INSTITUTE FOR DEFENSE ANALYSES

Metamodeling Techniques for Verification and Validation of Modeling and Simulation Data

John T. Haman, Project Leader

Curtis G. Miller

September 2022

Public release approved. Distribution is unlimited.

IDA Paper P-33230

Log: H 2022-000374

INSTITUTE FOR DEFENSE ANALYSES
730 East Glebe Road
Alexandria, Virginia 22305

About This Publication
This work was conducted by the Institute for Defense Analyses (IDA) under contract HQ0034-19-D-0001, Task BD-09-2299(90), Methods Development," for the Office of the Director, Operational Test and Evaluation. The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments
The IDA Technical Review Committee was chaired by Dr. V. Bram Lillard and consisted of Dr. Kelly M. Avery, Dr. Elliot A.J. Bartis, Dr. Andrew M. Bluestein, Dr. Madalina Colci O'Hara, and Dr. Megan L. Gelsinger from the Operational Evaluation Division.

For more information:
Dr. John T. Haman, Project Leader
jhaman@ida.org • (703) 845-2132

Dr. V. Bram Lillard, Director, Operational Evaluation Division
vlillard@ida.org • (703) 845-2230

INSTITUTE FOR DEFENSE ANALYSES

IDA Paper P-33230

Metamodeling Techniques for Verification and Validation of Modeling and Simulation Data

John T. Haman, Project Leader

Curtis G. Miller

Executive Summary

Modeling and simulation (M&S) outputs help the Director, Operational Test and Evaluation (DOT&E), assess the effectiveness, survivability, lethality, and suitability of systems. DOT&E desires the same degree of understanding and confidence in M&S outputs as in live test results; that is, models and simulators must be sufficiently verified and validated. Verification and validation (V&V) is a rigorous process that DOT&E has sought to improve since 2016 through policy, training, and methodological advancements (DOT&E 2016, 2017; Haman et al. 2022; Wojton, Avery, Freeman, et al. 2019; Wojton, Avery, Yi, et al. 2021).

This paper's purpose is to improve the state of V&V methods applied inside and outside the testable region[1] by recommending and demonstrating a set of statistical techniques—*metamodels* (also called *statistical emulators*)—to the M&S community. In 2017, DOT&E recommended that M&S evaluators use metamodels to understand M&S *outside* the testable region of the operational space.

A metamodel can accurately summarize the output of M&S, thus allowing model and system experts to better compare M&S with their subject matter expertise. Metamodeling thus combines subject matter expertise and model expertise, allowing experts to make judgments about the M&S in the most difficult modeling conditions. In short, a metamodel helps experts discover implausible M&S predictions. We expand upon DOT&E's existing guidance about metamodel usage by creating methodological recommendations the M&S community could apply to its activities.

While an M&S environment aims to emulate a live test, a metamodel aims to emulate an M&S environment. The advantages of a metamodel are numerous since they are:

1. Fast,
2. Portable,
3. Amenable to uncertainty quantification, and
4. Easier to understand than the M&S environment itself.

That said, metamodels will never render M&S environments or live testing obsolete. M&S environments are developed based on the physics and logic of the phenomenon being modeled. Metamodels are statistical fits with no understanding of why the outputs they were fit with emerged the way they did; they simply make predictions, matching what was

[1] The testable region is the region of the operational space for which evaluators have both live data and simulation outputs for comparison.

observed from an M&S environment. And since metamodels ultimately are derived from M&S environments and M&S environment outputs are not interchangeable with live test data, metamodels (being a surrogate of a surrogate) will never replace live testing either.

Metamodeling Techniques

Metamodeling is a methodology for analyzing, understanding, verifying, and validating M&S environments. A metamodel attempts to summarize output from an M&S environment with a statistical fit that interpolates or smooths the M&S output, essentially changing a complex computer simulation into a mathematical formula.

While an M&S environment usually is computationally complex, a metamodel generally is smaller and portable. The metamodel is a way to make conclusions about the M&S environment, as it summarizes the observed M&S outputs and predicts outputs at unobserved points. This summarization and prediction help check whether M&S outputs appear plausible and help identify an M&S environment that produces obviously incorrect outputs.

M&S environments vary widely in nature and implementation, and knowing how an M&S environment works is essential for analyzing its output and properly constructing metamodels for it. In our metamodeling framework, an important determinant of the statistical strategy is stochastic noise, or variation in the outcomes of an experiment such that no two runs of the experiment are identical or are identical only when using a common random seed.

We split M&S output based on whether the response variable of interest is deterministic or stochastic and discrete or continuous.[2] We make the following recommendations:

- For a deterministic, discrete response variable (e.g., threat classification or missile firing doctrine), we recommend nearest neighbor or decision tree interpolators. These interpolators are flexible enough to work in a wide variety of circumstances. Decision trees, in particular, may be dense but might be human readable.
- For a deterministic, continuous response variable (e.g., threat radar cross-section of a digital threat model), we recommend Gaussian process (GP) interpolation. This yields a mathematical function that connects observed M&S outputs and makes predictions at unobserved conditions based on observed outputs while accompanying those predictions with uncertainty estimates.

[2] A deterministic response has no random variation, while a stochastic response varies when the M&S environment generates output under identical conditions.

- For a stochastic response variable (either continuous or discrete, such as whether a threat was engaged or the range at which a threat was engaged), we recommend a generalized additive model (GAM). This statistical modeling framework works in many contexts and can yield flexible but also human-interpretable fits.

Evaluating Metamodel Fits

All of the techniques described above include many options for fitting a metamodel to M&S output. Analysts need M&S outputs to help them decide which of those options to use when estimating a metamodel. Broadly speaking, a metamodel needs to describe M&S outputs well, and metamodeling choices need to facilitate a good description of the M&S environment.

A well-calibrated metamodel makes predictions that generally match M&S environment outputs. We observe some M&S outputs and use those outputs for estimating a metamodel. A model that closely matches outputs already observed—known as in-sample outputs—is not good enough; the metamodel needs to describe hypothetical unobserved outputs—known as out-of-sample outputs—well too. We recommend using output splitting techniques that use only some of the available M&S outputs for model fitting and that use other outputs to evaluate the metamodel's predictive performance for data not involved in fitting. One should plan the use of such techniques prior to generating M&S outputs, and any design of experiments (DOE) plan for collecting M&S outputs should accommodate such techniques. We more precisely define model quality using metrics that either describe how much metamodel predictions deviate from observed outcomes or state how likely the outputs would be given the metamodel we fitted. Visualizations such as calibration plots show the relationship between predicted values and outputs.

If our metamodel assessment metrics remain consistent between training, screening, and evaluation sets, we can rely on its predictions; otherwise, the metamodel may be overfitting M&S environment outputs, meaning that it mostly repeats outputs observed in the training set without learning the larger patterns of the M&S environment it needs to emulate. The metamodel may also simply fail to make precise predictions—a phenomenon known as underfitting—though we do the best we can when working with the training set for a metamodel to make precise predictions without overfitting. If we are satisfied with the precision of the metamodel's predictions and if we observe no evidence of overfitting as we evaluate the model's performance with observations not used directly for fitting, we may declare the metamodel a sufficient representation of the M&S environment and use it.

In summary, we recommend:

- Using statistical metrics to assess a metamodel's ability to match observed M&S outputs;
- Using said metrics to make choices for estimating metamodels based on a resulting metamodel's ability to predict out-of-sample M&S outputs, as estimated by cross-validation and output splitting;
- Using visualizations to present the relationship between metamodel predictions and M&S outputs;
- Checking whether candidate metamodels overfit the training outputs by checking the metamodel's performance with screening and evaluation sets; and
- Accommodating data splitting in any plan to collect M&S observations for metamodel estimation.

Experimental Designs for Metamodeling

Metamodels require the collection of M&S outputs before a metamodel can be fit; DOE should guide output collection. We divide DOE into two classes: parametric DOE and space-filling designs (SFDs). Parametric DOE consists of DOE methodology designed for parametric regression metamodeling, including linear statistical models. SFDs place design points in such a way that the factor space is "filled" with points and the designs are model independent.

Many designs for parametric statistical metamodels use design points selected to produce good statistical properties in the fitted metamodels. Simple linear statistical models may prefer points near the edges of the factor space since the edges often are the best locations for placing points; such points minimize the statistical error of the metamodel coefficients.

SFDs do not attempt to be good for a *specified* model and instead try to explore the whole factor space. They are agnostic to the statistical methods planned, which leads to designs that should work well for the more flexible statistical fitting techniques recommended in this paper.

Both approaches have advantages and disadvantages, and the type of study dictates which approach is more appropriate. We make the following recommendations:

- In situations where variation in outcomes makes estimating effects difficult, statistical error is the biggest concern; parametric DOE should be used, such as D-optimal designs that attempt to minimize estimation error in the metamodel's parameters.

- In situations where variation in outcomes either does not exist or does not make discovering relative effects too difficult, we can tackle statistical model uncertainty using SFDs. In addition to the designs mentioned in Wojton, Avery, Yi, et al. (2021), we have another recommendation—the MaxPro design. This design sufficiently spreads points throughout the design space, ensures that all factors are adequately covered even if other factors are ignored, and can economically handle both categorical and quantitative factors. The MaxPro design can be used in instances where Wojton, Avery, Yi, et al. (2021) recommend the sliced Latin hypersquare design (SLHD), but when creating slices for all combinations of categorical factors would result in unreasonably large sample sizes to execute.

Software for Metamodeling

Many statistical software packages can estimate the metamodels we recommend and generate appropriate DOEs. JMP and JMP Pro can fit generalized linear statistical models, GP models, decision trees, and nearest neighbor interpolators and can make some DOEs. The free and open-source statistical software R can estimate every metamodel featured in this paper, and it was the software actually used in the paper's example analysis of a paper airplane flight simulator. R can also generate SFDs we recommend. In particular, we recommend the R package **mgcv** for estimating GAMs, **GPfit** for estimating GPs, and the **SLHD** or **MaxPro** packages for generating SFDs with both categorical and quantitative factors.

All our recommended techniques, from data collection to model fitting and evaluation, are demonstrated in this paper for a toy M&S environment simulating the flight paths of paper airplanes. The simulator runs on a common laptop, has no distribution restrictions, is relatively easy to use, and should be easily understood (most readers have likely thrown a paper airplane). This paper is accompanied with R code that runs the data example.

Contents

1. Introduction

Metamodeling is a methodology for analyzing, understanding, verifying, and validating modeling and simulation (M&S) environments. A *metamodel* (also called a *statistical emulator*) summarizes M&S output with a statistical fit that interpolates or smooths the M&S output, essentially changing a complex computer simulation into a mathematical formula (see Figure 1-1). The metamodel's purpose is to help M&S stakeholders understand the M&S output, quantify the uncertainty in the M&S predictions, and be a useful product in and of itself in contexts where the M&S cannot be used practically, either because of time or accessibility.

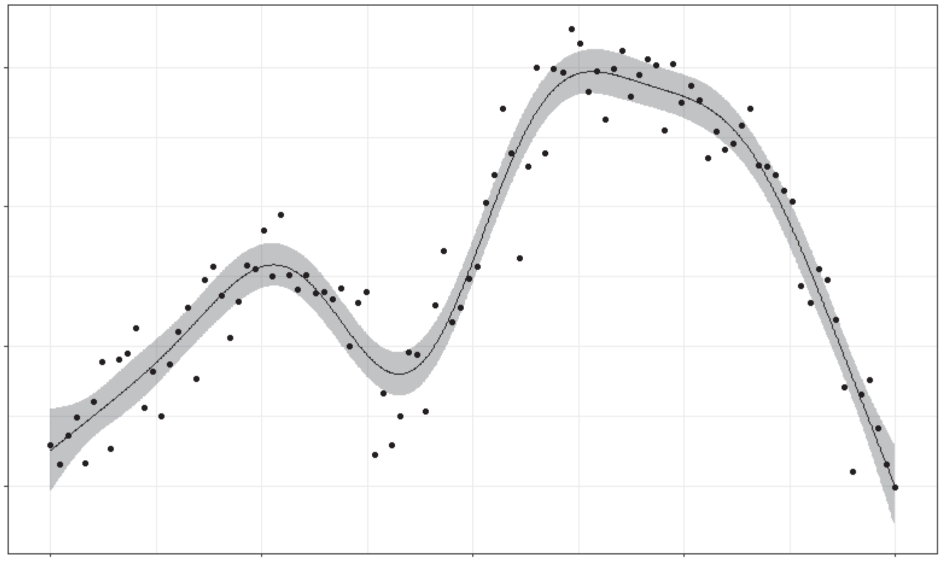

Figure 1-1. A hypothetical metamodel for an M&S environment with a stochastic (i.e., random) response variable depending on a single factor. The metamodel estimates the mean value of the response variable depending on the factor, with the estimate accompanied by a pointwise confidence interval shown as a shaded region; for any selected factor level, the upper and lower bounds of the shaded region represent the confidence interval for the mean response at that factor level.

While an M&S environment usually is computationally complex, a metamodel generally is smaller and portable. For example, the Navy's Environment Centric Weapons Analysis Facility (ECWAF) is a hardware-in-the-loop M&S environment for simulating torpedo performance. It consists of multiple computers, filling a large room, that run in real time while connected to a torpedo's motherboard; some computers are dedicated to calculating underwater sound trajectories. One could completely describe in a short email a fitted logistic regression statistical metamodel predicting hit probabilities based on ECWAF outputs.

The Director, Operational Test and Evaluation (DOT&E), considers M&S output in addition to live testing when assessing the effectiveness, suitability, survivability, and lethality of many systems under their oversight. The Institute for Defense Analyses has published recommendations for generating experimental designs to statistically study M&S environments (Wojton, Avery, Yi, et al. 2021). These recommendations help the operational testing community meet the guidance set by DOT&E, who stated that M&S verification, validation, and accreditation should "articulate a method for strategically varying the factors that affect system performance with respect to the response variables of interest" (DOT&E 2016, p. 3).

Wojton, Avery, Yi, et al. (2021) describe how to collect data from M&S environments but do not describe how to analyze that data beyond direct comparison. And Wojton, Avery, Freeman, et al. (2019) mention metamodels but do not describe them fully. DOT&E issued follow-on guidance stating that "empirical models (a.k.a., emulators or metamodels) should be used to understand M&S outcomes across the operational space and assist in the uncertainty quantification in areas where there are no live data" (DOT&E 2017, p. 1). Since the publication of that guidance, other documents have been published recommending the use of metamodels, such as OPTEVFOR Instruction 5000.1D (Commander, Operational Test and Evaluation Force 2022), which requires that M&S analysis "statistically analyze the V&V runs, identify factor effects, and generate an empirical metamodel of the M&S predictions when possible."

The metamodel is a way to study the M&S environment by directly modeling outputs. For example, if one wants to use M&S to find conditions that optimize a system's performance, or to determine what model parameters best replicate live test data, one can use a metamodel of the M&S environment instead of the M&S environment directly because the metamodel is much smaller, faster, and easier to manipulate. Hence, those working with M&S environments should be able to fit high-quality metamodels.

The present paper is a follow-on to Wojton, Avery, Freeman, et al. (2019) and Wojton Avery, Yi, et al (2021), and it specifies statistically rigorous methods for metamodeling. This paper describes:

- M&S environments and what one should consider when fitting their output with a metamodel;
- The goals of metamodeling, as well as appropriate and inappropriate uses of metamodels; and
- A handful of recommended metamodeling techniques, evaluation criteria, and experimental design approaches.

We split M&S output based on whether the response variable of interest is deterministic (with no random seed[3]) or random (deterministic up to a random seed), and we make the following recommendations:

- For a deterministic, discrete response variable, we recommend nearest neighbor or decision tree interpolators.
- For a deterministic, continuous response variable, we recommend Gaussian process interpolation.
- For a stochastic response case (either continuous or discrete), we recommend a generalized additive model (GAM).

We demonstrate our recommended models and their evaluation using a notional example of a paper plane flight simulator, a digital simulation analyzed as both a fully deterministic simulation and a stochastic simulation.

This paper is a starting point, setting an expectation of what good metamodeling looks like. While we offer recommendations, entire books have been written that present alternative techniques and perspectives. Hastie et al. (2009) in particular offer many statistical procedures one can use for essentially the same purpose. We cannot be as comprehensive in less than 100 pages. Nevertheless, our paper presents a standard for good metamodeling practice.

[3] A random seed is an important part of what allows a computer to emulate randomness. Computers usually generate pseudorandom numbers in lieu of truly random numbers; truly random numbers are impossible for logical systems like computers to generate. Pseudorandom numbers are generated from a complex mathematical process, and if one knows the first number in the process, one can perfectly predict the rest of the numbers. This first number is the random seed. When one doesn't know the first number, the sequence presents an illusion of randomness usually sufficient for statistical purposes. Many M&S environments include pseudorandom number generation as part of the model; some, such as hardware-in-the-loop simulations (including the ECWAF), also feature true randomness because the physics of communication between multiple computers involves real-time calculations sensitive to even small perturbations.

2. Classes of M&S Environments

M&S environments vary widely in nature and implementation, and knowing how an M&S environment works is essential for analyzing its output and properly constructing metamodels for it. In our metamodeling framework, an important determinant of the statistical strategy is *stochastic noise*, or variation in the outcomes of an experiment such that no two runs of the experiment are identical or are identical only when using a common random seed. Some M&S environments are noiseless or deterministic, such that running the M&S environment under the same initial conditions will yield the same output. Only mathematical and computer models can be noiseless in practice, restricting this class of M&S to digital simulation (DSIM), hardware-in-the-loop (HITL), or software-in-the-loop (SITL).

M&S environments with some randomness are called stochastic, regardless of whether that randomness arises naturally in the simulation or from a random seed. One usually analyzes outputs from these environments with statistical fits. What type of statistical fit to use depends on how much variation there is in outputs, how much data we can collect, and what phenomena we wish to detect.

If few M&S outputs can be generated and if outputs vary considerably under common input conditions, we may not be able to detect subtle phenomena, as the variation in the outputs has too pronounced an effect. Operational tests usually have these characteristics; for example, missile tests are expensive on a per-run basis and require significant manpower to execute, and run results can vary considerably despite similar conditions. Statistical methods likely cannot detect subtle phenomena, such as the difference between a linear and a not-quite-linear relationship between miss distance and range to threat, under these conditions. We call such situations *low signal-to-noise ratio* (SNR) *situations*, and they are handled best by relatively simple statistical fits, such as some form of linear statistical model.

If either the variation in outputs is low or the sample sizes are large enough that such subtleties can be detected with confidence, we can estimate statistical fits able to capture such nuances. Such situations are high SNR situations. For example, the Institute for Defense Analyses developed the Virtual Carrier Model (IVCM), a DSIM for studying the relationship between the reliability of systems onboard the aircraft carrier USS *Gerald R. Ford* (CVN-78) and the carrier's sortie generation rate. IVCM can generate thousands of outputs within a reasonable time frame. Thus, it can mostly overcome the variation in the simulation to allow fitting metamodels less prescriptive about the relationship between factors and response variables than linear statistical models; other M&S environments similarly allow a far larger number of runs, presenting opportunities to estimate more nuanced, less prescriptive statistical fits.

Table 2-1 is a nonexhaustive list of common simulation classes encountered in the Department of Defense.

Table 2-1. Some Simulation Environments Used to Support Operational Testing[a]

Simulation Environment	Acronym	Description
Digital Simulation	DSIM	A fully digital representation of a physical system and its intended operational environment. Can be deterministic or stochastic.
Hardware-in-the-Loop	HITL	A simulation that includes actual physical system hardware. Can be deterministic or stochastic but often is stochastic because of the architecture of such simulations.
Software-in-the-Loop	SITL	A simulation incorporating actual physical system software. May be deterministic or stochastic.
Operator-in-the-Loop	OITL	A simulation that includes inputs and decisions from at least one operator. Highly stochastic.
Natural Model	NM	A model that represents a system by another system that exists in the real world; for example, a model that uses one body of water to represent another. Highly stochastic, as natural models essentially are a form of live testing.
Physical Model	PM	A model whose physical characteristics resemble the physical characteristics of the system being modeled; for example, a plastic or wooden replica of an airplane, or a model of a human body filled with sensors to detect likely injury. Likely involved in highly stochastic situations.
Land-Based Test Facility	LBTF	A physical environment, constructed on an open range, that incorporates various aspects of DSIM, HITL, SITL, OITL, or live test assets.
Laboratory/Chamber	LAB	A facility allowing for simulation via DSIM, HITL, SITL, or OITL of various aspects of an operational system in a closed environment.
Threat Representation	TR	Any engineering representation (physical or digital) of a threat system or environment. If physical, should be seen as stochastic.
C4I System Integration Environments and Facilities	C4IEF	A command, control, communications, computers, and intelligence (C4I) environment that operates external to the system under test or the system of systems and that can be used to test system functionality and interoperability. May be deterministic or stochastic.
Reliability Simulation	RSIM	A simulation that provides reliability predictions to represent the system under test live or in captive-carry tests, chamber tests, or DSIM. Could be deterministic or stochastic.

Simulation Environment	Acronym	Description
Federation	N/A	A distributed system of interacting models, simulations, and supporting infrastructure that are based on a common understanding of the objects portrayed in the system of systems. If these connecting models are all DSIMs, the randomness of the federates would propagate to the federation, but if live communication of hardware or human interaction is involved, at least some stochasticity may emerge.
Federate	N/A	An individual system within a federation, such as a simulation, a tool, or an interface to live systems. May be deterministic or stochastic.

[a] See M&S Glossary (Defense Modeling and Simulation Enterprise 2020) and OPTEVFOR Instruction 5000.1D (Commander, Operational Test and Evaluation Force 2022).

The high SNR context interests us more in this paper than the low SNR context. One may discover that a simple linear fit still describes the outputs of an M&S environment well in the high SNR situation, but the data collection and analysis should at least allow the opportunity to discover unanticipated relationships and effects.

3. Goals and Appropriate Uses of Metamodeling

We view metamodeling as an activity that increases the impact of M&S environments in test and evaluation and perhaps throughout the defense community. Table 3-1 summarizes how metamodels could be used in different activities commonly undertaken across the defense enterprise.

Table 3-1. Potential Uses of Metamodeling in Defense Activities

Activity	Metamodel Usage
M&S Verification, Validation, and Accreditation (VV&A)	Can be used to determine whether outcomes observed from live data are consistent with M&S predictions, or to see whether M&S predictions are reasonable at face value given subject matter experts' understanding of the system being modeled. Predictions that are clearly wrong to an expert indicate a bad M&S environment even if no live data were collected.
Test Planning	Suggests system performance in the factor space and what regions may be most interesting to test. Useful for test scoping by facilitating power analysis and the study of VV&A sensitivity.
Predicting System Performance	If a metamodel matches both an M&S environment's outputs and a system's live data, the metamodel predicts the system's performance as a function of the studied factors.
Exercise Planning and Training	Suggests system performance in the factor space and what regions may be most interesting for exercises and training. If appropriate, may be used to simulate live outcomes.
M&S Creation and Execution	Can act as a purely digital surrogate for one M&S that operates as a component of another M&S, thus effectively allowing one model's output to inform another model. This link can overcome obstacles that otherwise would prevent one model from informing another, such as speed, availability, expense, location, etc.
Discrete Event Simulation / Agent-Based Modeling	Can be a part of a discrete event simulation (DES) or an agent-based model (ABM) that effectively allows the outputs of a different M&S environment dedicated to understanding one of the modeled entities to describe the behavior of that entity as a part of the DES or ABM. DESs/ABMs are a class of M&S.
Campaign Analysis	Can be a part of a campaign model by allowing the use of outputs of an M&S environment dedicated to describing the performance of a weapon system, sensor, platform, etc. to inform the behavior of that entity in the campaign model. Campaign models are DSIMs, often DESs. Using the DES STORM[a] as an example, a metamodel fit using ECWAF outputs could be used to determine whether a torpedo fired from a submarine hit its target in the simulation.
Wargaming	Can act as a fast and inexpensive outcome adjudicator informed by an M&S environment that otherwise would be too unwieldy to use in a live setting.

Activity	Metamodel Usage
Tactical Planning	Can leverage otherwise inaccessible or untimely M&S predictions to quantify risks and suggest the likelihood of outcomes for different courses of action when timely information is needed. For example, a metamodel of a weapon system could predict its performance characteristics in wartime, informing warfighter decision-making.
M&S Calibration	A method for leveraging observed data to inform parameter selection for any needed but unknown parameters in an M&S environment to ensure the M&S best mimics observed outcomes.
Parameter Estimation	Transforms M&S into a model that can be used to infer and quantify uncertainty for parameters describing observed data via model calibration.

[a] STORM is the Synthetic Theater Operations Research Model for campaign modeling, involving primarily air and naval assets.

M&S environments aim to realistically reproduce live phenomena in computer environments or to model a particular piece of a live test, such as a threat. Unfortunately, high-fidelity models are often large, slow, expensive, unwieldy, nonportable, and intractable, making them unsuitable for direct use in some activities. Also, usually only the M&S owners and maintainers can access an M&S environment and run it. Metamodels extend the influence of the high-fidelity models to other contexts by statistically summarizing their outputs and substituting for the high-fidelity model when it cannot be practically deployed.

Model validation seeks to determine the extent to which live data and simulation outputs agree, and—as noted by DOT&E (2017)—metamodeling can be part of the validation process. Because metamodels predict the outcome of the simulation, one can compare live data to metamodel fits to check for agreement. Building a metamodel is an intermediate step to M&S validation, and it is a component of M&S verification.

M&S results need to be distributed in a useful and digestible format. Having a “pocket” model that quickly works on a basic laptop can be useful in many instances. The pocket model can aid in planning live testing by identifying interesting regions of low operational performance or regions where performance is sensitive to other factors. If the model is trustworthy, it can be used in wargaming to aid decision-making or in the field to inform tactical decisions. Metamodeling does not, on its own, prove an M&S environment makes accurate predictions, but it can help reveal an unsuitable M&S environment.

Importantly, a metamodel can be used to describe M&S output across the entire operational domain for which the M&S will be accredited (DOT&E 2017) and to describe how much variation to expect in the M&S outputs. Such fitting is useful because it is never the case that live data are available across the full domain of operational conditions to validate an M&S environment. Metamodels summarize M&S environment outputs over the space and thus allow subject matter experts to determine whether the M&S environment is performing nominally well.

A metamodel's simpler representation of a full M&S allows activities that otherwise may not be feasible. For example, a metamodel can be used to select conditions under which optimal performance for the phenomenon under study can be achieved. The metamodel can allow *model calibration*, which is selecting parameter values that lead to M&S output that best mimics the real-world phenomenon being modeled (Smith 2014). The model calibration problem can be reformulated into a statistical parameter estimation problem, turning the M&S environment (via the metamodel) into a complex statistical model and thus allowing us to estimate the real-life value of that parameter from a data set; granted, one takes the leap of faith that the M&S environment (and its companion metamodel) reasonably represent reality. These calibration practices also come with statistical uncertainty procedures.

Metamodels do not eliminate the need for live data, and they do not render the original M&S obsolete either. Metamodels are statistical summaries of M&S outputs. They do not incorporate the physics or logic of the phenomena being modeled; at best, they imitate the outputs observed from an M&S environment with such understanding. Metamodels do not generate "runs for the record" but describe what was seen in actual runs for the record from the M&S environment. Situations where the original M&S environment is still needed include generating new outputs in situations not used in metamodel fitting or evaluating the effects of changes to the modeled system (e.g., missile guidance software).

4. Recommended Methods for Metamodeling Deterministic Simulations

Deterministic M&S environments should be analyzed differently than stochastic simulations. Output generated by deterministic M&S environments has no random noise component. Hence, there is no need to use methods that extract signal from noise, the typical problem statistics addresses. Instead, one should employ interpolation (i.e., connecting observations), which presumes that the values of the response variable under new and unobserved conditions either are identical to or very near the values observed under conditions most resembling those of the unobserved response; thus, interpolation predicts a value for a response that is between the values of its closest neighbors in the M&S output data set. This practice is distinct from statistical regression techniques, which presume there is random variation in the outputs that needs to be smoothed away. If the interpolator is given the input conditions under which an M&S output was observed, the interpolator will return that output exactly;[4] in contrast, a regression model almost never parrots original data.

What method to use for interpolation depends mostly on whether the response variable is discrete or continuous. If it is discrete (such as the classification of a threat or the number of threat missiles detected), a decision tree or nearest neighbor model is most appropriate. If it is continuous (such as the terminal range of a ballistic missile or radar signal emulator), we recommend Gaussian process (GP) interpolation.

GP interpolation requires the analyst to make some decisions after collecting data—in particular, choosing a family of covariance kernels to specify the shape of the interpolating function. Then, the analyst has to determine the values of any remaining needed parameters for achieving the best possible interpolation of the outputs. But once a GP is fitted, confidence bounds can be computed to describe the uncertainty associated with the GP's predictions of a response variable's values. These confidence bounds are a GP's primary attractor.

[4] Some interpolators do not return original outputs exactly but only because the interpolator may be easier for a computer to fit if a small error is allowed, not because of a statistical assumption that the value observed is perturbed by random noise.

A. Metamodels for Discrete, Deterministic Output

Discrete data are data that take values from a list. For discrete and nonrandom responses, we need to classify our data and find rules to perform that classification. Classification methods can be used here, such as nearest neighbor classifiers or decision trees. Hastie et al. (2009) discuss these interpolators in more detail.

1. Nearest Neighbor

A nearest neighbor interpolator predicts an output identical to the nearest observed output, reducing the prediction problem to figuring out which of the observed test points is nearest to the inputs. The advantage of the nearest neighbor interpolator is that it is simple to compute, simple to understand, and yields low-bias predictions.

The method is nonparametric; there is no parametric statistical theory explaining why nearby points are what we should use to make predictions. We simply believe that the response surface is largely continuous and that nearby points have similar values, so we should make predictions that agree with what is nearby.

2. Decision Tree

A decision tree is a sequence of nodes in a graph that asks a series of true or false questions; depending on the answers, one follows different branches of the tree. At the end of the tree is the predicted value of the response variable based on the decisions made. Like the nearest neighbor algorithm, decision trees perfectly predict the outcomes observed in the M&S environment's outputs, but they do so without having to reference the entire data set every time a prediction is requested.

Furthermore, a decision tree is more understandable than a nearest neighbor interpolation, so it is an effective way to summarize M&S output. Figure 4-1 shows decision tree predictions, and Figure 4-2 presents an example decision tree.

Figure 4-1 illustrates nearest neighbor and decision tree interpolations for a fictitious M&S environment predicting whether a large gun with two models (the Mk 1 and Mk 2) firing a shell at some initial velocity and some initial angle of elevation (shifted and scaled to be between 0 and 1) will hit its target. Angle of elevation and velocity are continuous factors while the gun model is a categorical factor. Colored regions denote the outcomes predicted by these two models for the factor combinations. Both fits make the same predictions for the M&S environment's observed outputs (seen at the marked points in the plots, with color corresponding to the observed response), but they make different predictions in the intermediate space. The decision tree generates rules and boundaries for making its predictions while the nearest neighbor interpolator bases its predictions on distance to the nearest point in the observed data set; hence, the irregularity of the latter's predictions stems from the irregularity of the observed data set.

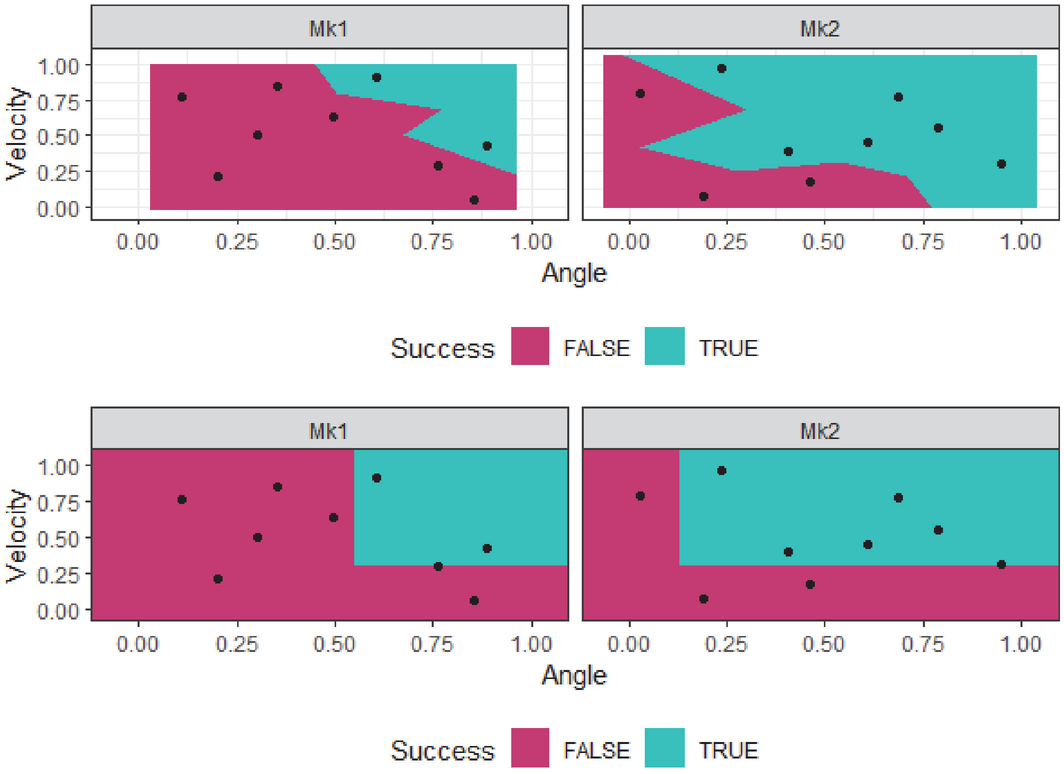

Figure 4-1. Visualizations of fits by a nearest neighbor (top) and a decision tree (bottom) for a deterministic M&S predicting whether a fictitious gun of two types (Mk 1 and Mk 2) will hit its target depending on initial velocity and angle of elevation. Actual M&S observations are marked with points, and the color of the region a point lies within is the output observed. Notice that despite the differing shapes of the regions in the top and bottom plots, the colored region containing each observed point does not change.

Figure 4-2 presents the decision tree used for making predictions in Figure 4-1. To read the tree, start from the top node, answer yes or no to the question proposed, and follow the resulting path ("yes" to the left, "no" to the right) to either the next node or the final prediction. The diagram includes some supplemental information, coloring based on the most common outcome at that level of the tree, reporting the proportion of successes at that node's level in the tree and what proportion of the data set that node covers. This decision-making sequence does not require tabulating the entire M&S environment's outputs, and it is more suggestive than the nearest neighbor interpolator about why predictions were made.

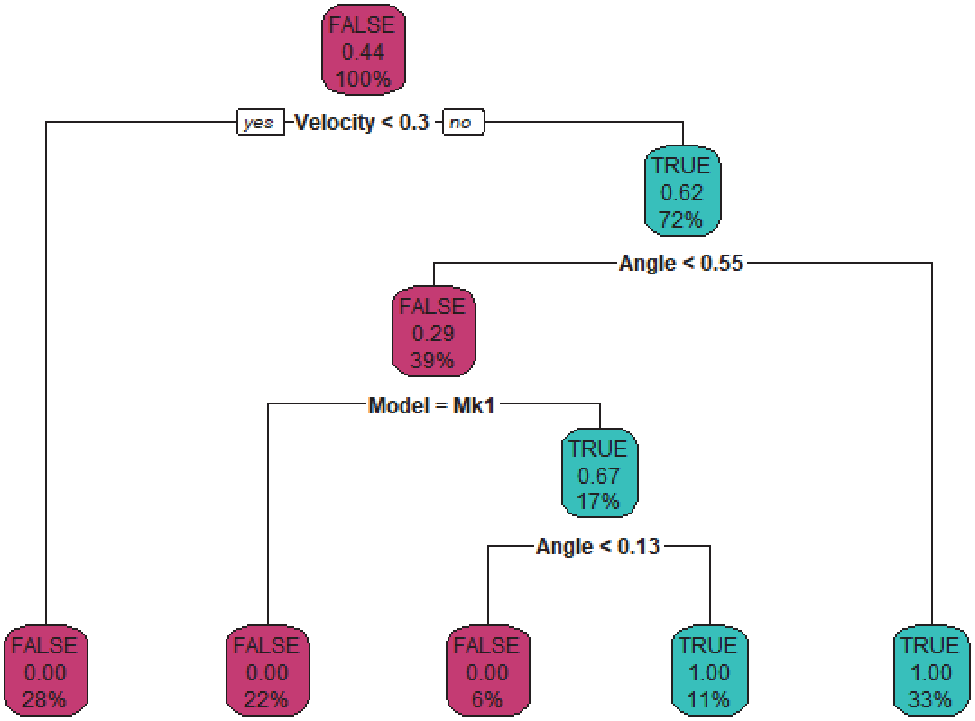

Figure 4-2. Visualization of rules of a fitted decision tree for the data given in Figure 4-1. The label at the top of a node and the node's color indicate the most common outcome at that node level, and the number below the label indicates what proportion of outcomes at that node level are true. The number inside the node at the bottom is the proportion of the data set covered by that node. Below the node is a statement that is either true or false. To make a prediction, answer the statement; if "yes," follow the left side of the tree; otherwise, follow the right side.

B. Metamodels for Continuous, Deterministic Output

We recommend GP interpolation, also known as kriging, for interpolating (or connecting) continuous data. GP interpolation includes a framework for quantifying the uncertainty in a response variable, along with all the features one would find in any other interpolation method, such as splines and piecewise interpolators (which we do not discuss). However, GP interpolation can be slower than other interpolation techniques. Still, GP interpolation may be the most common form of metamodel in all disciplines.

Our goal with GP interpolation remains predicting the output of the M&S environment at unobserved factor combinations using a data set of observed M&S outputs. However, because GPs are a type of probability model, we first need to discuss the properties of that probability model.

1. What Is a Gaussian Process?

A GP model is a probability model describing the distribution of a random *function*. It uses a GP to describe the values a partially observed function may take when assuming that function was generated by the random process.[5] After we estimate a GP given the M&S output, we can infer the likely values of the random function at unobserved test points, and we can put uncertainty bounds on those predictions. Since GPs often interpolate points with two or more factors, the interpolation is sometimes called a *response surface*; when one plots a two-factor interpolation, it resembles a surface someone could touch.

2. Essential Features of a Gaussian Process

GPs are determined fully by two parameters: the *mean function* $\mu(t)$ and the *covariance kernel* (a two-parameter function) $K(s,t)$.[6] GP interpolation requires picking these functions, and the choice strongly affects both the shape of the resulting metamodel and its uncertainty calculations.

In practice, the only interesting choice concerns the covariance kernel K; the mean function is usually chosen to be $\mu(t) = 0$.

The covariance kernel describes the relationship between any two points of the response surface being predicted. A kernel function needs to take two points as input and return the covariance between those two points as output. Scaling the kernel function scales the width of the prediction bands. The kernel function influences how strongly a nearby observed value influences predictions made by a resulting metamodel and its smoothness. Hence, picking the kernel function well matters greatly to the quality of the resulting metamodel.

There are many possible covariance kernels. Practitioners pick a *family* of functions that describe the covariance kernel of the metamodel, so the difference between two members of a common family depends on only a handful of parameters. Usually, most

[5] Astute readers may be wondering why we are using a probability model involving a random function to fit data from a nonrandom, fully deterministic simulation. We are using GPs as part of a Bayesian procedure and thus are adopting the Bayesian perspective on randomness and uncertainty. In that perspective, since we do not know what the M&S environment would do at unobserved conditions, we call it random in those conditions. Consult texts dedicated to Bayesian inference, such as Bernardo and Smith (1994), to learn more.

[6] The covariance between two random variables is equal to the product of their standard deviations and their correlation; hence, the covariance describes how two random variables are correlated. The covariance kernel gives correlation information about the values of the GP at any two points.

covariance kernels used are of the form $K(r)$, with $r = \|s - t\|$ being the Euclidean distance between the two points s and t.[7]

a. Gaussian Kernel

For deterministic simulations, we highlight two common families of covariance kernel functions. The most popular kernel is the Gaussian kernel,

$$K(r; \tau, \lambda) = \tau^2 \exp(-r^2/\lambda),$$

where τ is a scale parameter controlling overall variability in the curve, and λ is a positive number that controls correlation between two points. For a large λ, two points nearby each other would have similar values, while a small λ means less dependence between two nearby points. Thus, λ helps control how much wiggle there is in the resulting surface. One can think of wiggle as how much the resulting function or surface changes direction. A small λ allows a lot of wiggle while a large λ encourages smoothness. The fitted GP generated using a Gaussian kernel tends to be as smooth as possible. Figure 4-3 illustrates how the Gaussian kernel parameters relate to the shape of the potential functions being fitted.

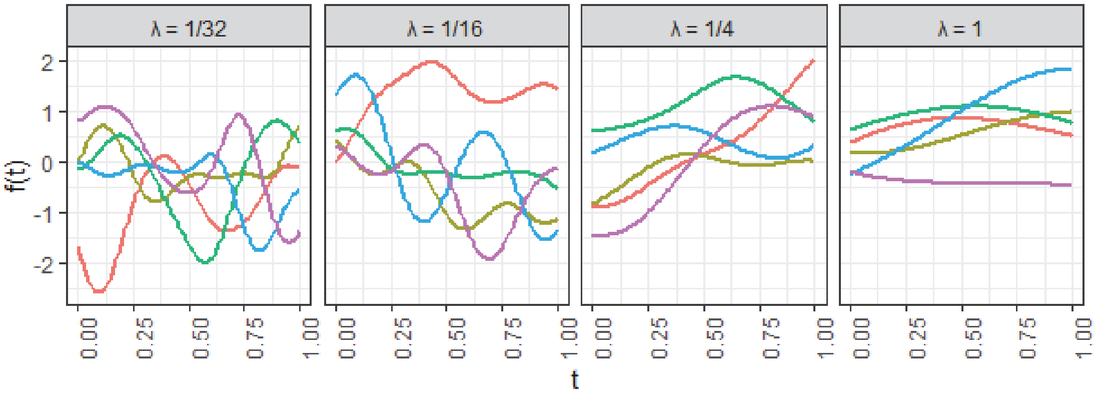

Figure 4-3. Random realizations of a GP with varying correlation parameter λ. A smaller λ implies weaker correlation over larger distances, and an increasing λ strengthens that correlation, influencing the resulting wiggle in the curve.

[7] Kernels with this property are said to be stationary. Situations exist where stationary kernels are inappropriate—specifically, situations where there is more variation in one region of the factor space than in another. We do not discuss such issues in this introductory paper.

b. Matérn Kernel

Some practitioners believe the Gaussian kernel may be too smooth to be believable. Kernels from the Matérn family have the form

$$K(r;\tau,\lambda,\nu) = \tau^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\left(r\sqrt{\frac{2\nu}{\lambda}}\right)^{\nu} Y_\nu\left(r\sqrt{\frac{2\nu}{\lambda}}\right),$$

where Γ is the gamma function, Y_ν is the Bessel function of the second kind of order ν, and ν controls smoothness.[8] The Matérn family of kernels resembles the Gaussian family described above by setting ν very big. Low ν yields metamodels that are jittery, while large ν yields smooth metamodels. Figure 4-4 illustrates the relationship between the parameters and the resulting function shape.

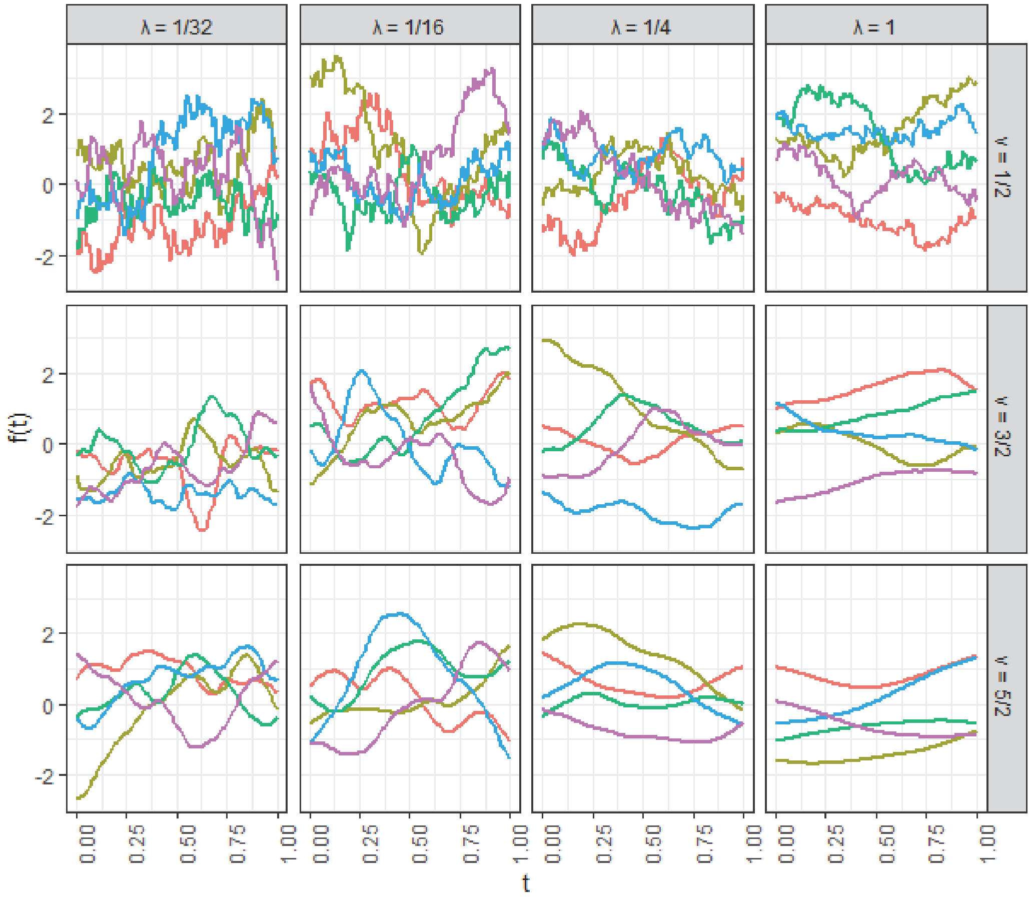

Figure 4-4. Random realizations of a GP using the Matérn kernel with different roughness (ν) and correlation (λ) parameters. The Gaussian kernel case (see Figure 4-3) can be thought of as the $\nu = \infty$ case, hence why the realizations with the Matérn kernel resemble those of the Gaussian kernel for larger ν.

[8] A GP realization with a Matérn kernel has $\lceil \nu \rceil - 1$ continuous derivatives. For example, setting $\nu = \frac{5}{2}$ would result in GP realizations that have continuous position, velocity, and acceleration.

c. Gaussian Processes That Do Not Interpolate M&S Outputs: Nugget Effects

GP models can be augmented for low-noise M&S outputs. The augmentation involves modifying the covariance kernel function to also incorporate a residual noise, σ^2. The augmentation also prevents the GP from interpolating. This residual noise is known as a *nugget effect*.

Allowing a nugget effect is appropriate when a GP is fitting outputs from a stochastic M&S environment; the effect basically allows for "noise" in the data. However, practitioners sometimes allow a nugget effect even for deterministic simulations, when ideally an interpolator would connect all observed outputs from the M&S environment. Forcing the metamodel to pass through each observation might be too difficult numerically, taking too much computer time. Loosening that requirement slightly might make fitting the metamodel easier, at the expense of a tiny bit of error in predictions at observed M&S outputs. Figure 4-5 illustrates what a process with a nugget effect looks like. Unlike the processes seen in Figures 4-3 and 4-4, the process in Figure 4-5 is not a smooth line; rather, it has scattered points yet with clear overall trends.

When two input points are the same, a nugget effect may be added to a kernel function by adding a small, positive number. Mathematically, such a kernel is

$$K(r) + \sigma^2\delta(r),$$

where $\delta(r) = 1$ when $r = 0$ and otherwise is zero.

Figure 4-5 presents a realization of a random GP with a substantial nugget effect. This illustration better resembles using GPs to analyze output with noise than it does allowing for interpolation with small error as described earlier, but the size of the nugget effect illustrates how it affects the GP. Rather than the smooth functions seen in Figures 4-3 and 4-4, the data here appear noisy. Actually, a GP with a nugget effect cannot be a smooth function directly, though it could be seen as a combination of an overall smooth function and a perturbation of that smooth function at observed sites via random noise following a normal distribution.

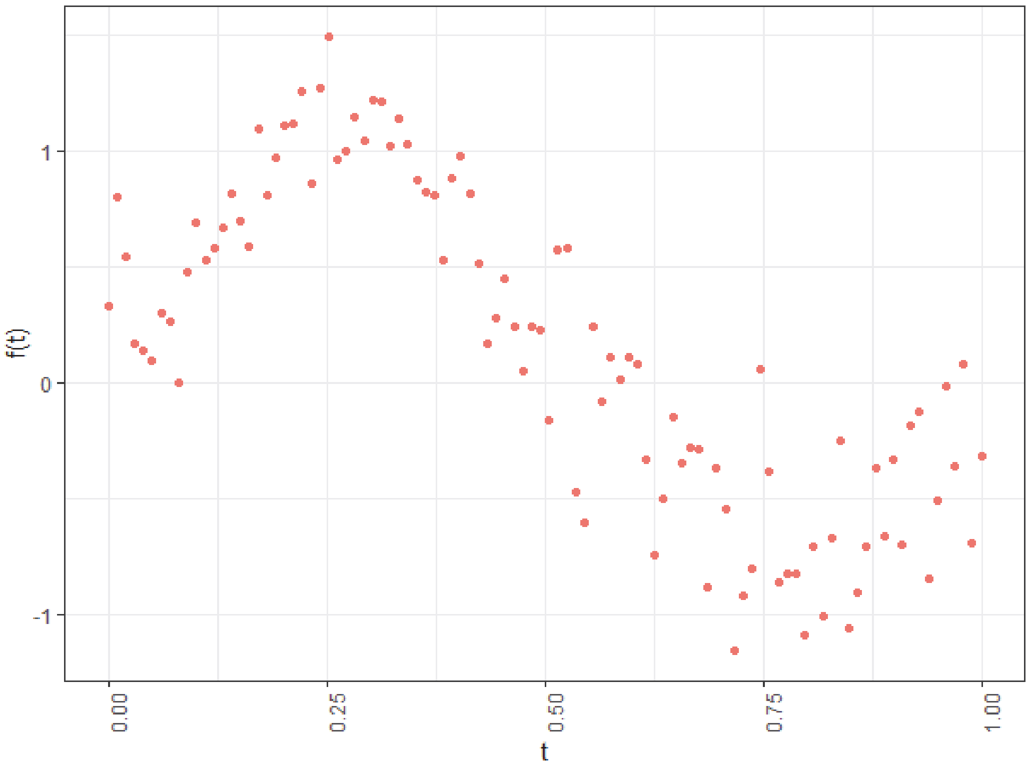

Figure 4-5. Random Realization of a Gaussian Process Using the Gaussian Kernel and with a Nugget Effect

3. Fitting a Gaussian Process Interpolator

It is easiest to start by describing the M&S outputs in pairs. Let x_i and x_j be the factor settings at which outputs y_i and y_j are observed.[9] Let Σ_n be the covariance matrix for the metamodel's prediction at the observed factor settings, with the element in row i and column j of the matrix, Σ_n^{ij}, being equal to the value of the covariance kernel K given the distance between the factor settings x_i and x_j, so $\Sigma_n^{ij} = K(\|x_i - x_j\|)$. Let Y_n be the vector consisting of outputs $y_1, \ldots, y_n$.

Unless a nugget effect is added to the covariance kernel, the predicted value of the GP interpolator at any observed factor setting combination will be exactly the value of the response variable at that test point, with a variance of zero. However, our goal is to make predictions at unobserved factor settings. Suppose $\hat{x}_1, \ldots, \hat{x}_m$ are each of the combinations of factor settings for which we would like to predict the M&S output. Let $\tilde{\Sigma}_m$ be like Σ_n described above but using $\hat{x}_1, \ldots, \hat{x}_m$ instead of $x_1, \ldots, x_n$. Let C_{mn} be a matrix with m rows and n columns such that the value of the matrix in row i and column j is $C_{mn}^{ij} = K(\|x_i - \hat{x}_j\|)$.

[9] x_i and x_j are vectors with length equal to the number of factors, and the factor settings are the vectors' entries.

If $\widehat{Y}_m$ is the vector of the new, unobserved M&S outputs, the conditional distribution of $\widehat{Y}_m$ given the observed outputs will be a multivariate normal distribution. The vector of means of this normal distribution will be $\hat{\mu}_m = C_{mn}\Sigma_n^{-1}Y_n$. *This vector includes the set of predictions for the unobserved M&S output.* The covariance matrix of the distribution is $\widehat{\Sigma}_m = \tilde{\Sigma}_m - C_{mn}\Sigma_n^{-1}C_{mn}^{\top}$. This matrix is needed to form confidence bands around the predictions.

Many software packages perform these calculations, including JMP, Python (in **sklearn**), R (in various packages; we use **GPfit** in our examples, discussed in Section 8), and other commercial statistical software packages.

After one picks a family of covariance kernel (see Sections 4.B.2.a–4.B.2.c, along with Section 6 for more general model selection strategies), fitting a GP interpolator to an output from an M&S environment depends on between two and four parameters. Below, we describe statistical strategies to pick the values of the parameters.

a. Frequentist Approach to Parameter Selection

As mentioned earlier, GPs treat each M&S observation as random and drawn from a normal distribution. The resulting probability model implies we can compute a likelihood function. For a GP model, the likelihood is

$$L_n = (2\pi\tau^2)^{-\frac{n}{2}}|\Sigma_n|^{-\frac{1}{2}}\exp\left(-\frac{1}{2\tau^2}Y_n^{\top}\Sigma_n^{-1}Y_n\right),$$

where Y_n is the observed M&S outputs at n observed test points and Σ_n is a $n \times n$ matrix that is the covariance matrix evaluated at each of the test points and that depends on unknown parameters, as described in Section 4.B.3.

We can estimate the parameters by choosing parameter values that maximize the likelihood function. Once parameters have been estimated, we can construct interval estimates for the output by directly inserting the parameter estimates into formulas involving the covariance kernel.

b. Bayesian Approach to Parameter Selection

In the Bayesian approach, we pair our likelihood function with a prior distribution for the parameters. In principle, a prior incorporates beliefs about what the parameters could be *before* observing data, but we also can see the prior as encapsulating what we *want* to encourage in a resulting metamodel. For example, if we use the Matérn covariance kernel, smoother functions with few wiggles are usually preferred, suggesting that small λ should be avoided unless strong evidence in the data suggests otherwise. Similarly, small ν may not be desirable either, as they generate metamodels with few derivatives and are thus rough.

The parameters of interest generally are positive numbers, so a reasonable prior should generate positive random variables. We may also assume under the prior that these parameters are independent. The prior distribution for the parameters and the likelihood function L_n can then be combined to obtain the posterior distribution.

4. Uncertainty Quantification

As described in Section 4.B.3, GP interpolation, at its core, estimates the conditional distribution of a random function given the observations made from an M&S environment. This distribution is normal, with mean and variance given by $\hat{\mu}_m$ and $\hat{\Sigma}_m$. Further, the distribution for the predicted value of the response variable at an unobserved test point is normal, with mean $\hat{\mu}_m$ and variance equal to the diagonal entry of the matrix $\hat{\Sigma}_m$. One can obtain an uncertainty interval by finding the appropriate quantiles of the corresponding normal distribution at each factor setting combination involved in the prediction. For example, a 95 percent interval will be about two standard deviations away from the mean prediction for each factor setting combination. Figure 4-6 illustrates how adding confidence bands to a GP interpolation describes the uncertainty associated with its predictions.

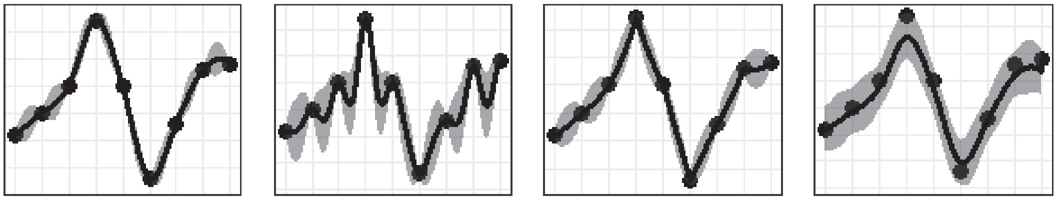

Figure 4-6. GP interpolation of a common data set with varying covariance kernels. From left to right: Gaussian kernel with $\lambda = 1/100$; Gaussian kernel with $\lambda = 1/400$; Matérn kernel with $\lambda = 1/25$ and $\nu = 1/2$; and Gaussian kernel with $\lambda = 1/100$ and a nugget effect with $\sigma = 1/4$. For all plots, the scaling parameter is $\tau = 1/2$, and 80 percent prediction intervals have been drawn. The horizontal and vertical black lines are the x and y axes, respectively.

But pulling directly from the normal distribution this way overlooks the procedures used to determine the covariance kernel's unknown parameters and does not yield a full accounting of uncertainty. If we use a Bayesian procedure to describe those parameters, we could use Monte Carlo procedures to obtain a confidence band.

First, we would sample from the posterior distribution of the covariance kernel parameters. Then, we would randomly generate normal realizations from the posterior distribution of the response variable at the unobserved test points using the randomly sampled covariance kernel parameters. Doing this many times to generate a distribution of the response variables will also account for the selection of the covariance kernel parameters.

5. Potential Problems with Using Gaussian Process Interpolators

a. Bounded Response

GP interpolation assumes no bounds on the potential values of the response, which can be problematic for some metamodels. For example, if the output of an M&S environment is predicted probabilities, then a metamodel should return numbers between 0 and 1.

The easiest solution is to apply a transformation to the observed output so that the transformed value is bounded.

- For probabilities, we recommend that practitioners use the logit transform, $\log\left(\frac{p}{1-p}\right)$.
- For positive data, such as miss distances or delay times, we recommend that practitioners use a log transformation.

Then, fit a GP to the transformed values. The link functions for generalized linear models in Table 5-3 are also possible transformations. To obtain predictions in terms of the original units, apply the inverse transformation to predicted values, paying careful attention when interpreting any related statistics to whether those statistics refer to the original or transformed scale.

b. Categorical Factors

We assumed factors in the above discussion are continuous, where distance between factor settings makes sense. This is not the case for categorical factors. When categorical factors are in the models, there is not an agreed measure of distance between test points.

One approach is to generate a separate GP for each combination of the possible values of the categorical values. The results from one combination are completely uninformative for an even slightly different combination, which may or may not be desired. This approach can be computationally burdensome. A lot of data will be needed if one wants to be able to put in many combinations of factors.

It is more computationally tractable to rule out factor combinations using subject matter input. Other techniques could be used to handle categorical factors; see Kang and Deng (2020) as an example of what could be done.

5. Recommended Methods for Metamodeling Stochastic Simulations

Stochastic simulations, unlike deterministic ones, yield new outcomes each time they are run. Metamodels for such simulations should describe aspects of the M&S output's probability distribution. Predicting a simulation's output with such a model usually amounts to estimating what the mean behavior will be over the operational space, though some methods focus on other aspects of the output's distribution.

The recommended tool in the stochastic case is a generalized additive model (GAM). Additive statistical models relate the response variable to the factors by adding the effects of these factors to the response variable's average value. Unlike with fully linear statistical models, the relationship between an individual factor and the response variable need not be linear, in the sense that changing the value of the factor results in a common change to the response that depends only on how much the factor itself was changed. Instead, additive statistical models make loose assumptions about the relationship between the factors and the response variable's average value, and they use a technique called *smoothing* to discover how the relationship between a factor and the response variable changes depending on the factor's value. Smoothing allows analysts to make weaker assumptions in metamodeling about the relationship between the factors and the M&S output. Additive models become GAMs when the response variable is assumed to follow some non-normal distribution (such as the binomial distribution for binary outcomes), which often means a link function needs to relate the mean of the response variable to the additive model.

Estimating an additive statistical model still requires the analyst to wisely make a number of decisions. The analyst still needs to state the functional form of the relationship between the factors and the response, including which factors should have a linear relationship, which may have an arbitrary smooth relationship, and which should interact and in what way. Fitting involves selecting parameters controlling how to smooth the data to find the relationships between factors and responses and what characteristics the smooth relationships should have—for example, whether they are cyclical (a reasonable assumption for a factor representing the initial heading of a ship, for example) or nondecreasing (such as the relationship between a vehicle's reliability and the distance its platoon travels in a campaign). The result is statistical modeling useful for prediction and inference (including confidence bounds), with more flexibility and generality than traditional linear modeling but with much of linear modeling's interpretability.

A. An Introduction to Additive Models

One perspective on metamodeling M&S system output is finding an appropriate smooth for the response surface. A smooth is a mostly arbitrary function that best fits the M&S output over the operational space.

Smooths can describe the relationship between multiple factors and a single response, but the more factors to be analyzed, the more difficult it is to conduct unstructured smoothing. While a smooth over multiple variables may work well, it can quickly become statistically and numerically untenable.[10] To address this problem, additive models require that the response depend on the factors in an additive manner. For example, if a response variable depends on three factors, the corresponding additive model we could fit would be

$$y = \alpha + f_1(x_1) + f_2(x_2) + f_3(x_3) + \epsilon.$$

This model says that the response variable y depends on three factors[11] via the sum of those three factors fed through unknown functions f_1, f_2, and f_3 and an intercept term α, plus the noise term ϵ. The modeling objective is to estimate the functions f_1, f_2, and f_3, the intercept, and the standard error of the noise term. The functions are mostly unrestricted in their shape. If we wanted to allow an arbitrary interaction between two factors, we could do so with the following model:

$$y = \alpha + f_1(x_1) + f_2(x_2) + f_3(x_3) + f_{23}(x_2, x_3) + \epsilon.$$

If we knew that y depended on x_1 linearly, we could instead fit the model

$$y = \alpha + \beta_1 x_1 + f_2(x_2) + f_3(x_3) + f_{23}(x_2, x_3) + \epsilon.$$

In this case we would need to estimate β_1, a slope term, rather than an entire function.

An additive model is a good compromise between a linear model and full smoothing over the entire factor space. And while the univariate or bivariate functions above may not admit an easy numerical description, they can be plotted. As with linear modeling, the additive model approach allows for statistical testing to determine what factors affect the response variable, what types of relationships exist in the data, and whether there are interactions between response variables.

[10] Potential problems for such smoothing include the curse of dimensionality, large numbers of categorical variables, an inability to incorporate assumptions, and a lack of interpretability.

[11] Note that for ease of presentation in this paper, we briefly change the meaning of the subscript of predictors x compared to its meaning in other contexts. Here, different factors have x with different subscript numbers.

Given these advantages, even though GP regression represents a reasonable metamodeling approach in general and can be used in the stochastic M&S case, we prefer additive metamodels for stochastic M&S prediction.

1. Estimating Additive Smooths

The functions in an additive model need to be estimated via smoothing techniques. Most smoothing techniques work by estimating the coefficients of a set of basis functions, but the smooths could be other functions, even GPs.

For the sake of simplicity, suppose we wish to fit the model

$$y = \alpha + f(x) + \epsilon.$$

Requiring $\int_a^b f\,(x)dx = 0$ ensures the model is uniquely estimable. This means that all functions in the model have an average value of 0 over their domain. As a result, the additive functions are centered around 0. When working with additive models, one must always be aware of identifiability problems, as they are more pernicious than when one works with linear models. Software should handle these issues automatically by imposing appropriate constraints, such as the integrate-to-zero constraint.

Now we discuss function estimation. We can write the function as

$$f(x) \approx \sum_{i=1}^{k} a_i\, b_i(x).$$

That is, we say that the function is approximately the sum of a finite number of known functions, b_i, each multiplied by unknown constants, a_i. The collection of functions $b_1, \dots, b_k$ are referred to as *basis functions*, and picking these functions is called picking the *basis* for estimation. Some bases require the selection of knot points, or points x_j corresponding with each basis function b_j such that $f(x_j) = a_j$. Knot points and knot point spacing are yet another fitting consideration that can affect fit quality for such bases, which is a nuisance; yet, other considerations (flexibility, ease of computation, etc.) may make those bases a good choice regardless.

Figure 5-1 illustrates common bases and some of their constituent basis functions, and Table 5-1 describes the strengths and weaknesses associated with each. Table 5-1 also presents more bases (not all illustrated) one could use.

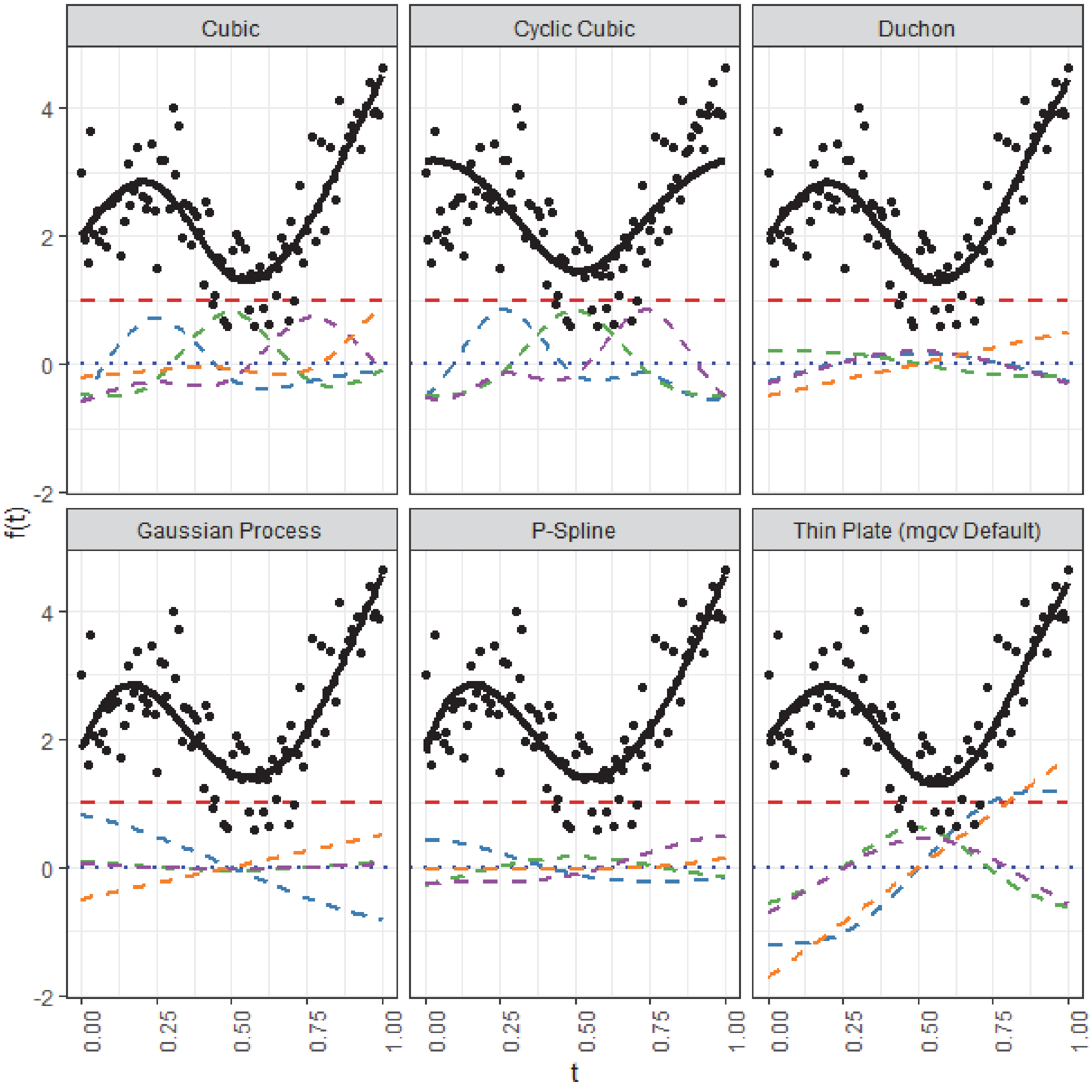

Figure 5-1. Visualization of a function smoothed using different bases. The data fitted, and the estimated smooth, are shown in black. Underneath the black line are plots of the constituent basis functions used to form the smooth, including the intercept (the red line).

Table 5-1. Bases used in smoothing; see Wood (2017) or Ramsay, Hooker, and Graves (2009) for more summaries.

Basis	Characteristics	Recommendations	Additional References
Cubic	Analytically simple to describe and thus common. Requires setting knots, with a resulting function interpolating the set values at the knots and having continuous first and second derivatives, thus looking smooth. Not difficult to compute or set up. Mathematically proven to be the smoothest interpolation.	The need to specify knot locations introduces yet another set of parameters that need to be determined and can affect fit quality, which is not desirable. A common basis nonetheless, and acceptable if no other is present.	Schoenberg (1964)
Cyclic Cubic	Cubic splines where the value and all derivatives at the end of the input domain are all equal, so that the value of the function at the end equals the value at the beginning along with still being a smooth connection.	Periodic phenomena should be modeled with a cyclic function; for example, daily average temperature over a year should be cyclic. This basis is a reasonable choice in such cases.	Schoenberg (1964)
Duchon	A large class of splines that includes thin-plate splines as a special case.	Best used for smoothing multiple variables in dimensions higher than two and where the variables are largely the same units. In particular, for smoothing covariates over a sphere (such as latitude and longitude of satellites orbiting Earth), Duchon splines seem to fair well.	Duchon (1977)
Gaussian Process	The individual functions in the fit are treated as Gaussian processes.	If one prefers Bayesian approaches to inference, this basis helps facilitate such inference.	Matheron (1963)
P-Spline	Based on B-splines, but with a difference penalty applied to the coefficients of the basis elements to control their wiggliness. Retains the property of B-splines of being mostly zero, making the basis extremely easy to set up and computationally fast. Permits adaptive smoothing, where not all regions of the covariate are equally smoothed.	An easily computed basis that is useful in situations where computational power is at a premium, such as Bayesian analysis using Markov chain Monte Carlo methods. Also useful in contexts where different levels of smoothing are needed for different values of the factor being smoothed, as evidenced by a single smoothing factor seeming to fail in some areas; this is known as adaptive smoothing.	Eilers and Marx (1996)

Basis	Characteristics	Recommendations	Additional References
Thin Plate	Handles one or more input variables automatically. Optimal basis in the sense that the approximation minimizes squared error plus a wiggliness penalty, the approximation's squared second derivative. Computationally cheap to evaluate. Does not require knot placement. Constructing the basis functions can be more computationally demanding than constructing the cubic spline basis.	Reasonable default basis, but if lots of basis functions are needed for some reason, a cubic basis may be better. For interaction smooths, recommended if the inputs are of the same scale, such as geographic coordinates; if not, tensor product smooths may be better.	Duchon (1977)
Polynomial	A sequence of polynomials added together (when multiplied by coefficients). Very well understood basis mathematically, and common in statistics. A very simple basis (such as $1, x, x^2, x^3, \ldots$) would have numerical problems, so usually an orthogonal polynomial basis (Chebyshev polynomials, Legendre polynomials, etc.) is used instead. No need for knot placement.	If a basis from scratch were needed and the knot placement problem unappealing, a polynomial basis is fine. In practice though, polynomials are more useful as an analytical tool than in computations.	Abramowitz and Stegun (1972)
Fourier	A sequence of sine and cosine functions with differing periods. Periodic, and does not require knots. Well-known theoretical properties.	This basis is common and a reasonable choice if available. The theory associated with the basis is also appealing, thanks to the relationship with Fourier transformations, so if theoretical interpretation of the basis itself is needed, the Fourier basis may be selected. However, the basis is not as flexible as the cyclic cubic basis for fewer numbers of basis elements.	Ramsay, Hooker, and Graves (2009)
B-Spline	Like the cubic basis, interpolates a series of knots and has continuous first and second derivatives, resulting in a smooth function. Only over a small region is any basis element non-zero, meaning only a few elements from the basis are needed for most computations, making the basis representation computationally efficient.	In extreme circumstances where computational power is at a premium, such as a large-scale smoothing problem with lots of inputs, this basis can be numerically efficient enough to work.	de Boor (1978)

Basis	Characteristics	Recommendations	Additional References
Cyclic P-Spline	A P-spline basis required to be cyclic, or to have equal value and continuous derivatives at the ends of the function's domain.	Used in conditions where a P-spline would be used and cyclicality is required (situations such as time of day, time of year, relative bearing, etc.).	Eilers and Marx (1996)
SCOP-Spline	Shape-constrained P-splines (SCOP-spline) have additional constraints on the shape of the resulting function, such as requiring that the function be an increasing function.	Used in shape-constrained additive models (SCAMs). When one knows more about the shape of the resulting function than that it is monotonic, these splines should be used.	Pya and Wood (2015)
Soap Film	A basis designed for smoothing in an irregular, nonrectangular region, such as over a body of water.	Used for irregular regions when one should also account for the region's irregular shape, which is often a geographic area of mixed type (such as including land and large bodies of water).	Wood (2008)
Tensor Product	This is a procedure for constructing a basis for a multivariate smooth from bases for univariate smooths. Notably, the covariates of the smooth do not need to have the same scale or units (unlike multivariate smoothing via Duchon or thin-plate splines).	This should be the default approach for multivariate smoothing unless the covariates are in the same units (specifically, spatial coordinates).	de Boor (1978) and Wood (2006)

We prefer that k be large, because a large k improves the ability of the basis to approximate the function f. However, a large k raises the possibility of overfitting or unstable estimation. We will require additional constraints to achieve the desired smoothing results.

The idea of fitting the output is achieved by solving

$$\min_{\alpha,f} \sum_{i=1}^{n} (y_i - \alpha - f(x_i))^2.$$

The equation above is the least squares minimization problem: a function best fits the output when the sum of the squared differences between the observed output value y_i and the predicted value $\alpha + f(x_i)$ is made as small as possible. Unfortunately, the space of possible f permitted by the basis may be so large that this minimization is too easy to solve. The resulting fit may superfluously interpolate the output rather than extract a correct pattern. This superfluous interpolation can be solved by modifying the problem by penalizing the f we dislike. Abstractly, we do this by instead solving

$$\min_{\alpha,f} \sum_{i=1}^{n} \left(y_i - \alpha - f(x_i)\right)^2 + \mathrm{Cost}(f).$$

In addition to requiring that the function fit the data, we also ask that it not pick a costly f that may be overfitting the data. In function smoothing, we equate cost with wiggliness. One way to define a wiggly function is to say that the area under the second derivative of f is big. This suggests that we solve

$$\min_{\alpha,f} \sum_{i=1}^{n} (y_i - \alpha - f(x_i))^2 + \lambda \int_a^b (f''(x))^2 \, \mathrm{d}x.$$

Keeping the variation of the second derivative of f small over the domain of x (while also requiring that the second derivative be continuous) encourages differentiable functions that do not wiggle too much; the ideal function by this metric is a straight line, a function with zero wiggle. We can now use a basis with lots of functions and not be too concerned about overfitting.

The parameter λ controls how costly a wiggly fit is. Figure 5-2 illustrates this tradeoff when attempting to smooth fictitious data. Having a large λ may result in underfitting the data and having fewer local phenomena, such as local extrema; the resulting fit looks like a flat trend line (panel for $\lambda = 100$). A small λ makes having a large second derivative cheap, thus allowing the function to wiggle more (panel for $\lambda = 1\text{e}{-}04$). But the danger of a small λ is overfitting.

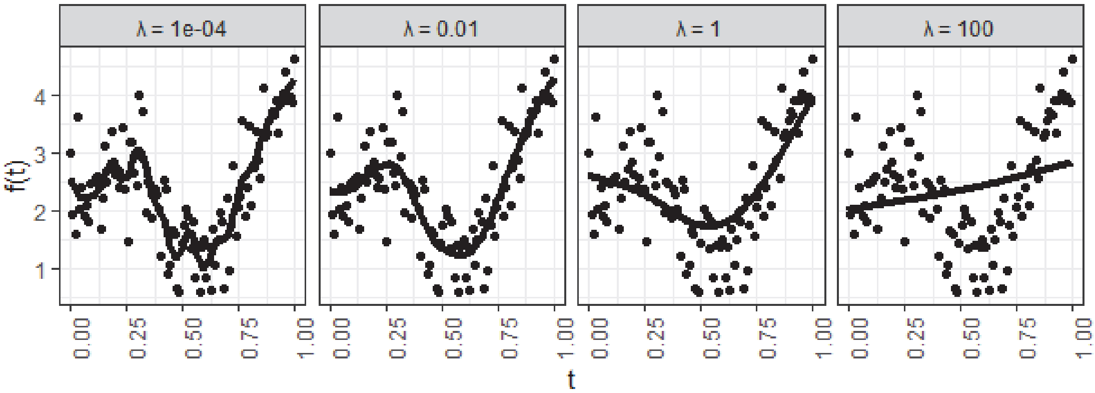

Figure 5-2. How different choices of smoothing parameter affect fitted function. The smooth in the leftmost panel is too wiggly and likely would not predict new observations well because it is too influenced by the data it saw and fails to generalize. The smooth in the rightmost panel makes biased predictions through most of the space because it is too smooth.

In principle, a good λ is one where the resulting fit predicts data well and does not depend too much on any individual data point; if a data point were not used for smoothing, the smooth function estimated with this omission would predict the missing data point well.

This idea motivates use of the generalized cross-validation (GCV) score, which estimates the error of the smoothed function for each data point if that data point were not used for smoothing. Picking a λ that minimizes the GCV results in a smooth that should neither underfit nor overfit the data.

In summary, smoothing involves three choices:

1. The set of basis functions,
2. The number of basis functions to include, and
3. The roughness penalty parameter λ.

We recommend picking a basis by its capabilities and ease of computation rather than fit quality. If one has no such concerns, software defaults likely are fine for many data problems.

The number of basis functions, k, to include is a secondary concern. The usual warning about including too many parameters in a fit does not apply here since overfitting is controlled by the cost function and λ. Hence, picking a large number of basis functions is reasonable if the fits are computationally tractable.

Assuming the operational space is well-specified, the roughness penalty parameter, λ, controls fit quality. The GCV score serves as a good method to pick λ, but there are other options, like the restricted maximum likelihood (REML) criterion. The value of λ can indicate model improvements. For example, if λ is very large and the resulting fitted function looks nearly linear, the model might be improved by replacing the general smooth function with a linear term. Table 5-2 summarizes methods for picking a smoothing parameter.

Table 5-2. Methods for selecting a smoothing parameter; see Wood (2017).

Method	Description	Recommendation	Additional References
REML	Restricted maximum likelihood; maximizes the likelihood of the data, modeled as the likelihood of the data given the parameters multiplied by the prior likelihood of the parameters, then integrates out the parameters, leaving the likelihood of the data given the smoothing parameter.	Preferred method, but not at all simple (if done from scratch) and can be numerically difficult.	Anderssen and Bloomfield (1974) and Wahba (1985)
C_p	Attempts to minimize squared error; requires a known scale parameter.	If one knows the scale parameter, this method works well, but that is rarely the case. The R package **mgcv** default is to use C_p if possible, but usually it uses GCV.	Mallows (1973)

Method	Description	Recommendation	Additional References
GCV	Leave-one-out cross-validation (n-CV) estimates the error made by a model predicting an outcome using all outputs except those associated with the selected observations. Generalized CV (GCV) has a similar interpretation but makes estimated error rotation-invariant. n-CV can also be generalized to leave-several-out CV, which reduces the variance associated with GCV estimates of accuracy at the expense of increasing bias, as described in Hastie, Tibshirani, and Friedman (2009). GCV seems more intuitive than REML. While REML asymptotically undersmooths relative to GCV (Wahba 1985), GCV is more likely to have multiple minima and undersmooth relative to REML (Reiss and Ogden 2009).	GCV is a good choice if the sample size is large, overfitting is worse than underfitting, or REML is infeasible.	Craven and Wahba (1979)
OSER	After obtaining a distribution describing the value of the smoothing parameter, select the smoothing parameter one standard error above the mean, thus erring on the side of oversmoothing (this is called the one-standard-error rule, or OSER). Requires using a procedure such as REML first to obtain such a distribution.	If other procedures seem to be producing models that are not smooth enough, try OSER.	Hastie, Tibshirani, and Friedman (2009)

2. Generalized Additive Models: Extending the Additive Model Framework to Arbitrary Response Types

GAMs combine the ideas of additive models described above (stating that the relationship between a response variable and some factors depends on smooth functions we need to estimate) with generalized linear models, a class of statistical models including least squares linear regression as well as logistic regression, survival analysis statistical models, and others. The theory allows fitting metamodels relating the response variables to the factors even when that response is binary (hit/miss), a discrete count (number of targets found), or non-negative (time to failure).

A GAM will fit a metamodel that says the mean of the response variable depends on the sum of smooth functions of the factors, with each such function needing to be estimated. A link function, determined by the type of response variable being handled (binary, count, survival time, etc.), connects this mean to the additive model. Hence the model estimates average values for the response variables given the input factors, and it does so while saying relatively little about what the relationship between the response and a specific factor is.

An example of such a model is

$$\eta(\mu_i) = \alpha + f_1(x_{1i}) + f_2(x_{2i}),$$

where μ_i is the mean of y_i given the value of the predicting factors x_{1i} and x_{2i} and η is a link function that depends on the statistical nature of the response variable. For example, if the response variable is binary, η is usually (but not required to be) the canonical link function[12] of the binomial distribution, which is the logit function, or $\eta(p) = \log\left(\frac{p}{1-p}\right)$. Table 5-3 lists common probability distributions, use cases for those distributions, and common link functions used for those distributions. After we estimate f_1 and f_2 using smoothing techniques, we can estimate the conditional mean of the response variable given the factors via

$$\hat{\mu}_i = \eta^{-1}\left(\hat{\alpha} + \hat{f}_1(x_{1i}) + \hat{f}_2(x_{2i})\right),$$

with $\hat{\mu}_i$, $\hat{\alpha}$, $\hat{f}_1$, $\hat{f}_2$ all signifying estimates.

Table 5-3. Common response distributions for Generalized Linear Models and GAMs, when to use them, and their properties. There is a star by the canonical link.

Distribution	Range	Common Use Case	Example Response	Link Functions
Normal	Unbounded	Measurement	Signed[a] miss distance	Identity*
Bernoulli	0,1	Success or failure	Hit or miss	Logit,* probit (normal inverse cumulative distribution function (CDF)), complementary log-log (CLL), identity, angular
Poisson	0,1,2, ...	Count	Number of failures in test	Log*
Negative Binomial	0,1,2, ...	Count	Number of warnings until failure	$\log\left(\frac{\mu}{(\mu+k)}\right)$,* identity, square root
Gamma	Positive	Time to event	Time until failure; radial miss distance	Inverse,* log
Inverse Gaussian	Positive	Time to event	Time until failure	Square inverse*

[a] A signed value in this context can either be positive or negative. An example of a signed miss distance is recording that a shell that landed to the left of the target had a negative miss distance, while if it landed to the right it had a positive miss distance. Realistically, we would record another signed missed distance for misses above and below the target.

[12] The canonical link function is a special link function specific to the distribution being modeled; it has slight numerical advantages in estimation. For more details, see McCullagh and Nelder (1989).

Hence, there are a number of moving parts when building GAMs:

- Selecting the distribution of the response.
- Selecting the link function, η.
- Selecting the factors.
- Selecting the linear terms, smooth terms, and interactions.
- Selecting the basis and number of basis functions for smooth terms.
- Fitting the model, with all the above components.

Some of these tasks are addressed well by software defaults, but the user needs to be aware of the issues at hand to use the software well.

Here we offer tips on making these decisions and fitting GAMs.

a. Response Distribution

The distribution of the response is best selected based on subject matter expertise. Below are some examples of data encountered and what distribution could be used for modeling:

- Success/fail data imply a Bernoulli response distribution.
- Count data imply a Poisson or negative binomial response distribution.
- Time-to-failure data imply a gamma response distribution.
- The normal distribution is a good, general choice for continuous data, though one can explore alternatives (gamma, inverse gamma, etc.) to see if a better fit is possible.
- The log normal distribution may be tried for skewed data as a first approach prior to trying another skewed distribution (such as gamma).

These are just a handful of starting points, and software packages have several more options.

b. Model Selection

Similarly, identifying which factors matter and how they should influence the response variable's mean should be driven by a combination of subject matter expertise and statistical expertise. The best-fitting model may not be interpretable. If understandability is desired, a highly predictive model may be unsatisfactory even when its predictive ability is good. One approach to promoting interpretability is to estimate a more easily interpreted but mis-specified model. Mis-specification comes with consequences; White (1982) discusses them in more depth, but a mis-specified model should be interpreted as the best approximation of the correct model, and it can come with less precise estimates of the model coefficients. Another approach is to liberally add terms and complexity to a model but decompose those effects into interpretable and opaque components; for example, decompose the effect of a variable on a response output into the sum of a linear and nonlinear effect. If the less interpretable parts of the model are either statistically insignificant or make a small contribution to overall changes in performance, the noninterpretable parts can be viewed as being essentially nuisance parameters and largely ignored in analysis and discussion, perhaps not even reported (or relegated to an appendix).

Information criteria (such as the Akaike information criterion or the Bayesian information criterion), p-values, and other metrics can assist with model selection, but metrics always provide only a single-number summary of model goodness, and other considerations can justify choosing one model over another regardless of what the metrics say.

c. Basis Selection

For general metamodeling with GAMs, we recommend a thin-plate spline basis for single-factor smooths, coupled with tensor product bases for interaction smooths. Other bases will be useful depending on the type of data being modeled. For example, if a factor has a cyclic effect, then use a cyclic cubic spline to account for the periodicity.

When one is estimating the smooth functions that form the basis of the resulting model, software will do a lot of the work. Choose a reasonable basis for the functions and a reasonable size for the basis, erring on the side of having many basis functions. Going with $10n^{2/9}$ basis functions is a reasonable default (see Kim and Gu 2004). In practice, modify the number of basis functions only if the software default is problematic.

d. Computational Issues with Metamodels

The fitting procedures for GAMs and GP interpolators are computationally complex, and they can take noticeable time to complete. Some properties that influence fitting time include:

- Large data sets,
- Large number of factors, and
- Large number of interactions.

Working with subsets of the data or with simpler metamodels at the beginning of an analysis can be useful to quickly experiment with important metamodel decisions before devoting more computer time to estimation.

3. Statistical Inference for Generalized Additive Models

Once a GAM is fitted, we can perform statistical inference. See Wood (2017) for details. Our recommended software package, the R package **mgcv**, handles these details. Here is a selection of inferences the analyst can make:

- Interval estimates around estimated conditional means and model parameters
- Pointwise interval bands around the smooth functions produced by the fit
- Testing the statistical significance of a function
- Analysis of variance (ANOVA)-type tests that check whether a *collection* of functions affect the conditional mean response

In Section 6, we discuss techniques one can use to judge the quality of a fitted GAM.

4. Extensions of the Generalized Additive Model Framework

The additive model framework used in GAMs can be extended for other purposes and to impose different constraints. Here are the most relevant extensions for metamodeling M&S system output:

- *Shape-constrained additive models* (SCAMs) incorporate smoothing functions that *cannot* take an arbitrary shape and need to satisfy some condition, such as being an increasing function (Pya and Wood 2015).
- *Additive quantile models* estimate conditional percentiles of the response variable (Koenker 2011).
- *Functional additive models* (FAMs) allow the factors or even the M&S outputs to be a function rather than a single number or label. For example, such a model could use as a predictor not only whether a missile hit or missed, or its miss distance, but also the missile's entire flight path, or it even could predict what the typical flight path will look like based on input factors (Müller and Yao 2008).
- *Vector GAMs* can model both location and dispersion, rather than location only. These models allow output variation to depend on the factors rather than be constant over the operational space. They also allow multiple response variables to be modeled simultaneously (Yee 2015).
- *Multiclass additive models* determine the probability that an outcome is one of a number of potential outcome classes (Zhang et al. 2019).

B. Linear Models as Metamodels

Linear models (or generalized linear models) can be used as metamodels of stochastic simulations. We recommend additive models since they have much less stringent requirements about the relationship between the operational space and operational metrics. Additive models include linear models as a subclass; thus, by recommending the use of additive models, we can say we still recommend the use of linear models but in an extended framework allowing the possibility and discovery of nonlinear relationships in addition to linear ones. One may discover that additive models' flexibility contributes little to overall fit and thus one may revert to linear effects only, but one should not do so without mitigating circumstances (usually difficulties in performing experiments, which is why linear models should still predominate in live testing).

It may not be possible to estimate an additive model. In these situations, the classic linear model may be a good fallback, especially in low SNR situations. Usually the reason additive models cannot be estimated is limitations in the observations due to both small sample size and insufficient variation of factors in the sample. D-optimal experimental designs for relatively simple linear models likely would preclude fitting additive models, while SFDs should vary points enough. We discuss data collection briefly in Section 7; Wojton et al. (2021) devote themselves to the topic.

C. Gaussian Process Models Can Be Used in the Additive Model Framework

We noted in Section 4 that adding a nugget effect leads to a GP that no longer interpolates the data but instead passes near it. The GP can then be seen as predicting the mean of the data at a given point, with some extra randomness still being possible. Otherwise, the GP discussion here is the same as in Section 4, including all discussions about picking a covariance kernel.

When one allows for a nugget effect, the GP can be interpreted as a smoother. In fact, GPs fit into the additive model theory, and software such as the R package **mgcv** can fit additive models with the additive functions being GPs.

6. Evaluating the Fit of a Metamodel

In this section, we discuss strategies for evaluating metamodel quality. Sections 4 and 5 present many options for fitting a metamodel to M&S outputs. Analysts need M&S outputs to help them choose which of those options to use when estimating a metamodel. Broadly speaking, a metamodel needs to describe M&S outputs well, and metamodeling choices need to facilitate a good description of the M&S environment. This section describes how to judge metamodel performance and improve it.

A well-calibrated metamodel makes predictions that generally match M&S environment outputs. We observe some M&S outputs and use those outputs for estimating a metamodel. Closely matching outputs already observed—known as *in-sample* outputs—is not good enough; the metamodel needs to describe hypothetical unobserved outputs—known as *out-of-sample* outputs—well too. Techniques such as *cross-validation* (CV) and splitting the outputs into *training*, *screening*, and *evaluation output sets* help quantify metamodel performance in contexts not used for fitting. One should plan the use of such techniques before generating M&S outputs, and DOE should accommodate these techniques. We more precisely define model quality using metrics such as Brier scores, mean-squared error (MSE), the Bayesian information criterion, and others; all of these metrics either describe how much metamodel predictions deviate from observed outcomes or state how likely the outputs would be given the metamodel we fitted. Visualizations such as calibration plots reveal the relationship between predictions and observed outputs.

If our metamodel assessment metrics remain consistent between training, screening, and evaluation sets, we can rely on the metamodel's predictions; otherwise, the metamodel may be *overfitting* M&S environment outputs, meaning that it mostly repeats outputs observed in the training set without learning the larger patterns of the M&S environment it needs to emulate. The metamodel may also simply fail to make precise predictions—a phenomenon known as *underfitting*—though we do the best we can in fitting a metamodel to make precise predictions without overfitting to the training set. If we are satisfied with the precision of metamodel predictions and we observe no evidence of overfitting as we evaluate its performance on observations not used directly for fitting, we may declare the metamodel a sufficient representation of the M&S environment and use it.

This section describes the components of this process in more detail. We start by discussing output splitting, then present metrics that help define model quality and fit. Finally, we discuss visualization techniques.

A. Sample Division for Evaluation

Our ultimate goal is for the statistical model to make good predictions about unobserved M&S environment output. Outputs used to fit a metamodel are called in-sample outputs, while outputs not used for fitting are called out-of-sample outputs. We want the model to predict both in-sample and out-of-sample outputs well. A model can predict in-sample outputs very well, perhaps perfectly, yet poorly predict out-of-sample outputs; a look-up table or nearest neighbor interpolator may have this property. For stochastic simulations, this phenomenon is called overfitting. (There is no such thing as overfitting for deterministic simulations.) Strategies to address overfitting attempt to emulate the in-sample/out-of-sample division and observe the metamodel's ability to predict emulated out-of-sample data.

Out-of-sample observations effectively become in-sample outputs when analysts use them frequently to measure overfitting. One should think of outputs used at any point in the fitting process as being "contaminated" and thus less able to reveal overfitting. The more the so-called out-of-sample outputs are referenced, the more contaminated they become. For this reason, we recommend that analysts keep some outputs separated from the metamodel fitting process until a final candidate metamodel has been determined, at which point the held-out sample can give a final assessment of the metamodel's performance.

1. Train-Screen-Evaluate Split for Metamodel Evaluation

When one can do so, assessing metamodel fit using out-of-sample outputs is better than using in-sample outputs. Out-of-sample outputs might be hard to come by, but one option is to create out-of-sample outputs by separating them out at the beginning of the metamodeling process. This held-out output is called the *evaluation output set*, while the M&S output still used for model fitting is called the *training output set*. This strategy of splitting the M&S outputs is known as the *train-evaluate split.*

Ideally, the evaluation set is used only for a final characterization of metamodel performance and is never used for metamodel selection. This is because model selection implicitly suggests refitting models until one with good out-of-sample performance is found. If an evaluation set is repeatedly queried, it ceases to be an out-of-sample output since one is effectively optimizing metrics computed on it with different statistical fits, even if the process is ad hoc. Hence, the test set is often viewed as being off limits until one has decided what metamodel to use and how it should be fitted, and then has estimated it. The evaluation set is only ever referenced to obtain final estimates of prediction error.

If the evaluation set is off limits until the very end of metamodel fitting, how can one get a momentary sense of out-of-sample performance for a candidate metamodel? Some propose splitting the training outputs yet again so there is a training output set and a *screening output set.* The screening output set emulates the evaluation set as it is not used in model fits directly but is used for intermediate metamodel evaluations. The difference between the screening set and the evaluation set is that analysts can repeatedly query the screening set to estimate out-of-sample performance. We call splitting the available M&S outputs this way the *train-screen-evaluate split.*[13]

The more the screening set is used, the more it behaves like the training output and effectively becomes a part of the fit, degrading its ability to emulate the evaluation set and out-of-sample outputs. Hence, while the screening set can be repeatedly queried, some restraint is wise. We recommend querying the screening set when one has obtained a metamodel that could be the final candidate metamodel but one has not committed to it, or when one has a small list of candidate metamodels and needs to downselect to a single metamodel.

There need to be enough M&S outputs to obtain a good fit and enough data to meaningfully assess the model's performance. Hastie, Tibshirani, and Friedman (2009) suggest that 50 percent of all outputs used be training outputs, 25 percent be screening outputs, and 25 percent be evaluation outputs. However, their rule of thumb likely emerged in a context where data are more numerous and less expensive than in the Department of Defense test and evaluation M&S context. Study planners can justify smaller sample sizes for the screening and evaluation sets via statistical sample sizing methods, such as choosing the sample size for the screening and evaluation sets such that the standard errors of the preferred evaluation metrics (MSE, accuracy, etc.) are acceptably small in both sets. In any case, we recommend that the screening and evaluation sets have the same sample sizes, since the screening set operates like the evaluation set but can be referenced multiple times.

We recommend making decisions regarding metamodel evaluation strategies prior to collecting data, so the data may be generated in a manner that supports the strategy. The sample size and experimental design for the training, screening, and evaluation sets should be generated and executed separately. By accounting for output splitting in study planning, we ensure that the output sets are well-balanced and representative of the factor space in which predictions will be made and statistical inference is needed. We also ensure that we retain the design properties we desire, such as the spread of the points throughout the factor space and the equal representation of each region of the factor space. Separating the designs helps ensure that the training, screening, and evaluation sets remain separated and thus serve as good representations of unseen data. Failing to do so, perhaps by generating one DOE that an analyst then splits randomly after collecting M&S observations, may

[13] To our knowledge, ours is the first use of this terminology. In statistics and machine learning literature, the screening set is called the validation data set, and the evaluation set is called the test data set; Hastie et al. (2009) use this terminology. We introduced different terminology here to avoid confusion with existing terminology in the Department of Defense test and evaluation community, but we note the difference when interfacing with other communities.

result in imbalance in these sets of observations and also result in the training set not having good spread throughout the factor space.

We realize our recommendation adds yet another consideration to the process of forming a good DOE for a study. Test and evaluation practitioners may not like having yet another concern. Our recommendation springs from our desire to ensure that the resulting statistical analysis be both simple and high quality. While the train-screen-evaluate split process is data-hungry, it is the simplest way to achieve an unbiased estimate of out-of-sample metamodel performance.

2. Cross-Validation for Metamodel Evaluation

Another technique used for assessing out-of-sample metamodel performance is *K-fold cross-validation* (K-CV). The strategy involves splitting the training M&S outputs into K subsets, known as *folds*. Then one fits K models, each model leaving out one of the K folds of data in the fitting process, and assesses the model's performance on its respective left-out data. This process yields K estimates of out-of-sample performance, which the analyst can analyze and aggregate as desired.

CV thus resembles the use of a screening set but not an evaluation set. It is a popular technique for choosing tuning parameters, such as the smoothing parameter for GAMs or some of the parameters in the covariance kernels of GPs. When one has chosen a metric by which to evaluate model performance, such as MSE, one can then select the value of a tuning parameter such that the CV performance is optimized, such as choosing a smoothing parameter that minimizes the cross-validated MSE. This does not guarantee that no overfitting has taken place, though. It may sound difficult to refit a model K times, but if the model and error metric are appropriately chosen, in some cases estimates of error can be computed with minimal effort (as is the case with GAMs and the MSE).

Common choices of K are 5, 10, or n. The case $K = n$ is known as *leave-one-out CV*, here called n-CV, as each observation is left out once as its own test set and all other data are used for estimation. n-CV produces an approximately unbiased estimate of the out-of-sample error, but it varies greatly because of individual variation in each of the observations (Hastie et al. 2009). Choosing fewer folds yields an estimate that may be biased, but the variance is smaller.

CV is often used in conjunction with the train-screen-evaluate split (in the training set only). CV is used freely to get a sense of a fitting procedure's out-of-sample performance as one makes fitting decisions. Often, CV is done repeatedly, and in cases such as K-CV, subsets for the folds are chosen randomly anew each time K-CV is performed. The screening set is referenced occasionally when one thinks they may have a small list of candidate metamodels for the "final" metamodel. The evaluation set is referenced most stringently, ideally only once at the end of the metamodel fitting process as a final descriptor of its predictive performance. Altogether, the data-splitting approach forms a robust framework for finding a model with the best predictive performance while resisting overfitting.

B. Prediction Metrics for Metamodels

Splitting outputs into subsamples only works if a metric exists that can describe how well a metamodel's predictions describe observations. We should define what good performance looks like. Many metrics facilitating such a definition exist, each with an intended use and distinct advantages and disadvantages. Hence we split our discussion based on the response variable under study. We start by considering metrics for binary response variables,[14] then consider metrics for continuous response variables. The metrics describe performance well but have disadvantages for metamodel selection. Information criteria handle metamodel selection better but are less intuitive and do not as directly describe the accuracy of metamodel predictions. We discuss them at the end of this section, along with best practices for using them well.

1. Preliminary Summary Statistics

Good practice before estimating complex statistical metamodels is to compute some basic summary statistics, such as sample proportions for categorical responses or means and standard deviations for quantitative responses. This should be done in the entire output set and then, after splitting the outputs into interesting subsets, based on some categorical factor believed to be important or splitting some important continuous variables into high and low values. A few splits are all that's needed, and one need not do too much splitting lest one find themselves accidentally fitting a decision tree to their outputs (contingency tables are a form of decision tree). The purpose of such summary statistics is to understand how much variation there is in the outputs and thus establish a baseline of what performance of a more complex metamodel should be. If a metamodel has no predictive capability over simply predicting the most common outcome observed in the outputs, it is a bad metamodel. Similarly, if the metamodel's root mean-squared error (RMSE, the square root of the mean-squared error) is higher than the standard deviation of quantitative outputs, the model has more error than a simple model that predicts the average outcome. An outcome so extreme seems unlikely, but hopefully the metamodel's RMSE is substantially lower than the outputs' standard deviation.

[14] Haman et al. (2022) is devoted to assessing model quality at predicting binary outcomes; we cite this publication as a reference for those wanting a longer discussion.

2. Metrics for Binary Data Metamodels

Analysts use *accuracy* to measure the errors associated with a metamodel's predictions. Accuracy is intuitive—it is just the proportion of responses that are correctly classified—but inadequate. Problems with accuracy include the following:

- It is not easy to optimize a metamodel by maximizing accuracy.
- Accuracy is a better measure for balanced observations, which are collections of observations where the proportion of possible outcomes (such as hit or miss) are roughly equal; if 90 percent of outcomes are hits and 10 percent are misses, the outcomes are imbalanced.
- Accuracy gives no partial credit for nearly predicting correctly.

Loss functions,[15] such as the Brier score, are a good alternative to accuracy. The Brier score in particular helps separate confident metamodels (with predictions commonly near 1 or 0) from underconfident metamodels (with predictions commonly near 0.5). If a success is called 1 and a failure called 0 in an output set with binary outcomes and for which predicted probabilities of success are computed, the Brier score is the average squared difference between the outcome (either 1 or 0) and the predicted probability of observing a success.[16] The lower the Brier score, the better the predictions generated. Brier scores are small when both predictions are correct and when the probability of that outcome is high. Brier scores for binary outcomes are high when the predicted probabilities of an outcome are near 50 percent or when confident-but-wrong predictions are frequent. The score also better handles imbalanced outcomes.

3. Metrics for Continuous Data Metamodels

In the case of continuous responses, we measure whether the metamodel predictions are close to the outcomes. Three common metrics are used for continuous response metamodel evaluation:

1. MSE, the average squared distance between prediction and outcome. (The measurement units of the MSE are the *squared* units of the M&S outputs under study, making MSE interpretation difficult.)
2. RMSE, the square root of the MSE, and thus in the same measurement units as the M&S outputs.

[15] A loss function is a summary of the discrepancy between predictions and outcomes. The function can be applied to in-sample data or out-of-sample data.

[16] For binary data, the Brier score and the MSE are identical.

3. Mean absolute error (MAE), the average distance between prediction and outcome.
4. Maximum absolute error (MxE), the maximum distance between prediction and outcome.

All of these metrics are effective summaries of the error distribution. We present formulas for these metrics in Table 6-1. Each formula offers a different perspective on error in the predictions; low MSE suggests the metamodel is good on average, low MAE suggests that model splits observed outputs well, and lower MxE suggests that worst-case prediction errors are well controlled. These metrics most intuitively correspond to the notion of quantifying how much error a metamodel makes.

None of these metrics, however, are fully suitable when trying to compare different competing metamodels using in-sample outputs. Suppose we are trying to decide which factors to include in a metamodel or how to use those factors (including whether or not to include interaction terms). MSE cannot increase when one includes an irrelevant factor, meaning it can only decrease and thus look artificially better with the inclusion of irrelevant information. (MAE and MxE share this malignancy.) Taken to an extreme, MSE encourages throwing in as many factors as possible, regardless of their relevance, to decrease the MSE. Such a strategy is nonsense and a clear path to overfitting.

Information criteria, such as the Akaike information criterion (AIC) or the Bayesian information criterion (BIC), seek to mitigate the problem of picking a metamodel with in-sample outputs.

4. Akaike Information Criterion

a. Basic AIC

AIC takes the simplest approach: add a term to the loss function to penalize complex metamodels. For linear metamodels, the term penalizing complexity is $2d$, where d is the number of parameters in the metamodel (including the intercept). For metamodels such as GAMs, d is replaced with a different measure of complexity, the effective degrees of freedom, which responds to the number of factors in the metamodel and the overfitting or underfitting from the smoothing parameter. AIC is

$$\text{Loss} + 2d.$$

Statistical theory implies that models should minimize the AIC.

While AIC can use the MSE or MAE to describe loss, the preferred approach is to use the log-likelihood function. This requires having a likelihood function describing the M&S output and the metamodel, which certainly is available for GAMs. If L_n is the likelihood function for the data for a generic set of parameters θ, then AIC becomes

$$\mathrm{AIC} = -2\log(L_n) + 2d.$$

AIC values can be compared under some conditions. Two metamodels built on the same data have comparable AICs and can use their AIC values to suggest which is superior, with lower AIC values being better.

b. Corrected AIC

For most metamodels practitioners encounter, the AIC prefers metamodels with many parameters, particularly in small samples. In response to this property, Hurvich and Tsai (1989) advocate using Sugiura's corrected AIC (1978):

$$\mathrm{AIC}_C = \mathrm{AIC} + \frac{2d^2 + 2d}{n - d - 1}.$$

Notice that as the sample size n increases, the difference between AIC_C and AIC becomes negligible, so the two measures are near equivalent for large sample sizes while AIC_C better picks models for small sample sizes. Hence, given the choice between the two, we recommend using AIC_C in all cases (though as discussed below, we prefer the BIC over either variant of the AIC). Given that these two statistics still behave similarly, we often refer to the AIC_C as the AIC.

5. Bayesian Information Criterion

BIC is a close cousin to AIC, but it is motivated by Bayesian statistical reasoning. In short, choosing the metamodel with the lowest BIC is equivalent to choosing the metamodel with the highest posterior probability of being the true metamodel among a set of candidate metamodels. The formula for BIC is similar to the formula for AIC,

$$\mathrm{BIC} = -2\log(L_n) + d\log(n).$$

AIC and BIC serve similar roles, so which to use? AIC is more forgiving of complex metamodels and overfitting than BIC; this is undesirable. That said, in smaller samples, BIC may overpenalize complex models. But BIC's easy interpretation in a Bayesian framework is appealing, as is having statistical theory suggesting that for large sample sizes it will select the correct model.

After considering benefits and disadvantages of these information criteria, our recommendation among the criteria is the BIC. However, we do not feel that either information criterion is a necessary reason to select a metamodel. Subject matter expertise should be part of selecting a metamodel. We see the information criteria as aids to the decision, not as having the goal to minimize themselves.

6. Warning Regarding Automated Model Selection Procedures

If one has a set of metamodels that each seem plausible for the M&S environment outputs and one needs to decide which to use, AIC is a reasonable way to decide. However, some software offers routines to automatically find the metamodel that minimizes AIC by adding or removing factors or changing functional forms, an approach sometimes called *stepwise procedures*. We do not recommend these procedures. Their search for a model will invalidate inference regarding the model's abilities. Statistical procedures usually do not account for such a selection process. See Derksen and Keselman (1992), Hurvich and Tsai (1990), Mantel (1970), Roecker (1991), and Tibshirani (1996) for more information.

Metamodel specification is an opportunity for analysts to describe the phenomenon in a way that reflects their subject matter expertise. Stepwise routines are a way to outsource to an algorithm the hard part of building a metamodel—that is, building the metamodel based on knowledge of what is being modeled. AIC is not a get-out-of-jail-free card for escaping the challenge of metamodel selection.

We recommend limiting use of the AIC to a handful of candidate metamodels with substantial differences between each other. If one suspects that a factor, set of factors, or some function of factors (e.g., interactions, quadratic terms) may be relevant to the response variable, include those terms in a candidate metamodel. Metamodels can include factors the analyst thinks are necessary to obtain a good statistical fit but are not interesting in their own right, sometimes called *nuisance factors* or *nuisance parameters*. If statistical fitting of such liberal metamodels yields nuisance parameter estimates that are either not statistically significant or have a small effect on response variable predictions in the factor space, such estimates can be reported only in a statistical appendix and largely ignored in discussion and interpretation. Using the AIC as a tiebreaker between metamodels in uninteresting edge cases may also be acceptable.

7. Summary of Metrics

Table 6-1 summarizes the metrics discussed up to this point, in addition to some metrics not discussed in depth.

Table 6-1. Common metrics used in assessing prediction quality and error; see Hastie, Tibshirani, and Friedman (2009).

Metric	Response	Formula	Interpretation
Accuracy	Discrete	$\frac{\text{Correct classification}}{n}$	Number of correct predictions made by the metamodel, with larger being better
Precision	Discrete	$\frac{\text{Correctly predicted } C}{\text{\# of cases of } C}$	How frequently a class was correctly called, with larger being better
Recall	Discrete	$\frac{\text{Correctly predicted } C}{\text{\# of predictions of } C}$	How frequently predictions were correct, or the true alarm rate, with larger being better
Brier Score	Binary	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{p}_i)^2$	MSE of predictions, with smaller being better
MSE	Continuous	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	Squared distance between a prediction and the observed outcome, with smaller being better (allows for loss compensation)
RMSE	Continuous	$\sqrt{MSE}$	Square root of MSE; this means the units of the metric match the units of the M&S outputs being analyzed, making interpretation easier
MAE	Continuous	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$	Distance between a prediction and the observed outcome, with smaller being better (does not allow for loss compensation)
MxE	Continuous	$\max_i\lvert y_i - \hat{y}_i\rvert$	The maximum error made, with smaller being better (focuses entirely on worst-case loss)
AIC	Any	$-2\log(L_n) + 2d$	Part of a formula that describes how likely a model is to minimize information loss in predictions and thus describe the outputs well, with smaller being better; requires a likelihood
Corrected AIC	Any	$\text{AIC} + \frac{2d^2 + 2d}{n - d - 1}$	Like AIC, but corrected to better control overfitting; requires a likelihood
BIC	Any	$-2\log(L_n) + d\log(n)$	Part of a formula that describes the probability of a metamodel being the true metamodel among alternative metamodels, with smaller BIC being better; requires a likelihood
Deviance-R^2	Any	$\frac{\text{Explained deviance}}{\text{Total deviance}}$	The percentage of variation in the sample that is described by the metamodel, with larger being better; requires a likelihood
Adjusted Deviance-R^2	Any	$1 - (1 - R^2)\frac{n-1}{n-d}$	Like R^2 but adjusted to punish metamodels with lots of parameters; requires a likelihood

C. Visualizations for Assessing Metamodel Calibration

All the above metrics roll up prediction errors into a single metric. While this gives a single number to focus on, aggregation removes nuance. Visualization methods provide a means to unwrap the predictions and further explore their relationship with observed outcomes. We can see for what predicted values the metamodel's predictions may be biased.

We focus on two plots for the continuous and binary case that compare metamodel predictions to observed output. Both of these visualization methods are types of *calibration plots*. A well-calibrated metamodel makes predictions that roughly correspond to observed outcomes. If a metamodel makes predictions inconsistent with observed outcomes, the metamodel may need revisiting, and different fitting decisions (like those discussed in Sections 3 and 4) may need to be applied.

1. Visualization for Checking Continuous Data Metamodels

We recommend plotting the predicted outcomes against the observed outcomes. Figure 6-1 presents an example of such a plot. If metamodel predictions are unbiased, the predictions and outcomes should center around an $\text{Observed} = \text{Predicted}$ relationship line. If the model predictions have low variance, the predictions hug the line closely. Figure 6-1 shows a (hypothetical) model with consistent in-sample and out-of-sample performance and thus no evidence of overfitting. Since observed and predicted outcomes seem to have a one-to-one relationship for all predicted values, the metamodel appears to be well-calibrated. Whether the deviation of observed outcomes from their predictions (the spread around the line) is acceptable depends on the context, but in this case we can see that the metamodel does have predictive capability and some may consider the observed deviation around the identity line reasonable.

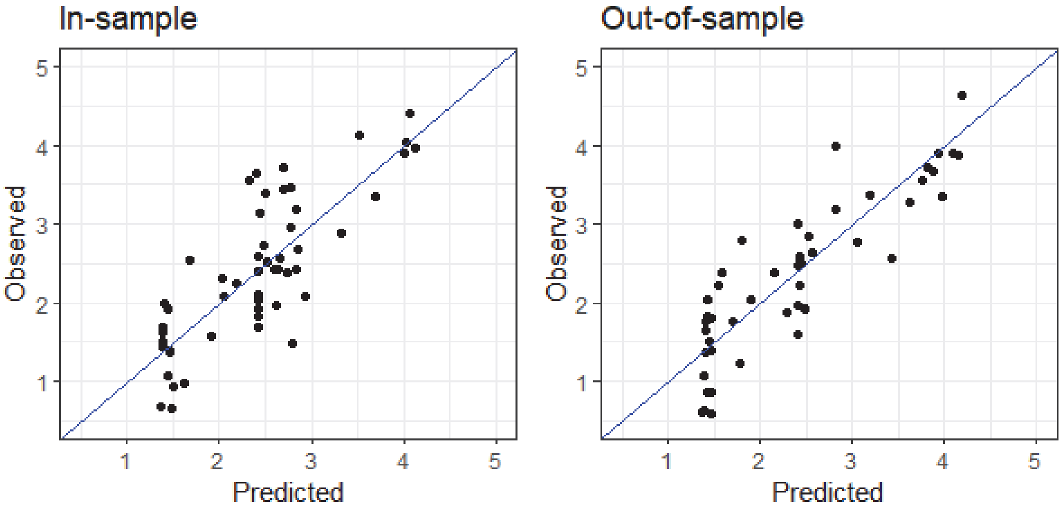

Figure 6-1. Observed versus Predicted Plot

2. Calibration Curve Visualization for Checking General Response Metamodels

For the binary case, the metamodel makes probabilistic predictions. Plotting outcome directly against probability would not yield the best plot due to outcomes being either 0 or 1. But if we fit another smoother to the observed outputs depending on the predictions, we can more easily see the relationship between predictions and outputs. Doing this is known as fitting a *calibration curve*, with the ideal curve closely matching the identity line (the line representing $\mathrm{Observed} = \mathrm{Predicted}$). All of the discussion in Section 5 about smooth fitting can be applied to effectively bin the outputs and handle nonconstant variance in predicted values. We can then examine the confidence intervals surrounding the curve.

Confidence intervals can suggest whether the metamodel overpredicts or underpredicts M&S outputs in some conditions, but they cannot say whether a calibration curve overall is statistically different from the identity line. Answering that question requires that we perform a statistical test asking whether the curve is identical to the $\mathrm{Observed} = \mathrm{Predicted}$ line. When fitting a calibration curve via a GAM, we automatically get statistical procedures to decide whether the model is appropriately calibrated. We can use the ANOVA test provided by **mgcv** (Wood 2017) to perform a statistical test deciding whether the fit is the identity line—the null hypothesis—or whether the fit is a general, smooth relationship that is not the identity line—the alternative hypothesis. If the corresponding p-value of the ANOVA test is small, we should reject the null hypothesis of perfect calibration, but this does not necessarily mean that the model should not be used. The confidence intervals of the calibration curve indicate where and how badly the model may be miscalibrated, thus allowing a nuanced decision on the model's usefulness.

These techniques are demonstrated in Figure 6-2. The top and bottom rows contain calibration plots for two different data sets; the top row's response variable is continuous, while the bottom row's response variable is discrete. These two rows are not directly connected and they are displayed in the same figure only for convenience. The left column shows a hypothetical model's fit with the observations used for fitting the metamodel, called in-sample observations; the right column shows the metamodel's predictive ability for observations not used in fitting, known as out-of-sample observations. The blue lines represent an ideal relationship between predictions and typical outcomes where predictions are equal to observations at least on average. The red lines are a smoother's estimate of the actual average value of an outcome as a function of the predicted value, surrounded by pink 95 percent pointwise confidence bands; if the blue line falls within the confidence region *at a particular predicted value*, there is insufficient statistical evidence to believe that the observed outputs differ on average from their predicted values. However, if the confidence band always contains the identity line, that does not imply necessarily that the metamodel's predictions correspond to observed outcomes on average everywhere. The number displayed in the pink box in the upper-left corner of each plot is the p-value of a statistical test checking whether the identity line and the smooth line are statistically distinguishable, with small p-values providing evidence the lines are not the same and thus the metamodel's predictions are miscalibrated. The metamodel in the top row of plots is well-calibrated in the sense that there is no evidence that its predictions differ from the ideal identity relationship, while the metamodel in the bottom row appears to be miscalibrated in both in-sample and out-of-sample outputs. The poor out-of-sample performance likely follows from the metamodel's inability to even predict outputs it saw and used for fitting, and the analyst should investigate why poor fitting is occurring and seek other modeling approaches that could improve the in-sample performance.

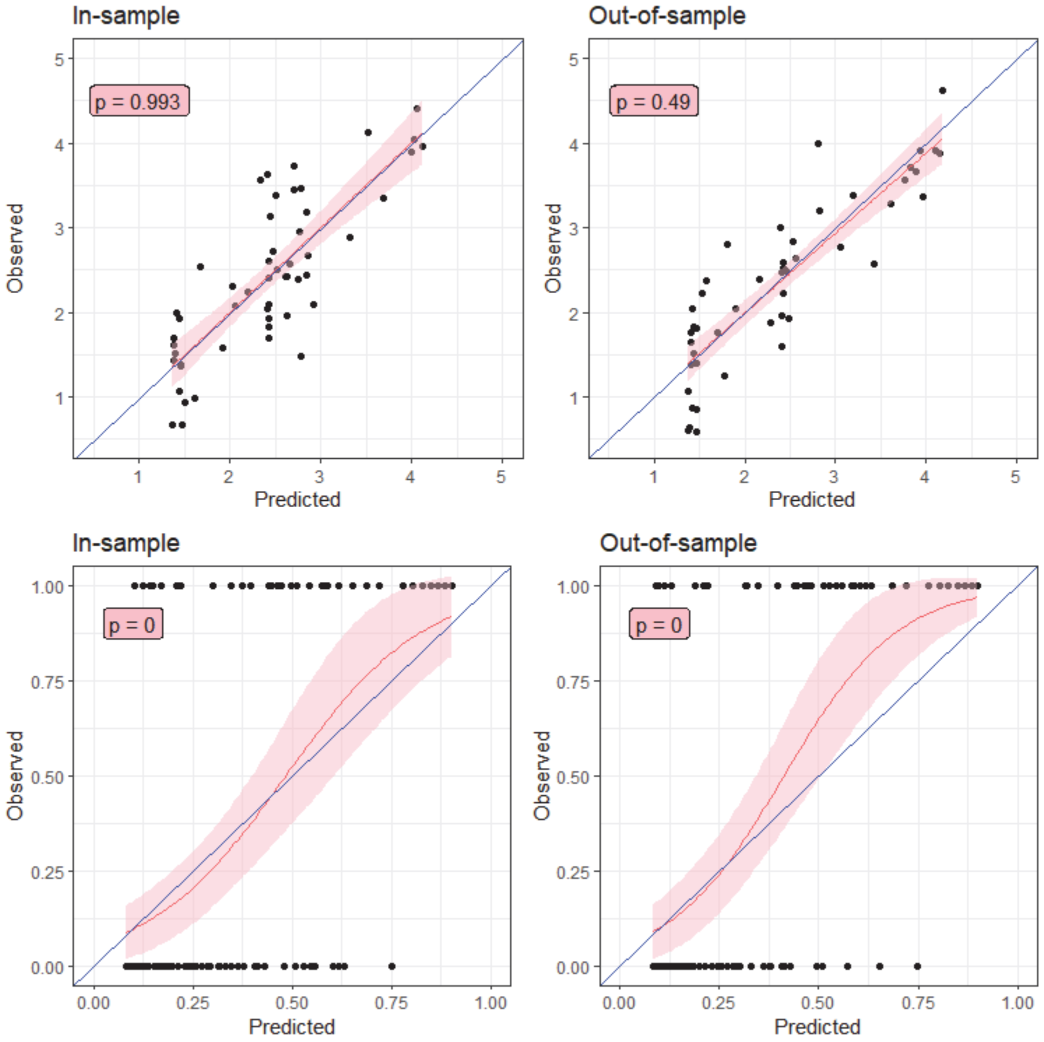

Figure 6-2. Calibration plots for in-sample and out-of-sample data, with the top panels for a continuous response and the bottom panels for a binary response. Such calibration plots provide interesting and potentially powerful plots and statistical procedures for checking whether predictions and outcomes match. Since these plots rely on smoothing though, they suffer all the complications of smooth fitting discussed in Section 4, inserting yet another layer of complexity into the analysis. If this additional complexity is unappealing, see Haman et al. (2022) for additional methods to validate models that produce probability predictions.

D. Fitting Screening and Evaluation Outputs After Metamodel Selection

The discussions above concerned selecting a good model. The screening set and evaluation set at first glance look like they were not used for model fitting, but that would be an unfair depiction of their role, as they provided information useful for selecting a model and understanding its capabilities. That said, one may wonder if, after we have selected a model, checked for pathologies such as overfitting, and quantified its predictive performance, the screening M&S outputs and the evaluation outputs could be used with the training outputs to fit a new model.

Asymptotic theory calls for using as much data as possible for model fitting. Statistical error in our estimates decreases with larger samples. If we have made all the decisions needed for fitting a metamodel and demonstrated those decisions yield metamodels that generalize beyond the sample used for fitting, getting even more precise estimates seems useful.

On the other hand, when we evaluated the final metamodel with the evaluation set, we could then refer to those estimates as good estimates of *that* metamodel's out-of-sample performance. If we then estimate a different metamodel using all data, including screening and evaluation sets, we no longer are using the metamodel that generated the metrics observed in the evaluation set. We no longer have either a screening or evaluation set, as all outputs are used for estimation. Hence, the metrics we obtained from the screening set are no longer valid for the metamodel we use.

These are the tradeoffs for refitting a model with all outputs observed. While others can disagree, we believe that refitting the metamodel with all observations after model selection, while technically invalidating the metrics obtained from the evaluation set, should not result in a grave error. We would be surprised if a set of fitting decisions yielding metamodels that consistently generalize in CV, screening, and evaluation outputs suddenly collapses when all outputs ever seen are used for fitting. A more likely culprit would be a more fundamental problem, such as changes to the M&S environment or the weapon system's software that cause different behavior never seen in the original outputs used for fitting the metamodel.

7. Recommended Experimental Designs for M&S Verification and Validation

We divide DOE into two classes: parametric DOE and SFDs. Parametric DOE consists of DOE methodology designed for parametric regression metamodeling, including linear statistical models. SFDs place design points in such a way that the factor space is filled with design points and the designs are model independent.

Many designs for parametric statistical metamodels use design points selected to produce good statistical properties in the fitted metamodels. Simple linear models prefer points near the edges of the factor space since the edges often are the best locations for placing points; such points minimize the standard errors of the metamodel coefficients.

SFDs do not attempt to be good for a specified model and instead try to explore the whole factor space. Both approaches have advantages and disadvantages, and the type of study dictates which approach is more appropriate. In low SNR situations, statistical error is the biggest concern; parametric DOE should be used. In high SNR situations, we can tackle model uncertainty using SFDs. We discuss both DOE approaches briefly.

More extensive recommendations are available in Wojton et al. (2019) and Wojton et al. (2021).

A. Space-Filling Designs: Designs for Low-Noise Metamodels

M&S environments may have little or no noise in their response variable. Classes of models for which this would be the case include DSIM, HITL, and SITL models, or federations involving these types of simulations. DSIM models in particular may have no noise at all. HITL models often execute in real time and involve communication between different computer systems. This can produce some natural variation in output.

SFDs are well-suited to collecting data for metamodels and are recommended over parametric DOE. These designs care mostly about distributing points throughout the entire space. Thanks to this property, they can more easily find interesting local phenomena in the data.

We generally recommend MaxPro designs (Joseph et al. 2020) for metamodeling. Depending on test-specific features, uniform designs or sliced maximin Latin hypersquare designs (maximin SLHDs) are recommended.

1. Latin Hypersquare Design

Latin hypersquare designs (LHDs) for continuous factors divide each factor into n levels of equal length (thus dividing the factor space into n^d rectangular regions if there are d factors), then choose combinations of these levels so that each level for each factor is used once, guaranteeing that all levels of each factor are seen once. This helps ensure that the factors in the design are equally well-covered individually, but this requirement does not fully specify the design, and some designs qualify as LHDs but would not be considered "space filling."

A test planner may add another requirement to the LHD, such as finding the LHD that makes the minimum distance between two points in the design as large as possible. Such a design is an LHD-maximin design.

2. Maximin Design

A maximin design maximizes the minimum distance between two points in the design. By ensuring no two points are closer to one another than they need to be, the design encourages points to spread throughout the space. Such designs tend to place points near the borders of the factor space, and when one looks at how the design fares when one or more factors are dropped from consideration, the resulting design may appear to not fill space as well. These issues can be alleviated by using an LHD-maximin design.

3. Sliced Latin Hypersquare Design

Generic LHDs are for continuous factors and do not automatically incorporate categorical factors. SLHDs incorporate categorical factors by generating "slices" such that not only do we have an LHD ignoring the slices, the data within a slice also are an LHD, thus preserving LHD coverage properties. The slices themselves correspond to combinations of the categorical factors. As with LHDs, we can couple SLHDs with the maximin criterion and thus have maximin SLHDs. A full factorial approach would assign all combinations of the categorical factors their own slice, but if there are many categorical factors, this can result in a combinatorial explosion in the number of slices and thus not be practical.

4. MaxPro Design

MaxPro designs are able to ensure good factor coverage like LHDs, spread points out like maximin designs, and handle categorical factors like maximin SLHDs. We recommend MaxPro designs over maximin SLHDs because MaxPro designs can better handle categorical factors when the number of such factors and the study's sample size are practical numbers.

The R package **MaxPro** can generate such designs. Our current process for generating designs with binary factors using the package is:

1. Generate a maximin SLHD with the appropriate number of continuous factors and some number of slices.

2. Join the slices of the maximin SLHD with the rows of a fractional factorial design with the appropriate number of categorical factors and with the number of rows matching the number of slices in the maximin SLHD.
3. Feed the resulting design into the *MaxProQQ()* function from the **MaxPro** package.

The resulting design generated by the software is a DOE we may execute to collect M&S observations.

4. Fast Flexible Filling Design

Fast flexible filling (FFF) designs rely on clustering to select test points. FFF designs can easily handle restrictions on factor combinations and categorical factors.

Figure 7-1 visually illustrates maximin LHD, SLHD, and FFF designs. A MaxPro design would resemble the maximin SLHD.

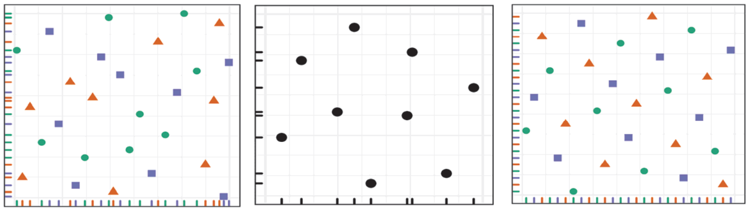

Figure 7-1. Space-filling designs. From left to right: maximin LHD, maximin SLHD, and FFF. Different colors or shapes indicate different levels of a single categorical factor.

B. Parametric Experimental Designs: Design of Experiments for High-Noise Metamodels

If testers can expect highly stochastic data, we may not want to fit a nonparametric model. This would be the case for models that are not purely digital, such as OITL models or models with physical features, such as natural models or physical models. If this is the case, parametric DOE may be preferred. These designs may care less about spreading points through the factor space and more about placing points in regions useful to fitting the desired statistical model.

1. Factorial Design

A full factorial design is a design in which each combination of factor levels is observed equally frequently. A full factorial design can be untenable when the design has more than a small number of factors. (For example, a full factorial design involving eight levels for each of four continuous factors would require a multiple of 4,096 runs, with the size growing exponentially with each additional continuous factor). Fractional factorial designs, which include some of the runs seen in full factorial designs, avoid this issue. Hence, fractional factorial designs often are more practical and thus more common.

2. Response Surface Design

Response surface designs place test points in a star orientation, with prominent placement of central points and points outside of a rectangular region. These designs allow for fitting models consisting of terms other than linear and interaction terms, such as quadratic terms. If the response variable truly has a nonlinear relationship with factors, this would be an important feature of the design. We recommend these designs in low SNR situations with few or zero categorical factors.

3. Optimal Design

In this paper, an optimal design is an experimental design for a regression model. Optimal designs place design points in such a way that the resulting design optimizes some quantity of interest of the model to be fit, such as minimizing parameter standard errors. Optimal designs are ideal if the assumed model is valid, but they can be fragile to deviations from those assumptions or be far less efficient if we deviate from the parametric model assumed. Generally, we recommend optimal designs if most or all factors are categorical and if there are restrictions on data collection.

8. Examples of Metamodeling with Generalized Additive Models and Gaussian Process Interpolators

We present an analysis of a paper plane simulator, which generates flight paths for paper planes. M&S analysts should easily understand this example and be able to replicate it; the software is small, there are no access restrictions compared to models of weapon systems, and most people reading this paper have thrown a paper plane at least once and know something about how they work. Also, one can run this example on any modern personal computer. The example represents an idealized M&S context, with the paper plane simulator designed specifically to allow easy metamodeling.

The paper plane simulator is an M&S environment that solves a set of ordinary differential equations (ODEs) to generate flight paths for a paper airplane.[17] The simulation tracks a plane's velocity, flight angle, height, and range. Gravitational force, air density, drag and lift coefficients, reference area (i.e., wing area), and mass contribute to the flight path; see Stengel (2004) for more information. The simulator is a numerical ODE solver for these equations, and thus it is a fully digital and deterministic simulation. However, if one were to add randomness to some inputs, such as a random error in the initial velocity or angle of the flight, it would become a stochastic simulation. We consider both cases here to demonstrate multiple techniques.

Factors of interest include *angle* and *airspeed*. The plane's design factors into the plane's reference area (i.e., the wing area) and the lift and drag coefficients. Rather than varying these like any other continuous factor, they should be considered part of a categorical factor of *plane design*.

[17] The ODE system is

$$\dot{V} = -C_D(\rho V^2/2)S/m - g\sin(\gamma)$$
$$\dot{\gamma} = (C_L(\rho V^2/2)S/m - g\cos(\gamma))/V$$
$$\dot{h} = V\sin(\gamma)$$
$$\dot{r} = V\cos(\gamma)$$

where V, γ, h, and r are the velocity, flight angle, height, and range of the flight path, respectively (a dot means a derivative with respect to time); C_D is the drag coefficient; C_L is the lift coefficient; ρ is the air density; g is the force of gravity; m is the mass of the paper airplane; and S is the reference area (i.e., wing area) of the paper airplane. See Stengel (2004) for more information.

We consider three response variables:

- The presence of a loop in the flight path,
- The range (horizontal distance) traveled, and
- The number of bumps in the flight path, which is a discrete outcome.

The metamodeling of these three variables form the different examples in this section. We devote one example to a deterministic metamodel for flight range, and we devote three examples to stochastic metamodels of each of these responses.

Cheap data generation allows many runs and easy variation, so we used a maximin SLHD to generate training, screening, and evaluation samples. For the deterministic case, since bump and loop counts are discrete variables, we used nearest neighbor interpolation for prediction. Range is a continuous variable, so we fit range with a GP using a Matérn kernel. Plotting the interpolator shows its ability to discover multiple unique phenomena that would be difficult to discover without interpolation, such as two local extrema for the world-record paper airplane (WRPA) design. Accompanying error estimates are also small, suggesting we achieved a good fit.

For the stochastic simulations, we fit GAMs to all three response variables of interest, but we demonstrated different fitting strategies and software. For fitting range, we used the R package **caret**; it offers an easy and fast interface for doing some common fitting tasks, but unfortunately it does not support many options for GAM fitting compared to using **mgcv** and taking a more manual approach, which we did for the categorical response variables. We found that using a t distribution and including initial airspeed and initial angle as smooth functions yielded the best fit. We implemented a train-screen-evaluate split for finding a metamodel to predict loops in flight. The best GAM we found was a logistic regression metamodel that uses a linear response for initial airspeed and a smooth term for the initial angle; the metamodel did well in predicting loops in the evaluation set, better than a simple prediction based only on the proportion of loops observed in the sample. Finally, for predicting the number of bumps in a flight path, we used K-CV (specifically, 10-CV) in the training sample in addition to a train-screen-evaluate split. The resulting best metamodel was a linear statistical model (without smooth terms) estimating the mean of a zero-inflated Poisson (ZIP) response distribution, which again demonstrates some superior predictive ability over simple summaries depending only on overall sample mean and standard deviation.

A. Experimental Design for Paper Plane Simulator

We created a 300-run experimental design investigating two continuous factors and one categorical factor: airspeed (meters per second), angle (usually stated in radians but sometimes in degrees), and plane design. We consider three paper plane designs: the dart, classic, and WRPA.[18] Figure 8-1 shows the differences between these plane designs, showing both the designs explored in this paper and the designs not explored in this paper but supported by the software. Initial height is fixed at 1.5 meters, and the planes are allowed to fly for a maximum of 15 seconds. Initial airspeed ranges from 1 to 12 meters per second, and angle ranges from 60° below ($-\frac{\pi}{3}$ radians) to 60° above ($\frac{\pi}{3}$ radians) the horizon.

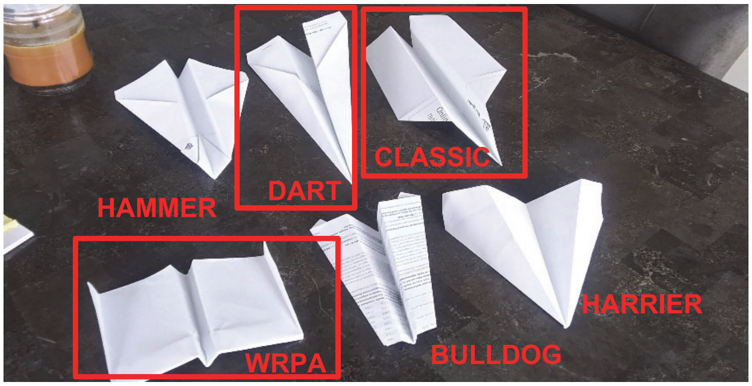

Figure 8-1. Different paper plane designs. Boxes indicate the designs used in computer experiments in this study.

Figure 8-2 shows simulator output for a plane flight.

[18] The WRPA plane once set a Guinness world record for the longest flight, but it has since been overtaken by other designs.

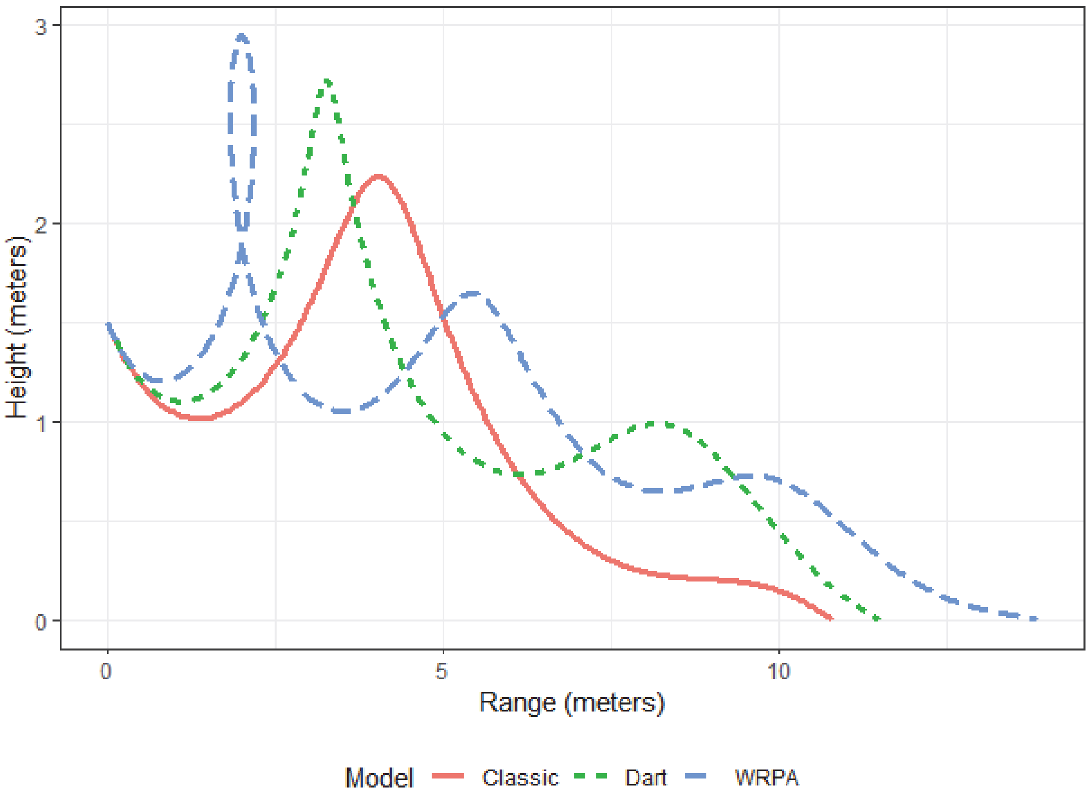

Figure 8-2. Flight paths for three different paper planes, with an initial height of 1.5 meters, initial airspeed of 8 meters per second, and initial angle of 40° below the horizon. Note the bumps in all flight paths and a loop in the flight path of the world-record paper airplane.

We generated a maximin SLHD experimental design, shown in Figure 8-3. There are three slices because there is one categorical factor (plane design) with three levels: dart, classic, and WRPA. The other factors (airspeed, angle) are continuous and thus are varied via the Latin hypersquare sampling scheme. This approach should spread points roughly evenly throughout the space while ensuring that factors are well-covered individually to give us a dense data set for metamodeling. For more information, see Wojton et al. (2021).

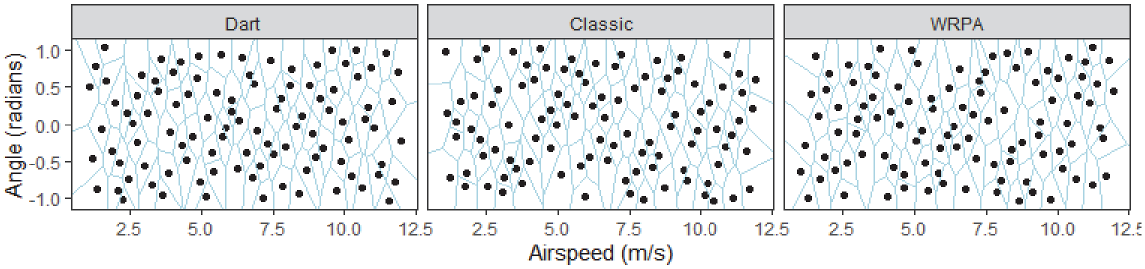

Figure 8-3. Visualization of the maximin SLHD experimental design used. Each dot represents where an observation for the experimental design will be collected. The dots are embedded in a Voronoi diagram, where all points within a cell are closest to the dot in the cell.

B. Deterministic Analysis Using Gaussian Processes

The paper plane simulator is a deterministic simulation. The plots in Figure 8-4 show terminal range as well as bump and loop counts, and the plots display the counts using the Voronoi cells shown earlier. While this figure looks information dense, the implied metamodel would be fine for the discrete response variables (bump and loop counts) but not for range, which we believe should be described with a continuous surface. Hence, to better describe range, we use a GP.

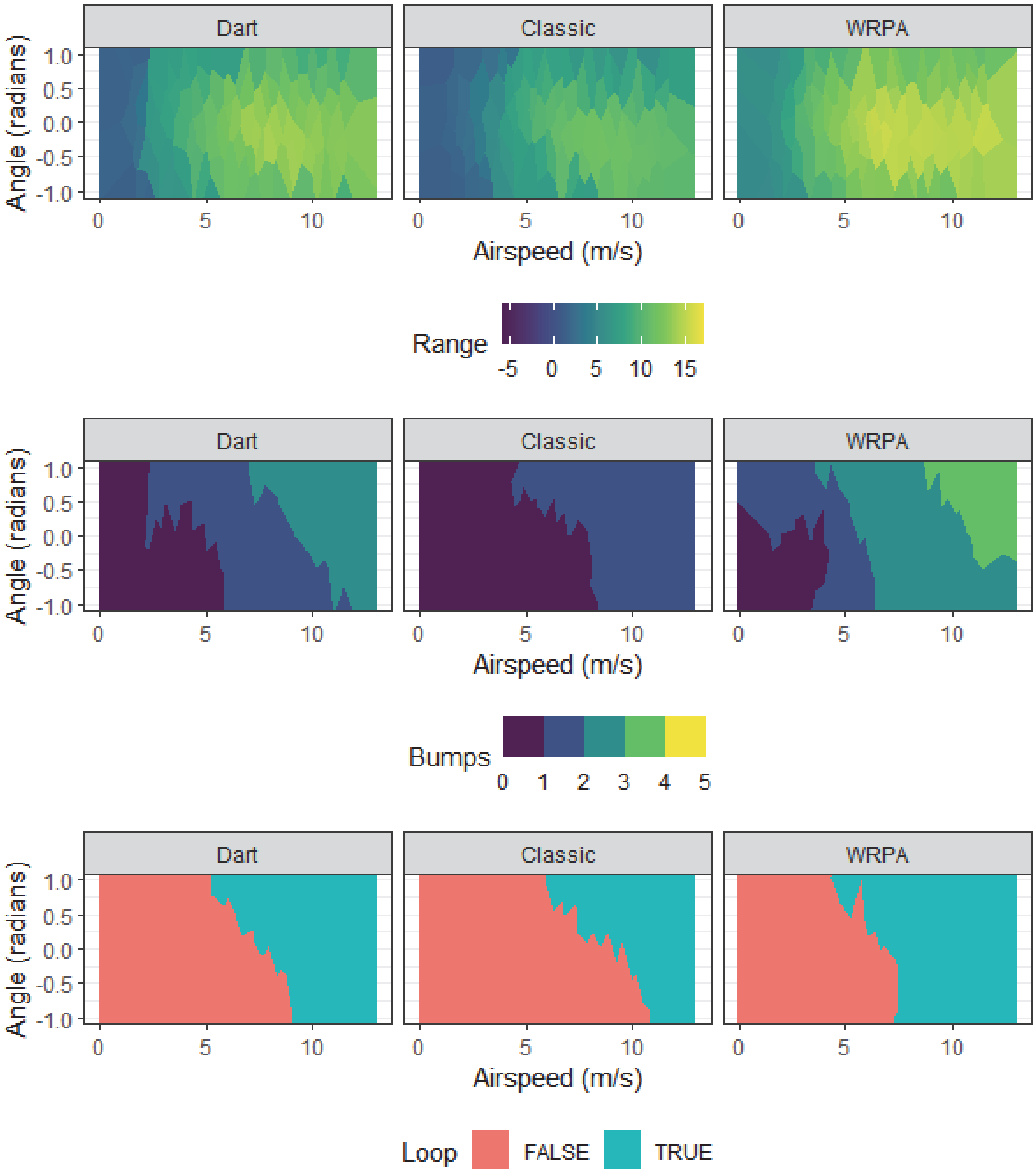

Figure 8-4. Observed response variables from the deterministic simulation using the sliced Latin hypersquare experimental design. To generate this figure, we colored in the cells of the Voronoi diagram shown in Figure 8-3 with the nearest response variable's value in the DOE. Hence, this figure can be seen as showing the predictions of a nearest neighbor interpolator.

We fit a separate, independent GP for each model of plane. We use the R package **GPfit** for fitting the GP (MacDonald, Ranjan, and Chipman 2015). We use GP with a Matérn kernel and with terminal range depending on initial airspeed and initial angle. Since ν cannot be estimated with **GPfit**, we set $\nu = \frac{9}{2}$, which corresponds to having four continuous derivatives. The other parameters of the GP will be estimated from the data via the maximum likelihood method. While interpolation is the goal, our fit includes a small nugget effect for computational ease.

After fitting the GP (shown in Figure 8-5), we can plot and consider the resulting interpolation. The resulting surface is nonlinear. The GP for the dart plane design is the smoothest and most regular, while the classic and WRPA plane designs feature interesting ridges in the surface. WRPA also has multiple local maxima.

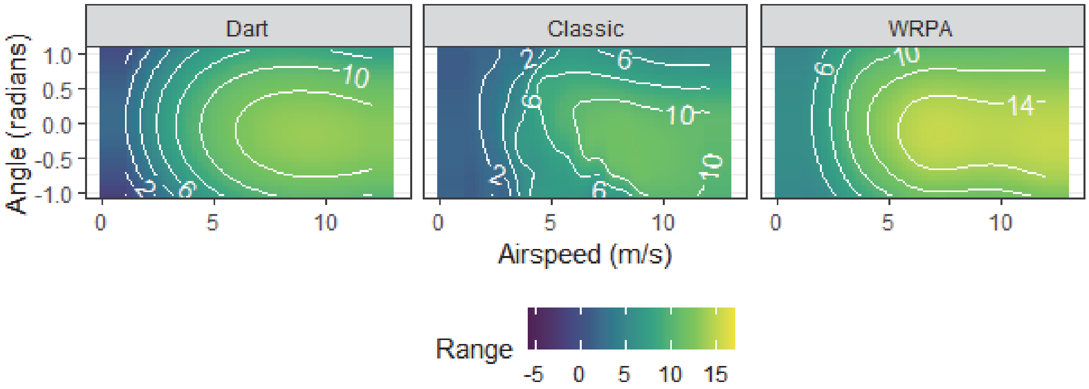

Figure 8-5. GP Interpolation of the Observed Simulation Ranges

GPs yield predictions and uncertainty estimates. We can use the estimated MSE of the process to create confidence bounds by adding and subtracting the square root of the MSE scaled by an appropriate quantile of the normal distribution[19] to the predicted value of the range. We can then plot the lower bound, upper bound, and width of these intervals, as done in Figure 8-6.

[19] In the case of 80 percent confidence level intervals, the quantile is approximately 1.28.

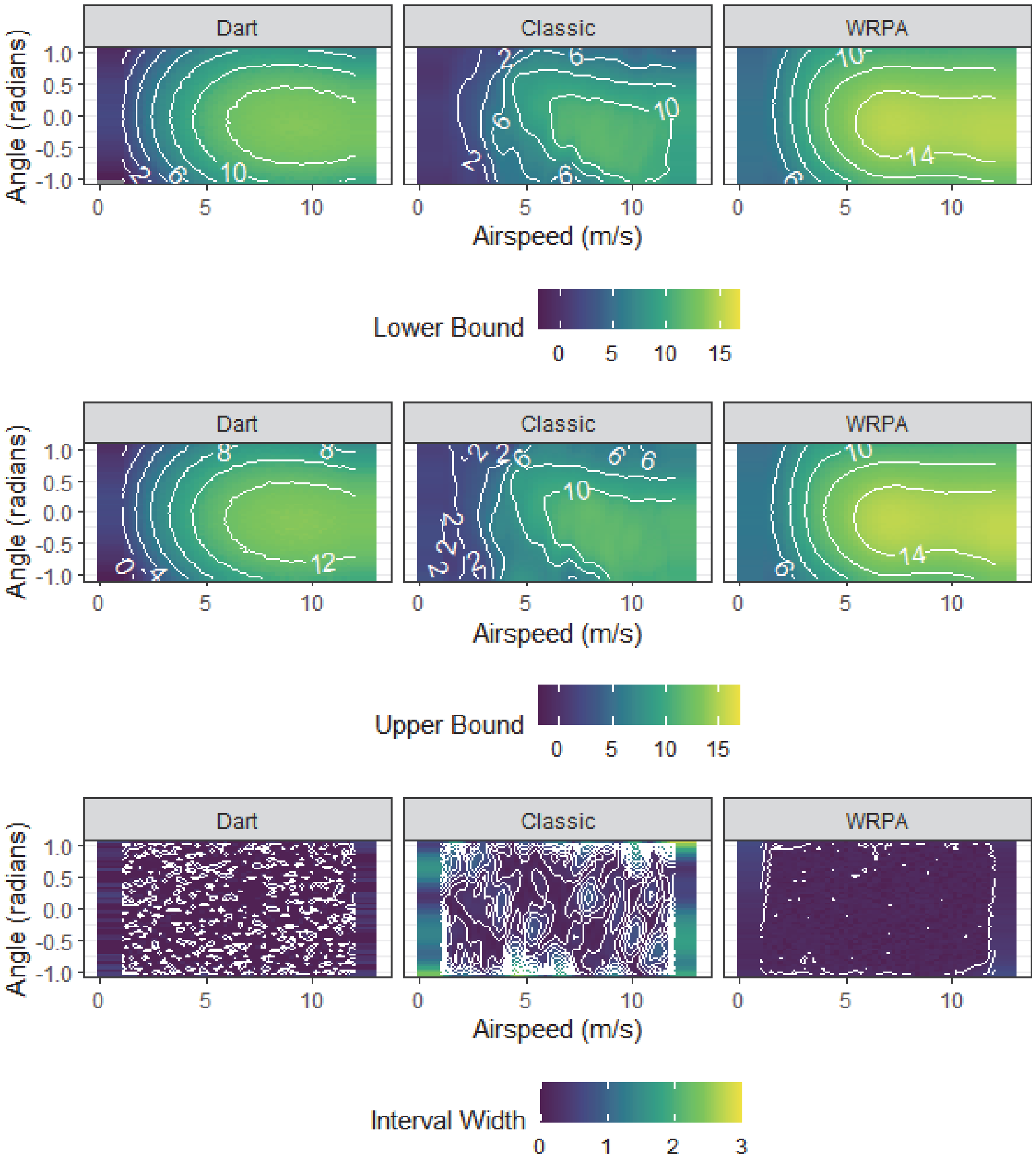

Figure 8-6. From Top to Bottom: the Lower Bound of the Confidence Region, the Upper Bound of the Region, and the Width of the Region

Because we use an SFD, uncertainty throughout most of the parameter space is low, especially for the WRPA design (which has the smoothest fitted surface). Uncertainty as measured by confidence interval width is highest near the edges of the input parameter space, which is not surprising as there are fewer outputs in these regions.

C. Stochastic Analysis Using Generalized Additive Models

Now we assume that the flight path depends on random initial airspeed and random initial angle, representing that people are unable to throw paper planes with perfect precision regarding either the initial speed or angle. Specifically, the initial airspeed and angle are normally distributed, with a standard deviation of 1 meter per second for airspeed and 15° ($\frac{\pi}{12}$ radians) for angle. (Now, the specified airspeed or angle is instead the mean airspeed or angle.) As a result, the range, number of bumps, and number of loops in a flight path are random as well, though how they are affected by the random perturbation is not immediately obvious.[20]

Our aim now is to estimate the average value of the response via smoothing. We fit GAMs to describe the terminal range of the paper plane as well as count bumps and the presence of loops. Terminal range is a continuous response, the presence of loops is a binary response, and the number of bumps is a discrete numeric response.

For these examples, we use the same experimental design as the one used in the deterministic case. Figure 8-7 plots the three outcomes.

[20] This simulation can be characterized as ordinary differential equations (ODEs) with random initial conditions. Since the ODEs and initial condition distributions are fully described, we could mathematically describe the distribution of the flight path's parameters at selected points in time using the procedures given in Chapter 6 of Soong (1973), Liouville's theorem in particular; from this, we could compute a distribution of the range.

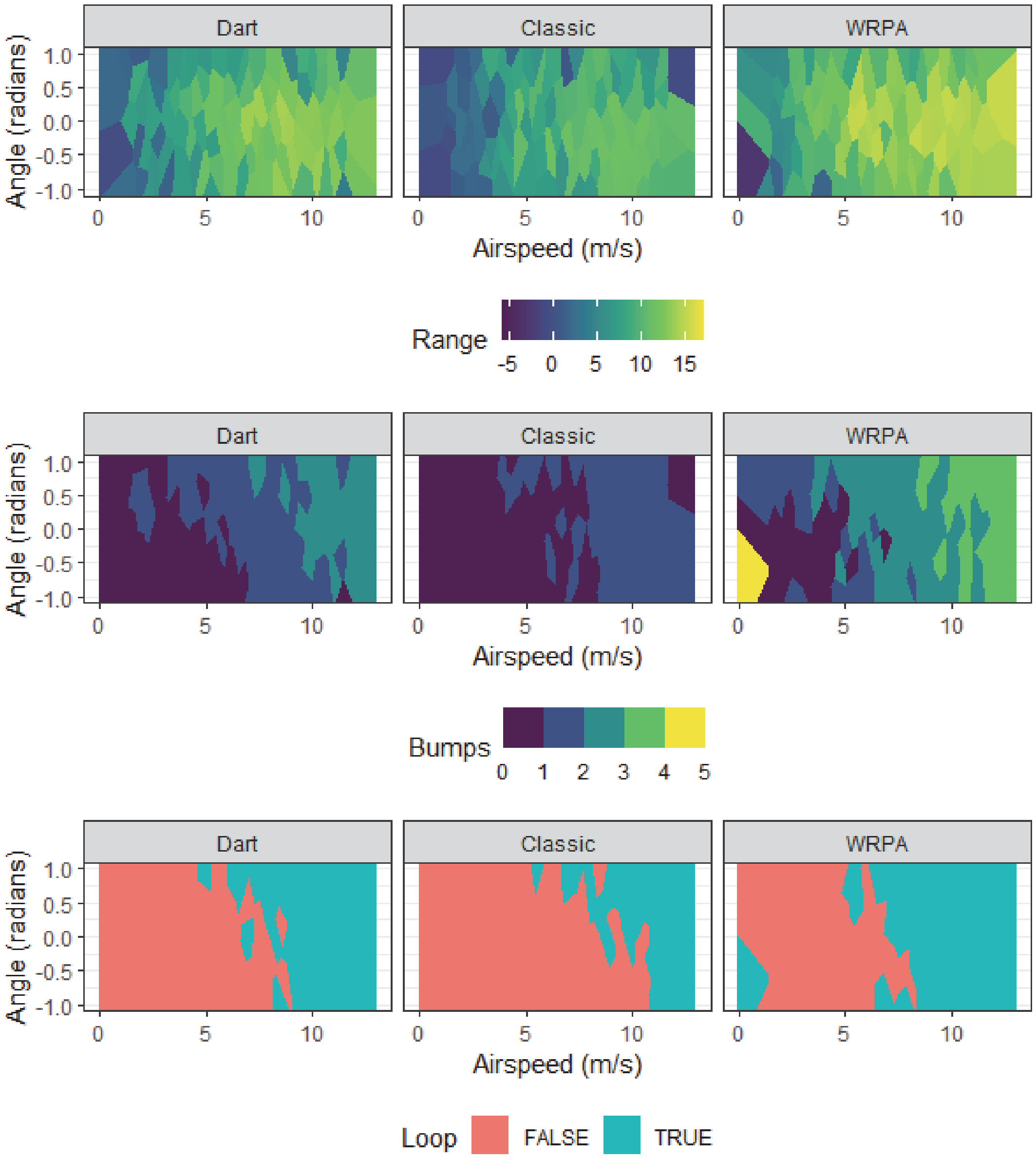

Figure 8-7. Observed response values from a stochastic paper plane simulation. Unlike those seen in Figure 8-4, these values are random.

We will use the above output set as the training outputs. We will have two other output sets: the evaluation output set, which we will generate later, and the screening output set. We have 50 observations per model in the screening output set and 50 per model in the evaluation output set. The designs for these output sets will also be maximin SLHD experimental designs.

We should avoid looking at these output sets, including visualizing their results, other than perhaps checking that the factors are adequately varied. Minimizing contact with them will preserve them as effective out-of-sample M&S output surrogates.

The type of response affects which distribution family to use in the metamodel, which in turn affects the link function and the likelihood function. GAMs are flexible enough to handle these varied specifications and will be applied to all stochastic responses in these examples.

Several R packages facilitate applying and exploring GAM fits. The R package **mgcv** (Wood 2017) provides excellent formula specification, fitting specification, and diagnostic tools. The **caret** package (Kuhn 2022) provides interfaces to facilitate model fitting while incorporating general purpose techniques like output splitting, CV, and others. The **caret** package supports models provided by other packages, including **mgcv**, so that its routines can be easily incorporated into many standard statistical models. We demonstrate metamodeling with GAMs using both **mgcv** and **caret**.

The **caret** package fits GAMs using the *train()* function, its general purpose model-fitting function capable of fitting lots of models. For GAMs, *train()* will attempt to select the smoothness parameter and which terms in the model should be smoothed when we tell the function how to make these decisions.

The R package **mgcViz** (Fasiolo et al. 2018) provides visualization tools for GAMs fitted using **mgcv**, and it works with the GAMs yielded by **caret**'s *train()*.

1. Example GAM for Predicting Range of Paper Planes

We start by developing a metamodel for range. We have three factors to include in our metamodel: initial airspeed, initial angle, and plane design. These factors can be incorporated in many ways, including whether and how we would account for interactions. We are not obligated to fit separate GAMs for each plane design, unlike in the GP case.

First, we use a normal distribution to describe the response distribution. To assess our distribution selection, we look at diagnostic plots to check for problems in the fit. Figure 8-8 presents a common diagnostic plot set. The Q-Q plot compares the deviance residuals (a type of residual from generalized linear model theory; see Wood (2017) for details) to their theoretical quantiles if the distribution of the response is correctly specified. This relationship should be a straight line; however, we see that these residuals are consistently under the line on the left-hand side of the plot and consistently over the line in the middle of the plot.

The histogram of the residuals should resemble a normal distribution, but there appear to be too many observations in the tails of the distribution than a normal distribution would imply.

The residuals versus linear predictor plot should look cloudy and lack a discernable pattern. Higher predictor values suggest less variability in the residuals, and midrange predictor values seem to be associated with higher variability in the residuals; ideally, the variation in the residuals should be constant throughout the range of the linear predictor.

Finally, when comparing the response to the predicted value, we should see a straight one-to-one line, with less variation around that line being better.

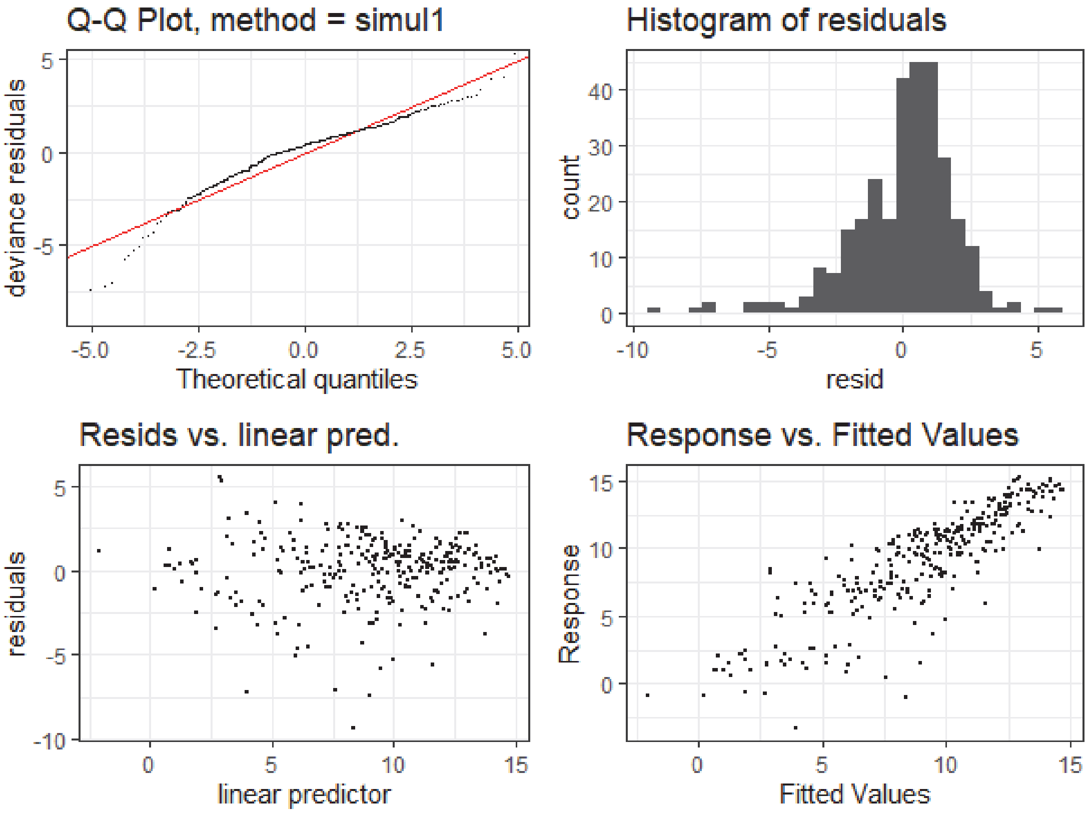

Figure 8-8. Diagnostic Plots of the Fitted GAM When Modeling the Response Variable with a Normal Distribution

Aside from the problems seen in the diagnostic plots, other metrics for metamodel fit tell a better story. The diagnostic report given by *check()* indicates no problems when estimating the GAM smooth. As for performance, with an RMSE of 1.94 meters, an MAE of 1.41 meters, and an R^2 of 75.4 percent, this fit does not look too bad, and it may be an adequate metamodel for the M&S environment. When switching the response distribution, we should see whether these metrics change. These metrics are all in sample, but if we compute the metamodel's performance in the screening set (using the fit obtained from the training outputs), we get an RMSE of 1.76 meters, an MAE of 1.36 meters, and an R^2 of 81 percent, again not bad. Since our goal is making a metamodel that can accurately predict, we should keep the metamodel that has better metrics.

The potential problems noted above could be due to assuming a normal response variable with constant variance. There are several remedial measures, including using a more robust model or a model for nonconstant variance. Hence, we will try one more distribution, the scaled Student's t distribution. The resulting diagnostic plots are shown in Figure 8-9. While the nonconstant variance concerns still prevail, the plots assessing the appropriateness of the distribution look better, albeit not great. This suggests that an improved metamodel may jointly model the location and variance of the flight distance; such approaches are out of scope of this tutorial, and thus we will not consider them. With an RMSE of 1.73 meters, an MAE of 1.31 meters, and an R^2 of 82 percent (all in the screening outputs), the metamodel using the scaled Student's t distribution appears to have good predictive properties. Given the other benefits, we will keep it.

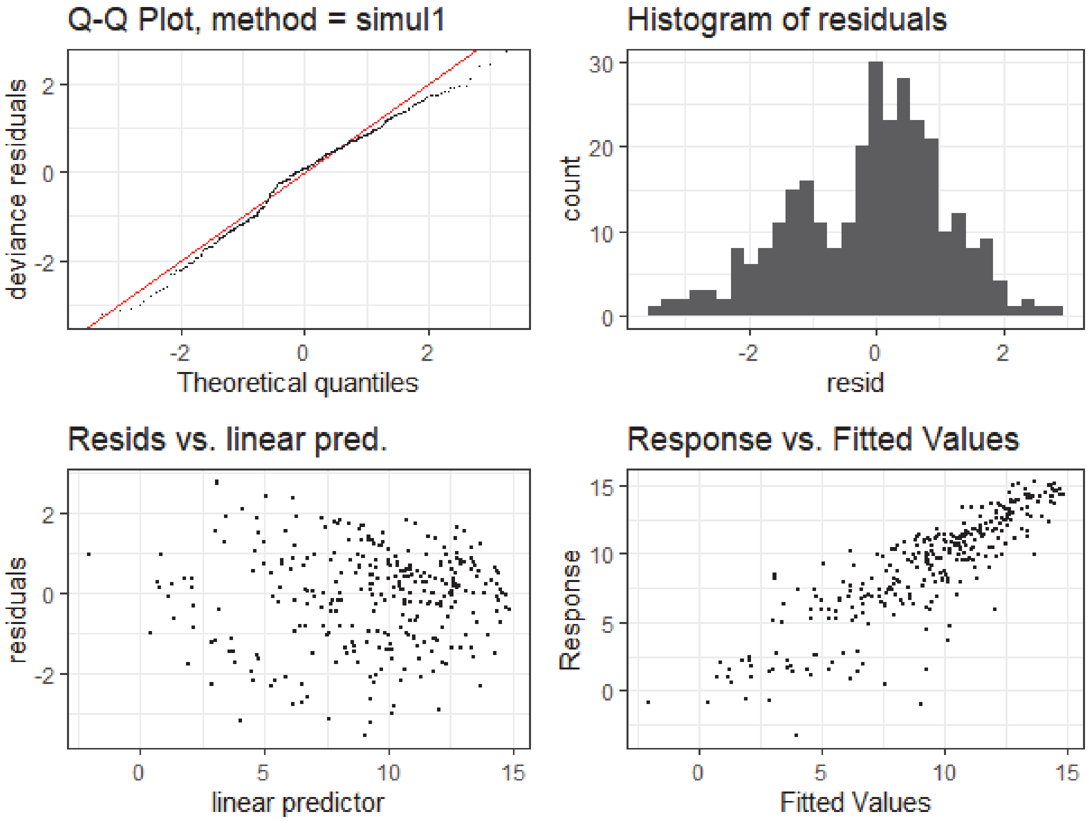

Figure 8-9. Diagnostic Plots of the Fitted GAM for the Range but Using a *t* Distribution for the Response Variable

The *train()* function determined which parameters should be smoothed parameters based on the number of unique levels in the parameter, not metamodel selection metrics such as AIC or BIC; hence, for GAMs as described in this paper, metamodel selection is not innately supported by **caret**.[21] To our knowledge, *train()*'s model determination procedure allows only univariate functions to appear in the GAM, precluding multivariate smoothing surfaces for M&S factors. For now, we will accept the metamodel generated by *train()*, but later we will use **mgcv** to fit and explore more complex metamodels.

Plots are useful for understanding the fitted effects. Figure 8-10 is an example. These plots show the marginal effect of initial airspeed and initial angle on terminal range. The

[21] The *gamboost* models in **caret** do support feature selection. These are GAMs using a procedure known as boosting; see Hastie, Tibshirani, and Friedman (2009).

grey error bands show pointwise confidence bounds, meaning confidence intervals describing uncertainty about the response function's value for particular airspeeds or particular angles, as appropriate. Recall that identifiability conditions for GAMs require that the integral of these functions each be 0; the values on the vertical axis of the plots reflect this requirement. To predict the terminal range of a paper plane from these plots, add the value of the curve at a selected initial airspeed and initial angle to a coefficient dependent on the plane design—approximately 9.3 meters for the dart design, 7.4 meters for the classic design, and 11.3 meters for the WRPA design. Hence, based on eyeballing the chart, if we threw a dart plane at an initial airspeed of 2.5 meters per second level to the ground, we would predict the plane would fly about 6.2 meters by starting with the baseline number for the dart plane (approximately 9.3 meters), adding the contribution of the initial airspeed at 2.5 meters per second (approximately −4.2 meters), and adding the contribution for throwing at a level angle (approximately 1.1 meters). Without considering the baselines for each plane design, the plots tell us how much terminal range changes when we compare potential factor levels; for example, doubling the initial airspeed from 2.5 meters per second to 5 meters per second increases the terminal range of any of the planes by approximately 6 meters on average.

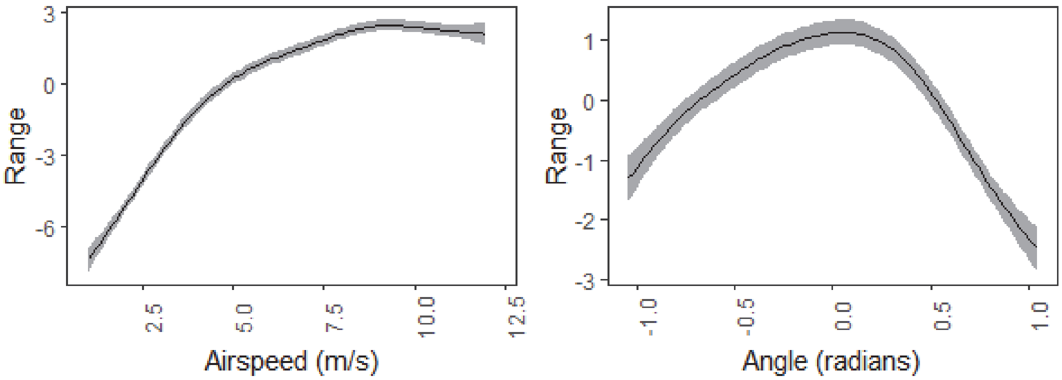

Figure 8-10. Estimated Response Functions for the GAM Predicting Paper Plane Terminal Range

The shapes of the functions also tell a story. For instance, the optimal angle to maximize range seems to be near zero. The relationship between initial angle and terminal range appears mostly but not quite symmetrical; in particular, the highest angle appears to have a considerably lower terminal range on average than the lowest angle. (We could fit a GAM that assumes a symmetric relationship and perform a hypothesis test to determine if the relationship is symmetrical, but we do not do so in this tutorial; Wood (2017) demonstrates such testing.) The launch airspeed's effect on range diminishes around 7.5 meters per second, and the change appears abrupt. Combining these functions with the categorical effects yields a surface of predicted values, as shown in Figure 8-11.

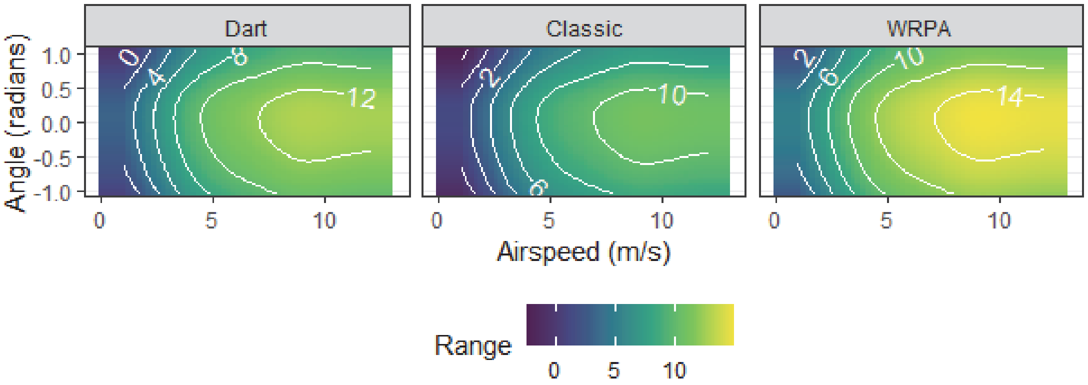

Figure 8-11. Prediction Surfaces for Terminal Range Using the Fitted GAM

If we create a calibration plot estimating a smooth relationship between observed and predicted values, we can get a better sense of whether the metamodel is miscalibrated and for what predicted values. Figure 8-12 shows that while the metamodel is mostly correctly calibrated, there are regions where it seems to predict the M&S output poorly. In particular, the variation in predictions does not appear to be constant; there's more variation for larger predicted terminal ranges (over 10 meters) than for predicted terminal ranges in middle regions (between 5 and 10 meters). However, the predictions appear to be correct on average throughout the factor space.

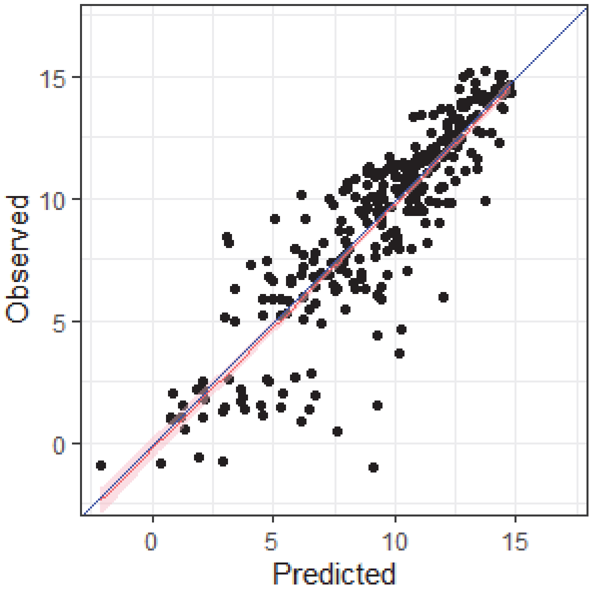

Figure 8-12. Calibration Plot for Observed versus Predicted Terminal Range

To assess lack of calibration, we conduct a statistical test at the 80 percent confidence level. We can conclude there is not statistically significant evidence for miscalibration ($p \approx 0.7674$). The confidence intervals tell us the difference may be greatest at moderate predicted values, but overall the metamodel makes good predictions.

At this point, we have decided to use the GAM fitted by *train()* with the chosen family being the scaled t distribution. Now we can quantify final performance of the fitted metamodel in the evaluation set. In the evaluation outputs, the metamodel has an RMSE of 1.94 meters, an MAE of 1.48 meters, and an R^2 of 76 percent. These metrics did not change much from the training output metrics, suggesting we likely avoided overfitting while having a metamodel with reasonably good predictive performance. The calibration curve given in Figure 8-13 similarly suggests the model is well-calibrated; the corresponding statistical test does not reject the possibility of the identity relationship for the calibration curve ($p \approx 0.3266$).

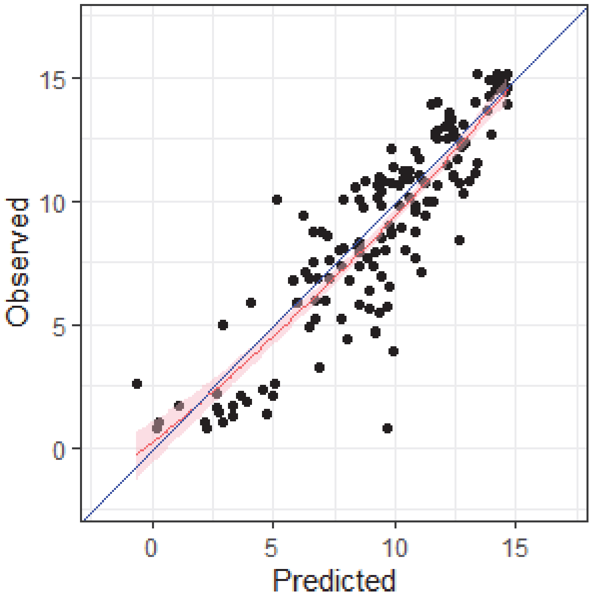

Figure 8-13. Calibration Plot for Observed versus Predicted Terminal Range in the Evaluation Set

2. Example GAM for Predicting Loops of Paper Plane Paths (Bernoulli Response)

In this example, we focus more on metamodel comparison to demonstrate some of the metamodel evaluation techniques from Section 6. We will create a metamodel for predicting loops. This is a binary response.

Before metamodeling, we should compute some basic summary statistics, such as those shown in Tables 8-1 and 8-2. These statistics establish a baseline predictive ability for any model we construct; if a model cannot beat a simple classifier of "predict the most common label for that group," the model should not be used. In the training data, 38 percent of flights had a looping flight path. Further breaking down the data from Table 8-1, 40 percent of dart plane flights had a loop, 25 percent of classic plane flights had a loop, and 48 percent of WRPA flights had a loop. We further disaggregate the data, as shown in Table 8-2, by splitting the continuous factors, initial airspeed and angle, into high and low categories and similarly computing proportions. That level of disaggregation, though, is harder to comprehend and thus less useful for establishing a baseline metamodel performance expectation.

Table 8-1. Within-Group Proportions of Loops in Data Subsets by Model Type

Design	Loop Probability	Sample Size
Dart	0.40	100
Classic	0.25	100
WRPA	0.48	100

Table 8-2. Within-Group Proportions of Loops in Data Subsets by Factor

Design	Airspeed Group[a]	Angle Group[b]	Loop Probability	Sample Size
Dart	Low	Low	0	24
Dart	Low	High	0.0769	26
Dart	High	Low	0.654	26
Dart	High	High	0.875	24
Classic	Low	Low	0	23
Classic	Low	High	0.037	27
Classic	High	Low	0.259	27
Classic	High	High	0.739	23
WRPA	Low	Low	0.037	27
WRPA	Low	High	0.087	23
WRPA	High	Low	0.826	23
WRPA	High	High	0.963	27

[a] Airspeed split into low and high based on being below or above 6.5 meters per second.

[b] Angle split into low and high based on being below or above 0 radians.

This time we will use **mgcv** directly for metamodel fitting so we can have more control. We considered 30 models for fitting, varying the following choices we could make:

- The functional form of the additive model. All candidate forms are listed in Table 8-3.
- The distribution the outputs are assumed to follow. While at first glance the data are clearly binary (since we either do or do not see a loop) and should be modeled with the binomial distribution, logistic regression-type procedures like those considered here can handle a phenomenon known as *overdispersion* using the quasibinomial distribution. See McCullagh and Nelder (1989) for more information on overdispersion and why it needs to be accounted for. Hence, in addition to modeling the response with the binomial distribution, we consider the quasibinomial distribution.

- The link function used, either the logistic, probit (inverse standard normal cumulative distribution function), or complementary log-log (CLL) link function. Formulas for these link functions are given in Table 8-4.

Table 8-3. Model Forms Considered for Loops in Paper Plane Flights

Description	Functional Form[a]
Smooth Main Effects	$\eta\big(P(\text{loop}_i)\big) = \beta_0 + \beta_{\text{cl}}\text{classic}_i + \beta_{\text{wr}}\text{wrpa}_i + f_{\text{as}}(\text{airspeed}_i) + f_{\text{an}}(\text{angle}_i)$
Smooth Interaction	$\eta\big(P(\text{loop}_i)\big) = \beta_0 + \beta_{\text{cl}}\text{classic}_i + \beta_{\text{wr}}\text{wrpa}_i + f_{\text{as,an}}(\text{airspeed}_i, \text{angle}_i)$
Plane Design-Specific Smooth Terms	$\eta\big(P(\text{loop}_i)\big) = \beta_0 + f_{\text{as}}(\text{airspeed}_i) + f_{\text{an}}(\text{angle}_i)$ $+ \Big(f_{\text{as,cl}}(\text{airspeed}_i) + f_{\text{an,cl}}(\text{angle}_i)\Big)\,\text{classic}_i$ $+ \Big(f_{\text{as,wr}}(\text{airspeed}_i) + f_{\text{an,wr}}(\text{angle}_i)\Big)\,\text{wrpa}_i$
Smooth Saturated	$\eta\big(P(\text{loop}_i)\big) = \beta_0 + f_{\text{as,an}}(\text{airspeed}_i, \text{angle}_i) + f_{\text{as,an,cl}}(\text{airspeed}_i, \text{angle}_i)\text{classic}_i$ $+ f_{\text{as,an,wr}}(\text{airspeed}_i, \text{angle}_i)\text{wrpa}_i$
Fully Linear Main Effects	$\eta\big(P(\text{loop}_i)\big) = \beta_0 + \beta_{\text{cl}}\text{classic}_i + \beta_{\text{wr}}\text{wrpa}_i + \beta_{\text{as}}\text{airspeed}_i + \beta_{\text{an}}\text{angle}_i$

[a] η is the link function, in this example either the logistic, probit, or CLL link function. $P(\text{loop}_i)$ is the probability of a loop for observation i. Coefficients for linear terms are denoted with β. Smooth functions are denoted with f. Subscripts for coefficients or smooth terms indicate what factors that coefficient or smooth term are for and whether it is an interaction term. classic_i and wrpa_i are indicator variables, meaning they equal 1 if observation i is a plane of that design and equal 0 otherwise. airspeed_i and angle_i are numeric and equal to the value of the initial airspeed or initial angle of the plane for that observation.

Table 8-4. Link Functions for Regression on Binary Data

Name	Formula
Logistic[a]	$\eta(p) = \log\left(\frac{p}{1-p}\right)$ [b]
Complementary Log-Log (CLL)	$\eta(p) = \log(-\log(1-p))$
Probit	$\eta(p) = \Phi^{-1}(p)$ [c]

[a] The logistic link function is the canonical link function for the binomial distribution. See McCullagh and Nelder (1989) for more information on the significance of canonical link functions.

[b] $\log$ is the natural logarithm, or $\log(\text{e}) = 1$.

[c] Φ is the cumulative distribution function of the normal distribution function.

We fit a GAM for each combination of distribution, link function, and functional form. We estimate smooth functions using univariate thin-plate splines for air speed and angle. **mgcv** can use n-CV to choose the smoothing parameter, though we use here only our preference, the restricted maximum likelihood (REML) approach. After fitting, we compute the in-sample BIC and the out-of-sample Brier score, accuracy, recall, and precision for each metamodel, using the screening output set as the out-of-sample outputs. Not all metamodels are easily estimable; hence, an easy way to cut down our options is to remove any metamodel for which we could not compute a BIC. We then investigate the three estimated metamodels with the best BIC, Brier, and accuracy score among those estimated. We see the final metamodels and their associated metrics in Table 8-5.

Table 8-5. Evaluation Metrics for Model Candidates Fitting Flight Loops

Formula	Distribution	Link	BIC	OOS Brier	OOS Accuracy	OOS Recall (No Loop)	OOS Recall (Loop)	OOS Precision (No Loop)	OOS Precision (Loop)
Smooth Main Effects	Binomial	Logit	163.516	0.063	0.92	0.97	0.86	0.90	0.95
Smooth Main Effects	Binomial	Probit	171.206	0.061	0.90	0.95	0.83	0.88	0.93
Smooth Main Effects	Binomial	CLL	148.437	0.067	0.90	0.97	0.81	0.87	0.95

Note: OOS means out of sample.

The metamodel the BIC recommends is a GAM using a binomial distribution as the response distribution, a CLL link function, and smooth main effects with no interactions. The other metrics of interest—accuracy, precision, and recall—look good for this metamodel. The three metamodels proposed differ only in the chosen link function, with functional form and distribution being the same. We will select the metamodel with the lowest BIC for further refinement.

While the evaluation metrics are high for our chosen metamodel, plotting the smooth terms in the metamodel reveals opportunities to improve it. In Figure 8-14, we see the function linking initial airspeed to the response is almost perfectly linear. This linearity suggests we may want to change this term in our model from an arbitrary smooth function to a linear effect. We will keep all other parameters suggested by our metamodel selection procedures. This change makes the metamodel even simpler. Note that this implies the final functional form used is not listed in Table 8-3; it is

$$\eta\left(P\left(\text{loop}_i\right)\right) = \beta_0 + \beta_{\text{cl}}\text{classic}_i + \beta_{\text{wr}}\text{wrpa}_i + \beta_{\text{an}}\text{angle}_i + f_{\text{as}}\left(\text{airspeed}_i\right),$$

where η is the CLL function.

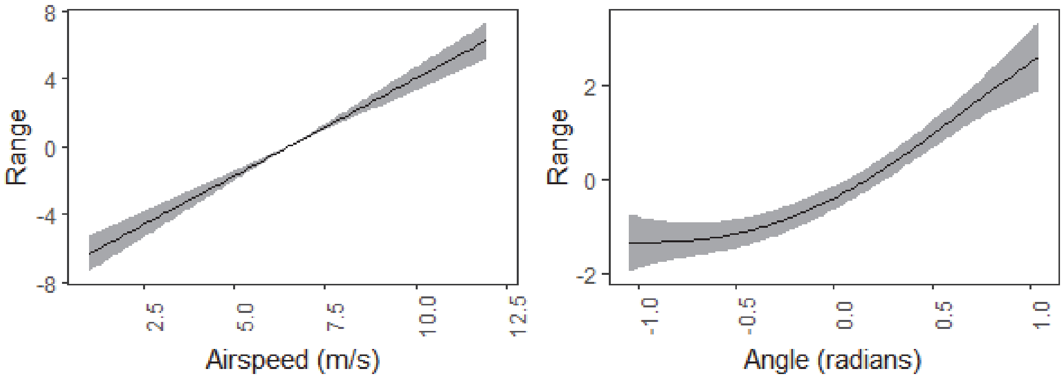

Figure 8-14. Estimated Response Functions for the GAM Predicting Paper Plane Probability of Looping

The final model is summarized by **mgcv**. Table 8-6 reports model results. Figure 8-15 shows the loop prediction regions.

Table 8-6. Metamodel Parameter Estimates and Metrics for the Presence of a Loop in a Paper Plane Flight Path

Parametric Coefficients	Estimate	Std. Error[a]	t-value	p-value
$\boldsymbol{\beta}_{\mathbf{0}}$	−9.1928	1.2070	−7.6164	< 0.0001
$\boldsymbol{\beta}_{\mathbf{cl}}$	−1.9537	0.4641	−4.2096	< 0.0001
$\boldsymbol{\beta}_{\mathbf{wr}}$	1.1033	0 .4171	2.6450	0.0082
$\boldsymbol{\beta}_{\mathbf{as}}$	1.1487	0.1491	7.7029	< 0.0001
Smooth Terms	**EDF[b]**	**Ref. DF[c]**	**F-value**	**p-value**
$\boldsymbol{f}_{\mathbf{as}}(\mathbf{airspeed})$	2.3447	2.9167	30.7587	< 0.0001
Model Metrics	**Dev. Exp.[d]**	**Adj. $\boldsymbol{R}^2$**	**−REML[e]**	**Scale**
	72.6%	0.775	30.7587	1

[a] The standard error of parameter estimates.

[b] Estimated degrees of freedom.

[c] Reference degrees of freedom.

[d] Deviance explained.

[e] Restricted maximum likelihood.

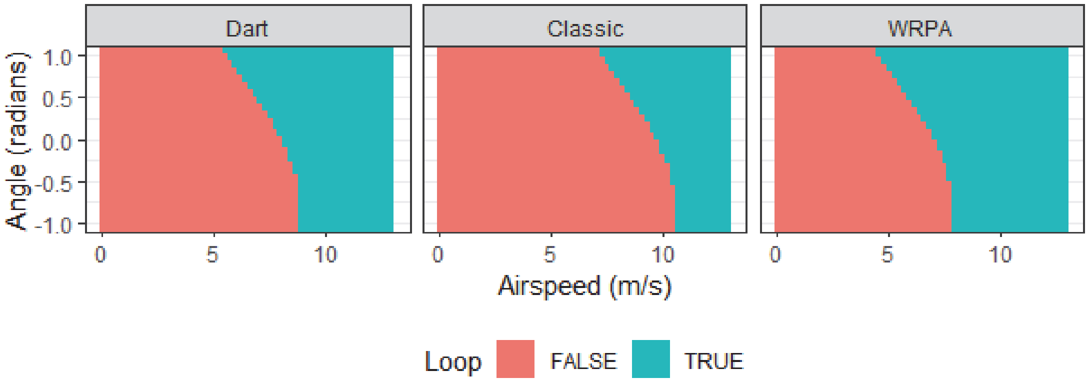

Figure 8-15. Prediction Regions for the GAM Modeling Probability of Looping

Unfortunately, the parameters of the chosen model are not easily interpreted when using a CLL link function, but the sign[22] of the coefficients can still tell us whether a loop is more or less likely under certain conditions. In this case, we learn that the classic planes are less likely overall to generate a loop and the world-record planes are most likely. Higher initial airspeed makes a loop more likely. Higher initial angles make loops more likely than with lower initial angles, but in a nonlinear way; loops become more likely the higher the angle is when the angle is above the horizon, but low angles seem to have a roughly constant probability of generating a loop.

The in-sample calibration curve in Figure 8-16 suggests the model predicts loops well. One would hope that would be the case for in-sample data. The calibration curve for the validation data, though, looks terrible, but is this due to the model actually being miscalibrated in the validation set or to bad smoothing in the calibration curve? A smooth fit via a different method[23] does not look as bad, as seen in Figure 8-17, but it is not perfect either. These issues could prompt further investigation into why the calibration curves do not look good. We do not do so here as those issues are out of scope for this paper.

[22] The sign of a number denotes whether the number is positive or negative.

[23] The second calibration curve fit in the validation data set uses a locally estimated scatterplot smoothing (LOESS) smoother from the R package **gam**. This smoother is fit via different methods than those described in this paper, based on a moving average-type procedure rather than penalized regression. This smoother generates a calibration curve that more closely resembles that of Haman and Johnson (2022), and it fits into the original GAM fitting procedures proposed by Hastie and Tibshirani (1986).

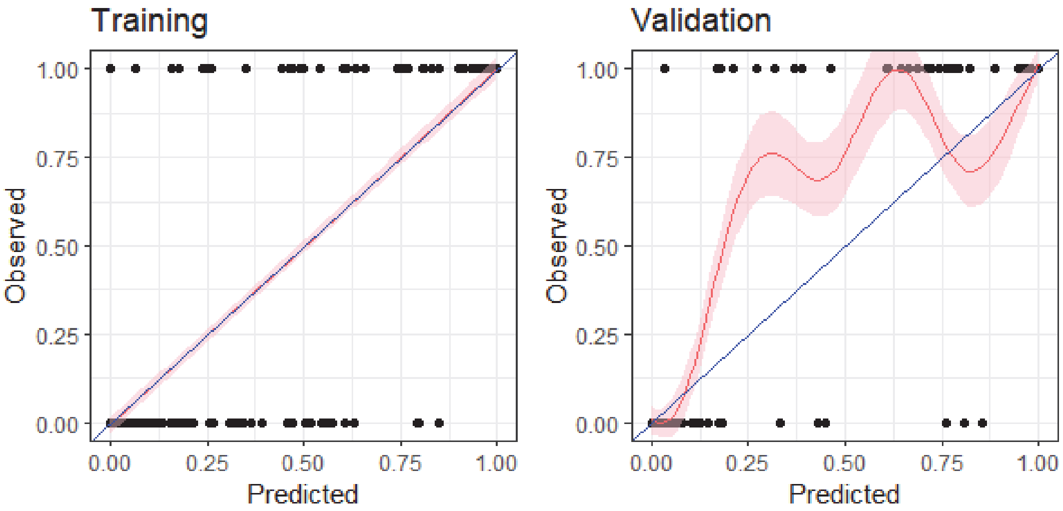

Figure 8-16. Calibration plots of the selected loop model in both the training and screening output sets. The smoothed calibration curve obtained for the validation set seems implausible.

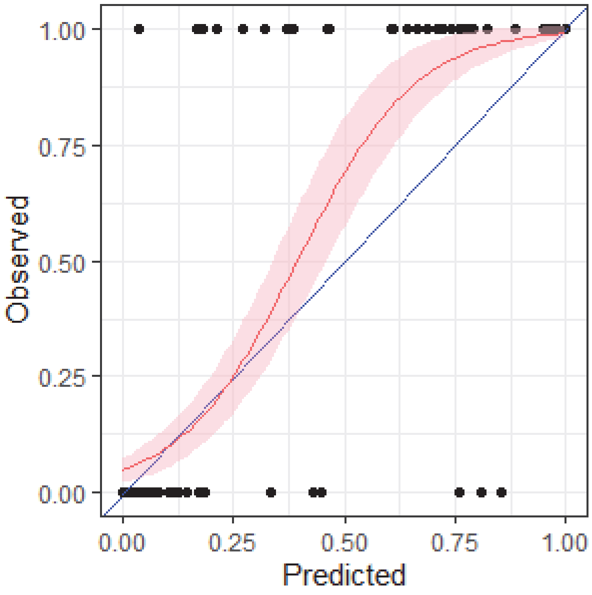

Figure 8-17. Calibration Curve Estimated via a Different Smoothing Technique, Locally Estimated Scatterplot Smoothing (LOESS)

Having decided on our model, we evaluate its performance in the evaluation set. Fortunately, there is no strong evidence of degradation; we observe an accuracy of 89 percent, a recall rate of 92 percent for no loops and 83 percent for loops, a precision of 89 percent for no loops and 88 percent for loops, and a Brier score of 0.11. If we had predicted simply the most common outcome observed (no bump), we would be correct 62 percent of the time overall, and Table 8-1 suggests our predictive capability would max at 75 percent. This suggests our model was not a waste of effort and that it improved predictive performance over a simple rule of predicting the most common outcome. The calibration curve given in Figure 8-18 looks good. This metamodel appears to be an effective summary of the M&S environment.

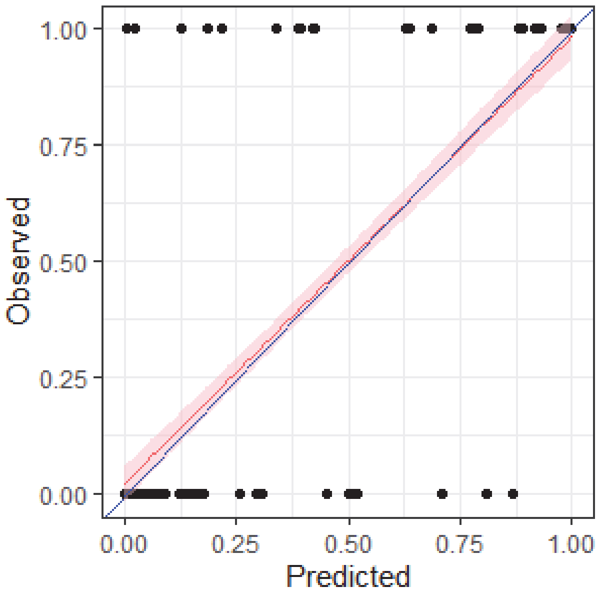

Figure 8-18. Calibration Plot for the Loop Model in the Test Data Set

3. Example GAM for Predicting the Number of Bumps in a Paper Plane Path (Count Data Response)

Our last metamodel treats bumps as the response variable. The responses are counts and thus different distributional families are involved in modeling.

We explore the following GAM metamodeling options in this example:

- Which distribution to treat as describing the count data, including Poisson, quasi-Poisson (which is Poisson but allows for overdispersion), ZIP (which allows the probability of no bumps to be greater than specified by an unadjusted Poisson distribution), and negative binomial. We will not vary link function use and instead will use the default link function for each candidate response distribution.
- Use of GCV or REML for smoothing parameter selection.
- The functional form of the additive model. All candidate functional forms are listed in Table 8-7.

Table 8-6. Model Forms Considered for Bump Counts in Paper Plane Flights

Description	Functional Form[a]
Smooth Main Effects	$\eta\big(E(\mathrm{bump}_i)\big) = \beta_0 + \beta_{\mathrm{cl}}\mathrm{classic}_i + \beta_{\mathrm{wr}}\mathrm{wrpa}_i + f_{\mathrm{as}}(\mathrm{airspeed}_i) + f_{\mathrm{an}}(\mathrm{angle}_i)$
Smooth Interaction	$\eta\big(E(\mathrm{bump}_i)\big) = \beta_0 + \beta_{\mathrm{cl}}\mathrm{classic}_i + \beta_{\mathrm{wr}}\mathrm{wrpa}_i + f_{\mathrm{as,an}}(\mathrm{airspeed}_i, \mathrm{angle}_i)$
Plane Design-Specific Smooth Terms	$\eta\big(E(\mathrm{bump}_i)\big) = \beta_0 + f_{\mathrm{as}}(\mathrm{airspeed}_i) + f_{\mathrm{an}}(\mathrm{angle}_i)$ $+ \Big(\mathrm{f}_{\mathrm{as,cl}}(\mathrm{airspeed}_i) + \mathrm{f}_{\mathrm{an,cl}}(\mathrm{angle}_i)\Big)\,\mathrm{classic}_i$ $+ \Big(\mathrm{f}_{\mathrm{as,wr}}(\mathrm{airspeed}_i) + \mathrm{f}_{\mathrm{an,wr}}(\mathrm{angle}_i)\Big)\,\mathrm{wrpa}_i$
Smooth Saturated	$\eta\big(E(\mathrm{bump}_i)\big) = \beta_0 + f_{\mathrm{as,an}}(\mathrm{airspeed}_i, \mathrm{angle}_i) + f_{\mathrm{as,an,cl}}(\mathrm{airspeed}_i, \mathrm{angle}_i)\mathrm{classic}_i$ $+ f_{\mathrm{as,an,wr}}(\mathrm{airspeed}_i, \mathrm{angle}_i)\mathrm{wrpa}_i$
Fully Linear Main Effects	$\eta\big(E(\mathrm{bump}_i)\big) = \beta_0 + \beta_{\mathrm{cl}}\mathrm{classic}_i + \beta_{\mathrm{wr}}\mathrm{wrpa}_i + \beta_{\mathrm{as}}\mathrm{airspeed}_i + \beta_{\mathrm{an}}\mathrm{angle}_i$
Saturated Linear Model	$\eta\big(E(\mathrm{bump}_i)\big) = \beta_0 + \beta_{\mathrm{cl}}\mathrm{classic}_i + \beta_{\mathrm{wr}}\mathrm{wrpa}_i + \beta_{\mathrm{as}}\mathrm{airspeed}_i + \beta_{\mathrm{as,cl}}\mathrm{airspeed}_i{\times}\mathrm{classic}_i$ $+ \beta_{\mathrm{as,wr}}\mathrm{airspeed}_i{\times}\mathrm{wrpa}_i + \beta_{\mathrm{an}}\mathrm{angle}_i + \beta_{\mathrm{an,cl}}\mathrm{angle}_i{\times}\mathrm{classic}_i$ $+ \beta_{\mathrm{an,wr}}\mathrm{angle}_i{\times}\mathrm{wrpa}_i + \beta_{\mathrm{as,an}}\mathrm{airspeed}_i{\times}\mathrm{angle}_i$
Linear Model with Airspeed-Angle Interaction	$\eta\big(E(\mathrm{bump}_i)\big) = \beta_0 + \beta_{\mathrm{cl}}\mathrm{classic}_i + \beta_{\mathrm{wr}}\mathrm{wrpa}_i + \beta_{\mathrm{as}}\mathrm{airspeed}_i + \beta_{\mathrm{an}}\mathrm{angle}_i$ $+ \beta_{\mathrm{as,an}}\mathrm{airspeed}_i{\times}\mathrm{angle}_i$

[a] η is the link function, depending on the distribution used for describing the response. $E(\mathrm{bump}_i)$ is the expected number of bumps for observation i. Coefficients for linear terms are denoted with β. Smooth functions are denoted with f. Subscripts for coefficients or smooth terms indicate what factors that coefficient or smooth term are for and whether it is an interaction term. $\mathrm{classic}_i$ and wrpa_i are indicator variables, meaning they equal 1 if observation i is a plane of that design and equal 0 otherwise. $\mathrm{airspeed}_i$ and angle_i are numeric and equal to the value of the initial airspeed or initial angle of the plane for that observation.

We did not use K-CV in our loop prediction example, but we involve it more here. CV will generate a distribution of performance metrics representing out-of-sample performance rather than a single number; this means that the evaluation of a metamodeling approach will be based on a summary of the CV results, and we can study the resulting distributions to obtain a better sense of how much uncertainty there is in selecting a metamodeling approach.

We first establish a baseline understanding of the M&S output set with summary statistics. We look at the mean number of bumps and the standard deviation of the bump count; for the response variable we are considering, these are good metric choices. We present the summary statistics in Tables 8-8 and 8-9. The classic plane design has the lowest average number of bumps, at 1.27 bumps in a flight path; the dart design comes in second with 1.8 bumps; and the WRPA design has the most, with 2.56 bumps. The standard deviations for these designs are 0.75, 0.866, and 1.15 bumps, respectively. Of these statistics, the standard deviation is the more noteworthy, since if the RMSE exceeds the standard deviation, the model likely makes terrible predictions.

Table 8-7. Within-Group Bump Count Metrics in Data Subsets by Model Type

Design	Mean Bumps	Bump Std. Dev.	Sample Size
Dart	1.76	0.866	100
Classic	1.27	0.75	100
WRPA	2.56	1.15	100

Table 8-8. Within-Group Bump Count Metrics in Data Subsets by Factor

Design	Airspeed Group[a]	Angle Group[b]	Mean Bumps	Bump Std. Dev.	Sample Size
Dart	Low	Low	0.708	0.55	24
Dart	Low	High	1.62	0.571	26
Dart	High	Low	2.15	0.543	26
Dart	High	High	2.54	0.509	24
Classic	Low	Low	0.435	0.59	23
Classic	Low	High	1.07	0.616	27
Classic	High	Low	1.74	0.447	27
Classic	High	High	1.78	0.422	23
WRPA	Low	Low	1.63	1.28	27
WRPA	Low	High	2.09	0.848	23
WRPA	High	Low	3.09	0.668	23
WRPA	High	High	3.44	0.506	27

[a] Airspeed split into low and high based on being below or above 6.5 meters per second.

[b] Angle split into low and high based on being below or above 0 radians.

We use 10 folds for CV (abbreviated as 10-CV). These folds are unchanging throughout the fits. With 10 folds, we can consider not only which metamodel seems to do best according to 10-CV metrics but also how much the metrics we study vary.

After fitting all the candidates, we will narrow down the metamodels to those with the smallest median BIC, smallest median out-of-sample MSE, and smallest median out-of-sample MAE. We will also consider the distribution of BIC, MSE, and MAE in the 10-CV samples when making our selection to assess metric variability. A plot easily facilitates such assessments. Table 8-10 presents candidate models and their associated metrics. Figure 8-19 visualizes the distribution of these metrics in the 10-CV samples.

Table 8-9. Candidate Generalized Additive Models for the Bump Count Data with Goodness-of-Fit Metrics

Formula	Distribution	Smoothing Penalty Method	BIC	OOS RMSE[a]	OOS MAE
Smooth Interaction	Poisson	REML	747.893	0.526	0.193
Smoothed Interaction	ZIP	GCV-Cp[b]	702.299	0.540	0.187
Smoothed Interaction	ZIP	REML	702.299	0.540	0.187
Linear Model with Airspeed-Angle Interaction	ZIP	GCV-Cp	684.814	0.547	0.194
Linear Model with Airspeed-Angle Interaction	ZIP	REML	684.814	0.547	0.194

[a] Out-of-sample root mean-squared error.

[b] Generalized cross-validation with Mallows C_p.

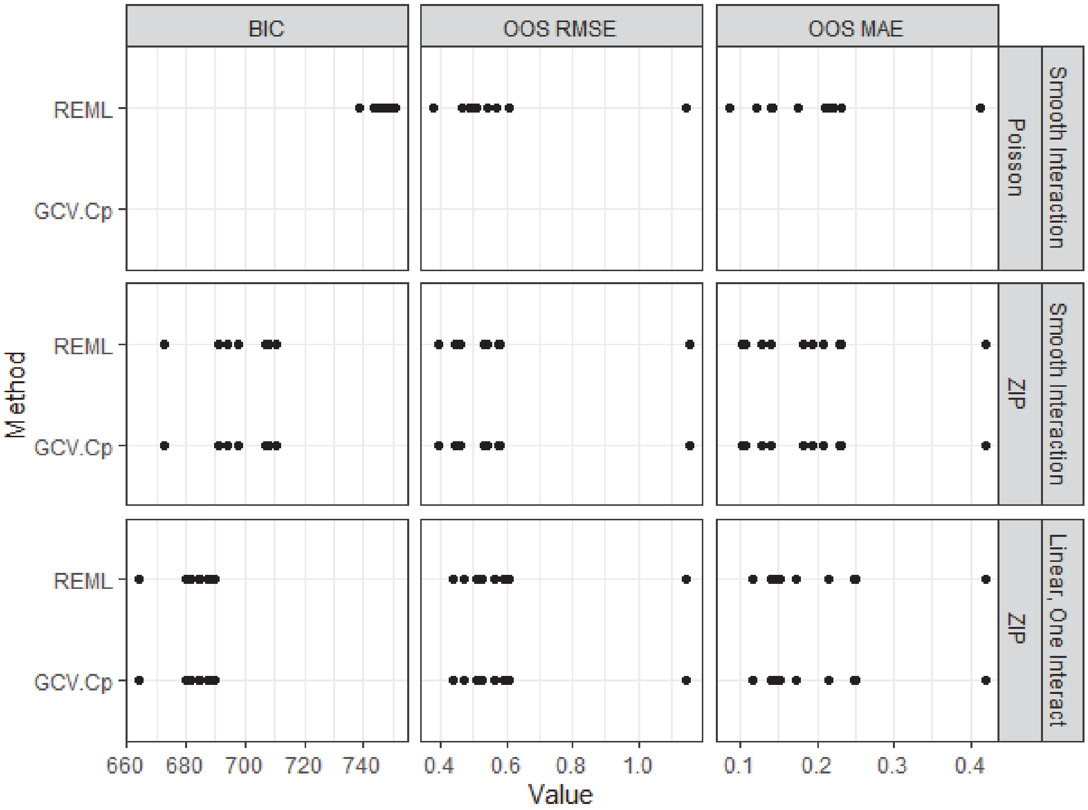

Figure 8-19. Metric values observed in 10-CV folds. OOS means out of sample, and ZIP means zero-inflated Poisson.

We chose five metamodels for the final candidates, but four are redundant. The plots reveal that REML and 10-CV generated identical metamodels, since the 10-CV metric distributions are identical. There was one exception, where only the model obtained via the REML smoothing parameter selection appeared as a candidate (when the distribution family is Poisson).

After we eliminate the duplicate metamodels, we see that two of the three final metamodels use the ZIP distribution as the distribution of the data. Of the three metrics, BIC seems the most decisive when one considers the variation in the metrics across the 10-CV samples. RMSE and MAE have very similar distributions across metamodels, while BIC is clearly smaller for the linear metamodel that interacts airspeed and angle but treats the paper plane design as a simple additive effect; it is the simplest model of the three candidate metamodels. Hence, it seems that BIC prefers the linear model on the basis of its simplicity, as the smoothing models (which are smooth surfaces, interacting airspeed and angle) lack convincingly better predictive performance than the simple linear model. Furthermore, the RMSE is smaller than the standard deviations in our earlier baseline estimates, suggesting that the model has some predictive ability.

Hence, the 10-CV results suggest a generalized linear model interacting airspeed and angle. We fit the final metamodel using the full training sample instead of the 10-CV samples. As an additional check before seeing our results on the evaluation set, we check the metamodel's predictive performance in the screening output set. If we see significant degradation in the RMSE and MAE, we could have an overfitted metamodel that will not generalize well.

Figure 8-20 presents diagnostic plots we use to further assess metamodel performance. These diagnostic plots do not suggest we have obtained a good fit; the Q-Q plot and the residual histogram, in particular, suggest that the distribution of the response variable is not correct. Such pathologies would be reason to further explore our metamodeling options. We will not do so now and will ignore these warning signs, since attempting a fix is outside the scope of this paper. Metrics relating to the metamodel's predictive ability look better. The metamodel explains 75.9 percent of the deviance in the training data. The in-sample RMSE is 0.6 bumps and the in-sample MAE is 0.4 bumps. In the screening output set, the metamodel's RMSE is 0.5 bumps and the MAE is 0.4. These numbers do not suggest there is any overfitting. Table 8-11 gives more information about the fit.

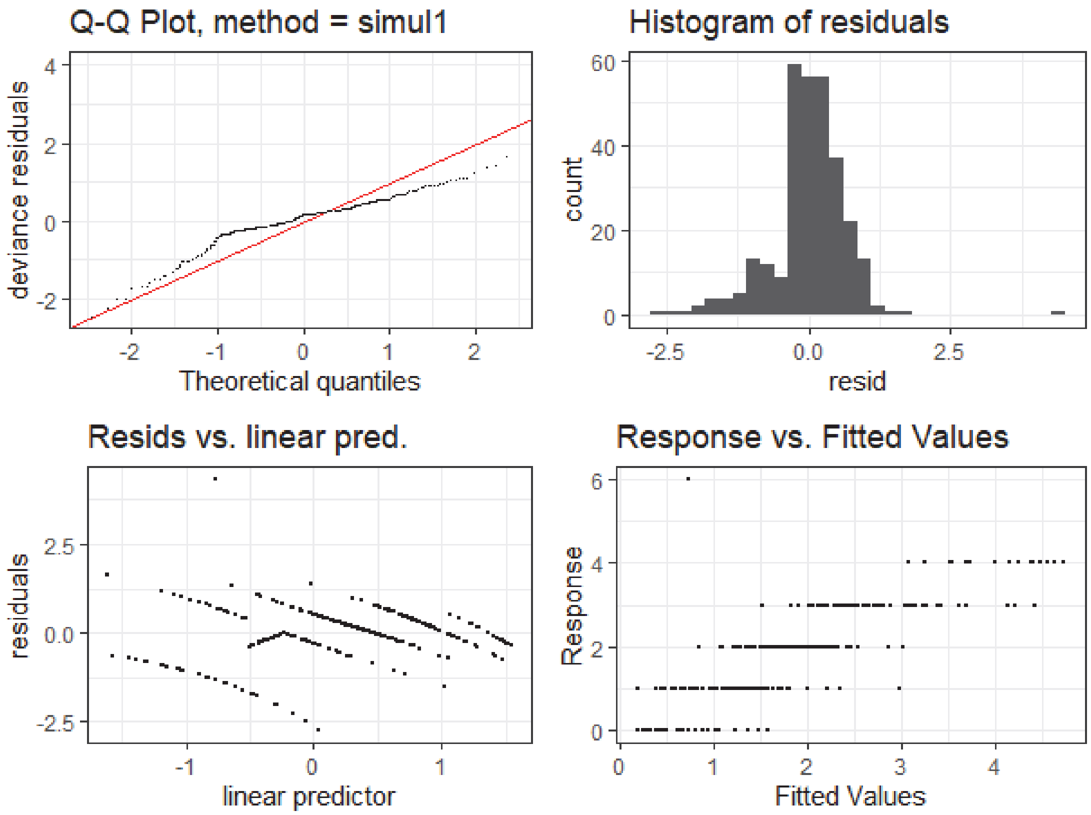

Figure 8-20. Diagnostic Plots of the Zero-Inflated Poisson Model for the Number of Bumps in a Paper Plane Flight Path

Table 8-11. Metamodel Parameter Estimates and Metrics for the Number of Bumps in a Paper Plane Flight Path

Parametric Coefficients	Estimate	Std. Error	t-value	p-value
$\boldsymbol{\beta}_0$	−0.9306	0.1120	−8.3052	< 0.0001
$\boldsymbol{\beta}_{\text{cl}}$	−0.3937	0.1065	−3.6956	0.0002
$\boldsymbol{\beta}_{\text{wr}}$	0.4370	0.1009	4.3320	< 0.0001
$\boldsymbol{\beta}_{\text{as}}$	0.1711	0.0129	13.2789	< 0.0001
$\boldsymbol{\beta}_{\text{an}}$	0.7700	0.1627	4.7315	< 0.0001
$\boldsymbol{\beta}_{\text{as,an}}$	−0.0654	0.0211	−3.1038	0.0019
Model Metrics	**Deviance Explained**	**ZIP Intercept**	**ZIP Slope**	**ZIP *b***
	75.9%	1.246	0.587	0

The selected distribution of the response variable uses the identity link function, which means that we can directly interpret the parameters of the linear metamodel. However, we should still exercise caution since the mean of the response variable should still be positive, and a naïve interpretation of predictions could lead us to infer negative means.

The classic plane design has, on average, 0.4 fewer bumps than the dart model (the baseline), while the WRPA design has 0.4 more bumps on average. The effects of the initial airspeed and initial angle are harder to interpret because of the interaction term. It is small relative to the angle effect and thus we can say that, within the range of the data we observed, for every radian increase in initial angle, a plane sees, on average, 0.8 more bumps.[24] The magnitude of the airspeed effect resembles the magnitude of the interaction term's effect, making such interpretations harder, but in general we see, on average, more bumps with higher airspeeds. The interaction term seems to mostly dampen what otherwise is an increasing number of bumps as we increase both initial airspeed and initial angle. The plots of the surface and predicted values given in Figure 8-21 show the increasing number of predicted bumps as we increase both initial angle and initial airspeed.

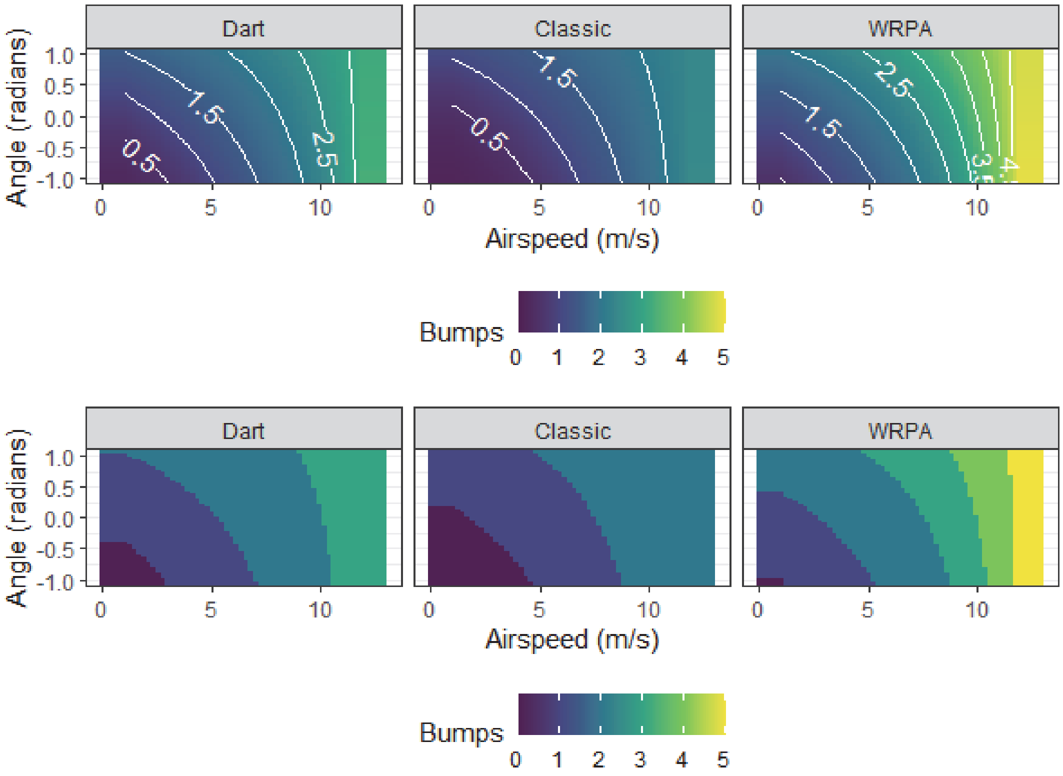

Figure 8-21. Prediction surface for the number of bumps in a paper plane flight path estimated by the zero-inflated generalized linear model. The top plots do not have the predictions rounded (allowing for decimal number predictions, interpreted as the average number of bumps in a flight path), while the bottom plots are rounded to the nearest integer.

Calibration curves in the training and screening samples, shown in Figure 8-22, suggest the model is reasonably well-calibrated. The corresponding ANOVA statistical

[24] In degrees, that is 0.01 bumps per degree.

tests would disagree (with *p*-values near 0) and suggest a nonidentity relationship between observed and predicted values; the plots suggest that if there is miscalibration, it would be near the ends of the observed range, with the predicted number of bumps perhaps being off by half a bump or more.

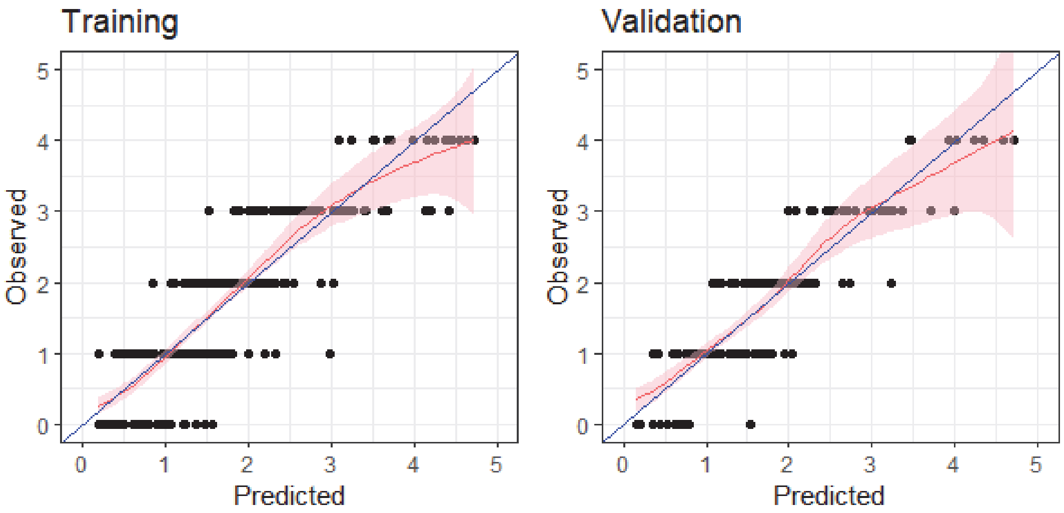

Figure 8-22. Calibration Curves for the Bump Model, Both in the Training Sample and the Validation Sample

That said, the metamodel seems to predict outcomes somewhat well, so we check the model's performance in the evaluation set. There, we observe an RMSE of 0.5 bumps and an MAE of 0.4 bumps. This is lower than the observed variation in the M&S observations and variation in the subsets shown in Table 8-9. The calibration curve, seen in Figure 8-23, is similar to what was seen in the validation sample, which we deemed acceptable before and which again suggests the metamodel's predictive abilities have not changed when moving to out-of-sample outputs. These checks all bode well for the final metamodel's predictive capabilities.

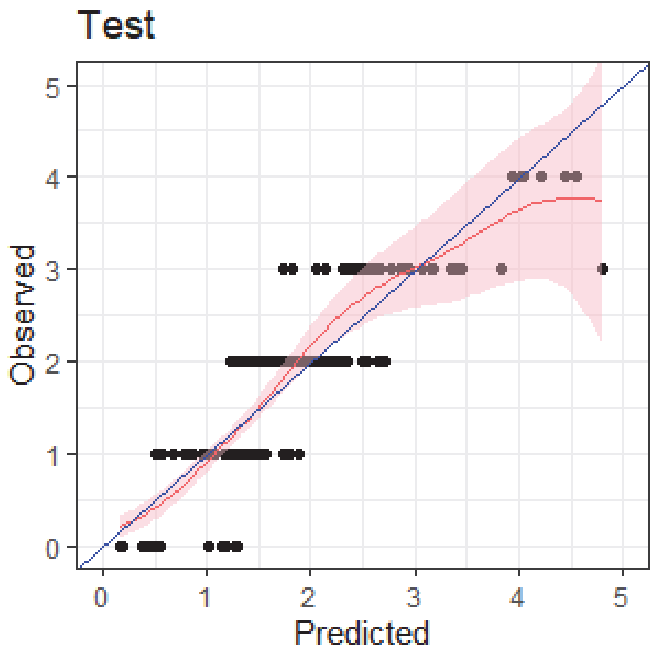

Figure 8-23. Calibration Curves for the Bump Model in the Evaluation Output Set

9. Conclusions

Metamodeling is a mature methodology for analyzing M&S environments and their outputs, one that provides rich information about the system's performance. The metamodel can provide information about M&S characteristics by describing the M&S environment's behavior. It can also be a key component in solving other problems, such as finding optimal settings under which the modeled system performs, calibrating the M&S environment's settings to best mimic real-world behavior, or inferring from observed data the values of important parameters. It may be a useful product in and of itself by replacing the M&S from which it was trained in contexts where speed or understandability matter, such as exercises, wargames, other M&S environments, and perhaps even real tactical decisions.

Metamodeling resembles other statistical modeling and prediction activities. We cannot claim to have exhaustively described all the modeling techniques that could be used because that would fill books; Hastie, Tibshirani, and Friedman (2009) offer many models, including those discussed here, that one could apply. Even the two methods emphasized in this paper—GPs and GAMs—are still actively researched and have many extensions; see Gramacy (2020) for more information about GPs and Wood (2017) for more information about GAMs. Hence, this paper is not the end point but a useful starting point and standard from which build metamodels.

Our examples also give standards by which to judge models. No single process was applied as a recipe. The statistical fitting process generally never is—and never should be—purely algorithmic, without any intervention by a person; to quote DOT&E (2017), "There is no cookbook approach." The result of unthinking analysis is "cargo-cult statistics" (Stark and Saltelli 2018), the imitation of statistical procedures and methods without understanding why to apply them, what the results mean, and under what circumstances to avoid them. Statisticians are not algorithmic when analyzing data, looking at lots of different metrics and strategies and considering the story each one tells and their individual weaknesses, which then inform the statistician's overall opinion. One should read the examples not just to see the procedures used but also to appreciate the story of how one goes from an M&S ready for analysis to a final metamodel with reasonable descriptive and predictive abilities. We always encourage a thoughtful approach.

References

Abramowitz, Milton, and Irene A. Stegun, eds. 1972. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. 9th ed. New York, NY: Dover.

Anderssen, R. S., and Peter Bloomfield. 1974. “A Time Series Approach to Numerical Differentiation.” *Technometrics* 16, no. 1 (February): 69–75. https://doi.org/10.2307/1267494.

Bernardo, José M., and Adrian F. M. Smith. 1994. *Bayesian Theory*. West Sussex, England: Wiley.

Commander, Operational Test and Evaluation Force. 2022. “Use of Modeling and Simulation in Operational Test.” OPTEVFOR Instruction 5000.1D. Norfolk, VA: U.S. Navy.

Craven, Peter, and Grace Wahba. 1979. “Smoothing Noisy Data with Spline Functions.” *Numerische Mathematik* 31 (December): 377–403.

de Boor, Carl. 1978. *A Practical Guide to Splines*. New York, NY: Springer.

Defense Modeling and Simulation Enterprise. Last updated September 15, 2020. “M&S Glossary.” https://www.msco.mil/MSReferences/Glossary/MSGlossary.aspx.

Derksen, Shelley, and H. J. Keselman. 1992. “Backward, Forward and Stepwise Automated Subset Selection Algorithms: Frequency of Obtaining Authentic and Noise Variables.” *British Journal of Mathematical and Statistical Psychology* 45, no. 2 (November): 265–282. https://doi.org/10.1111/j.2044-8317.1992.tb00992.x.

DOT&E (Director, Operational Test and Evaluation). 2016. “Guidance on the Validation of Models and Simulation Used in Operational Test and Live Fire Assessments.” Washington, DC: DOT&E. https://www.dote.osd.mil/Portals/97/pub/policies/2016/20140314_Guidance_on_Valid_of_Mod_Sim_used_in_OT_and_LF_Assess_(10601).pdf?ver=2019-08-19-144201-107.

DOT&E (Director, Operational Test and Evaluation). 2017. “Clarifications on Guidance on the Validation of Models and Simulation Used in Operational Test and Live Fire Assessments.” Washington, DC: DOT&E. https://www.dote.osd.mil/Portals/97/pub/policies/2017/20170117_Clarification_on_Guidance_on_the_Validation_of_ModSim_used_in_OT_and_LF_Assess(15520).pdf?ver=2019-08-19-144121-1237.

Duchon, Jean. 1977. “Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces.” Chap 7 in *Constructive Theory of Functions of Several Variables*, edited by Walter Schempp and Karl Zeller, 85–100. Berlin, Germany: Springer.

Eilers, Paul H. C., and Brian D. Marx. 1996. “Flexible Smoothing with B-Splines and Penalties.” *Statistical Science* 11, no. 2 (May): 89–121. https://doi.org/10.1214/ss/1038425655.

Fasiolo, Matteo, Raphaël Nedellec, Yannig Goude, and Simon N. Wood. 2018. “Scalable Visualisation Methods for Modern Generalized Additive Models.” *Arxiv Preprint*. https://arxiv.org/abs/1809.10632.

Gramacy, Robert B. 2020. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. Boca Raton, FL: CRC Press.

Haman, John T., Thomas H. Johnson, David Grimm, Kerry Walzl, and Lindsey Butler. 2022. *Predicted Probabilities Validation*. Alexandria, VA: Institute for Defense Analyses.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. 2nd ed. New York, NY: Springer. http://www-stat.stanford.edu/ tibs/ElemStatLearn/.

Hurvich, Clifford M., and Chih-Ling Tsai. 1989. “Regression and Time Series Model Selection in Small Samples.” *Biometrika* 76, no. 2: 297–307.

Hurvich, Clifford M., and Chih-Ling Tsai. 1990. “The Impact of Model Selection on Inference in Linear Regression.” *The American Statistician* 44, no. 3 (August): 214–217. https://doi.org/10.2307/2685338.

Joseph, V. Roshan, Evren Gul, and Shan Ba. 2020. “Designing Computer Experiments with Multiple Types of Factors: The MaxPro Approach.” *Journal of Quality Technology* 52, no. 4: 343–354. https://doi.org/10.1080/00224065.2019.1611351.

Kang, Xiaoning, and Xinwei Deng. 2020. “Design and Analysis of Computer Experiments with Quantitative and Qualitative Inputs: A Selective Review.” *WIREs Data Mining and Knowledge Discovery* 10, e1358 (January): 1–9. https://doi.org/10.1002/widm.1358.

Kim, Young-Ju, and Chong Gu. 2004. “Smoothing Spline Gaussian Regression: More Scalable Computation via Efficient Approximation.” *Journal of the Royal Statistical Society Series B* 66, no. 2: 337–356.

Koenker, Roger. 2011. “Additive Models for Quantile Regression: Model Selection and Confidence Bands.” *Brazilian Journal of Probability and Statistics* 25, no. 3 (November): 239–262. https://doi.org/10.1214/10-BJPS131.

Kuhn, Max. Published August 9, 2022. “Caret: Classification and Regression Training.” https://CRAN.R-project.org/package=caret.

MacDonald, Blake, Pritam Ranjan, and Hugh Chipman. 2015. “GPfit: An R Package for Fitting a Gaussian Process Model to Deterministic Simulator Outputs.” *Journal of Statistical Software* 64, no. 12 (April): 1–23. https://doi.org/10.18637/jss.v064.i12.

Mallows, C. L. 1973. “Some Comments on c_p.” *Technometrics* 15, no. 4 (November): 661–675. https://doi.org/10.2307/1267380.

Mantel, Nathan. 1970. “Why Stepdown Procedures in Variable Selection.” *Technometrics* 12, no. 3 (August): 621–625. https://doi.org/10.2307/1267207.

Matheron, G. 1963. “Principles of Geostatistics.” *Economic Geology* 58, no. 8 (December): 1246–1266. http://dx.doi.org/10.2113/gsecongeo.58.8.1246.

McCullagh, P., and J. A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. London, UK: Chapman and Hall.

Müller, Hans-Georg, and Fang Yao. 2008. "Functional Additive Models." *Journal of the American Statistical Association* 103, no. 484 (December): 1534–1544. https://doi.org/10.1198/016214508000000751.

Pya, Natalya, and Simon N. Wood. 2015. "Shape Constrained Additive Models." *Statistics and Computing* 25, no. 3 (May): 543–559. https://doi.org/10.1007/s11222-013-9448-7.

Ramsay, James O., Giles Hooker, and Spencer Graves. 2009. *Functional Data Analysis with R and MATLAB*. New York, NY: Springer.

Reiss, Philip T., and R. Todd Ogden. 2009. "Smoothing Parameter Selection for a Class of Semiparametric Linear Models." *Journal of the Royal Statistical Society: Series B* 54, no. 3: 507–554. https://doi.org/10.1111/j.1467-9868.2008.00695.x.

Roecker, Ellen B. 1991. "Prediction Error and Its Estimation for Subset-Selected Models." *Technometrics* 33, no. 4 (November): 459–468.

Schoenberg, I. J. 1964. "Spline Functions and the Problem of Graduation." *Proceedings of the National Academy of Sciences* 52, No. 4 (October): 947–950. https://doi.org/10.1073/pnas.52.4.947.

Smith, Ralph C. 2014. *Uncertainty Quantification*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Soong, T. T. 1973. *Random Differential Equations in Science and Engineering*. London, UK: Academic Press.

Stark, Philip B., and Andrea Saltelli. 2018. "Cargo-Cult Statistics and Scientific Crisis." *Significance* 15, no. 4 (July): 40–43. https://doi.org/10.1111/j.1740-9713.2018.01174.x.

Stengel, Robert F. 2004. *Flight Dynamics*. Princeton, NJ: Princeton University Press.

Sugiura, Nariaki. 1978. "Further Analysis of the Data by Akaike's Information Criterion and the Finite Corrections." *Communications in Statistics – Theory and Methods* 7, no. 1: 13–26. https://doi.org/10.1080/03610927808827599.

Tibshirani, Robert. 1996. "Regression, Shrinkage, and Selection Via the LASSO." *Journal of the Royal Statistical Society, Series B* 58, no. 1: 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Wahba, Grace. 1985. "A Comparison of GCV and GML for Choosing the Smoothing Parameter in the Generalized Spline Smoothing Problem." *The Annals of Statistics* 13, no. 4 (December): 1378–1402.

White, Halbert. 1982. "Maximum Likelihood Estimation of Misspecified Models." *Econometrica* 50, no. 1 (January): 1–25. https://doi.org/10.2307/1912526.

Wojton, Heather, Kelly M. Avery, Laura J. Freeman, Samuel H. Parry, Gregory S. Whittier, Thomas H. Johnson, and Andrew C. Flack. 2019. *Handbook on Statistical Design and Analysis Techniques for Modeling and Simulation Validation*.

Alexandria, VA: Institute for Defense Analyses. https://testscience.org/wp-content/uploads/sites/16/formidable/20/Handbook-on-Statistical-Design-Analysis-Techniques-for-Modeling-Simulation-Validation-Reduced.pdf.

Wojton, Heather, Kelly Avery, Han Yi, and Curtis Miller. 2021. *Space-Filling Designs for Modeling and Simulation Validation*. Alexandria, VA: Institute for Defense Analyses. https://testscience.org/wp-content/uploads/sites/16/formidable/20/SFD_Literature_Review_Final.html.

Wood, Simon N. 2006. "Low-Rank Scale-Invariant Tensor Product Smooths for Generalized Additive Mixed Models." *Biometrics* 62, no. 4 (December): 1025–1036.

Wood, Simon N. 2008. "Fast Stable Direct Fitting and Smoothness Selection for Generalized Additive Models." *Journal of the Royal Statistical Society, Series B* 70, no. 3 (July): 495–518. https://doi.org/10.1111/j.1467-9868.2007.00646.x.

Wood, Simon N. 2017. *Generalized Additive Models: An Introduction with R*. 2nd ed. Boca Raton, FL: CRC Press.

Yee, Thomas W. 2015. *Vector Generalized Linear and Additive Models with an Implementation in R.* New York, NY: Springer.

Zhang, Xuezhou, Sarah Tan, Paul Koch, Yin Lou, Urszula Chajewska, and Rich Caruana. 2019. "Axiomatic Interpretability for Multiclass Additive Models." In *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 226–234. New York, NY: Association for Computing Machinery. https://doi.org/10.1145/3292500.3330898.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 09/30/2022	**2. REPORT TYPE** OED Draft	**3. DATES COVERED** *(From - To)* N/A

4. TITLE AND SUBTITLE Metamodeling Techniques for Verification and Validation of Modeling and Simulation Data	**5a. CONTRACT NUMBER** HQ0034-19-D-0001
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Curtis G. Miller John T. Haman	**5d. PROJECT NUMBER** BD-9-2299(90)
	5e. TASK NUMBER 229990
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Institute for Defense Analyses 730 East Glebe Road Alexandria, Virginia 22305	**8. PERFORMING ORGANIZATION REPORT NUMBER** P-33230 ▼ H 2022-000374 ▼
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Director, Operational Test and Evaluation 1700 Defense Pentagon Washington, DC 20301	**10. SPONSOR/MONITOR'S ACRONYM(S)** DOT&E
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
This OED Draft has not been approved by the sponsor for distribution and release. Reproduction or use of this material is not authorized without prior permission from the responsible IDA Division Director.

13. SUPPLEMENTARY NOTES

Project Leader: John T. Haman

14. ABSTRACT

Modeling and simulation (M&S) outputs help DOT&E assess the effectiveness, survivability, lethality, and suitability of systems. To use M&S outputs, DOT&E needs models and simulators to be verified and validated. The purpose of this paper is to improve the state of verification and validation by recommending and demonstrating a set of statistical techniques—metamodels, also called statistical emulators—to the M&S community. The paper expands on DOT&E's guidance about metamodel usage by creating methodological recommendations the M&S community could apply to its activities. For a deterministic, discrete response variable, we recommend a nearest neighbor or decision tree model. For a deterministic, continuous response variable, we recommend Gaussian process interpolation. For a stochastic response variable, we recommend a generalized additive model. We also present techniques that testers can use to assess the adequacy of their metamodels and conclude with an example.

15. SUBJECT TERMS

Gaussian Process Modeling; Generalized Additive Models; Modeling and Simulation (M&S) Validation; Paper Plane Simulation; Statistical Methods

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			John T. Haman
Unclassified	Unclassified	Unclassified	Unlimited	TDB	**19b. TELEPHONE NUMBER** *(Include area code)* 703-845-2132