NISS

Bayesian Validation of a Computer Model for Vehicle Crashworthiness

M. J. Bayarri, J. O. Berger, M. C. Kennedy,
A. Kottas, R. Paulo, J. Sacks,
J. A. Cafeo, C. H. Lin, J. Tu

Technical Report Number 163
July 2005

National Institute of Statistical Sciences
19 T. W. Alexander Drive
PO Box 14006
Research Triangle Park, NC 27709-4006
www.niss.org

Bayesian Validation of a Computer Model for Vehicle Crashworthiness*

M.J. Bayarri, J.O. Berger, M.C. Kennedy, A. Kottas, R. Paulo, J. Sacks
and
J.A. Cafeo, C.H. Lin, J. Tu †

July 23, 2005

Abstract

A key question in evaluation of computer models is *Does the computer model adequately represent reality?* A complete Bayesian approach to answering this question is developed for the challenging practical context in which the computer model (and reality) produce functional data. The methodology is particularly suited to treating the major issues associated with the validation process: quantifying multiple sources of error and uncertainty in computer models; combining multiple sources of information; and being able to adapt to different – but related – scenarios through hierarchical modeling. It is also shown how one can formally test if the computer model reproduces reality.

The approach is illustrated through study of a computer model developed to model vehicle crashworthiness.

Keywords and phrases: Computer models; Validation; Functional data; Bias; Bayesian analysis; Hierarchical modeling; Hypothesis testing.

1 Introduction

1.1 Validation framework

Bayarri et al. (2005b) described a general framework for validation of complex computer models and applied the framework to two (relatively) simple examples. Here we consider an extension of the framework to handle functional data, allow hierarchical modeling for dealing with related computer models, and enable formal testing of the validity of a computer model. The extensions are motivated

*This research was supported in part by grants from General Motors and the National Science Foundation (Grant DMS-0073952) to the National Institute of Statistical Sciences.

†Affiliations: Bayarri — Universitat de València; Berger — Duke University; Kennedy — University of Sheffield; Kottas — University of California, Santa Cruz; Paulo — University of Bristol; Sacks — National Institute of Statistical Sciences; Cafeo, Lin and Tu — General Motors.

by methodological needs in analyzing the CRASH computer model of vehicle crashworthiness, described in the next section.

The framework described in Bayarri et al. (2005b) consists of the following six-step procedure:

Step 1. Specify the inputs to the computer model and the unknown model parameters, along with associated uncertainties or ranges—the Input/Uncertainty (I/U) map.

Step 2. Determine criteria under which the model is to be evaluated.

Step 3. Design and perform validation experiments (obtain computer model runs, and field data from the real process).

Step 4. If the computer model is computationally slow, develop a fast approximation.

Step 5. Perform output analysis, sensitivity analysis and compare computer model output with field data. This includes in particular: *i)* modeling the relation of reality to the computer model; *ii)* statistical modeling of the data (computer runs and field); *iii)* tuning/calibrating model input parameters based on the field data; *v)* updating uncertainties in the parameters (given the data); and *vi)* assessing accuracy of predictions given the data

Step 6. Feedback information into current validation exercise and feed-forward information into future validation activities.

The first four steps in the strategy are common to virtually all validations of computer models. In some cases the design step (Step 3) may be absent because data are collected in an informal way. Significant aspects of the analyses required for Step 5 will ordinarily vary from problem to problem. This paper is an extension of Bayarri et al. (2005b), treating the following important – and technically challenging – problems that greatly widen the applicability of the six-step strategy.

Problem 1 – Functional Output: In Bayarri et al. (2005b), the examples considered involved real-valued model outputs. The output of CRASH, however, is a time-dependent function (or functions), as indicated in Figure 1. Thus, in validating the computer model with field data, one must compare functions, a more difficult enterprise.

Problem 2 – Hierarchical Modeling: A common problem in computer modeling is that codes (and field experiments) can be run under differing conditions, where the differences are not completely quantifiable and where data may be scant – or even lacking – for some of the conditions. In CRASH, this arises when treating collisions of different barriers; there are few data for center pole collision or for right-angle collision but there are reasonable amounts of data for straight-frontal and left-angle collision (see Table 1). By use of hierarchical modeling, data from the various experiments can be combined to overcome scarcity and obtain useful predictions. The engineering implications of this are substantial:

- Prediction, inference and validation can be done even for situations where data may be scant for some of the particular conditions of interest.
- The methods can be used for prediction under untried conditions, as long as the new conditions are deemed to be compatible with the hierarchical modeling assumptions.

Problem 3 – Testing Model Correctness: The validation question initially asked by computer modelers is almost invariably "Can you establish that the computer model is correct?" In Section 6 it is shown how one can formally conduct a test to answer this question. The result of the test will virtually always be that there is conclusive evidence that the computer model is not correct, but formally providing an answer to this question will often make it easier to convince the modelers to move on to the more interesting question of assessing accuracy of predictions using the computer model.

1.2 The computer model for vehicle crashworthiness

The CRASH computer model simulates the effect of a collision of a vehicle with different types of barriers. Proving ground tests with prototype vehicles must ultimately be made to meet mandated standards for crashworthiness, but the computer model plays an integral part in the design of the vehicle to assure crashworthiness before manufacturing the prototypes. How well the model performs is therefore crucial to the design process.

CRASH is implemented using a non-linear dynamic analysis (commercial) code, LS-DYNA, using a finite element representation of the vehicle. The main focus is on the velocity changes after impact at key positions on the vehicle, such as the driver seat and radiator.

Geometric representation of the vehicle and the material properties play critical roles in the behavior of the vehicle after impact and the necessary detailing of these inputs leads to very time consuming computer runs (from 1 to 5 days on a standard work station). Obtaining field data involves crashing of full vehicles, so that field data is obviously also limited. Studying CRASH is thus inherently data-limited – both in terms of computer runs and field data – presenting a basic challenge to assessing the validity of the model.

There are many variables and sources of uncertainty in the vehicle manufacturing process and proving ground test procedures that, in turn, induce uncertainties in the test results. The acceleration and velocity histories (i.e., the acceleration and velocities over the initial 100 millisecond time interval of the crash) of two production vehicles of the same type, subjected to 30mph zero degree rigid barrier frontal impact tests, are shown in Figure 1. These test results indicate the variability inherent in the field data.

Details of the computer model and sources of uncertainty are given in Appendix A.

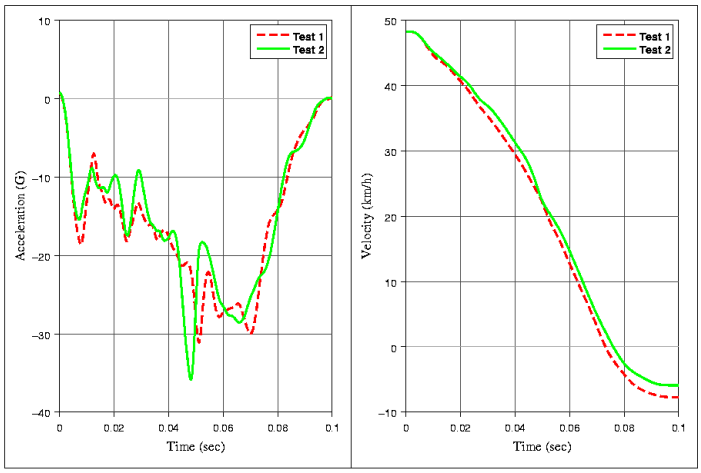

Figure 1: Acceleration and velocity pulses in the occupant compartment from 30mph zero degree rigid barrier frontal impact tests for two production vehicles of the same type.

1.3 Background and motivation

General discussions of the entire Validation and Verification process can be found in Roache (1998), Oberkampf and Trucano (2000), Cafeo and Cavendish (2001), Easterling (2001), Pilch et al. (2001), Trucano et al. (2002), and Santner et al. (2003). We focus here on the last stage of the process: that of assessing the accuracy of the computer model in predicting reality, and in using both the computer model and field data to make predictions, especially in new situations.

Because, as mentioned above, a computer model can virtually never be said to be a completely accurate representation of the real process being modelled, the relevant question is "Does the model provide predictions that are accurate enough for the intended use of the model?" Thus predictions need to come with what were called *tolerance bounds* in Bayarri et al. (2005b), indicating the magnitude of the possible error in prediction.

This focus on giving tolerance bounds, rather than stating a yes/no answer as to model validity, arises for three reasons:

1. Models rarely give highly accurate predictions over the entire range of inputs of possible interest, and it is often difficult to characterize regions of accuracy and inaccuracy. For instance, we will see that, for certain inputs, the computer model of vehicle crashworthiness yields quite accurate predictions, while for others it does not.

2. The degree of accuracy that is needed can vary from one application of the computer model to another, which again will be illustrated by different uses of CRASH.

3. Tolerance bounds account for *model bias*, the principal symptom of model inadequacy; accuracy of the model cannot simply be represented by a variance or standard error.

The key components of the approach outlined here are the use of Gaussian process response-surface approximations to a computer model, following on work in Sacks et al. (1989), Currin et al. (1991), Welch et al. (1992), and Morris et al. (1993), and introduction of Bayesian representations of model bias and uncertainty, following on work in Kennedy and O'Hagan (2001) and Kennedy et al. (2002). A related approach to Bayesian analysis of computer models is that of Craig et al. (1997), Craig et al. (2001), Goldstein and Rougier (2003) and Goldstein and Rougier (2004), which focus on utilization of linear Bayes methodology to address the problem.

1.4 Overview

Section 2 outlines the key elements of the problem, and discusses the basic strategy that is utilized to deal with functional data. Approximation of the computer model is considered in Section 3, while the Bayesian validation analysis is given in Section 4. Section 5 introduces the hierarchical methodology for dealing with related scenarios (differing barrier collisions in CRASH), while Section 6 considers the formal testing of model validity. Most of the details of the computations are relegated to the Appendix.

2 Key Elements of the Problem

2.1 Inputs

The inputs to the computer model will be denoted by a vector $\boldsymbol{x}$. In the case of CRASH, the two key inputs are $x_1 =$ *velocity at impact* and *barrier type*. The methodology from Bayarri et al. (2005b) that we are adopting does not, however, accept qualitative inputs such as barrier type. Hence, to deal with the full range of barrier types, we will resort to the hierarchical analysis.

Three of the barrier types (left angle, straight frontal, and right angle) can be converted into the quantitative input $x_2 =$ *angle of impact*. (One might think that left and right angles should be symmetric, but vehicles are not symmetric.) For most of the paper we will analyze only the data for straight frontal collision (as this was the most extensive data set), and will then be using $\boldsymbol{x}$ to just represent *impact velocity*, but in Section 5.2 we also consider angle and so $\boldsymbol{x}$ will represent the pair (*impact velocity, angle*) in that section.

In CRASH, the selected inputs for running the computer model and collecting field data were chosen for reasons other than conducting a model validation (by engineers, previous to the involvement of the authors), and are given in Table 1; note that replicates exist of the field data, which is highly useful for model validation.

Each computer run or field test resulted in acceleration data curves as in Figure 1, corresponding to various locations on the vehicle. (For the field data, these acceleration curves were from sensors placed at various locations.) The primary output of interest, and that which we shall focus on here, is the relative velocity of the "Sensing and Diagnostic Module", SDM, situated under the driver's

Impact velocity (km/h) used in model	barrier type	Impact velocity (km/h) of field tests
19.3	straight frontal	19.3
25.5	straight frontal	25.5
28.9	straight frontal	28.9
32.1	straight frontal	32.1
35.3	straight frontal	35.3
38.4	straight frontal	38.4
41.3	straight frontal	41.3, 41.3
49.3	straight frontal	49.4, 49.2, 49.4, 49.3, 49.3, 49.4
56.4	straight frontal	56.4
22.5	left angle	22.5
32.2	left angle	32.2
40.2	left angle	40.2, 41.4, 41.5
41.9	left angle	41.9
49.3	left angle	49.5, 49.2
56.2	left angle	56.2
57.3	left angle	57.3
28.9	right angle	28.9
31.9	right angle	31.9
41.7	right angle	41.7, 41.8
48.3	right angle	48.3
19.3	center pole	19.3
25.5	center pole	25.5
32.0	center pole	32.0
36.8	center pole	36.8
40.3	center pole	40.3
48.6	center pole	48.6

Table 1: Data is available at these inputs.

seat, relative to a free-flight dummy. This relative velocity is obtained by subtracting the impact velocity v from the actual SDM velocity (it being assumed that the dummy maintains velocity v over the time interval of interest). The resulting functions vary (at least theoretically) between 0 at the time of impact $t = 0$ and $-v$ at the time the vehicle is stationary.

The collection of all inputs for which the computer model was exercised will be denoted

$$D^M = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^L\}.$$

These inputs are given in Table 1. Note that L ranges from 4 to 9 depending on the barrier type (or is 17 when the angle is used to represent the first three barrier types). The collection of all

inputs to field trials (proving ground crashes of vehicles) will be denoted

$$D^F = \{\boldsymbol{x}^{*1}, \ldots, \boldsymbol{x}^{*l}\},$$

where there could be repeat inputs (i.e., replications in the field trials). These are also given in Table 1 and note that l ranges from 5 to 15 depending on the barrier type (or is 30 when the angle is used to represent the first three barrier types).

The first step in the model validation process is to construct the I/U Map, indicating the main uncertainties in the computer model or its inputs. Indeed, this exercise was carried out by experts, the results of which are given in Appendix A. The key uncertainties only become relevant, however, when experimentation can be performed over the range of uncertainties. In our situation, only one vehicle type was tested, and the inputs *impact velocity* and *barrier type* did not have uncertainties associated with them, so we did not need to utilize the I/U map in the analysis (except for recognizing that impact velocity and barrier type are the crucial inputs).

2.2 Incorporating functional outputs

There are several possible approaches that can be taken to adapt the methods of Bayarri et al. (2005b) to the setting of functional data of the type in Figure 1.

One common approach is to represent the functions that arise through a basis expansion (e.g., a polynomial expansion), taking only a finite number of terms of the expansion to represent the function. The coefficients of the terms in this expansion would then be viewed as a collection of real-valued output functions, with the methodology from Bayarri et al. (2005b) being applied to each. Examples can be found in Bayarri et al. (2005a), where a wavelet representation is used for 'rough' output functions.

In our model, the functional data for velocity (which is the primary interest) is rather smooth and so we turn instead to the most direct possibility, which is to combine t with $\boldsymbol{x}$, thus enlarging the input space to $\boldsymbol{z} = (\boldsymbol{x}, t)$. Since we only consider discrete input values here, we further must pretend that we have only observed the function at a discrete set of points, $D^T = \{t_1, \ldots, t_T\}$. In essence, we are thus 'throwing away available data.' However, it is clear that, if T is chosen large enough and the points at which we record the function are chosen well , then the function values at these T points will represent the entire function very well.

The price to be paid with this approach is that the number of observations becomes much larger; at each input value $\boldsymbol{x}$ in the original data set, there are now T time points that must also be included as inputs. The total number of inputs thus becomes $m \equiv LT$ for the model runs and $n \equiv lT$ for the field runs. Computational complexity grows rapidly with the number of inputs so, at first sight, this approach is untenable.

Luckily, if we choose the same set, D^T, of time points for each of the $\boldsymbol{x}$ inputs in the model runs or field runs, and if we make a reasonable simplifying assumption as to the nature of the

probabilistic model for the processes, a considerable simplification is effected that reduces the computational burden to something like the sum of the burdens for $(L + l)$ and T inputs. This is shown in Section 3.

Since we assume that the functions are discretized at the same points in D^T for all $\boldsymbol{x}$ in the data, the overall design spaces (the sets of $(\boldsymbol{x}, t)$ points at which model-run and field data are obtained) can be written as the products $D^M \times D^T$ and $D^F \times D^T$.

In the CRASH example, $D^T = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65\}$ms was chosen. More points were chosen in the region $t < 20$ms since information from this region is more important in estimating the primary quantity of interest, CRITV, that will be discussed shortly. Times above 65ms are unimportant for the same reason, and so there was no need to include such points in the design. These 19 points provide a good compromise between adequate representation of the function (for the desired purposes) and numerical efficiency.

2.3 Data

As discussed above, the data we subsequently analyze is the relative SDM velocity in the field runs and computer model runs over the (augmented) input domains $D^F \times D^T$ and $D^M \times D^T$. These data will be denoted $\boldsymbol{y}^M \equiv \{y^M(\boldsymbol{x}, t) : \boldsymbol{x} \in D^M, t \in D^T\}$ and $\boldsymbol{y}^F \equiv \{y^F(\boldsymbol{x}, t) : \boldsymbol{x} \in D^F, t \in D^T\}$, being vectors of length m and n, respectively, constructed from lexicographic ordering of the inputs.

2.4 Evaluation criteria

Computer models are typically used for many purposes, so overall fidelity to reality of the computer model predictions is an issue of considerable interest. Hence we will be interested in comparing the entire function predicted by the computer model with the function observed from the field data, over a wide range of inputs. Unfortunately, computer models are virtually never accurate predictors over the full range of inputs, and so assessing the accuracy of the computer model for particular features of interest is also important.

One such interesting feature that we will consider is the SDM velocity calculated 30ms before the time the SDM displacement (relative to the free-flight dummy), DISP, reaches 125mm. Call this quantity CRITV. The airbag takes around 30ms to fully deploy, which is why this particular evaluation criterion, CRITV, is important. Our analysis takes account of the dependence between displacement and velocity (displacement is the integral of velocity) by working with the probability distribution of the velocity and then finding the implied distribution of the displacement; it is a useful feature of the Bayesian approach we adopt that dealing with composite criteria, such as this, is no more difficult than dealing directly with the outputs.

The process we follow can be adapted to treat other evaluation criteria such as: *i)* time at which SDM displacement reaches 125mm; *ii)* SDM velocity when SDM displacement reaches 250mm and 350mm. On the other hand, the evaluation criterion 'Velocity at the center of the radiator, RDC,

30ms before SDM displacement reaches 125mm' poses different issues because it requires a combined analysis of the functional data from two sensors, one located at the radiator center, the other under the driver's seat. This would require joint analysis of multivariate output.

3 Approximating the Computer Model

The CRASH model can take several days to run, and a faster approximation is needed for the statistical analysis (which in our case is done via Markov Chain Monte Carlo simulation, requiring thousands of computer model runs).

We denote the (now scalar) output of the model at input $\boldsymbol{z} = (\boldsymbol{x}, t)$ by $y^M(\boldsymbol{z})$. An expensive computer model is only run at some few inputs (here $\boldsymbol{z} \in D^M \times D^T$) and should thus be viewed as an unknown function at all other values of $\boldsymbol{z}$.

The goal is to approximate $y^M(\boldsymbol{z})$ by a function $\hat{y}^M(\boldsymbol{z})$, to be called the *model approximation*, which is much easier to compute. In addition, it is desirable to have a variance function $V^M(\boldsymbol{z})$ that measures the accuracy of $\hat{y}^M(\boldsymbol{z})$ as an estimate of $y^M(\boldsymbol{z})$. A response surface approach that achieves both these goals is the Gaussian process response surface approximation (GASP), described in Sacks et al. (1989) and Kennedy and O'Hagan (2001). Using the notation in Bayarri et al. (2005b), we thus assume that $y^M(\boldsymbol{z})$ has a prior distribution given by a Gaussian process with mean $\mu(\cdot)$, correlation function $c^M(\cdot,\cdot)$, and unknown precision (1/variance)= λ^M, to be denoted by GP($\mu(\cdot), \frac{1}{\lambda^M} c^M(\cdot,\cdot)$).

Process mean: In the spotweld example studied in Bayarri et al. (2005b), the mean of the Gaussian process was a constant. Here, however, there is a clear trend in the velocity output as can be seen in Figure 1 (although recall that we will be considering the relative velocity, found by subtracting the initial impact velocity, x_1). Since the relative velocity thus starts at 0 and declines to something near $-x_1$, it is reasonable to introduce a term that reflects this trend, the simplest being of the form $\mu^M x_1\, t$, with μ^M being unknown. (An intercept term is not needed since the relative velocity is 0 at $t = 0$).

Process correlation: We choose this to be of the form

$$c^M(\boldsymbol{z}, \boldsymbol{z}^\star) = \exp\left(-\sum_{j=1}^{d} \beta_j^M |z_j - z_j{}^\star|^{\alpha_j^M} \right). \tag{3.1}$$

Here, d is the number of coordinates in $\boldsymbol{z}$: either 2 or 3 for CRASH (with inputs augmented by t), depending on whether *angle* is included as an input. The α_j^M are numbers between 1 and 2, and the β_j^M are in $[0, \infty)$. The product form of the correlation function (each factor is itself a correlation function in one-dimension) helps the computations made later. Prior beliefs about the smoothness properties of the function will affect the choice of $\boldsymbol{\alpha}^M$. For example, the choice $\alpha_j^M = 2$ for all j reflects the belief that the function is infinitely differentiable, plausible for many engineering and

scientific models. Denote all the covariance parameters by $\boldsymbol{\theta}^M = (\lambda^M, \boldsymbol{\alpha}^M, \boldsymbol{\beta}^M)$ which, together with μ^M, are the unknown parameters of the GASP approximation to the computer model.

Finite dimensional distribution: As before, let $\boldsymbol{y}^M$ denote the vector of model evaluations at the set of inputs $D^M \times D^T$. From the Gaussian process assumption, $\boldsymbol{y}^M$, conditional on the hyperparameters, is multivariate normal with covariance matrix $\boldsymbol{\Gamma}^M = \boldsymbol{C}^M(D^M \times D^T, D^M \times D^T)/\lambda^M$, where $\boldsymbol{C}^M(D^M \times D^T, D^M \times D^T)$ is the matrix with (i,j) entry $c^M(\boldsymbol{z}_i, \boldsymbol{z}_j)$, for each pair $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ in $D^M \times D^T$. Once $\boldsymbol{y}^M$ is observed, this yields a likelihood function for the parameters $\boldsymbol{\theta}^M$ and μ^M (based solely on the observed $\boldsymbol{y}^M$).

Prediction (approximation) at a new input: If $\boldsymbol{z}$ is a new input value, then the conditional distribution of $y^M(\boldsymbol{z})$, given $\boldsymbol{y}^M$, μ^M and $\boldsymbol{\theta}^M$ is normal. Formally, the posterior density, $p(y^M(\cdot) \mid \boldsymbol{y}^M, \mu^M, \boldsymbol{\theta}^M)$, is a Gaussian process with mean and covariance function given by

$$E[\, y^M(\boldsymbol{z}) \mid \boldsymbol{y}^M, \mu^M, \boldsymbol{\theta}^M] = \mu^M x_1\, t + \boldsymbol{r_z}'(\boldsymbol{\Gamma}^M)^{-1}(\boldsymbol{y}^M - \mu^M \boldsymbol{\Psi}) \tag{3.2}$$

$$\mathrm{Cov}[\, y^M(\boldsymbol{z}), y^M(\boldsymbol{z}^\star) \mid \boldsymbol{y}^M, \mu^M, \boldsymbol{\theta}^M] = \frac{1}{\lambda^M}\, c^M(\boldsymbol{z}, \boldsymbol{z}^\star) - \boldsymbol{r_z}'(\boldsymbol{\Gamma}^M)^{-1}\boldsymbol{r_{z^\star}}, \tag{3.3}$$

where $\boldsymbol{r}_z{}' = \frac{1}{\lambda^M}\ (c^M(\boldsymbol{z}, \boldsymbol{z}_1), \ldots, c^M(\boldsymbol{z}, \boldsymbol{z}_m))$, $\boldsymbol{\Gamma}^M$ is given above, and $\boldsymbol{\Psi}$ is the column vector consisting of the values of $x_1\, t$ corresponding to the inputs in $D^M \times D^T$.

With specification of $(\mu^M, \boldsymbol{\theta}^M)$, the GASP thus behaves as a Kalman filter, yielding a posterior mean function (3.2) that can be used as the fast approximation or inexpensive emulator for $y^M(\cdot)$. Indeed, the response surface approximation to $y^M(\boldsymbol{z})$, given $(\mu^M, \boldsymbol{\theta}^M)$, is simply $E[\, y^M(\boldsymbol{z}) \mid \boldsymbol{y}^M, \mu^M, \boldsymbol{\theta}^M]$, and the variance measuring the uncertainty in this approximation is given by the right-hand side of (3.3) with $\boldsymbol{z}^\star$ replaced by $\boldsymbol{z}$. Note that this variance is zero at the design points at which the function was actually evaluated. The model approximation obtained through the GASP theory can thus roughly be thought of as an interpolator of the data.

The hyper-parameters $(\mu^M, \boldsymbol{\theta}^M)$ are unknown, and will be dealt with in a Bayesian fashion. This will be detailed in Appendix C, and leads to a posterior sample $(\mu_i^M, \boldsymbol{\theta}_i^M), i = 1, \ldots, N$. The resulting Bayesian approximation to $y^M(\cdot)$ is

$$\hat{y}^M(\boldsymbol{z}) = \frac{1}{N} \sum_{i=1}^{N} \left[\, \mu_i^M x_1\, t + \boldsymbol{r}_i{}'_{\boldsymbol{z}}(\boldsymbol{\Gamma}_i^M)^{-1}(\boldsymbol{y}^M - \mu_i^M \boldsymbol{\Psi})\right] ,$$

where $\boldsymbol{r}_i{}'_{\boldsymbol{z}}$ and $\boldsymbol{\Gamma}_i^M$ are computed using the generated values $(\mu_i^M, \boldsymbol{\theta}_i^M)$. An estimate of the variance of this approximation can be obtained by adding two terms: the posterior expectation of (3.3) and the posterior variance of (3.2). In practice, these terms are estimated, respectively, by the sample average of (3.3) and by the sample variance of (3.2) evaluated at the generated values $(\mu_i^L, \boldsymbol{\theta}_i^M)$.

Note that one can also just generate directly from the posterior distribution of $y^M(\boldsymbol{z}_{\mathrm{new}})$, by first sampling $(\mu_i^M, \boldsymbol{\theta}_i^M)$ from its posterior distribution and then sampling the multivariate normal with mean and covariance given by (3.2) and (3.3) with the sampled hyper-parameters.

Key computational simplification: The major difficulty in the above computations is the inversion of the many thousands of matrices $\mathbf{\Gamma}_i^M$, or equivalently (and dropping the i subscripts) the inversions of the correlation matrices $\boldsymbol{C}^M(D^M \times D^T, D^M \times D^T)$. These matrices are of dimension $m = LT$ which, in CRASH, will be seen to be as large as 323. Additionally, the sampling mechanism utilized to obtain samples from the posterior distributions of the unknowns forces one to compute the inverse (and determinant) of the correlation matrix for values of the parameters that lead to highly ill-conditioned matrices. This leads to very unstable calculations that become increasingly unreliable as the dimension of the matrices increases.

It is here that crucial use is made of the choice of a product form for the correlation function, together with use of the product input space $D^M \times D^T$. It follows directly that

$$\boldsymbol{C}^M(D^M \times D^T, D^M \times D^T) = \boldsymbol{C}_x^M(D^M, D^M) \otimes \boldsymbol{C}^T(D^T, D^T), \tag{3.4}$$

where $\boldsymbol{C}_x^M$ and $\boldsymbol{C}^T$ are correlation matrices corresponding to separate use of the $\boldsymbol{x}$ and t components of the correlation functions, and $\otimes$ refers to the Kronecker product defined as: $\boldsymbol{A} \otimes \boldsymbol{B}$, for matrices $\boldsymbol{A}_{m\times n}$ and $\boldsymbol{B}_{p\times q}$, is the $mp \times nq$ matrix whose i, j block is $a_{ij}\boldsymbol{B}$.

The advantage of the Kronecker product structure (see Bernardo et al. (1992) for a related use) is that then

$$(\boldsymbol{C}^M(D^M \times D^T, D^M \times D^T))^{-1} = (\boldsymbol{C}_x^M(D^M, D^M))^{-1} \otimes (\boldsymbol{C}^T(D^T, D^T))^{-1}, \tag{3.5}$$

and inverting L and T-dimensional matrices (and multiplying the inverses together) is much cheaper, and more stable, than inverting an LT-dimensional matrix. Also,

$$\left|\boldsymbol{C}^M(D^M \times D^T, D^M \times D^T)\right| = \left|\boldsymbol{C}_x^M(D^M, D^M)\right|^T \left|\boldsymbol{C}^T(D^T, D^T)\right|^L . \tag{3.6}$$

4 Analysis of Model Output

4.1 Notation and statistical modeling

Reality and bias: It is crucial to represent the computer model as a biased representation of reality, "reality = model + bias;" formally

$$y^R(\boldsymbol{x}, t) = y^M(\boldsymbol{x}, t) + b(\boldsymbol{x}, t)\,, \tag{4.1}$$

where $y^R(\boldsymbol{x}, t)$ is the value of the 'real' process at input $(\boldsymbol{x}, t)$ and $b(\boldsymbol{x}, t)$ is the unknown bias function.

Modeling the field response functions: The field response functions at input $\boldsymbol{x}^{*i}, i = 1, \ldots, l$,

are modeled as

$$y^F(\boldsymbol{x}^{*i}, t) = y^R(\boldsymbol{x}^{*i}, t) + \epsilon_i^F(t)\,, \tag{4.2}$$

where the $\epsilon_i^F(t)$ are independent $\mathrm{GP}(0, \frac{1}{\lambda^F}\, c^T(\cdot,\cdot))$, i.e., are independent Gaussian processes with mean function 0, precision λ^F, and correlation function

$$c^T(t, t^\star) = \exp\left(-\beta^T |t - t^\star|^{\alpha^T}\right). \tag{4.3}$$

The assumption that $\epsilon^F(\cdot)$ has mean zero is formally the assumption that the field observations have no bias. If the field observations do have bias, the situation is quite problematic, in that there is then no purely data-based way to separate the field bias from the model bias. Estimates of bias that arise from our methodology could still be interpreted as the systematic difference between the computer model and field observations, but this is of little interest, in that prediction of reality (not possibly biased field data) is the primary goal. Assuming that each field response function error is itself a draw from a Gaussian process, and one of the same form (as a function of t) as the computer model, seems quite natural.

Stochastic modeling of the bias: Since we will be performing a Bayesian analysis, the bias must be assigned a prior distribution. It is natural to choose this to be another Gaussian process with constant mean function $\mu^b(\cdot) \equiv \mu^b$ and correlation function as in (3.1), but with its own set of hyper-parameters $\boldsymbol{\alpha}^b, \boldsymbol{\beta}^b, \lambda^b$. However, we restrict attention to smooth bias functions by fixing all components of the vector $\boldsymbol{\alpha}^b$ to be two (except for the t component, to be discussed shortly). In part, this is done for technical reasons; since the bias cannot be observed directly, there is very little information available about $\boldsymbol{\alpha}^b$, and numerical computations are more stable with $\boldsymbol{\alpha}^b$ specified. There is also the notion that the bias process might typically be smoother than the model process; for instance, the model process might only be 'off' by a level-shift, because of something forgotten or inappropriately specified in the model. Indeed, there is both empirical and 'folklore' evidence of this. Empirically, in the examples we have looked at, the maximum likelihood estimates of $\boldsymbol{\alpha}^b$ have mostly been near 2. As to folklore, it is often claimed that even biased models are typically accurate for predicting small changes, which would not be true if bias were not smoother than the model outputs. Finally, note that the bias can still assume the form of any infinitely differentiable function.

The choice of a constant mean function for the Gaussian process of the bias was to allow some flexibility in the level of the bias, but to avoid confounding with the linear structure being assumed for the computer model Gaussian process.

Key assumption: We assume that the Gaussian process correlation parameters corresponding to the input t – namely α^T and β^T – are the same for the computer model, the field error, and the bias. As will be seen in Appendix B.8, this is a necessary assumption for computational implementation, in that it allows all the Bayesian computations to take advantage of the type of Kronecker product

simplification illustrated in (3.5). It is also not unreasonable scientifically: the variations of the three functions with respect to t are not being assumed to be the same, but are merely being assumed to have the same correlation structure.

4.2 Bayesian analysis

To proceed with a Bayesian analysis of the situation, it is only necessary to specify a prior density for the remaining unknown parameters, namely $(\mu^M, \mu^b, \lambda^M, \lambda^b, \lambda^F, \alpha^T, \beta^T, \boldsymbol{\beta}^M, \boldsymbol{\alpha}^M, \boldsymbol{\beta}^b)$, and apply Bayes theorem. The details of the prior assignment are given in Appendix B.9, and the Bayesian implementation via MCMC is discussed in Appendix C. Here we focus on discussion of the possible outputs of the analysis.

The MCMC results in a sample of all unknowns, including the key functions $y^M(\boldsymbol{x}, t)$ and $b(\boldsymbol{x}, t)$ at specific input values $\boldsymbol{x}$ of interest. (Note that equation (4.1) can then be used to produce a sample from $y^R(\boldsymbol{x})$ as well.) Denote the MCMC sample by $\{y_i^M(\boldsymbol{x}, t), b_i(\boldsymbol{x}, t); i = 1, \ldots, N\}$. A technical point is that one must again discretize t and do the function predictions at these discretized values; but this can be done at a much finer set of t values than D^T, since the computations involved in prediction are much faster than those involved in obtaining a posterior sample of unknown parameters. In CRASH, the discretization $D_t^P = (3,\ 6,\ \ldots,\ 81)$ was used at this prediction stage, and was quite adequate for reconstruction of the functions. Details of this aspect of the calculations are in Appendix C.2.

A final point is that, when predicting curves for initial velocities that are not part of the model and field data that was originally collected, we must introduce the added information that the initial relative velocity is zero (else the posterior realizations would, inappropriately, have varying initial relative velocities). Incorporation of such constraints is straightforward, as explained in detail in Appendix C.3.

4.2.1 Bias estimates

The estimated bias function is obtained from the posterior sample as

$$\hat{b}(\boldsymbol{x}, t) = \frac{1}{N} \sum_{i=1}^{N} b_i(\boldsymbol{x}, t).$$

This estimated bias function for SDM velocity at the input $x_1 = 56.3$ km/h (impact velocity) is the central curve in Figure 2.

It is also important to give tolerance or confidence bands for any estimated functions. Pointwise 80% posterior intervals for the bias are also given in Figure 2. These were found by simply taking the 10^{th} and 90^{th} percentiles of the N function values at each (discretized) t. Note that these confidence bands are wide enough that there is no clear indication of bias at the given input value.

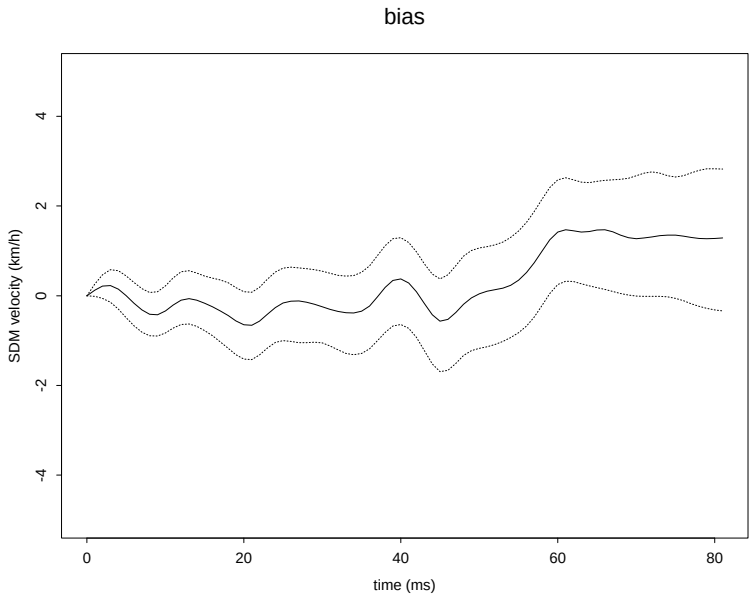

Figure 2: Estimate and 80% posterior intervals for SDM velocity bias, at 56.3km/h impact.

(This, of course, is not a proof that no bias exists – it is simply the observation that the data are not conclusive in this regard.)

In contrast, Figure 3 shows the posterior estimate of the bias for a 30km/h impact, along with the 80% confidence bands. The bias is clearly larger in the 20-59ms interval than it was for the 56.3km/h impact, and now the confidence bands do not cover 0. Hence this is clear evidence that the computer model is biased at this input value (for t in the 20-59ms interval). This was also noticed in other contexts, leading to the conclusion that the computer model appears to be more severely biased at low impact velocities than at higher impact velocities, a finding of considerable interest to the modelers.

4.3 Predicting the computer model and reality

Estimates of $y^M(\boldsymbol{x}, t)$ and $y^R(\boldsymbol{x}, t)$ at a (new) input $\boldsymbol{x}$ are given by the MCMC estimate of the posterior means of the functions, namely

$$\hat{y}^M(\boldsymbol{x}, t) = \frac{1}{N} \sum_{i=1}^{N} y_i^M(\boldsymbol{x}, t) \quad \text{and} \quad \hat{y}^R(\boldsymbol{x}, t) = \frac{1}{N} \sum_{i=1}^{N} [y_i^M(\boldsymbol{x}, t) + b_i(\boldsymbol{x}, t)] \,. \tag{4.4}$$

This is assuming that the computer model is not run at the new input $\boldsymbol{x}$; the situation when it is subsequently run at $\boldsymbol{x}$ is considered in Appendix C.3. The estimate $\hat{y}^R(\boldsymbol{x}, t)$ is called the *bias-corrected prediction of reality* in Bayarri et al. (2005b) because it equals $\hat{y}^M(\boldsymbol{x}, t) + \hat{b}(\boldsymbol{x}, t)$.

Posterior confidence bands for both estimates can also be computed, using the samples generated

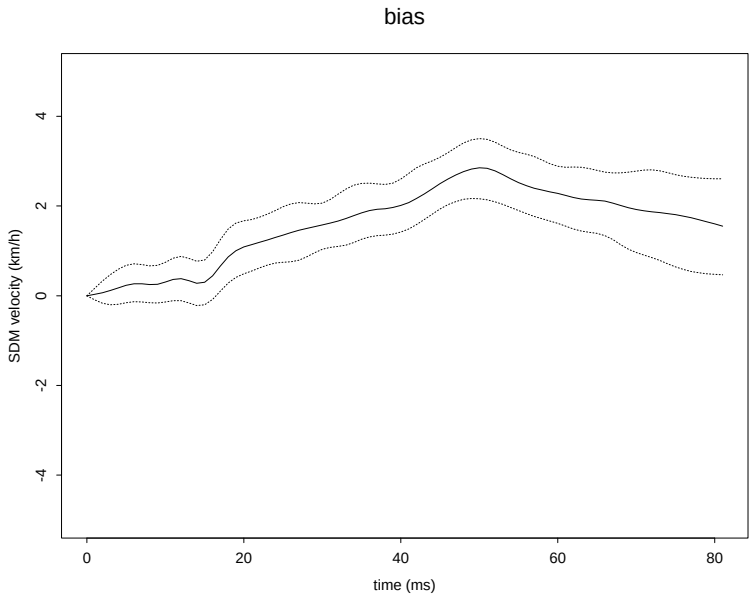

Figure 3: Estimate and 80% posterior intervals for SDM velocity bias, at 30km/h impact.

from the MCMC. The estimates and confidence bands are given in Figure 4 for SDM velocity when the impact is 56.3km/h.

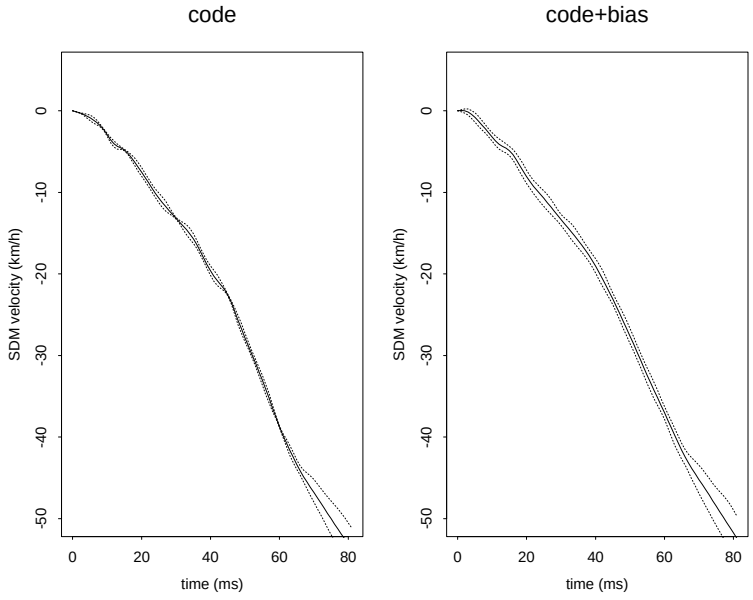

Figure 4: Estimate and 80% posterior intervals of SDM velocity for y^M (left figure) and for y^R (right figure), when the impact velocity is 56.3km/h.

The tight confidence bands around $\hat{y}^M$ indicate that the GASP approximation to the computer model code is quite accurate at the 56.3km/h input, except for times exceeding 65ms. The bias-corrected prediction of reality also appears to be quite accurate over the same region.

The confidence bands for the computer model prediction are not indicators of the accuracy of the computer model in predicting reality; they simply indicate the accuracy of the GASP approximation to the computer model. Accuracy of the computer model, while observable by plotting the bias, is best quantified by producing *tolerance bands*, as in Bayarri et al. (2005b). Tolerance bands are constructed so that one can, e.g., make the statement "with probability 0.80, the prediction $\hat{y}^M(\boldsymbol{x}, t)$ is within a specified tolerance τ of the true $y^R(\boldsymbol{x}, t)$." (Note that this is a pointwise statement.) Symmetric 80% tolerance bands are simply found by choosing τ so that 80% of the posterior samples satisfy

$$|\hat{y}^M(\boldsymbol{x}, t) - [y_i^M(\boldsymbol{x}, t) + b_i(\boldsymbol{x}, t)]| < \tau\,. \tag{4.5}$$

Examples are given in Table 4.

4.4 Prediction of CRITV

Recall that the primary evaluation function for CRASH was CRITV, the SDM velocity calculated 30ms before the time the SDM displacement (DISP) reaches 125mm. Note that DISP is just integrated velocity, i.e. $\text{DISP}(t) = -\int_0^t y^R(\boldsymbol{x}, v)dv$. Thus $\text{CRITV} = y^R(\boldsymbol{x}, \text{DISP}^{-1}(125) - 30)$.

Obtaining a posterior sample for CRITV is relatively straightforward. For bias-corrected prediction of reality, one takes each sample function $y_i^R(\boldsymbol{x}, t)$ from the posterior and simply solves for the corresponding CRITV_i using the above formulas. The result is a sample $\{\text{CRITV}_i, i = 1, \ldots, N\}$ from the posterior distribution of CRITV. (Note that CRITV is a very involved function of the other parameters, yet the MCMC technique produces its posterior very easily.) Likewise, one could use the posterior sample $y_i^M(\boldsymbol{x}, t)$ of predictions of the computer model to estimate CRITV^M, by which we mean the value of CRITV that would result from actually exercising the computer model at $\boldsymbol{x}$ and computing CRITV from the resulting $y^M(\boldsymbol{x}, t)$.

We first consider the impact velocity $x_1 = 56.3$km/h. Figure 5 gives the posterior distribution of CRITV (lower figure) and CRITV^M (upper figure).

The fact that the posterior for CRITV^M is quite tight shows that using the GASP approximation to the computer model is effective. Of course, the posterior for CRITV^M does not tell us directly about the quantity of interest, CRITV, and the lower figure shows that there is considerably more uncertainty as to its actual value. Indeed, 80% tolerance bands for the computer model prediction of CRITV itself are -5.11 ± 0.43, while the 80% tolerance bands for CRITV^M are the much tighter -5.11 ± 0.17.

For bias-corrected prediction, the 80% tolerance bands are -5.21 ± 0.42, very close to the tolerance bands obtained when using the computer model. This suggests that the computer model can be viewed as providing an accurate estimate of CRITV at initial impact velocity 56.3km/h.

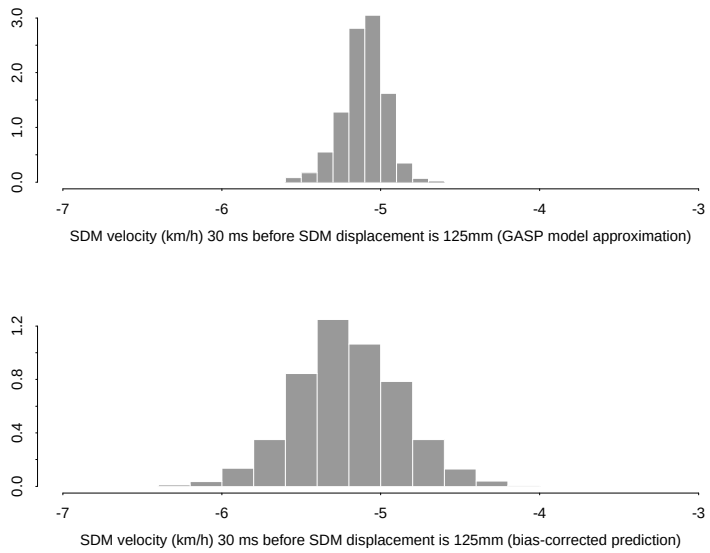

Figure 5: Posterior distribution of CRITV (lower figure) and CRITV^M (upper figure), at a 56.3km/h impact velocity.

This is not very surprising, of course, given the fact that Figure 2 showed little evidence of bias in the prediction of $y^R(\boldsymbol{x}, t)$ by $y^M(\boldsymbol{x}, t)$.

On the other hand, Figure 3 did indicate the presence of bias in the computer model at impact velocity 30km/h. However, 80% tolerance bands for CRITV from the computer model predictions and bias-corrected predictions, respectively, are -6.53 ± 0.64 and -6.56 ± 0.63, which are remarkably similar. This serves as a potent reminder that validity of a computer model can depend strongly on the evaluation criterion of interest.

5 Extrapolation Past the Range of the Data

One of main motivations for using computer models is the hope that they can adequately predict reality in regions outside the range of the available field data. We have advocated use of bias-correction – incorporating the field data – to improve (typically biased) computer model predictions. The difficulty is that the estimates of bias may not extrapolate well outside the range of the actual field data. When this is the case, the Bayesian methodology will tend to return very large tolerance bands; while one is at least not then making misleading claims of accuracy, the large bands may make assertion of predictive accuracy of the model impossible. (On a technical note, the best way to minimize the size of the tolerance bands in extrapolation is to choose the mean, $\mu^M(\cdot)$, of the model Gaussian process and the mean of the bias Gaussian process to be as accurate representations of the real process as possible.)

One 'solution' to this difficulty is to simply make the scientific judgment that the bias estimates do extrapolate. For instance, in CRASH, the entire analysis was performed for a fixed vehicle configuration. If, say, an element of the vehicle frame were increased in thickness by 5%, one might reasonably judge that the bias estimates would extend to this domain, even though no field data was obtained for varying thicknesses of the element. Of course, any such assumption should be reported, along with the conclusions of predictive accuracy of the model.

5.1 Hierarchical modeling

Bayesian methodology allows a weaker (and typically much more palatable) way of extrapolating past the range of the data. The idea is to use hierarchical modeling to relate the new situation to those for which data is available, but not to insist that the situations are identical in terms of bias and predictive accuracy.

Hierarchical modeling applies most directly to scenarios in which there are K different function outputs, to be denoted by subscripts i, each coming from different configurations of a computer model (or even from different computer models). Each of these functions can be modeled as was done in Sections 3 and 4, through Gaussian process priors. We will be particularly concerned with settings where the Gaussian processes for y_i^M and b_i can be assumed to share common features, typically where the parameters governing the priors are drawn from a common distribution. This induces connections among the individual models and enables us to combine information from the separate models, sharpen analyses and reduce uncertainties.

Implementation of these ideas will depend heavily on what data, both computer and field, are available as well as the legitimacy of the assumptions imposed. Here we informally state and comment on these assumptions for the simplest structure we will impose. (Full details can be found in Appendix D.)

Assumption 1. The smoothness of the model approximation processes are identical across the K models being considered. This is a reasonable assumption in the contexts for which hierarchical modeling would typically be employed. Hence, we assume that all models and field data have common α's and common β's, which are assigned priors as in the single model case.

Assumption 2. The variances $1/\lambda^M$ of the model approximation processes are equal, across the various cases, as are the variances $1/\lambda^F$ of the field data. Again, this is typically reasonable, and the priors used for these parameters are as in the single model case.

Assumption 3. The unknown mean parameters, μ_i^M, from the GASP approximations for the K computer models, are assumed to arise from the following two stage-hierarchical model:

- $\mu_i^M \mid \mu, \tau \sim \mathrm{N}(\mu, \tau^{-1})$
- (μ, τ) is assigned a reference prior, detailed in Appendix D.

This prior distribution will allow the computer models to vary considerably between the K situations, but still ensures that information is appropriately pooled in their estimation.

Assumption 4. The mean biases, μ^b, for the K situations are assumed to be equal (their common value being assigned a constant prior as in the single model case), while the variances of the bias processes are related in a fashion described by a parameter q, whose value must be specified. This parameter describes the believed degree of similarity in the biases for the K different computer models; indeed, $1 + q$ can be interpreted as an upper bound on the believed ratio of the standard deviations of the biases, or, stated another way, the proportional variation in the bias is q. Specifying $q = 0.1$ is stating that the biases are expected to vary by about 10% among the various cases being considered.

Specifically, we assume that $\log(\lambda_i^b) \sim \mathrm{N}(\eta, 4q^2)$, where q is the proportional variation in the bias that is expected among models (e.g., $q = 0.1$). A constant prior density is assigned to η. q must be specified.

Note that specification of q is a judgment as to the *comparative* accuracy of the K different computer models, as opposed to their *absolute* accuracy (which need not be specified). The reason we require specification of this parameter by the engineer/scientist is that there is typically very little information about this parameter in the data (unless K is large). Specifying q to be zero could be reasonable, if one is unsure as to the accuracy of the computer models but is quite sure that the accuracies are the same across the various K. As a final comment, note that the bias process means, μ^b, are less important in the analysis than the bias process variances, λ^b, and having the latter vary between models was felt to be allowing sufficient variation in the bias processes, given the limited amount of available data; hence the process means were set equal to a common (unknown) value.

The analysis reported in Section 4 was for the data and model for rigid barrier, straight frontal impact. By use of hierarchical modeling we can simultaneously treat rigid barrier, left angle and right angle impacts as well as center pole impact. Thus we use the hierarchical model with $K = 4$ related situations. The analyses and predictions reported below are for a 56.3km/h impact (this is at the high end of the data). The hierarchical model was used with $q = 0.1$. For simplicity, however, the prior distributions used for the GASP parameters were chosen to be the same as those used for the straight frontal analysis (and described in Appendix B.9); this is reasonable, since the priors are relatively non-informative and the straight frontal dataset is by far the largest of the 4 categories.

Figure 6 shows the posterior distributions of $\log \lambda_i^b$ and μ_i^M for individual barrier types. Note that, while the assumed similarity between the models allows information to be passed from 'large data' to 'small data' models, the models are still allowed to vary significantly.

Figure 7 shows the differing posterior predicted SDM velocity curves and pointwise uncertainty bands for each of the four barrier types. The straight frontal and left angle posterior intervals in Figure 7 are tight, in part because there are data with inputs close to 56.3km/h for these barrier types. In contrast, the intervals are not tight for the other barriers because data near 56.3km/h are

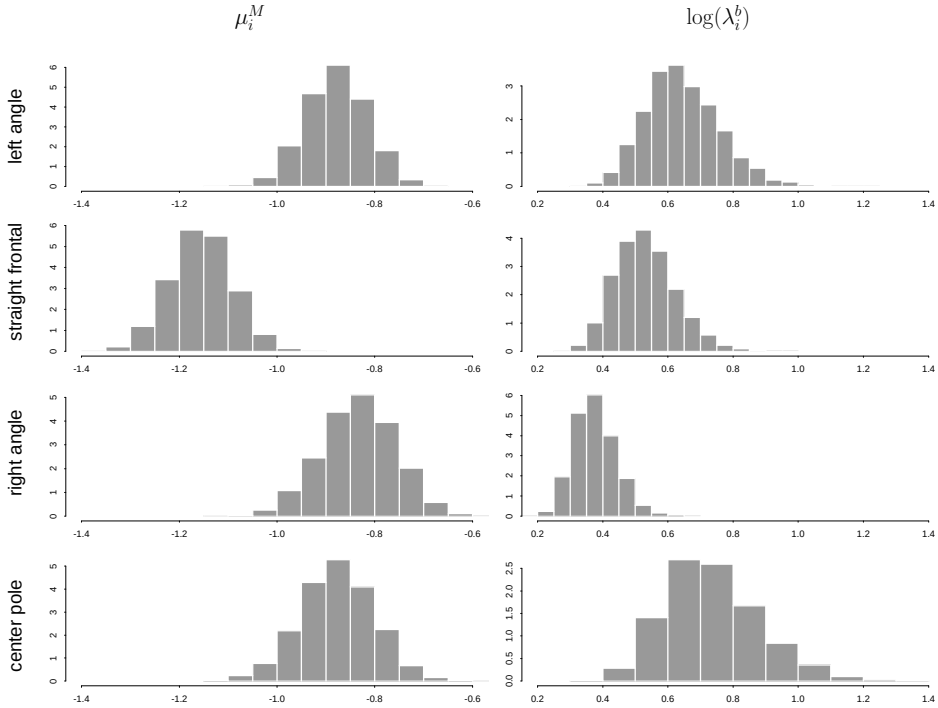

Figure 6: Posterior distributions of μ_i^M and $\log \lambda_i^b$ for the 4 barrier types, based on the hierarchical model.

lacking. (This thus reinforces the potential value of making a model run at a new desired input.) Note that Figure 4 and the straight frontal pictures in Figure 7 are similar, so that the hierarchical analysis did not greatly affect the answers for this barrier type (which had the most data).

Figure 8 gives the estimates of the four bias functions and the associated pointwise uncertainties. Because of the large uncertainties in the bias estimates, the only case in which the bias seems clearly different from zero is for left angle impacts, after 43ms. (While we cannot clearly assert that there is bias in the other cases, the tolerance bounds for predictions will be quite large, reflecting the uncertainty in the bias estimates.)

For CRITV, Table 2 presents the mean and standard deviations of CRITV for each barrier type for the GASP model approximation and for the bias-adjusted prediction of CRITV.

5.2 Angle as an additional input

If we only consider the three rigid barrier impacts (frontal, right angle and left angle) and ignore the center pole impact data, we could proceed without use of hierarchical modeling by incorporating the angle of impact, x_2, as an input to the model. The smoothness assumption required for the Gaussian process analysis is plausible: it is reasonable to assume that small changes in the angle

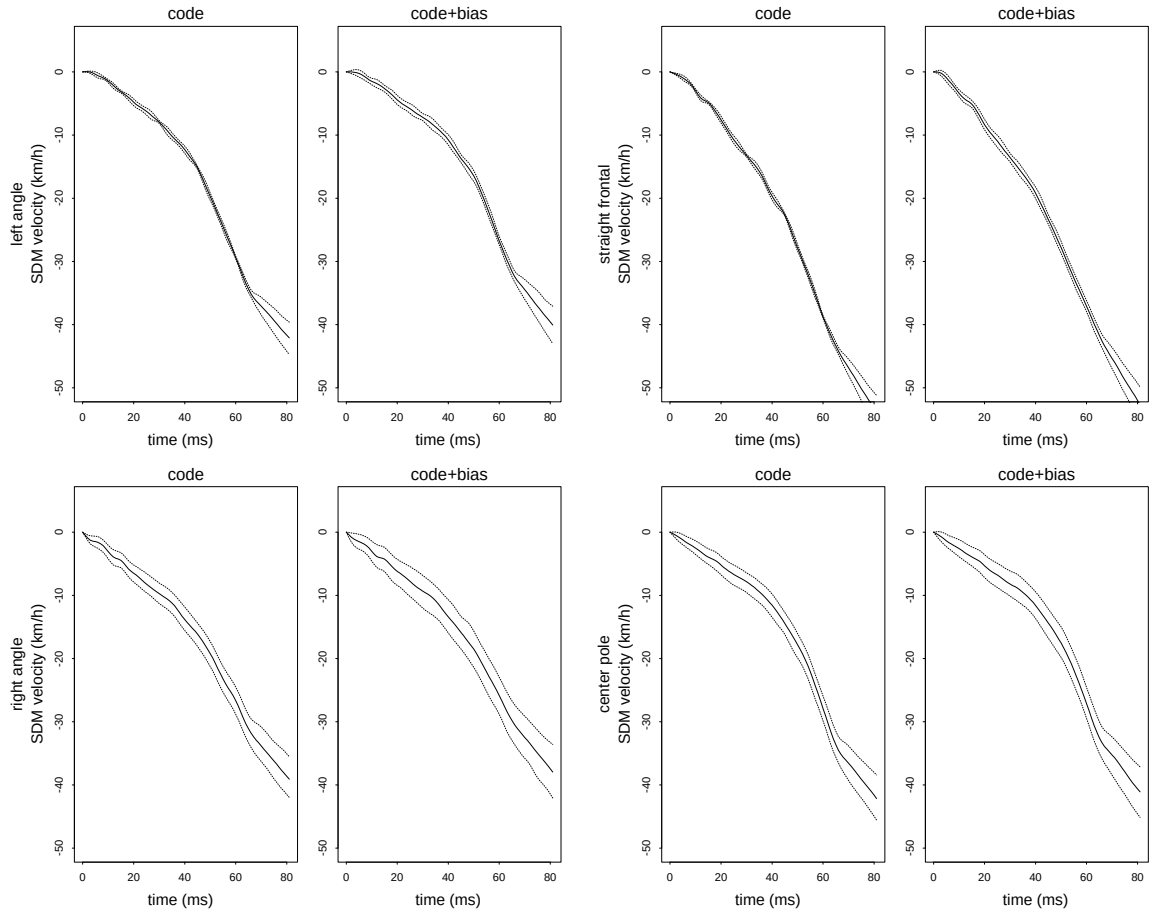

Figure 7: Pointwise 80% posterior intervals for SDM velocity under 4 barrier types, based on the hierarchical model, and arising from the GASP model approximation estimate and the bias-corrected prediction, respectively, at the input velocity 56.3km/h.

	Hierarchical model		Using frontal data only	
Barrier type	CRITV^M	CRITV	CRITV^M	CRITV
left angle	-6.08 (0.34)	-6.34 (0.49)		
straight frontal	-5.13 (0.13)	-5.22 (0.30)	-5.11 (0.13)	-5.21 (0.33)
right angle	-6.89 (0.65)	-6.80 (0.96)		
center pole	-6.55 (0.74)	-6.54 (0.91)		

Table 2: Posterior mean and standard deviation of CRITV^M and CRITV, at the input velocity 56.3km/h.

will result in small changes in the velocity-time curve so that y^M is a smooth function of x_2.

Combining the data from left angle ($x_2 = 0.0$), right angle ($x_2 = 1.0$), and straight frontal ($x_2 = 0.5$) barrier impacts led to computations that were considerably more expensive than in the

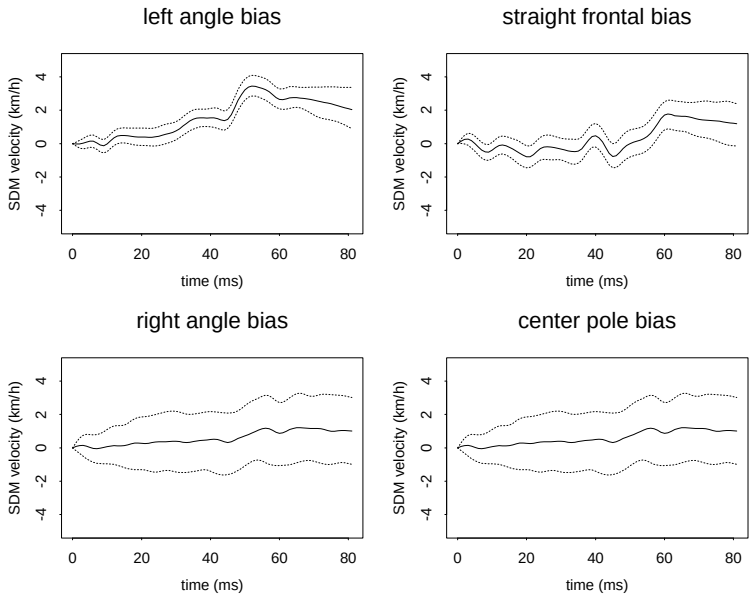

Figure 8: Pointwise 80% posterior intervals for bias, under 4 barrier types, based on the hierarchical model and at the input velocity 56.3km/h.

hierarchical model because we now need to invert 26×26 matrices instead of the smaller matrices encountered in dealing with the individual barrier types.

Comparison of the results in Tables 3 and 2 generally shows close agreement between using angle as input and the hierarchical model. There are differences associated with the right angle CRITV, reflecting the paucity of data for that angle. (The hierarchical model makes weaker assumptions about the relationship between the various cases than does incorporation of angle as an input variable, and hence is more affected by the shortage of data for right angle.)

Finally, Table 4 gives *tolerance bounds* for CRITV – computed as in (4.5) – assuming that the computer model (more strictly, the approximation to the computer model) is used without bias-correction. These are somewhat larger than the corresponding 80% tolerance bands for bias-corrected prediction (which can roughly be found by multiplying the standard deviations in the second column of Table 2 by 1.28). If there were more bias, the tolerance bands when using the computer model alone to predict CRITV would have been much larger (see Bayarri et al. (2005b)).

Barrier type	CRITV^M	CRITV
left angle	-6.04 (0.36)	-6.29 (0.53)
straight frontal	-5.12 (0.15)	-5.24 (0.30)
right angle	-6.88 (0.64)	-6.69 (0.90)

Table 3: Posterior mean and standard deviation of CRITV^M and CRITV, based on the non-hierarchical model with *angle* as an additional input, and at the input velocity 56.3km/h.

Barrier type	Hierarchical model	Angle input x_2	Frontal data only
left angle	-6.08 $\pm$ 0.70	-6.04 $\pm$ 0.76	
straight frontal	-5.13 $\pm$ 0.40	-5.12 $\pm$ 0.41	-5.11 $\pm$ 0.43
right angle	-6.89 $\pm$ 1.27	-6.88 $\pm$ 1.17	
center pole	-6.55 $\pm$ 1.18		

Table 4: Posterior mean and 80% tolerance bounds for CRITV, arising from the GASP model approximation estimate of y^M, at the input velocity 56.3km/h.

6 Merging Predictive and Physical Approaches to Validation

6.1 Introduction

We have focused on the predictive approach to model validation, that of determining the accuracy of the predictions of the computer model, assuming that some bias exists. In what could be called the physical school of modeling, a modeler that has carefully constructed and exhaustively tested each component of a model (including component interfaces) might argue that the model is correct by construction, i.e., that there can be no bias. Such claims are rarely believed without at least some confirming data, but how much confirming data is needed?

This same question arises in the pure view of the scientific process. A scientist proposes a new theory, which makes precise predictions of a process, say $y^M(\boldsymbol{x}) \pm 0.0001$ at input $\boldsymbol{x}$ (dropping the t index for simplicity). Other scientists then try to devise an experiment that can test this theory, i.e., an experiment that, for some input $\boldsymbol{x}^\star$, will provide a field observation $y^F(\boldsymbol{x}^\star)$ that is within, say, 0.00001 of the true process value $y^R(\boldsymbol{x}^\star)$. If the experiment is conducted and $y^F(\boldsymbol{x}^\star)$ is within ± 0.0001 of $y^M(\boldsymbol{x}^\star)$, then the scientific theory is viewed as being validated. The key to this scientific process is that, if the scientist makes even one very precise prediction, and the prediction turns out to be true, then that would seem to be considerable evidence in favor of the hypothesis. If the prediction of the scientist were not very precise, then a single observation could disprove, but not really confirm the theory.

The natural language in which to discuss and implement these ideas is the Bayesian language. The proposed new theory (or proposed computer model) is $\mathcal{M}_0$, and one asks the question (after seeing one or more field observations) "What is the probability, given the data, that $\mathcal{M}_0$ is correct?"

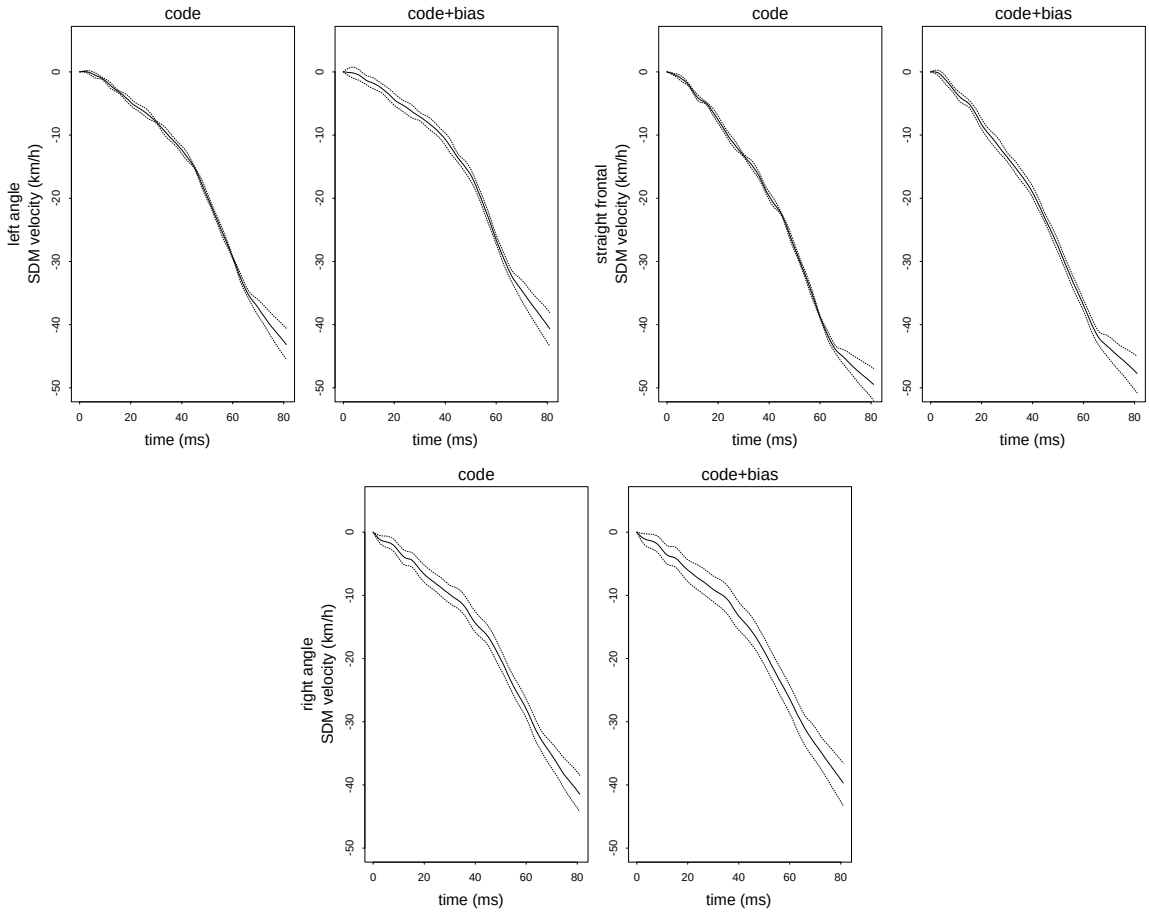

Figure 9: 80% pointwise posterior intervals for SDM velocity, using angle as additional input, and arising from the GASP model approximation estimate and the bias-corrected prediction, respectively, at the input velocity 56.3km/h.

This question can be asked – and answered – through the Bayesian approach, and the result behaves as the scientific intuition from the previous paragraph would suggest. In particular, this probability can be quite high in the scientific context of a precise theory, after even one confirming precise field observation, while it will not be high in the case of an imprecise theory (or an imprecise field observation).

This is a problem of hypothesis testing or model selection. In the classical approach to hypothesis testing, one can essentially show that $\mathcal{M}_0$ is false, if the data so suggest, but it is much harder to show that $\mathcal{M}_0$ is true. (There is widespread abuse here; far too often a classical test of $\mathcal{M}_0$ is performed and, if it does not reject, it is concluded that $\mathcal{M}_0$ is validated. This is simply bad logic.) The Bayesian approach does allow direct answer of the primary question of interest. Several ingredients are needed to implement the Bayesian approach:

1. A prior probability, π_0, that $\mathcal{M}_0$ is true. This can, of course, vary from one individual to another. The modeler might feel π_0 to be quite high, while a skeptic might judge it to be low. Often the default choice $\pi_0 = 1/2$ is made, in order to 'see what the data has to say.'

2. An alternative model $\mathcal{M}_1$.

3. Suitable prior probability distributions on unknown parameters of $\mathcal{M}_0$ and $\mathcal{M}_1$.

In the context of evaluation of computer models, we have already constructed these needed ingredients. In particular,

- The prior distribution on the parameters of the computer model, $\mathcal{M}_0$, is that provided by the I/U map (not needed for CRASH) and the assignment of priors for the GASP parameters.

- The alternative model, $\mathcal{M}_1$, is the model constructed in Sections 3 and 4, which includes the bias term $b(\boldsymbol{x})$.

- The prior distribution on the parameters of the alternative model (including the unknown bias) are as constructed for the predictive validation, and detailed in Appendix B.9.

The result of the analysis is the *posterior probability that* $\mathcal{M}_0$ *is true*, to be denoted $P(\mathcal{M}_0 \mid \boldsymbol{y})$, where here we generically let $\boldsymbol{y}$ refer to all the data.

Technical point: Use of improper priors is typically not possible when interest lies in computation of the posterior probability of models. This is a potential concern because the Bayesian predictive analysis discussed previously assigned constant improper priors to both $\boldsymbol{\mu}^M$ and μ^b. However, the parameters $\boldsymbol{\mu}^M$ are 'location-parameters' that occur in both models and so can be assigned the constant prior, as justified in Berger et al. (2001). Unfortunately, μ^b only occurs in $\mathcal{M}_1$, and so it cannot be assigned a constant prior. Hence, for the computation in this section, we simply assumed that $\mu^b = 0$.

For CRASH, implementing this analysis resulted in a posterior probability of near 0 that the computer model is true (assuming an initial prior probability of $\pi_0 = 1/2$). This was actually also apparent from earlier graphs of the estimated bias, and illustrates an important point: if a computer model has statistically significant bias over any part of the domain under study, the model will have essentially zero posterior probability of being correct. This, of course, is as it should be, but does point out the reason that looking at predictive accuracy of the model (which can vary over the input domain) is greatly superior to simply asking yes/no questions.

Before discussing the details of the analysis, another feature of the Bayesian approach deserves highlighting, namely that the conclusions regarding accuracy of predictions will now be a weighted average of the accuracy statements arising from $\mathcal{M}_0$ and $\mathcal{M}_1$. For instance, if it is desired to know the probability that $y^R(\boldsymbol{x}^\star)$, at specified input $\boldsymbol{x}^\star$, lies within the interval $(8, 10)$, the answer would be

$$P(\mathcal{M}_0 \mid \boldsymbol{y})\, P(8 < y^R(\boldsymbol{x}^\star) < 10 \mid \mathcal{M}_0) + (1 - P(\mathcal{M}_0 \mid \boldsymbol{y}))\, P(8 < y^R(\boldsymbol{x}^\star) < 10 \mid \mathcal{M}_1)\,,$$

with the $P(8 < y^R(\boldsymbol{x}^\star) < 10 \mid \mathcal{M}_i)$ being computable from our previous analyses. In this expression, we see a complete merging of the physical and predictive approaches to model validation. The physical approach would produce the accuracy statement $P(8 < y^R(\boldsymbol{x}^\star) < 10 \mid \mathcal{M}_0)$, while the predictive approach would produce the accuracy statement $P(8 < y^R(\boldsymbol{x}^\star) < 10 \mid \mathcal{M}_1)$. The overall correct answer is their weighted average, with the weights being the posterior probability that each of the models is true.

6.2 Formal testing of model validity

In carrying out the Bayesian computation of $P(\mathcal{M}_0 \mid \boldsymbol{y})$ for a slow computer model, the approximation introduced in Section 3 will be required. In this case, $\mathcal{M}_0$ is like the overall model $\mathcal{M}_1$, but with the bias function $b(\cdot) = 0$.

Let $\boldsymbol{\phi}_i$ be the entire parameter vector for model $\mathcal{M}_i$, $i = 0, 1$ (including all parameters of the mean functions and the Gaussian processes involved). In addition, for model $\mathcal{M}_i$, $i = 0, 1$, denote by $f_i(\boldsymbol{y} \mid \boldsymbol{\phi}_i)$, $p_i(\boldsymbol{\phi}_i)$ and $p_i(\boldsymbol{\phi}_i \mid \boldsymbol{y})$ the likelihood function of the full data vector $\boldsymbol{y}$ (both computer model and field data), the prior density and the posterior density for the parameter vector, respectively. The form of the likelihood function and the approaches for prior specification and posterior inference, using MCMC methods, for model $\mathcal{M}_0$ are similar to the corresponding ones for model $\mathcal{M}_1$, described earlier and detailed in appendices B.9 and C.

Letting $\pi_1 = 1 - \pi_0$ denote the prior probability of $\mathcal{M}_1$, Bayes theorem gives that the posterior probability of $\mathcal{M}_0$ is given by

$$P(\mathcal{M}_0 \mid \boldsymbol{y}) = \frac{\pi_0 m_0(\boldsymbol{y})}{\pi_0 m_0(\boldsymbol{y}) + \pi_1 m_1(\boldsymbol{y})}\,, \tag{6.1}$$

where

$$m_i(\boldsymbol{y}) = \int f_i(\boldsymbol{y} \mid \boldsymbol{\phi}_i) p_i(\boldsymbol{\phi}_i) d\boldsymbol{\phi}_i \tag{6.2}$$

is the marginal likelihood for model $\mathcal{M}_i$, $i = 0, 1$.

Although we can analytically integrate in (6.2) over part of the parameter vector $\boldsymbol{\phi}_i$, part of the integration must be done numerically. There are a variety of possible methods for such numerical integration (see, e.g., Han and Carlin (2001)). We considered two approaches: that of Chib and Jeliazkov (2001), which uses samples from the posterior distribution $p_i(\phi_i \mid \boldsymbol{y})$ along with extra samples drawn from the proposal distribution, and importance sampling, utilizing a t distribution (with mean and variance estimated from the MCMC output) as the importance function. (See Appendix C.4 for details.)

For $\pi_0 = 1/2$, the two computational estimates lead to posterior probabilities of the model $\mathcal{M}_0$ of 2.694×10^{-26} and 1.334×10^{-23}, respectively. The computation via importance sampling was stable and accurate (easy to assess because the importance samples are i.i.d.), but the computation using the method of Chib and Jeliazkov did not seem to fully stabilize; while the estimate by this

latter method is known to converge to the correct answer, the convergence process appears to have very heavy tails.

References

Bayarri, M., Berger, J., Garcia-Donato, G., Palomo, J., Sacks, J., Walsh, D., Cafeo, J., and Parthasarathy, R. (2005a), "Computer Model Validation with Function Output," Tech. rep., National Institute of Statistical Sciences.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C. H., and Tu, J. (2005b), "A Framework for Validation of Computer Models," Tech. rep., National Institute of Statistical Sciences.

Berger, J. O., De Oliveira, V., and Sansó, B. (2001), "Objective Bayesian analysis of spatially correlated data," *Journal of the American Statistical Association*, 96, 1361–1374.

Bernardo, C., Buck, R., Liu, L., Nazaret, W., Sacks, J., and Welch, W. (1992), "Integrated circuit design optimization using a sequential strategy," *IEEE Transactions on Computer-Aided Design*, II., No. 3.

Cafeo, J. and Cavendish, J. (2001), "A Framework For Verification And Validation Of Computer Models and Simulations," Internal general motors document, to be published, GM Research & Development Center.

Chib, S. and Jeliazkov, I. (2001), "Marginal likelihood from the Metropolis-Hastings output," *Journal of the American Statistical Association*, 96, 270–281.

Craig, P. S., Goldstein, M., Rougier, J. C., and Seheult, A. H. (2001), "Bayesian forecasting for complex systems using computer simulators," *Journal of the American Statistical Association*, 96, 717–729.

Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A. (1997), "Pressure matching for hydrocarbon reservoirs: a case study in the use of Bayes linear strategies for large computer experiments," in *Case Studies in Bayesian Statistics: Volume III. Gatsonis, C., Hodges, J. S., Kass, R. E., McCulloch, R., Rossi, P. and Singpurwalla, N. D. (eds.)*, pp. 36–;93.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," *Journal of the American Statistical Association*, 86, 953–963.

Easterling, R. G. (2001), "Measuring the Predictive Capability of Computational Models: Principles and Methods, Issues and Illustrations," Tech. Rep. SAND2001-0243, Sandia National Laboratories.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995), *Bayesian data analysis*, Chapman & Hall Ltd.

Goldstein, M. and Rougier, J. C. (2003), "Calibrated Bayesian forecasting using large computer simulators," Tech. rep., Statistics and Probability Group, University of Durham, http://www.maths.dur.ac.uk/stats/physpred/papers/CalibratedBayesian.ps.

— (2004), "Probabilistic formulations for transferring inferences from mathematical models to physical systems," Tech. rep., Statistics and Probability Group, University of Durham, http://www.maths.dur.ac.uk/stats/physpred/papers/directSim.pdf.

Han, C. and Carlin, B. (2001), "Markov chain Monte Carlo methods for computing Bayes factors: a comparative review," *Journal of the American Statistical Association*, 96, 1122–1132.

Kennedy, M. C. and O'Hagan, A. (2001), "Bayesian calibration of computer models (with discussion)," *Journal of the Royal Statistical Society B*, 63, 425–464.

Kennedy, M. C., O'Hagan, A., and Higgins, N. (2002), "Bayesian analysis of computer code outputs," in *Quantitative Methods for Current Environmental Issues. C. W. Anderson, V. Barnett, P. C. Chatwin, and A. H. El-Shaarawi (eds.)*, Springer-Verlag: London, pp. 227–243.

Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993), "Bayesian design and analysis of computer experiments: Use of derivatives in surface prediction," *Technometrics*, 35, 243–255.

Oberkampf, W. and Trucano, T. (2000), "Validation Methodology in Computational Fluid Dynamics," Tech. Rep. 2000-2549, American Institute of Aeronautics and Astronautics.

Paulo, R. (2005), "Default priors for Gaussian processes," *Annals of Statistics*, 33, 556–582.

Pilch, M., Trucano, T., Moya, J. L., Froehlich, G. Hodges, A., and Peercy, D. (2001), "Guidelines for Sandia ASCI Verification and Validation Plans - Content and Format: Version 2.0," Tech. Rep. SAND 2001-3101, Sandia National Laboratories.

Roache, P. (1998), *Verification and Validation in Computational Science and Engineering*, Albuquerque: Hermosa Publishers.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and analysis of computer experiments (C/R: p423-435)," *Statistical Science*, 4, 409–423.

Santner, T., Williams, B., and Notz, W. (2003), *The Design and Analysis of Computer Experiments*, Springer-Verlag.

Trucano, T., Pilch, M., and Oberkampf, W. O. (2002), "General Concepts for Experimental Validation of ASCII Code Applications," Tech. Rep. SAND 2002-0341, Sandia National Laboratories.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D. (1992), "Screening, predicting, and computer experiments," *Technometrics*, 34, 15–25.

Yang, R. and Chen, M. (1995), "Bayesian analysis for random coefficient regression models using noninformative priors," *Journal of Multivariate Analysis*, 55, 283–311.

A Modeling for vehicle crashworthiness

The finite element model includes the following components: complete body in white including windshield, cradle, bumper system, doors, engine/transmission, suspension, exhaust system, rear axle and drive shaft, radiator, steering system, instrument panel beam, brake booster, and others. Added mass is often used to represent nonstructural components and non-essential objects, while maintaining the actual vehicle test weight and its center of gravity. Generally, element size is approximately between 10 mm to 15 mm. Holes of smaller than 15 mm diameter are not modeled unless in critical areas and fillets, rounds, and radii less than 10 mm is ignored. Mesh quality is tied closely to the correlation of the analysis such that warpage, skew angle, aspect ratio, element interior angle, and taper should be less than the limited values to ensure quality models. These suggested values are available in the procedure. Typically, the number of elements ranges in size from fifty thousand elements in the early 90s to three hundred thousand elements in current multi-purpose models. Duration of the simulations is limited to 100 msec to 150 msec for most of the frontal impact conditions if post impact dynamics is not required. Typical vehicle models are shown in Figure 10.

The finite element vehicle model consists of, mostly, shell and solid elements. Shell elements are used to model numerous sheet metals, such as rail, frame, and stamped/deep-drawn sheet panels, while solid elements are used to model bumper foam, radiator, battery, or even the suspension system. An engine is usually modeled as shell elements on the exterior surface with properly assigned mass and moments of inertia, to represent the massive engine block. Since analysis of vehicle crashworthiness is primarily concerned with the crash behavior of the vehicle, it is imperative that considerable detail of the crush zone structure/components and the connections between them be accurately represented in the model.

Guidelines for crashworthiness are set to make sure that accurate results can be calculated from a commercial solver, LS-DYNA, within a viable computational turn around time, usually within a day or two with the help from powerful Symmetric Multi-Processor (SMP) and Massively Parallel

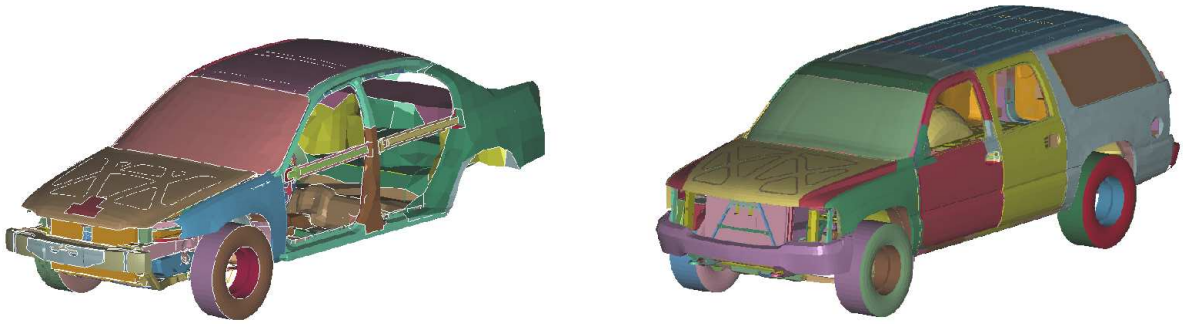

Figure 10: Typical finite element vehicle models: (a) A car finite element model; (b) A sport utility vehicle model

Processor (MPP) technologies. LS-DYNA is a non-linear dynamic analysis code and its explicit time integration code, advantages in both the economy and reliability of computations, is mostly used for crashworthiness simulation.

There are many variables and sources of uncertainty in the vehicle manufacturing process and testing procedure to a greater or less extent – and these in turn influence error and uncertainty in the computational models. An I/U map organizes this uncertainty information. It contains a list of features or inputs of potential importance, a ranking of the importance of each input (from 1, only minor likely impact on prediction error, to 5, significant potential impact), uncertainty of each input, and current status (that is, how is the input currently treated in the model). Table 5 give the complete I/U map for CRASH.

B Summary of Notation and Assumptions

It is common for computer models to also have tuning or calibration parameters $\boldsymbol{u}$ that are unknown. That was not the case in CRASH, but the methodology can be easily extended to this case, and so we include this option in the general description given below. We also allow for an arbitrary number of inputs and 'time parameters', for generality.

B.1 Model inputs

$\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{t})$, where

$\boldsymbol{x}$ is a p-vector of controllable and specified inputs (e.g., velocity)

$\boldsymbol{u}$ is a q-vector of uncertain inputs.

$\boldsymbol{t}$ is an r-vector of arguments for function output.

INPUT		IMP ACT	UNCERTAINTY	CURRENT STATUS
Geometry	Element size about 10mm	5	unspecified	fixed
	Holes < 10mm, fillets, and rounds not meshed	4	unspecified	fixed
	Use of design surfaces, not surfaces after stamping	4	unspecified	fixed
	Spot weld locations are approximated	5	unspecified	fixed
	Thickness variation from location to location	3	Can be specified with c.v. of 2% for whole components	Controllable in some degree
Material Properties	Dynamic stress/strain curves	3	May be approximated with c.v. of 5% for major components	Controllable in some degree
	Spot weld failure force	3	unspecified	fixed
	Joints separation	2	Approximated with 5% c.v.	Controllable
	Damping factor	5	Controllable	fixed
	Friction coefficients between part	4	unspecified	fixed
	Material density	5	unspecified	fixed
Boundary/ Initial Conditions	Vehicle mass/speed	5	Can be matched with the test	fixed
	Barrier variation (plywood condition and barrier angle)	3	unspecified	fixed
	Vehicle test attitude	5	unspecified	fixed
	Testing environment (humidity & temperature)	5	unspecified	fixed
Restraint System	Steering column stroking force	2	5% of c.v.	controllable
	Airbag deployment time	4	5% of c.v.	controllable
	Seatbelt retractor force	2	5% of c.v.	controllable
	Airbag mass flow rate	2	5% of c.v.	controllable
	Occupant position	3	$\pm\frac{1}{2}$ inch on horizontal plan and $\pm\frac{1}{4}$ inch on vertical	controllable

Table 5: The input/uncertainty map for CRASH.

B.2 Gaussian process

A random process $y(z)$ is a Gaussian process with mean function μ and covariance function C if, for any finite set $\{z_1, \ldots, z_n\}$, $(y(z_1), \ldots, y(z_n))'$ is multivariate normal with mean vector $(\mu(z_1), \ldots, \mu(z_n))'$ and covariance matrix $[C(z_i, z_j)]_{ij}$.

B.3 Approximating the mean process

$$y^M(\boldsymbol{z}) = \Psi(\boldsymbol{z})\boldsymbol{\mu}^M + \epsilon^M(\boldsymbol{z}), \tag{B-1}$$

in which we can allow a general mean structure $\Psi(\boldsymbol{z})\boldsymbol{\mu}^M$, where $\Psi(\boldsymbol{z})$ is a $(1 \times k)$ vector function (e.g., $\Psi(\boldsymbol{z}) = x_1 t$ in the Crash model), $\boldsymbol{\mu}^M$ is a $k \times 1$ vector of unknown linear parameters of the

process mean (μ^M in CRASH), and

$$\epsilon^M(\boldsymbol{z}) \mid \boldsymbol{\theta}^M, \boldsymbol{\theta}^T \sim \mathrm{GP}(0, \frac{1}{\lambda^M} c^M(\cdot,\cdot)), \tag{B-2}$$

$$c^M(\boldsymbol{z}, \boldsymbol{z}^\star) = \prod_{i=1}^{p} \exp\{-\beta_i^M |x_i - x_i^\star|^{\alpha_i^M}\} \prod_{i=1}^{q} \exp\{-\beta_{i+p}^M |u_i - u_i^\star|^{\alpha_{i+p}^M}\} c^T(\boldsymbol{t}, \boldsymbol{t}^\star), \tag{B-3}$$

$$c^T(\boldsymbol{t}, \boldsymbol{t}^\star) = \prod_{k=1}^{r} \exp\{-\beta_k^T |t_k - t_k^\star|^{\alpha_k^T}\}, \tag{B-4}$$

$$\boldsymbol{\theta}^M = (\lambda^M, \boldsymbol{\alpha}^M, \boldsymbol{\beta}^M), \quad \boldsymbol{\theta}^T = (\boldsymbol{\alpha}^T, \boldsymbol{\beta}^T). \tag{B-5}$$

B.4 Model for field data

Letting $\tilde{\boldsymbol{z}} = (\boldsymbol{x}, \boldsymbol{t})$,

$$y^F(\tilde{\boldsymbol{z}}) = y^R(\tilde{\boldsymbol{z}}) + \epsilon^F(\tilde{\boldsymbol{z}}), \tag{B-6}$$

$$\epsilon^F(\tilde{\boldsymbol{z}}) \mid \lambda^F, \boldsymbol{\theta}^T \sim \mathrm{GP}(0, \frac{1}{\lambda^F} c^F(\cdot,\cdot)), \tag{B-7}$$

$$c^F(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{z}}^\star) = 1_{\{\boldsymbol{x}=\boldsymbol{x}^\star\}} c^T(\boldsymbol{t}, \boldsymbol{t}^\star). \tag{B-8}$$

B.5 Reality and bias

$$y^R(\tilde{\boldsymbol{z}}) = y^M(\boldsymbol{z}) + b(\tilde{\boldsymbol{z}}), \tag{B-9}$$

$$b(\tilde{\boldsymbol{z}}) \mid \boldsymbol{\theta}^b, \boldsymbol{\theta}^T \sim GP(\mu^b, \frac{1}{\lambda^b} c^b(\cdot,\cdot)), \tag{B-10}$$

$$c^b(\tilde{\boldsymbol{z}}, \tilde{\boldsymbol{z}}^\star) = \prod_{i=1}^{p} \exp\{-\beta_i^b (x_i - x_i^\star)^2\} c^T(\boldsymbol{t}, \boldsymbol{t}^\star), \tag{B-11}$$

$$\boldsymbol{\theta}^b = (\lambda^b, \boldsymbol{\beta}^b) = (\lambda^b, \beta_1^b, \ldots, \beta_p^b). \tag{B-12}$$

Note that we choose the exponents of $|x_i - x_i^\star|$ in the bias correlation function to be 2, reflecting the fact that we expect the bias to be a smooth function. Note also that $\boldsymbol{z}$ in the first expression should be interpreted as $\tilde{\boldsymbol{z}}$ together with the unknown true or 'best fit' value of $\boldsymbol{u}$.

B.6 Model data

Recall that we assume that the functional data $\boldsymbol{t}$ is taken at the same points $D^T = \{\boldsymbol{t}^1, \ldots, \boldsymbol{t}^T\}$ for each value of the other inputs in $D^M = \{(\boldsymbol{x}^1, \boldsymbol{u}^1), \ldots, (\boldsymbol{x}^L, \boldsymbol{u}^L)\}$. Thus the *design space* for the model data is $D^M \times D^T$, so that the augmented model design points are $\boldsymbol{z}_1 = (\boldsymbol{x}^1, \boldsymbol{u}^1, \boldsymbol{t}^1), \ldots, \boldsymbol{z}_T = (\boldsymbol{x}^1, \boldsymbol{u}^1, \boldsymbol{t}^T), \boldsymbol{z}_{T+1} = (\boldsymbol{x}^2, \boldsymbol{u}^2, \boldsymbol{t}^1), \ldots, \boldsymbol{z}_m = (\boldsymbol{x}^L, \boldsymbol{u}^L, \boldsymbol{t}^T)$, where $m = LT$. The corresponding values of the model-run data are denoted by the $m \times 1$ vector $\boldsymbol{y}^M$.

B.7 Field data

Similarly, we assume that the functional data $\boldsymbol{t}$ is taken at the same points $D^T = \{\boldsymbol{t}^1, \ldots, \boldsymbol{t}^T\}$ for each value of the other inputs in $D^F = \{\boldsymbol{x}^{*1}, \ldots, \boldsymbol{x}^{*l}\}$. Thus the *design space* for the field data is $D^F \times D^T$, so that the augmented field design points are $\tilde{\boldsymbol{z}}_1^\star = (\boldsymbol{x}^{*1}, \boldsymbol{t}^1), \ldots, \tilde{\boldsymbol{z}}_T^\star = (\boldsymbol{x}^{*1}, \boldsymbol{t}^T), \tilde{\boldsymbol{z}}_{T+1}^\star = (\boldsymbol{x}^{*2}, \boldsymbol{t}^1), \ldots, \tilde{\boldsymbol{z}}_n^\star = (\boldsymbol{x}^{*l}, \boldsymbol{t}^T)$, where $n = lT$. The corresponding values of the field data are denoted by the $n \times 1$ vector $\boldsymbol{y}^F$.

B.8 Likelihood

Define $C^f(D^g, D^h)$ to be the matrix with (i, j) entry $c^f(\boldsymbol{w}_i, \boldsymbol{w}_j^\star)$ (with any missing variables in the domain of definition of $c^f(\cdot, \cdot)$ being set equal to zero), where $\boldsymbol{w}_i$ and $\boldsymbol{w}_j^\star$ are, respectively, the i^{th} and j^{th} points in the design spaces D^g and D^h. To define the likelihood, we must allow augmentation of the field data by the unknown parameters $\boldsymbol{u}$. When this is needed, we denote the augmented design space by $D_{\boldsymbol{u}}^F$ (which is the same design space as D^F, except that we simply replace $\tilde{\boldsymbol{z}}_i^\star$ by $(\boldsymbol{x}_i^\star, \boldsymbol{u}, \boldsymbol{t}_i^\star)$). Then, given the unknown parameters, the data $\boldsymbol{y} = ((\boldsymbol{y}^M)', (\boldsymbol{y}^F)')'$ has a multivariate normal distribution with mean vector $\boldsymbol{X}\boldsymbol{\theta}$ and covariance matrix $\boldsymbol{\Sigma} \otimes \boldsymbol{C}^T(D^T, D^T)$ where $\boldsymbol{\theta} = ((\boldsymbol{\mu}^M)', \mu^b)'$,

$$\boldsymbol{X} = \begin{pmatrix} \Psi(\boldsymbol{z}_1) & 0 \\ \cdots & \cdots \\ \Psi(\boldsymbol{z}_m) & 0 \\ \Psi(\tilde{\boldsymbol{z}}_1^\star) & 1 \\ \cdots & \cdots \\ \Psi(\tilde{\boldsymbol{z}}_n^\star) & 1 \end{pmatrix}, \tag{B-13}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \frac{1}{\lambda^M}\boldsymbol{C}^M(D^M, D^M) & \frac{1}{\lambda^M}\boldsymbol{C}^M(D^M, D_{\boldsymbol{u}}^F) \\ \frac{1}{\lambda^M}\boldsymbol{C}^M(D^M, D_{\boldsymbol{u}}^F)' & \frac{1}{\lambda^M}\boldsymbol{C}^M(D_{\boldsymbol{u}}^F, D_{\boldsymbol{u}}^F) + \frac{1}{\lambda^b}\boldsymbol{C}^b(D^F, D^F) + \frac{1}{\lambda^F}\boldsymbol{I}_{n\times n} \end{pmatrix}. \tag{B-14}$$

For notational convenience, define $\boldsymbol{\xi} = (\lambda^M, \lambda^b, \lambda^F, \boldsymbol{\beta}^M, \boldsymbol{\beta}^b, \boldsymbol{\beta}^T, \boldsymbol{\alpha}^M, \boldsymbol{\alpha}^T)$, so that $\boldsymbol{\Sigma}(\boldsymbol{\xi}) \equiv \boldsymbol{\Sigma} \otimes \boldsymbol{C}^T(D^T, D^T)$ only depends on $\boldsymbol{\xi}$. Hence, we can write

$$\boldsymbol{y} \mid \boldsymbol{\theta}, \boldsymbol{\xi}, \boldsymbol{u} \sim \mathrm{N}(\boldsymbol{X}\boldsymbol{\theta}, \boldsymbol{\Sigma}(\boldsymbol{\xi})) \ , \tag{B-15}$$

the density for which we denote by $\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}, \boldsymbol{\Sigma}(\boldsymbol{\xi}))$.

B.9 Prior distributions

Since $\boldsymbol{\theta}$ is a location-vector, we utilize the standard constant prior density $p(\boldsymbol{\theta}) = 1$. The prior density for $\boldsymbol{u}$ is given in the I/U map (but recall that there is no $\boldsymbol{u}$ in CRASH). It thus remains only to specify the prior density for $\boldsymbol{\xi}$.

For the smoothness parameters $\boldsymbol{\alpha}^M$ and $\boldsymbol{\alpha}^T$, we choose uniform priors on their ranges (each α_i being in the interval $(1, 2)$). The precisions and range parameters are given independent Exponential priors, with means set at 10 times the marginal likelihood estimate of the parameter. This was essentially an empirically driven choice. When the mean was set at 10 times the marginal MLE, the prior had little influence on the answer (so that the 'double use' of the data in defining the centering of the prior is not a significant sin), and the exponential tail did help considerably with the convergence of the MCMC.

The marginal MLE of $\boldsymbol{\xi}$ is defined as the MLE from the integrated likelihood, obtained by integrating out $\boldsymbol{\theta}$ and $\boldsymbol{u}$ in (B-15) with respect to their priors. With CRASH there is no $\boldsymbol{u}$, and integrating out $\boldsymbol{\theta}$ results in the expression (cf. Berger et al. (2001))

$$L^I(\boldsymbol{\xi} \mid \boldsymbol{y}) = |\boldsymbol{\Sigma}(\boldsymbol{\xi})|^{-1/2} \; |\boldsymbol{X}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi})\boldsymbol{X}|^{-1/2} \exp(-S^2/2) \; , \tag{B-16}$$

where $S^2 = \boldsymbol{y}'\boldsymbol{Q}\boldsymbol{y}$, $\boldsymbol{Q} = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi})\{\boldsymbol{I} - \boldsymbol{X}[\boldsymbol{X}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi})\boldsymbol{X}]^{-1}\boldsymbol{X}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi})\}$.

Paulo (2005) has shown that the Fisher information associated with (B-16) is given by

$$I(\boldsymbol{\xi}) = [\mathrm{tr}(\mathbf{W}_i \; \mathbf{W}_j)/2]_{i,j}$$

where $\mathbf{W}_i = \dot{\boldsymbol{\Sigma}}^i(\boldsymbol{\xi})\boldsymbol{Q}$ and with $\dot{\boldsymbol{\Sigma}}^i(\boldsymbol{\xi})$ denoting the matrix whose entries result from those of $\boldsymbol{\Sigma}(\boldsymbol{\xi})$ by differentiation with respect to ξ_i, the i-th component of $\boldsymbol{\xi}$. Taking advantage of this formula, Paulo (2005) utilizes Fisher's scoring method to numerically obtain the marginal maximum likelihood estimate of $\boldsymbol{\xi}$.

Using this method, we obtained that, in CRASH, the marginal MLE of $(\lambda^M, \lambda^b, \lambda^F, \beta^T, \beta^M, \beta^b)$ is $(0.051, 0.77, 1.24, 2.63, 0.25, 31.7)$, and hence the prior on, e.g., β^T is an exponential distribution with mean 10×2.63. The marginal MLE for α^T and α^M are, respectively, 1.10 and 1.03 (recall that α^b was fixed at 2), but these are not used in the prior specification since these parameters are considered uniformly distributed on their range *a priori*.

C Monte Carlo methods for posterior inference

C.1 Generating samples from the posterior distributions of the unknown parameters

Because of the focus on the application to CRASH, we outline the MCMC analysis only for the case where $\boldsymbol{u}$ is absent. (The more general case can be handled by incorporating $\boldsymbol{u}$ with the $\boldsymbol{\xi}$ block of variables below.)

The posterior distribution is proportional to the product of the likelihood in (B-15) and the priors given in the previous section. The form of $\boldsymbol{\Sigma}(\boldsymbol{\xi})$ does not result in simple expressions for full conditional distributions of parameters contained in $\boldsymbol{\xi}$, so we utilized a block Gibbs sampler

to generate from the posterior distribution. In particular, we work with two full conditional distributions, $p(\boldsymbol{\theta} \mid \boldsymbol{\xi}, \boldsymbol{y})$ and $p(\boldsymbol{\xi} \mid \boldsymbol{\theta}, \boldsymbol{y})$, and, given the current draw $(\boldsymbol{\theta}^{(\text{old})}, \boldsymbol{\xi}^{(\text{old})})$, update according to

(a) $\boldsymbol{\theta}^{(\text{new})} \sim p(\boldsymbol{\theta} \mid \boldsymbol{\xi}^{(\text{old})}, \boldsymbol{y})$

(b) $\boldsymbol{\xi}^{(\text{new})} \sim p(\boldsymbol{\xi} \mid \boldsymbol{\theta}^{(\text{new})}, \boldsymbol{y})$.

Step (a) is easily implemented since $p(\boldsymbol{\theta} \mid \boldsymbol{\xi}, \boldsymbol{y}) \propto \mathrm{N}(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}, \boldsymbol{\Sigma}(\boldsymbol{\xi}))$ and, based on standard calculations, we obtain that $p(\boldsymbol{\theta} \mid \boldsymbol{\xi}, \boldsymbol{y})$ is the density of a $(k+1)$-variate normal distribution with mean vector $(\boldsymbol{X}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi})\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi})\boldsymbol{y}$ and covariance matrix $(\boldsymbol{X}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi})\boldsymbol{X})^{-1}$.

To sample from $p(\boldsymbol{\xi} \mid \boldsymbol{\theta}, \boldsymbol{y}) \propto \mathrm{N}(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}, \boldsymbol{\Sigma}(\boldsymbol{\xi}))\, p(\boldsymbol{\xi})$, where $p(\boldsymbol{\xi})$ denotes the prior defined in the previous appendix, we utilize a Metropolis-Hastings algorithm. To facilitate the choice of a good proposal distribution, we transform $\boldsymbol{\xi}$ to $\boldsymbol{\xi}^{\star} = g(\boldsymbol{\xi}) \in \mathbb{R}^{3+3p+2r}$, where

$$g(\boldsymbol{\xi}) = \boldsymbol{\xi}^{\star} = \left(\log \lambda^{M}, \log \lambda^{b}, \log \lambda^{F}, \log \boldsymbol{\beta}^{M}, \log \boldsymbol{\beta}^{b}, \log \boldsymbol{\beta}^{T}, \log \left(\frac{\boldsymbol{\alpha}^{M} - 1}{2 - \boldsymbol{\alpha}^{M}} \right), \log \left(\frac{\boldsymbol{\alpha}^{T} - 1}{2 - \boldsymbol{\alpha}^{T}} \right) \right)$$

and the operations involving vectors should be interpreted in a componentwise fashion.

In this context, a proposal distribution with which we have had good empirical results is a t-density with ν degrees of freedom, centered at $g(\boldsymbol{\xi}^{(\text{old})})$, and with scale matrix proportional to $I^{-1}(\hat{\boldsymbol{\xi}^{\star}})$. Here, $\hat{\boldsymbol{\xi}^{\star}}$ is the (marginal) MLE for $\boldsymbol{\xi}^{\star}$, obtained from the (marginal) MLE $\hat{\boldsymbol{\xi}}$ for $\boldsymbol{\xi}$, and $I(\hat{\boldsymbol{\xi}^{\star}})$ is the information matrix, obtained from $I(\hat{\boldsymbol{\xi}})$, the corresponding matrix for $\boldsymbol{\xi}$ (both given in the previous appendix). Note that the actual proposal distribution for $\boldsymbol{\xi}$ is induced by the multivariate t proposal for $\boldsymbol{\xi}^{\star}$, through the transformation g, and is not symmetric. Denote such induced density by $q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^{(\text{old})})$. To choose the constant c to multiply $I(\hat{\boldsymbol{\xi}^{\star}})$, we can start by trying $c = 2.4/\sqrt{d}$, where $d = 3 + 3p + 2r$, as suggested by Gelman et al. (1995), and then increasing or decreasing c in order to achieve a suitable acceptance rate.

More explicitly, step (b) is implemented as follows:

$\rightarrow$ Draw $\tilde{\boldsymbol{\xi}^{\star}} \sim t_{\nu}(g(\boldsymbol{\xi}^{(\text{old})}), c\, I^{-1}(\hat{\boldsymbol{\xi}^{\star}}))$

$\rightarrow$ Compute $\tilde{\boldsymbol{\xi}} = g^{-1}(\tilde{\boldsymbol{\xi}^{\star}})$

$\rightarrow$ Set $\boldsymbol{\xi}^{(\text{new})} = \tilde{\boldsymbol{\xi}}$, with probability ρ, and $\boldsymbol{\xi}^{(\text{new})} = \boldsymbol{\xi}^{(\text{old})}$, with probability $1 - \rho$, where

$$\rho = \rho(\boldsymbol{\xi}^{(\text{old})}, \tilde{\boldsymbol{\xi}} \mid \boldsymbol{\theta}^{(\text{new})}, \boldsymbol{y}) = \min \left\{ 1, \frac{\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}^{(\text{new})}, \boldsymbol{\Sigma}(\tilde{\boldsymbol{\xi}}))\, p(\tilde{\boldsymbol{\xi}})}{\mathrm{N}(\boldsymbol{y} \mid \boldsymbol{X}\boldsymbol{\theta}^{(\text{new})}, \boldsymbol{\Sigma}(\boldsymbol{\xi}^{(\text{old})}))\, p(\boldsymbol{\xi}^{(\text{old})})} \frac{q(\boldsymbol{\xi}^{(\text{old})} \mid \tilde{\boldsymbol{\xi}})}{q(\tilde{\boldsymbol{\xi}} \mid \boldsymbol{\xi}^{(\text{old})})} \right\}. \quad \text{(C-1)}$$

For the CRASH example, a proposal distribution that gave good mixing of the chain had $\nu = 5$ and $c = 1.5$.

C.2 Simulating realizations from the posterior distribution of functions

Again we restrict consideration to the CRASH situation, with $\boldsymbol{u}$ absent and t a scalar. One usually wants to predict functions such as $y^M(\boldsymbol{x},t)$, $y^R(\boldsymbol{x},t)$ and $b(\boldsymbol{x},t)$ at a finer set of t values than D^T. This is possible because the computations involved in prediction are much faster than those involved in obtaining a posterior sample of unknown parameters. Thus suppose prediction is desired at the n_P time points $D_t^P = \{t_1, \ldots, t_{n_P}\}$. (Note that Kronecker product simplifications will not then be possible, but that is not a problem here.) If prediction is desired at input $\boldsymbol{x}$, it follows that it is desired to predict the functions at the combined inputs $D^P = \{(\boldsymbol{x}, t_1), \ldots, (\boldsymbol{x}, t_{n_P})\}$.

We need to generate a sample from the posterior distribution of the combined vector

$$\boldsymbol{r}(D^P) = \begin{pmatrix} \boldsymbol{y}^M(D^P) \\ \boldsymbol{b}(D^P) \end{pmatrix} \tag{C-2}$$

where $\boldsymbol{y}^M(D^P)$ and $\boldsymbol{b}(D^P)$ denote the vectors of sampled y^M and b function points respectively. Conditional on $\boldsymbol{\theta}, \boldsymbol{\xi}$, it follows from Appendix B that this vector is multivariate normal with prior mean vector $E(\boldsymbol{r}(D^P)) = \boldsymbol{X}(D^P)\boldsymbol{\theta}$ and prior covariance matrix

$$\mathrm{Cov}(\boldsymbol{r}(D^P)) = \begin{pmatrix} \frac{1}{\lambda^M}\boldsymbol{C}^M(D^P, D^P) & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{\lambda^b}\boldsymbol{C}^b(D^P, D^P) \end{pmatrix},$$

where $\boldsymbol{X}(D^P)$ is defined as in (B-13) but with $n = m = n_P$ and the $\{\boldsymbol{z}_i\}$ and $\{\boldsymbol{z}_i^\star\}$ replaced by the elements of D^P.

The MCMC algorithm produces a sample from the posterior distribution of the parameters $(\boldsymbol{\theta}, \boldsymbol{\xi})$ of the Gaussian processes. For each point $(\boldsymbol{\theta}_i, \boldsymbol{\xi}_i)$ from this sample, the posterior distribution $[\boldsymbol{r}(D^P) \mid \boldsymbol{\theta}_i, \boldsymbol{\xi}_i, \boldsymbol{y}]$ is also multivariate normal with mean and covariance given by

$$\boldsymbol{\mu}_i^r = \boldsymbol{X}(D^P)\boldsymbol{\theta}_i + \mathrm{Cov}(\boldsymbol{r}(D^P), \boldsymbol{y})\boldsymbol{\zeta}_i \tag{C-3}$$

$$\boldsymbol{\Sigma}_i^r = \mathrm{Cov}(\boldsymbol{r}(D^P)) - \mathrm{Cov}(\boldsymbol{r}(D^P), \boldsymbol{y})\{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi}_i) \otimes \boldsymbol{C}^T(D^T, D^T)^{-1}\}\,\mathrm{Cov}(\boldsymbol{r}(D^P), \boldsymbol{y})', \tag{C-4}$$

where $\boldsymbol{\zeta}_i = \{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\xi}_i) \otimes \boldsymbol{C}^T(D^T, D^T)^{-1}\}\{\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}_i\}$ and $\mathrm{Cov}(\boldsymbol{r}(D^P), \boldsymbol{y})$ is the matrix of prior covariances between points in $\boldsymbol{r}(D^P)$ and points in $\boldsymbol{y}$, which can be expressed in block form as

$$\mathrm{Cov}(\boldsymbol{r}(D^P), \boldsymbol{y}) = \begin{pmatrix} \frac{1}{\lambda^M}\boldsymbol{C}^M(D^P, D^M) & \frac{1}{\lambda^M}\boldsymbol{C}^M(D^P, D^F) \\ \boldsymbol{0} & \frac{1}{\lambda^b}\boldsymbol{C}^{b,T}(D^P, D^F) \end{pmatrix}. \tag{C-5}$$

We can use standard simulation to generate a vector $\boldsymbol{z}$ of n_P independent $\mathrm{N}(0,1)$ variates. Then $\boldsymbol{\mu}_i^r + (\boldsymbol{\Sigma}_i^r)^{1/2}\boldsymbol{z}$ is a draw from the posterior distribution of $\boldsymbol{r}(D^P)$, where $(\boldsymbol{\Sigma}_i^r)^{1/2}$ denotes the Cholesky decomposition of $\boldsymbol{\Sigma}_i^r$.

For the results reported earlier in the paper for CRASH, the MCMC computations were done

by discarding the first 10000 iterations (burn-in), storing the next 40000 values of the unknown parameters generated by the MCMC, and selecting every 40th point from this collection. These 1000 selections were then used to generate 1000 realizations of $y^M(\boldsymbol{x}, t)$, $y^R(\boldsymbol{x}, t)$, and $y^F(\boldsymbol{x}, t)$ at the inputs $\boldsymbol{x}$ of interest, for t in $D_t^P = (3, 6, \ldots, 81)$.

C.3 Updating

The model approximation is exact only at the observed model-run data points. Sometimes the values of the model output are also constrained at other points; for instance, in CRASH, all relative velocity curves are (by definition) equal to zero at $t = 0$. If one tried to incorporate this constraint apriori, the Kronecker product computational simplification would no longer apply, resulting in an impractical MCMC algorithm. Such constraints can, however, be easily incorporated at the prediction stage, resulting in an implementable and reasonable approximation to the exact answer.

Another crucial instance of the conditioning idea is when an additional computer model run, $y^M(\boldsymbol{x}, t)$, becomes available. Indeed, this is how the computer model is often utilized for a new input $\boldsymbol{x}$ of interest. The difficulty here is that it may not be feasible to go back and re-run the entire MCMC computation with this new data point. The solution to both these problems is simply to condition the existing posterior on the additional constraint or data point; i.e., use the additional information in the Kalman filter part of the analysis, but not to obtain the posterior for tuning parameters or parameters in the Gaussian processes.

To illustrate this process, consider updating the posterior by the constraint that the relative velocity is 0 at $t = 0$. This is equivalent to introducing the 'new (extended) observations' $y^M(x_1, 0) = y^R(x_1, 0) = 0$. Let $\boldsymbol{\Sigma}^+$ denote the prior covariance matrix of the new data vector

$$\boldsymbol{y}^+ = \begin{pmatrix} \boldsymbol{y}^M \\ \boldsymbol{y}^F \\ y^M(x_1, 0) \\ y^R(x_1, 0) \end{pmatrix}. \tag{C-6}$$

$\boldsymbol{\Sigma}^+$ can be written in block matrix form as

$$\begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma}_{12}^+ \\ \boldsymbol{\Sigma}_{12}^{+\prime} & \boldsymbol{\Sigma}_{22}^+ \end{pmatrix}. \tag{C-7}$$

$\boldsymbol{\Sigma}_{22}^+$ is simply the covariance matrix for the new data

$$\boldsymbol{\Sigma}_{22}^+ = \frac{1}{\lambda^M} \begin{pmatrix} 1 & 1 \\ 1 & (1 + \lambda^M/\lambda^b) \end{pmatrix}. \tag{C-8}$$

For $\boldsymbol{\Sigma}_{12}^{+}$ we can use the Kronecker product structure

$$\begin{aligned}\boldsymbol{\Sigma}_{12}^{+} &= \begin{pmatrix} \mathrm{Cov}(\boldsymbol{y}^M, y^M(x_1,0)) & \mathrm{Cov}(\boldsymbol{y}^M, y^R(x_1,0)) \\ \mathrm{Cov}(\boldsymbol{y}^F, y^M(x_1,0)) & \mathrm{Cov}(\boldsymbol{y}^F, y^R(x_1,0)) \end{pmatrix} \\ &= \tfrac{1}{\lambda^M} \begin{pmatrix} \boldsymbol{C}^M(D^M,\{x_1\}) & \boldsymbol{C}^M(D^M,\{x_1\}) \\ \boldsymbol{C}^M(D^F,\{x_1\}) & \boldsymbol{C}^M(D^F,\{x_1\}) + (\lambda^M/\lambda^b)\boldsymbol{C}^b(D^F,\{x_1\}) \end{pmatrix} \otimes \boldsymbol{C}^T. \end{aligned} \tag{C-9}$$

To calculate the inverse $(\boldsymbol{\Sigma}^{+})^{-1}$ we make use of a standard block matrix result which uses the precalculated $\boldsymbol{\Sigma}^{-1}$

$$(\boldsymbol{\Sigma}^{+})^{-1} = \begin{pmatrix} \boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{12}^{+}\boldsymbol{G}{\boldsymbol{\Sigma}_{12}^{+}}^{T}\boldsymbol{\Sigma}^{-1} & -\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{12}^{+}\boldsymbol{G} \\ -\boldsymbol{G}{\boldsymbol{\Sigma}_{12}^{+}}^{T}\boldsymbol{\Sigma}^{-1} & \boldsymbol{G} \end{pmatrix} \tag{C-10}$$

where

$$\boldsymbol{G} = (\boldsymbol{\Sigma}_{22}^{+} - {\boldsymbol{\Sigma}_{12}^{+}}^{T}\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{12}^{+})^{-1}. \tag{C-11}$$

Any posterior quantities can now be recalculated by replacing the inverse of the data covariance matrix $\boldsymbol{\Sigma}^{-1}$ by $(\boldsymbol{\Sigma}^{+})^{-1}$ and replacing $\boldsymbol{y}$ by $\boldsymbol{y}^{+}$.

C.4 Computing marginal likelihoods

Here we describe the details of the methods used to compute the marginal likelihoods under each of the competing models, $\mathcal{M}_i$, $i = 0, 1$, which are given by (6.2). As mentioned before, the structure of the competing models is very similar, and hence in what follows we drop the model index i for clarity of exposition.

We first consider the method of Chib and Jeliazkov (2001). Note that, for any fixed value $\boldsymbol{\xi}^o$ of $\boldsymbol{\xi}$,

$$m(\boldsymbol{y}) = \frac{\kappa\ L^I(\boldsymbol{\xi}^o \mid \boldsymbol{y})\ p(\boldsymbol{\xi}^o)}{p(\boldsymbol{\xi}^o \mid \boldsymbol{y})}\,, \tag{C-12}$$

where $\kappa = (2\pi)^{-(m+n-d)/2}$ with d denoting the rank of the matrix $\boldsymbol{X}$ (k in the model $\mathcal{M}_0$ and $k+1$ in the model $\mathcal{M}_1$). Also, L^I is given by the right-hand side of (B-16), and $p(\cdot)$ and $p(\cdot \mid \boldsymbol{y})$ denote, respectively, the prior and marginal posterior densities of $\boldsymbol{\xi}$, normalizing constants included. As a consequence of the formula above, as noted by Chib and Jeliazkov and by other authors before them, all that is needed to obtain an estimate of $m(\boldsymbol{y})$ is an estimate of the marginal posterior of $\boldsymbol{\xi}$ evaluated at a particular value of the parameter vector, $\boldsymbol{\xi}^o$, e.g. its posterior mean or maximum likelihood estimate.

To that end, Chib and Jeliazkov (2001) note that the acceptance probability (C-1) is such that it satisfies detailed balance, and hence

$$\rho(\boldsymbol{\xi}, \boldsymbol{\xi}^o \mid \boldsymbol{\theta}, \boldsymbol{y})\ q(\boldsymbol{\xi}^o \mid \boldsymbol{\xi}, \boldsymbol{y})\ p(\boldsymbol{\xi} \mid \boldsymbol{\theta}, \boldsymbol{y}) = \rho(\boldsymbol{\xi}^o, \boldsymbol{\xi} \mid \boldsymbol{\theta}, \boldsymbol{y})\ q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^o, \boldsymbol{y})\ p(\boldsymbol{\xi}^o \mid \boldsymbol{\theta}, \boldsymbol{y})\,, \tag{C-13}$$

where $p(\cdot \mid \boldsymbol{\theta}, \boldsymbol{y})$ denotes the full-conditional density of $\boldsymbol{\xi}$, normalizing constants included.

Multiplying both sides of (C-13) by $p(\boldsymbol{\theta} \mid \boldsymbol{y})$, integrating over $\boldsymbol{\xi}$ and $\boldsymbol{\theta}$ and applying standard calculations we have

$$p(\boldsymbol{\xi}^o \mid \boldsymbol{y}) = \frac{\int \rho(\boldsymbol{\xi}, \boldsymbol{\xi}^o \mid \boldsymbol{\theta}, \boldsymbol{y})\ q(\boldsymbol{\xi}^o \mid \boldsymbol{\xi}, \boldsymbol{y})\ p(\boldsymbol{\theta}, \boldsymbol{\xi} \mid \boldsymbol{y})\ d\boldsymbol{\theta} d\boldsymbol{\xi}}{\int \rho(\boldsymbol{\xi}^o, \boldsymbol{\xi} \mid \boldsymbol{\theta}, \boldsymbol{y})\ q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^o, \boldsymbol{y})\ p(\boldsymbol{\theta} \mid \boldsymbol{\xi}^o, \boldsymbol{y})\ d\boldsymbol{\theta} d\boldsymbol{\xi}}. \tag{C-14}$$

Based on draws $(\boldsymbol{\xi}_l, \boldsymbol{\theta}_l)$, $l = 1, ..., N_1$, from $p(\boldsymbol{\theta}, \boldsymbol{\xi} \mid \boldsymbol{y})$, a Monte Carlo estimate for the integral in the numerator is

$$N_1^{-1} \sum_{l=1}^{N_1} \rho(\boldsymbol{\xi}_l, \boldsymbol{\xi}^o \mid \boldsymbol{\theta}_l, \boldsymbol{y})\ q(\boldsymbol{\xi}^o \mid \boldsymbol{\xi}_l, \boldsymbol{y}). \tag{C-15}$$

For the denominator, our setting is simpler than the general one in Chib and Jeliazkov (2001) since $p(\boldsymbol{\theta} \mid \boldsymbol{\xi}^o, \boldsymbol{y})$ has a convenient form (cf. Appendix C.1) and the proposal density doesn't depend on $\boldsymbol{\theta}$. For a Monte Carlo estimate for the integral in the denominator, for $j = 1, ..., N_2$ we draw (i.i.d.) $\boldsymbol{\xi}_j \sim q(\boldsymbol{\xi} \mid \boldsymbol{\xi}^o, \boldsymbol{y})$ and (independently) draw (i.i.d.) $\boldsymbol{\theta}_j \sim p(\boldsymbol{\theta} \mid \boldsymbol{\xi}^o, \boldsymbol{y})$. Then the estimate is

$$N_2^{-1} \sum_{j=1}^{N_2} \rho(\boldsymbol{\xi}^o, \boldsymbol{\xi}_j \mid \boldsymbol{\theta}_j, \boldsymbol{y}). \tag{C-16}$$

Finally, (C-15) and (C-16) along with (C-14) and (C-12) yield the estimate of $m(\boldsymbol{y})$.

The importance sampling approach is much simpler to explain. Indeed,

$$m(\boldsymbol{y}) = \kappa \int L^I(\boldsymbol{\xi} \mid \boldsymbol{y})\ p(\boldsymbol{\xi})\ d\boldsymbol{\xi} = \int \frac{\kappa\ L^I(\boldsymbol{\xi} \mid \boldsymbol{y})\ p(\boldsymbol{\xi})}{q(\boldsymbol{\xi})}\ q(\boldsymbol{\xi})\ d\boldsymbol{\xi} \tag{C-17}$$

so that if, for $j = 1, \ldots, N$, we draw i.i.d $\boldsymbol{\xi}_j \sim q(\boldsymbol{\xi})$, a Monte Carlo estimate of $m(\boldsymbol{y})$ is given by

$$N^{-1}\ \kappa \sum_{j=1}^{N} \frac{L^I(\boldsymbol{\xi}_j \mid \boldsymbol{y})\ p(\boldsymbol{\xi}_j)}{q(\boldsymbol{\xi}_j)}\ . \tag{C-18}$$

For the importance function q, we used a variation around the proposal distribution we constructed in Appendix C.1 to sample from the full-conditional of $\boldsymbol{\xi}$. Indeed, rather than centering the t distribution at the previous iteration of the chain, we center it at the posterior mean of $g(\boldsymbol{\xi})$, and the covariance matrix is set at the posterior variance of $g(\boldsymbol{\xi})$. These posterior moments are estimated using the MCMC sample from the posterior distribution of the parameters of the model.

D Hierarchical model assumptions and analysis

Suppose we have K related models. For the i^{th} model, we use the subscript i for the parameters $\boldsymbol{\mu}^M$, and λ^b that are allowed to vary between models. The assumptions made in the hierarchical

analysis are as follows.

1. All models and field data have common α's, common β's, common μ^b's, common λ^M's, and common λ^F's. The choice of priors for these is explained in Appendix B.9.

2. $\boldsymbol{\mu}_i^M \mid \boldsymbol{\mu}, \boldsymbol{\Upsilon} \sim \mathrm{N}_k(\boldsymbol{\mu}, \boldsymbol{\Upsilon})$. The hyperparameters $\boldsymbol{\mu}$ and $\boldsymbol{\Upsilon}$ are assigned a reference prior, following Yang and Chen (1995). This prior is given as follows, conditional on the model GASP parameters:

   - Note that $\boldsymbol{y}_i^M$, the model-run data from computer model i, has a $\mathrm{N}(\boldsymbol{X}_i^M \boldsymbol{\mu}_i^M, \boldsymbol{C}_i/\lambda^M)$ distribution, where $\boldsymbol{X}_i^M \equiv (\Psi(\boldsymbol{z}_{i1})', \ldots, \Psi(\boldsymbol{z}_{im_i})')'$ and $\boldsymbol{C}_i \equiv \boldsymbol{C}_i^M(D_i^M, D_i^M) \otimes \boldsymbol{C}^T(D^T, D^T)$.
   - The reference prior can be written $p(\boldsymbol{\mu}, \boldsymbol{\Omega} \mid \lambda^M, \boldsymbol{\theta}^M, \boldsymbol{\theta}^T) = |\boldsymbol{G} \sum_{i=1}^K (\boldsymbol{B}_i \otimes \boldsymbol{B}_i) \boldsymbol{G}'|^{1/2}$, in terms of the random variables $\boldsymbol{\mu}$ and $\boldsymbol{\Omega} = \lambda^M \boldsymbol{\Upsilon}$, where
     - $\boldsymbol{B}_i \equiv \boldsymbol{X}_i^{M\prime} (\boldsymbol{X}_i^M \boldsymbol{\Omega} \boldsymbol{X}_i^{M\prime} + \boldsymbol{C}_i)^{-1} \boldsymbol{X}_i^M$,
     - $\boldsymbol{G}$ denotes the $(K(K+1)/2) \times K^2$ matrix of zeroes and ones formed as

       $$\partial(\mathrm{vec}(V))/\partial(\mathrm{vecp}(V)) \ ,$$

       where V is any $K \times K$ symmetric matrix with nonzero entries, $\mathrm{vec}(\cdot)$ is the matrix operator that arranges the columns of a matrix into one long column, and $\mathrm{vecp}(\cdot)$ is the matrix operator that arranges the columns of lower left corner of a symmetric matrix into one long column.

   When $k = 1$ as for CRASH, so that the hierarchical prior is $\mu_i^M \mid \mu, \tau \sim \mathrm{N}(\mu, \tau^{-1})$, the reference prior can be shown (using the matrix identity $(\boldsymbol{A} + \boldsymbol{w}\boldsymbol{w}')^{-1} = \boldsymbol{A}^{-1} - (1 + \boldsymbol{w}'\boldsymbol{A}^{-1}\boldsymbol{w})^{-1}\boldsymbol{A}^{-1}\boldsymbol{w}\boldsymbol{w}'\boldsymbol{A}^{-1}$) to be

   $$p(\mu, \tau \mid \lambda^M, \boldsymbol{\theta}^M, \boldsymbol{\theta}^T) = \frac{1}{\tau} \left| \sum_{i=1}^K \left(1 + \frac{\tau}{\lambda^M \boldsymbol{X}_i^M \boldsymbol{C}_i^{-1} \boldsymbol{X}_i^{M\prime}} \right)^{-2} \right|^{1/2} .$$

   (For $k > 1$, it is probably most reasonable to let $\boldsymbol{\Upsilon}$ be a diagonal matrix, since there is such limited and indirect data concerning these hyperparameters.)

3. $\log(\lambda_i^b) \sim N(\eta, 4q^2)$, where q is the proportional variation in the bias that is expected among models (e.g., $q = 0.1$) and η is given a constant prior.

Finally, we note that the Monte Carlo methods described in Section C can be extended to provide a simulation-based model fitting technique for the hierarchical model (see the analysis presented in Section 5).

In terms of the sampling mechanism, and restricting attention to the case illustrated by CRASH, where $k = 1$, the full conditionals of μ_i^M, μ^b, μ, and η are Normal, and hence can be sampled directly.

The common parameters are sampled as a block, using a Metroplis-Hastings step with a proposal similar to the one used in the single barrier analysis, except that now λ^b is excluded from this step as it is not common across the different computer models. In the CRASH example, we started by using the scale matrix utilized in the straight frontal impact analysis as the scale matrix in the t density proposal. We then used pilot runs to re-estimate this matrix.

To sample from the full conditional of the λ_i^b, we follow a similar strategy: we reparametrize in terms of $\log \lambda_i^b$ and use a t density proposal centered at the previous iteration and with a variance that is initially set at a multiple of the corresponding entry of the covariance matrix used when sampling from the proposal distribution of the straight frontal impact model. A pilot run is then used to estimate individual variances for each of the λ_i^b.

To draw from the full conditional distribution of τ requires more tuning. We again reparameterized in terms of the log-precision and utilized the idea of sampling from a t density proposal, centered at the previous iteration and with a variance that in this case had to be tuned by trial and error. Here, there does not seem to be any natural value at which to start the pilot runs.