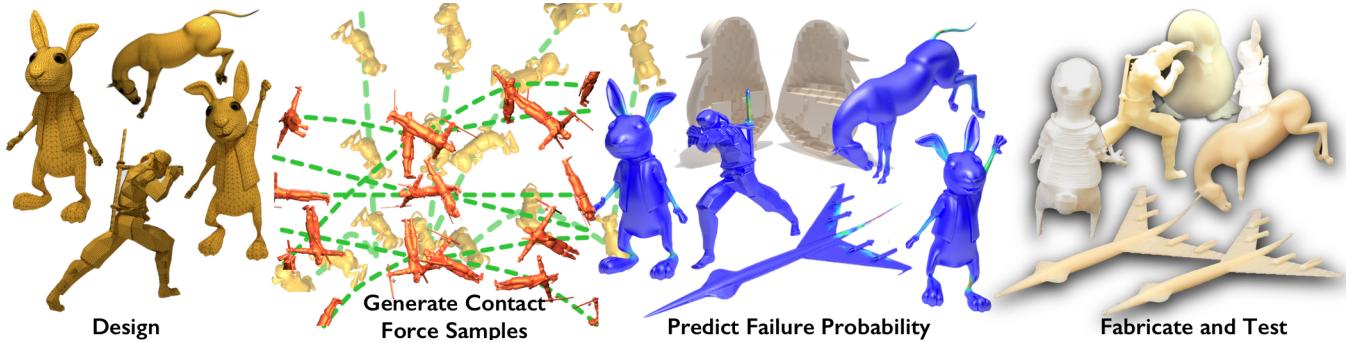


# Stochastic Structural Analysis for Context-Aware Design and Fabrication

Timothy Langlois<sup>\*†||</sup> Ariel Shamir<sup>\*‡</sup> Daniel Dror<sup>\*‡</sup> Wojciech Matusik<sup>\*§</sup> David I.W. Levin<sup>\*¶</sup>

\*Disney Research †Cornell University ‡Adobe Research ‡The Interdisciplinary Center §MIT CSAIL ¶University of Toronto



**Figure 1:** Our new stochastic finite element method uses sampled rigid body trajectories (left) to predict the failure modes of objects under real-world circumstances (middle). We tested our method on more than 30 3D printed models (right) and found that our algorithm predicted over 90% of observed failures.

## Abstract

In this paper we propose failure probabilities as a semantically and mechanically meaningful measure of object fragility. We present a stochastic finite element method which exploits fast rigid body simulation and reduced-space approaches to compute spatially varying failure probabilities. We use an explicit rigid body simulation to emulate the real-world loading conditions an object might experience, including persistent and transient frictional contact, while allowing us to combine several such scenarios together. Thus, our estimates better reflect real-world failure modes than previous methods. We validate our results using a series of real-world tests. Finally, we show how to embed failure probabilities into a stress constrained topology optimization which we use to design objects such as weight bearing brackets and robust 3D printable objects.

**Keywords:** structural analysis, computational design, FEM

**Concepts:** •Computing methodologies → Modeling and Simulation; Uncertainty Quantification;

## 1 Introduction

Every designer must master the art of compromise. Whether designing a child’s toy or a bridge, one must balance usability and reliability with aesthetic considerations. For instance, a designer may reinforce a toy so that it can survive a fall from a child’s hand, but may not be willing to sacrifice the design aesthetics to allow the toy to survive being deliberately thrown to the ground. For a real-world bridge, a designer may make the opposite choice – sacrificing

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. © 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM.

SA ’16 Technical Papers., December 05 - 08, 2016, Macao

ISBN: 978-1-4503-4514-9/16/12

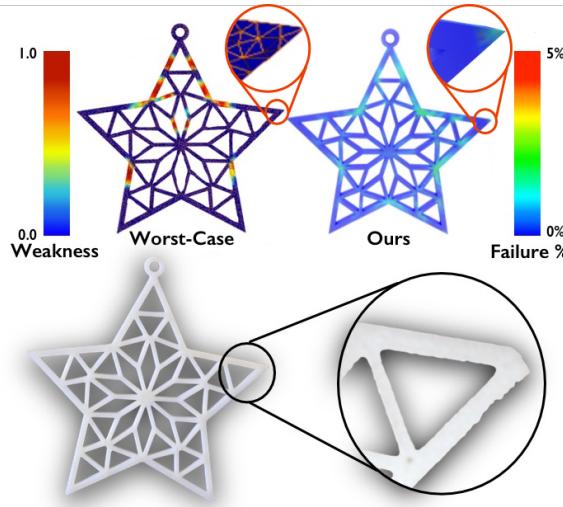
DOI: <http://dx.doi.org/10.1145/2980179.2982436>

the design for sturdiness in all scenarios. What becomes clear is that the susceptibility of an object, or its various parts, to failure depends on the “typical” use cases of the object. For instance, a delicate internal part of a toy may not be so vulnerable if it is protected by other protrusions. In this paper, we explore the use of probability of failure as a semantically and computationally meaningful measure of reliability, and a compact tool for encoding aggregate object behavior over a range of real-world scenarios. We use this probabilistic formulation of object failure for both forward analysis tasks and robust inverse computational design.

The primary computational technique for failure analysis of solid structures is the finite element method [Belytschko et al. 2013], which allows for large-scale analysis of complicated structures and materials. Of particular interest to us are so-called “worst-case” methods [Stava et al. 2012; Zhou et al. 2013] which attempt to identify the most flimsy parts of a design so that users can modify them. These methods suffer from at least one of two fundamental limitations. The first is that presenting mechanical stress, or arbitrary, spatially varying weakness ratings to a designer is nonintuitive. Without further understanding of structural mechanics, the designer cannot reliably interpret how often these stresses/weak components will lead to breakage. This means the designer cannot fully understand whether he should compromise the aesthetic of the object in search of reliability. Second, the “worst-case” scenario, the one in which the object fails most easily, may never be encountered during real-world usage of the object. Thus, designing to account for these can lead to ‘over-engineering’ that costs time and money, and can sacrifice the design aesthetics.

In contrast to these methods, we use techniques from stochastic finite element analysis to compute the spatially varying probability of failure from simulations of an object’s real-world usage. Stochastic finite element methods (SFEM) are an extension of standard finite elements which deal with force distributions rather than single instances of an applied force. Thus, to use SFEM, we need a method for generating realistic force distributions and to study their effect on an object. Our method is the first to use an external physics simulation to estimate real world loading conditions for an object.

Given a 3D object, we use a fast rigid-body physics engine to generate contact force samples on its surface. Rather than perform finite element analysis on all samples, we use randomized singular



**Figure 2:** A comparison between worst-case structural analysis (top left) and our method (top right) for a star geometry falling to the ground. Worst-case analysis [Zhou et al. 2013] incorrectly predicts interior struts as being weak while also failing to predict that the corners of the star may break. Our method avoids both these faults and agrees more closely with the results of a real-life drop tests (as seen at the bottom).

value decomposition in the contact sample space to derive a low-dimensional, reduced force-space. This accelerates our computation of object stress samples from our contact samples by approximately 100 $\times$ . With these stress samples we can compute the failure probability density function, which gives us the probability that the maximum stress experienced by an object will exceed a given threshold. We do this without resorting to coupled spatial-probability discretizations common to other perturbation-based stochastic finite element schemes [Stefanou 2009].

Our probability of failure measure is easy to understand. For instance, a toy that will survive 99% of its interactions with the world can be considered robust. Using probability also allows designers to make more informed design decisions, and avoids over-engineering by ensuring that virtual testing matches real-world usage. Figure 2 illustrates the difference by comparing our method to worst-case structural analysis [Zhou et al. 2013]. First, our method produces meaningful failure probability maps. Second, our use of real world scenarios avoids errors such as labeling interior struts as vulnerable, and correctly predicting that the corners of the object will chip. Finally, unlike previous methods, our probabilities are comparable between objects and thus induce a reliability ordering over sets of designs. This allows a designer to evaluate a particular design relative to others.

We apply our new method to both forward analysis and inverse design problems. In forward design, we make similar assumptions to previous work, that the object only deforms infinitesimally (larger deformations cause breakage), and focus our analysis on stiff materials such as ABS plastic. The input to the forward design problem is a 3D surface mesh with volumetric material assignments, as well as a definition of the typical usage scenario for the object. The outputs of our method are a map of the spatially varying probabilities for fracture, and a total probability of object fracture (under the given usage). This map can be used by designers and engineers to manually modify and reinforce their designs. Figure 1 shows the basic stages of our algorithm along with a sample of our virtual and fabricated results.

Solving the inverse design problem requires not only predicting the likelihood of object fracture, but also using computation to automatically correct design flaws. In this paper we also illustrate the use of our stochastic finite element method to perform robust topology optimization. The input to the inverse problem are: a 3D mesh with a volumetric material assignment, the typical usage scenario, as well as a minimum fracture probability. Our method automatically tries to reduce the weight of the object, while guaranteeing the object's probability of breaking is always less than the specified input value. To facilitate this, we derive efficient formulas for computing the gradient of our failure probabilities. This allows us to use well-tested, continuous optimization schemes. We demonstrate the algorithm by designing optimized weight bearing brackets and other robust, 3D printable objects.

Our paper makes the following technical contributions:

- An efficient stochastic finite element method which can derive compact force distributions from external simulations or data,
- a reduced-space approach that accelerates the computation of object failure probabilities,
- a topology optimization formulation using the above numerical method,
- a fast method for evaluating the gradients of the stochastic design problem, allowing the use of efficient continuous optimization methods,
- application of the method to structural failure prediction and automatic design reinforcement.

The remainder of this paper outlines our approach to the stochastic finite element method as well as describes the efficient implementation of both the forward stochastic simulation and several inverse design problems. Armed with these tools we show several virtual and fabricated examples that illustrate the ability of our method to tackle the inherently unpredictable context-aware structural analysis.

## 2 Related Work

If one is to build anything, it is critical to know whether or not it will fall apart. It is for this reason that structural analysis of elastica has been extensively studied. In the late 19th and early 20th centuries, Antoni Gaudi used a hanging chain model [Tomlow 1989] to ensure the cathedrals he was designing would stay upright. In a similar vein engineers and mathematicians have studied the rigidity and stability of truss and beam structures in attempts to ensure their reliability [Crapo 1979; McCormac and Elling 1984; Crapo and Whiteley 1989]. For more complicated, multi-material objects, the go-to technique for structural analysis is the finite element method (FEM) [Hughes 2012; Belytschko et al. 2013] which has been used to study everything from cracks in beams [Gounaris and Dimarogonas 1988] to human beings [Melosh 1974].

In a similar vein, the fabrication sub-field of computer graphics [Prévost et al. 2013; Bächer et al. 2014; Musalski et al. 2015; Musalski et al. 2016] has also dealt with preventing fabricated objects from failing. Zhou et al. [2013] present a method for performing “worst-case” structural analysis, in which they compute a contact pressure field which produces the worst possible stress distribution within an object. They use this to produce a scaled “weakness” map which indicates areas of likely failure. As illustrated in Figure 2, such a method can sometimes generate unrealistic stress distributions and erroneously predict areas of failure. Umetani and Schmidt [2013] use a simplified beam model which allows for fast computation of structural weakness. These methods all suffer from the same limitations. First, they produce arbitrary weakness maps which are not semantically meaningful – they do not qualitatively demonstrate how fragile one design is when compared to

another. Second, the analysis is performed without any regard for how the object will interact with the world. The notion of weakness is different for a child's toy than it is for the landing gear of a Jumbo Jet. We believe that context is important and that structural analysis methods should take it into account.

Injecting context into standard analysis methods can be difficult. Stava et al [2012] make some progress in this direction by performing structural analysis for a limited set of cases, load under gravity and a two fingered pinch but we seek a more general solution. This is the raison-d'être of the Stochastic Finite Element Method (SFEM). These method are a relatively recent invention of the engineering community which are receiving much attention [Ma 1987; Der Kiureghian and Ke 1987; Ang and Tang 2007] (see Stefanou [2009] for a review).

The standard SFEM problem is to determine the response of an object with stochastic material properties to an applied load. SFEM has a long history in engineering [Liu et al. 1986b; Mahadevan and Haldar 1991; Matthies and Keese 2005], and there are two major approaches for such a problem: Monte Carlo [Schuller 2006] and Perturbation Methods [Liu et al. 1986a]. Monte Carlo methods sample the material property space, instantiating a large number of deterministic FEM matrices and solving them, while Perturbation methods use a local expansion around the mean of the material properties to arrive at a large system of equations that can be solved to yield the stochastic response. Both methods are performance intensive, Monte Carlo methods require solving a huge number of finite element systems while Perturbation Methods give rise to huge linear systems which must be inverted [Eiermann et al. 2007]. Perturbation methods can also yield inaccurate responses when variance in forces is high [Liu et al. 1986a], as such, Monte Carlo methods are often used as the gold-standard comparison [Stefanou 2009].

In this work we solve a different problem. We assume the material properties of our object are fixed while the applied load is stochastic and of unknown variance. Specifically, we concern ourselves with the non-trivial interaction of linearly elastic, infinitesimally deforming objects with the world, including persistent and transient frictional contact. We exploit fast rigid body simulation [Coulmans et al. 2015] to build an efficient Monte Carlo method for reliability analysis under these conditions. Contact problems are relatively unexplored in the SFEM literature and previous work tends to rely on highly specialized models for specific scenarios [Faravelli and Bigi 1990]. This is quite different from our general approach. Our method also shares similarities with the Response Surface Method [Zhang et al. 2014]; however rather than compute the surface as an intermediate step, we directly compute object failure probabilities from our contact force samples. Our major contribution is speeding up Monte Carlo based approaches for stochastic analysis by an order of magnitude.

Stochastic finite elements can also be used to solve optimization problems. In this work we focus on the topology optimization problem, wherein the topology of an object is optimized to meet certain requirements [Bendsøe and Sigmund 2009]. Most topology optimization problems attempt to minimize the compliance of an object given a certain amount of material [Sigmund 1997], such problems have recently been explored in computer graphics as well [Wang et al. 2013; Lu et al. 2014; Dumas et al. 2015; Martínez et al. 2015]. Minimum compliance, however, cannot tell us whether an object will fail or not – such an outcome is determined by the internal stresses developed under load. Instead, we solve a stochastic variant of the stress constrained topology optimization problem ([Lee et al. 2012]) wherein one attempts to minimize the weight of an object subject to a constraint on the yield stress. While stochastic topology optimization is quite new there has been much work done in the engineering community [Chen et al. 2010; Evgrafov et al. 2003;

Maute 2014]. The key difference between this work and our own is twofold; first, unlike previous approaches we optimize for the structure of an object given a constraint on its probability of failure. Second, we propose a fast, PCA-based, Monte Carlo sampling approach that allows us to perform this optimization efficiently, under complicated loading conditions (such as contact forces) generated by an external simulator.

### 3 Background on Structural Analysis

**Notation.** Here, we briefly outline the notational conventions followed in this work. We express scalar variables and functions as lower-case letters ( $f$ ), vector-valued ones as lower-case, boldfaced letters ( $\mathbf{v}$ ) and those that are matrix-valued as upper-case letters ( $A$ ). When necessary to denote a specific entry in a vector, or a column of a matrix we use a lower-right subscript ( $v_j$  indicates the  $j^{th}$  component of vector  $v$  while  $A_j$  denotes the  $j^{th}$  column of matrix  $A$ ). The discussion of finite element techniques requires that we have the ability to discuss certain properties in global (whole-object) or per-element scope. We use a superscript  $e$  to denote that a particular property is being used in a per-element fashion (i.e.  $v$  is a vector associated with the whole-object while  $v^e$  is the component of that vector associated with the  $e^{th}$  element). This per-element notation applies to functions as well, that is, if function  $f^e(\cdot)$  computes a per-element scalar quantity, then  $\mathbf{f}(\cdot)$  returns a vector wherein the  $e^{th}$  component contains the value  $f^e(\cdot)$ . Finally, the description of our algorithm necessitates that we reference per-sample quantities. We reserve the index  $i$  to represent the  $i^{th}$  sample and place it as a left hand superscript (i.e.  ${}^i \mathbf{f}$  is a vector,  $\mathbf{f}$ , associated with the  $i^{th}$  sample).

**The Finite Element Method.** At the heart of our approach lies the finite element method (FEM) for linearly elastic objects. FEM allows us to compute the stresses induced on an object by the applied forces. Briefly, the finite element method begins by discretizing an object using a computational mesh of simple volumetric cells. In our case we rely on a hexahedral, embedded FEM which divides our object into a number of axis-aligned hexahedra. The static equilibrium conditions which result from this discretization are

$$K\mathbf{u} = \mathbf{f}, \quad (1)$$

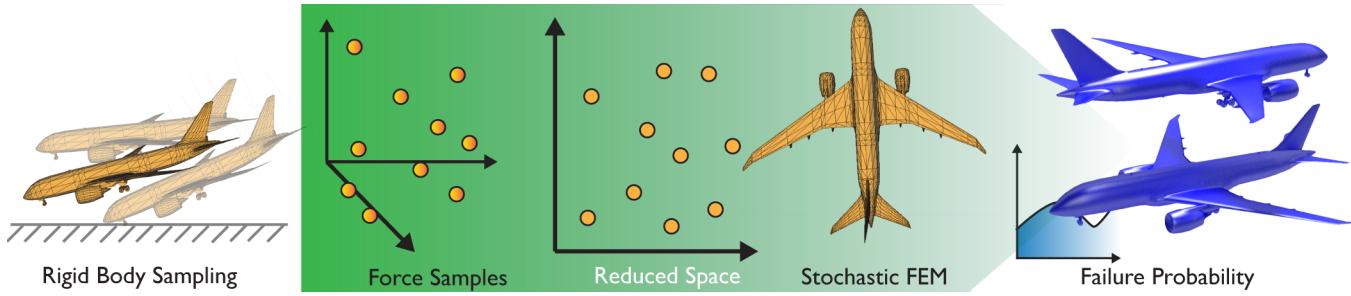
where  $K \in \mathbb{R}^{3n \times 3n}$  is the stiffness matrix,  $n$  is the number of vertices in the computational mesh,  $\mathbf{u} \in \mathbb{R}^{3n}$  are the nodal displacements and  $\mathbf{f} \in \mathbb{R}^{3n}$  are the applied external forces.

Crucial to computing whether or not an object will break is the ability to compute the internal stresses acting on the object. These stresses are given by the simple formula

$$\boldsymbol{\sigma} = CB\mathbf{u}, \quad (2)$$

where  $\boldsymbol{\sigma} \in \mathbb{R}^{6m}$  is a vector of object's Cauchy stresses,  $C \in \mathbb{R}^{6m \times m}$  is the constitutive matrix (determined by the material chosen),  $m$  is the number of elements in the computational grid and  $B \in \mathbb{R}^{m \times 3n}$  computes the Cauchy strain from the nodal displacements  $\mathbf{u} = K^{-1}\mathbf{f}$ .

**Detecting Failure using Yield Stresses.** We detect object failure on a per-element level, by investigating the per-element stress ( $\boldsymbol{\sigma}^e$ ), which is a 6D vector that can be extracted from the global stress vector  $\boldsymbol{\sigma}$ . An object fractures when any per-element stress passes outside of a "yield surface" in the 6-dimensional stress space. For many applications one can use a simplified yield surface which is based on computing the von Mises stress. The von Mises stress is



**Figure 3:** Solving for failure probabilities begins with the use of a rigid body simulator to compute inertial and contact force distributions. Randomized SVD is used to find a reduced space which accounts for 95% of the force variance. This reduced space is used to accelerate Stochastic Finite Element analysis which yields both spatially varying and per-object failure probabilities. Here we see that this plane can land without incident under normal circumstances.

a function of  $\sigma^e$  and is given by

$$\tilde{\sigma}^e(\sigma^e) = \frac{1}{2} [(\sigma_1^e - \sigma_2^e)^2 + (\sigma_2^e - \sigma_3^e)^2 + (\sigma_3^e - \sigma_1^e)^2] + 3(\sigma_4^{e2} + \sigma_5^{e2} + \sigma_6^{e2}), \quad (3)$$

We decorate the von Mises stress with a buckling bar since it is a scalar measure of total element deformation. For any material there is a scalar threshold stress  $\hat{\sigma}$ , called the yield stress. If the von Mises stress exceeds the yield stress  $\tilde{\sigma}^e(\sigma^e) > \hat{\sigma}$ , the element (and thus the object) will fracture – indicated by the “broken” bar above the stress sign. The von Mises stress has been shown experimentally to be a good predictor of fracture. This, coupled with its straightforward formulation, makes it ideal for use in our stochastic finite element method.

**Stochastic Finite Elements.** On the surface, stochastic FEM is a simple modification to the standard version defined by Equation 1 wherein the deterministic variables  $\mathbf{u}$  and  $\mathbf{f}$  are replaced by random variables. This allows us to study the behavior of a mechanical system in a probabilistic way, computing expected displacements, variances, and most important for us, the probability of certain events such as fracture. In the next section, we will outline how exactly we use the stochastic FEM formalism to compute these probabilities.

## 4 Fracture Probabilities using Stochastic Finite Elements Method

Our method for computing fracture probabilities takes as input a 3D triangle mesh with associated volumetric material assignment, and the usage scenario for the object. It generates, as output, both per-element fracture probabilities and the probability of fracture for the entire object (Figure 3). Computing fracture probabilities proceeds in five stages:

1. Generate a force distribution using an external simulator (Section 4.1).
2. Compute per-element stress distributions from force distributions (Section 4.2).
3. Convert per-element von Mises stress distributions into the 1-dimensional probability of maximum stress experienced by the entire object (Section 4.3).
4. Compute the cumulative distribution function (CDF) of the maximum stress distribution (Section 4.4).
5. Compute the probability that the stress will exceed the yield stress using the CDF.

In the next sections we explore each of these stages in detail.

### 4.1 Generating Force Distributions

Our algorithm uses an off-the-shelf rigid body simulator [Coulmans et al. 2015] to generate force distributions from simulations of an object in the real world. We make two assumptions that allow us to generate force distributions in this way. First, we assume that our object only deforms infinitesimally. This means that we do not need to update the object’s geometry as it moves through the world. Second, we assume that the object instantaneously returns to its undeformed state after a force is applied (as is common for many relatively stiff materials such as wood, ABS plastic and metal). This means that we can assume our object’s geometry and inertial properties are fixed over time and thus it can be simulated as a rigid body. We regard our rigid body simulator as a blackbox function

$$\mathcal{F} = \mathbf{S}(\phi_0), \quad (4)$$

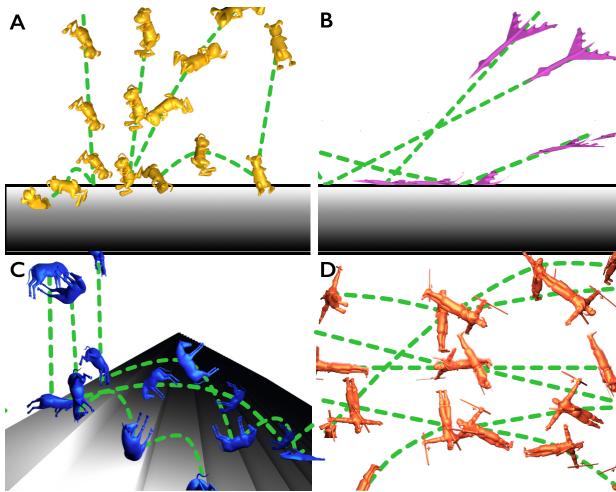
where  $\mathcal{F}$  is a random vector representing the force distribution generated by simulating initial conditions represented by a second random variable  $\phi_0$ , a random 12-vector which stores the body’s center-of-mass position and orientation in the world space, as well as its linear and angular velocities (each represented by 3 scalar values).

We represent this distribution as a matrix of force samples  $F^{rigid}$  where each column,  $F_i^{rigid}$  is a single force sample generated by a single initial condition sampled from  $\phi_0$ . Concretely each column of  $F^{rigid}$  has the following form:

$$F_i^{rigid} = \begin{pmatrix} {}^i\mathbf{f}_1^c \\ \vdots \\ {}^i\mathbf{f}_s^c \\ {}^i\mathbf{f}^{COM} \\ {}^i\boldsymbol{\tau}^{COM} \end{pmatrix}, \quad (5)$$

where  ${}^i\mathbf{f}_k^c \in \mathbb{R}^3$  is a contact force sample acting on the surface of the object (we use the surface of our finite element simulation mesh),  $s$  is the total number of surface points,  ${}^i\mathbf{f}^{COM} \in \mathbb{R}^3$  is the inertial force acting on the object’s center of mass and  ${}^i\boldsymbol{\tau}^{COM}$  is the inertial torque acting on the object’s center of mass. Our matrix,  $F^{rigid}$ , consists of  $N_s$  of these columns, where  $N_s$  is the total number of force samples.

Simulating a single rigid body is fast, and creating samples for  $\phi_0$  is embarrassingly parallel, thus we can very quickly generate thousands of force samples. Furthermore, since contact forces tend to be sparsely located over the surface of the object, we can efficiently store this data which we use in subsequent steps to predict fracture probability.



**Figure 4:** We encode context into our analysis by using a rigid body simulator to generate a large number of force samples from user supplied scenarios. Here we show the four scenarios used in our paper, the Drop Test (A), Plane Landing (B), Stairs Fall (C) and Two-Object Collision (D).

#### 4.1.1 Scenarios for Context-Aware Analysis

Altering the initial conditions ( $\phi_0$ ) allows us to encode real-world context into our algorithm. In this paper we explore four different scenarios with varying degrees of complexity (Figure 4), they are

1. *Drop Test*: A single, randomly oriented object is dropped from a randomly determined height onto flat ground.
2. *Plane Landing*: A randomly oriented object is thrown towards the ground at a shallow, but randomly generated, angle of descent.
3. *Stair Fall*: A randomly oriented object is dropped down a set of stairs from a randomly determined height.
4. *Two-Object Collision*: Two randomly oriented objects are collided "head on" at randomly determined velocities.

Table 1 gives parameters for all scenarios. Additional scenarios can be designed either by hand or using real-world data and seamlessly used with our algorithm.

#### 4.2 Per-Element Stress Distribution

Equipped with our force samples,  $F$ , we can now attempt to convert them into a set of per-element stress samples. Naively this would involve solving Equation 1 for each of our force samples, an approach that is clearly computationally intractable. Instead, we propose to learn a reduced basis which represents our force samples. This significantly reduces the number of linear solves required by our algorithm. Note that our approach differs significantly from previous efforts which perform model reduction [Barbić and James 2005] on the object geometry, thus potentially losing the ability to capture the extremely local nature of the contact-induced deformation. Our method exploits the fact that forces are generated from a relatively low dimensional space of initial conditions, leading to efficient reduced space representations.

To map our contact force samples from a rigid body representation to our finite element mesh we use the projection operator  $J$ , a matrix of size  $3n \times (3s + 6)$  given by

$$J = [J_c \quad J_{com} \quad J_\tau], \quad (6)$$

where  $J_c \in \mathbb{R}^{3n \times 3s}$ ,  $J_{com} \in \mathbb{R}^{3n \times 3}$  and  $J_\tau \in \mathbb{R}^{3n \times 3}$ . Here  $J_c$  maps surface degrees of freedom to full volume degrees of freedom, while  $J_{com}^T$  and  $J_\tau^T$  are matrix operators which compute the linear and angular velocities at the center of mass of our finite element mesh (e.g.  $\mathbf{v}_{com} = J_{com}^T \mathbf{v}$  and  $\mathbf{v}_\tau = J_\tau^T \mathbf{v}$  where  $\mathbf{v}$  is an arbitrary vector of nodal velocities on the FEM mesh). This gives us a finite element force sample matrix  $F = JF^{rigid}$

We use randomized singular value decomposition (SVD) [Halko et al. 2011] to perform principal component analysis on  $F$  to create a reduced space. However, performing the SVD on this matrix, of size  $3n \times N_s$  (where  $N_s$  is the total number of force samples), would be prohibitively slow. We accelerate this procedure using the QR decomposition of  $J$  which yields  $Q \in \mathbb{R}^{3n \times (3s+6)}$  and  $R \in \mathbb{R}^{(3s+6) \times (3s+6)}$  where  $(3s + 6) \ll 3n$ . This can be computed quickly due to the "tall and thin" shape of  $J$ . Using the QR decomposition we can write

$$F = JF^{rigid} = QRF^{rigid} = QU'SV^T = USV^T, \quad (7)$$

where  $U'$ ,  $V$  and  $S$  are the singular vectors and singular values of  $RF$  and  $U$  are the right singular vectors of  $JF^{rigid}$ . This follows from the orthogonality of both  $Q$  and  $U'$ . Therefore, we can compute the principal components of  $F$  by performing an SVD on the much smaller matrix  $RF^{rigid}$  which reduces computation time. Figure 5 shows both some sample basis vectors from a horse Drop Test as well as a comparison of original samples and their reconstructions in the reduced space. While not exact, our reduced space approximates the locality of the applied contact samples, their applied direction, and their magnitudes well.



**Figure 5:** Top Row: Six (of 108) basis vectors computed for a Drop Tested horse. Note that our basis vectors consist of sparse collections of contact forces (zoom in). Bottom Row: Comparison of original force samples to their reconstructions in the basis. Here we focus only on the contact area, forces are zero elsewhere in both the original sample and its reduced representation. Colors represent the  $x$  (red),  $y$  (green) and  $z$  (blue) components of the applied force.

The mean sample vector and the principal components form a reduced basis for our finite element force samples, denoted  $\bar{F}$ . We can then represent the  $i^{th}$  force sample as  $\bar{F}^i \alpha$  where  $i \alpha = \bar{F}^T F_i$ . This, in combination with Equation 2, allows us to represent the  $i^{th}$

Scenario	Position		Velocity	
	Center-of-Mass	Orientation	Linear	Angular
Drop Test	$(0, 1.5m, 0) \pm (0, 0.2m, 0)$	Uniform	$(0, 0.2m/s, 0) \pm (0, 0.05m/s, 0)$	$0.4 \pm 0.2$
Plane Landing	0.5m along plane heading	Pitch: $30^\circ \pm 10^\circ$ Roll: $0^\circ \pm 20^\circ$	$7m/s \pm 2m/s$ along plane heading	0.0
Stair Fall	$(0, 1.5m, 0) \pm (0, 0.2m, 0)$	Uniform	$(0, 0.2m/s, 0) \pm (0, 0.05m/s, 0)$	$0.4 \pm 0.2$
Collision	Objects start 1m apart	Uniform	$5m/s \pm 2m/s$ in direction of other object	$3.0 \pm 1$

**Table 1:** Definitions of our four testing scenarios. For random variables, we sample from Gaussian distributions unless otherwise specified. per-element stress sample

$${}^i\boldsymbol{\sigma} = CBK^{-1}\bar{F}{}^i\boldsymbol{\alpha}, \quad (8)$$

wherein we immediately observe that all stress samples can be computed using a number of linear solves equal to the number of columns in  $\bar{F}$  – a significant improvement over the naive solution. In our algorithm we retain a number of principal components such that we preserve 95% of the variance in the force sample data.

### 4.3 PDF for Maximum Stress

If the maximum von Mises stress in any element exceeds the yield stress, the object will fail, e.g. fracture. Thus, we relate the probability of object failure to the probability that an object will experience a given maximum von Mises stress,  $\mathbf{P}(\tilde{\sigma})$ .

We approximate this probability density function (PDF) using a piecewise linear function. To construct this approximation we begin by computing the histogram,  $H$ , which requires binning the maximum von Mises stress of each stress sample, given by  ${}^i\tilde{\sigma} = \max({}^i\boldsymbol{\sigma})$ . Here max is the component-wise maximum taken over the global vector of per-element von Mises stresses, computed by the function  $\tilde{\sigma}^e({}^i\boldsymbol{\sigma}^e)$  from Equation 3. We then discretize  $\mathbf{P}(\tilde{\sigma})$  using a 1-dimensional finite element grid where the  $e^{th}$  element is equipped with  $k$  finite element shape functions  $\psi_i^e(x)$ .

We note that we can represent  $H$  as

$$H(x) = \sum_{i=1}^{N_s} \delta(x - {}^i\tilde{\sigma}) \quad (9)$$

where  $\delta(y)$  is a discrete Dirac delta, defined by

$$\delta(y) = \begin{cases} \frac{1}{N_s} & y = 0 \\ 0 & \text{otherwise} \end{cases}. \quad (10)$$

In each finite element of  $\mathbf{P}(\tilde{\sigma})$  we estimate the distribution as a linear combination of shape functions,  $\mathbf{P}^e(\tilde{\sigma}) = \sum_{l=1}^k a_l \psi_l^e(\tilde{\sigma})$ , where  $a_l$  are unknown coefficient values. These values can be estimated in a Galerkin fashion. Each element,  $e$ , in our probability space discretization emits two linear equations of the form

$$\sum_{l=1}^k a_l \int_y \psi_g^e(y) \psi_l^e(y) dy^e = \int_{\tilde{\sigma}} \psi_g^e(y) H(y) dy, \quad (11)$$

where  $g$  indexes the element shape functions and we integrate over the range of  $\tilde{\sigma}$  covered by element  $e$ . Due to the properties of the discrete Dirac delta, we can simplify this equation to

$$\sum_{l=1}^k a_l \int_{\tilde{\sigma}^e} \psi_g^e(y) \psi_l^e(y) dy^e = \sum_{i=1}^{N_s} \psi_g^e({}^i\tilde{\sigma}) \frac{1}{N_s}, \quad (12)$$

which can be assembled into a tridiagonal linear system of the form  $\mathbf{Na} = \mathbf{b}$ , where  $\mathbf{a}$  is the stacked vector of coefficients  $a_l$ . Due to its sparse nature this system can be solved efficiently. By concatenating

all element shape functions into a matrix  $\Psi(y)$  we can express our entire PDF using the simple equation

$$\mathbf{P}(\tilde{\sigma}) = \Psi(\tilde{\sigma}) N^{-1} \mathbf{b}, \quad (13)$$

### 4.4 CDF and Yield Stresses

Recall that we are interested in computing the probability that the object will not break, i.e., that the maximum von Mises stress stays below the yield stress. This can be expressed as  $\mathbf{P}(\tilde{\sigma} < \hat{\sigma})$ , where  $\hat{\sigma}$  is the yield stress for a given material. This is simply the CDF of the PDF computed above. Another advantage of the piecewise PDF representation is that we can easily modify it to directly compute the CDF by inserting the appropriate integral into Equation 12 so that it becomes

$$\mathbf{P}(\tilde{\sigma} < \hat{\sigma}) = \int_0^{\hat{\sigma}} \Psi(y) N^{-1} \mathbf{b} dy, \quad (14)$$

Since only the  $\Psi$  matrix depends on  $y$  we can integrate this matrix quickly as a preprocess. One of the major advantages of this CDF representation is that it is easy to differentiate, a property that will become crucial when we discuss solving inverse problems later on.

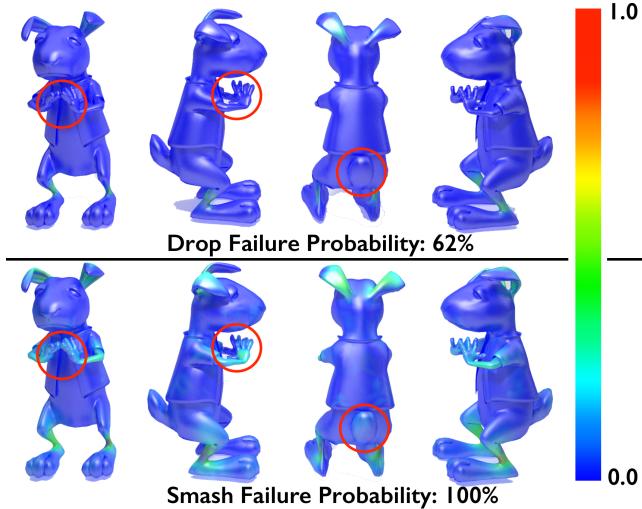
Once this is done, a single linear system solve (additional to our FEM solves), followed by a binary search across our finite element probability grid allows us to efficiently compute the probability that the object will not break. We compute the probability of failure as the compliment of this value, or  $\mathbf{P}_{\text{fail}}(\tilde{\sigma} < \hat{\sigma}) = 1 - \mathbf{P}(\tilde{\sigma} < \hat{\sigma})$ .

## 5 The Forward Analysis Problem

By computing  $\mathbf{P}_{\text{fail}}$  we have solved our forward analysis problem. For any input mesh, accompanying material assignment and collection of usage scenarios we can estimate the probability of object failure. We can also compute a spatially varying map of failure probabilities using our stress samples. In this case, for any point in our object,  $x \in \mathbb{R}^3$ , we can compute local probability of failure,  $\mathbf{P}_{\text{fail}}(x)$ , by evaluating Equation 14 on the per-element von Mises stress samples i.e., without taking the max over the entire object. We can display both  $\mathbf{P}_{\text{fail}}$  and the spatially varying  $\mathbf{P}_{\text{fail}}(x_i)$  to a designer or engineer, allowing them to make design decisions guided by our measure of real world robustness. Two examples can be seen in Figure 6, which highlight the importance of context sensitive analysis. Different scenarios lead to different points of failure for the same object. Notice how the bunny's hands are robust during a simple drop test, but suffer extensive damage when collided with another bunny. This is due to the different nature of the contacts, something that can only be revealed using our method. More results are shown in section 7. We can also combine sample sets to perform an aggregate analysis across multiple scenarios (Figure 7). We compare our combined result to one computed via worst-case analysis (Figure 7). In this case both methods agree that the horse's legs are its weak point but worst-case analysis both fails to predict the non-fragile legs in the stairs and drop cases as well as misses the fragile body and mouth regions in the smash and combined cases.

One of the additional features of failure probability as a metric is that it is comparable between objects. This induces a “reliability

“ordering” on our designs which allows designers to explicitly trade-off reliability for artistic considerations (see Figure 9).



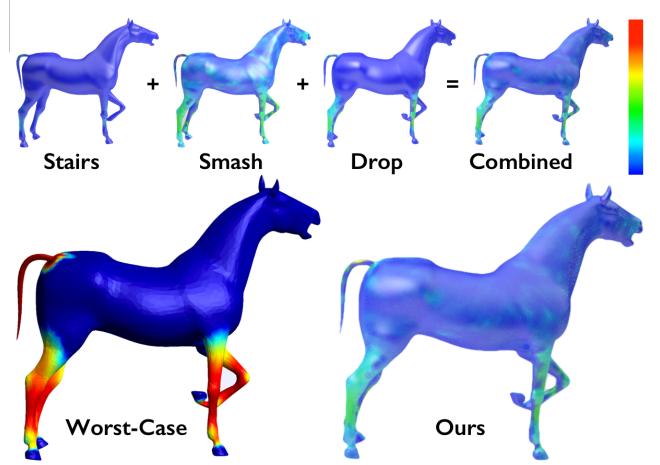
**Figure 6:** Example probability maps for one object (shown in four views) under two different scenarios, our Drop Test scenario and the Two-Object Smash scenario. We have highlighted the major differences in the computed probability maps in red. More complex contacts in the Smash scenario lead to different local reliability estimates (i.e. the breaking arms). Note that to better visualize differences in failure probability we normalize the spatially varying maps with respect to their maximum probability.

## 6 Inverse Design – Topology Optimization

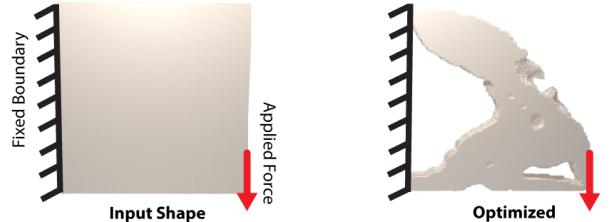
In addition to being used as a guide for designers and engineers, our failure probabilities can also assist in guiding topology optimization. Topology optimization involves either adding or removing material from an initial object design in order to optimize its structure. There are many ways in which topology optimization could utilize failure probabilities but in this paper we explore the specific case of object weight reduction. We minimize the amount of material in each element while maintaining the probability of failure under a given threshold. This can be formulated as:

$$\begin{aligned} \mathbf{w}^* = & \arg \min_{\mathbf{w}} \sum_{e=1}^{N_{el}} w_e \int_{\Omega_e} \rho_e d\Omega, \\ \text{s.t. } & \mathbf{P}(\tilde{\sigma} < \hat{\sigma}) > \Theta \\ & \mathbf{Ku} = \mathbf{f} \\ & 0 < \omega_{min} \leq \mathbf{w} \leq 1 \end{aligned} \quad (15)$$

where  $w_e \in [0, 1]$  is a “fill ratio”, for the  $e^{th}$  finite element where  $\omega_{min}$  indicates the element is completely empty,  $\mathbf{w}$  is the stacked vector of all  $w_e$ ,  $\Omega_e$  and  $\rho_e$  are, respectively, the volume occupied by and the density of the  $e^{th}$  finite element. The small threshold  $\omega_{min}$  (we use 1e-5) is used instead of 0 to avoid singularity. We use  $\mathbf{P}(\tilde{\sigma} < \hat{\sigma})$  to constrain the robustness of the object being optimized. This gives the designer an intuitive handle for trading between the amount of material used during fabrication and the durability of the object. We enforce sparsity of the weights by using the Solid Isotropic Material with Penalization (SIMP) material model with an exponent of 5 [Bendsøe and Sigmund 2009]. We also augment our cost function to suppress checkerboard patterns using the energy term from Schumacher et al. [2015], but with a very small weight (0.01).



**Figure 7:** Top: Our method supports seamlessly combining samples from multiple scenarios to provide an aggregate analysis. Bottom: We compare our combined results to the worst-case analysis for the same pose. Worst-case both fails to predict the non-fragile legs in the stairs and drop cases as well as misses the fragile body and mouth regions in the smash and combined cases.



**Figure 8:** Topology optimization is used to create a bracket (right) from an input slab of material (left) along with appropriate boundary conditions and applied forces.

Like our forward design problem, our topology optimization scheme takes a triangle mesh with volumetric material assignment, and a usage scenario as input. Additionally, one must specify the failure threshold  $\Theta$ . The output,  $\mathbf{w}^*$ , describes which elements in the original finite element mesh are either full or have been removed to save weight. One limitation of our method is that we must constrain elements on the surface of our object to remain full ( $w_e = 1.0$ ). This gives us a non-evolving contact surface which allows us to avoid computing complicated, non-smooth derivatives of the contact forces with respect to the change in contact surface.

### 6.1 Algorithm Overview

We use the method of moving asymptotes [Svanberg 1987] to solve Equation 15. Each iteration of the optimization scheme requires the execution of the following steps:

1. Compute the cost:  $\sum_{e=1}^{N_{el}} w_e \int_{\Omega_e} \rho_e d\Omega$
2. Compute the cost gradient:  $\nabla_{\mathbf{w}} \sum_{e=1}^{N_{el}} w_e \int_{\Omega_e} \rho_e d\Omega$
3. Compute the failure probability:  $P(\tilde{\sigma} < \hat{\sigma})$
4. Compute the gradient of failure probability:  $\nabla_{\mathbf{w}} P(\tilde{\sigma} < \hat{\sigma})$

where  $\nabla_{\mathbf{w}}$  is the gradient operator taken with respect to the per-element fill ratios.

The first two steps of this process are computationally trivial and

we have already presented an efficient algorithm for evaluating  $P(\tilde{\sigma} < \hat{\sigma})$  in step three. The major remaining hurdle is evaluating  $\nabla_{\mathbf{w}} P(\tilde{\sigma} < \hat{\sigma})$  and we address this challenge below.

**Remarks on Notation** During the derivation of the failure probability gradients we often require the derivative of a matrix w.r.t. a vector. This yields a third order tensor. In an effort to keep our notation clean we avoid resorting to indicial notation. Instead, we point out to the reader that our third order tensors always have  $\mathbf{w}$  as a free dimension and are only ever multiplied with standard matrices and vectors. Hence, we use standard linear algebra notation to denote these operations which should be thought of as occurring for each  $w_e$ .

## 6.2 Failure Probability Gradients

Computing the gradient of our probability functions is a computationally difficult task given that we require the derivative with respect to each  $w_e$  where  $|\mathbf{w}|$  is equal to the total number of finite elements in our computational mesh.

The first step in making this derivative tractable is to rephrase  $\mathbf{P}(\tilde{\sigma} < \hat{\sigma})$  by normalizing with respect to the yield stress allowing us to re-express  $\mathbf{P}_{\tilde{\sigma}}(\tilde{\sigma} < \hat{\sigma}) > \Theta$  as  $\mathbf{P}_{\tilde{\sigma}}(\frac{\tilde{\sigma}}{\hat{\sigma}} < 1) > \Theta$ . Unfortunately, the standard von Mises stress (Equation 3) has been shown to have a singularity when used for topology optimization. To avoid this, we use a modified von Mises stress measure [Lee et al. 2012] which is given by

$$S^e(\boldsymbol{\sigma}^e) = w_e^{1/2} \frac{\tilde{\sigma}^e(\boldsymbol{\sigma}^e)}{\hat{\sigma}}. \quad (16)$$

We then use the whole object maximum,  $S = \max(S(\boldsymbol{\sigma}))$  to compute  $\mathbf{P}(S < 1) > \Theta$  as a stable replacement for our previous probability.

Because we use a piecewise linear representation for the CDF (Equation 14) we can easily evaluate its derivative. After normalization, our CDF becomes

$$\mathbf{P}(S < 1) = \int_0^1 \Psi(y) N^{-1} \mathbf{b} dy, \quad (17)$$

where  $N$  and  $\mathbf{b}$  are computed using  $S$ . The derivative of this equation with respect to  $\mathbf{w}$  is given by

$$\frac{\partial \mathbf{P}(S < 1)}{\partial \mathbf{w}} = \int_0^1 \Psi(y) N^{-1} \frac{\partial \mathbf{b}}{\partial \mathbf{w}} dy, \quad (18)$$

where  $\frac{\partial \mathbf{b}}{\partial \mathbf{w}}$  can be derived from Equation 12 and Equation 16 to be

$$\frac{\partial \mathbf{b}}{\partial \mathbf{w}} = \frac{1}{N_s} \sum_{i=1}^{N_s} \frac{\partial \psi_s^e}{\partial S} \frac{\partial S}{\partial \mathbf{w}}. \quad (19)$$

This requires the derivative of our shape functions, which are easy to compute, and the derivatives of our samples with respect to the fill parameters. Naively, one could use finite differencing to compute this gradient, but recall that  $|\mathbf{w}|$  is large. Since each finite difference solve would require a Finite Element solve this solution is clearly not viable. Instead we show how to drastically reduce the number of solves needed to compute this gradient using the Adjoint method [McNamara et al. 2004].

**Computing the von Mises Stress Gradient.** Our first major road block in computing  $\frac{\partial S}{\partial \mathbf{w}}$  arises because  $S = \max(S(\boldsymbol{\sigma}))$ . The max function does not have a smooth derivative and so instead we use a common substitution, the  $L^p$  norm [Lee et al. 2012]. As  $p$

approaches infinity  $L^p$  approaches max. Thus we must now evaluate

$$\frac{\partial S}{\partial \mathbf{w}} = \left[ \sum_{e=1}^{N_{el}} S^{ep} \right]^{\frac{1}{p}-1} \sum_{e=1}^{N_{el}} S^{e(p-1)} \left( \frac{\partial S^e}{\partial \mathbf{w}} (\boldsymbol{\sigma}^e) + \frac{\partial S^e}{\partial \tilde{\sigma}^e} \frac{\partial \tilde{\sigma}^e}{\partial \boldsymbol{\sigma}^e} \frac{\partial \boldsymbol{\sigma}^e}{\partial \mathbf{w}} \right), \quad (20)$$

where we have found that  $p = 8$  works well experimentally. Here all derivatives except for  $\frac{\partial \boldsymbol{\sigma}^e}{\partial \mathbf{w}}$  have straightforward analytical solutions.

**Computing the Sample Stress Gradient.** Now we arrive at the per-sample evaluation of the stress gradient. From Equation 8 we can show that

$$\begin{aligned} \frac{\partial \boldsymbol{\sigma}_i}{\partial \mathbf{w}} = & -CBK^{-1} \frac{\partial K}{\partial \mathbf{w}} \bar{U}^i \boldsymbol{\alpha} + CBK^{-1} \frac{\partial \bar{F}}{\partial \mathbf{w}} {}^i \boldsymbol{\alpha} \\ & + CB\bar{U} \frac{\partial {}^i \boldsymbol{\alpha}}{\partial \mathbf{w}} \end{aligned} \quad (21)$$

where  $\bar{U} = K^{-1} \bar{F}$ ,  $\frac{\partial K}{\partial \mathbf{w}}$  is a sparse, third order tensor representing the derivative of the finite element stiffness matrix with respect to the fill in parameters,  $\frac{\partial \bar{F}}{\partial \mathbf{w}}$  is the derivative of our reduced force basis and  $\frac{\partial {}^i \boldsymbol{\alpha}}{\partial \mathbf{w}}$  is the derivative of the sample coordinates in that space.

We can exploit the fact that our force samples are generated by a rigid body simulation, and are thus only dependent on the center-of-mass and moments-of-inertia in the object's current state. This allows us to evaluate  $\frac{\partial \bar{F}}{\partial \mathbf{w}}$  and  $\frac{\partial {}^i \boldsymbol{\alpha}}{\partial \mathbf{w}}$  using a total of 18 finite differences per sample. Because rigid body simulation of a single object is fast, and because sampling is parallel, this step is relatively efficient (see supplemental material).

**Efficient Gradient Evaluation using the Adjoint Method.** On the surface, it still appears as though evaluating Equation 20 will require a number of inverses of  $K$  on the order of the  $N_s \times N_{el}$ . However we can reduce this by considering the gradient formula as a whole, rather than as a product of individual derivatives. We spare the details for our supplemental material, but note here that we can express  $\frac{\partial \mathbf{P}(S < 1)}{\partial \mathbf{w}}$  as

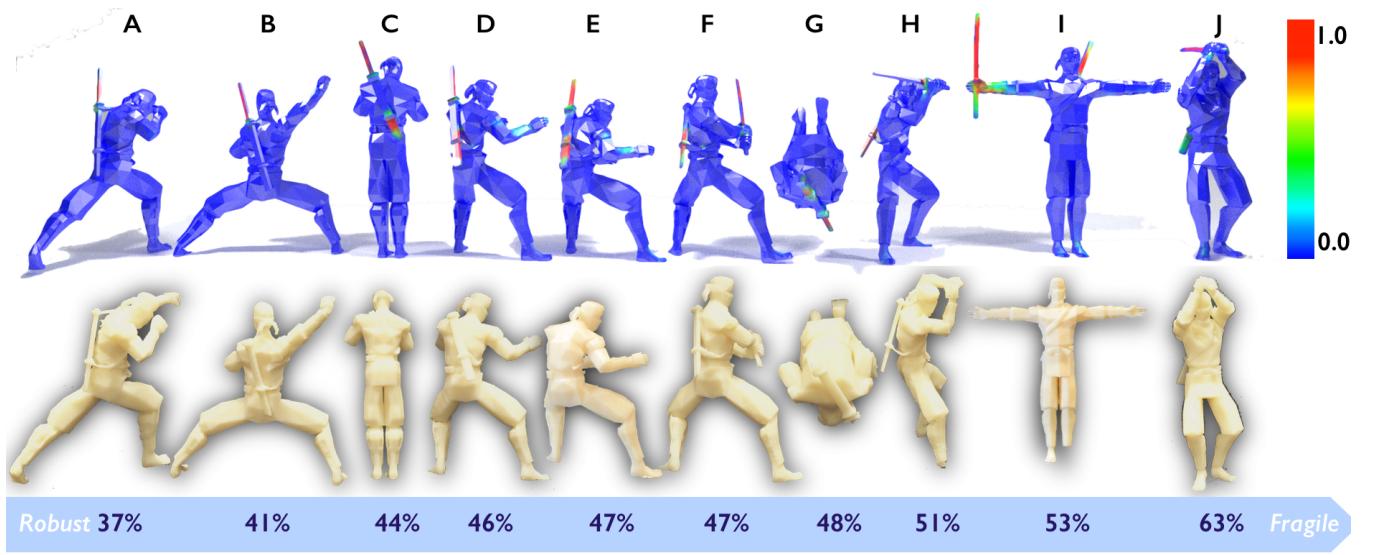
$$\frac{\partial \mathbf{P}(S < 1)}{\partial \mathbf{w}} = \sum_{i=1}^{N_s} a_i \left( \mathbf{b}_i + \frac{\partial {}^i \boldsymbol{\alpha}}{\partial \mathbf{w}} \mathbf{c}_i \right), \quad (22)$$

where  $a_i = \int_0^1 \Psi(y) dy \left[ \sum_{e=1}^{N_{el}} S^{ep} \right]^{\frac{1}{p}-1}$ ,  $\mathbf{b}_i = \sum_{e=1}^{N_{el}} S^{e(p-1)} \frac{\partial S^e}{\partial \mathbf{w}} (\boldsymbol{\sigma}^e)$  and  $\mathbf{c}_i = \sum_{e=1}^{N_{el}} S^{e(p-1)} \frac{\partial S^e}{\partial \tilde{\sigma}^e} \frac{\partial \tilde{\sigma}^e}{\partial \boldsymbol{\sigma}^e}$ .

By substituting Equation 21 into Equation 22 we can re-express the gradient as

$$\frac{\partial \mathbf{P}_{\tilde{\sigma}}}{\partial \mathbf{w}} = (K^{-1} X) : \frac{\partial K}{\partial \mathbf{w}} + (K^{-1} Y) : \frac{\partial \bar{F}}{\partial \mathbf{w}} + \mathbf{x} + \mathbf{b} \quad (23)$$

where  $X = (\sum_{i=1}^{N_s} B^T C^T \mathbf{c}_i \otimes {}^i \boldsymbol{\alpha}) \bar{U}^T$ ,  $Y = (\sum_{i=1}^{N_s} B^T C^T \mathbf{c}_i \otimes {}^i \boldsymbol{\alpha})$ ,  $\mathbf{x} = \sum_{i=1}^{N_s} \frac{\partial {}^i \boldsymbol{\alpha}}{\partial \mathbf{w}} \bar{U}^T B^T C^T \mathbf{c}_i$  and  $\mathbf{b} = \sum_{i=1}^{N_s} a_i \mathbf{b}_i$ . Notice that  $X = Y \bar{U}^T$  and  $Y \in \mathbb{R}^{n \times N_b}$  where  $N_b$  is the number of reduced space basis vectors. This means we can solve  $K^{-1} Y$  using only  $N_b$  linear solves to compute the entire gradient. This is a significant improvement from the naive solution and makes our stochastic topology optimization possible.



**Figure 9:** Computed failure probabilities for a series of ninja characters. We order the characters from most robust (left) to most fragile (right) which allows a toy designer to balance structural soundness with intended expressivity. We also show representative drop test results performed with 3D printed characters.

## 7 Results

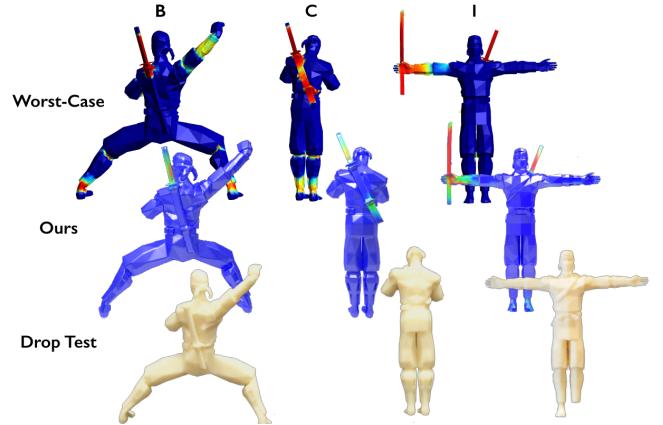
Below we review the results produced by our algorithm beginning with spatially-varying and per-object failure probabilities and concluding with some topology optimization results.

### 7.1 Forward Design Results

We illustrate a scenario where a designer needs to create a plastic toy character and wants to test different possibilities to pose the character. On the one hand, these poses should be expressive and attractive, but on the other hand they should be sturdy. Using worst case analysis will not provide a measure to distinguish the different poses as the fragile parts of the object would be the same for all poses. Using our method, we can sort the different poses according to their failure probabilities, as well as show specific areas that are more vulnerable than others (see Figure 9, and more results in the supplemental file). This allows the designer to gain insight into the effect of object configuration on reliability, and easily find the pose that has the best tradeoff between expressiveness and sturdiness.

To provide this analysis, we virtually drop each character 5000 times from a uniformly random orientation with uniformly random linear and angular velocities from a height of 1.5m. Note that in our analysis some poses are surprisingly robust such as the ninja in Figure 9A. Such an example may be disregarded by a designer if they were operating on intuition alone or guided by worst-case analysis. Figure 9 also shows representative, real-world drop test results for 3D printed realizations of each pose. In general, we observe that objects fail where predicted and never fail where the failure probability is close to 0%. Our supplemental material contains representative drop test results for two other characters, a horse and a bunny.

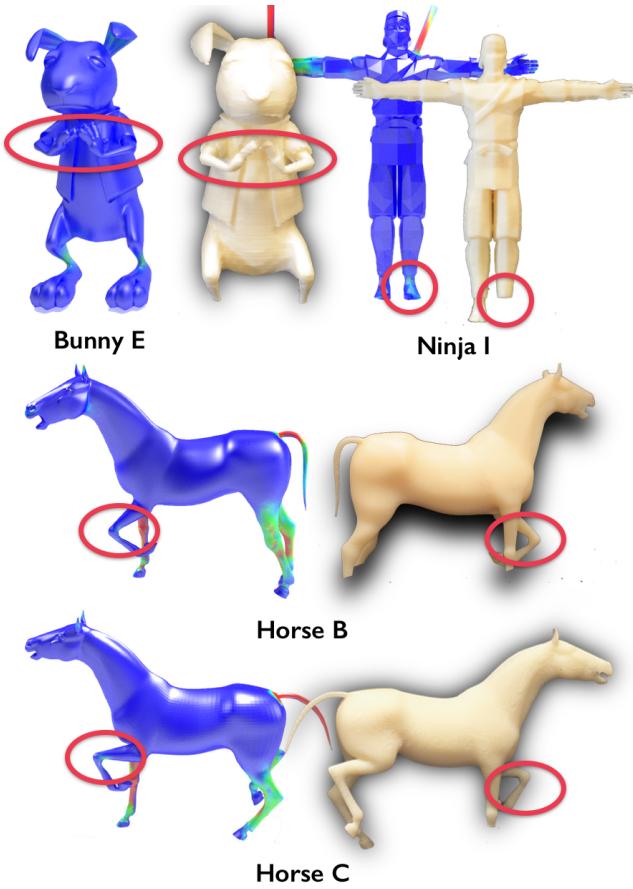
It is tempting to say that objects will always fail at thin features. However as can be seen in Figure 2, the reality is much more subtle, and our analysis allows to capture this subtlety. For example, by virtue of the head shielding the sword from impact, Ninja A (Figure 9) becomes very difficult to break. Figure 10 shows a comparison of our method to both WCA and fabricated results for three poses of the ninja. In some cases our method and WCA predict similar failure regions (Ninja C) but for other cases WCA either overpredicts (ankles and arm of Ninja B) or underpredicts (ankle of



**Figure 10:** Comparisons of our method to Worst-Case Analysis (WCA) and fabricated results. While in some cases WCA and our predictions match well (C), in other cases WCA has a tendency to over predict weakness (ankles of B) or miss subtle breaking points (ankles of I).

Ninja I) weakness. Figure 11 shows more such examples illustrating our method’s ability to detect both common and rare points of failure. The discovering of these unintuitive outcomes is where our stochastic finite element analysis excels.

Equally as important is our method’s ability to perform analysis in a real-world context. Figure 12 illustrates using different scenarios in the analysis of a plane model. The first is performed using a landing scenario (Figure 3), while the second is performed using the same drop test from 1.5m. Using our method we can predict that the object is much more robust during landing, something that is borne out by real-world experiments we carried out. We also compare our analysis to worst-case analysis [Zhou et al. 2013], which mis-predicts the reliability in both the landing and the drop test case. Here, we see the advantage of using our context-based method: a designer or engineer can get a realistic picture of the likely object performance in real-world scenario. This prevents them from making potentially unnecessary design decisions such as reinforcing the wings of this



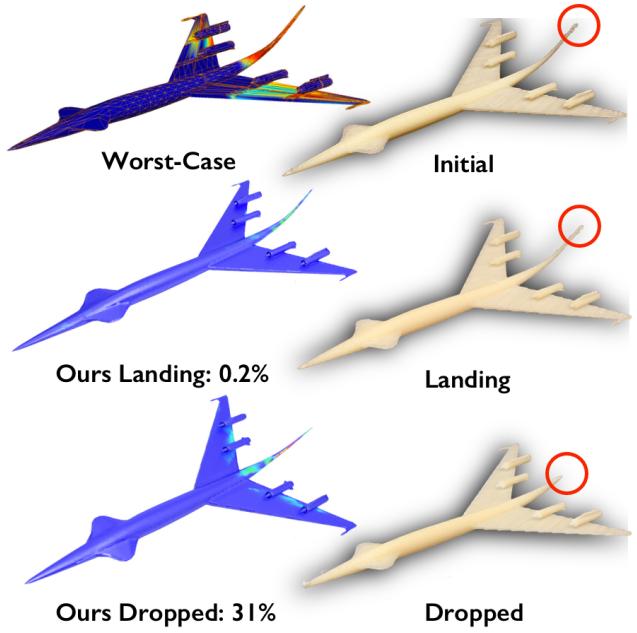
**Figure 11:** Examples of non-intuitive robust or fragile features detected by our method and validated in real-world drop tests. The thin arms of Bunny E are well protected during falls and are unlikely to break, as are the bent legs of Horses B and C. Ninja I features a rare and surprising vulnerability in the left-foot which actually caused failure during drop testing.

plane.

## 7.2 Topology Optimization

We present some of our results generated using stochastic topology optimization. First, in Figure 13 we show an example of a simple bracket. We optimized for two brackets, one with an applied, deterministic load of  $10N$  and a second using a stochastic load. The stochastic load was prescribed by sampling from two Gaussian distributions, one for vertical force (mean:  $10N$ , std:  $5N$ ) and a second for transverse forces (mean:  $0N$ , std:  $5N$ ). As expected, the deterministic optimization reduces the weight of the bracket but makes it thin as it only has to balance the vertical load. The stochastic optimization widens the bracket in order to resist these lateral loads. This shows the advantage of stochastic topology optimizations – we can succinctly describe the wide range of forces an object may experience and produce appropriately optimized designs.

Figure 14 and Figure 15 show penguin meshes that have been optimized for varying degrees of robustness using our Drop Test and Stairs scenarios. In each figure, the top row shows a more fragile penguin while the bottom row shows a more robust design. This illustrates how failure probabilities can be used to guide a topology optimization scheme, in this case to reinforce 3D prints to particular real-world loading scenarios. Figure 16 shows a 3D print of



**Figure 12:** Analysis and drop test results for a plane model. We test two cases, a landing case wherein the plane approaches the ground at a shallow angle, and a random drop test from 1.5 meters. Notice that our method accurately reflects the spatially-varying and overall reliability of the object well while Worst-Case analysis over estimates the objects fragility.

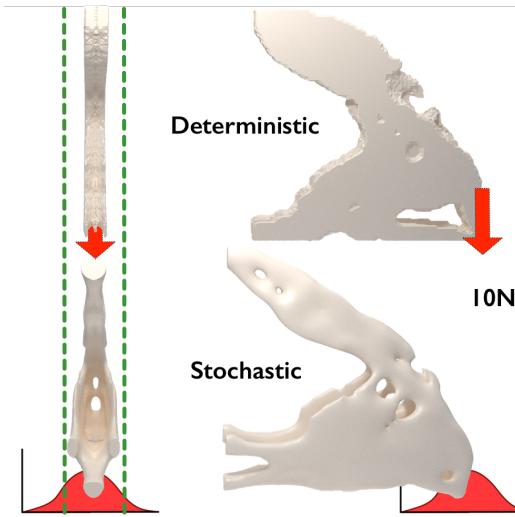
our optimized 40% failure penguin and Figure 17 shows the results of drop testing both the 50% and 40% penguins (Figure 14). We dropped each penguin from identical, randomly generated heights and orientations until a failure occurred. After 4 drops the beak on the more failure-prone penguin fractured.

## 7.3 Performance

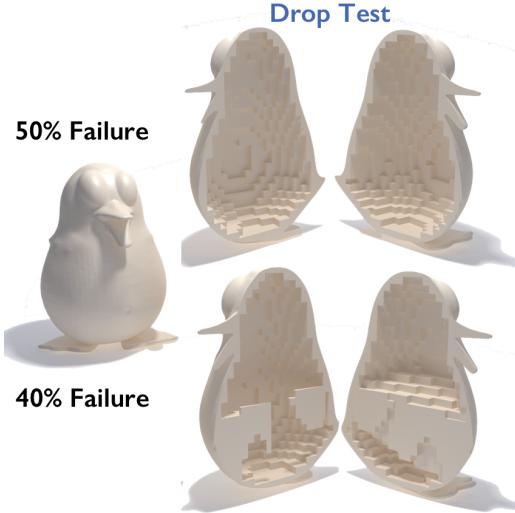
Table 2 shows the performance of our algorithm for both forward and inverse problems. Our benchmark machine was equipped with 4 AMD Opteron 6378 processors (16 cores each) and 256GB of RAM. We made moderate effort to multi-thread our code. For the forward problem, our reduced space approach accelerates the linear solver stage of our method by  $32\times$  to  $64\times$ , resulting in an  $9\times$  to  $36\times$  improvement in algorithm performance. To solve the linear systems, we used Intel's MKL Pardiso solver. Figure 18 shows the accuracy of our approach, compared to computation using all samples. At 95% variance the total relative error between spatially-varying maps is less than 7% and the probability error is less than 5%. As expected, these errors converge to zero as we increase the number of basis vectors.

Currently a large part of the runtime is taken by the probability computation itself which involves several independent, per-voxel, dense linear algebra operations. Because of its parallel nature, we feel that this stage of the method is ripe for a GPU implementation which could improve its performance drastically.

For the inverse problem, our fast gradient computation gives us a massive performance boost. Even though our examples are relatively small, they are impossible to solve with naïve methods. We abandoned a single optimization step after 12 hours of wall time but estimate the total time required to be over 11,000 hours (using  $N_s \times N_{el} \times \text{solver time}$ ). While not fast, our gradient formulation is efficient, garnering a 10,000x speedup over the naïve approach



**Figure 13:** Two brackets created using topology optimization. Surfaces are extracted isocontours from the optimized material density field. The top bracket is created using a deterministic load while the bottom is created using our stochastic method. Notice how the bottom bracket is wider to account for lateral loading.

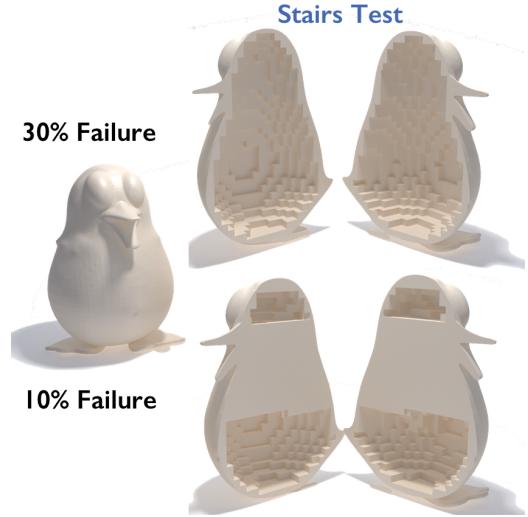


**Figure 14:** Two examples of an optimized penguin mesh using our Drop Test scenario. We use the original triangular mesh to represent the outer surface of our object and optimize the voxelized interior. Top Row: This penguin will remain intact with 50% certainty when dropped from a height of two meters. Bottom Row: A more robust penguin which will remain intact 60% of the time.

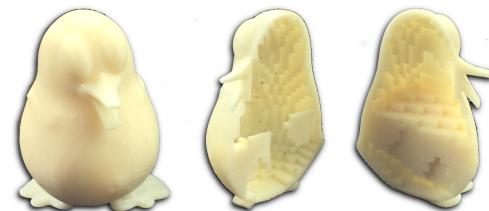
and allowing us to solve these previously intractable problems. One of the bottlenecks in our method is sampling time. This arises because the simulator we are using does not feature a straight forward method to accelerate collision detection for non-convex objects. We imagine this time can be reduced dramatically with more careful implementation of the simulation stage.

## 8 Conclusion and Discussion

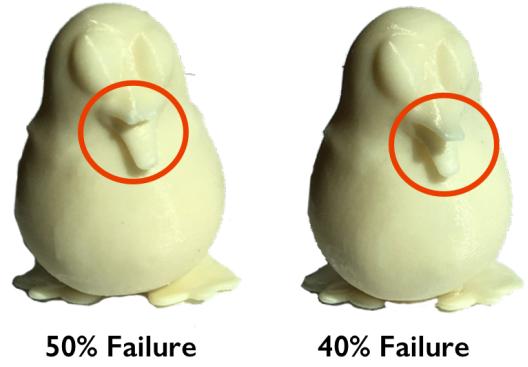
In this paper we have presented a new, context-aware, stochastic finite element method for predicting both the per-object and spatially varying probability of failure. We leverage a reduced space approach and novel gradient formulation to solve both forward and previously intractable inverse design problems. Our method opens up a new,



**Figure 15:** Two examples of an optimized penguin mesh using our Stair Test scenario. We use the original triangular mesh to represent the outer surface of our object and optimize the voxelized interior. Top Row: This penguin will remain intact with 70% certainty when dropped from a height of two meters. Bottom Row: A more robust penguin which will remain intact 90% of the time.



**Figure 16:** A 3D print of our 40% failure penguin produced using a Stratysys Fortus 250mc FDM printer.



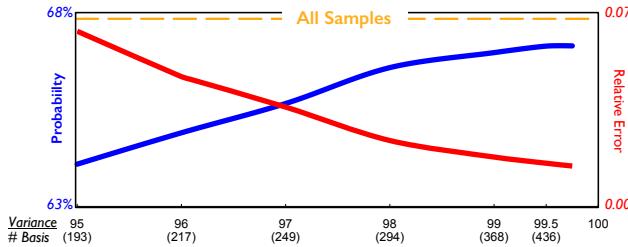
**Figure 17:** Drop Tests of our 50% and 40% failure probability penguins. Note the broken beak on the 50% failure penguin.

robust space of computational structural analysis algorithms for artists and engineers across a myriad of fields.

Our stochastic finite element method reveals a wealth of opportunities for future work which range from relaxing the assumptions made in our physical model to improving its performance further. For instance, we assume that the objects being tested only deform infinitesimally. This allows us to decouple rigid body simulation and deformable body simulation in order to accelerate computation. While the space of infinitesimally deforming objects is huge, extending the method so that it can be applied to finitely deforming

Example	Grid Size	Rank	Total	Sampling	Linear Solve	Probability and Gradient	Speedup
Ninja (C) - naive	$42 \times 116 \times 40$		9,618	504	8,929	185	1×
Ninja (C) - ours	$42 \times 116 \times 40$	139	1027	504	275	185	9.4×
Ninja (E) - naive	$50 \times 96 \times 59$		14,013	635	13,172	206	1×
Ninja (E) - ours	$50 \times 96 \times 59$	128	1301	635	382	206	11×
Ninja (H) - naive	$49 \times 99 \times 58$		19,569	835	18,629	105	1×
Ninja (H) - ours	$49 \times 99 \times 58$	130	1474	835	462	105	13×
Rabbit (G) - naive	$50 \times 108 \times 51$		17,561	770	16,660	132	1×
Rabbit (G) - ours	$50 \times 108 \times 51$	150	1311	770	333	132	13×
Rabbit (H) - naive	$40 \times 100 \times 60$		17,685	277	16,639	153	1×
Rabbit (H) - ours	$40 \times 100 \times 60$	146	852	277	316	153	21×
Rabbit (I) - naive	$55 \times 105 \times 46$		15,504	672	14,610	222	1×
Rabbit (I) - ours	$55 \times 105 \times 46$	155	1434	672	459	222	11×
Plane - naive	$75 \times 23 \times 133$		14,883	138	14,715	30	1×
Plane - ours	$75 \times 23 \times 133$	146	411	138	229	30	36×
Horse (B) - naive	$33 \times 80 \times 86$		13,566	353	13,004	209	1×
Horse (B) - ours	$33 \times 80 \times 86$	157	847	353	218	209	16×
Horse (F) - naive	$36 \times 104 \times 142$		71,782	1155	70,310	317	1×
Horse (F) - ours	$36 \times 104 \times 142$	141	4399	1155	2730	317	16×
<b>Inverse:</b> Penguin (40%) - naive, 1 step	$26 \times 34 \times 28$		est. 11,754 hrs	51	570	> 12 hrs (est. 11,754 hrs)	1×
<b>Inverse:</b> Penguin (40%) - ours, 1 step	$26 \times 34 \times 28$		3,713	51	57.3	65 min	10,849×

**Table 2:** Performance of our algorithm on representative forward and inverse problems. All timings are given in seconds unless explicitly stated. For inverse problems we record the time for a single optimization step. We also report the rank of the reduced space (number of basis vectors).



**Figure 18:** Variance test for a rabbit model. As the reduced space is increased to capture more of the variance (number of basis vectors shown in parentheses), the total breakage percentage (blue solid line) approaches the brute force, Monte Carlo analysis result (horizontal line at top). Correspondingly, The relative L2 error of the spatially varying breakage probability (red dashed line) tends toward zero.

objects would further broaden its scope to objects such as soft casts for athletes or foam-latex props and costumes used in feature films.

Our method also requires distributions of initial conditions to initialize rigid body sampling via simulation. While not strictly a limitation of our method, acquiring this data is time consuming and this makes adding new analysis cases a labor intensive process. Exploiting computer vision techniques to automatically capture and generate these distributions would be of great practical benefit. Building databases of such distributions would also help expedite the adoption of techniques such as ours. There is also the question of how to best evaluate the output probabilities produced by stochastic analysis methods. Statistically meaningful validation would require 3D printing hundreds or thousands of models, something which is cost and labor prohibitive even on the cheapest 3D printers. More involved manufacturing procedures would only increase this burden. Devising new experimental methodologies for validation of stochastic methods is thus necessary to push research in this area forward.

Another barrier to widespread adoption is performance. While our method is efficient and can certainly solve offline design problems in reasonable amounts of time, larger datasets and interactive applications are still beyond our reach. There are two main reasons for this, first solving very large, sparse linear systems to high levels of accuracy is still slow, though progress is being made to accelerate this [Wu et al. 2016]. In conjunction with exploring the use of such fast linear solvers, we would like to explore using model reduction in

more phases of the algorithm, both to reduce the number of degrees-of-freedom used for topology optimization and to accelerate each structural solve. Second, aggregating probabilities on the mesh is surprisingly time consuming. We would like to explore GPU-based algorithms for accelerating this step of our stochastic method.

Finally, both the graphics and engineering communities are just beginning to understand how to control the appearance of topology optimized models and to constrain them to fabricable designs. Adding these appearance and fabrication constraints to our method is also a promising area of future work.

## Acknowledgements

Thanks to our Reviewers for helpful suggestions, Disney Research (esp. Jessica Hodgins for guidance and David Zimmerman for I.T. help), Doug James for use of a workstation, Qingnan Zhou for WCA code, and Desai Chen and Javier Ramos for fabrication help. A. Shamir and D. Dror were partly supported by ISF grants 2216/15 and 324/11. T. Langlois was partly supported by NSF grant DGE-1144153.

## References

- ANG, A. H.-S., AND TANG, W. H. 2007. *Probability concepts in engineering planning and design*. Wiley.
- BÄCHER, M., WHITING, E., BICKEL, B., AND SORKINE-HORNUNG, O. 2014. Spin-it: Optimizing moment of inertia for spinnable objects. *ACM Trans. Graph.* 33, 4 (July), 96:1–96:10.
- BARBIĆ, J., AND JAMES, D. L. 2005. Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM Trans. Graph.* 24, 3 (July), 982–990.
- BELYTSCHKO, T., LIU, W. K., MORAN, B., AND ELKHODARY, K. 2013. *Nonlinear finite elements for continua and structures*. John Wiley & Sons.
- BENDSØE, M. P., AND SIGMUND, O. 2009. *Topology Optimization*. World Scientific.
- CHEN, S., CHEN, W., AND LEE, S. 2010. Level set based robust shape and topology optimization under random field uncertainties. *Structural and Multidisciplinary Optimization* 41, 4, 507–524.

- COUMANS, E., ET AL. 2015. Bullet physics library. *Open source: bulletphysics.org*.
- CRAPO, H., AND WHITELEY, W. 1989. The geometry of rigid structures. *Encyclopedia of Mathematics and its Applications*.
- CRAPO, H. 1979. Structural rigidity. *Structural Topology 1*, 26–45.
- DER KIUREGHIAN, A., AND KE, J.-B. 1987. The stochastic finite element method in structural reliability. In *Stochastic Structural Mechanics*, Y. Lin, G. Schuller, and P. Spanos, Eds., vol. 31 of *LNE*. Springer Berlin Heidelberg, 84–109.
- DUMAS, J., LU, A., LEFEBVRE, S., WU, J., AND DICK, C. 2015. By-Example Synthesis of Structurally Sound Patterns. *ACM Trans. Graph.* 34, 4 (July).
- EIERMANN, M., ERNST, O., AND ULLMANN, E. 2007. Computational aspects of the stochastic finite element method. *Computing and Visualization in Science 10*, 1, 3–15.
- EVGRAFOV, A., PATRIKSSON, M., AND PETERSSON, J. 2003. Stochastic structural topology optimization: existence of solutions and sensitivity analyses. *ZAMM 83*, 7, 479–492.
- FARAVELLI, L., AND BIGI, D. 1990. Stochastic finite elements for crash problems. *Structural Safety 8*, 1, 113 – 130.
- GOUNARIS, G., AND DIMAROGONAS, A. 1988. A finite element of a cracked prismatic beam for structural analysis. *Computers & Structures 28*, 3, 309–313.
- HALKO, N., MARTINSSON, P.-G., AND TROPP, J. A. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review 53*, 2, 217–288.
- HUGHES, T. J. 2012. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation.
- LEE, E., JAMES, K., AND MARTINS, J. 2012. Stress-constrained topology optimization with design-dependent loading. *Structural and Multidisciplinary Optimization 46*, 5, 647–661.
- LIU, W. K., BELYTSCHKO, T., AND MANI, A. 1986. Probabilistic finite elements for nonlinear structural dynamics. *Comput Method Appl M 56*, 1, 61 – 81.
- LIU, W. K., BELYTSCHKO, T., AND MANI, A. 1986. Random field finite elements. *International Journal for Numerical Methods in Engineering 23*, 10, 1831–1845.
- LU, L., SHARF, A., ZHAO, H., WEI, Y., FAN, Q., CHEN, X., SAVOYE, Y., TU, C., COHEN-OR, D., AND CHEN, B. 2014. Build-to-last: Strength to weight 3d printed objects. *ACM Trans. Graph.* 33, 4 (July), 97:1–97:10.
- MA, F. 1987. Extension of second moment analysis to vector-valued and matrix-valued functions. *International Journal of Non-Linear Mechanics 22*, 3, 251 – 260.
- MAHADEVAN, S., AND HALDAR, A. 1991. Practical random field discretization in stochastic finite element analysis. *Structural Safety 9*, 4, 283 – 304.
- MARTÍNEZ, J., DUMAS, J., LEFEBVRE, S., AND WEI, L.-Y. 2015. Structure and appearance optimization for controllable shape design. *ACM Transactions on Graphics (TOG) 34*, 6, 229.
- MATTHIES, H. G., AND KEESE, A. 2005. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput Method Appl M 194*, 1216, 1295 – 1331.
- MAUTE, K. 2014. Topology optimization under uncertainty. In *Topology Optimization in Structural and Continuum Mechanics*, G. Rozvany and T. Lewiski, Eds., vol. 549 of *CISM*. Springer Vienna, 457–471.
- MCCORMAC, J. C., AND ELLING, R. E. 1984. *Structural analysis*. Harper & Row New York.
- MCNAMARA, A., TREUILLE, A., POPOVIĆ, Z., AND STAM, J. 2004. Fluid control using the adjoint method. *ACM Trans. Graph.* 23, 3 (Aug.), 449–456.
- MELOSH, R. J. 1974. Finite element analysis of automobile structures. Tech. rep., SAE Technical Paper.
- MUSIALSKI, P., AUZINGER, T., BIRSAK, M., WIMMER, M., AND KOBBELT, L. 2015. Reduced-order shape optimization using offset surfaces. *ACM Trans. Graph.* 34, 4 (July), 102:1–102:9.
- MUSIALSKI, P., HAFNER, C., RIST, F., BIRSAK, M., WIMMER, M., AND KOBBELT, L. 2016. Non-linear shape optimization using local subspace projections. *ACM Trans. Graph.* 35, 4 (July), 87:1–87:13.
- PRÉVOST, R., WHITING, E., LEFEBVRE, S., AND SORKINE-HORNUNG, O. 2013. Make it stand: Balancing shapes for 3d fabrication. *ACM Trans. Graph.* 32, 4 (July), 81:1–81:10.
- SCHUMACHER, C., BICKEL, B., RYS, J., MARSCHNER, S., DARAIO, C., AND GROSS, M. 2015. Microstructures to control elasticity in 3d printing. *ACM Trans. Graph.* 34, 4.
- SCHULLER, G. 2006. Developments in stochastic structural mechanics. *Archive of Applied Mechanics 75*, 10–12, 755–773.
- SIGMUND, O. 1997. On the design of compliant mechanisms using topology optimization\*. *J Struct Mech 25*, 4, 493–524.
- STAVA, O., VANEK, J., BENES, B., CARR, N., AND MĚCH, R. 2012. Stress relief: Improving structural strength of 3d printable objects. *ACM Trans. Graph.* 31, 4 (July), 48:1–48:11.
- STEFANOU, G. 2009. The stochastic finite element method: Past, present and future. *Comput Method Appl M 198*, 912, 1031 – 1051.
- SVANBERG, K. 1987. The method of moving asymptotes- a new method for structural optimization. *International journal for numerical methods in engineering 24*, 2, 359–373.
- TOMLOW, J. 1989. The model–antoní gaudí hanging model and its reconstruction—new light on the design of the church of colonia güell. *Institute for Lightweight Structures (IL): University of Stuttgart. Germany*, 34.
- UMETANI, N., AND SCHMIDT, R. 2013. Cross-sectional structural analysis for 3d printing optimization. In *SIGGRAPH Asia 2013 Technical Briefs*, ACM, New York, NY, USA, SA ’13, 5:1–5:4.
- WANG, W., WANG, T. Y., YANG, Z., LIU, L., TONG, X., TONG, W., DENG, J., CHEN, F., AND LIU, X. 2013. Cost-effective printing of 3d objects with skin-frame structures. *ACM Trans. Graph.* 32, 6 (Nov.), 177:1–177:10.
- WU, J., DICK, C., AND WESTERMANN, R. 2016. A system for high-resolution topology optimization. *IEEE T Vis Comput Gr* 22, 3 (Mar.), 1195–1208.
- ZHANG, Z., JIANG, C., HAN, X., HU, D., AND YU, S. 2014. A response surface approach for structural reliability analysis using evidence theory. *Advances in Engineering Software 69*, 37 – 45.
- ZHOU, Q., PANETTA, J., AND ZORIN, D. 2013. Worst-case structural analysis. *ACM Trans. Graph.* 32, 4 (July), 137:1–137:12.