Non-isometric Curve to Surface Matching with Incomplete Data for Functional Calibration

ARASH POURHABIB AND BALABHASKAR BALASUNDARAM*

August 7, 2015

Abstract

Calibration refers to the process of adjusting features of a computational model that are not observed in the physical process so that the model matches the real process. We propose a framework for calibration when the unobserved features, i.e. calibration parameters, do not assume a single value, but are functionally dependent on other inputs. We demonstrate that this problem is curve to surface matching where the matched curve does not possess the same length as the original curve. Therefore, we perform non-isometric matching of a curve to a surface. Since in practical applications we do not observe a continuous curve but a sample of data points, we use a graph-theoretic approach to solve this matching of incomplete data. We define a graph structure in which the nodes are selected from the incomplete surface and the weights of the edges are decided based on the response values of the curve and surface. We show that the problem of non-isometric incomplete curve to surface matching is a shortest path problem in a directed acyclic graph. We apply the proposed method, graph-theoretic non-isometric matching, to real and synthetic data and demonstrate that the proposed method improves the prediction accuracy in functional calibration.

Keywords: Matching, non-isometry, calibration, computer experiments, graph shortest path

1 Introduction

Calibration refers to the problem of specifying the values of some features, known as calibration parameters, in a computational model, which are unobservable in the associated physical system (Kennedy and O’Hagan, 2001; Han et al., 2009; Higdon et al., 2004; Tuo and Wu, 2014). Here, a computational model refers to a set of mathematical formulations and assumptions implemented through computer codes, and is intended to behave

*School of Industrial Engineering and Management, Oklahoma State University, \{arash.pourhabib,baski\}@okstate.edu
similar to the physical system, i.e. they both generate the same response for a given input variable. If in the physical system, a calibration parameter assumes a range of values as a function of other variables in the system, we call this problem functional calibration. Functional calibration is motivated by examples in nano-manufacturing (Pourhabib et al., 2015), resistance spot-welding (Bayarri et al., 2007), and bioenergy production (Kumar et al., 2009). For example, in Poly-vinyl Alcohol (PVA)-treated buckypaper fabrication, we are interested in understanding the relation between the response, the tensile strength, and the input variable, the PVA amount. Here, the calibration parameter is the percentage of PVA absorbed, which may change in the physical system depending on the values of the input variable, the PVA amount (Pourhabib et al., 2015).

From a geometric perspective, functional calibration is a curve to surface matching problem. Here, all the possible values for the input variable and the response constitute a one-dimensional curve. Note that the reason the physical system is denoted by a curve in a two-dimensional space is we cannot measure or observe the calibration parameter, also referred to as the latent variable (Pourhabib et al., 2015), in the physical process. As such, what we actually observe only consists of input variables and the responses which form a curve in the input-response space. On the other hand, in the computational model, we can specify the values of both the input and the calibration parameter. Consequently, all the possible values of the variable, calibration parameter, and the computational model response together form a surface (see Figure 1). If the problem is point calibration, i.e., there exists one single value for the calibration parameter, then we need to match the curve from the physical system to the surface from the computational model, since the computational model is intended to represent the physical process.

We note that the physical curve we observe is in fact a projection of a curve in the three dimensional input-parameter-response space. The reason lies behind the nature of parameters in a physical process: for each input there exists an (unknown) value for the parameter, and these two features, i.e the input and parameter, determine one single response. However, since we do not observe the actual value of the parameter in the physical process, we only see a projected curve on input-response surface. Hence, this problem is to recover the original curve, or in other words, determine a non-isometric match of a curve to a surface. The non-isometry is due to the fact that the curve in the projected space has a different length than the original curve. Therefore, this is in principle different from isometric matching problems (Gruen and Akca, 2005, Bronstein et al., 2005, Baltsavias et al., 2008).

Furthermore, in functional calibration we do not observe a complete curve or surface, but a set of scattered data points obtained from the physical experiments or the computational model. The data points from the former lie on the projected curve we observe and the data from the latter lie on the input-parameter-response surface. Therefore, what we observe is incomplete data and we aim to match non-isometrically an incomplete curve to an incomplete surface, which is equivalent to solving the functional calibration problem.

The incompleteness of the data, that is scattered on a curve or a surface instead of
a continuous function, can be addressed by using kernel interpolation, such as Gaussian processes (GPs) (Stein, 1999). In fact, to handle functional calibration, both parametric and non-parametric approaches (Pourhabib et al., 2015, 2014) use kernel interpolation. After interpolating the data points to get a continuous surface, the former assumes a parametric relationship between the inputs and calibration parameters and then estimates the unknown parameters of that function, whereas the latter, the non-parametric approach, formulates the problem using the concepts of reproducing kernel Hilbert spaces (RKHS) and then finds a solution using the representer’s theorem (Schölkopf et al., 2001).

However, using interpolation introduces a new source of error: assuming a GP, for example, relies on the joint Gaussian distribution which is not always the case in practice. Therefore, we argue if we operate on the very data points we will obtain a better matching in this case. In addition, the existing methods appeal to iterative numerical techniques for optimization which are prone to get trapped in local optima. The interaction between these two issues, interpolation and iterative optimization, further exacerbates the quality of the optimal solution, as the methods converge to a local optimum of the model based on the interpolated surface.

In this paper, we propose a graph-theoretic approach for non-isometric curve to surface matching from incomplete data for functional calibration. We reformulate the functional calibration problem as a shortest path problem on a directed acyclic graph (DAG) which can be solved efficiently with a fast polynomial-time algorithm. The shortest path found identifies anchor points from the computational model points that balance two basic criteria: First, the anchors should offer the best match to the response of the physical system, and second, they should facilitate fitting of a smooth function (not exhibit erratic fluctuations in values). The anchor points are then used to fit a response function that can then be used to make predictions on the physical process. Applying the overall framework developed in this paper on both synthetic and real data, we demonstrate that the new approach outperforms the alternative approaches in terms of prediction accuracy.

The rest of this paper is organized as follows: Section 2 formally defines the problem and reviews the relevant literature. Section 3 outlines our proposed approach to solve the non-isometric curve to surface matching. Section 4 presents the numerical results and makes suggestions regarding parameter selection. Section 5 concludes the paper.

2 Non-isometric matching

2.0.1 Problem Definition

We consider a physical system that operates according to a set of (possibly unknown) physical laws. In this system, we assume there is a functional relationship between a group of features and an output. The physical system has an input variable denoted by \( x \in \mathbb{R} \). Input variables represent those values that can be observed or specified. As such, the vector of input variables are also called control variables. We also assume that once in the
physical system the control variables are set (either observed or specified), the physical process generates a real valued response. However, the response is not merely a function of those control variables, but a function of both the control variables and parameters \( \theta \in \mathbb{R} \). The latter denotes those features of the physical systems that are either hard to measure or control. Hence, to express this functional relationship we denote the response as \( y^p(x; \theta) \). Likewise, we denote the output of the computational model by \( y^c(x, \theta) \), although here, \( \theta \) is also an input argument. An example of such a complex system is PVA-treated buckypaper fabrication, where we are interested in finding the relation between the input variable PVA amount and the response, namely the buckypaper tensile strength. Here, the percentage of PVA absorbed is a calibration parameter, since it cannot be measured in the physical system.

Note that, assuming our computational model is constructed according to the laws governing the physical system, in a computational model the same concepts hold: we have a “system” whose structure is determined by the interactions between some input/control variables and parameters. However, since in a computer code we are not restricted to measuring any physical features, we can set the values of both \( x \) and \( \theta \) arbitrarily. Of course, the goal is to design a high-fidelity computational model that operates similar to the physical system. To this end, we need to obtain the physical responses \( \{y_1^p, \ldots, y_m^p\} \) at a set of physical inputs \( D = \{x_1^p, \ldots, x_m^p\} \), and the computational model responses \( \{y_1^c, \ldots, y_n^c\} \) at the model inputs \( G = \{(x_1^c, \theta_1^c), \ldots, (x_n^c, \theta_n^c)\} \), where \( y_j^p = y^p(x_j^p; \theta_j^p) \) and \( y_j^c = y^c(x_j^c, \theta_j^c) \), for \( j = 1, \ldots, m \) and \( i = 1, \ldots, n \). In general, due to cost considerations, \( m \ll n \), i.e., obtaining data points from the computational model is generally cheaper than getting data from the physical experiment. As such, we also assume that for each \( x_j^p \in D \) there exists at least an \( i \in \{1, \ldots, n\} \) such that \( x_i^c = x_j^p \), i.e., for any physical input we have at least one model input \( (x_i^c, \theta_i^c) \) where the control variable in the computational model is the same as that in the physical system. The goal in calibration is to adjust the parameter \( \theta \), such that the computational model represents the physical system in the sense that the computational model can predict the physical response at any input location \( x_i^c \).

We call any method seeking a single value \( \theta \in \mathbb{R} \) that matches the computational model with the physical system, point calibration (this class of methods is simply called calibration in the literature; see [Tuo and Wu, 2014; Bayarri et al., 2007; Joseph and Melkote, 2009; Higdon et al., 2004b; Han et al., 2009b; Xiong et al., 2009]). On the other hand, functional calibration refers to finding a function \( f \), where \( \theta = f(x) \), such that the output of the computational model matches that of the physical system for any given \( x \), i.e. \( f \) is a solution to

$$\min_{f \in \mathcal{F}} \{ \mathcal{L}(y^p(x; \theta), y^c(x, \theta)) | \theta = f(x) \},$$

(1)

where \( \mathcal{F} \) is a space of functions and \( \mathcal{L}(\cdot, \cdot) \) is a loss function that measures the discrepancy between the responses of the physical system and the computational model. What we call functional calibration is also referred to as local calibration (Pourhabib et al., 2014) or
latent variable estimation (Pourhabib et al., 2015).

Functional calibration has an interesting geometric interpretation. Note that $y_p(x; \theta)$ as a function of mere $x$ constitutes a one-dimensional curve. On the other hand, $y_c(x, \theta)$ as a function of both $x$ and $\theta$ is a two-dimensional surface. So, the optimization problem (1) is matching a one-dimensional curve to a two-dimensional surface. However, this is different from traditional matching problems (Gruen and Akca, 2005), since the length of the one-dimensional curve is not preserved. Specifically, if $(x, \theta, y_p(x; \theta))$ represents the true physical curve, what we observe is its projection $(x, 0, y_p(x; \theta)) \in \mathbb{R}^3$, which is a one-dimensional curve. On the other hand, the two-dimensional surface is set of all points $(x, \theta, y_c(x, \theta)) \in \mathbb{R}^3$. Now, we are trying to find $\theta = f(x)$ such that the curve $(x, \theta, y_p(x; \theta))$ lies on the surface $(x, \theta, y_c(x, \theta)) \in \mathbb{R}^3$; however, the length of the curve $(x, 0, y_p(x; \theta))$ is larger than that of $(x, 0, y_c(x; \theta))$, and consequently the matching procedure is non-isometric. As such, we note that functional calibration for the case where $\theta$ and $x \in \mathbb{R}$ is non-isometric curve to surface matching.

In practice we do not have the entire curve or surface, but scattered data points from those sets. Recall that we are provided with the physical inputs $D = \{x_p^j : j = 1, \ldots, m\}$ and model inputs $G = \{(x_p^i, \theta_p^i) : i = 1, \ldots, m\}$ from the computational model. As such, we have $\{(x_p^1, 0, y_p^1), \ldots, (x_p^m, 0, y_p^m)\}$, and $\{(x_c^1, \theta_c^1, y_c^1), \ldots, (x_c^m, \theta_c^m, y_c^m)\}$ sampled from the projected curve and the surface, respectively, where $y_p^j = y_p(x_p^j; f(x_p^j))$ for $j = 1, \ldots, m$, and $y_c^i = y_c(x_c^i, \theta_c^i)$ for $i = 1, \ldots, n$. Under this setting, the problem is then non-isometric matching of an incomplete curve to an incomplete surface, which we call non-isometric matching with incomplete data. Once the function $f$ is estimated, we can predict the physical response at any point $x_c^*$ by $y_c^*(x_c^*, f(x_c^*))$. See Figure 1 for an illustration.

2.0.2 Related Work

The characterization of function $f$ determines the solution approach to problem (1). One way is to characterize $f$ as an element of a reproducing kernel Hilbert space (RKHS) (Tuoc and Wu, 2014). Assume $\Phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ denotes a symmetric positive semidefinite kernel, i.e.,

$$\int \Phi(x, x') f(x) f(x') d\mu(x) d\mu(x') \geq 0,$$

for any $f \in L_2(\mathbb{R}^r, \mu)$ where $\mu$ is a measure (Rasmussen and Williams, 2006), which can define a space of smooth functions: For a given $x_j \in \mathbb{R}$, the function $\Phi(x, x_j)$ is a real-valued function. Then, define the space of all functions constructed as a linear combination of $\Phi(x, x_j)$ for $x_j \in \mathbb{R}$, as:

$$F_{\Phi}(\mathbb{R}) = \left\{ \sum_{j=1}^{\infty} \beta_j \Phi(\cdot, x_j), \beta_j \in \mathbb{R}, x_j \in \mathbb{R} \right\}.$$
Figure 1: Curve to surface matching for a synthetic dataset used in Section 4.0.3. The top plot shows the complete surface and curve. The computational model data points lie on a two-dimensional surface. The true physical curve needs to be recovered by projecting back the observed physical curve to the 3D space. In practice, however, what we observe are scattered data points sampled from the complete curve and surface, which is depicted in the bottom plot.
We can show that the closure of $F_\Phi(\mathbb{R})$ with respect to the norm induced by the inner product

$$\left\{ \sum_{j=1}^{\infty} \beta_j \Phi(., x_j), \sum_{\ell=1}^{\infty} \gamma_{\ell} \Phi(., x_{\ell}') \right\} := \sum_{j=1}^{\infty} \sum_{\ell=1}^{\infty} \beta_j \gamma_{\ell} \Phi(x_j, x_{\ell}'),$$

is an RKHS (Wendland, 2005; Fasshauer, 2011). Then, non-parametric functional calibration (NFC) (Pourhabib et al., 2014), given a kernel, seeks a function $f$ in that RKHS as a solution to a variation of problem (1). Specifically, NFC solves

$$\min \frac{1}{m} \sum_{j=1}^{m} \left\{ y^p(x_j^p; f(x_j^p)) - y^c(x_j^p, f(x_j^p)) \right\}^2 + \gamma \| f \|^2,$$

(2)

where $f = \sum_{j=1}^{\infty} \beta_j \Phi(., x_j)$, and $\| f \|^2 = \sum_{j=1}^{\infty} \sum_{\ell=1}^{\infty} \beta_j \beta_{\ell} \Phi(x_j, x_{\ell})$, $\gamma > 0$ is a regularization parameter. The generalized representer’s theorem (Schölkopf et al., 2001) guarantees that any local optimum $f(x) = \sum_{j=1}^{m} \hat{\beta}_j \Phi(x, x_j^p)$, i.e. a linear combination of the kernel evaluations at $\mathcal{D} = \{x_1^p, \ldots, x_m^p\}$. Therefore, we need to find $\hat{\beta}$ such that

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^m} \frac{1}{m} \sum_{j=1}^{m} \left\{ y^p \left( x_j^p; f(x_j^p) \right) - y^c \left( x_j^p, f(x_j^p) \right) \right\}^2 + \gamma \sum_{j=1}^{m} \sum_{\ell=1}^{m} \beta_j \beta_{\ell} \Phi(x_j^p, x_{\ell}^p),$$

where

$$f(x_j^p) = \sum_{\ell=1}^{m} \beta_{\ell} \Phi(x_j, x_{\ell}^p),$$

(3)

which is a non-convex optimization problem in general, due to the fact that $\beta$ is an argument of the non-convex function $y^c(\cdot, \cdot)$.

Note that in optimization problem (3), we have $y^c(x_j^p, f(x_j^p))$ which means we evaluate the computational model at $\mathcal{D}$. This poses no issue since we assumed for each $x_j^p \in \mathcal{D}$ there exists at least an $i \in \{1, \ldots, n\}$ such that $x_i^c = x_j^p$. However, since we solve (3) using iterative numerical techniques, such as trust region methods (Byrd et al., 2000; Conn et al., 2000), or steepest descent (Luenberger and Ye, 2008), we may have to evaluate $y^c(\cdot, \cdot)$ at points other than those in $\mathcal{G}$. Therefore, we need to create a surrogate model that interpolates the computational model. Specifically, let $\hat{y}^c(x, \theta)$ denote a real valued function defined on $\mathbb{R}^2$, where $\hat{y}^c(x_i^c, \theta_i^c) = y^c(x_i^c, \theta_i^c)$, for every $(x_i^c, \theta_i^c) \in \mathcal{G}$, i.e. $\hat{y}^c(\cdot, \cdot)$ interpolates $y^c(\cdot, \cdot)$. Furthermore, if at any point $(x, \theta) \in \mathbb{R}^2 \setminus \mathcal{G}$, $\hat{y}^c(\cdot, \cdot)$ provides a “good guess” of what $y^c(\cdot, \cdot)$ would be, we call $\hat{y}^c(\cdot, \cdot)$ a surrogate model for $y^c(\cdot, \cdot)$. A popular choice for the surrogate model is based on GP regression (Stein, 1999; Rasmussen and Williams, 2006).
An alternative to using a non-parametric form for \( f \), namely a function in the RKHS, is to assume a parametric form. For example, guided by the physics of the buckypaper fabrication process (Pourhabib et al., 2015), we can assume \( \theta = f(x) = \alpha \exp(-\beta x) \). Then, the functional calibration problem seeks \((\alpha^*, \beta^*)\) such that

\[
(\alpha^*, \beta^*) = \arg\min_{\alpha, \beta} \sum_{j=1}^{m} \left[ y^p(x^p_j; \alpha \exp(-\beta x^p_j)) - \hat{y}^c(x^p_j, \alpha \exp(-\beta x^p_j)) \right]^2 ,
\]

where we use \( \hat{y}^c(\cdot, \cdot) \) again as a surrogate model as solving (4) requires appealing to iterative numerical techniques that need to evaluate the computational model at points not necessarily in \( \mathcal{G} \). We call this approach parametric functional calibration (PFC) (which is also referred to as latent variable estimation in (Pourhabib et al., 2015)).

PFC is well suited for the buckypaper fabrication problem, however, for most applications we cannot assume a parametric form for the calibration function \( f \) \textit{a priori}. In addition, using a surrogate model introduces a new source of error, since we are interpolating the computational model surface. We cannot overcome the issue of using a surrogate model in PFC, or even NFC, since they are both solved using iterative numerical techniques that require evaluating the objective function at arbitrary points on its domain. Note that once we use a surrogate model in the optimization procedure, the point identified as optimal may be far way from the actual optimal point. In the following section, we propose an alternative framework in which we employ the computational model data points directly to determine the actual physical curve.

### 3 Graph Shortest Path for Non-isometric Matching

The intuition that the functional calibration problem is non-isometric curve to surface matching, and that we are provided with scatter data points from the curve and the surface, motivates us to view the problem through a discrete combinatorial lens. In this approach, we model functional calibration as a combinatorial optimization problem on a graph.

Unlike NFC and PFC discussed in Section 2.0.2, we do not identify a class of functions \textit{a priori}. Instead we seek to identify anchor points, among computational model points, which, intuitively speaking, are “close” to the points on the true physical curve. In other words, the anchor points are positioned such that the true physical curve passes through neighborhoods of those points. To find these anchor points, we require two properties to hold: (i) the computational model response should be close to the physical response for a given input \( x \); and (ii) the parameter values for two consecutive anchor points should be close to each other. The former ensures the computational model identifies the points that have the same response as that of the physical system, and the latter ensures the smoothness of the estimated physical curve. Next, we show how we translate this idea and the properties into a rigorous graph-theoretic framework.
Without loss of generality, we assume that both the physical inputs $\mathcal{D} = \{x^p_j : j = 1, \ldots, m\}$ and model inputs $\mathcal{G} = \{(x^c_i, \theta^c_i) : i = 1, \ldots, n\}$ are strictly ordered such that $x^p_j < x^p_{j+1}$, for $j = 1, \ldots, m-1$, and $x^c_i < x^c_{i+1}$, for $i = 1, \ldots, n-1$. Then, we construct a directed, edge-weighted graph, $G = (V, E)$ where $V = V^0 \cup \{0, n+1\}$, and the vertices in $V^0 = \{1, \ldots, n\}$ correspond to the model inputs in $\mathcal{G}$.

We partition the $V^0$ into $m$ clusters $\{C_1, \ldots, C_m\}$ as follows: Consider any vertex $i \in V^0$ corresponding to $(x^c_i, \theta^c_i) \in \mathcal{G}$. Then, $i$ is assigned to a unique cluster $C_j$ for some $j \in \{1, \ldots, m\}$ as follows:

$$i \in C_j \text{ if } j = \arg \min \{|x^p_\ell - x^c_i| : \ell = 1, \ldots, m\}. \tag{5}$$

Note that, if the minimum in (5) is achieved by more than one $j \in \{1, \ldots, m\}$, then the tie can be broken arbitrarily. As a consequence, each cluster $C_j$ is in 1-to-1 correspondence with the physical input $x^p_j$. We can now describe the set of directed edges $E$ as follows:

$$E = \bigcup_{j=1, \ldots, m-1} \{(u, v) : u \in C_j, v \in C_{j+1}\} \bigcup \{(0, u) : u \in C_1\} \bigcup \{(u, n+1) : u \in C_m\} \tag{6}$$

The first term corresponds to all the directed edges that go from vertices in $C_j$ to vertices in $C_{j+1}$. The second term corresponds to the directed edges from the artificial vertex 0 to each vertex in $C_1$. The third term corresponds to directed edges from vertices in $C_m$ to the artificial vertex $n+1$.

The final critical step is assigning weights $w_{uv}$ to each edge $(u, v) \in E$. Consider two consecutive clusters $C_j$ and $C_{j+1}$ for any $j = 1, \ldots, m$ and let $u \in C_j$ and $v \in C_{j+1}$. Define $w_{uv}$ as follows.

$$w_{uv} = |y^p_u - y^p_v| + \lambda|\theta^c_u - \theta^c_v|, \tag{7}$$

where $\lambda > 0$ is a scaling parameter. The weights $w_{0,u} \forall u \in C_1$ and $w_{u,n+1} \forall u \in C_m$ are identically zero. Notice that the edge-weight for any edge between two consecutive clusters $j$ and $j+1$ consists of two parts: the first representing the difference between the model response and physical response; the second part represents the difference between the calibration parameters of $j$ and $j+1$.

On this graph $G$, we intend to solve the shortest (simple, directed) path problem from origin vertex 0 to destination vertex $n+1$. Every path from 0 to $n+1$ in $G$ has exactly $m+1$ arcs, by construction. Therefore, the $m$ internal vertices on the shortest path that is found will serve as the anchor points for fitting the function. See Figure 2 for illustration.

Observe that, $G = (V, E)$ as described here is a directed acyclic graph (DAG). Furthermore, we claim that $0, 1, \ldots, n, n+1$ is a topological ordering of vertices of $G$. To establish this claim it suffices to show that if $(u, v) \in E$, then $u < v$. Note that $u < v \iff x^p_u < x^c_v$ as we have assumed the inputs to be strictly ordered. Suppose $u \in C_j$ and $v \in C_{j+1}$ where $j \in V^0 \setminus \{m\}$. Hence, $x^p_j < x^p_{j+1}$, $j = \arg \min \{|x^p_\ell - x^c_u| : \ell = 1, \ldots, m\}$ and $j+1 = \arg \min \{|x^p_\ell - x^c_v| : \ell = 1, \ldots, m\}$. Since, $x^p_j, x^c_u, x^c_j$ and $x^p_{j+1}$ are points on a real
Figure 2: Graph representation of the functional calibration problem for the case where $m = 5$ and $n = 13$. Crosses represent data points from the computational model and solid circles show the physical data points which correspond to clusters $C_1$ through $C_m$. The darker crosses represent the anchor points, and the solid arrows identify the edges in the shortest path.

line, we must have $x^c_u < x^c_v$. Therefore, we can solve the shortest path problem in $O(|E|)$ complexity using the reaching algorithm for DAGs (Lawler, 1976) (see Algorithm 1).

**Algorithm 1** Reaching algorithm for DAGs

```
1: dist(0) := 0, dist(u) := ∞ ∀ u ∈ V \ {0}
2: pred(0) := 0, pred(u) := −1 ∀ u ∈ V \ {0}
3: for u = 0, . . . , n do
4:    for each (u, v) ∈ E do
5:        if dist(v) > dist(u) + w_{uv} then
6:            dist(v) := dist(u) + w_{uv}
7:            pred(v) := u
8:        end if
9:    end for
10: end for
```

Suppose $0 - v_1 - v_2 - \cdots - v_m - (n + 1)$ is the shortest path identified. The points $\{(x^c_{v_1}, \theta^c_{v_1}), \ldots, (x^c_{v_m}, \theta^c_{v_m})\}$ serve as the anchor points for fitting the function. Note that we are only interested in identifying these “optimal” anchor points that provide us with the information to estimate the “best” function $f$ (within this discrete optimization framework).

Given the anchor points, one approach is to fit a piecewise-linear function with the anchor points as break-points, or identify linear pieces via regression within clusters, with the additional constraint that the linear pieces coincide at the boundary resulting in a continuous piecewise-linear function. Alternately, we can use GP regression to interpolate between these points. Based on our experiments, the latter, i.e., GP regression, provided a
better solution in terms of predictive power. Specifically, we fit a GP on the anchor points\
\{(x^c_{v_1}, \theta^c_{v_1}), \ldots, (x^c_{v_m}, \theta^c_{v_m})\}, where \{x^c_{v_1}, \ldots, x^c_{v_m}\} are the index set and \{\theta^c_{v_1}, \ldots, \theta^c_{v_m}\} are noise-contaminated output of the GP \( f \), i.e. \( \theta^c_{v_j} = f(x^c_{v_j}) + \epsilon_j \), for \( j = 1, \ldots, m \), where \( \epsilon_j \sim N(0, \sigma^2) \) and \( \sigma^2 > 0 \).

We call this approach graph non-isometric matching (GNM), which characterizes \( F \) in optimization (1) as the space of smooth GP posterior functions. The proximity of the anchor points to the physical responses is emphasized by our construction (clustering and edge-weights), enables a good match between the physical system and the computational model.

4 Experiments

This section presents the numerical results for the proposed method, GNM, and compares its performance with two competing methods, PFC and NFC. We use one real and two synthetic datasets. The measure of performance is prediction accuracy; after using a training dataset we select a set of test data points, \( T = \{x^*_1, x^*_2, \ldots x^*_n\} \) and then for each method we find the predicted response values \( \{\hat{y}_1, \hat{y}_2, \ldots \hat{y}_n\} \). We compare the predicted values with the actual physical responses at those points, i.e. \( \{y^p(x^*_1; \theta^*_1), y^p(x^*_2; \theta^*_2), \ldots y^p(x^*_n; \theta^*_n)\} \).

Then, we compute the root mean squared error (RMSE)

\[
\text{RMSE} = \sqrt{\frac{1}{n_t} \sum_{i=1}^{n_t} (\hat{y}_i - y^p(x^*_i; \theta^*_i))^2},
\]

for each method. A method with a smaller RMSE is deemed superior.

To create training and test cases we follow two approaches: we can select the test data set \( T \) to be the set of inputs in the training data set, i.e. \( T = \{x^p_1, x^p_2, \ldots x^p_m\} \). The second approach is to use 4-fold cross validation [Hastie et al., 2001]. We call the error found through the latter the test error and the former the training error.

For synthetic datasets, since we know the actual functional relationship between the calibration parameter and the input variable, we can also calculate the RMSE for the calibration parameter prediction. Specifically,

\[
\text{RMSE}_\theta = \sum_{i=1}^{n_t} (\hat{\theta}_i - \theta^*_i)^2,
\]

where \( \hat{\theta}_i \) is the estimated parameter at the input \( x^*_i \), i.e., \( \hat{\theta}_i = f(x^*_i) \), where \( f(\cdot) \) is the estimated calibration function, and \( \theta^*_i \) is the actual parameter.

4.0.3 Synthetic Data

We use two synthetic datasets, initially designed to test NFC [Pourhabib et al., 2014]; however, we use different ranges for the input variables and slightly modify the way the
input locations are selected as explained. First, we consider a physical system with a response function $y^p(x; \theta) = \exp\left(\frac{x}{10}\right) \sin(x)$ and an associated computational model $y^s(x, \theta) = \exp\left(\frac{x}{10}\right) \sin\left(\frac{x}{20}\right)$. We note that when $\theta = .5x$ the computational model matches the physical response. As such, the true functional relationship is given by $f(x) = 0.5x$. We name this dataset SD1. The second synthetic dataset we consider displays a more complicated physical response and a nonlinear relationship between the calibration parameter and control variables. Specifically, the physical system follows $y^p(x; \theta) = \cos(2x) \sin\left(\frac{x}{2}\right)$ and the associated computational model surface $y^s(x, \theta) = \cos(2x) \sin\left(\frac{x}{2}\right) \sin\left(\frac{\pi \theta}{2(x-2)^2+2}\right) \cos\left(\frac{2\pi \theta}{(x-2)^2+1}\right) \exp\left(\frac{\theta}{2(x-2)^2+2} - \frac{1}{2}\right)$. Note that if the parameter $\theta = (x - 2)^2 + 1$ the computational model matches the physical system. We name this dataset SD2. We use two-fold cross validation to select the regularization parameter $\lambda$, which suggests $\lambda = 0.4$ for SD1 and $\lambda = 0.3$ for SD2. Figure 1 illustrates the problem for SD2.

To calculate the training error, we use $m = 15$ physical data points and $n = 450$ data points for the computational model. To calculate the test error, we use $m = 30$ physical data points and $n = 900$ points from the computational model. For SD1 the physical data points are uniformly sampled from $[9\pi/8, 5\pi/2]$ and the computational model data points are sampled from $[9\pi/8, 5\pi/2] \times [\pi/4, 3\pi/4]$. For SD2 the physical data points are sampled from $[3\pi/8, \pi]$, and the computational model data points are sampled from $[3\pi/8, \pi] \times [\pi/4, \pi]$. Table 1 shows the results for both SD1 and SD2. For each training or test case, we present the results for response prediction, measured by RMSE, and parameter prediction, measured by $\text{RMSE}_\theta$. For test cases, the numbers in parentheses are the standard deviations of the four folds. The table shows that GNM outperforms NFC and PFC for all the comparison cases. In the following, we analyze some of the results in more detail.

Comparing training cases of SD1 with SD2, we observe that the performance of PFC deteriorates on the dataset with a quadratic relationship between the control variables and the parameters, whereas for GNM and NFC the nonlinearity of function $f$ does not affect the performance since neither fixes a parametric functional from a priori. An interesting observation for the test errors in SD1 is while NFC has a relatively small RMSE, its $\text{RMSE}_\theta$ is significantly worse. A closer look at the results revealed that the large value for $\text{RMSE}_\theta$ can be attributed to two reasons. First, for the test cases each method is provided with a smaller number of data points. For NFC, this resulted in the algorithm finding a local optimum in the region which has a similar response surface as the training region, but different values of the parameters. This occurs due to the fact that NFC uses function approximation for the computational model, thereby it assumes the function following similar trends over a large region. Note that this problem does not occur for all the possible ranges for the inputs, like the ranges used in (Pourhabib et al., 2014) for building synthetic data. In fact, we realized this issue while trying different ranges for the data, and then used the aforementioned range in this paper to demonstrate this point. This can be a major drawback for some applications in which not only a good prediction for the response is desired, but also the relationship between the control variables and the parameters is
Table 1: Comparing NFC, PFC and GNM on synthetic datasets.

|       | Training |       |       |       |
|-------|----------|-------|-------|-------|
|       | RMSE     | RMSEθ | RMSE  | RMSEθ |
| NFC   | 0.267    | 0.454 | 0.180 | 0.134 |
| PFC   | 0.220    | 0.303 | 0.351 | 0.131 |
| GNM   | 0.029    | 0.056 | 0.031 | 0.017 |

|       | Training |       |       |       |
|-------|----------|-------|-------|-------|
|       | RMSE     | RMSEθ | RMSE  | RMSEθ |
| NFC   | 0.175    | 0.173 | 0.290 | 0.126 |
| PFC   | 0.380    | 0.329 | 0.571 | 0.0503|
| GNM   | 0.017    | 0.027 | 0.043 | 0.022 |

In summary, the significantly lower RMSE and RMSEθ for GNM, compared to NFC and PFC can be explained by noting that GNM does not model the computational model as a continuous surface. In addition, the way the graph G is constructed adequately describes the non-isometric matching nature of this problem.

4.0.4 Real Data

We use a real dataset from fabrication of buckypaper which is a nano-manufactured product composed of thin sheets of carbon-nano tubes \cite{Tsai2011}. An approach to enhance the mechanical properties of buckypaper involves adding PVA to the fabrication process. A computational model is then used to model the relationship between the output, measured by Young’s modulus of the buckypaper, and the input, the PVA amount \cite{Wang2013}. Here, the calibration parameter is the percentage of the PVA absorbed in the process. Based on the understanding of the physics of the problem, this parameter is related, through a functional relationship, to the input variable, the PVA amount \cite{Pourhabib2014}. As such, this problem falls in the category of functional calibration or, based on the framework we proposed in this paper, non-isometric matching.

We use \( m = 11 \) physical data points, where the input variables are uniformly distributed between 0.5 and 1. The number of computational model data points is 150. Since we do not know the true functional relationship between the parameter and the input, we can only evaluate the methods by calculating their response prediction.
using both training error and 4-fold cross validation. For this dataset, NFC provides the best result for the training error. For the test case, GNM provides the best result. This demonstrates when we have a smaller number of training physical data points, the graph-theoretic approach is more reliable than both NFC and PFC.

Table 2: Comparing NFC, PFC and GNM on buckypaper data.

| Method | Training RMSE | Test RMSE |
|--------|---------------|-----------|
| NFC    | 0.121         | 0.301(0.248) |
| PFC    | 0.291         | 0.609(0.441) |
| GNM    | 0.276         | 0.285(0.164) |

4.1 Selection of regularization parameter

Here we discuss the impact of the numerical value of the regularization parameter $\lambda$ in equation (7). If $\lambda = 0$ the weight for the edges in the graph is solely decided by the difference in the responses of the physical data points and the simulated data points in their neighborhood. This may result in a large difference between the second values of the parameters of the two consecutive points selected in model (7). On the other hand, if $\lambda$ is a very large number, the selected computational model points, which are used to estimate the true physical curve, may be very close to each other and the estimated curve may deviate from the actual physical curve. In our analysis we use 2-fold cross validation to select the parameter $\lambda$. Based on our analysis, this approach identifies a good value for the regularization parameter. To test this, Figure 3 displays the test error for the two synthetic datasets as a function of $\lambda$. The test error is calculated by randomly partitioning each dataset into 75% of training points and 25% for testing. The figure suggests that the value of the error initially reduces as a function of $\lambda$ then it sharply increases. For SD2, due to shape of the computational model surface, even a very small number close to zero provides a small prediction error. The values for $\lambda$ we obtained in Section 4.0.3 using cross validation error, which were 0.4 and 0.3 for SD1 and SD2, respectively, are very close to the optimal values in this figure.

5 Conclusion

In this paper, we developed a novel graph-theoretic approach for non-isometric curve to surface matching from incomplete data for functional calibration. The motivating example is the PVA-treated buckypaper fabrication process in which the incomplete curve is the set
of physical data points and the incomplete surface is the set of computational model data points. In this framework, the functional calibration problem was recast as a shortest path problem on a directed acyclic graph. The solution to the shortest path problem is used to identify anchor points that are then used to fit a response function. On both synthetic and real data, we demonstrated that our new approach outperforms competing approaches in terms of prediction accuracy.

An interesting question, which is of importance in many manufacturing problems, is how to generalize GNM to match low dimensional surfaces to higher dimensional surfaces. This problem arises when the physical process has more than one input, such as the welding example discussed in (Bayarri et al., 2007). A major challenge in this case is that the points do not possess an inherent ordering like in the one-dimensional case. Therefore this generalization is not a straightforward task and will constitute our future research.

Acknowledgment

Author Balasundaram would like to acknowledge the support of the Air Force Office of Scientific Research grant FA9550-12-1-0103 and the National Science Foundation grant CMMI-1404971. This work was completed utilizing the High Performance Computing Center facilities of Oklahoma State University at Stillwater. The authors also acknowledge Dr. Chuck Zhang from Georgia Institute of Technology for providing the authors with the buckypaper fabrication data.
References

Baltsavias, E., A. Gruen, H. Eisenbeiss, L. Zhang, and L. Waser (2008). High-quality image matching and automated generation of 3d tree models. International Journal of Remote Sensing 29(5), 1243–1259.

Bayarri, M. J., J. O. Berger, J. Cafeo, G. Garcia-Donato, F. Liu, J. Palomo, R. J. Parthasarathy, R. Paulo, J. Sacks, and D. Walsh (2007). Computer model validation with functional output. The Annals of Statistics 35(5), 1874–1906.

Bayarri, M. J., J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C. H. Lin, and J. Tu (2007). A framework for validation of computer models. Technometrics 49(2), 138–154.

Bronstein, A. M., M. M. Bronstein, and R. Kimmel (2005). Three-dimensional face recognition. International Journal of Computer Vision 64(1), 5–30.

Byrd, R. H., J. C. Gilbert, and J. Nocedal (2000). A trust region method based on interior point techniques for nonlinear programming. Mathematical Programming 89(1), 149–185.

Conn, A. R., N. I. M. Gould, and P. L. Toint (2000). Trust-Region Methods. SIAM.

Fasshauer, G. E. (2011). Positive definite kernels: Past, present and future. Dolomite Research Notes on Approximation 4, 21–63.

Gruen, A. and D. Akca (2005). Least squares 3d surface and curve matching. ISPRS Journal of Photogrammetry and Remote Sensing 59(3), 151–174.

Han, G., T. J. Santner, and J. J. Rawlinson (2009a). Simultaneous determination of tuning and calibration parameters for computer experiments. Technometrics 51(4), 464–474.

Han, G., T. J. Santner, and J. J. Rawlinson (2009b). Simultaneous determination of tuning and calibration parameters for computer experiments. Technometrics 51(4), 464–474.

Hastie, T., R. Tibshirani, and J. Friedman (2001). Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer.

Higdon, D., M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne (2004a). Combining field data and computer simulations for calibration and prediction. SIAM Journal of Scientific Computing 26, 448–466.

Higdon, D., M. Kennedy, J. C. Cavendish, J. A. Cafeo, and R. D. Ryne (2004b). Combining field data and computer simulations for calibration and prediction. SIAM Journal on Scientific Computing 26(2), 448–466.

Joseph, V. R. and S. N. Melkote (2009). Statistical adjustments to engineering models. Journal of Quality Technology 41(4), 362–375.

Kennedy, M. C. and A. O’Hagan (2001). Bayesian calibration of computer models (with discussion). Journal of the Royal Statistical Society: Series B 63(3), 425–464.

Kumar, A., H. Noureddini, Y. Demirel, D. Jones, M. Hanna, et al. (2009). Simulation of corn stover and distillers grains gasification with aspen plus. Transactions of the ASABE 52(6), 1989–1995.

Lawler, E. (1976). Combinatorial Optimization: Networks and Matroids. New York: Holt, Rinehart, and Winston.

Luenberger, D. G. and Y. Ye (2008). Linear and nonlinear programming, Volume 116. Springer Science & Business Media.
Pourhabib, A., J. Z. Huang, K. Wang, C. Zhang, B. Wang, and Y. Ding (2015). Modulus prediction of buckypaper based on multi-fidelity analysis involving latent variables. *IIE Transactions* 47(2), 141–152.

Pourhabib, A., R. Tuo, J. Z. Huang, and Y. Ding (2014). Local calibration of computer models. *Manuscript*.

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. The MIT Press.

Schölkopf, B., R. Herbrich, and A. J. Smola (2001). A generalized representer theorem. In *Computational learning theory*, pp. 416–426. Springer.

Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Verlag.

Tsai, C., C. Zhang, J. A. David, B. Wang, and R. Liang (2011). Elastic property prediction of single-walled carbon nanotube buckypaper/polymer nanocomposites: stochastic bulk response modeling. *Journal of Nanosience and Nanotechnology* 11(3), 2132–2141.

Tuo, R. and C. F. J. Wu (2014). A theoretical framework for calibration in computer models: parametrization, estimation and convergence properties. Technical report, Chinese Academy of Sciences and Georgia Institute of Technology.

Wang, K. (2013). *Statistics-enhanced multistage process models for integrated design and manufacturing of poly(vinyl alcohol) treated buckypaper*. Ph. D. thesis, Florida State University, Tallahassee, FL.

Wendland, H. (2005). *Scattered Data Approximation*. Cambridge University Press.

Xiong, Y., W. Chen, K.-L. Tsui, and D. W. Apley (2009, March). A better understanding of model updating strategies in validating engineering models. *Computer Methods in Applied Mechanics and Engineering* 198(15-16), 1327–1337.