this empirical period.

The simplest FEM model that can give us an indication of the true dynamic behavior of a complex building is the lumped mass shear system which is approximately valid for buildings that are short and dominated by shear.

Suppose our initial model has fundamental period $T_1 \neq T_D$ and that there exists a factor $c$ such that if we multiplied our initial stiffnesses by $c$ it would yield a model with $T_1 \approx T_D$, then the algorithm is:

1. START with an initial set of inertias per storey $I = [I_1, I_2, ..., I_n]$.
2. then transform the current building temporarily into a lumped mass system using inertias $I_i$ at storey $i$.
3. obtain $T_1$ by solving the associated eigensystem.
4. if $T_1 << T_D$, this means our building is too stiff, then scale by selecting $c \in (0, 1)$, go to step 3
5. if $T_1 >> T_D$, this means our building is too flexible, then scale $c \in (1, \infty)$, go to step 3
6. if $T_1 \approx T_D$, transform the model back in to a MDOF building using inertias $cI$, END.

It is important to mention that this procedure also should 'chunk' the stiffness of our building in blocks so that the cross-sections do not change every story, rather, they change every other 2, 3, 4, or maybe even every $\lfloor\sqrt{N_{\text{storeys}}}\rfloor$ storeys. Such chunking procedures resemble more realistically how buildings are designed and constructed in structural engineering firms.

### CodeRSA

When the fundamental period of our pre-designed building is somewhat realistic, we need to estimate the design forces acting upon it. There is an ingenious way to estimate the mean value of the peak response to an ensemble of earthquake excitations without actually performing the response history analyses (RHA). This widely-known technique is called response spectrum analysis (RSA). The goal of this class is to automate a Code-based RSA to compute the design forces acting upon our building.

Suppose we can perform basic structural analyses on our models such as gravity, eigensystem analysis, and pushover. The details of the implementations are postponed chapter 6. Then, let $\Omega^2, \Phi$ be the eigenvalue and eigenvector (modal) matrices, where each column $\phi_n$ of $\Phi$ is the $n$-th natural vibration mode and $\Omega^2$ is diagonal, ordered from lowest natural frequency $\omega$ (rad/s) to highest.

Suppose that the modal matrix is unitarily mass normalized i.e. $\Phi^T M \Phi = I_n$ where $I_n$ is the identity matrix of order $n$. This is important for the procedure to give dimensionally consistent results [Clough and Penzien, 1993].

Let $M$, $\Gamma$ be diagonal matrices, where $M$ is the mass matrix and

$$\Gamma = \begin{pmatrix} \Gamma_1 & 0 & 0 & 0 \\ 0 & \Gamma_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \Gamma_n \end{pmatrix} = I_n(\Phi^T M 1)$$

Then, the 'mass shape' matrix that corresponds to the modal expansion of effective inertial (earthquake) forces is

$$S = \Phi \Gamma M$$

where each column $s_n$ has the mass contributions at each story for each mode. Since both $\Phi$ and $\Gamma$ are unitless, this matrix has units of mass and is called the *effective mass matrix*.

Let $Q, Q', R, \gamma_{\max}$ be the Building Code's seismic behavior factor, spectral ordinate reduction factor, over-strength factor, and the maximum allowed drift which depend upon the structure's fundamental period, building materials, location, among others.

Furthermore, let $A_n$ be the diagonal matrix whose $n$-th entry is the design spectral pseudo-acceleration corresponding to a single-degree-of-freedom system with elastic period equal to period $T_n$ of the structure. Where $T_1$ is the (longest)

fundamental period of the structure. The matrix of effective static earthquake forces $F_{ij}$ at storey $i$ for mode $j$ is simply

$$F_{ij} = \frac{1}{Q'R} SA_n$$

Using this matrix, the vector of lateral design forces per storey is computed using a modal combination rule. We recommend the combination rule *SRSS* as it is simple and works well in this scenario since the natural frequencies of frames are usually not closely spaced [Newmark and Rosenblueth, 1971]. In this procedure, each entry in the matrix is squared, then the columns are summed together into a column vector, and finally, the square root of each resulting entry is taken.

$$F_D = \sqrt{\sum_{n=1}^{N} F_n^2} \tag{4.1}$$

Note that peak values computed by this procedure underestimate the true response for RC-SMRF buildings by around 15-25% [Chopra, 2012]. A realistic design procedure must consider this. The reader is referred to [Cruz and Miranda, 2017] for more details.

Afterward, lateral design forces $F_D$ obtained using Eq 4.1 are used to perform an elastic static analysis directly on the proposed FEM to get the final member design forces $(P, V, M)$.

Finally, a lateral deformation check must be performed

$$\gamma_0 Q \leq \gamma_{\max}$$

Where $\gamma_0$ is the maximum absolute drift of any storey of the structure under the static design forces. When those checks are passed, we end the procedure. The complete pipeline is shown in Fig 4.3.

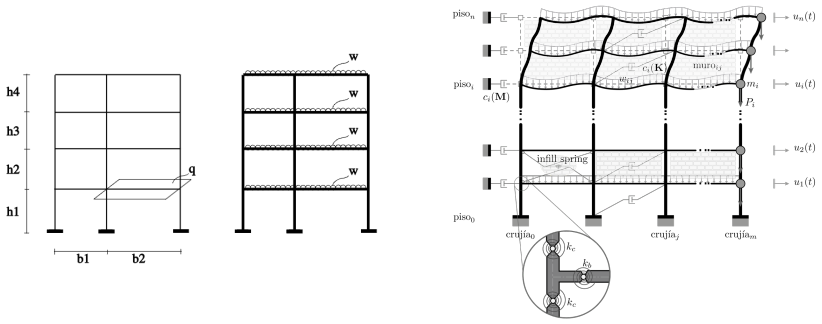
## Reinforced concrete element selection and design

For any reinforced concrete member and $(P, V, M)$ obtained in the previous section, an element *design procedure* selects the longitudinal and stirrup steel ratios and geometry which will produce the desired axial, shear and flexure strengths $(P_y, V_y, M_y)$. In other words, members much be given enough strength to resist the design forces, plus enough detailing such that it will develop full flexure ductility before failing brittlely in shear. Building codes also dictate that the ratio of resistances at the joints must follow a strong-column weak-beam criterion

$$M_c/M_b > \kappa \qquad \kappa \approx 1.5$$

The design procedure must take this into account.

Once a member has been *designed*, it is equivalent to the member being physically built, as such, all geometrical and material properties are therefore




(a) Initial stages of design (pre-design)

(b) Final stage of the design procedure (FEM model)

Figure 4.3: Complete design pipeline, from specification to a model with masses and stiffnesses, and finally to a very detailed FEM model capable of being implemented in a computer program.

fixed. Physical properties of interest include rotational capacities and ductility such as $\theta_y, \theta_u$, member strength hardening ratio, and energy dissipation capacity $\Lambda$, among others. These inform the analyst of the complexity of the model, the quality of the member's detailing, and allow for more realistic structural analyses to be implemented.

## 4.3 What is an asset?

As discussed on Sec.4.2.1, a *designed instance* must contain an exhaustive description of the building's assets. An asset is part of a building that holds value to its stakeholders, in other words, an asset is something that can cause monetary loss. Stakeholders will suffer this monetary loss when they are forced to choose to either repair, replace, or discard an asset. This will occur sometime after they have made this decision. This implies that loss does not happen instantly. However, for simplicity of this investigation, the loss will be considered to occur instantly after a seismic event. Along the same lines, we recognize that after a building is furbished, the contents, non-structural elements, and human occupancy change with time, however, we will initially treat those as constant due to simplicity.

The initial state of the building's assets is defined when the building is constructed and furbished and depends primarily on the occupancy that the building has *e.g.* a hospital will hold significantly different assets than a school even if they have the same architectural specification.

The problem of specifying assets for a given occupancy class together with an arbitrary specification is solved to a first-order approximation by what we call in this research a *collocation algorithm* and will be expounded in detail in Sec 4.5.2. For a design procedure to be considered complete, it must also

generate an exhaustive description of the assets instances which include their cost, location, classification, categorization, behavior, and response to actions, among others.

## 4.4 Asset categorization and classification

Historically, there are three mutually exclusive categories an asset can fall into:

1. Structural
2. Non-structural
3. Contents

The fundamental difference between the non-structural elements and contents is that the former refers to fixed elements, while the latter is usually mobile and is typically inventoried.

In the literature, human health and business uptime are sometimes considered assets. In general, they require separate treatment and will not be discussed further herein. Henceforth, all assets are physical objects located somewhere in or on the outside of a building.

Subcategories for structural elements include:

- columns
- beams
- non-partition walls and membrane-like elements
- floor slabs
- foundation and sub-structure elements
- braces and
- dampers
- isolators

Subcategories for non-structural elements include:

- electrical and mechanical services
- fire services
- lifts and escalators
- sanitary plumbing and drainage
- roofing and exterior fabrics

- windows
- internal finishing (floors, partition walls, ceilings)
- fitting and fixtures

Subcategories of contents include:

- doors
- furniture (chairs, tables, cabinets, etc.)
- office equipment (computers, printers, plotters, etc.)
- kitchen equipment (food, glassware, fridges, etc.)

These are visualized in Fig 4.4. For a more complete and exhaustive breakdown, the reader is referred to [Taghavi et al., 2003].

Conceptually, assets are considered structural when they need to be considered when performing structural analyses, this means that they either contribute directly in stiffness or strength, whereas otherwise, they would only contribute in mass and weight (and therefore indirectly to damping).

A study of buildings in New Zealand found that the ratio of non-structural & contents to the structural elements cost is around 2:1 [Dhakal and Aninthaneni, 2021]. Taghavi and Miranda [Taghavi et al., 2003] found that there is a high variability of cost distributions, with offices holding the largest portion of structural costs (18%), while hospitals have the lowest (8%). While for contents, hospitals tend to have most of the asset cost concentrated in their contents. Structural engineers and stakeholders must recognize that optimizing the cost of structural elements during the design phase is biased and should move towards a more holistic treatment of assets.

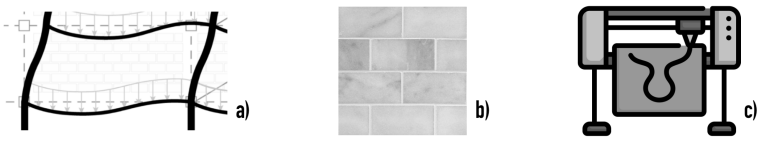



Figure 4.4: Examples of assets from the three different categories a) structural elements (beams and columns), b) non-structural element (such as marble tilings), c) contents (plotter).

Another useful classification pertains to the way that an asset responds to actions. This means paying attention to where loss can come from during events. In this spirit, we say an asset is *sensitive* to a demand parameter such as acceleration, displacement (drift), force, or velocity when this is the primary cause of loss. For example, a flower vase (Fig 4.5a.) is acceleration sensitive, while most lateral force resisting structural elements (Fig 4.5b.) are drift sensitive.

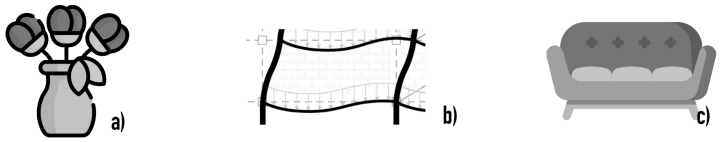



Figure 4.5: Example representation of; a) acceleration sensitive asset (vase), b) drift sensitive assets (beams and columns), c) rugged asset (sofa).

An asset might be sensitive to multiple parameters simultaneously, although this is not considered herein for simplicity.

Conversely, we say an asset is *rugged* (Fig 4.5c.) when it is unaffected by any local action, and therefore the only cause for monetary loss would be total or partial building collapse. The terminology comes from a rug which is the representative element of this class. Other examples of this class would be sofas and clothing or shoes.

For completeness, we present another interesting classification pattern, the so-called 'partition-like' assets [Ramirez and Miranda, 2009] whose loss is conditionally dependent upon the damage state of the partition and the components therein. When a partition component needs to be replaced due to damage, then the entirety of the partition will be replaced. This means there is a spatial interaction between elements of the partition. Examples of such assets would be windows plus framing, or ceilings that are coupled to electrical wiring and plumbing. One of the difficulties with this approach is that the joint-probability distributions of their fragilities are extremely hard to obtain. There is also the possibility that assets can be damaged by fire, wind, or water due to floods or anti-fire sprinklers. These events are ignored for the present study.

## 4.5 Asset distribution

Distribution refers to where the assets are located within or outside of the building. This is important for loss estimation. In general, structural elements are largely invariant during the lifetime of the building, while contents tend to vary the most. The variation of assets during the lifetime of the building is not considered here.

In Sec 4.2.1 we discuss the case where two identical buildings with distinct occupancies produced in general very different asset distributions *e.g.* a hospital versus a school. Therefore it is paramount that we capture this notion with the concept of an 'occupancy class'.

### 4.5.1 Occupancy classes

The primary usage of a building determines its occupancy class. The use of classifying a building of a given occupancy class is to be able to determine

approximately the non-structural and content assets in a building without recurring to pre-defined inventory sheets [FEMA, 2012a]. For our purposes, the occupancy class must also dictate the distribution and location of the assets.

For framed buildings, the most important classes of occupancy include

- residential (apartments)
- hospitals
- commercial offices
- hotels
- retail
- schools
- warehouses
- museums
- industrial installations such as laboratories

Although occupancy classes have a predominant role in the expected loss, their influence has not been fully explored yet. In part, because we lack the tooling to simulate them adequately. These classes determine the quantity and specific non-structural assets and contents. As a useful simplification, a building must have one and only one occupancy class. They must also dictate the probability distribution for the number and location of occupants at any instant. This is called the 'population model', however, since we do not deal with human losses or business uptime, this is not considered in the present study.

The problem of determining asset distribution has usually been treated somewhat heuristically, this is because of the many uncertainties pertaining to asset distribution, among them; the lack of curated empirical inventory sheets, coupled with the high variability across multiple buildings of the same occupancy class alongside contractor variability [Ramirez, 2009].

Reynoso and Jaimes [Reinoso and Jaimes, 2013] present a set of real inventory sheets of contents for different occupancy classes. In particular, they show the schematic distribution of assets within public offices (Fig 4.6). One notices the similarities across buildings and floors of the same building. For example, the plotters are present on almost every floor while the distribution of tables and chairs is very similar. These observations hint towards a conceptual understanding of occupancy classes in terms of pre-defined blocks of usage.

An office might be defined as a space with the following blocks

1. generic bathrooms
2. business meeting room
3. lobby

4. cubicles

Each block has a well-defined set of assets required to fulfill its intended functionality *e.g.* a business meeting room must have at least 1 table with around 6 to 12 chairs on average, plus the carpet and ceiling finishes, projector, etc.

We posit that this characterization of an office building in terms of blocks of single functionality captures what we intuitively mean by the assets of this occupancy. Conversely, when we speak of an office building we imply the existence of this block configuration.

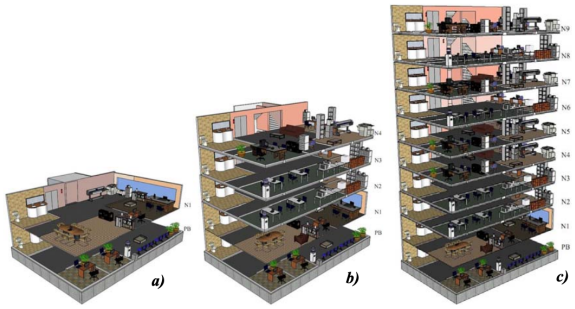



Figure 4.6: Office buildings of a) two b) five c) ten floors. Notice the similarities between the distribution of assets within and across buildings e.g. chairs, tables, plotters, bathrooms, etc. (taken from [Reinoso and Jaimes, 2013])

On a similar note, [FEMA, 2018] presents a conceptualization of a hospital in terms of 'blocks', see Fig 4.7

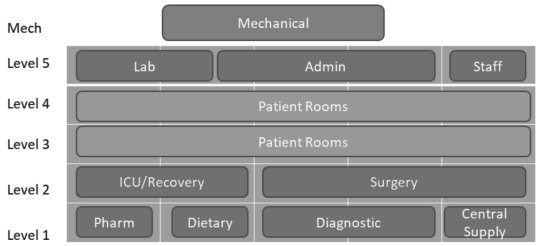



Figure 4.7: Schematic representation of a mid-rise hospital as a series of blocks *i.e.* a department-block-diagram, each block has a well-understood functionality. (taken from [FEMA, 2018])

It is implied that those blocks are similar across all conceivable mid-rise hospitals (although one must be careful to distinguish between acute and non-acute care occupancies). The description of occupancy and the distribution of

assets within the building is easily explainable in terms of those department-blocks.

This motivated the author to introduce the notion of a *collocation phase* of assets in terms of department-block specifications.

### 4.5.2 Collocation algorithm

A collocation algorithm is a mapping from a department-block specification and a FEM instance to a set of assets.

```
CollocationAlgorithm: (BlockSpec, FEM) ↦ Assets
```

This idea implies that asset distribution is somewhat independent of the building's storeys, bays, and dimensions, this means that for example, most offices are very similar to each other in their layout, independent of the building specification.

A collocation algorithm simulates placing those assets in specified locations to resemble architectural averages *i.e.* how many ceiling/electrical wires/plumbing/furniture/etc are usually contained in a unit area for a given occupancy. We stress that this procedure be fully deterministic. Since a FEM can have any number of storeys or bays of any height or length, it may be the case that a simplistic collocation algorithm will produce unrealistic distribution of assets.

As an example of a diagram-block specification for an office is:

```
Reception:
  units:
  - waiting_room
  - printing_station
  - cubicle
OfficeFloorMedium:
  units:
    - small_kitchen
    - printing_station
    - two_cubicles
    - meeting_room
    - bathroom
```

A 'block' such as a Reception contains a set of units. A unit contains a set of assets with a specified area, as an example we specify a printing station unit as:

```
- unit_name: printing_station
  dx: 2.0
  assets:
    - Minicomponent
    - 60inPlotter
    - CopyMachine
    - FaxMachine
```

```
- FaxMachine
- MediumShelf
- WaterDispenser
- Multiprinter
- SmallPrinter
- DigitalCamera
- DigitalCamera
- NonStructuralGenericElectric
- NonStructuralGenericRoofTilings
- NonStructuralGenericWallFinishes
- SlimJar
- MediumTable
- MediumTable
```

The collocation algorithm we suggest attempts to fit as many units as possible within a floor. For a public office with the aforementioned diagram-block specification, the algorithm produces the set of assets shown in Fig 4.8.

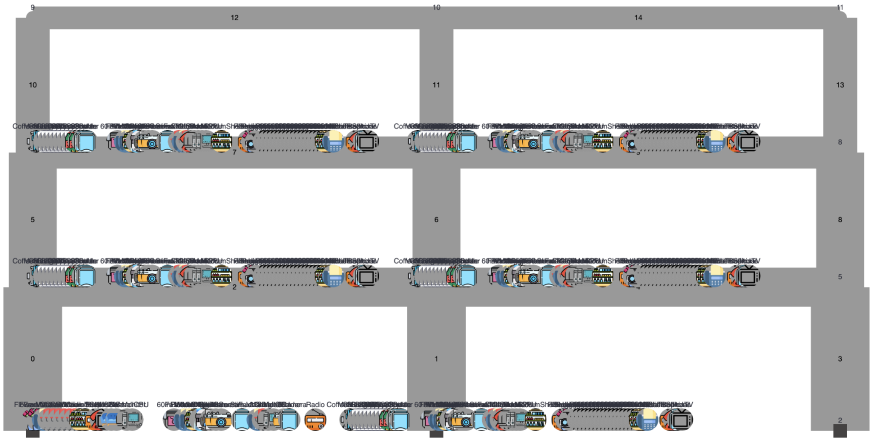


Figure 4.8: Assets produced for a frame with a Private Office occupancy specified by the diagram-block specification previously shown. Every circle represents an asset. The first floor is a Reception block while the upper levels are of block-type OfficeFloorMedium.

## 4.6 Responsibility and objective of the module

This module must produce a finite element model capable of being executed by nonlinear structural analysis software. Furthermore, it must produce a complete description of the assets of a building, information which is required to perform a loss computation after the structural analysis results are obtained. Henceforth

everything is in SI system of units for engineering; forces in kilonewton (kN), distances in meters (m) and time in seconds (s).

We show a summary of what each class does and explain at a high level the responsibility of each class and what it should accomplish. The source code of the initial release is public.

### 4.6.1 UML Diagrams

In the PlantUML notation, software classes are specified by rectangles with their name at the top. Their properties (called attributes) are shown below their name, while their most relevant methods (own functions) are listed at the bottom third of the rectangle.

Relationships between classes are given as lines, the nature of this relationship is specified by the arrowhead type:

1. △ represents implementations of interfaces.
2. ▲ represents a relationship of composition or being contained by the source class.

Fig 4.9 shows the main objects in the module. Most properties are not primitive but rather custom types, however, their implementations are omitted for brevity. The reader is invited to consult the public source code for the complete implementation.

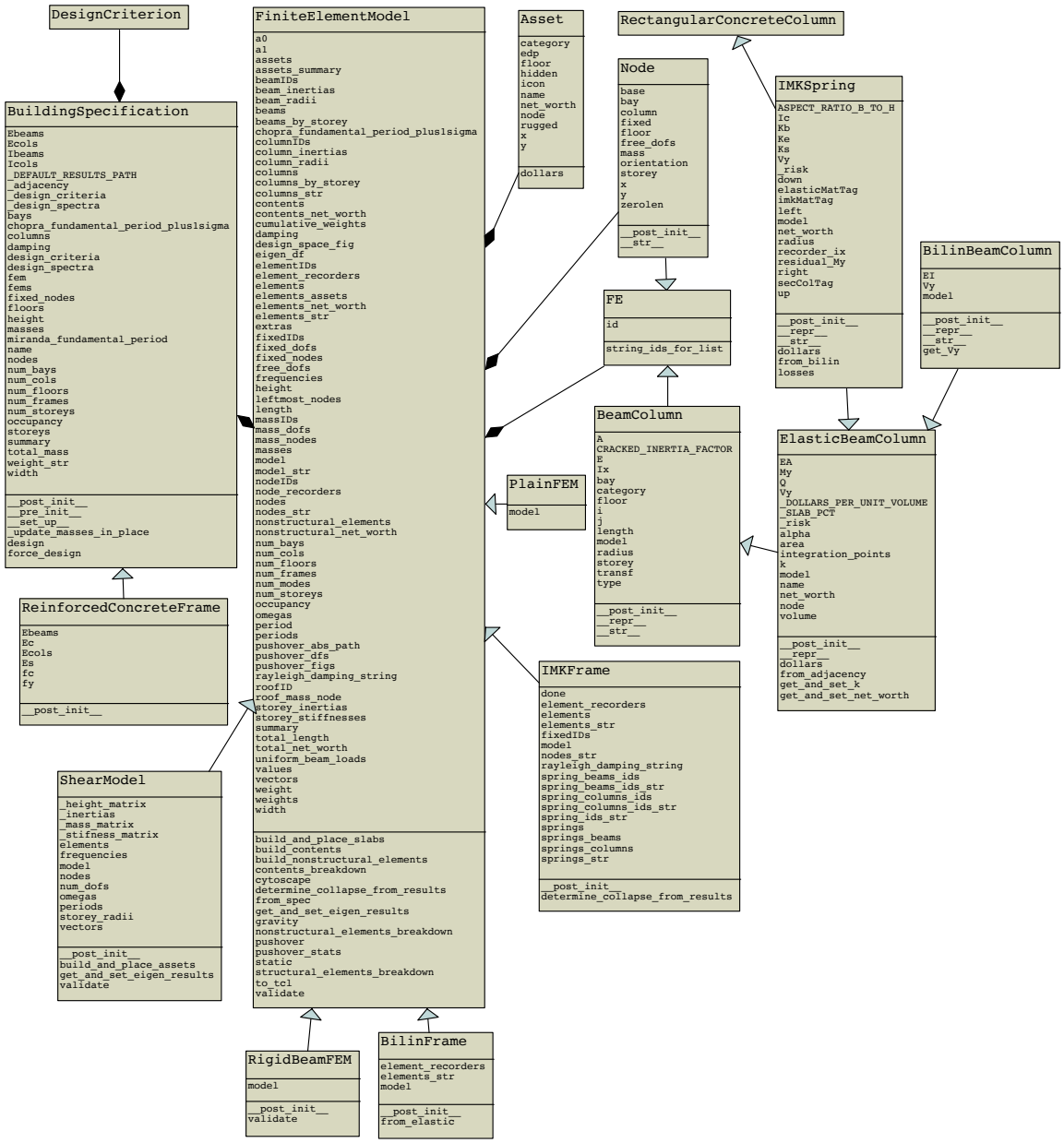



Figure 4.9: UML diagrams for the design module. Each row in a box represents an attribute of the class, notice 'BuildingSpecification' and 'FiniteElementModel' are interfaces which many classes implement (as seen via all $\triangle$ that point towards it.)

# Chapter 5

# Hazard module

## 5.1 Introduction

Seismic hazard expresses the relationship between a measure of seismic excitation and its annual rate of exceedance at a specific location. This relationship is usually given by a hazard curve $\nu(a)$ (Fig.5.1) of a chosen intensity measure (IM). Risk-based assessment needs to bind structural responses to rates-of-exceedance using one or more intensity measures such as spectral displacements or pseudo-accelerations.

There are two main methods for this; intensity-based scaling and spectral matching techniques. The former method modifies the amplitude of the record while preserving the original non-stationary content while the latter modifies the frequency content to match its response spectrum to the target spectrum.

The goal of intensity-based scaling methods is to provide scaling factors for records so that nonlinear analyses of the structure for these scaled records give an accurate estimate of the median value of the demand parameters. We require that this procedure is also *efficient*, i.e., it minimizes the record-to-record variability in the structural response [Kalkan and Chopra, 2010] [Eads, 2013] [Sextos, 2014].

The following measures are shown to exhibit both efficient and accurate properties [Aslani and Miranda, 2005]

1. Elastic displacement spectral ordinate
2. Normalized peak ground acceleration by the factor $4\pi^2/T_1^2$ , where the denominator is the first period of vibration of the MDOF model.
3. Inelastic spectral displacement computed using a single-degree-of-freedom with a predefined yield displacement.
4. A combination of spectral displacements evaluated at two vibration periods equal to the geometric mean of the elastic spectral displacement

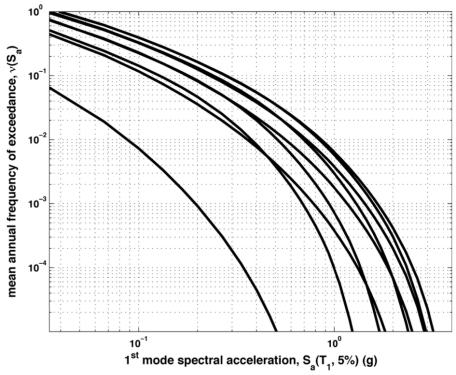



Figure 5.1: Mean annual rate of exceedance curve for multiple first-period spectral pseudo-accelerations at 5% viscous damping, (taken from [Fragiadakis and Papadrakakis, 2008])

evaluated at the fundamental period and twice the fundamental period respectively.

The state of structural engineering practice, scales recorded natural acceleration records to match the first-mode spectral pseudoacceleration as IM [Fragiadakis and Papadrakakis, 2008]. Then, they are applied directly as excitation to a numerical model of the structure. Recent studies have validated the performance of this approach in estimating different median demand parameters for buildings and comparing intensity measures from simulation with empirical ground motion prediction equation models [Haselton et al., 2012].

Many ground motions (and associated response history analyses) are required to estimate full distributions of response for risk-based assessments. Past studies have used 40 ground motions per intensity level, at 10 intensity levels [Haselton et al., 2012]. However, with recent increased access to cheap computational power, this can increase by an order of magnitude or two. Unfortunately, this approach has several limitations including the scarce availability of naturally recorded ground motions for the specific earthquake characteristics of interest (magnitude, distance, type of faulting, site conditions, etc.), the doubtful meaningfulness of the scaling procedure [Grigoriu, 2011] and the questionable choice of the intensity measure used as a reference for scaling. Ground motion selection is a very active research topic [Marques et al., ], however, for our purposes, we need a pragmatic procedure that produces acceptable results.

### 5.1.1 Responsibility and objective of this module

The responsibility of this module is to hold a representation of the seismic hazard via a mean annual rate of exceedance curve $\nu(a)$, where the intensity $a$ is the first mode pseudo-acceleration at 5% viscous damping.

This module also holds a set of seismic records that are consistent with the hazard, however, the consistency of the records is not validated and is left to the user. This module must also produce the correct scaling factors for the provided records to bind $a$ to a structural response and to ultimately be able to compute losses using $\nu(a)$.

Therefore the objective of this module is to produce an object representing the intended hazard, this means both the hazard curve as well as its associated records. This technique allows the user to generate multiple 'hazard scenarios' to perform comparative or parametric studies.

#### UML diagrams and code structure

In the PlantUML notation, classes are specified by rectangles with their name at the top. Their properties (called attributes) are shown below their name, while their most relevant methods (own functions) are listed at the bottom third of the rectangle.

Relationships between classes are given as lines, the nature of this relationship is specified by the arrowhead type:

1. $\triangle$ represents implementations of interfaces.
2. $\blacktriangle$ represents a relationship of composition or being contained by the source class.

Fig 5.2 shows the main objects in the module. The reader is invited to consult the public source code for the complete implementation.

### 5.1.2 Hazard container class

**Hazard** is the main class of the module. It is a container for both **Record** and the **HazardCurve**. It provides methods to add and delete records and wrappers to obtain the necessary scaled intensity for IDAs.

### 5.1.3 HazardCurve interface class

Implementations of this interface are responsible for representing the actual hazard function $\nu(a)$ of the rate of exceedance for an intensity measure. They can also provide implementations for

- generating a uniform set of samples of $a$
- interpolating sets of intensities
- generating bins of intensities to perform IDAs

The program provides two sample implementations; the first, a **ParetoCurve** specifies the rate of exceedance as:

$$\nu(a) = \nu_0 \left(\frac{a_0}{a}\right)^r$$

where the parameters $a_0, \nu_0, r$ are the minimum cutoff intensity, the minimum cutoff rate of exceedance for $a_0$, and a shape factor, respectively.

The second implementation, **UserDefinedCurve** (Fig 5.3), allows the user to specify $(x = a, y = \nu(a))$ directly by providing a discrete set of points to be interpreted as the intensity and mean annual rate of exceeding such intensity.

### 5.1.4 Record class

This class is responsible for holding a pointer to the time-history input file corresponding to an accelerogram. It has a set of attributes that describe the properties of the time-history input file such as duration, scale, steps, etc.

Its **RecordKind** denotes the physical units of the time-history input file i.e. acceleration, displacement, velocity or Fourier, etc, and is used only as type enforcement and validation. It indirectly references the multiple **Spectra** that it generates for a fixed critical damping level.

This class provides methods such as computing the scaling factor needed for the record and intensity and validating that the input file is physically consistent and can be used in RHA.

### 5.1.5 Spectra class

This class represents the responses of a set of single-degree-of-freedom systems with given periods (usually from 0.1 to 5s in increments of 0.1s) and a critical damping ratio (usually 5%) to a given record. It holds references to the files on disk that contain the previously computed spectral ordinates, as well as references to frequently requested values such as the maximum absolute value, minimum value, and maximum value of the response, as well as the time where it occurred. These responses are the following: displacement, pseudo-velocity, pseudo-acceleration, absolute velocity, or absolute acceleration. This class provides wrappers to get the ordinate value for a given period or set of periods.

An example of a (Record, $S_a$ Spectra) pair produced by the program is shown in Fig 5.4

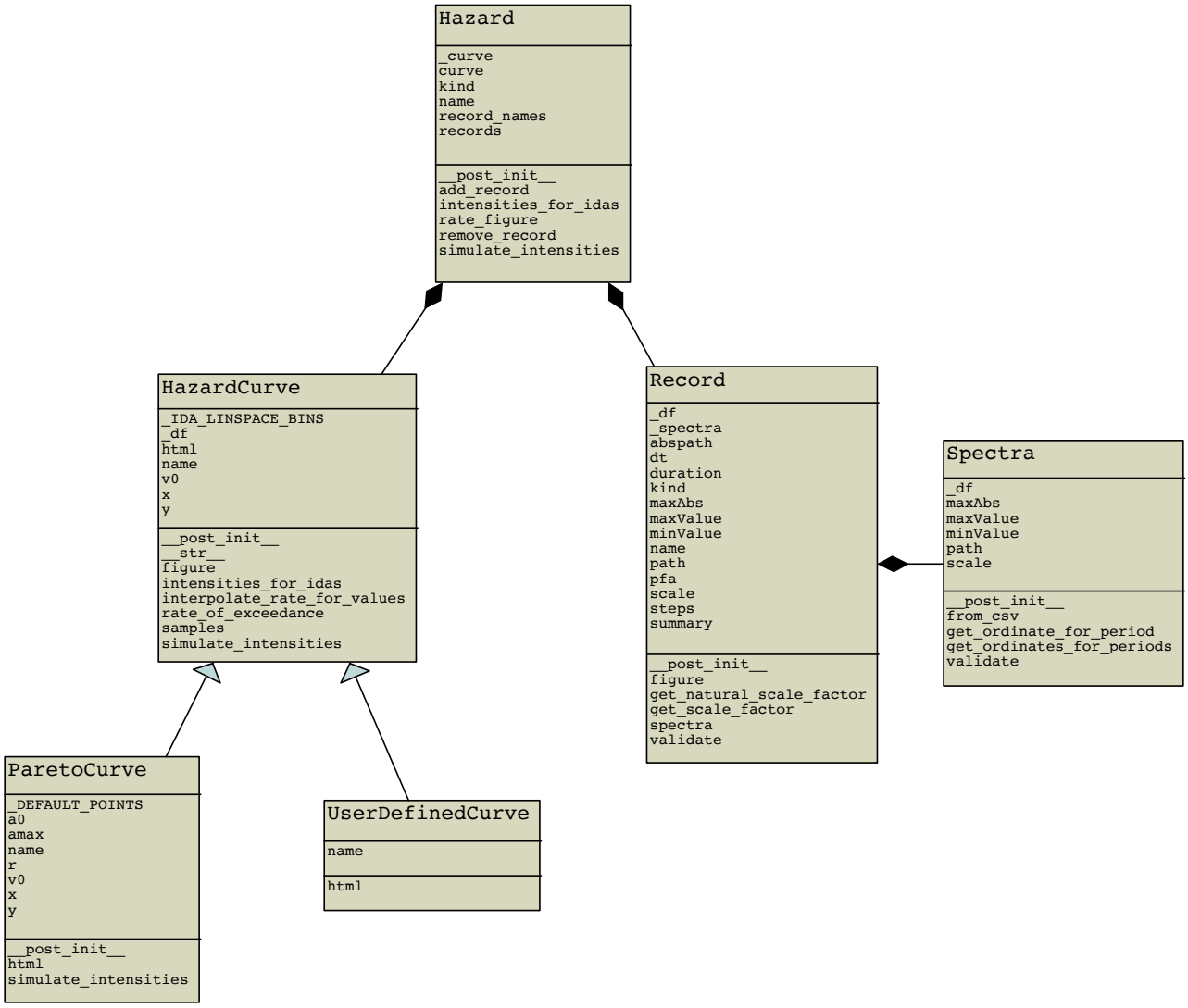



Figure 5.2: UML diagrams for the hazard module. Each row in a box represents an attribute of the class, notice 'HazardCurve' is an interface implemented by 'ParetoCurve' and 'UserDefinedCurve'.

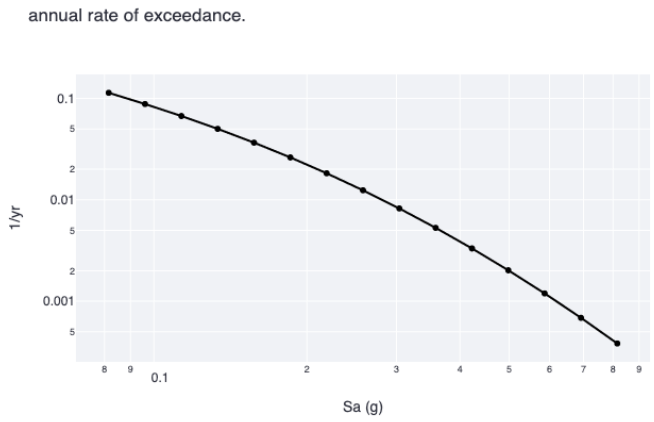



Figure 5.3: User-defined mean annual rate of exceedance curve consistent with the records for the Mexico City lakebed soil zone.

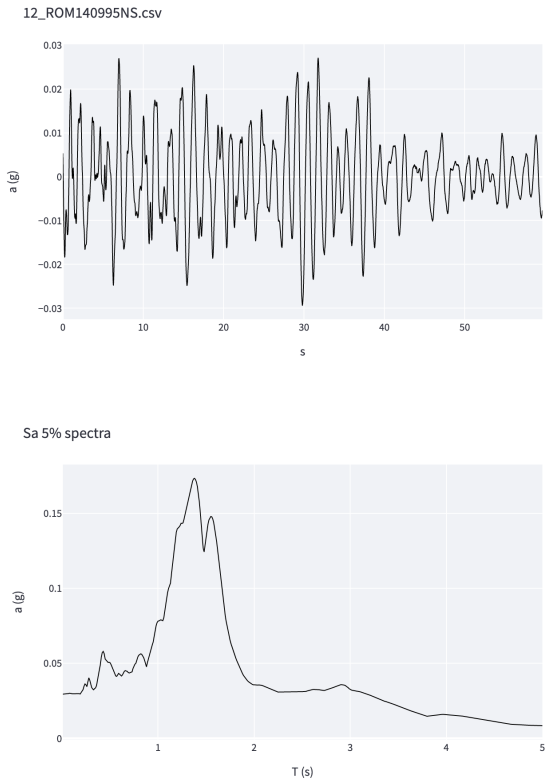



Figure 5.4: Time-history record and $S_a$ spectra for record 14-ROM14099NS captured in the Mexico City lakebed soil zone.

# Chapter 6

# Structural Analysis module

## 6.1 Introduction

The main objective of structural analyses is to obtain the instantaneous displacements and forces of a structure subjected to external loading, and consequently its internal stresses and strains. There are a variety of response parameters of potential interest to an analyst *e.g.* peak transient drift in each story, floor accelerations, component inelastic deformation, element forces, dissipated energy, etc. Modern structural analysis programs allow the analyst to capture the complete time-history response for all degrees of freedom for displacements, velocities, accelerations, and forces as well as element internal stresses and strains.

Despite this, linear elastic analysis of buildings are still useful to engineers because they provide insight and are easy to perform, however, real structural behavior of buildings is nonlinear mostly due to material damage.

Geometric nonlinearity affects structural engineers as well, most notably due to P-Delta effects which can be accommodated in the analyses via the co-rotational approach or the tangent stiffness matrix approach [López, 2015] [Argyris et al., 1979].

However, material damage is considerably more difficult to model realistically because the uncertainties in the material behavior during high-intensity events are so large that the best we can hope for is to capture the main aspects of building behavior [Powell, 2010]. Therefore, the fundamental objective of structural analyses could be rephrased as:

> "To produce results that capture the essential behavior of a real structure."

This implies that it is erroneous to think that a more sophisticated model always implies more accurate results. We must not forget that we are dealing with a very crude representation of reality, therefore the analyst must strive to reduce the complexity of the model to a minimum.

The goal of this chapter is to show how to attempt to implement a real framed building in a computer. First, we show how to move from from simple

linear elastic analysis to multiple nonlinear time-history analyses correctly, so that we are certain our results are correct. Then we discuss the most important concepts in modeling such as structural element model selection, damping, material damage and treatment of collapse. Afterward, we show the abstract classes needed to implement them in computer code. Finally we show a sample OpenSes input code for a static pushover and dynamic analysis of a 3 storey reinforced concrete building.

## 6.2 Implementation of the equations of motion

Our goal is to understand how buildings behave under seismic action, and so we ultimately need to perform nonlinear dynamic analyses. Hence, the exact dynamical response $u(t)$ of a structure to an acceleration record with a finite number of degrees of freedom is given by

$$M\ddot{u}(t) + C(u)\dot{u}(t) + K(u)u(t) = -M\ddot{u}_g(t) \tag{6.1}$$

Where $M, C(u),\ K(u)$ are the mass, damping, and stiffness matrix, and $\ddot{u}_g$ is the base free-field acceleration.

In general, Eq.6.1 must be solved incrementally at discrete instants $t_1, \ldots, t_n$ and iteratively because $C(u),\ K(u)$ are nonlinear due to structural damage primarily [Zienkiewicz and Taylor, 2013]. Integrating Eq.6.1 is done via Finite Element Method (FEM) code-bases or kernels, which implement assembler and solver algorithms and strategies. In this work, we use OpenSees as our FEM kernel.

### 6.2.1 FEM kernels

The main idea behind an analysis kernel such as OpenSees [McKenna et al., 2004] is to abstract away the numerical implementation of the pre-processors, assemblers, and solvers for the Finite Element Method and provide interfaces for creating a wide variety of models easily. As such, OpenSees is the main kernel used by researchers in earthquake engineering and is still active with contributors, new models, and algorithms being added to this day.

OpenSees contains a variety of solvers and strategies for integrating eq 6.1, and is not opinionated in this regard. Indeed, the original author Frank McKenna has stated that the analyst **must** decide on which algorithms and strategies to choose for a given model and analysis type, alongside the convergence parameters and criteria. This process can get so complex however, to require special 'convergence scripts' that switch algorithms and parameters at each analysis step [López, 2015]. The author thinks this is excessive for a practitioner, especially since nonlinear analysis is usually not taught in undergraduate courses.

Often, to quickly get results, the analyst ends up copy-pasting 'convergence scripts' in hopes of them working without really understanding why a model is not producing the expected results.

To avoid this, the author now believes that the solution algorithms and convergence strategies should be "hidden" from the engineer and implemented very robustly so that when the analysis does not converge, there is almost surely an error with the input structure. This is not possible in general, however, it could work for clearly defined subdomains such as framed buildings, bridges, dams, etc. Each of these would have its own internal "solver" procedures, tailored to their peculiarities and abstracted away from the end-user. Even with free, open-source and powerful software such as OpenSees, it is the author's opinion that modeling real building behavior continues to be hard, parts of it are still too low-level to be usable by practitioners. Furthermore, it lacks an interface and idioms for what structural and geotechnical engineers need. As it stands, it is difficult, time-consuming and error-prone to implement the types of analyses that most structural and geotechnical engineers want in most cases e.g. static, gravity, pushover, modal, dynamic, IDAs, etc. In this work, we show how to integrate OpenSees with the main program by producing 'input files', in this manner we can solve 6.1 without having to write any FEM code and generate those files based on the analysis we wish to conduct.

## 6.3 Analysis types

Regardless of the choices for the structural model and FEM kernel, the author recommends the following analysis to be performed in order

1. gravity
2. small lateral loads
3. eigen
4. pushover (switch to nonlinear models)
5. free-vibration
6. time-history nonlinear (acceleration records)
7. incremental dynamic analysis (IDA)

For each analysis type, the author provides rules of thumb in the form of checklists (see summary 6.1) to ensure that the computer model captures the essence of a real building and is correctly specified in the program. It is fundamental to start with a simple elastic model and switch slowly to a nonlinear model once we pass the eigen stage of analysis.

The following provide necessary conditions to assert that we have implemented the model correctly and that the result can be trusted.

Table 6.1: Recommended checks for different analysis types in order of increasing complexity.

| Name | Nature | Type | PDelta | Checks |
|---|---|---|---|---|
| Gravity | Static | Elastic | No | Base reactions, connectivity |
| Lateral | Static | Elastic | No | Base reactions |
| Eigen | Modal | Elastic | No | Periods, shapes |
| Pushover | Static | Nonlinear | No | Strengths, drifts |
| Free-vibration | Dynamic | Elastic | No | Damping |
| Time-history | Dynamic | Nonlinear | Yes | Earthquake intensity, dynamic capacity |
| IDA | Dynamic | Nonlinear | Yes | Collapse capacity, drift |

### 6.3.1 Gravity analysis

The goal of gravity analysis is to assert that

- the model transmits gravity loads to its base
- the connectivity of the elements and nodes is correctly specified.
- the reactions are numerically correct
- the beams have the correct shear and bending moment distributions
- the columns have the correct axial loads

For OpenSees we recommend the following strategy:

```
pattern Plain 1 Linear {
    eleLoad -ele <eleIDs> -type beamUniform -<load>
}
constraints Transformation
numberer Plain
system BandGeneral
test NormDispIncr 1.0e-08 100
algorithm Newton
integrator LoadControl 0.01 1
analysis Static
initialize
analyze 100
remove recorders
loadConst -time 0.0
```

This applies the gravity loads slowly and resets the virtual time. The author has found that incorrectly specifying the magnitude or sign in the uniform load or in applying the load too quickly can lead to incorrect results which are difficult to detect in subsequent stages.

### 6.3.2 Lateral force analysis

The goal of elastic lateral force analysis is to assert that

- the model transmits lateral loads to the base
- the model has adequate lateral stiffness
- the reactions are numerically correct, both moments and shears
- the columns have the correct bending moment distributions

It is not recommended to check for an exact numerical storey stiffness, both because that concept is not well-defined and because this step acts more as a sanity check, furthermore, the stiffness of our model will be verified subsequently in modal analysis.

For OpenSees we recommend the following strategy:

Place lateral loads at each master node as follows:

```
pattern Plain 2 Linear {
    load 4 1010.0485 0.0 0.0
    load 8 -152.9684 0.0 0.0
    load 12 17.05205 0.0 0.0
    load 16 -0.34394 0.0 0.0
}
constraints Transformation
numberer Plain
system BandGeneral
test NormDispIncr 1.0e-08 100
algorithm Newton
integrator LoadControl 0.01 1
analysis Static
initialize
analyze 100
remove recorders
loadConst -time 0.0
```

The author has found that applying small lateral loads in 100 steps is adequate for all purposes.

### 6.3.3 Eigen analysis

The fundamental goal of eigen analyses is to assert that

1. periods (frequencies) are correct, this implies the ratio of stiffness to mass is adequate
2. spectral shapes are adequate, for a framed building one usually checks that the first or second mode shapes are similar to a cantilever and with a single zero crossing, respectively

Modal or eigen analysis allows us to understand if we have correctly specified the masses and the stiffness of our elements. For framed buildings, the ratio of mass and stiffness that we input into the program must yield a fundamental period of approximately $n/10$ or $n/8$ seconds, where $n$ is the number of storeys of our building.

For OpenSees we recommend the following strategy to compute the modes and periods of a building

```
set eigenvalues [eigen $num_storeys]
set eigen_values_file [open $outdir/eigen_values.csv "w"]
puts $eigen_values_file $eigenvalues
close $eigen_values_file
set eigen_vectors_file [open $outdir/eigen_vectors.csv "w"]
set storeys {1 2 3 4}
set massNodes {4 8 12 16}
foreach mode $storeys {
  foreach m $massNodes {
      lappend eigenvector($mode) [lindex [nodeEigenvector $m $mode 1] 0]
  }
  puts $eigen_vectors_file $eigenvector($mode)
}
set Ts {}
set periods_file [open $outdir/periods.csv "w" ]
foreach val $eigenvalues {
set w [expr {sqrt($val)}]
set period [expr {2*3.1416/$w}]
lappend Ts $period
}
puts $periods_file $Ts
close $periods_file
```

Once the elastic model has passed these checks. The author recommends *slowly* switching the elements to inelastic behavior and performing the checks again.

### 6.3.4 Pushover analysis

The fundamental goal of pushover analysis is to gain insight into the nonlinear behavior of our building, both in strengths and displacements. The analyst should assert that

- the global strength as measured by base shear is adequate
- the yield displacement (drift) of our model is adequate
- the analysis converges past the yielding point

Furthermore, the analyst should inspect one possible collapse mechanism (for the given load pattern) to assert strong-column/weak-beam behavior if

that is included in the design criteria. The author recommends plotting the normalized base shear $V_y/W \approxeq c_s$ against the normalized roof displacement (the total drift), see Fig.6.1, and to define the approximate yield drift and yield strength by the point where the tangent is around 15-25% of the original elastic tangent. The yield drift of reinforced-concrete buildings is around 0.5-2.5% and the normalized yield base shear should be close to the seismic design coefficient.

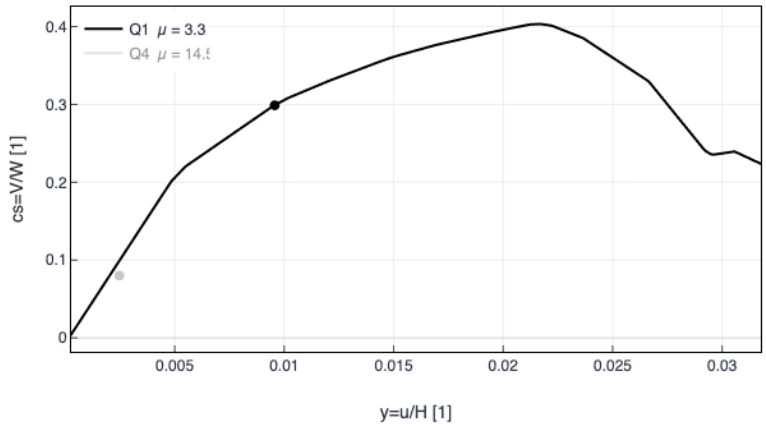



Figure 6.1: Sample normalized pushover for 3 storey building, note there is no obvious yield drift or yield strength point. A good first approximation would be around 1% drift at $c_s = 0.3$, or perhaps 0.5% at $c_s = 0.2$

For OpenSees we recommend the following strategy; after applying gravity loads and resetting the virtual time, define a loading pattern *e.g.* first, by applying a very small lateral load with that pattern as follows:

```
pattern Plain 2 Linear {
    load 4  0.0002097 0.0 0.0
    load 8  0.0002852 0.0 0.0
    load 12 0.000371  0.0 0.0
    load 16 0.000425  0.0 0.0
}
constraints Transformation
numberer Plain
system BandGeneral
test NormDispIncr 1e-09 100 2
algorithm KrylovNewton
integrator DisplacementControl 16 1 0.00084 1000
analysis Static
```

```
analyze 500
```

Framed buildings tend to produce banded matrices, therefore a 'BandGeneral' system is used, although this does not have a significant effect on the overall compute time as our models are still relatively small. We recommend to use the 'RCM' (reverse Cuthill-McKee) numberer [Cuthill and McKee, 1969] as this reduces the bandwidth and the memory requirements of the structure on the system, however, one could also use the 'Plain' numberer with low computational detriment. The author has found that 500 steps up until 5% total drift with the convergence strategies and tolerances shown works well for most cases.

### 6.3.5 Free-vibration analysis

The goal of elastic free-vibration analysis is to assert that

- the model has the correct damping
- the model behaves well under dynamic analysis

The author has found this damping check useful to verify that the model is ready for time-history analyses.

For OpenSees, we recommend the following strategy; let the $\omega_1, \omega_2$ be the chosen frequencies (Sec. 6.6) and $\xi$ be the desired critical damping percentage, then, subject the structure to an initial displacement and let it vibrate freely:

```
set zeta 0.03
# mass damping coefficient based on first and nth modes
set a0 [expr $zeta*2.0*$w1*$w2/($w1 + $w2)];
# stiffness damping coefficient based on first and nth modes
set a1 [expr $zeta*2.0/($w1 + $w2)];
# assign stiffness proportional damping to frame beams & columns
region 1 -eleRange 1 3 -rayleigh 0.0 0.0 $a1 0.0;
# assign mass proportional damping to structure (assign to nodes with mass)
region 2 -node 3 3 -rayleigh $a0 0.0 0.0 0.0;

system FullGeneral
constraints Transformation
numberer Plain
integrator LoadControl 0.1 10
test NormDispIncr 1.0e-8 100 2
algorithm KrylovNewton
analysis Static

recorder Node -file $outdir/disp.csv -time -node 3 -dof 1 disp

timeSeries Constant 1 -factor 1
pattern Plain 1 1 {
    sp 3 1 0.001 ;# horizontal displacement of 0.001 at node 3
}

analyze 10
wipeAnalysis
setTime 0.0
remove loadPattern 1
# begin dynamic
integrator Newmark 0.5 0.25
analysis Transient

recorder Node -file $outdir/base_shear.csv -time -node 1 -dof 1 reaction
recorder Drift -file $outdir/drift.csv -time -iNode 1 -jNode 2 -dof 1 -perpDirn 2

analyze 1000 0.01
```

Afterward, plot the free vibration displacements and note two subsequent peak displacements: $u_1, u_2$. Check that the viscous damping $\xi$ entered in

OpenSees and the approximate first-mode damping percentage are close numerically [Chopra, 2012]:

$$\xi \approx \frac{1}{2\pi} \ln \frac{u_1}{u_2}$$

To integrate the equations of motion, the author recommends the use 'Newmark' integrator with $\gamma = 1/2$ and $\beta = 1/4$ i.e. the constant acceleration method, as it is unconditionally stable and well documented in the literature.

### 6.3.6 Nonlinear time-history analyses

The fundamental goal of nonlinear time-history analyses is to assert that

- the earthquake intensity and record parameters are correctly set
- the earthquake intensity that causes first yield is approximately $S_{ay}$
- the model behaves well under high accelerations and does not have an unexpected collapse mechanism

This step is generally used to prepare the model for IDAs, by slowly turning up the intensity from elastic, to $Sa_y$ up until the onset of instability.

To solve the nonlinear equations of dynamical equilibrium, it is usual to employ Newton-like methods as they are the most straightforward to understand and offer potential quadratic convergence speeds. The disadvantage of these methods is that they can be computationally intensive because the tangent stiffness matrix must be computed at each step. If this is the case, one could use Quasi-Newton methods that do no need this matrix at every iteration, and instead use the initial tangent matrix (by adding '-initial'). It is important to provide acceleration records with small and constant time steps of $dt = 0.01$ or smaller. The recommended dynamic analysis options for a framed building in OpenSees are as follows:

```
constraints Transformation
numberer RCM
integrator Newmark 0.5 0.25
# integrator TRBDF2
# integrator HHT 0.6
system FullGeneral
analysis Transient
test NormDispIncr 1e-8 1000 2
set ts "Series -dt <record_dt> -filePath <record_dir> -factor <factor>"
pattern UniformExcitation 2 1 -accel $ts
set duration 108.915
set record_dt 0.005
analyze <duration>*<record_dt>
```

The choice of a convergence test is still relatively ad-hoc, however, the author has found 'NormUnbalance' or 'NormDispIncr' to work well. For Newton-like methods, the tolerance chosen should be close to half the machine precision (1.0e-8) to guarantee that the next residual will achieve full precision [Zienkiewicz and Taylor, 2013]. A maximum number of iterations of 1000 is recommended, if convergence is still not achieved, this means that the model is near or has crossed a limit point. When this happens, this is indicative of some type of instability [Ibarra and Krawinkler, 2005].

In this case, to integrate the equations of motion the author recommends trying out the Hilbert-Hughes-Taylor method [Hilber et al., 1977], which introduces some numerical damping and high-frequency dissipation which might help achieve convergence [Chung and Hulbert, 1993].

### 6.3.7 Incremental dynamic analysis (IDA)

The fundamental goal of an IDA is to capture the variability of the structural response against a set of records with distinct frequency contents and increasing intensity to gain insight into the dynamical behavior of a structure close to the onset of collapse.

In the classical IDA [Vamvatsikos and Cornell, 2002] [Vamvatsikos, 2004], one employs a suite of selected ground motions characterized by sources of a similar magnitude. For each ground motion, the structure is subjected to a monotonically increasing intensity (in $S_a$), obtaining for each record and each intensity, the structural response in terms of EDPs, *e.g.* peak inter-storey drift. In this manner it is conjectured that the response of the building across these intensity levels and with distinct frequency content could offer insight into its dynamical capacity and performance, starting from the elastic stage up until the onset of collapse.

The set of IDA curves plotted in EDP vs IM fashion $(\gamma, S_a(5\%, T_1))$ is called an IDA-plot (Fig.6.2). Using these results, one can statistically summarize the structural response as a conditional probability distribution of the EDP given an intensity. Moreover, one can estimate the probability of collapse given a level of intensity, which is a useful quantity in damage analysis and risk analysis in general.

For example, one can compute the median intensity in which collapse (dynamic instability) is achieved. In this procedure, structural collapse can be identified as a softening in each curve that can be identified visually as a flat line. This softening corresponds to numerical instability, and if the model satisfies some robustness properties, then this corresponds to dynamical instability [López, 2015]

When performing automated IDAs, the author recommends to assert that

- median collapse $S_a/g$ is somewhat above the seismic design coefficient.
- median inter-storey collapse drift is reasonable, around 4-10%, although this has high variance

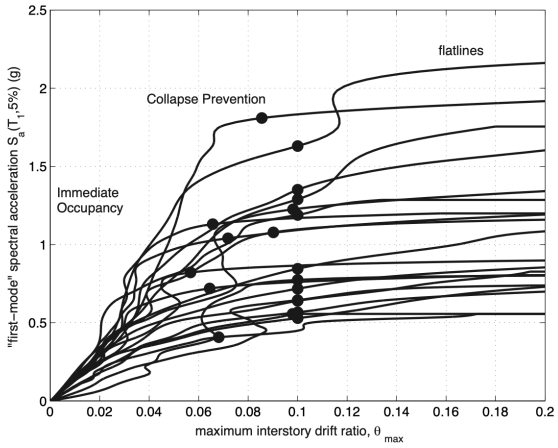



Figure 6.2: A set of IDA curves for a 9 storey steel building, plotting the maximum interstorey drift ratio attained at any time in the structural analysis against the first mode spectral acceleration at a 5% critical damping level. Each line corresponds to a given record at increasing levels of intensity. The black dots indicate a 80% decrease in "stiffness" and correspond to a definition of the collapse-prevention limit state which is at the onset of dynamic instability, [Vamvatsikos, 2004]

For OpenSees we recommend the following strategy:

```
set tol 1e-6
constraints Transformation
numberer RCM
integrator TRBDF2
# integrator HHT 0.6; #the smaller the alpha, the greater the numerical damping
system FullGeneral
analysis Transient
test NormDispIncr $tol 1000 0; # switch to 2 for output

set algorithms {
 "Newton"
 "KrylovNewton"
 "BFGS"
 "Broyden"
}

set num_algorithms [llength $algorithms]
set num_subdivisions 10
set max_subdivisions 2 ;# divide original timestep into 10^2 = 100 steps

# set results_file $results_dir/results.csv
set results_file results.csv
set fp [open $results_file w+]
```

```
set aix 0
set break_outer 0
set converged 0
set time [getTime]
while {$time <= $duration && !$break_outer} {
  # reset to dt=record_dt, algorithm_0 = Newton
  set time [getTime]
  set analysis_dt $record_dt
  set algorithm [lindex $algorithms $aix]
  # puts "Trying $algorithm..."
  algorithm $algorithm
  set converged [analyze 1 $analysis_dt]
  set subdivision_retries 0
  # puts $converged
  while {$converged != 0} {
    # subdivide timestep
    incr subdivision_retries
    if {$subdivision_retries >= $max_subdivisions} {
      set time [getTime]
      puts $fp "Algorithm $algorithm did not converge at time
       $time $converged $subdivision_retries"
      incr aix
      if {$aix >= $num_algorithms} {
        puts $fp "FAILURE, couldn't converge with any algorithm or reduced step"
        puts $fp "time $time"
        # puts $fp "max drift $time"
        set break_outer 1
      }
      break
    }
    set n [expr {$num_subdivisions ** $subdivision_retries}]
    set analysis_dt [expr {$record_dt/$n}]
    set time [getTime]
    set converged [analyze $n $analysis_dt]
    puts $converged
    if {$converged == 0} {set aix 0}; # if we converged then reset to algorithm_0
  }
}

close $fp
```

To achieve convergence of the equilibrium equations close to a singular stiffness matrix (hypostatic condition) a special 'converge script' was used [López, 2015]. This procedure detects "big" displacement norms and divides the integration step in 10 sub-intervals twice (until 100 sub-steps), wherein different integration strategies are tried until convergence is achieved or the program exits indicating numerical instability [Haselton et al., 2009]. Using this script, the convergence is 'forced' and the run ends when we encounter astronomically large displacements, this manifests itself in the IDA curves as flat lines.

Although simpler and more powerful collapse converge algorithms exist that accurately stop when the structure has collapsed [Scott and Fenves, 2010], the resulting IDA curves do not flatline, instead, they go vertical at the collapse capacity for the record.

## 6.4 Mechanical model

The mechanical model used consists of bar elements (beam-columns), for which their non-linear behavior is concentrated at their ends as infinitesimal plastic hinges, see Fig.6.3. The springs used in these hinges implement the IMK hysteretic model [Ibarra et al., 2005] with moment-rotation relationships that are representative of the structural behavior of reinforced concrete elements. This implementation can accurately describe the structural response of an element from the onset of cracking to total stiffness and strength degradation due to phenomena such as concrete crushing, and rebar buckling, among others. The implementation of IMK in OpenSees has been calibrated with more than 500 experimental results and has been used successfully in the past to build models that attempted to estimate numerical instability *i.e.* the onset of structural collapse [Haselton et al., 2009].

The degrees of freedom of the frame are the rotations at each joint, and the lateral displacements of each story due to a rigid diaphragm hypothesis. The columns at the base are fixed and the gravitational loads are placed directly on the beams, at the same time, the story masses exert a P-Delta effect on the structure (illustrated as $m$'s on the rightmost joints) which is necessary can model collapse due to dynamical instability. Structural damping is considered classical (Rayleigh), which is proportional to both mass and stiffness and is illustrated with viscous dampers (mass) at the leftmost part of the frame and with inter-storey diagonal dampers (stiffness).

The option to use 'PDelta' is used to incorporate second-order moments into the equations. This is implemented for the column element tag definitions. The author recommends starting without P-Delta and then turning it on after most checks have passed. This model has a good balance that captures the overall behavior of framed reinforced concrete structures and is also relatively fast, performing a single IDA in a couple of seconds with a commercially available laptop.

### IMK springs implementation

This is a phenomenological model that is calibrated with lab tests to accurately represent cyclical behavior with stiffness and strength degradation until collapse. They model the global behavior of the cross-section, without attending to a specific phenomenon, such as concrete cracking, rebar fracture, etc. Therefore, these models do not offer insight into what really would have happened.

This model uses a trilinear backbone curve for the moment-rotation response with parameters which are derived from laboratory tests, see Fig.6.4.

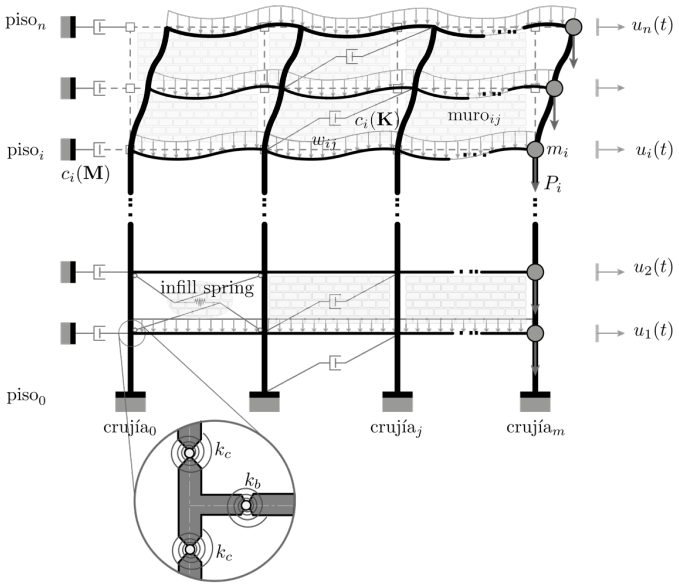



Figure 6.3: Example of a 2D framed mechanical model with infinitesimal springs for nonlinear beam-column elements, infill walls (not included in this thesis). Note the degrees of freedom are the rotations at each join, and the lateral displacements of each storey.

Ibarra and Krawinkler [Ibarra and Krawinkler, 2005] show in Appendices B.2 and B.3 how to implement such spring models, however, they have to be modified for a member with two springs and an elastic beam-column element in series.

To obtain the proper response of the member, the elastic deformation should be completely contained in the interior beam-column element, and the plasticity in the springs. However, OpenSees cannot implement such infinite stiffness elements and has issues inverting these stiffness matrices. Although the formulas shown in these appendices are correct only for a single spring and beam-column element, the correct formulas for a spring-column-spring in series needed for most beams and columns will be shown.

To avoid numerical instabilities, choose a natural number $n >> 1$, such as 10, 20 or 50. Let the stiffness of the spring be proportional to that of the elastic beam-column $K_s = nK_{bc}$. Since the member is in series the stiffnesses are:

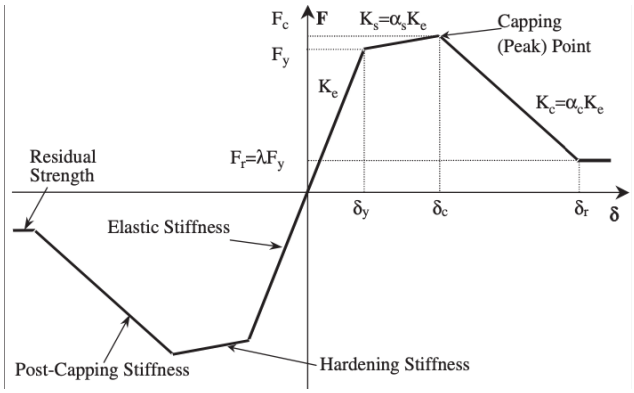



Figure 6.4: Backbone curve for an IMK material spring, showing trilinear behavior. From elastic stiffnesses, to a hardening phase and ultimately a softening branch representing a loss of strength. It is possible to set residual strength which is numerically useful as well.

$$K_{mem} = \frac{K_s K_{bc}}{K_s + 2K_{bc}}.$$

Therefore,

$$K_{bc} = \frac{n+2}{n} K_{mem}, \qquad K_s = (n+2)K_{mem}.$$

Since the member is in series, the total rotation of the member is the sum of the rotation of the spring plus the beam-column:

$$\Delta\theta_{mem} = \Delta\theta_s + \Delta\theta_{bc} = \frac{\Delta M}{\alpha_{s,s} K_s} + \frac{\Delta M}{K_{bc}} = \frac{\Delta M}{\alpha_{mem} K_{mem}}.$$

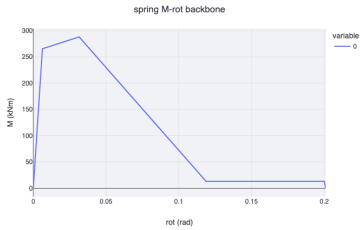
Then the post-yielding coefficient of the spring is

$$\alpha_{s,s} = \frac{\alpha_{mem}}{n + 2 - n\alpha_{mem}}.$$

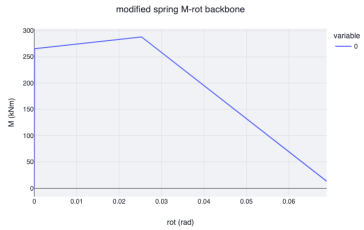
Thus, the backbone for the spring is quite different to the intended backbone of the member (see Fig.6.5).

For the implementation of the IMK-spring building models in OpenSees, the author recommends the following code:

```
#!/usr/local/bin/opensees
# IMKFrame, storeys 3, bays 2
wipe
model BasicBuilder -ndm 2 -ndf 3
geomTransf Linear 1
geomTransf PDelta 2
node 0 0.0 0.0
fix 0 1 1 1
node 1 8.0 0.0
fix 1 1 1 1
node 2 16.0 0.0
fix 2 1 1 1
node 3 0.0 4.5 -mass 256.0000 1e-9 1e-9
node 4 8.0 4.5
```




(a) Expected member moment-rotation backbone.




(b) Spring moment-rotation backbone required to achieve the member's backbone.

Figure 6.5: Backbone curves for IMK material spring, Fig a. shows the intended moment-rotation backbone, Fig b. shows the modified spring backbone to achieve Fig a.

```
node 5 16.0 4.5
node 6 0.0 7.7 -mass 256.0000 1e-9 1e-9
node 7 8.0 7.7
node 8 16.0 7.7
node 9 0.0 10.9 -mass 256.0000 1e-9 1e-9
node 10 8.0 10.9
node 11 16.0 10.9
node 12 0.0 0.0
equalDOF 0 12 1 2
node 13 8.0 0.0
equalDOF 1 13 1 2
node 14 16.0 0.0
equalDOF 2 14 1 2
node 15 0.0 4.5
equalDOF 3 15 1 2
node 16 0.0 4.5
equalDOF 3 16 1 2
node 17 0.0 4.5
equalDOF 3 17 1 2
node 18 8.0 4.5
equalDOF 4 18 1 2
node 19 8.0 4.5
equalDOF 4 19 1 2
node 20 8.0 4.5
equalDOF 4 20 1 2
node 21 8.0 4.5
equalDOF 4 21 1 2
node 22 16.0 4.5
equalDOF 5 22 1 2
node 23 16.0 4.5
equalDOF 5 23 1 2
node 24 16.0 4.5
equalDOF 5 24 1 2
node 25 0.0 7.7
equalDOF 6 25 1 2
node 26 0.0 7.7
equalDOF 6 26 1 2
node 27 0.0 7.7
equalDOF 6 27 1 2
node 28 8.0 7.7
equalDOF 7 28 1 2
node 29 8.0 7.7
equalDOF 7 29 1 2
node 30 8.0 7.7
equalDOF 7 30 1 2
node 31 8.0 7.7
equalDOF 7 31 1 2
node 32 16.0 7.7
equalDOF 8 32 1 2
node 33 16.0 7.7
equalDOF 8 33 1 2
node 34 16.0 7.7
equalDOF 8 34 1 2
node 35 0.0 10.9
equalDOF 9 35 1 2
node 36 0.0 10.9
equalDOF 9 36 1 2
```

```
node 37 8.0 10.9
equalDOF 10 37 1 2
node 38 8.0 10.9
equalDOF 10 38 1 2
node 39 8.0 10.9
equalDOF 10 39 1 2
node 40 16.0 10.9
equalDOF 11 40 1 2
node 41 16.0 10.9
equalDOF 11 41 1 2
equalDOF 3 4 1
equalDOF 3 5 1
equalDOF 6 7 1
equalDOF 6 8 1
equalDOF 9 10 1
equalDOF 9 11 1
element elasticBeamColumn 2 12 15 1e+05 2.60308e+07 0.0782855 2
element elasticBeamColumn 5 13 18 1e+05 2.60308e+07 0.0782855 2
element elasticBeamColumn 8 14 22 1e+05 2.60308e+07 0.0782855 2
element elasticBeamColumn 17 17 25 1e+05 2.60308e+07 0.0347935 2
element elasticBeamColumn 20 20 28 1e+05 2.60308e+07 0.0347935 2
element elasticBeamColumn 23 23 32 1e+05 2.60308e+07 0.0347935 2
element elasticBeamColumn 32 27 35 1e+05 2.60308e+07 0.00869838 2
element elasticBeamColumn 35 30 37 1e+05 2.60308e+07 0.00869838 2
element elasticBeamColumn 38 33 40 1e+05 2.60308e+07 0.00869838 2
element elasticBeamColumn 11 16 21 1e+05 2.60308e+07 0.0154638 1
element elasticBeamColumn 14 19 24 1e+05 2.60308e+07 0.0154638 1
element elasticBeamColumn 26 26 31 1e+05 2.60308e+07 0.0068728 1
element elasticBeamColumn 29 29 34 1e+05 2.60308e+07 0.0068728 1
element elasticBeamColumn 41 36 39 1e+05 2.60308e+07 0.0017182 1
element elasticBeamColumn 44 38 41 1e+05 2.60308e+07 0.0017182 1
uniaxialMaterial ModIMKPeakOriented 200001 275494.17 0.13 0.13 919.25 -919.25 17.787 17.787 17.787 17.787
     1. 1. 1. 1. 0.01750088 0.01750088 0.02792747 0.02792747 0.01 0.01 0.04876507 0.04876507 1. 1.
element zeroLength 1 0 12 -mat 200001 -dir 6
uniaxialMaterial ModIMKPeakOriented 200003 275494.17 0.13 0.13 919.25 -919.25 17.787 17.787 17.787 17.787
     1. 1. 1. 1. 0.01750088 0.01750088 0.02792747 0.02792747 0.01 0.01 0.04876507 0.04876507 1. 1.
element zeroLength 3 15 3 -mat 200003 -dir 6
uniaxialMaterial ModIMKPeakOriented 200004 275494.17 0.13 0.13 919.25 -919.25 17.787 17.787 17.787 17.787
     1. 1. 1. 1. 0.01750088 0.01750088 0.02792747 0.02792747 0.01 0.01 0.04876507 0.04876507 1. 1.
element zeroLength 4 1 13 -mat 200004 -dir 6
uniaxialMaterial ModIMKPeakOriented 200006 275494.17 0.13 0.13 919.25 -919.25 17.787 17.787 17.787 17.787
     1. 1. 1. 1. 0.01750088 0.01750088 0.02792747 0.02792747 0.01 0.01 0.04876507 0.04876507 1. 1.
element zeroLength 6 18 4 -mat 200006 -dir 6
uniaxialMaterial ModIMKPeakOriented 200007 275494.17 0.13 0.13 919.25 -919.25 17.787 17.787 17.787 17.787
     1. 1. 1. 1. 0.01750088 0.01750088 0.02792747 0.02792747 0.01 0.01 0.04876507 0.04876507 1. 1.
element zeroLength 7 2 14 -mat 200007 -dir 6
uniaxialMaterial ModIMKPeakOriented 200009 275494.17 0.13 0.13 919.25 -919.25 17.787 17.787 17.787 17.787
     1. 1. 1. 1. 0.01750088 0.01750088 0.02792747 0.02792747 0.01 0.01 0.04876507 0.04876507 1. 1.
element zeroLength 9 22 5 -mat 200009 -dir 6
uniaxialMaterial ModIMKPeakOriented 200010 162596.06 0.13 0.13 612.83 -612.83 16.682 16.682 16.682 16.682
     1. 1. 1. 1. 0.01760536 0.01760536 0.03852301 0.03852301 0.01 0.01 0.05989741 0.05989741 1. 1.
element zeroLength 10 3 16 -mat 200010 -dir 6
uniaxialMaterial ModIMKPeakOriented 200012 162596.06 0.13 0.13 612.83 -612.83 16.682 16.682 16.682 16.682
     1. 1. 1. 1. 0.01760536 0.01760536 0.03852301 0.03852301 0.01 0.01 0.05989741 0.05989741 1. 1.
element zeroLength 12 21 4 -mat 200012 -dir 6
uniaxialMaterial ModIMKPeakOriented 200013 162596.06 0.13 0.13 612.83 -612.83 16.682 16.682 16.682 16.682
     1. 1. 1. 1. 0.01760536 0.01760536 0.03852301 0.03852301 0.01 0.01 0.05989741 0.05989741 1. 1.
element zeroLength 13 4 19 -mat 200013 -dir 6
uniaxialMaterial ModIMKPeakOriented 200015 162596.06 0.13 0.13 612.83 -612.83 16.682 16.682 16.682 16.682
     1. 1. 1. 1. 0.01760536 0.01760536 0.03852301 0.03852301 0.01 0.01 0.05989741 0.05989741 1. 1.
element zeroLength 15 24 5 -mat 200015 -dir 6
uniaxialMaterial ModIMKPeakOriented 200016 208338.54 0.13 0.13 735.40 -735.40 17.047 17.047 17.047 17.047
    1. 1. 1. 1. 0.01754794 0.01754794 0.03268178 0.03268178 0.01 0.01 0.05375955 0.05375955 1. 1.
element zeroLength 16 3 17 -mat 200016 -dir 6
uniaxialMaterial ModIMKPeakOriented 200018 208338.54 0.13 0.13 735.40 -735.40 17.047 17.047 17.047 17.047
     1. 1. 1. 1. 0.01754794 0.01754794 0.03268178 0.03268178 0.01 0.01 0.05375955 0.05375955 1. 1.
element zeroLength 18 25 6 -mat 200018 -dir 6
uniaxialMaterial ModIMKPeakOriented 200019 208338.54 0.13 0.13 735.40 -735.40 17.047 17.047 17.047 17.047
     1. 1. 1. 1. 0.01754794 0.01754794 0.03268178 0.03268178 0.01 0.01 0.05375955 0.05375955 1. 1.
element zeroLength 19 4 20 -mat 200019 -dir 6
uniaxialMaterial ModIMKPeakOriented 200021 208338.54 0.13 0.13 735.40 -735.40 17.047 17.047 17.047 17.047
     1. 1. 1. 1. 0.01754794 0.01754794 0.03268178 0.03268178 0.01 0.01 0.05375955 0.05375955 1. 1.
element zeroLength 21 28 7 -mat 200021 -dir 6
uniaxialMaterial ModIMKPeakOriented 200022 208338.54 0.13 0.13 735.40 -735.40 17.047 17.047 17.047 17.047
     1. 1. 1. 1. 0.01754794 0.01754794 0.03268178 0.03268178 0.01 0.01 0.05375955 0.05375955 1. 1.
element zeroLength 22 5 23 -mat 200022 -dir 6
uniaxialMaterial ModIMKPeakOriented 200024 208338.54 0.13 0.13 735.40 -735.40 17.047 17.047 17.047 17.047
     1. 1. 1. 1. 0.01754794 0.01754794 0.03268178 0.03268178 0.01 0.01 0.05375955 0.05375955 1. 1.
element zeroLength 24 32 8 -mat 200024 -dir 6
uniaxialMaterial ModIMKPeakOriented 200025 120568.05 0.13 0.13 490.27 -490.27 17.146 17.146 17.146 17.146
     1. 1. 1. 1. 0.01767534 0.01767534 0.04570061 0.04570061 0.01 0.01 0.06744225 0.06744225 1. 1.
element zeroLength 25 6 26 -mat 200025 -dir 6
uniaxialMaterial ModIMKPeakOriented 200027 120568.05 0.13 0.13 490.27 -490.27 17.146 17.146 17.146 17.146
     1. 1. 1. 1. 0.01767534 0.01767534 0.04570061 0.04570061 0.01 0.01 0.06744225 0.06744225 1. 1.
element zeroLength 27 31 7 -mat 200027 -dir 6
uniaxialMaterial ModIMKPeakOriented 200028 120568.05 0.13 0.13 490.27 -490.27 17.146 17.146 17.146 17.146
     1. 1. 1. 1. 0.01767534 0.01767534 0.04570061 0.04570061 0.01 0.01 0.06744225 0.06744225 1. 1.
element zeroLength 28 7 29 -mat 200028 -dir 6
uniaxialMaterial ModIMKPeakOriented 200030 120568.05 0.13 0.13 490.27 -490.27 17.146 17.146 17.146 17.146
     1. 1. 1. 1. 0.01767534 0.01767534 0.04570061 0.04570061 0.01 0.01 0.06744225 0.06744225 1. 1.
element zeroLength 30 34 8 -mat 200030 -dir 6
```

```
uniaxialMaterial ModIMKPeakOriented 200031 148074.06 0.13 0.13 588.32 -588.32 16.895 16.895 16.895 16.895
     1. 1. 1. 1. 0.01765357 0.01765357 0.04346107 0.04346107 0.01 0.01 0.06508778 0.06508778 1. 1.
element zeroLength 31 6 27 -mat 200031 -dir 6
uniaxialMaterial ModIMKPeakOriented 200033 148074.06 0.13 0.13 588.32 -588.32 16.895 16.895 16.895 16.895
     1. 1. 1. 1. 0.01765357 0.01765357 0.04346107 0.04346107 0.01 0.01 0.06508778 0.06508778 1. 1.
element zeroLength 33 35 9 -mat 200033 -dir 6
uniaxialMaterial ModIMKPeakOriented 200034 148074.06 0.13 0.13 588.32 -588.32 16.895 16.895 16.895 16.895
     1. 1. 1. 1. 0.01765357 0.01765357 0.04346107 0.04346107 0.01 0.01 0.06508778 0.06508778 1. 1.
element zeroLength 34 7 30 -mat 200034 -dir 6
uniaxialMaterial ModIMKPeakOriented 200036 148074.06 0.13 0.13 588.32 -588.32 16.895 16.895 16.895 16.895
     1. 1. 1. 1. 0.01765357 0.01765357 0.04346107 0.04346107 0.01 0.01 0.06508778 0.06508778 1. 1.
element zeroLength 36 37 10 -mat 200036 -dir 6
uniaxialMaterial ModIMKPeakOriented 200037 148074.06 0.13 0.13 588.32 -588.32 16.895 16.895 16.895 16.895
     1. 1. 1. 1. 0.01765357 0.01765357 0.04346107 0.04346107 0.01 0.01 0.06508778 0.06508778 1. 1.
element zeroLength 37 8 33 -mat 200037 -dir 6
uniaxialMaterial ModIMKPeakOriented 200039 148074.06 0.13 0.13 588.32 -588.32 16.895 16.895 16.895 16.895
     1. 1. 1. 1. 0.01765357 0.01765357 0.04346107 0.04346107 0.01 0.01 0.06508778 0.06508778 1. 1.
element zeroLength 39 40 11 -mat 200039 -dir 6
uniaxialMaterial ModIMKPeakOriented 200040 82469.38 0.13 0.13 392.21 -392.21 22.028 22.028 22.028 22.028
     1. 1. 1. 1. 0.01783197 0.01783197 0.05000000 0.05000000 0.01 0.01 0.07258783 0.07258783 1. 1.
element zeroLength 40 9 36 -mat 200040 -dir 6
uniaxialMaterial ModIMKPeakOriented 200042 82469.38 0.13 0.13 392.21 -392.21 22.028 22.028 22.028 22.028
     1. 1. 1. 1. 0.01783197 0.01783197 0.05000000 0.05000000 0.01 0.01 0.07258783 0.07258783 1. 1.
element zeroLength 42 39 10 -mat 200042 -dir 6
uniaxialMaterial ModIMKPeakOriented 200043 82469.38 0.13 0.13 392.21 -392.21 22.028 22.028 22.028 22.028
     1. 1. 1. 1. 0.01783197 0.01783197 0.05000000 0.05000000 0.01 0.01 0.07258783 0.07258783 1. 1.
element zeroLength 43 10 38 -mat 200043 -dir 6
uniaxialMaterial ModIMKPeakOriented 200045 82469.38 0.13 0.13 392.21 -392.21 22.028 22.028 22.028 22.028
     1. 1. 1. 1. 0.01783197 0.01783197 0.05000000 0.05000000 0.01 0.01 0.07258783 0.07258783 1. 1.
element zeroLength 45 41 11 -mat 200045 -dir 6
```

It is recommended to not use distributed loads on these elements, rather use point loads and moments on the nodes.

## 6.5 Modeling damage and post-yielding cyclic behavior in reinforced-concrete beam-columns

For framed buildings, the correct simulation of structural damage in beam-column elements is paramount to capture the influence of the deterioration of strength and stiffness in the response and to simulate structural collapse, which has profound consequences in risk analysis. The selection of structural element models is therefore of foremost importance, primarily with the aim of linking the damage suffered to a loss measure. The author recommends that this link is deterministic, based only on the mechanical properties of the piece and its cross-section.

Along these lines, [Eldawie, 2020] and [George, 2018] provide comprehensive summaries on the different modeling schemes for reinforced concrete elements and frames from the elastic phase until the onset of collapse. They present good guidance on how to calibrate models and estimate their parameters via semi-empirical equations.

There are two main categories of nonlinear elements: concentrated and distributed plasticity. [Fragiadakis and Papadrakakis, 2008] gives a good overview of the pros and cons of each approach. Among them, the tradeoffs between choosing 'fiber' elements vs 'springs' are discussed. One of the main advantages of fiber models is that they can capture axial-moment interactions, as opposed to springs which can only account for this interaction indirectly. However, a disadvantage of fiber elements is that they require more computational resources, and calibrating them to experimental results is inherently harder than the spring counterpart.

However, fiber models can capture $(P, M_x, M_y)$ interaction accurately, and they are excellent for monotonic simulations and to reproduce yielding and pre-yielding behavior. Unfortunately, realistic degradation of stiffness and strength is difficult to reproduce, very difficult to accurately simulate ultimate states of curvature/rotation and the effect of stirrups directly i.e the effect of confinement is done through choosing a different concrete model for the core. Classical models do not consider the effect of shear in the global behavior of the reinforced-concrete member, [Elwood, 2004], this must be considered separately in a post-processing phase.

## Semi-empirical equations for deformation capacity of reinforced-concrete beam-columns

Due to these limitations of fiber models to capture the cyclical reversible behavior very far from post-yielding where where the member is reaching maximum capacity, we are somewhat forced to employ semi-empirical models such as IMK. As discussed on the previous section 6.4, the author recommends using infinitesimal springs that implement the IMK hysteretic rule [Ibarra et al., 2005].

Chapter 7 expounds linking structural damage to a monetary loss computation through a physical approach based on dissipated energy and post-yielding states.

[Haselton et al., 2016] provides rigorous statistical calibration to experimental test data which we follow in this work. Effective stiffness, peak and post-peak inelastic forces, deformations, and energy dissipation are functions of the geometrical and mechanical (material) properties of the member instance itself and the loading conditions *i.e.* axial load primarily but also shear loading.

Plastic rotation capacity $\theta_p$ is associated with the components' behavior before local instabilities such as buckling of reinforcing bars. This represents the initiation of yielding and the loss of strength:

$$\theta_p = 0.12(1 + 0.55a_{sl})(0.16)^{n_0}(0.02 + 40\rho_w)^{0.43}(0.54)^{0.01f'_c}(0.66)^{0.1s_n}(2.27)^{10\rho}$$

Post-capping plastic rotation capacity $\theta_{pc}$ is associated with component behavior after the occurrence of local instabilities. The smaller the value, the sooner the component reaches zero bending strength capacity:

$$\theta_{pc} = 0.76(0.031)^{n_0}(0.02 + 40\rho_w)^{1.02} \leq 0.10$$

Where $a_{sl}$ is a slippage coefficient that is 1 if bar slippage of the longitudinal bars from their anchorage beyond the section of the maximum moment is possible and 0 otherwise, $n_0$ is the axial load ratio, $\rho_w$ is the web steel ratio and $s_n$ is the separation of stirrups. It is important to note that these are capacities *e.g.* deltas, such that

$$\theta_u = \theta_y + \theta_p + \theta_{pc}.$$

## 6.6 Damping in framed buildings

The true damping mechanisms in buildings are not fully understood, moreover, when structural damage occurs the unknown total elastic damping increases considerably. For example: in reinforced-concrete buildings, this damping goes from 2-5% in the linear elastic range to 7-10% or higher when there is considerable structural damage.

Damping forces are usually modeled via viscous damping mechanisms contained in matrix $C$, however, estimation of $C$ based on the initial material and geometrical properties of the structure is not possible. Therefore, a widely used alternative is to postulate $C$ as a linear combination of the mass and the stiffness matrix

$$C = a_0 M + a_1 K$$

this is called Rayleigh damping and has dubious theoretical justification [Chopra, 2012], especially the mass-proportional term which violates physical law [Wilson, 2002] [Chopra and McKenna, 2016]. However, the numerical implementation is straightforward for finite element models and can include the increase in damping due to damage via the tangent stiffness matrix. Therefore Rayleigh damping is a good compromise between exactness, uncertainty and computational effort required.

To implement Rayleigh damping, however, the engineer can only select two "representative" modes, besides the percentage of critical damping of the complete structure $\xi$:

$$a_0 = 2\xi \frac{\omega_i \omega_j}{\omega_i + \omega_j}, \quad a_1 = \frac{2\xi}{\omega_i + \omega_j}$$

A good algorithm chooses them such that the ones that contribute more effective modal mass have ratios closer to the desired $\xi$

$$\xi_i = \frac{a_0}{2\omega_i} + \frac{a_1}{\omega_i}$$

In other words, choose the modes that minimize $\sum_i M_i^* e_i$ where $M_i^*$ is the effective modal mass for mode $i$ and $e_i$ is the relative error for mode $i$:

$$e_i = \left| \frac{\xi_i - \xi}{\xi} \right|$$

As with most modal selection schemes, higher modes end up having higher damping and therefore will have a negligible impact on the response. A good test for the algorithm is that it will tend to select the first or second mode alongside a higher mode that is usually not the last. A rule of thumb for tall buildings used in practice is to select the first and the fourth modes.

## 6.7 Responsibility and objective of the module

The main responsibility of this module is to perform **IDA**s on a given designed building instance. It must produce a complete and exhaustive set of results that contain the structural time-history response of all DOFs for displacements, velocities, accelerations, and forces as well as the internal stresses and strains for the structural elements at every instant. These results will be used by the Loss Module to compute different loss and risk measures.

#### UML diagrams and code structure

In the PlantUML notation, classes are specified by rectangles with their name at the top. Their properties (called attributes) are shown below their name, while their most relevant methods (own functions) are listed at the bottom third of the rectangle.

Relationships between classes are given as lines, the nature of this relationship is specified by the arrowhead type:

1. △ represents implementations of interfaces.
2. ▲ represents a relationship of composition or being contained by the source class.

Fig 6.6 shows the main objects in the module. The reader is invited to consult the public source code for the complete implementation.

### 6.7.1 Structural analysis interface class

The goal of **StructuralAnalysis** is to build the correct **Recorder** classes for the desired analysis type. This class is a coordinator between an in-memory reference to the **FEM** instance. They have a 'path' attribute that references where the results will be written. The important method is '.run()' which needs a **'Recorder'** object

### Recorder interfaces

The objective of this class provide pointers to a **StructuralResultView** object of the corresponding analysis type. Implementations of this class are wrappers around the available OpenSees recorders, and can provide the string representation of those recorders based on the node and element definitions in the **FEM** instance. They also reference the different solver classes used in OpenSees for the selected analysis type.

The following hierarchy of recorders has been implemented, and can be composed to perform multiple analyses in a row:

1. EigenRecorder (modal analysis)
2. KRecorder (stiffness matrix computation)

3. Gravity
4. Static
5. Pushover
6. Timehistory
7. Gravity > Static
8. Gravity > Pushover
9. Gravity > Timehistory

## StructuralResultView class

These are wrapper classes around reading the data produced by the kernel, this means they only hold pointers to the files in disk where the results of the OpenSees recorders write to. Callers can consult the result of different EDPs *e.g.* to consult the moment-rotation response of springs, the object figures out how to get the filepath of the given analysis, and reads the time-history files, merges them into a dataframe so they can be used later.

## IDA class

The objective of this class is to run IDA analysis.

The author recommends summarizing the results as flat tables in column-major fashion. The correct abstraction for this is known as a dataframe:

| | Sa | collapse | freq | inf | intensity | record |
|---|---|---|---|---|---|---|
| 0 | 0.380055 | False | 0.054995 | 0.040775 | 0.081549 | 12_ROM140995NS.csv |
| 1 | 0.760110 | False | 0.036356 | 0.122324 | 0.163099 | 12_ROM140995NS.csv |
| 2 | 1.140165 | False | 0.011906 | 0.203874 | 0.244648 | 12_ROM140995NS.csv |
| 3 | 1.520220 | True | 0.006012 | 0.285423 | 0.326198 | 12_ROM140995NS.csv |
| 4 | 1.900275 | True | 0.001638 | 0.366972 | 0.407747 | 12_ROM140995NS.csv |
| 5 | 2.280329 | True | 0.001061 | 0.448522 | 0.489297 | 12_ROM140995NS.csv |
| 6 | 0.380055 | False | 0.054995 | 0.040775 | 0.081549 | 11_TL101294_2EW.csv |
| 7 | 0.760110 | False | 0.036356 | 0.122324 | 0.163099 | 11_TL101294_2EW.csv |
| 8 | 1.140165 | False | 0.011906 | 0.203874 | 0.244648 | 11_TL101294_2EW.csv |
| 9 | 1.520220 | False | 0.006012 | 0.285423 | 0.326198 | 11_TL101294_2EW.csv |
| 10 | 1.900275 | False | 0.001638 | 0.366972 | 0.407747 | 11_TL101294_2EW.csv |
| 11 | 2.280329 | False | 0.001061 | 0.448522 | 0.489297 | 11_TL101294_2EW.csv |
| 12 | 2.660384 | True | 0.000667 | 0.530071 | 0.570846 | 11_TL101294_2EW.csv |
| 13 | 3.040439 | True | 0.000406 | 0.611621 | 0.652396 | 11_TL101294_2EW.csv |
| 14 | 0.380055 | False | 0.054995 | 0.040775 | 0.081549 | 97_CO140995NS.csv |
| 15 | 0.760110 | False | 0.036356 | 0.122324 | 0.163099 | 97_CO140995NS.csv |
| 16 | 1.140165 | False | 0.011906 | 0.203874 | 0.244648 | 97_CO140995NS.csv |
| 17 | 1.520220 | False | 0.006012 | 0.285423 | 0.326198 | 97_CO140995NS.csv |
| 18 | 1.900275 | False | 0.001638 | 0.366972 | 0.407747 | 97_CO140995NS.csv |
| 19 | 2.280329 | True | 0.001061 | 0.448522 | 0.489297 | 97_CO140995NS.csv |
| 20 | 2.660384 | True | 0.000667 | 0.530071 | 0.570846 | 97_CO140995NS.csv |
| 21 | 0.380055 | False | 0.054995 | 0.040775 | 0.081549 | 100_SCT190917.csv |
| 22 | 0.760110 | False | 0.036356 | 0.122324 | 0.163099 | 100_SCT190917.csv |
| 23 | 1.140165 | False | 0.011906 | 0.203874 | 0.244648 | 100_SCT190917.csv |
| 24 | 1.520220 | False | 0.006012 | 0.285423 | 0.326198 | 100_SCT190917.csv |
| 25 | 1.900275 | False | 0.001638 | 0.366972 | 0.407747 | 100_SCT190917.csv |
| 26 | 2.280329 | False | 0.001061 | 0.448522 | 0.489297 | 100_SCT190917.csv |
| 27 | 2.660384 | True | 0.000667 | 0.530071 | 0.570846 | 100_SCT190917.csv |
| 28 | 3.040439 | True | 0.000406 | 0.611621 | 0.652396 | 100_SCT190917.csv |
| 29 | 3.420494 | True | 0.000101 | 0.693170 | 0.733945 | 100_SCT190917.csv |
| 30 | 0.380055 | False | 0.054995 | 0.040775 | 0.081549 | 36_PE140995EW.csv |
| 31 | 0.760110 | False | 0.036356 | 0.122324 | 0.163099 | 36_PE140995EW.csv |
| 32 | 1.140165 | False | 0.011906 | 0.203874 | 0.244648 | 36_PE140995EW.csv |
| 33 | 1.520220 | False | 0.006012 | 0.285423 | 0.326198 | 36_PE140995EW.csv |
| 34 | 1.900275 | True | 0.001638 | 0.366972 | 0.407747 | 36_PE140995EW.csv |

Each row contains a pointer to the folder on disk where the results of that run are stored. These results are then subsequently used in the Loss Module.

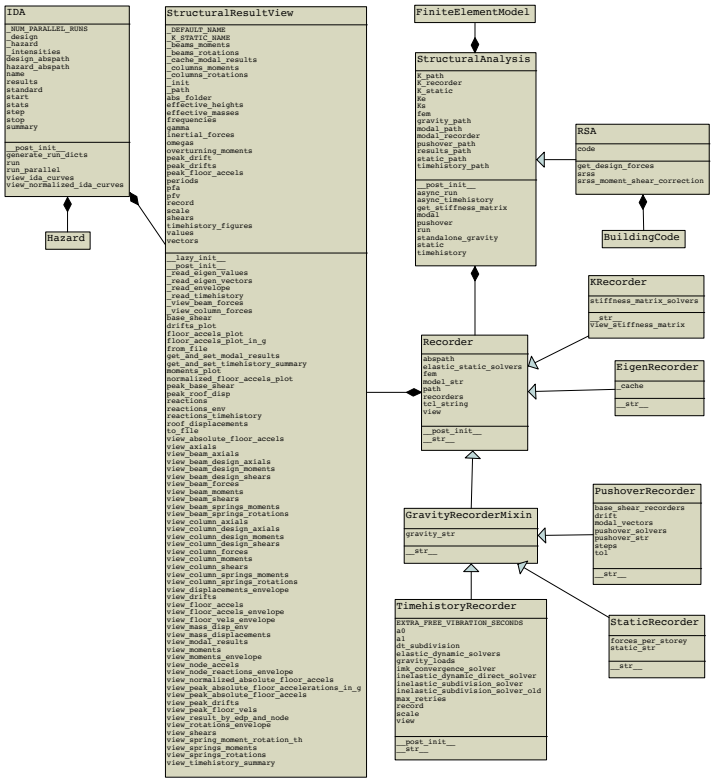



Figure 6.6: UML diagrams for the hazard module. Each row in a box represents an attribute of the class, notice 'StructuralAnalysis' is an interface implemented by 'RSA'.

# Chapter 7

# Loss module

## 7.1 Introduction

Loss is what a stakeholder suffers due to an event, by nature of having to repair or replace a damaged asset. Since the earthquake excitation is unknown, the loss suffered is foremost a random variable. Therefore, the responsibility of this module is to estimate the conditional probability of loss given an intensity, for the totality of the assets of a building. For this purpose, it uses both the results from the IDA, alongside the hazard curve, and the specification of the assets generated by the design module. The module must also provide disaggregation methods such that an analyst and the program can inspect and introspect the sources of loss, respectively.

In the literature, human health and business uptime are sometimes also considered assets. In general, these losses due to death and downtime require separate treatment from the dollar approach and will not be considered as such nor discussed further herein. Henceforth, all assets are physical objects located somewhere in or on the outside of a building.

## 7.2 Loss measures

Since we are interested in optimal design, an important measure to compute is the the average annual loss (AAL), which expresses the mean annual loss the building will suffer in its lifetime, and can be computed as the product of the expected loss for each event $k$ of the set and its corresponding annual frequency $f_k$:

$$\mathrm{AAL} = \sum_k L_k f_k \tag{7.1}$$

Another related measure is the variance of the annual loss (VAL):

$$\text{VAL} = \sum_i (L_i - \text{AAL})^2 / T$$

Where $i$ corresponds to the $i$-th *year* (and not the event), and $T$ is the total time simulated.

Another important measure is the *expected loss* for an asset, which expresses the mean value of loss for the next random event, and can be computed as follows.

$$\mathbb{E}[L] = \frac{1}{N} \sum_j L_j = \int \mathbb{E}[L \mid a]\, p(a) da \tag{7.2}$$

where $N$ is the number of simulations each producing a loss variate $L_j$, in practice this is easier than solving the integral on the right.

The total expected loss for the building is:

$$E[L_T] = \sum_i E_i[L]$$

Since we are generating independent random numbers for each asset $i$, and the expectation is linear, the total expected loss for the building is the sum of expected losses for each asset.

A good sanity check is to assert that AAL coincides with this :

$$\text{AAL} = \mathbb{E}[L] \frac{N}{T} = \mathbb{E}[L] \nu_0$$

Where $\nu_0$ is the rate-of-exceedance for the base intensity $a_0$.

Another related measure that is perhaps less used, is the variance of loss for the building. To compute this, we need to ask; what is the covariance between component losses? this is a complex problem. The reader is referred to [Aslani and Miranda, 2005] for more details. For our purposes, its value is not required.

Finally, regardless of the procedures used, the loss measure we are most interested in is the annual rate-of- exceedance-of-loss curve (see Fig.7.1). This curve tells us how frequently a given level of loss is exceeded, and contains all the information to compute many other loss measures, such as the average annual loss (AAL) or the expected loss.

This curve is obtained in the following manner;

1. Fix a level of loss, say: $L = l$.
2. For all records and all intensities, sum the corresponding frequency of events which caused losses greater than or equal to $l$, this is the rate of exceedance of that level of loss $\nu(l)$.
3. Repeat this procedure for all levels of loss from 0 to the total building cost.

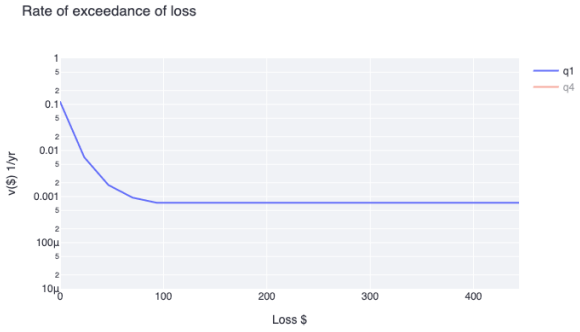



Figure 7.1: Example annual rate-of-exceedance-of-loss curve for a 4-storey reinforced concrete building with a total cost of 450k dollars.

## 7.3 Loss computation for assets

### 7.3.1 Risk objects

A risk object must produce a loss variate given an event, which is a 'record, intensity' pair. Each asset must implement a risk object and specify how to produce this loss value, this will depend mostly on the category of the asset. Therefore, the complete response of the asset is needed, this means that, if possible, the analysis kernel should capture the set of all DOF time-histories of displacements, drifts, rotations, moments, etc. These are called engineering-demand-parameters or EDPs for short, and are also called field or response variables $u$.

Therefore a risk object implements the following mapping:

```
Risk: EDPs ↦ Dollar | Downtime | Death
u ↦ dollars
```

### 7.3.2 Loss computation for non-structural assets and contents

#### Fragility and vulnerability objects

The uncertainty in the loss is modeled in two phases: first, determine the damage state the asset falls into, this is done by specifying the probability of incurring in any damage state given an intensity called the 'fragility function'. Afterward, the conditional probability of exceeding a loss given that the model is in damage state 'ds' is used to compute the loss, this is usually termed the 'vulnerability' of the object [FEMA, 2012a] [FEMA, 2012b].

The algorithm to compute the loss of this risk object consists of implementing these two steps; first, the damage state of an asset for a run is obtained as follows:

1. Select an EDP obtained from the structural analysis run to which the asset is sensible such as peak drifts.

2. Generate a random number $r$ between 0 and 1, and enter the ordinates of the fragility curves with a horizontal line at $r$.

3. Meet the value of the EDP and select the first curve above $r$, this is the damage state of the asset.

Then, to determine the loss suffered, conditional on a damage state; simulate a random variate via the inverse method using the vulnerability curve cumulative distribution function (CDF) that matches the selected damage state [Benjamin and Cornell, 1970].

As an example application of this algorithm, consider a drift-sensitive asset (reinforced concrete column) with 7 damage states (see Fig.7.2). We simulate a random number between 0 and 1; $r = 0.66$, and we enter the ordinates with a 'right arrow' on that value, considering the peak drift for the story for this event was $\gamma = 0.029$, we enter the abscissas with an 'up-arrow' on that value.

Afterward, the first curve that the 'up-arrow' touches after touching the 'right-arrow' in this case is the purple curve, therefore the asset is in a damage state DS3 (cracking). Finally, given that the asset is in damage state DS3, we simulate another random number between 0 and 1; $r = 0.89$, and follow the right-arrow until we touch the purple curve, then read from the abscissa a loss of 0.71, which means a 71% percent loss of the total asset cost.

This scheme has some flexibility since a deterministic event can be modeled with two lognormal distributions with uncertainty 0, this implies that for any event, it is always damaged in damage state $ds_0$, and it always costs a certain amount, say $\beta_0$, to repair.

### 7.3.3 Loss for structural elements

Loss computation for structural elements can be implemented in the same manner as the other asset categories. That strategy, however, estimates loss without attending to the specific mode or cause. Strength degradation occurs when existing microcracks due to initial imperfections propagate into macrocracks due to the combination of external loads and structural imperfections. This, in turn, leads to cracking, buckling, yielding of steel reinforcement, crushing of concrete, and ultimately, collapse. [D. A. Makhloof, 2021].

One disadvantage of the probabilistic approach is the impossibility to know by which mechanism the damage originated. Another disadvantage is that the empirical procedure is ultimately based on simulation and has some amount of variance. Furthermore, it uses just a single value *e.g.* maximum absolute drift, which is too simplistic, while there is much more information regarding the time-history response of the member which could give the analyst more insight into the sources of loss, and ultimately, guide the design procedure towards an optimum.

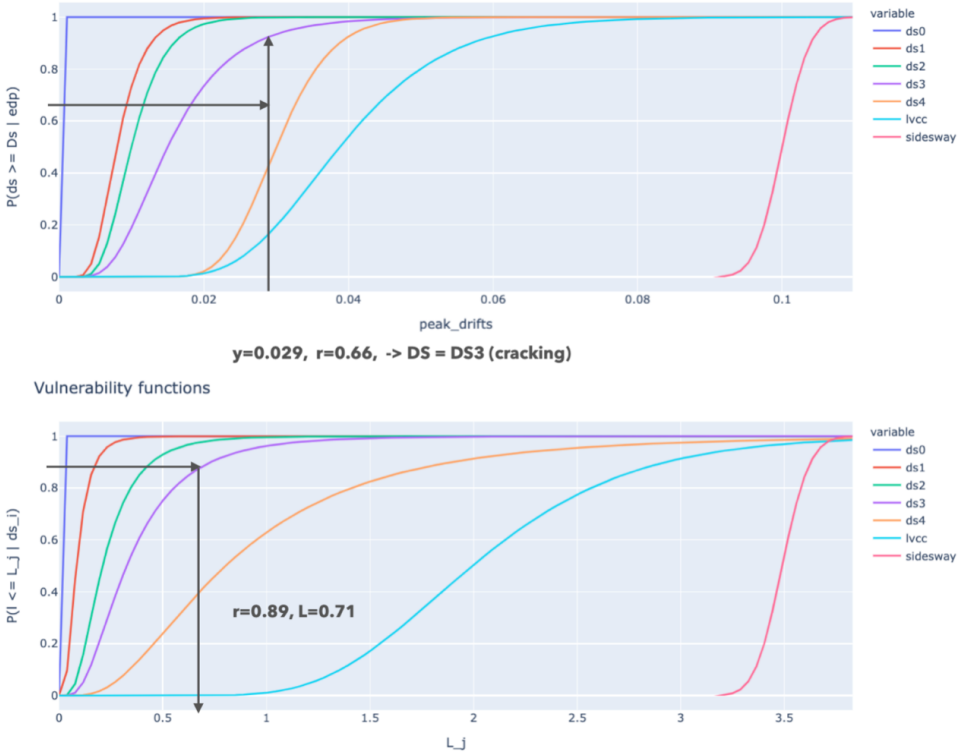



Figure 7.2: Fragility (top traces) and vulnerability (bottom traces) curves for a drift-sensitive asset (reinforced concrete column) with 7 damage states. The EDP of this asset is peak drifts (top figure abscissa). Loss is given as a fraction of the total cost (bottom figure abscissa).

The main contribution of this module is to show how to compute the direct loss for structural elements based on physical principles. We propose a method for reinforced concrete beams and columns based on a modification of the energy dissipation capacity index proposed by Park and Ang. This method is deterministic and is rooted in physical principles. This damage index is a continuous measure from 0 to 1, where 0 indicates no damage and 1 indicates total local collapse.

**Physical approach towards a continuous damage index for axial-flexural behavior**

The capacity of a beam-column element must consider three coupled phenomena: axial load, moment capacity, and shear capacity. If the column has enough detailing to develop full flexural ductility, shear capacity can be decoupled from the axial-flexural response both in the structural analysis stage as well as the

loss computation stage. In this spirit, building codes usually specify minimum reinforcement guidelines to avoid brittle failure which makes the flexurally dominant model valid and useful. One could propose residual strength, dissipated energy, or ductility as measures of capacity and therefore as indicators of local damage.

Makhloof [D. A. Makhloof, 2021] presents a state-of-the-art review on concrete damage indices; most of these approaches are based on the Park and Ang damage index, while others are based on more elaborate cycle-counting and cross-counting techniques. Coincidentally, Medina [Ibarra et al., 2005] mentions that the IMK damage model deterioration is equivalent to Park and Ang's.

Therefore, hysteretic response and energy dissipation is the most natural damage index to use as a base model. This criterion postulates that any member possesses a reference inherent hysteretic energy dissipation capacity independent of loading history and that the piece can dissipate a certain amount of half cycles worth of energy before it collapses from degradation.

According to this hypothesis, the energy capacity is:

$$E_t = \Lambda M_y \theta_p = 1/\beta_{PA} M_y \theta_p \tag{7.3}$$

Where $M_y, \theta_p$ is the yield moment and plastic rotation, respectively.

This means the member can dissipate $1/\beta_{PA}$ half cycles worth of energy before it collapses from cyclic degradation only.

It has been observed [Park and Ang, 1985] [Kunnath et al., 1997] that while cyclic degradation is important, a more sound approach is to consider a linear combination of cyclic capacity and monotonic ductility:

$$\mathrm{DI_{Ku}} = \frac{\max(\theta - \theta_y,\ 0)}{\theta_u - \theta_y} + \beta_{\mathrm{PA}} \frac{\int dE}{M_y \theta_p} \tag{7.4}$$

Where $\beta_{\mathrm{PA}}$ is the classic Park and Ang damage index, $\theta_u, \theta_y$ are the ultimate and yield rotations, respectively, and $\theta$ is the maximum absolute rotation obtained during the analysis.

The first term in 7.4 is related to the ratio of the maximum rotation to the ductility achieved by the section during the loading cycles. It can be inferred that a ductile member will have a bigger denominator $\theta_u - \theta_y$, and therefore less damage for a fixed $\theta$. The second term is based on the hysteretic energy dissipated proportionally to the inherent energy capacity. Another way to rewrite this expression is as $\int dE/E_t$ , that is, the proportion of plastic deformation energy with respect to the total energy capacity.

This model states that failure corresponds to $\mathrm{DI_{Ku}} \geq 1$, therefore the damage index is:

$$\mathrm{DI} = \min\left(1,\ \mathrm{DI_{Ku}}\right) \tag{7.5}$$

Jiang *et.al.* [Jiang et al., 2011] give calibrated empirical expressions to estimate $\beta_{PA}$ as functions of the member geometric and material properties plus the mean axial load sustained throughout the loading:

$$\beta_{PA} = (0.023\ L/h + 3.352 n_0^{2.35}) 0.818^{100\alpha\rho_w f_{yw}/f'_c} + 0.039 \tag{7.6}$$

where $\alpha_s$ is the confinement factor dictated by EuroCode2004, $\rho_w$ is the stirrup ratio, $f_{yw}$ is the stirrup yield strength and $n_0$ is the axial load ratio. Other important member properties such as $\theta_y, \theta_u$ can be estimated with the semi-empirical formulas of [Panagiotakos and Fardis, 2001].

One notes that once a member has been designed, its properties are fixed. Therefore, for any hysteretic response, the damage index and loss can be readily computed.

An example of the application of this criterion can be seen in (Fig.7.3), the top left trace just barely entered the nonlinear behavior and has DI=0.03, while the bottom right trace has exhausted most of its cyclical capacity, and coupled with its monotonic nonlinear behavior achieves a DI of 1 *i.e.* failure and total loss (Fig.7.3 bottom right trace).

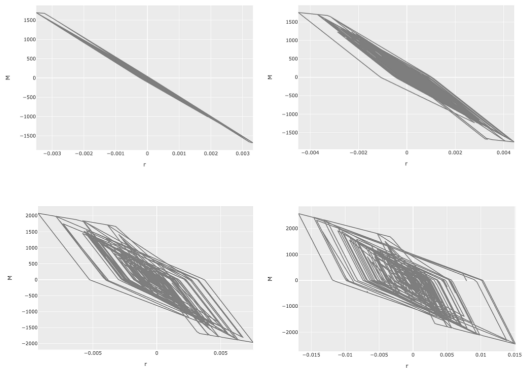


Figure 7.3: Damage progression in a reinforced-concrete column joint (chord rotation vs moment), loss is computed directly from the physical hysteresis observed, that is, a combination of the energy dissipated during the loading cycles and the ductility demand.

This method can be improved with the following observation; while a member suffers damage during an event, the total loss suffered is also a function of the repair strategy. Therefore the functional relationship between the damage index and loss is a map from this continuous index to piecewise constant step functions in the loss space. This is consistent with how a repair engineer would act in practice. Such repair strategies are not implemented in this study, rather, the damage index will directly be translated into losses by directly computing $L = \text{DI} \cdot \beta$ on any given run. This is the simplest strategy which simulates restoring the member immediately to its initial state.

## 7.4 The role of collapse in loss estimation

Although earthquakes whose intensities can completely collapse a building occur infrequently, the loss suffered by the stakeholders is so high that the contribution to the mean annual loss is non-negligible. This is shown schematically in figure 7.4, where the contribution of collapse dominates the higher levels of loss, in such a way that essentially all the low rates of exceedance have an almost complete contribution from the collapse case [Aslani and Miranda, 2005].

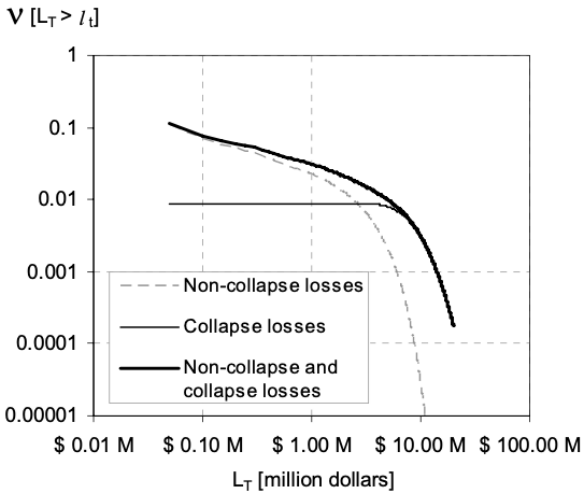



Figure 7.4: Mean annual frequency of exceedance of loss curve disaggregation into the non-collapse and collapse cases [Aslani and Miranda, 2005]

Furthermore, the contribution of collapse towards the expected annual loss can range from 10% to over 55% [Aslani and Miranda, 2005]. Therefore, collapse must be considered explicitly. This is arguably the hardest and most important structural phenomenon to take into account for a realistic loss computation.

Global building collapse can be total or partial, the latter happens when a part of the building must be demolished, such as the collapse of the upper stories which has been historically observed. Although the cases where the stakeholders decide to refurbish these stories instead of demolishing the building are rare, they must nevertheless be considered. On the other hand, total collapse can happen in two main ways; side-sway due to dynamic instability [**?**] loss of vertical carrying capacity. However, the causes of these two types of collapse are numerous and the uncertainties are very large [Haselton and Deierlein, 2007].

Other scenarios are equivalent to collapse in the loss that they produce, even though the building did not technically collapse. This happens when the building must be demolished due to excessive foundation settlement, excessive residual drifts, and some forms of loss of vertical carrying capacity, among many

others. These do not usually occur *during* the event but are considered after.

There are two main ways of considering collapse in the loss computation pipeline;

1. intra-analysis collapse
2. extra-analysis collapse via postprocessing

The program should provide a way to include and exhaust most of possibilites, this is the only way to achieve realistic loss estimations.

### 7.4.1 Estimation of intra-analysis collapse losses

Sidesway collapse is associated with the intra-analysis condition of dynamic instability. This can be detected numerically when the instantaneous stiffness matrix is not positive definite. In turn, this condition manifests itself via an in-cycle degradation of the load-displacement curve of the system, in other words, in the presence of a 'negative stiffness' branch [FEMA, 2009] [López, 2015].

Structural models that implement progressive degradation of strengths and stiffnesses of its elements are prone to exhibit this 'negative-stiffness' branch. Well-calibrated IMK-based models [Ibarra et al., 2005] allow realistic transitions from the undamaged stage to the deteriorated stage, up to dynamic instability which is representative of side-sway collapse [**?**]. P-Delta effects, also called second-order geometric effects, increase the moment-demand due to the action of a vertical load $P$ acting at a distance $\Delta$ over the static equilibrium configuration of the structure. This phenomenon can induce or amplify the 'negative-stiffness', thus inducing or accelerating side-sway collapse [López, 2015]. Therefore it is paramount that the model can incorporate these effects in the structural analysis stage as well.

On the other hand, some refined models with state-of-the-art convergence algorithms allow simulating collapse because of loss of vertical carrying capacity or due to progressive lateral failures [Scott and Fenves, 2010]. For our purposes, we consider side-sway collapse as the only shear-flexural collapse mechanism. This can be detected intra-analysis when the instantaneous displacement of any lateral DOF is unrealistically large *i.e.* $10^5$. For framed structures, a good rule of thumb is to consider side-sway collapse to occur when any interstorey drift exceeds 0.1 or perhaps as big as 0.2.

We show an IDA analysis script that detects side-sway collapse and subsequently writes information about the phenomenon to a file on disk, including metadata about the run;

```
set tol 1e-6
constraints Transformation
numberer RCM
integrator TRBDF2
system FullGeneral
analysis Transient
test NormDispIncr $tol 1000 0;

set algorithms {
 "Newton"
 "KrylovNewton"
 "BFGS"
```

```
 "Broyden"
}

set num_algorithms [llength $algorithms]
set num_subdivisions 10
set max_subdivisions 1 ;

set results_file $results_dir/results.csv
set collapse_file $results_dir/collapse.csv
set fp [open $results_file w+]

set aix 0
set break_outer 0
set converged 0
set time [getTime]
while {$time <= $duration && !$break_outer} {
    set time [getTime]
    set analysis_dt $record_dt
    set algorithm [lindex $algorithms $aix]
    algorithm $algorithm
    set converged [analyze 1 $analysis_dt]
    set subdivision_retries 0
    while {$converged != 0} {
        incr subdivision_retries
        if {$subdivision_retries >= $max_subdivisions} {
            set time [getTime]
            puts $fp "$algorithm did not converge at time $time $converged $subdivision_retries"
            incr aix
            if {$aix >= $num_algorithms} {
                set cf [open $collapse_file w+]
                puts $fp "FAILURE, couldn't converge with any algorithm or reduced step"
                puts $fp "FAILURE, couldn't converge with any algorithm or reduced step"
                puts $cf "FAILURE, couldn't converge with any algorithm or reduced step"
                puts $cf "time $time"
                # puts $fp "max drift $time"
                set break_outer 1
                close $cf
            }
            break
        }
        set n [expr {$num_subdivisions ** $subdivision_retries}]
        set analysis_dt [expr {$record_dt/$n}]
        set time [getTime]
        set converged [analyze $n $analysis_dt]
        puts $converged
        if {$converged == 0} {set aix 0};
    }
}

close $fp
```

We consider side-sway collapse to induce a unitary ($\beta = 1$) loss, that is, the complete cost of the building. Although it would be possible to incur on greater losses due to demolition costs, this is not considered herein.

### 7.4.2 Estimation of out-of-analysis collapse losses

The model's inability to capture every single failure mode must be compensated by a post-processing stage. This means that, even though the model did not strictly collapse under the earthquake excitation, a real structure would have produced more loss than the raw analysis data would suggest.

These extra failure modes include phenomena such as excessive residual drift, excessive and unrepairable damage across floors, excessive foundation settlement or plumb, axial or shear failure of the columns, partial floor collapses, among others. Interestingly, some conditions such as excessive residual drifts will cause the demolition of the building, however, most contents and non-structural assets could potentially survive, and therefore the loss suffered would not be monetarily equivalent to a total collapse.

### Shear failure of reinforced concrete columns

As mentioned, shear and moment-axial behavior are usually decoupled in the structural analysis model, thus, it is not possible to detect pure shear collapse intra-analysis. Therefore one must check the shear failure of columns in a post-processing stage.

However, the spirit of all modern Building Codes is to move away from a brittle failure mode (such as shear-failure) towards more ductile failure modes such as a purely flexural or combined shear-flexure mode which is primarily characterized by concrete crushing, reinforcing bar buckling, bar fracture, and bond failure, among others. The recommendations for rebar reinforcement ratios, placements and detailing of all modern Building Codes, coupled with the strong-column/weak-beam design principles intend to forbid a purely shear failure mode, even in scenarios where the intensity is much greater than the design intensity. Furthermore, for designs with a force-reduction factor of 1 or close to 1, the spirit of the strong-column/weak-beam criterion is to assert that beams yield before columns do, that is, that the plastic collapse mechanism is controlled by the beams and not the columns.

The design procedure should reflect these recommendations and uphold the principles of ductile design. Although with force-reduction factors close to 1, it is not strictly necessary to do a capacity check, we assert that the rebar percentage and spacing of the stirrups are such that the base column capacities $V_R$ are greater than the pushover base shear, even during the post-yield stage. This would give the designer confidence that the resulting model upholds the desired design principles.

A more stringent "pseudo-capacity" condition would check that the design produces a structure such that in a pushover, any beam would yield before any column reaches the onset of rebar stress, this means that plastic hinges start to form in the beams before the column shear forces reach the concrete shear capacity $V_{CR} = 0.5\sqrt{f'_c}$. If this pseudo-capacity condition is met, no scenario would lead to a purely shear building collapse, this would save us the task of explicitly including checks at every instant. However, if this condition is not fulfilled, then in general we would have to check the capacity of all columns against the instantaneous shear demands. Unfortunately, it is unclear whether a single or multiple columns failing brittely in shear would constitute a total building collapse. This is an inherent limitation of flexural spring models.

## 7.5 Disaggregation of loss

A fundamental objective of this module is to understand the relative importance of the different sources of loss. A useful tool for this purpose is the so-called 'disaggregation plots' which allow the analyst or stakeholders to understand immediately where loss is coming from. This is highly important since the decision to build, refurbish, or retrofit a building should be based on risk information, such as whether collapse contributes significantly towards the expected annual

loss, or whether a story is incorrectly designed and causing excessive losses in its contents or structural elements, or to improve the placements of the assets, etc.

As an example of loss disaggregation, we show an expected-loss vs intensity plot broken down into each asset category (see figure 7.5).

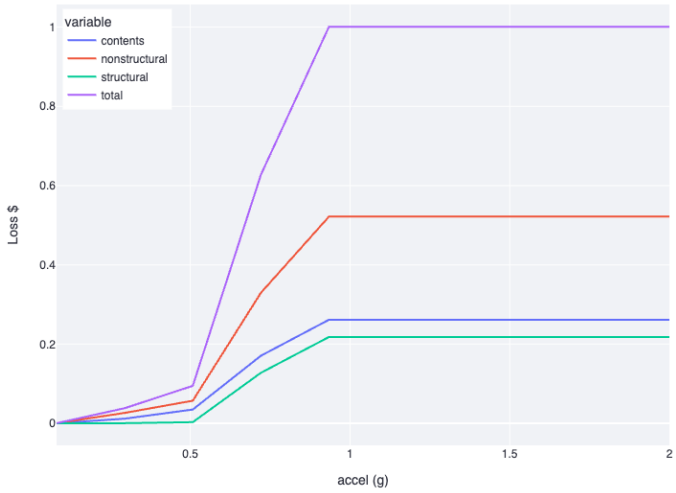



Figure 7.5: Expected-loss vs intensity disaggregation plot by asset category.

In this example, the gross loss is dominated by the non-structural assets. Note that the values are not normalized with respect to their initial cost, and therefore the initial cost distribution affects the overall disaggregation curves. As another example, figure 7.4 presents a rate-of-exceedance of loss curve disaggregated by collapse cases.

The loss module should provide facilities to perform loss disaggregation by asset name, asset category, floor, and collapse cases. It is interesting to take this a step further, and perform a second disaggregation phase to understand the relative cross-contributions of the different sources of loss. This second round of disaggregation can be shown in a matrix-style plot called a 'heatmap', wherein the brightest colors being highest contributions to the overall loss.

For example; figure 7.6 presents a category vs floor double disaggregation heatmap.

We can see that in this example, the nonstructural assets of the third floor contribute the highest to the overall loss, while structural beams on the rooftop contribute the lowest. This could perhaps be understood intuitively if the non-structural assets are acceleration-sensitive, since the upper floors tend to have the highest accelerations. While the upper floors also tend to have low drifts, thus the upper beams would contribute little to the total loss.

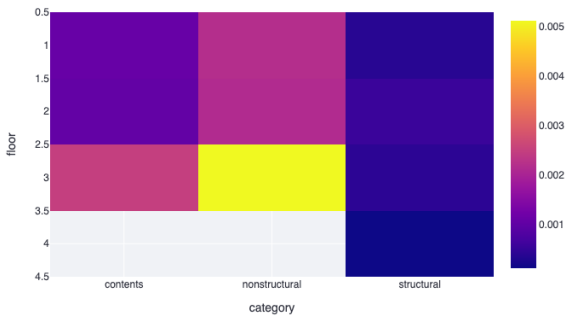



Figure 7.6: Heatmap disaggregation plot of category vs floor relative cross contributions.

## 7.6 Responsibility and objective of the module

The main responsibility of this module is to compute the loss suffered by the stakeholders of a building, given an **StructuralAnalysisResult**. It must provide filtering and disaggregation methods for the analyst, including filtering or disaggregating by asset category, name, floor, etc.

### UML diagrams and code structure

In the PlantUML notation, classes are specified by rectangles with their name at the top. Their properties (called attributes) are shown below their name, while their most relevant methods (own functions) are listed at the bottom third of the rectangle.

Relationships between classes are given as lines, the nature of this relationship is specified by the arrowhead type:

1. △ represents implementations of interfaces.
2. ▲ represents a relationship of composition or being contained by the source class.

Fig.7.7 shows the main objects in the module. The reader is invited to consult the public source code for the complete implementation.

### Loss interface

The main class of this module is an interface (ABC), which in essence is a container for loss statistics such as the AAL, the expected loss, the rate of exceedance of loss, etc. For a given structural analysis run, each asset implements the **Asset** interface and knows how to compute its loss and produce a **Loss** instance.

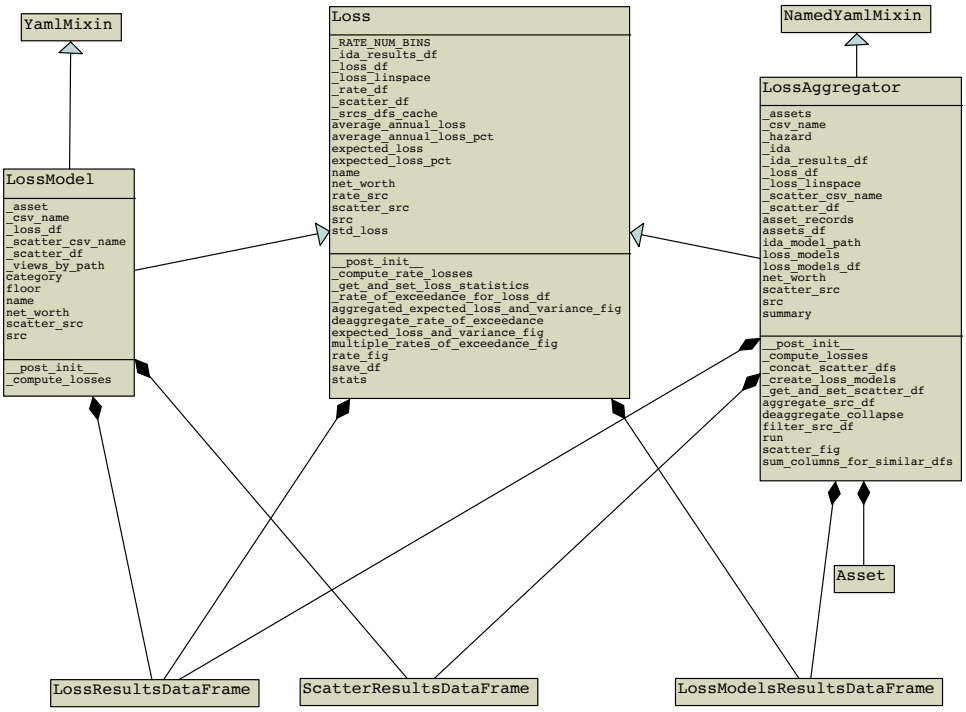



Figure 7.7: UML diagrams for the hazard module. Each row in a box represents an attribute of the class.

This class must be able to retrieve and process the information needed to produce the figures such as Fig.7.5 or Fig.7.6.

This interface is implemented both by single assets (**LossModel**) as well as the complete building (**LossAggregator**). The main methods are:

1. **_get_and_set_loss_statistics()**
2. **expected_loss_and_variance_fig()**
3. **rate_fig()**
4. **compute_rate_losses()**

## Asset object

Interface that maps a structural analysis run to dollars lost.

Uses a **IDAResultsDataFrame** and *can* but need not use **StructuralResultViews** internally, which contains all the information for the given run. For example, we can specify a given EDP such as rotations envelope or moments

envelope, and the asset will figure out which results file it needs to load from disk to compute its loss, based on its properties such as DOF, floor, etc.

**RiskAsset** implements the **Asset** interface, and uses the strategy discussed in section 7.3.2 to compute its loss.

**IMKSpring**, which represents a node in a reinforced concrete element implements the **Asset** interface, and uses the strategy shown in section 7.3.3 to compute its loss.

## LossModel

Its purpose is to hold a reference between an **Asset** to a run (**IDAResultsDataFrame**)

Its main method is **compute_losses()** which produces a **LossResultsDataFrame**. It also produces a **ScatterResultsDataFrame**, which we will discuss in the next section.

## LossAggregator

Represents the aggregate loss of the assets *i.e.* the building. Has methods to summarize and aggregate the loss of the different **LossModel**s, and also to create and manage them.

## Loss results

The author recommends summarizing the results of each run as flat as possible, such as tables in column-major fashion. The correct abstraction for is known as a *dataframe*. This allows us to think of **Loss** instances as 'record' data structures, and think in terms of data transformations more easily, such as filters, concatenations and pivots, etc, divorced from the object-oriented paradigm.

### LossModelsResultsDataFrame

As its name implies, it is only the set of **LossModel**s turned into a table.

### ScatterResultsDataFrame

ScatterResultsDataFrame: contains the most info; which is one row per run per asset.

Columns include: Sa/Say_design, collapse, freq, inf, intensity, intensity_str, path, peak_drift, peak_drift/drift_yield , pfa, pfv, record, sup, losses, loss, asset name, asset category, asset floor

### LossResultsDataFrame

Summarized acceleration vs loss values, a 'pivoted' version of the previous dataframe.

# Chapter 8

# Case study

We study the influence of the design procedure on the cost function (Eq.1.1) of a public office building designed using two different force reduction coefficients from the 2023 Mexican code [Gob.CDMX, 2023c] [Gob.CDMX, 2023a] [Gob.CDMX, 2023b]. This code uses the modal spectral design method, using force reduction factor $Q'$, which reduces the design lateral force which is given by the following expression:

$$V = c_s \frac{W}{R'Q'}$$

Where $c_s$ is the seismic coefficient, $W$ is the weight of the building, $R'$ is an overstrength factor while $Q'$ is the force-reduction factor that is proportional to a reference theoretical ductility factor $Q$.

## 8.1 Building configuration and earthquake hazard

With this in mind, we have selected as case studies the values $Q' = 1$ and $Q' = 4$, holding everything else in the design process constant. These designs are henceforth called Q1 and Q4, correspondingly. Intuitively, Q1 is a stronger and costlier design, while Q4 should have more ductile capacity at the expense of some strength. It is worth noting that while it is possible to use the Mexican Code to design a reinforced concrete building with a force factor equal to $Q' = 1$ tied to the theoretical ductility of 1, the percentage of minimum longitudinal and transversal rebar and the detailing aimed to prevent a brittle shear failure will increase the global ductility and overstrength of the designed building significantly.

The seismic hazard for the buildings planned to be built in the soft-soil lakebed zone (zone IIIB) and with a natural period of 1.10 s was computed with CRISIS [Ordaz et al., 2021], see Fig.8.1. A set of 50 representative strong motion records tied to the presented seismic hazard was selected.

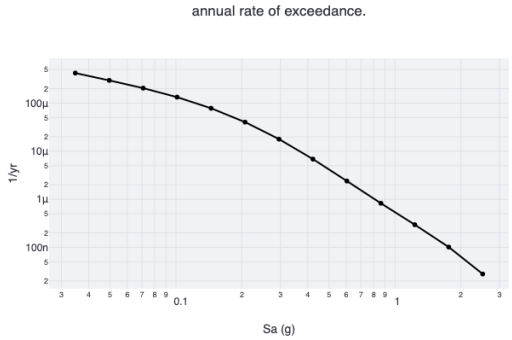



Figure 8.1: Rate of exceedance curve for $T = 1.10$s used for the sample buildings, which is representative of the soft lakebed soil zone in Mexico City.

The geometrical configuration of the building consists of 3 stories with heights 5, 4 and 4m respectively and two bays of equal widths of 8m. The fundamental period of this building is 1.10 s.

The distribution of assets follows empirical data of public offices commonly found in Mexico City [Reinoso and Jaimes, 2013].

Figure 8.2: Geometrical configuration of the framed RC building, with a schematic representation of its assets.

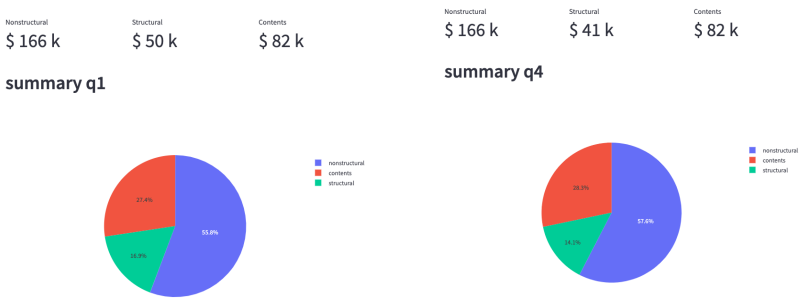
### 8.1.1 Asset cost distribution

Nonstructural and content costs are numerically equal, while structural cost increased from \$41k USD to \$50k USD, going from Q4 to Q1, which is a 18% relative increment in structural costs and only a 3% in total cost.

Due to the change in the strength of the building between Q1 and Q4, the former costs around 10% more than the latter in structural costs, owing to the

relative increase in longitudinal vs transversal steel required for Q1 and Q4 respectively.

Figure 8.3 shows the distribution of costs as functions of the asset categories.




(a) Q1 asset cost distribution.

(b) Q2 asset cost distribution.

Figure 8.3: Asset cost distribution (in categories: blue-nonstructural red-contents green-structural) for Q1 and Q4, while nonstructural and contents remain the same, Q4 costs $9k less than Q1.

### 8.1.2 Pushover curves

The first mode static pushover of both buildings is shown in Fig.8.4. Interestingly, even though Q1 was designed with a theoretical ductility of 1, the actual ductility developed in a static pushover analysis is around 2, while for the building Q4, the global ductile capacity was around 10. This indicates that in general, it seems hard to build a real structure with no reserve ductility using a codified design. This is in part due to the minimum reinforcement requirements specified in the codified design, and a consequence of the fact that most non-trivial structures contain some reserve strength and ductility that are not present in single-degree-of-freedom systems.

### 8.1.3 IDA curves

An IDA was performed with the 50 selected records at increasing intensities. Figure 8.5 shows the median normalized IDA curves for both buildings. The abscissas are normalized with respect to the approximate yield drift taken from the first-mode pushover. The median intensities are normalized with respect to their design intensity, which implies that the ordinates represent a sort of "dynamic overstrength ratio". In this case study, the maximum overstrength ratio for Q1 is about 1.8, while for Q4 it is about 4. Furthermore, we can observe that the curves are relatively similar in their shape; this highlights the fact that in this case study, only trading strength for ductility did not intrinsically change the median dynamical behavior of the building.

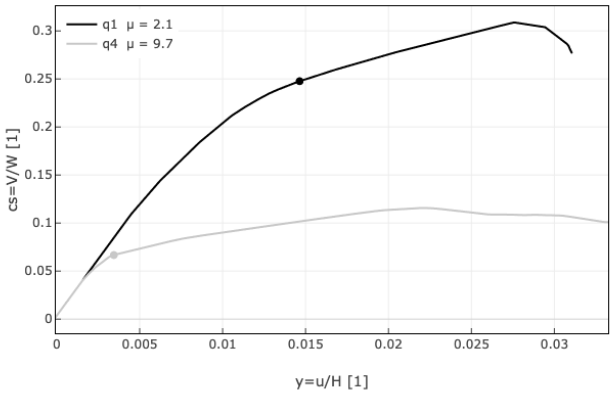



Figure 8.4: Normalized first-mode pushover curves for Q1 (grey) and Q4 (black). The ordinates are normalized with respect to the total weight (and are therefore numerically equal to the seismic coefficient) and the abscissas are normalized with respect to the height i.e. total drift.

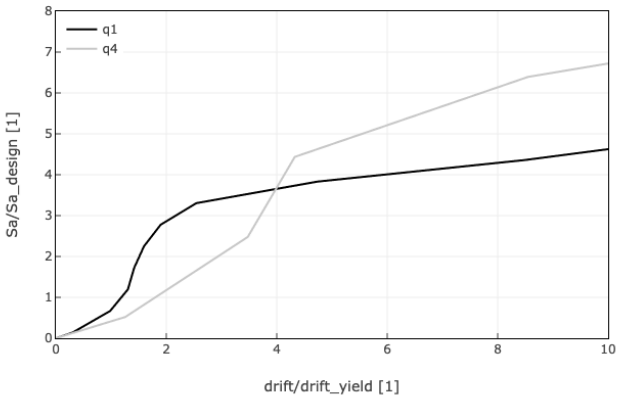



Figure 8.5: Normalized mean IDA curves for Q1(black) and Q4 (grey). The ordinates are normalized with respect to the design $S_a$ while the abscissas are normalized to their corresponding approximate drift at yield computed from the static pushover.

### 8.1.4 Rate of exceedance of loss

We computed the annual rate of exceedance of loss for both buildings, see Fig.8.6; we can observe that Q1 is always below Q4, which means that on average for a fixed loss, Q1 suffers that loss less frequently than Q4; this makes sense as Q1 has more strength.

Notice that the curves flatline after a level of loss, this means that after say $L_0$, a bigger loss happens as often as $L_0$. This is in part a consequence of the assumption in the model that there are no partial collapses. Furthermore, the loss suffered by non-rugged assets in a run without collapse is relatively small with respect to the total cost of the building. This leads to rugged assets and the collapse cases dominating the loss computation after some intensity threshold is crossed, which would indicate that the building configuration and modeling assumptions lead to a sensitiveness to collapse in its loss computation.

Upon seeing those results, it would intuitively seem that Q1 is a better design than Q4, as the latter does not seem to show any advantages so far. It is worth noting however, that the AAL of Q1 is 3k USD, while the AAL of Q4 is 5k USD, this means that although Q1 is more expensive to build, it suffers less future losses than Q4, which is cheaper to build today. This agrees with experience.

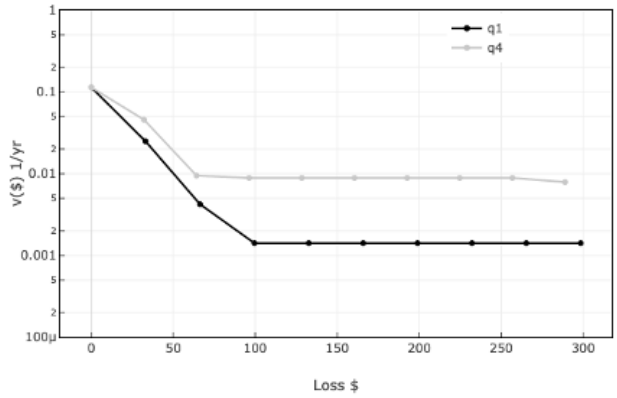



Figure 8.6: Annual rate of exceedance of loss curves for Q1 (black) and Q4 (grey). Note that for all levels of loss, Q1 is below Q4, this implies that it suffers the same loss less frequently.

### 8.1.5 Expected present-day cost comparison

Finally, we present a comparison of the expected present-day cost of constructing Q1 and Q4 as a function of $\gamma$ the monetary discount factor (Fig.8.7).

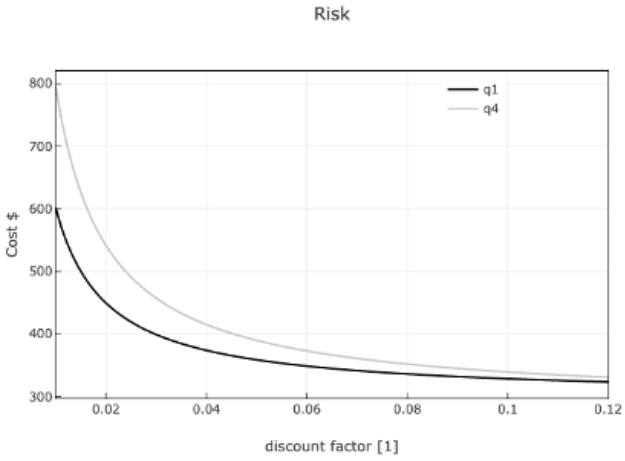



Figure 8.7: Expected present-day cost of constructing Q1 (black) and Q4 (grey) as functions of $\gamma$, the monetary discount factor.

## Observations

Using the interpretation of $\gamma$, the results shown allow us to infer something very interesting: if this factor is low, this means that future losses really matter and so the stronger design would be the better design choice; Conversely, when $\gamma$ is high we care little about future losses, in which case the design with the lowest initial cost tends to win; thus, Q4 would intuitively be a better design choice.

This means that trying to find the better design option from the first mode pushover, median IDA curves, and rate of exceedance of loss curves is perhaps not correct.

In our study case, we observe that the curves will eventually cross when $\gamma$ is very high, which means that *eventually* Q4 will be the better design. However, the $\gamma$ values for which this would happen are not realistic. Therefore, Q1 is the better design choice here, owing to its relatively small increase in cost with respect to big gains in capacity and dynamic stability.

As seen, this tool allows for a quick comparison of design choices; we could therefore perform a search in the space of all reinforced-concrete designs, fixing the layout and element dimensions but varying the amount of longitudinal and lateral steel.

To find the optimum building, we could introduce an optimization algorithm (gradient or gradient-free methods) while fixing the layout and occupancy class. This new design would be associated with a different distribution of forces and theoretical force reduction factors $Q'$. This evidences that the force reduction factor chosen a-priori is not highly important, which implies that we need to shift away from such simple rules towards searching for effective design algo-

rithms. One such algorithm, based on classical ideas by [Wilson, 2002], would be a multi-objective design procedure that attempts to minimize steel cost and static strain energy simultaneously during a first-mode pushover or due to modal spectral forces. Perhaps there are design algorithms that iteratively produce the optimum design starting from a simple base design *i.e.* fixed-point methods. This would be a powerful discovery.

# Chapter 9

# Conclusions

The bases for a conceptual framework that allows the designer to achieve a more rational, risk-informed, and hopefully optimal design of structures were presented. This framework was implemented in a program that allows the engineer to perform structural design and analysis quickly, and therefore choose the best options among many building designs.

The author believes this tool is useful for both research and practice as it allows a quick comparison of many design alternatives by automating and abstracting away the difficult and computationally expensive processes such as structural design, analysis, and asset loss estimation.

Using this implementation, the influence of the design procedure on the structural risk was studied, wherein the expected present cost of undertaking the construction (Eq.1.1) was shown as a function of the monetary discount factor.

This comparison was performed between two buildings, where the selection of a force reduction factor $Q$ was shown to influence the initial costs and future losses suffered directly. The main finding of this research so far is that for the selection of the optimum building, it is not enough to look at measures of behavior and structural performance such as pushover or median IDA curves, nor to base our criterion on the rate of exceedance of loss curve, or any other loss measures on their own. Rather, Eq.1.1 must be analyzed for the building options at hand.

The study case showed that the optimum design was the one with higher strength for all realistic levels of $\gamma$. This is owed to the fact that for this case, the increase in strength imparted the structure with considerable gains in capacity and dynamical stability even if the initial cost was somewhat higher.

Preliminary results from this investigation suggest that both a realistic estimation of the monetary discount factor alongside a realistic asset cost model are crucial for arriving at the optimum design.

While formalizing structural design as an algorithmic process via the "object-oriented" programming approach, some difficulties and inconsistencies were found. As an example; designing structural elements for different force reduc-

tion factors $Q$ produces a scenario where the structural element needs to know what the 'outside' force-reduction factor is and adjust its transversal steel ratio and separation accordingly. Somehow the elements are now themselves $Q = 1$ or $Q = 4$ elements, which is strange. Moreover, due to the spring models used, this increase in transversal reinforcement ratio from $Q = 1$ to $Q = 4$ is not fully reflected in the ductility of the building, this is because the empirical equations that dictate the member's ductility only consider this ratio indirectly. This is inconsistent with our idea of the design force-reduction factor tied directly to a theoretical ductility and indicates that perhaps this design procedure is not very efficient.

The present value cost of future losses is highly sensitive to the asset models used; the simplistic model used in this work overlooks many complex phenomena such as contracting economic fluctuations which are highly uncertain. In this vein, a good strategy to obtain more realistic models would be to fit empirical data to the complete building, and then divide the cost approximately between the components themselves. Finally, we also need to recognize that the strategy to repair or replace assets is usually not performed strictly component-wise, this is because most assets are physically interconnected (partition-like components). Therefore improvements need to be made to the loss computation procedures to better reflect this reality.

Future work should improve upon Eq.1.1, refining the criteria of optimum design including human injuries/fatalities plus business downtime. Furthermore, the process should always strive to make the building as realistic as possible with the right level of fidelity, while still being fast enough to be used en-masse for catastrophe models. In this regard, the most important concepts to get right are (in order):

- Improve the mechanical models of the structure, paying special attention to achieving realistic structural response and the correct estimation of the onset of collapse.
- Improve upon the asset cost model, both the asset placement algorithm and their fragility/vulnerability curves, or introduce new algorithms to compute loss using all the information available from the structural analyses results.
- Improve upon the collapse modes; such as including partial collapse, excessive residual drift, excessive and unrepairable damage across floors, etc.
- Include foundation assets and soil-structure interaction.

Future work should study the consequences of a wider variety of building archetypes, construction materials, and design algorithms on the optimum design while revising the notion that simple rules can effectively lead us to tolerable losses.

# Bibliography

[Argyris et al., 1979] Argyris, J., Hilpert, O., Malejannakis, G., and Scharpf, D. (1979). On the geometrical stiffness of a beam in space—a consistent v.w. approach. *Computer Methods in Applied Mechanics and Engineering*, 20(1):105–131.

[ASCE and SEI, 2006] ASCE and SEI (2006). Minimum design loads for buildings and other structures. Technical report, ASCE/SEI.

[Aslani and Miranda, 2005] Aslani, H. and Miranda, E. (2005). *Probabilistic Earthquake Loss Estimation and Loss Disaggregation in Buildings*. PEER Report. Pacific Earthquake Engineering Research Center.

[Benjamin and Cornell, 1970] Benjamin, J. and Cornell, A. (1970). *Probability Statistics and Decision for Civil Engineers*. Mc-Graw Hill, New York.

[Board and Eddington, 2013] Board, M. I. and Eddington, P. R. (2013). *Mars Climate Orbiter: Phase I Report*. Nimble Books LLC, Ann Arbor, MI, USA.

[Brooks, 1982] Brooks, F. (1982). *The Mythical man-month : essays on software engineering*. Reading, Mass. : Addison-Wesley Pub. Co., 1982. ©1975.

[Brzev et al., 2013] Brzev, S., Scawthorn, C., Charleson, A. W., Allen, L., Greene, M., Jaiswal, K., and Silva, V. (2013). Gem building taxonomy (version 2.0). Report.

[Chopra, 2012] Chopra, A. (2012). *Dynamics of Structures*. Prentice Hall, 4 edition.

[Chopra and Goel, 2000] Chopra, A. and Goel, R. (2000). Building period formulas for estimating seismic displacements. Technical report, Earthquake Spectra.

[Chopra and McKenna, 2016] Chopra, A. K. and McKenna, F. (2016). Modeling viscous damping in nonlinear response history analysis of buildings for earthquake excitation. *Earthquake Engineering & Structural Dynamics*, 45(2):193–211.

[Chung and Hulbert, 1993] Chung, J. and Hulbert, G. M. (1993). A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-alpha method. *Journal of Applied Mechanics*, 60(2):371–375.

[Clough and Penzien, 1993] Clough, R. and Penzien, J. (1993). *Dynamics of Structures*. Civil engineering series. McGraw-Hill.

[Cruz and Miranda, 2017] Cruz, C. and Miranda, E. (2017). First-mode damping ratios inferred from the seismic response of buildings.

[Cuthill and McKee, 1969] Cuthill, E. and McKee, J. (1969). Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th National Conference*, ACM '69, New York, NY, USA. Association for Computing Machinery.

[D. A. Makhloof, 2021] D. A. Makhloof, A. R. Ibrahim, X. R. (2021). Damage assessment of reinforced concrete structures through damage indices: A state-of-the-art review. *Computer Modeling in Engineering & Sciences*, 128(3):849–874.

[Dhakal and Aninthaneni, 2021] Dhakal, R. and Aninthaneni, P. (2021). Are stronger, stiffer buildings indeed costlier? - case study of rc frame buildings. *Christchurch, NZ: Annual Conference of NZ Society for Earthquake Engineering*.

[Eads, 2013] Eads, L. (2013). *Seismic collapse risk assessment of buildings: effects of intensity measure selection and computational approach*. PhD thesis, Stanford University.

[Eldawie, 2020] Eldawie, A. (2020). *Collapse modeling of reinforced concrete frames under seismic loading*. PhD thesis, Ohio State University.

[Elwood, 2004] Elwood, K. (2004). Modelling failures in existing reinforced concrete columns. *Canadian Journal of Civil Engineering - CAN J CIVIL ENG*, 31:846–859.

[Esteva, 1967] Esteva, L. (1967). Criterios para la construcción de espectros de diseño sísmico. *Proceeding of the 3rd Pan-American Symposium of Structures*.

[Esteva, 1970] Esteva, L. (1970). Regionalización sísmica de méxico para fines de ingeniería. *Institute of Engineering Series, UNAM*.

[Felippa, 2001] Felippa, C. (2001). A historical outline of matrix structural analysis: a play in three acts. *Computers & Structures*, 79(14):1313–1324.

[FEMA, 2009] FEMA (2009). Fema440A - Effects of Strength and Degradation on Seismic Response. Technical report, Applied Technology Council.

[FEMA, 2012a] FEMA (2012a). *FEMA-P58 - Seismic Performance Assessment of Buildings Volume 1 – Methodology*.

[FEMA, 2012b] FEMA (2012b). *FEMA-P58 - Seismic Performance Assessment of Buildings Volume 2 – Implementation Guide*.

[FEMA, 2018] FEMA (2018). *FEMA-P58-5 Next-Generation Performance Assessment of Buildings*.

[Fragiadakis and Papadrakakis, 2008] Fragiadakis, M. and Papadrakakis, M. (2008). Modeling, analysis and reliability of seismically excited structures: computational issues. *International Journal of Computational Methods*, 05(04):483–511.

[Gamma et al., 1995] Gamma, E., Vlissides, J., Helm, R., and Johnson, R. (1995). *Design patterns : elements of reusable object-oriented software*. Reading, Mass. : Addison-Wesley, ©1995.

[George, 2018] George, C. (2018). *Seismic Analysis & Fragility Assessment of Reinforced Concrete Structures through Numerical Modeling*. PhD thesis, University College London.

[Giovinazzi et al., 2015] Giovinazzi, S., Kongar, I., Bocchini, G. M., and Ottonelli, D. (2015). Damage to buildings: Modeling. *Encyclopedia of Earthquake Engineering*, pages 506–524.

[Gob.CDMX, 2023a] Gob.CDMX (2023a). Reglamento de Construcciones para la Ciudad de México, Normas Técnicas Complementarias para diseño por sismo.

[Gob.CDMX, 2023b] Gob.CDMX (2023b). Reglamento de Construcciones para la Ciudad de México, Normas Técnicas Complementarias para diseño y construcción de estructuras de concreto.

[Gob.CDMX, 2023c] Gob.CDMX (2023c). Reglamento de Construcciones para la Ciudad de México, Normas Técnicas Complementarias sobre criterios y acciones para el diseño estructural de edificaciones.

[Grigoriu, 2011] Grigoriu, M. (2011). To scale or not to scale seismic ground-acceleration records. *Journal of Engineering Mechanics*, 137:284–293.

[Guan et al., 2020] Guan, X., Burton, H., and Sabol, T. (2020). Python-based computational platform to automate seismic design, nonlinear structural model construction and analysis of steel moment resisting frames. *Engineering Structures*, 224:111199.

[Gunturi, 1992] Gunturi, S. (1992). *Building Specific Earthquake Damage Estimation*. Stanford University.

[Hadjian, 2001] Hadjian, A. (2001). A general framework for risk-consistent seismic design. *Earthquake Engineering and Structural Dynamics*, (31):601–626.

[Haselton and Deierlein, 2007] Haselton, C. and Deierlein, G. (2007). Assessing seismic collapse safety of modern reinforced concrete moment frame buildings. Technical report, The John A. Blume Earthquake Engineering Center, Stanford University.

[Haselton et al., 2009] Haselton, C., Liel, A., and Deierlein, G. (2009). Simulating structural collapse due to earthquakes: model idealization, model calibration and numerical solution algorithms. *Computational Methods in Structural Dynamics and Earthquake Engineering - 2nd international conference on Computational Methods in Structural Dynamics and Earthquake Engineering*.

[Haselton et al., 2016] Haselton, C., Liel, A., Taylor-Lange, S., and Deierlein, G. (2016). Calibration of model to simulate response of reinforced concrete beam-columns to collapse. *ACI Structural Journal*, 113.

[Haselton et al., 2012] Haselton, C., Whittaker, A., Hortacsu, A., Baker, J., Bray, J., and Grant, D. (2012). Selecting and scaling earthquake ground motions for performing response-history analyses. In *Proceedings of the 15th world conference on earthquake engineering*, pages 4207–4217. Earthquake Engineering Research Institute.

[Hilber et al., 1977] Hilber, H. M., Hughes, T. J. R., and Taylor, R. L. (1977). Improved numerical dissipation for time integration algorithms in structural dynamics. *Earthquake Engineering & Structural Dynamics*, 5(3):283–292.

[Hill and Rossetto, 2008] Hill, M. and Rossetto, T. (2008). Comparison of building damage scales and damage descriptions for use in earthquake loss modelling in europe. *Bulletin of Earthquake Engineering*, 6:335–365.

[Ibarra and Krawinkler, 2005] Ibarra, L. and Krawinkler, H. (2005). Global collapse of frame structures under seismic excitations. Technical report, PEER.

[Ibarra et al., 2005] Ibarra, L., Medina, R., and Krawinkler, H. (2005). Hysteretic models that incorporate strength and stiffness deterioration. *Earthquake Engineering & Structural Dynamics*, 34:1489 – 1511.

[Jiang et al., 2011] Jiang, H., Chen, L., and Chen, Q. (2011). Seismic damage assessment and performance levels of reinforced concrete members. *Procedia Engineering*, 14:939–945. The Proceedings of the Twelfth East Asia-Pacific Conference on Structural Engineering and Construction.

[Kalkan and Chopra, 2010] Kalkan, E. and Chopra, A. K. (2010). Practical guidelines to select and scale earthquake records for nonlinear response history analysis of structures. *US geological survey open-file report*, 1068(2010):126.

[Kay, 1996] Kay, A. C. (1996). *The Early History of Smalltalk*, page 511–598. Association for Computing Machinery, New York, NY, USA.

[Kunnath et al., 1997] Kunnath, S., El-Bahy, A., Taylor, A., and Stone, W. (1997). Cumulative seismic damage of reinforced concrete bridge piers.

[López, 2015] López, S. (2015). *A displacement-based seismic design method for framed structures involving sidesway-collapse prevention*. PhD thesis, Instituto de Ingeniería, UNAM.

[Marques et al., ] Marques, M., Macedo, L., Araújo, M., Martins, L., Castro, J. M., Sousa, L., Silva, V., and Delgado, R. *Influence of Record Selection Procedures on Seismic Loss Estimations*, pages 1756–1766.

[McCormack and Rad, 1997] 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.

[McKenna et al., 2004] McKenna, F., Fenves, G., and Scott, M. (2004). Open system for earthquake engineering simulation. Technical report, Pacific Earthquake Engineering Research Center.

[Miranda, 1999] Miranda, E. (1999). Approximate seismic lateral deformation demands in multistory buildings. *Journal of Structural Engineering*, 125(4):417–425.

[Miranda and Reyes, 2002] Miranda, E. and Reyes, C. J. (2002). Approximate lateral drift demands in multistory buildings with nonuniform stiffness. *Journal of Structural Engineering*, 128(7):840–849.

[Möller et al., 2015] Möller, O., Foschi, R. O., Ascheri, J. P., Rubinstein, M., and Grossman, S. (2015). Optimization for performance-based design under seismic demands, including social costs. *Earthquake Engineering and Engineering Vibration*, 14(2):315–328.

[Newmark and Rosenblueth, 1971] Newmark, N. and Rosenblueth, E. (1971). *Fundamentals of earthquake engineering*. Prentice-Hall Civil engineering and engineering mechanics series. Prentice-Hall.

[Ordaz et al., 2021] Ordaz, M., Salgado-Gálvez, M., and Giraldo Grisales, S. (2021). R-crisis: 35 years of continuous developments and improvements for probabilistic seismic hazard analysis. *Bulletin of Earthquake Engineering*, 19.

[Ordaz et al., 2017] Ordaz, M., Salgado-Gálvez, M. A., Rocha, L. E. P., and Cardona, O. D. (2017). Optimum earthquake design coefficients based on probabilistic seismic hazard analyses: theory and applications. *Earthquake Spectra*, 33.4:1455–1476.

[Panagiotakos and Fardis, 2001] Panagiotakos, T. and Fardis, M. (2001). Deformations of Reinforced Concrete Members at Yielding and Ultimate. *American Concrete Institute Structural Journal*, 98(2).

[Park and Ang, 1985] Park, Y. and Ang, A. (1985). Mechanistic seismic damage model for reinforced concrete. *Journal of Structural Engineering*, 111(4):722–739.

[Picketty, 2014] Picketty (2014). *Capital in the Twenty-First century*. Harvard University Press.

[Powell, 2010] Powell, G. (2010). *Modeling for Structural Analysis: Behavior and Basics*. Computers and Structures.

[Priestley et al., 2007] Priestley, M., Calvi, G., and Kowalsky, M. (2007). *Displacement-based seismic design of structures*. IUSS Press.

[Ramirez and Miranda, 2009] Ramirez, C. and Miranda, E. (2009). Building-specific loss estimation methods & tools for simplified performance-based earthquake engineering. Technical Report 171, The John A. Blume Earthquake Engineering Center, Stanford University.

[Ramirez, 2009] Ramirez, M. (2009). *Building specific loss estimation methods & tools for simplified performance-based earthquake engineering*. PhD thesis, Department of Civil and Environmental Engineering, Stanford.

[Reinoso and Jaimes, 2013] Reinoso, E. and Jaimes, M. (2013). Estimación de pérdidas por sismo de contenido de edificios. Technical report, Instituto de Ingeniería, UNAM.

[Rojahn et al., 1985] Rojahn, C., Sharpe, R. L., Council, A. T., and Agency, U. S. F. E. M. (1985). *Earthquake damage evaluation data for California*. Applied Technology Council Redwood City, California, Redwood City, California.

[Rojas et al., 2011] Rojas, H., Foley, C., and Pezeshk, S. (2011). Risk-based seismic design for optimal structural and nonstructural system performance. *Earthquake Spectra*, 27:857–880.

[Rosenblueth, 1976a] Rosenblueth, E. (1976a). Optimum design for infrequent disturbances. *Journal of the Structural Division*, (ST9):1807–1825.

[Rosenblueth, 1976b] Rosenblueth, E. (1976b). Towards optimum design through building codes. *Journal of the Structural Division*, ST3:591–607.

[Salgado-Gálvez et al., 2015] Salgado-Gálvez, M., Bernal, G., Barbat, A., Tibaduiza, M., and Cardona, O. (2015). Probabilistic estimation of annual lost economic production due to premature deaths because of earthquakes. *Human and Ecological Risk Assessment*.

[Scott and Fenves, 2010] Scott, M. and Fenves, G. (2010). Krylov subspace accelerated newton algorithm: Application to dynamic progressive collapse simulation of frames. *Journal of Structural Engineering-asce - J STRUCT ENG-ASCE*, 136.

[Sextos, 2014] Sextos, A. G. (2014). Selection of ground motions for response history analysis. *Encyclopedia of Earthquake Engineering*, pages 1–10.

[Shibata and Sozen, 1976] Shibata, A. and Sozen, M. (1976). Substitute-structure method for seismic design in r/c. *Journal of the Structural Division*, 102:1–18.

[Silva et al., 2019] Silva, V., Akkar, S., Baker, J., Bazzurro, P., Castro, J., Crowley, H., Dolsek, M., Galasso, C., Lagomarsino, S., Monteiro, R., Perrone, D., Pitilakis, K., and Vamvatsikos, D. (2019). Current challenges and future trends in analytical fragility and vulnerability modeling. *Earthquake Spectra*, 35.

[Silva et al., 2017] Silva, V., Akkar, S., Baker, J., Bazzurro, P., Castro, J. M., Crowley, H., Dolsek, M., Galasso, C., Lagomarsino, S., Monteiro, R., Perrone, D., Pitilakis, K., and Vamvatsikos, D. (2017). Current challenges and future trends in analytical fragility and vulnerability modelling. *Earthquake Spectra*.

[Stroustrup, 2013] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley Professional, 4th edition.

[Taghavi et al., 2003] Taghavi, S., Miranda, E., and Miranda, M. (2003). *Response Assessment of Nonstructural Building Elements*. PEER Report. Pacific Earthquake Engineering Research Center.

[Vamvatsikos, 2004] Vamvatsikos, D. (2004). Applied incremental dynamic analysis. *Earthquake Spectra*, 20(2):523–553.

[Vamvatsikos and Cornell, 2002] Vamvatsikos, D. and Cornell, C. (2002). Incremental dynamic analysis. *Earthquake Engineering & Structural Dynamics*, 31:491 – 514.

[VonNeumann and Morgernstern, 1953] VonNeumann, J. and Morgernstern, J. (1953). *Theory of Games and Economic Behavior*. Princeton University Press.

[Whitman and Cornell, 1976] Whitman and Cornell, A. (1976). Chapter 9; design. *Seismic Risk and Engineering Decisions*, pages 339–379.

[Whitman et al., 1973] Whitman, R., Reed, J., and Hong, S. (1973). Earthquake damage probability matrices. *Proceedings of the 5th World Conference on Earthquake Engineering, Rome, Italy*, 2.

[Wilson, 2002] Wilson, E. (2002). *Three-Dimensional Static and Dynamic Analysis of Structures A Physical Approach With Emphasis on Earthquake Engineering*. Computers and Structures.

[Zienkiewicz and Taylor, 2013] Zienkiewicz, O. and Taylor, R. (2013). *The Finite Element Method for Solid and Structural Mechanics*. The Finite Element Method. Elsevier Science.