Technical Report PAR-10021867-092020-3

Interpolating tabular data for granular material models

Biswajit Banerjee
Parresia Research Limited, Auckland, New Zealand
b.banerjee.nz@gmail.com

David M. Fox
CCDC Army Research Laboratory, Aberdeen Proving Ground, MD, USA
david.m.fox1.civ@mail.mil

Richard A. Regueiro
University of Colorado, Boulder, CO, USA
richard.regueiro@colorado.edu

Monday, 9 November, 2020

Abstract

Experimental and microscale simulation data are used to design material models for granular materials. These data are collected in tabular form, often as the outcome of a design-of-experiments process when multiple independent variables are expected to affect the result of an experiment. Tabular data are collected densely for one independent variable, typically the strain. Data are obtained sparsely in the other dimensions. In this paper, we discuss possible approaches to using tabular data directly in material models without attempting to design and fit closed-form expressions. A dry, poorly-graded, concrete sand is used as the test material and we focus on the plastic strain-dependent bulk modulus. Linear interpolation, radial basis functions, and kriging are explored. The kriging and linear interpolators are found to produce acceptable predictions provided bulk moduli are computer from the input data and then interpolated. We recommend kriging for models that involve more than two independent variables.

1 Introduction

Continuum mechanics-based simulations require material models for closure of the governing system of partial differential equations. Though the primary features of these material models, such as elasticity, can vary significantly, many standardized models are available. A user can find parameters for these models from handbooks, research articles, or through standardized experiments. However, many of these models require secondary features such as density dependence, strain-rate dependence, coupled elasto-plasticity, etc. for accurate representation of the physics.

Granular materials such as soils and rocks are particularly problematic because of the wide range of possible secondary material model features. Experimental data for these materials are typically saved as tables where the primary independent variable (e.g., strain) is densely sampled while secondary independent variables (e.g., density, plastic strain, etc.) are sparsely sampled. One approach of dealing with such materials is to develop closed-form expressions for the models and then to fit parameters.

For example, in the KAYENTA model (Brannon et al., 2015; Banerjee and Brannon, 2019) for granular solids, experimental data are fitted to the bulk modulus (K) model:

$$K(I_1, \varepsilon_v^p) = f_K \left[\{ b_0 + b_1 \exp(-b_2/|I_1|) \} - b_3 \exp(-b_4/|\varepsilon_v^p|) \right] \tag{1}$$

where I_1 is the trace of the stress, ε_v^p is the plastic volumetric strain, f_K is a joint degradation factor, and (b_0, b_1, b_2, b_3, b_4) are fitted parameters. When an attempt is made to fit this model to data on sand and clay, it is quickly discovered that the model is inadequate and has to be reformulated.

The process of reformulating and refitting models whenever a new material is encountered is inefficient and cumbersome, and therefore avoided in practice. Instead, the tabular experimental data are directly used for simulations. However, there is no universally accepted method of interpolating tabular data of the form encountered in mechanics.

In previous work, we have discussed support vector machines (Banerjee, Fox, and Regueiro, 2020b) and artificial neural networks (Banerjee, Fox, and Regueiro, 2020a) for interpolating tabular experimental data. In this paper, we discuss approaches for interpolating unstructured tabular data and focus on simulations where there are multiple independent variables. Linear interpolation, radial basis function interpolation, and kriging interpolation are explored. These approaches are applied to the problem of interpolation of the bulk modulus of a dry poorly-graded concrete sand.

Section 2 discusses the three interpolation approaches explored in this paper. The experimental data used to compare the three approaches are briefly described in Section 3. Results are shown in Section 4 and some concluding remarks are given in Section 5.

2 Interpolating tabular data

Multivariate experimental data are often collected in the form of ragged tables (Browne et al., 1995). All independent variables, except one, are kept fixed and dependent variables are measured as the single independent variable is varied. This process gives a structure to the data because the measured independent variable is usually identical for each experiment and sampled densely. On the other hand, the remaining independent variables are sampled relatively sparsely. For example, elastic unloading curves in soil plasticity may have the form shown in Table 1.

A schematic of ragged tabular data can be seen in Figure 1. In this particular case there are three independent variables: the plastic strain (β), the saturation (α), and the total volumetric strain (ε). The pressure (p) is the dependent variable. The data represent a function of the form $p = p(\varepsilon, \alpha, \beta)$. For a numerical simulation to be able to use such tabular data, we have to compute the pressure (p_0) and its derivatives ($\partial p/\partial\alpha$, $\partial p/\partial\beta$, $\ldots$) given an input point (ε_0, α_0, β_0) in the three-dimensional independent variable space. This process involves interpolation. In general, there can be n independent variables and m dependent variables. Clearly, the data and the associated interpolation process can become quite complex as the number of dimensions increases.

Table 1 – *A typical set of tables for the elastic unloading of sand. In the table α is the water saturation, β is the volumetric plastic strain, ε is the total volumetric strain, and p is the pressure (mean stress).*

α	0						0.1			
β	0		0.06		0.14		0		0.07	
	ε	p	ε	p	ε	p	ε	p	ε	p
	0	0	0	55	0.062	155	0	0	0.001	150
	0.0003	41	0.0002	266	0.063	486	0.0001	50	0.002	250
	0.0005	77	0.0005	479	0.064	817	0.0005	101	0.003	400
	0.0008	113	0.0007	691	0.065	1150	0.001	205	…	…
	0.001	150	0.001	904	0.066	1480	0.0015	305	…	…
	…	…	…	…	…	…	…	…	0.08	20000
	0.027	16162	…	…	…	…	0.15	20000		
			0.081	16193	…	…				
					0.090	63152				

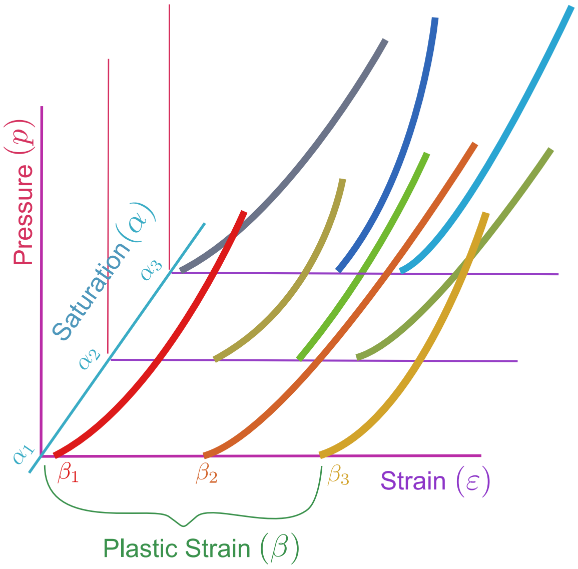


Figure 1 – *Schematic plot of tabular material data for a soil with elastic-plastic coupling. The circle in blue is the data point for which we would like to find the pressure.*

Mutivariable curve interpolation has a long history and was initially applied to data that could be mapped topologically to a planar rectangular grid (Ferguson, 1964). A more general approach that could be applied to parameterized curves was developed by Coons (Coons, 1967). Though the idea continues to be in widespread use for two-dimensional surfaces embedded in three dimensions, the original report explains how surfaces can be fitted and interpolated in higher dimensions. Mathematical proofs of the quality of approximation generated by Coons' blending functions have also been available for quite some time (Gordon, 1971).

A detailed survey and evaluation of algorithms for the interpolation of multivariate data by Franke (Franke, 1979; Franke, 1982) indicated that, at the time, there were six major classes of methods in use: inverse distance weighted interpolation, triangle-based blending, finite element interpolation, Foley's methods, nodal basis function methods, and Franke's spline based interpolation methods. However, Franke did not consider Coons' geometric approach based on parametric boundary curves, perhaps because a continuous interpolating parameterization of point data is expensive (Floater and Surazhsky, 2006).

Improvements in geometric interpolation have continued to be made (Peters, 1991; Barequet and Sharir, 1996; Lazarus, 1997; Dam, Koch, and Lillholm, 1998; Johan, Koiso, and Nishita, 2000; Goodman, 2001; Surazhsky and Elber, 2002). In particular, geodesic interpolation methods developed in recent years have led to high quality interpolants that are both mathematically accurate and visually satisfactory (Grazzini, Soille, and Bielski, 2007; Sprynski et al., 2008; Farouki, Szafran, and Biard, 2009). However, the focus of these methods has continued to be limited to three-dimensional data even though the domains of application have broadened to include not only CAD designs (Xu, Chen, and Feng, 2002) but also curve and image morphing (Surazhsky and Elber, 2002), reconstruction of three-dimensional scan data (Baloch et al., 2005), etc.

Since Franke's work, distance weighted methods have gained in popularity because of their ease of implementation. In particular, radial basis function methods (Broomhead and Lowe, 1988; Schaback, 1995; Buhmann, 2000) and kriging (Oliver and Webster, 1990; Remy et al., 2002; Gu, 2003) have exhibited clear advantages over parameterized geometry based method when applied to three or more dimensions. In this paper, we explore linear interpolation, radial basis functions, and kriging because these do not require meshing and can therefore be easily extended to higher dimensions. However, we have found that geometric interpolation methods work better for two- and three-dimensional data.

2.1 Linear interpolation

To illustrate the linear interpolation process let us consider the unknown function $p = p(\alpha, \varepsilon)$ where p is the pressure, and the two independent variables are the saturation (α) and the total strain (ε) as shown in Figure 2(a). We seek to find the value of the pressure (p_0) at the location (α_0, ε_0).

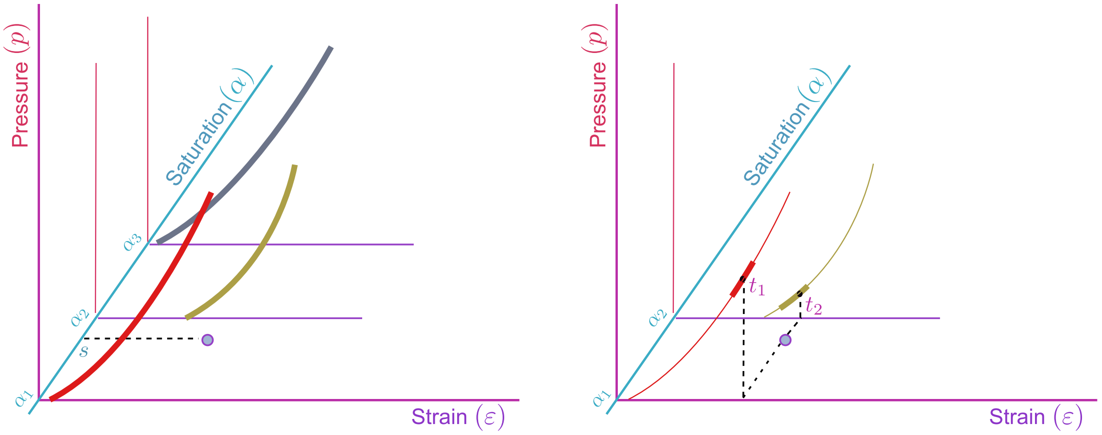
The procedure assumes that the α values are sorted in ascending order. If $\alpha_0 \notin [\alpha_1, \alpha_N]$, linear extrapolation is used to extend the data that have been provided. Notice that at least two sets of data are needed for the interpolation procedure to work.

The first step in the process is to search for the pressure-strain data from the ragged table that are needed for the interpolation process. This can be accomplished by iterating through the αs and finding a value of the parameter $s \in [0, 1]$ where

$$s = \frac{\alpha_0 - \alpha_k}{\alpha_{k+1} - \alpha_k} \ , \quad k = 1, 2, \dots, N-1 \tag{2}$$

where N is the number of values of α for which data are available. If α_0 is such that $s < 0$ or $s > 1$, the first two sets of curves or the last two sets in the ragged table are chosen for extrapolation.

Once the two curves needed for interpolation have been identified, the next step is to find the segments of the pressure-strain curves that correspond to the independent variable ε_0. These segments



(a) Stage 1.

(b) Stage 2.

Figure 2 – *Schematic of the first two steps of linear interpolation for a three variable table of material data. The circle in blue is the input point for which we would like to find the pressure.*

are highlighted with thick lines in Figure 2(b). The two associated parameters t_1 and t_2 are calculated using

$$\begin{aligned} t_1 &= \frac{\varepsilon_0 - \varepsilon_{j,k}}{\varepsilon_{j,k+1} - \varepsilon_{j,k}} \;, \quad k = 1, 2, \dots, M_j - 1 \\ t_2 &= \frac{\varepsilon_0 - \varepsilon_{j+1,k}}{\varepsilon_{j+1,k+1} - \varepsilon_{j+1,k}} \;, \quad k = 1, 2, \dots, M_{j+1} - 1 \end{aligned} \tag{3}$$

where $\varepsilon_{j,k}$ is a point on the pressure-strain curve for saturation α_j, and M_j is the number of points on the curve.

We can now compute the pressures at these two points, using

$$\begin{aligned} p_1 &= (1 - t_1)p_{j,k} + t_1 p_{j,k+1} \\ p_2 &= (1 - t_2)p_{j+1,k} + t_2 p_{j+1,k+1} \end{aligned} \tag{4}$$

The final step of the process is to compute the interpolated pressure p_0 using

$$p_0 = (1 - s)p_1 + s p_2 \,. \tag{5}$$

If the total strain exceeds the range of data provided in the tables, linear extrapolation can be used to estimate the pressure. An alternative is to use the end points of the curves as hard limits and interpolate using these end points. A schematic of this operation is shown in Figure 3.

This recursive approach can be extended to multiple dimensions. However, an implementation of the algorithm becomes nontrivial when the number of independent variables exceeds five. Also, the accuracy of interpolation decreases as the number of independent variables increases unless the data points correspond to Chebyshev nodes (Trefethen, 2019).

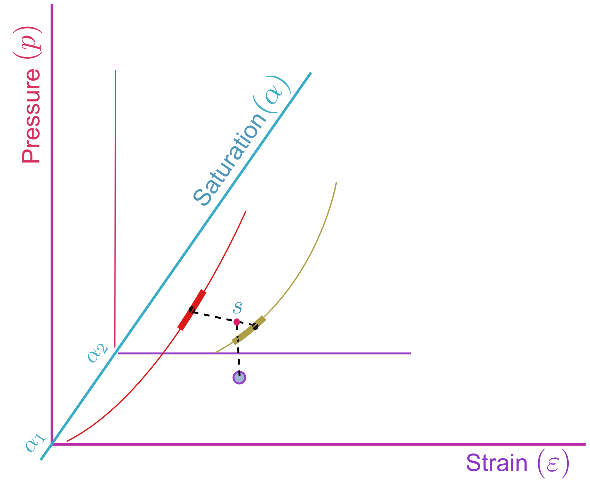


Figure 3 – *Final stage of interpolation of a three variable table of material data. The circle in blue is the input point and the red circle is the interpolated value.*

2.2 Radial basis function interpolation

The radial basis function interpolation process approximates a function as a sum of functions that are centered on a set of input data points. For example, $p = p(\mathbf{x})$ where $\mathbf{x} = (\alpha, \beta, \varepsilon)$ can be approximated as the function that interpolates a set of input points $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$:

$$p(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|) = \sum_{i=1}^{n} \lambda_i \varphi(\|\mathbf{r}_i\|) = \sum_{i=1}^{n} \lambda_i \varphi(r_i) , \quad \|\mathbf{r}\| = r := \sqrt{\mathbf{r} \cdot \mathbf{r}} . \tag{6}$$

With this approximation, the interpolation problem reduces to finding the weights (λ_i) corresponding to a set of input points and a choice of radial basis function (φ). Radial basis functions that are multiquadric can be efficiently evaluated and a common form is

$$\varphi(\|\mathbf{x} - \mathbf{x}_i\|) = \varphi(r) = \sqrt{1 + (\xi r)^2} \tag{7}$$

where ξ is a parameter that can be varied to increase or decrease the region of support of the basis function.

The weights (λ_i) can be found by solving a system of equations that is formed by evaluating (6) at the input points such that

$$p(\mathbf{x}_j) = \sum_{i=1}^{n} \lambda_i \varphi(\|\mathbf{x}_j - \mathbf{x}_i\|) = \sum_{i=1}^{n} \lambda_i \varphi(r_{ji}) , \quad j = 1, \dots, n \tag{8}$$

The resulting system of equations has the form

$$\begin{bmatrix} \varphi(r_{11}) & \varphi(r_{21}) & \dots & \varphi(r_{n1}) \\ \varphi(r_{12}) & \varphi(r_{22}) & \dots & \varphi(r_{n2}) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(r_{1n}) & \varphi(r_{2n}) & \dots & \varphi(r_{nn}) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} p(\mathbf{x}_1) \\ p(\mathbf{x}_2) \\ \vdots \\ p(\mathbf{x}_n) \end{bmatrix} \tag{9}$$

As long as points $\mathbf{x}_i$ are not repeated, the system is non-singular. However, in some situations the matrix can have a small condition number. The matrix is also dense and can become large if the number of input points is large. Once the values of λ_i have been found, the value of p at a given location $\mathbf{x}_0 = (\alpha_0, \beta_0, \varepsilon_0)$ is computed using

$$p(\mathbf{x}_0) = \sum_{i=1}^{n} \lambda_i \varphi(\|\mathbf{x}_0 - \mathbf{x}_i\|) \,. \tag{10}$$

The problem of inversion of large matrices, as well as the associated cost of evaluating a sum of a large number of functions, can be ameliorated by reducing the number of inputs points and performing a local approximation. For tabular data of the form of interest in this work, we can represent each input curve as a k-dimensional tree (kd-tree) (Moore, 1991). Nearest neighbors to these curves can then be found efficiently. A small set of nearest neighbors on each curve surrounding an interpolation point ($\mathbf{x}_0$) can then be used to form and solve equation (9) as depicted in the schematic in Figure 4.

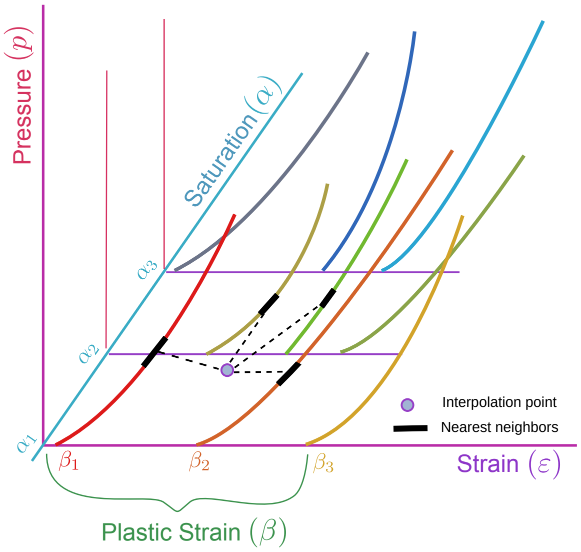


Figure 4 – *Nearest neighbor based interpolation used in radial basis function and kriging interpolation.*

While the approach shown in the figure is efficient, it suffers from the problem of discontinuous derivatives because of the local nature of the approximation. However, this issue can be eliminated if the derivatives of the input curves are also interpolated.

2.3 Kriging interpolation

Kriging also involves a distance-weighted approximation and interpolates using a function similar to equation (6):

$$p(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i p(\mathbf{x}_i) \tag{11}$$

If the additional constraint

$$\sum_{i=1}^{n} \lambda_i = 1 \tag{12}$$

is applied, then the process is called "ordinary" kriging. The kriging method can be classified as a Gaussian process model (Schobi, Sudret, and Wiart, 2015).

To find the weights (λ_i), kriging treats the variable p as the realization of a random variable $P(\mathbf{x})$ in the space spanned by $\mathbf{x}$. With that interpretation, the problem of finding λ_i can be expressed as a least squares minimization problem:

$$\underset{\boldsymbol{\lambda}}{\text{minimize}} \quad \text{Var}\left[P^\star(\mathbf{x}) - P(\mathbf{x})\right] \tag{13}$$

where $\text{Var}(\cdot)$ is the variance, $P^\star(\mathbf{x})$ is the estimate, and $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_n)$. Note that mean value terms are often included in the objective function to arrive at equations that involve covariance matrices. However these mean values cancel out in ordinary kriging.

The objective can also be expressed in terms of expected values,

$$\underset{\boldsymbol{\lambda}}{\text{minimize}} \quad \mathbb{E}\left[\left(P^\star(\mathbf{x}) - P(\mathbf{x}) - \mathbb{E}\left[P^\star(\mathbf{x}) - P(\mathbf{x})\right]\right)^2\right] \tag{14}$$

where $\mathbb{E}(\cdot)$ indicates the expected value. Alternatively,

$$\underset{\boldsymbol{\lambda}}{\text{minimize}} \quad \mathbb{E}\left[\left(P^\star(\mathbf{x}) - P(\mathbf{x})\right)^2\right] - \left(\mathbb{E}\left[P^\star(\mathbf{x}) - P(\mathbf{x})\right]\right)^2 \tag{15}$$

With the unbiasedness constraint, $\mathbb{E}[P^\star(\mathbf{x})] = \mathbb{E}[P(\mathbf{x})]$, the problem simplifies to

$$\underset{\boldsymbol{\lambda}}{\text{minimize}} \quad \mathbb{E}\left[\left(P^\star(\mathbf{x}) - P(\mathbf{x})\right)^2\right] \tag{16}$$

To minimize the objective function, we take derivatives with respect to $\boldsymbol{\lambda}$ and set the values to zero:

$$\frac{\partial I}{\partial \boldsymbol{\lambda}} := \frac{\partial}{\partial \boldsymbol{\lambda}} \mathbb{E}\left[\left(P^\star(\mathbf{x}) - P(\mathbf{x})\right)^2\right] = 2\mathbb{E}\left[\left(P^\star(\mathbf{x}) - P(\mathbf{x})\right) \frac{\partial}{\partial \boldsymbol{\lambda}}\left(P^\star(\mathbf{x}) - P(\mathbf{x})\right)\right] = \mathbf{0} \tag{17}$$

Using the same notation, the kriging approximation (11) can be written as

$$P^\star(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i P(\mathbf{x}_i) =: \boldsymbol{\lambda} \cdot \mathbf{P} \tag{18}$$

where $\mathbf{P} = (P(\mathbf{x}_1), P(\mathbf{x}_2), \dots, P(\mathbf{x}_n))$. Substitution into (17) gives

$$\mathbb{E}\left[\left(\boldsymbol{\lambda} \cdot \mathbf{P} - P(\mathbf{x})\right) \frac{\partial}{\partial \boldsymbol{\lambda}}\left(\boldsymbol{\lambda} \cdot \mathbf{P} - P(\mathbf{x})\right)\right] = \mathbf{0} \tag{19}$$

or

$$\mathbb{E}\left[\left(\boldsymbol{\lambda} \cdot \mathbf{P} - P(\mathbf{x})\right) \mathbf{P}\right] = \mathbf{0}\,. \tag{20}$$

Reorganizing,

$$\mathbf{E}[\boldsymbol{P} \otimes \boldsymbol{P}] \cdot \boldsymbol{\lambda} = \mathbb{E}[P(\mathbf{x})\boldsymbol{P}] \quad \text{where} \quad (\boldsymbol{P} \otimes \boldsymbol{P})_{ij} = P(\mathbf{x}_i)P(\mathbf{x}_j)\,. \tag{21}$$

This system can be used to solve for the weights (λ_i) given a set of input points ($\mathbf{x}_i$) and an interpolation point ($\mathbf{x}_0$). Expressed in matrix form:

$$\begin{bmatrix} \mathbb{E}[P(\mathbf{x}_1)P(\mathbf{x}_1)] & \mathbb{E}[P(\mathbf{x}_1)P(\mathbf{x}_2)] & \dots & \mathbb{E}[P(\mathbf{x}_1)P(\mathbf{x}_n)] & 1 \\ \mathbb{E}[P(\mathbf{x}_2)P(\mathbf{x}_1)] & \mathbb{E}[P(\mathbf{x}_2)P(\mathbf{x}_2)] & \dots & \mathbb{E}[P(\mathbf{x}_2)P(\mathbf{x}_n)] & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbb{E}[P(\mathbf{x}_n)P(\mathbf{x}_1)] & \mathbb{E}[P(\mathbf{x}_n)P(\mathbf{x}_2)] & \dots & \mathbb{E}[P(\mathbf{x}_n)P(\mathbf{x}_n)] & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ \mu \end{bmatrix} = \begin{bmatrix} \mathbb{E}[P(\mathbf{x}_0)P(\mathbf{x}_1)] \\ \mathbb{E}[P(\mathbf{x}_0)P(\mathbf{x}_2)] \\ \vdots \\ \mathbb{E}[P(\mathbf{x}_0)P(\mathbf{x}_n)] \\ 1 \end{bmatrix} \tag{22}$$

where μ is a dummy parameter used to enforce the constraint on λ. The quantities $\mathbb{E}[P(\mathbf{x}_i)P(\mathbf{x}_j)]$ are called semi-variances and are typically estimated using variograms, defined as (Matheron, 1967)

$$\gamma(\mathbf{h}) = \gamma(\mathbf{x}_i, \mathbf{x}_j) = \tfrac{1}{2}\mathbb{E}[P(\mathbf{x}_j) - P(\mathbf{x}_i)]^2 \,, \quad \mathbf{h} = \mathbf{x}_j - \mathbf{x}_i \,. \tag{23}$$

A discrete model that can be used to estimate semi-variances is the Matheron estimator

$$\hat{\gamma}(\mathbf{h}) = \frac{1}{2N_h}\sum_{i=1}^{N_h}(P(\mathbf{x}_i + \mathbf{h}) - P(\mathbf{x}_i))^2 \quad \text{where} \quad N_h = \#N(\mathbf{h}) \,, \;\; N(\mathbf{h}) = \left\{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_j - \mathbf{x}_i = \mathbf{h}\right\} \,. \tag{24}$$

Since few pairs of points will occur at exactly the same distance, $\|\mathbf{h}\|$, a binning procedure is typically applied and points that fall in each bin are summed. A continuous model is then typically fitted to the discrete model using a least squares curve fit and used for the interpolation process. For example, the spherical variogram model has the form (Cressie, 1985)

$$\gamma(\mathbf{h}) = \begin{cases} C_0 + C_s\left[\frac{3}{2}\frac{\|\mathbf{h}\|}{A_s} - \frac{1}{2}\left(\frac{\|\mathbf{h}\|}{A_s}\right)^3\right] & \text{for} \quad 0 < \|\mathbf{h}\| \le A_s \\ C_0 + C_s & \text{for} \quad \|\mathbf{h}\| \ge A_s \end{cases} \tag{25}$$

where C_0 is called the nugget, C_s the sill, and A_s is an effective range. With these definitions, the kriging problem reduces to solving the system of equations

$$\begin{bmatrix} \gamma(\mathbf{x}_1, \mathbf{x}_1) & \gamma(\mathbf{x}_1, \mathbf{x}_2) & \dots & \gamma(\mathbf{x}_1, \mathbf{x}_n)] & 1 \\ \gamma(\mathbf{x}_2, \mathbf{x}_1) & \gamma(\mathbf{x}_2, \mathbf{x}_2) & \dots & \gamma(\mathbf{x}_2, \mathbf{x}_n)] & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(\mathbf{x}_n, \mathbf{x}_1) & \gamma(\mathbf{x}_n, \mathbf{x}_2) & \dots & \gamma(\mathbf{x}_n, \mathbf{x}_n) & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \\ \mu \end{bmatrix} = \begin{bmatrix} \gamma(\mathbf{x}_0, \mathbf{x}_1) \\ \gamma(\mathbf{x}_0, \mathbf{x}_2) \\ \vdots \\ \gamma(\mathbf{x}_0, \mathbf{x}_n) \\ 1 \end{bmatrix} \tag{26}$$

Once the value of λ_i have been found, the interpolated value $p_0 = p(\mathbf{x}_0)$ is estimated using

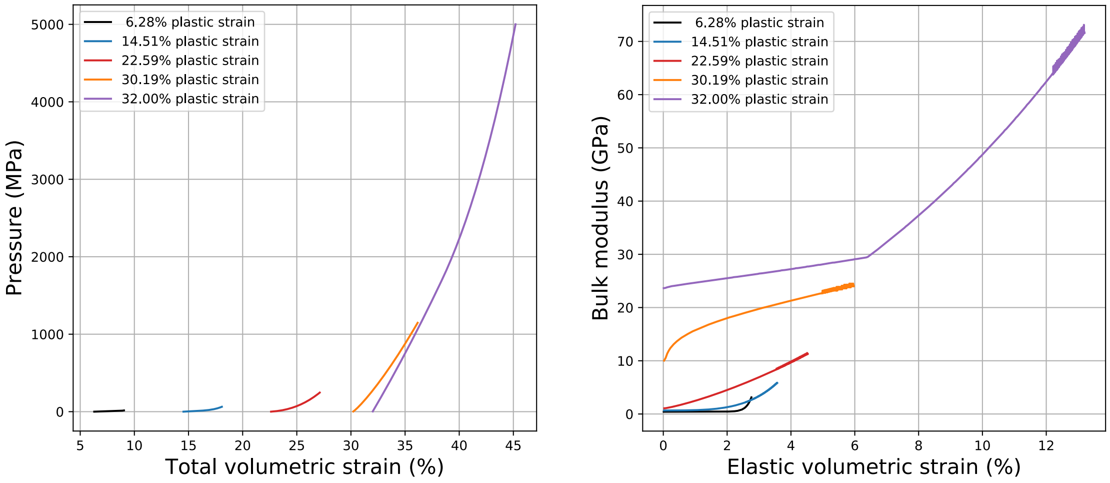
$$p(\mathbf{x}_0) = \sum_{i=1}^{n} \lambda_i p(\mathbf{x}_i) \,. \tag{27}$$

The kriging interpolator, like the radial basis function one, can be either applied in the global sense to the full input data or to local patches of the data as shown in Figure 4. In the local case, the procedure for finding the nearest neighbors is identical to that for radial basis function interpolation. The caveats discussed previously for computing derivatives of local radial basis function approximations also apply to local kriging interpolation.

3 Experimental data

The experimental data that have been used as a test-bed for the interpolation process are for a dry, poorly-graded, concrete sand described by Fox et al. (2014) and tested at the University of Maryland.[1] Further details on the particular set used in this work can be found in (Banerjee, Fox, and Regueiro, 2020b).

The hydrostatic elastic unloading data for that sand can be seen in Figure 5(a). Tangent bulk moduli at intermediate values of volumetric elastic and plastic strains are computed by interpolating between the input data. Tangents to the unloading curves represent the bulk modulus and have been plotted in Figure 5(b).



(a) Elastic unloading curves

(b) Tangent bulk modulus curves

Figure 5 – *Hydrostatic elastic unloading data and the corresponding tangent bulk moduli for dry poorly-graded concrete sand.*

As discussed in Banerjee, Fox, and Regueiro (2020a), the data are extended and padded prior to interpolation to prevent nonphysical behavior outside the available range. The extended unloading and bulk modulus curves are shown in Figure 6.

4 Interpolation of experimental data

4.1 Linear interpolation

The process of linear interpolation of the experimental tabular data is the most straightforward, though a generalized implementation in a host code becomes complex when many independent variables are involved. For the two variable data discussed in the previous section, the linear interpolation process leads to the pressure vs. strain predictions shown in Figure 7. If we examine the curves predicted for volumetric plastic strains (ε_v^p) of 10% and 20%, we notice that the interpolation

[1] Stephen Akers, 2018, Private communication, CCDC Army Research Laboratory, Aberdeen Proving Ground, MD, USA

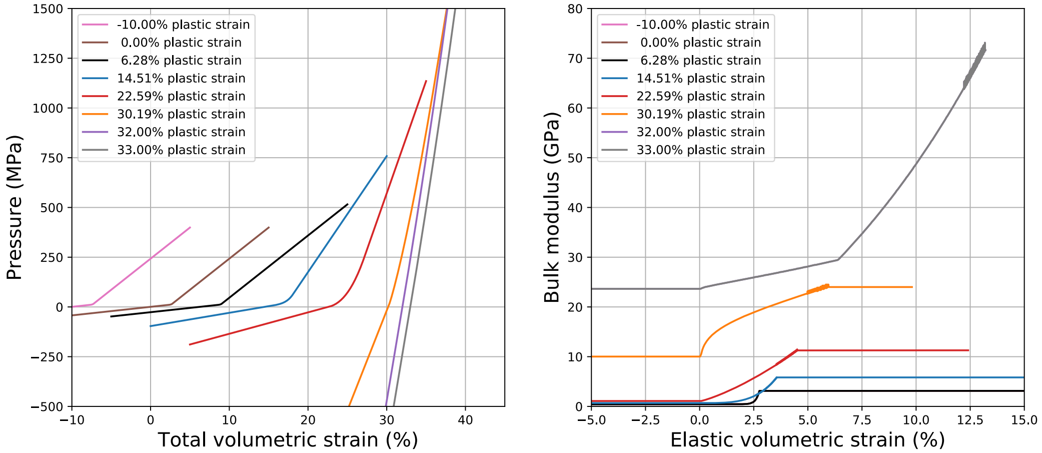


Figure 6 – *Hydrostatic elastic unloading data and the corresponding tangent bulk moduli for dry poorly-graded concrete sand.*

does not appear to be accurate for a given value of pressure, particularly at low pressures. This apparent error is caused by the fact that the curvatures of the two adjacent input curves are not identical for a given value of the total volumetric strain. A more visually appealing interpolation can be found if we use pressure as an independent variable. However, the fundamental problem still remains.

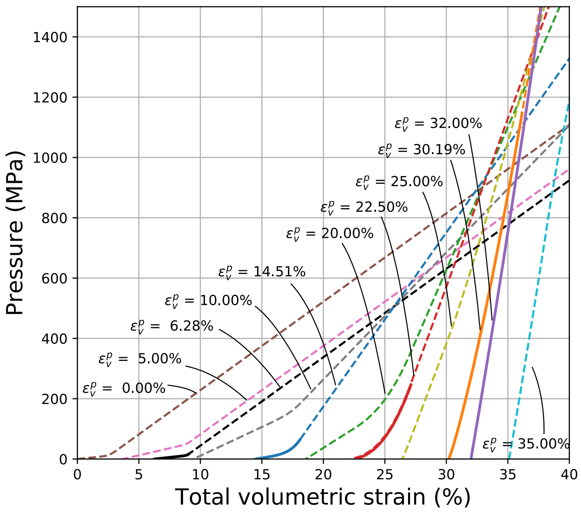
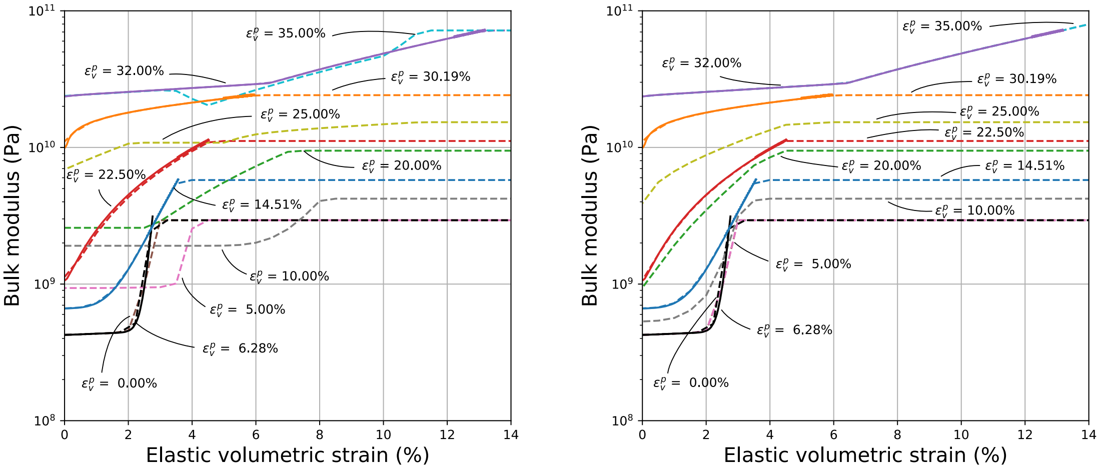


Figure 7 – *Linear interpolation of the pressure vs. total strain data for a dry poorly-graded concrete sand. Predictions for various values of volumetric plastic strain (ε_v^p) are shown as dashed lines while the solid lines are the input experimental data.*

Since we are concerned with computing the bulk modulus rather than the pressure, it is worth comparing the bulk moduli computed from the data shown in Figure 7 with those from direct linear interpolation of input bulk moduli curves. Figure 8(a) shows the moduli computed from interpolated pressures while Figure 8(b) shows directly interpolated moduli. The moduli computed from interpolated pressures are inaccurate at smaller values of strain and sometimes deviate from monotonic behavior. On the other hand, directly interpolated moduli behave much better because derivatives do not need to be computed.



(a) Computed from interpolated pressures.

(b) Directly interpolated moduli.

Figure 8 – *Predicted bulk moduli for a dry poorly-graded concrete sand using linear interpolation. Predictions are shown as dashed lines and the input data as solid lines.*

4.2 Radial basis function interpolation

When using radial basis functions, if the number of input data points is relatively small, a global distance matrix (see equation (9)) can be formed and used to compute the weights needed for interpolation. However, that matrix can be quite large and even though it has to be inverted only once, each interpolated point requires a sum over all input points (equation (10)) many of which contribute little to the interpolated value.

An alternative is to select a subset of the nearest neighbors to the interpolation point ($\mathbf{x}_0$) using a distance-based criterion. This reduces the matrix size. However, a matrix has to be created and inverted for each value of $\mathbf{x}_0$. Also, even though the matrix is not singular, the condition number can be small and a Moore-Penrose pseudo-inverse often has to be computed.

Figure 9 shows the predicted values of pressure using radial basis functions to interpolate between the input curves. Nearest neighbors are determined based on all the input data and a minimum of 100 data points were chosen. The plots of pressure as a function of elastic volumetric strain in the figure show that the interpolation is not smooth and significant fluctuations occur when there are insufficient nearby data points for interpolation. This issue is also clearly evident in the plots of pressure vs. total volumetric strain. As a consequence, the bulk moduli computed from these interpolated curves also shown significant errors and cannot be used in numerical simulations.

An improved interpolation can be computed if the two closest curves to a the interpolation point are found and then nearest neighbors on both curves are determined as depicted in Figure 4. The two

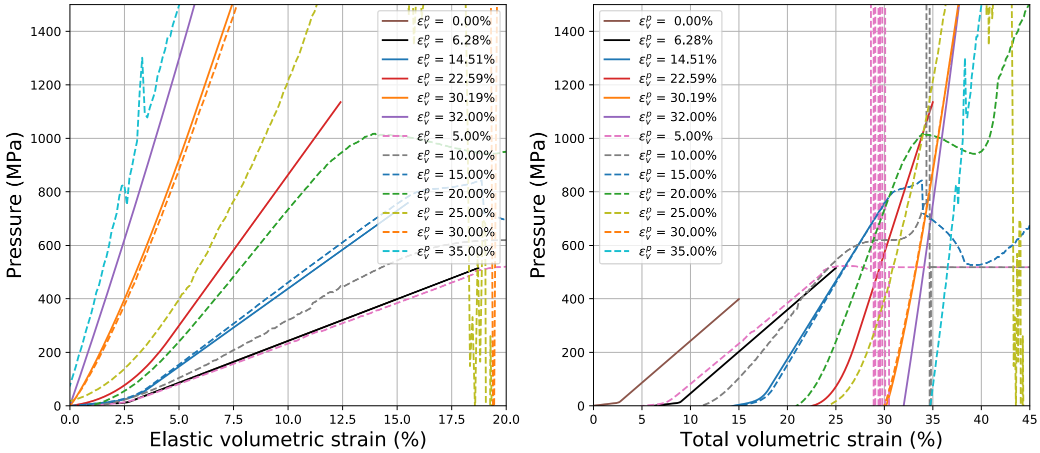


Figure 9 – *Radial basic function interpolation of the pressure vs. strain data for a dry poorly-graded concrete sand. All of the input data were used for nearest neighbor searches. Solid curves represent the input data.*

closest curves are determined based on the volumetric plastic strain value to be interpolated. The outcome of that process is shown in Figure 10. Though the interpolated curves are smoother than those in Figure 9, there are jumps in the values in regions where sufficient data are not available for accurate interpolation.

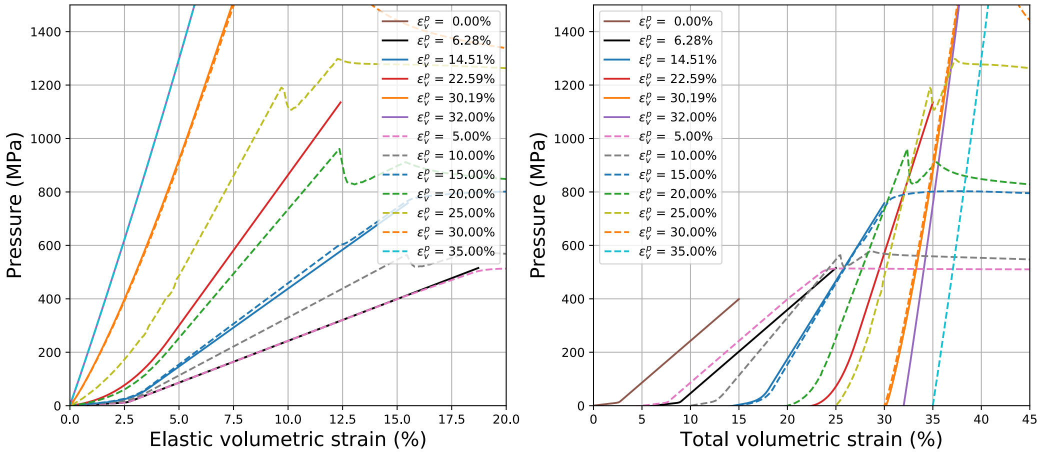
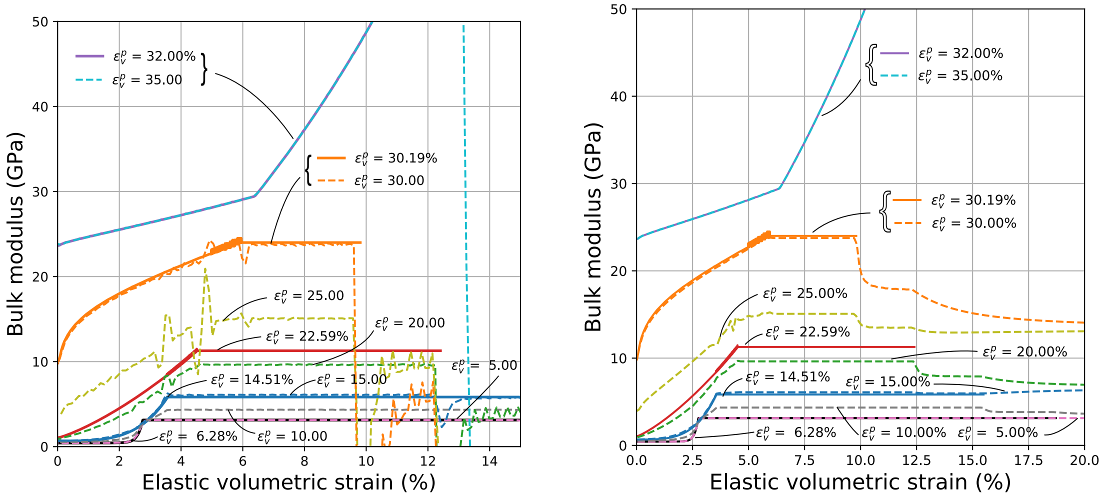


Figure 10 – *Radial basic function interpolation of the pressure vs. strain data for a dry poorly-graded concrete sand. Two closest input data curves were used for nearest neighbor searches. Solid curves represent the input data.*

Figure 11(a) shows the bulk moduli computed from the pressure-strain curves in Figure 10. The predicted bulk moduli do not vary smoothly and jumps are observed at several points. On the other hand, if the radial basic interpolation process is applied directly to bulk modulus data, predicted values vary more smoothly (see Figure 11(b)). Though there are some jumps in the predicted bulk moduli, and the predicted values outside the range of input data are dubious, the interpolated values are reasonable and can be used in simulations.



(a) Computed from interpolated pressures.

(b) Directly interpolated moduli.

Figure 11 – *Predicted bulk moduli for a dry poorly-graded concrete sand using radial basis function interpolation. Both plots have used nearest neighbors from the two curves closest to the plastic volumetric strain to be interpolated. Predictions are shown as dashed lines and the input data as solid lines.*

4.3 Kriging interpolation

If we replace the radial basis function interpolation process with kriging, and use the entire input data set for nearest neighbor searches, the predicted pressure-strain curves are as seen in Figure 12. The predicted curves are smoother than those computed using radial basis functions and the interpolated values appear reasonable. However, the curves tend to flatten out in regions where sufficient nearest neighbor data are not available.

Instead, if we use nearest neighbors from the two curves closest to an interpolated value of volumetric plastic strain, we observe the interpolated curves shown in Figure 13. The interpolated curves from this process exhibit a slight improvement over those in Figure 12. However, the bulk moduli computed from the interpolated pressure curves show cyclic behavior with periods that depend on the number of nearest neighbors used in kriging and the distance between adjacent curves. This variability is problematic for coupled elastic-plasticity because derivatives of the bulk modulus are needed in the associated stress update algorithms.

The variability in the bulk moduli computed from interpolated pressure-strain curves can be avoided if the bulk moduli are computed from the input data before interpolation. In Figure 14(a), interpolated values of bulk moduli are shown for the cases where nearest neighbors are selected from the entire input data set. The predicted curves are observed to be smooth even in regions where rapid changes in slope are observed in the input data. On the other hand, if nearest neighbors are com-

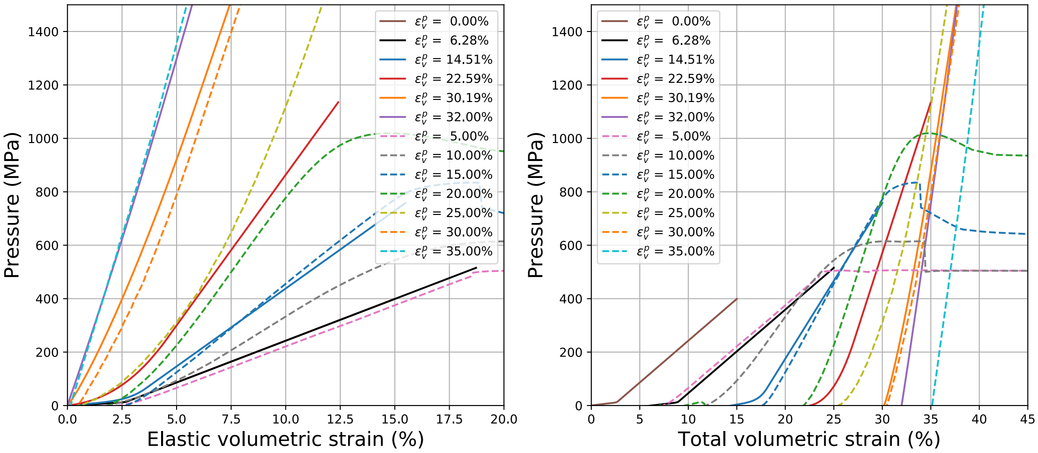


Figure 12 – *Kriging predictions of pressure for a dry poorly-graded concrete sand. Nearest neighbors have been computed from the entire input data set, shown as solid lines.*

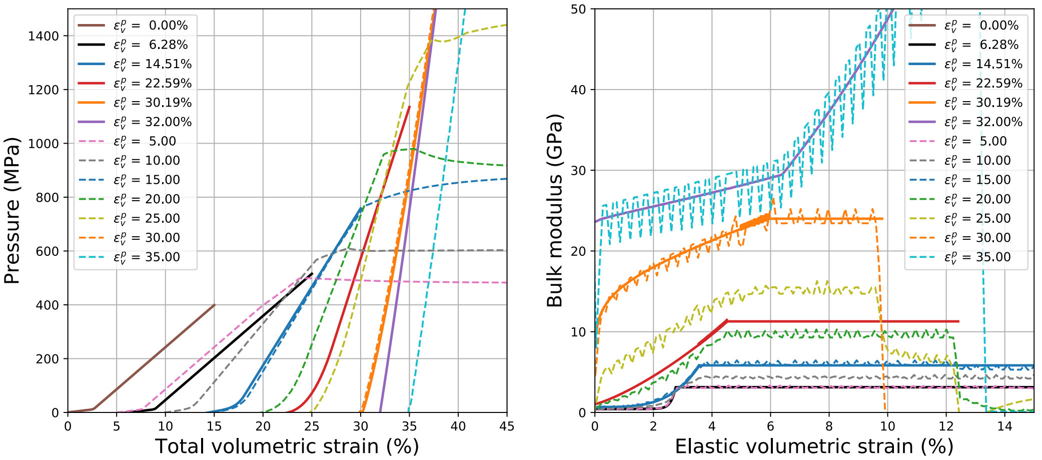
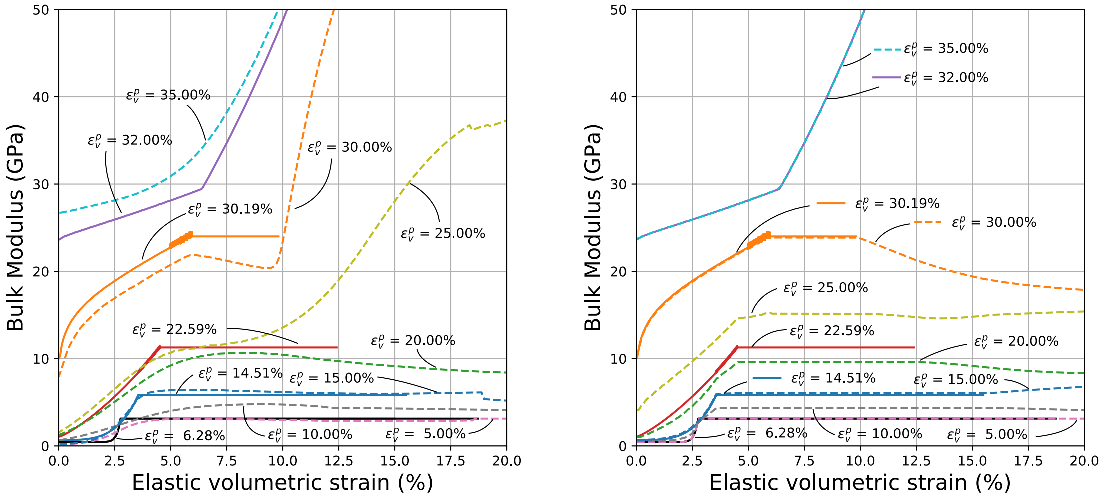


Figure 13 – *Kriging-based interpolated data (pressure and bulk modulus) a dry poorly-graded concrete sand using the two input curves closest to an interpolation point. Predictions are shown as dashed lines and the input data as solid lines.*

puted only from two curves adjacent to the interpolation point, we observe the predictions shown in Figure 14(b). The predicted values are better behaved and reasonably accurate in this case.



(a) Nearest neighbors from all input data.

(b) Nearest neighbors from two closest curves.

Figure 14 – *Kriging interpolation of precomputed bulk moduli for a dry poorly-graded concrete sand. The input data are depicted with solid lines.*

5 Concluding remarks

Linear interpolation is an efficient and accurate method of interpolating tabular data that can be represented as $y = f(x)$, where x is a scalar independent variable. If the data can be arranged such that the values of x can be sorted without changing the function f, a binary search can be used to locate a point in $O(\log n)$ where n is the number of data points.

However, if $y = f(\mathbf{x})$ where $\mathbf{x}$ is a vector of independent variables, not only is interpolation procedure not unique but the accuracy of the interpolation degrades with increasing dimensions. We have discussed one approach for linear interpolation in this work and examined the bulk moduli for a dry poorly-grade concrete sand predicted by this approach. Our results show that better interpolations are produced when, instead of derivatives of interpolated data, direct interpolation of derivatives are used.

Linear interpolation can potentially be improved upon by the use of radial basis function interpolations. We have found that, for the sand data set, radial basis functions lead to poorer predicted values when a fixed search distance is used to find nearest neighbors. Instead, if we select closest points from a set of nearest data curves, a superior result is obtained. Also, similar to what was observed for linear interpolation, direct interpolations of derivatives produce better estimates of bulk moduli.

Finally, we have explored kriging-based interpolations of the sand data. The interpolations produced by kriging are quite accurate whether nearest neighbors from the entire data set or nearest neighbors from the closest curves are used. However, derivatives of interpolated curves are not smooth. If kriging interpolation is applied to bulk moduli curves computed from the input data, interpolation of data from the closest curves appear to produce better bulk modulus predictions.

The theoretical accuracy of kriging interpolations are known to be better than those for linear interpolation, unless the input data are at Chebyshev points, But visual examination of our results does not indicate any significant reason for choosing one over the other when data from only the closest curves are used for interpolation. On the other hand, as the number of independent variables increases kriging becomes preferable because the implementation effort remains small while accuracy improves over linear interpolation.

Acknowledgements

This research has been partially funded by the US Office of Naval Research PTE Federal award number N00014-17-1-2704. The first author is also indebted to anonymous commentors on an earlier version of this work who pointed out the applicability of kriging to the process.

References

Baloch, SH et al. (2005). "3D curve interpolation and object reconstruction". In: *IEEE International Conference on Image Processing 2005*. Vol. 2. IEEE, pp. II–982 (cit. on p. 4).

Banerjee, B. and R. M. Brannon (2019). "Continuum modeling of partially saturated soils". In: *Shock Phenomena in Granular and Porous Materials*. Ed. by T. Vogler and A. Fredenburg. Springer (cit. on p. 2).

Banerjee, B., D. M. Fox, and R. A. Regueiro (2020a). *Multilayer perceptron neural networks as multi-variable material models*. Tech. rep. PAR-10021867-092020-2. Parresia Research Limited (cit. on pp. 2, 10).

— (2020b). *Support vector regression for fitting multi-variable material models*. Tech. rep. PAR-10021867-092020-1. Parresia Research Limited (cit. on pp. 2, 10).

Barequet, Gill and Micha Sharir (1996). "Piecewise-linear interpolation between polygonal slices". In: *Computer vision and image understanding* 63.2, pp. 251–272 (cit. on p. 4).

Brannon, R. M. et al. (2015). "KAYENTA: Theory and User's Guide". In: *Sandia National Laboratories report SAND2015-0803* (cit. on p. 2).

Broomhead, David S and David Lowe (1988). *Radial basis functions, multi-variable functional interpolation and adaptive networks*. Tech. rep. Royal Signals and Radar Establishment Malvern (United Kingdom) (cit. on p. 4).

Browne, Shirley et al. (1995). "The Netlib mathematical software repository". In: *D-lib Magazine* 1.9 (cit. on p. 2).

Buhmann, Martin D (2000). "Radial basis functions". In: *Acta numerica* 9, pp. 1–38 (cit. on p. 4).

Coons, Steven A (1967). *Surfaces for computer-aided design of space forms*. Tech. rep. MASSACHUSETTS INST OF TECH CAMBRIDGE PROJECT MAC (cit. on p. 3).

Cressie, Noel (1985). "Fitting variogram models by weighted least squares". In: *Journal of the international Association for mathematical Geology* 17.5, pp. 563–586 (cit. on p. 9).

Dam, Erik B, Martin Koch, and Martin Lillholm (1998). *Quaternions, interpolation and animation*. Vol. 2. Citeseer (cit. on p. 4).

Farouki, Rida T, Nicolas Szafran, and Luc Biard (2009). “Existence conditions for Coons patches interpolating geodesic boundary curves”. In: *Computer Aided Geometric Design* 26.5, pp. 599–614 (cit. on p. 4).

Ferguson, James (1964). “Multivariable curve interpolation”. In: *Journal of the ACM (JACM)* 11.2, pp. 221–228 (cit. on p. 3).

Floater, Michael S and Tatiana Surazhsky (2006). “Parameterization for curve interpolation”. In: *Studies in Computational Mathematics*. Vol. 12. Elsevier, pp. 39–54 (cit. on p. 4).

Fox, D. M. et al. (2014). “The effects of air filled voids and water content on the momentum transferred from a shallow buried explosive to a rigid target”. In: *International Journal of Impact Engineering* 69, pp. 182–193 (cit. on p. 10).

Franke, Richard (1979). “A critical comparison of some methods for interpolation of scattered data”. In: *U.S. Naval Postgraduate School, Technology Report NPS-53-79-003* (cit. on p. 4).

— (1982). “Scattered data interpolation: tests of some methods”. In: *Mathematics of computation* 38.157, pp. 181–200 (cit. on p. 4).

Goodman, T (2001). *Shape preserving interpolation by curves. In. Algorithms For Approximation IV. Proceedings of the 2001 International Symposium*. Tech. rep. DUNDEE UNIV (UNITED KINGDOM) DEPT OF MATHEMATICS (cit. on p. 4).

Gordon, William J (1971). “Blending-function methods of bivariate and multivariate interpolation and approximation”. In: *SIAM journal on numerical analysis* 8.1, pp. 158–177 (cit. on p. 3).

Grazzini, Jacopo, Pierre Soille, and Conrad Bielski (2007). “On the use of geodesic distances for spatial interpolation”. In: *Proceedings of the 9th International Conference on GeoComputation, GeoComputation, Maynooth, Ireland*. Citeseer (cit. on p. 4).

Gu, Lei (2003). “Moving kriging interpolation and element-free Galerkin method”. In: *International journal for numerical methods in engineering* 56.1, pp. 1–11 (cit. on p. 4).

Johan, Henry, Yuichi Koiso, and Tomoyuki Nishita (2000). “Morphing using curves and shape interpolation techniques”. In: *Proceedings the Eighth Pacific Conference on Computer Graphics and Applications*. IEEE, pp. 348–454 (cit. on p. 4).

Lazarus, Francis (1997). “Smooth interpolation between two polylines in space”. In: *Computer-aided design* 29.3, pp. 189–196 (cit. on p. 4).

Matheron, G (1967). “Kriging or polynomial interpolation procedures”. In: *CIMM Transactions* 70, pp. 240–244 (cit. on p. 9).

Moore, AW (1991). “An introductory tutorial on kd-trees. Extract from PhD”. PhD thesis. Cambridge University (cit. on p. 7).

Oliver, Margaret A and Richard Webster (1990). “Kriging: a method of interpolation for geographical information systems”. In: *International Journal of Geographical Information System* 4.3, pp. 313–332 (cit. on p. 4).

Peters, Jörg (1991). “Smooth interpolation of a mesh of curves”. In: *Constructive Approximation* 7.1, pp. 221–246 (cit. on p. 4).

Remy, Nicolas et al. (2002). “GsTL: The geostatistical template library in C++”. In: *Computers & Geosciences* 28.8, pp. 971–979 (cit. on p. 4).

Schaback, Robert (1995). “Creating surfaces from scattered data using radial basis functions”. In: *Mathematical methods for curves and surfaces* 477 (cit. on p. 4).

Schobi, Roland, Bruno Sudret, and Joe Wiart (2015). “Polynomial-chaos-based Kriging”. In: *International Journal for Uncertainty Quantification* 5.2 (cit. on p. 8).

Sprynski, Nathalie et al. (2008). “Surface reconstruction via geodesic interpolation”. In: *Computer-Aided Design* 40.4, pp. 480–492 (cit. on p. 4).

Surazhsky, Tatiana and Gershon Elber (2002). “Metamorphosis of planar parametric curves via curvature interpolation”. In: *International Journal of Shape Modeling* 8.02, pp. 201–216 (cit. on p. 4).

Trefethen, Lloyd N (2019). *Approximation theory and approximation practice*. Vol. 164. Siam (cit. on p. 5).

Xu, Zhiming, Jincheng Chen, and Zhengjin Feng (2002). “Performance evaluation of a real-time interpolation algorithm for NURBS curves”. In: *The International Journal of Advanced Manufacturing Technology* 20.4, pp. 270–276 (cit. on p. 4).