Received: 2 July 2020 | Revised: 11 September 2020 | Accepted: 14 September 2020

DOI: 10.1002/nme.6553

RESEARCH ARTICLE

WILEY

A strong form based moving Kriging collocation method for the numerical solution of partial differential equations with mixed boundary conditions

Felix S. Bott | **Michael W. Gee**

Mechanics and High Performance Computing Group, Technical University of Munich, Munich, Germany

Correspondence
Michael W. Gee, Technical University of Munich, Parkring 35, Garching, 85748 Munich, Germany.
Email: gee@tum.de

Abstract

In this article, a novel mesh-free, moving Kriging (MK) based collocation scheme for the numerical solution of partial differential equations (PDEs) is introduced. In contrast to methods that are based on a Galerkin weak form of the governing PDEs, the MK collocation (MKC) approach, which is strong form based, is truly mesh-free in the sense that no background mesh is required for numerical integration. In fact, the presented approach does not require the evaluation of any integrals. Since the approximation function in the MK framework can be conditioned on point value- as well as derivative-information, the pointwise exact imposition of essential as well as natural boundary conditions is rendered straightforward. By incorporating an explicit linear basis into the MK framework, the first-order consistency condition is fulfilled, and thus rigid body motions are captured accurately. Moreover, Kriging functions may be conceived that comply with constraints on higher order derivatives such as the PDE at hand at certain locations. This possibility proves useful in improving the solution accuracy in the vicinity of Dirichlet boundaries. This article provides a study of the method's characteristics by means of 2D linear elasticity examples. It concludes with a suggestion on how to apply MKC to nonlinear PDEs.

KEYWORDS

linear elasticity, mesh-free methods, moving Kriging collocation, solid mechanics

1 | INTRODUCTION

We introduce a Kriging[1,2] based mesh free method for the numerical solution of partial differential equations (PDEs). The objective is to tackle a persisting problem that is shared by all common mesh free discretization schemes: A precise and robust technique for imposing boundary conditions (BCs). The presented scheme will allow for a pointwise exact imposition of essential (Dirichlet) as well as natural (Neumann) BCs. It is at least first order complete, that is, it is able to recover linear polynomials exactly. In, for example, solid mechanics problems, this property is essential to reproduce rigid body motions exactly.

[The copyright line for this article was changed on 22 December 2020 after original online publication.]

Kriging techniques have already been used to construct mesh free discretization schemes.[3-7] Here, we employ Kriging techniques the first time to solve the strong forms of PDEs describing mechanical systems. In particular, the novelty of the presented method lies in the way how BCs are imposed, that is by defining the space in which an approximation is searched for to contain only those functions that a priori comply with the BCs. Interestingly, this space can be restricted further to comprise only those functions that also satisfy the PDE itself at discrete locations on the Dirichlet boundary prior to solving a global system of equations. This possibility proves to be very useful in improving the accuracy of the approximation on and in the vicinity of Dirichlet boundaries.

The development of a collocation method with the described features is tedious mostly for two reasons. First, constructing mesh free basis functions that possess the Kronecker delta property and thus allow for a straight forward imposition of essential BCs is nontrivial, especially if the spatial arrangement of discretization nodes is irregular. Second, in contrast to Galerkin weak form based approaches, collocation methods do not imply a unique and consistent strategy for imposing natural BCs. A strategy commonly used therefore is to enforce either a BC or the PDE at an individual discretization node, but never both at the same time. This inconsistency leads to a deterioration of the approximation accuracy in regions close to boundaries and it may thus also impair the global result. The proposed method aims at alleviating these deficiencies.

It is further worth mentioning that moving Kriging collocation (MKC) does not require the numerical evaluation of spatial integrals at any stage. In addition to the spatio-temporal discretization, the numerical integration constitutes a second layer of approximation in established algorithms. In Galerkin weak form based approaches such integration procedures, which are often performed by means of a background mesh of high resolution, are expensive. In other methods, such as smoothed particle hydrodynamics[8,9] or peridynamcis,[10,11] a significantly cheaper nodal integration strategy is employed. Such nodal integration schemes, however, promote the occurrence of zero energy modes.[12]

What the proposed method has in common with existing mesh free approaches is its purpose. In a multitude of scenarios, mesh free discretization methods for the numerical solution of PDEs are more efficient and flexible than conventional, mesh based approaches. By using mesh free methods, setting up an appropriate discretization of a physical domain becomes a rather straight forward task that can be achieved without human assistance. This feature is crucial in, for example, applications with complex geometries, rapidly evolving interfaces, and large deformations. It is reported that mesh free methods function robustly in such scenarios.[13] In mesh based techniques, on the other hand, pathologically distorted grids are often a source of trouble. Moreover, if the discretization does not rely on a topologically connected grid, finite deformation phenomena can be accounted for more naturally.[14]

Originally, mesh free methods gained their popularity through applications in which the exact imposition of BCs was of minor importance. In such scenarios, the employment of numerical methods which satisfy BCs with only low accuracy may be deemed appropriate. Also, in the field of solid mechanics such methods may be legitimately employed if the phenomena of interest are largely independent of information that is provided via the domain surface. The method presented in this article, however, is to be seen in the context of problems in which a lot of information is passed across surfaces, that is, problems in which BCs should be captured as accurately as possible. As an example of one such scenario one may think of surface coupled multiphysics applications.

To this day, MK basis functions have almost exclusively been investigated in a Galerkin weak form context.[3-6,15,16] One recent exception is the paper of Dehghan et al,[7] in which the employment of MK basis functions within a collocation scheme is proposed for the numerical solution of fractional PDEs. Their strategy of imposing BCs can be very much improved by the consistent approach that is proposed in this work. Here, we employ the probabilistic Kriging framework to derive basis functions which are then used within a deterministic collocation scheme. Fasshauer et al,[17] Cockayne et al,[18] Raissi et al,[19] and others[20] have conducted research on how to apply Kriging-like techniques directly for the numerical treatment of (stochastic) PDEs. The fully stochastic nature is what conceptually distinguishes their approach from the one presented in this work.

The high level structure of this article consists of three parts. The first part is dedicated to the derivation of an interpolation scheme (Sections 2 and 3). It is described in the second part (Section 4), how, by insertion of this interpolation scheme into the PDE, a collocation scheme is obtained. Finally, implementation issues and the results of numerical experiments are discussed and a suggestion for a generalization of MKC for nonlinear PDEs is outlined (Sections 5 to 8).

2 | KRIGING INTERPOLATION

Kriging is an interpolative inference technique originally employed in the field of geostatistics.[2,21] It can be viewed as a special instance of Gaussian Process regression,[1] which, as an inference technique, is more general in the sense that

the assumption of noise free observations is abandoned. The following short introduction of the Kriging interpolant is inspired by the book of Rasmussen et al[1] and is given for the sake of understanding.

Remark on the notation: An explicit discussion of the notation concept employed in this manuscript is important because certain letters will be used in different fonts and in combination with additional markers in order to denote related, yet fundamentally different objects. Consider, for instance the letter g. Without any additional markers, g denotes a deterministic function. The exponent in brackets $(\cdot)^{(a)}$ represents a discrete value. Stochastic processes, i.e., "random functions", are marked with a bar $\overline{(\cdot)}$. Parametrized, deterministic functions which interpolate or approximate a dataset are marked with a hat $\hat{(\cdot)}$. If a symbol is in bold, it represents a collection of discrete values in the form of a vector or matrix. The following table summarizes the notation concept:

Notation feature	Example	Meaning
nothing	$g(\cdot)$	$g(\cdot)$ is a function with unspecified argument.
exponent in brackets $(\cdot)^{(a)}$	$g^{(a)} = g(x^{(a)})$	$g^{(a)}$ is the value of the function g at discrete location $x^{(a)}$.
bar $\overline{(\cdot)}$	$\overline{g}$	$\overline{g}$ is a stochastic process.
hat $\hat{(\cdot)}$	$\hat{g}$	$\hat{g}$ is a parametrized, deterministic function that interpolates or approximates a dataset.
bold symbol	$\boldsymbol{g} = \begin{bmatrix} g^{(1)} & \dots & g^{(n)} \end{bmatrix}^\top$	$\boldsymbol{g}$ is a vector comprising the discrete values $g^{(a)}$ with $a \in \{1, \dots, n\}$.
bold symbol with bar	$\overline{\boldsymbol{g}} = \begin{bmatrix} \overline{g}^{(1)} & \dots & \overline{g}^{(n)} \end{bmatrix}^\top$	$\overline{\boldsymbol{g}}$ is the random vector comprising all discrete random values $\overline{g}^{(a)}$ with $a \in \{1, \dots, n\}$ of the stochastic process $\overline{g}$.

2.1 | The Kriging interpolant

We consider a parametrized function $\hat{f} : \Omega \subset \mathbb{R}^d \to \mathbb{R}^m, x \mapsto \hat{f}(x)$ that satisfies certain constraints. Let there be K different constraint types which are indexed by $\kappa \in \{1, \dots, K\}$. Each constraint type κ can be expressed in terms of a linear constraint operator $\mathcal{C}_\kappa$ and a corresponding constraint function $c_\kappa : \Omega_\kappa \to \mathbb{R}^{m_\kappa}, x \mapsto c_\kappa(x)$ acting on a constraint set $\Omega_\kappa \subset \Omega$. The constraint set Ω_κ is to be understood as the subset of Ω on which the κth constraint type is active. Let there be a target function $f : \Omega \to \mathbb{R}^m$ from which the constraint functions $c_\kappa : \Omega \to \mathbb{R}^{m_\kappa}$ can be derived, that is,

$$c_\kappa(x) = \mathcal{C}_\kappa(f(x); x). \tag{1}$$

Our objective consists in constructing the function $\hat{f}$ such that it approximates the target function f by satisfying the constraints $\kappa \in \{1, \dots, K\}$ at a finite number of locations $x_\kappa^{(a)} \in \Omega_\kappa$ with $a \in \{1, \dots, n_\kappa\}$. The notation $x_\kappa^{(a)}$ implies that there are K different sets of discrete locations (also called *nodes* or *particles* in a collocation context). The nodes of each set are associated with a constraint type, that is, they "carry" this constraint. Using the constraint operator and constraint function concept, the objective can be formulated as

$$\mathcal{C}_\kappa(\hat{f}(x); x) \begin{cases} = c_\kappa^{(a)} & \text{if } x = x_\kappa^{(a)} \\ \approx c_\kappa(x) & \text{if } x \in \Omega_\kappa \end{cases}, \tag{2}$$

where $c_\kappa^{(a)} = c_\kappa(x_\kappa^{(a)})$ are constraint functions evaluations at discrete locations $x_\kappa^{(a)} \in \Omega_\kappa$. For the case in which $K = 1$ and $\mathcal{C}_\kappa$ is an identity operator, the described objective is equivalent to a standard interpolation problem and thus the function $\hat{f}$ is from now on referred to as the *interpolant*. In an empirical inference context, the discrete locations $x_\kappa^{(a)}$ are to be seen as *inputs*, whereas the constraint function values $c_\kappa^{(a)}$ can be interpreted as *generalized observations*.

As a fundamental principle in Kriging, the interpolant $\hat{f}$ is identified with the mean of a smooth stochastic process $\overline{f}$ which is conditioned on noise-free observations. Before it is demonstrated how the conditioning of the stochastic process $\overline{f}$ on observations can be achieved, the following assumptions need to be put in place:

$$\mathbb{E}[\bar{f}(x)] = 0 \tag{3}$$

$$[\mathbb{C}\mathrm{ov}(\bar{f}(x))]_{ij} = \delta_{ij} k(x, x'), \tag{4}$$

where δ_{ij} is the Kronecker delta and $[\mathbb{C}\mathrm{ov}(\bar{f}(x))]_{ij}$ denotes the covariance between the i-th and the jth component of the random function values $\bar{f}(x)$ and $\bar{f}(x')$, respectively. Assumption 3 states that, in average, the value at a certain location x of a realization of the stochastic process $\bar{f}$ equals zero. Assumption 4 implies that all components of the stochastic process $\bar{f}$ are independent. Moreover, it states that the covariance between random function value components $[\bar{f}(x)]_i$ and $[\bar{f}(x')]_i$ at locations x and x' is given as a function $k(x, x')$ of the locations x and x'. The function k is termed kernel- or covariance function and can take various, however not arbitrary forms (cf. Reference 1). In this work, the choice of the covariance function is restricted to the squared exponential covariance function

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\lambda^2}\right), \tag{5}$$

where λ is the so-called characteristic length scale parameter which controls the complexity or flexibility of the kriging function. It will be discussed in more detail in Section 5.2. The introduction of a priori assumptions regarding the unconditioned process reflects the Bayesian nature of this approach. In order to arrive at an expression for the conditioned stochastic process, it is convenient to define a joint random vector $\bar{\boldsymbol{f}}^\top = \begin{bmatrix} \bar{f}(x)^\top & \bar{\boldsymbol{c}}^\top \end{bmatrix}$ involving the random constraint function values

$$\bar{\boldsymbol{c}} = \begin{bmatrix} \bar{\boldsymbol{c}}_1 \\ \vdots \\ \bar{\boldsymbol{c}}_K \end{bmatrix} \quad \text{where} \quad \bar{\boldsymbol{c}}_\kappa = \begin{bmatrix} C_\kappa(\bar{f}(x_\kappa^{(1)}); x) \\ \vdots \\ C_\kappa(\bar{f}(x_\kappa^{(n_\kappa)}); x) \end{bmatrix}. \tag{6}$$

In accordance with the aforementioned prior assumptions, one can write $\bar{\boldsymbol{f}}$ as a Gaussian random vector

$$\bar{\boldsymbol{f}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}) \qquad \text{with} \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{ff}(x) & \boldsymbol{\Sigma}_{fc}(x) \\ \text{sym.} & \boldsymbol{\Sigma}_{cc} \end{bmatrix}, \tag{7}$$

where

$$\boldsymbol{\Sigma}_{ff}(x) = I\ k(x, x), \quad \boldsymbol{\Sigma}_{fc}(x) = \begin{bmatrix} \boldsymbol{\Sigma}_{fc_1}(x) & \dots & \boldsymbol{\Sigma}_{fc_K}(x) \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\Sigma}_{cc} = \begin{bmatrix} \boldsymbol{\Sigma}_{c_1c_1} & \dots & \boldsymbol{\Sigma}_{c_1c_K} \\ & \ddots & \vdots \\ \text{sym.} & & \boldsymbol{\Sigma}_{c_Kc_K} \end{bmatrix}. \tag{8}$$

The matrix $\boldsymbol{\Sigma}_{fc}(x)$ describes the covariance between the random function value $\bar{f}(x)$ at an arbitrary location $x \in \Omega$ and the random constraint function values at discrete locations $x_\kappa^{(a)} \in \Omega_\kappa$. Accordingly, $\boldsymbol{\Sigma}_{cc}$ describes the covariances among random constraint function values. The values of these matrices need to be inferred consistently from the covariance function k. Details on how this is done for specific choices of constraint operators are given in Section 2.3.

Let the generalized observations be given in the form of a vector $\boldsymbol{c}$ (note: no bar over the symbol $\boldsymbol{c}$ as it represents a deterministic vector) which is defined as

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_1 \\ \vdots \\ \boldsymbol{c}_K \end{bmatrix} \quad \text{with} \quad \boldsymbol{c}_\kappa = \begin{bmatrix} c_\kappa^{(1)} \\ \vdots \\ c_\kappa^{(n_\kappa)} \end{bmatrix}. \tag{9}$$

Then, the distribution $p(\bar{f}(x)|\bar{\boldsymbol{c}} = \boldsymbol{c})$ of $\bar{f}(x)$ conditioned on the observations in $\boldsymbol{c}$ can be found by means of elementary algebraic manipulations of multivariate Gaussian distributions:

$$p(\bar{f}(x) \mid \bar{\boldsymbol{c}} = \boldsymbol{c}) = \mathcal{N}(\underbrace{\boldsymbol{\Sigma}_{fc}(x)\boldsymbol{\Sigma}_{cc}^{-1}\boldsymbol{c}}_{\hat{f}(x)},\ \boldsymbol{\Sigma}_{ff}(x) - \boldsymbol{\Sigma}_{fc}(x)\boldsymbol{\Sigma}_{cc}^{-1}\boldsymbol{\Sigma}_{fc}^{\top}(x)). \tag{10}$$

As indicated in (10), the mean of the conditional distribution $p(\bar{f}(x) \mid \bar{\boldsymbol{c}} = \boldsymbol{c})$ yields the sought-after interpolant

$$\hat{f}(x) = \boldsymbol{A}(x)\boldsymbol{c}, \tag{11}$$

where $\boldsymbol{A}(x) = \boldsymbol{\Sigma}_{xc}\boldsymbol{\Sigma}_{cc}^{-1}$ shall from now on be referred to as *linear interpolation operator*. The hat marker on the interpolant $\hat{f}$ indicates that it is a deterministic, parametrized function. Its parameters are the values contained in the vector $\boldsymbol{c}$.

2.2 | *P*th order complete Kriging interpolant

If the observations in $\boldsymbol{c}$ originate from a polynomial model, the interpolant $\hat{f}(\cdot)$ defined in the previous section is potentially inapt to recover said polynomial exactly. In a solid mechanics context this implies that in general, if particles are subjected to a rigid body motion, the interpolated displacement field is distortive. In order to achieve what is called *P*th order completeness, meaning that the interpolant recovers polynomials of order P exactly given sufficiently many data points, explicit basis functions are introduced to the Kriging framework.[1,22,23]

The structure of the unconditioned stochastic process which is enriched by explicit basis functions is given by

$$\bar{f}^{\mathrm{P}}(x) = \bar{f}(x) + h^{\top}(x)\beta, \tag{12}$$

where $\bar{f}$ is the zero-mean Gaussian process with covariance function $k(\cdot,\cdot)$, $h(\cdot)$ is a vector of explicit basis functions (in the present case a complete Pth order polynomial basis), and β is a vector of corresponding coefficients. Within the Bayesian paradigm, the parameters β are to be viewed as random variables. A zero-mean Gaussian prior distribution with covariance matrix B is therefore assigned to these parameters, that is, $\beta \sim \mathcal{N}(0, B)$.

We define the vector $h_\kappa(\cdot)$ of constraint-transformed basis functions such that $h_\kappa(x)^{\top}\beta = \mathcal{C}_\kappa(h(x)^{\top}\beta; x)$. Evaluating the constraint-transformed basis function $h_\kappa(\cdot)$ at all locations $x_\kappa^{(a)}$ with $a \in \{1, \dots, n_\kappa\}$ for each constraint type κ, gives rise to the matrix $\boldsymbol{H}_c$:

$$\boldsymbol{H}_c = \begin{bmatrix} \boldsymbol{H}_{c_1} \\ \vdots \\ \boldsymbol{H}_{c_K} \end{bmatrix} \quad \text{with} \quad \boldsymbol{H}_{c_\kappa} = \begin{bmatrix} h_\kappa^{\top}(x_\kappa^{(1)}) \\ \vdots \\ h_\kappa^{\top}(x_\kappa^{(n_\kappa)}) \end{bmatrix}. \tag{13}$$

Using these definitions, a new random vector is introduced:

$$\bar{\boldsymbol{f}}^{\mathrm{P}} = \begin{bmatrix} \bar{f}^{\mathrm{P}}(x) \\ \bar{\boldsymbol{c}}^{P} \end{bmatrix} \quad \text{with} \quad \bar{\boldsymbol{c}}^{\mathrm{P}} = \bar{\boldsymbol{c}} + \boldsymbol{H}_c\beta. \tag{14}$$

As a linear transformation of Gaussian random variables, the random vector $\bar{\boldsymbol{f}}^{\mathrm{P}}$ is also a Gaussian. In order to obtain its distribution, one needs to determine its mean $\boldsymbol{\mu}^{\mathrm{P}}$ and its covariance matrix $\boldsymbol{\Sigma}^{\mathrm{P}}$:

$$\boldsymbol{\mu}^{\mathrm{P}} = \mathbb{E}(\bar{\boldsymbol{f}}^{\mathrm{P}}) = \mathbb{E}(\bar{\boldsymbol{f}}) + \mathbb{E}(\boldsymbol{H}\beta) = \boldsymbol{0} \tag{15}$$

$$\boldsymbol{\Sigma}^{\mathrm{P}} = \mathbb{C}\mathrm{ov}(\bar{\boldsymbol{f}}^{\mathrm{P}}) = \begin{bmatrix} \mathbb{C}\mathrm{ov}(\bar{f}^{\mathrm{P}}(x)) & \mathbb{C}\mathrm{ov}(\bar{f}^{\mathrm{P}}(x), \bar{\boldsymbol{c}}^{\mathrm{P}}) \\ \text{sym.} & \mathbb{C}\mathrm{ov}(\bar{\boldsymbol{c}}^{\mathrm{P}}, \bar{\boldsymbol{c}}^{\mathrm{P}})) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_{ff}(x) + h(x)^{\top}Bh(x) & \boldsymbol{\Sigma}_{fc}(x) + h(x)^{\top}B\boldsymbol{H}_c \\ \text{sym.} & \boldsymbol{\Sigma}_{cc} + \boldsymbol{H}_c^{\top}B\boldsymbol{H}_c \end{bmatrix}. \tag{16}$$

Similar to the proceeding in previous sections, the Kriging interpolant $\hat{f}^{\mathrm{P}}(\cdot)$ that incorporates explicit basis functions is obtained as the conditional mean of a stochastic process. The stochastic process that is employed to this end is the limit

of $\bar{f}^{\mathrm{P}}(\cdot)$ as $B^{-1} \to 0$. As can be inferred from the results below, taking this limit is necessary in order for the interpolant to reproduce linear combinations of functions in the explicit basis $h(\cdot)$ exactly. Moreover, this limit reflects the fact that, in practice, the prior knowledge regarding the values of the explicit basis parameters β tends to zero. As demonstrated in O'Hagan et al,[23] by using the matrix inversion lemma, it is possible to determine the conditional mean of $\bar{f}^{\mathrm{P}}(\cdot)$ for the limiting case $B^{-1} \to 0$ as follows:

$$\mathbb{E}(\bar{f}^{\mathrm{P}}(x) \,\big|\, \bar{\boldsymbol{c}}^{\mathrm{P}} = \boldsymbol{c}) = (\boldsymbol{\Sigma}_{fc}(x) + h^{\top}(x)B\boldsymbol{H}_c^{\top})(\boldsymbol{\Sigma}_{cc} + \boldsymbol{H}_c B \boldsymbol{H}_c^{\top})^{-1}\boldsymbol{c} \tag{17}$$

$$= (\boldsymbol{\Sigma}_{fc}(x)\boldsymbol{\Sigma}_{cc}^{-1} + (h^{\top}(x) - \boldsymbol{\Sigma}_{fc}(x)\boldsymbol{\Sigma}_{cc}^{-1}\boldsymbol{H}_c)(B^{-1} + \boldsymbol{H}_c^{\top}\boldsymbol{\Sigma}_{cc}^{-1}\boldsymbol{H}_c)^{-1}\boldsymbol{H}_c^{\top}\boldsymbol{\Sigma}_{cc}^{-1})\boldsymbol{c}. \tag{18}$$

From expression (18), the limit of $\mathbb{E}(\bar{f}^{\mathrm{P}}(x) \,|\bar{\boldsymbol{c}}^{\mathrm{P}} = \boldsymbol{c})$ as $B^{-1} \to 0$ can readily be deduced, giving rise to the explicit-basis-enhanced interpolant

$$\hat{f}^{\mathrm{P}}(x) = \boldsymbol{A}^{\mathrm{P}}(x)\boldsymbol{c} \quad \text{with} \quad \boldsymbol{A}^{\mathrm{P}}(x) = \boldsymbol{A}(x) + (h^{\top}(x) - \boldsymbol{\Sigma}_{fc}(x)\boldsymbol{\Sigma}_{cc}^{-1}\boldsymbol{H}_c)(\boldsymbol{H}_c^{\top}\boldsymbol{\Sigma}_{cc}^{-1}\boldsymbol{H}_c)^{-1}\boldsymbol{H}_c^{\top}\boldsymbol{\Sigma}_{cc}^{-1}, \tag{19}$$

where $\boldsymbol{A}^{\mathrm{P}}(x)$ shall from now on be referred to as *explicit basis linear interpolation operator*. In order to proof that $\hat{f}^{\mathrm{P}}(\cdot)$ reproduces any linear combination of functions in the explicit basis $h(\cdot)$ exactly, let $\boldsymbol{c}^* = \boldsymbol{H}_c\beta^*$, where β^* is a vector of arbitrary coefficient values. Then, $\hat{f}^{\mathrm{P}*}(x) = \boldsymbol{A}^{\mathrm{P}}(x)\boldsymbol{c}^* = h^{\top}(x)\beta^*$.

Derivatives of the Kriging interpolant

Derivatives of the Kriging interpolant are readily available. As can be seen in (19), the derivatives of the Kriging interpolant are obtained by differentiation of the linear interpolation operator. This differentiation, in turn, is achieved by differentiating the explicit basis- and the covariance functions.

2.3 | Constraint types

In this section, the constraint operators for several constraint types that are required in subsequent sections are presented. In addition, explicit expressions for the corresponding covariance matrices $\boldsymbol{\Sigma}_{fc}$ and $\boldsymbol{\Sigma}_{cc}$ are provided. To this end, these matrices are decomposed into blocks:

$$\boldsymbol{\Sigma}_{fc_\kappa}(x) = \begin{bmatrix} \mathbb{C}\mathrm{ov}(\bar{f}(x), \bar{c}_\kappa^{(1)}) & \dots & \mathbb{C}\mathrm{ov}(\bar{f}(x), \bar{c}_\kappa^{(n_\kappa)}) \end{bmatrix}, \quad \boldsymbol{\Sigma}_{c_\kappa c_\lambda} = \begin{bmatrix} \mathbb{C}\mathrm{ov}(\bar{c}_\kappa^{(1)}, \bar{c}_\lambda^{(1)}) & \dots & \mathbb{C}\mathrm{ov}(\bar{c}_\kappa^{(1)}, \bar{c}_\lambda^{(n_\lambda)}) \\ \vdots & \ddots & \vdots \\ \mathbb{C}\mathrm{ov}(\bar{c}_\kappa^{(n_\kappa)}, \bar{c}_\lambda^{(1)}) & \dots & \mathbb{C}\mathrm{ov}(\bar{c}_\kappa^{(n_\kappa)}, \bar{c}_\lambda^{(n_\lambda)}) \end{bmatrix}, \tag{20}$$

with $\mathbb{C}\mathrm{ov}(\bar{f}(x), \bar{c}_\lambda^{(a)}) \in \mathbb{R}^{m \times m_\kappa}$ and $\mathbb{C}\mathrm{ov}(\bar{c}_\kappa^{(a)}, \bar{c}_\lambda^{(b)}) \in \mathbb{R}^{m_\kappa \times m_\lambda}$, where m_κ and m_λ are the dimensions of the constraint types κ and λ, respectively. In $\bar{c}_\kappa^{(a)}$, the bar indicates that this quantity is a random variable that is obtained as the constraint-transformation of the random function value $\bar{f}$ at location $x_\kappa^{(a)}$, that is,

$$\bar{c}_\kappa^{(a)} = C_\kappa(\bar{f}(x_\kappa^{(a)}); x_\kappa^{(a)}). \tag{21}$$

The definitions of the constraint operators given below are formulated in terms of the generic function $g : \mathbb{R}^d \to \mathbb{R}^m$, $x \mapsto g(x)$.

2.3.1 | Identity constraints

The simplest constraint operator is the identity operator which maps a function $g(\cdot)$ onto itself. In a collocation context this operator will be used to achieve two things: First, it enables the introduction of degrees of freedom (DOFs) that can be identified with function values of the interpolant at certain discrete locations ($\kappa = 1$). Second, this operator is used to

enforce essential BCs ($\kappa = 2$). Hence,

$$C_\kappa(g(x);x) = g(x) \quad \text{with} \quad \kappa \in \{1,2\}. \tag{22}$$

The covaraince between a random function value at an arbitrary location x and the random constraint value of type 1 or 2 at discrete location $x_\kappa^{(a)}$ thus is

$$[\mathbb{C}\text{ov}(\bar{f}(x), \bar{c}_\kappa^{(a)})]_{ij} = k(x, x_\kappa^{(a)})\delta_{ij} \quad \text{with} \quad \kappa \in \{1,2\} \tag{23}$$

$$[\mathbb{C}\text{ov}(\bar{c}_\kappa^{(a)}, \bar{c}_\lambda^{(b)})]_{ij} = k(x_\kappa^{(a)}, x_\lambda^{(a)})\delta_{ij} \quad \text{with} \quad \kappa, \lambda \in \{1,2\}. \tag{24}$$

By means of this expression, the covariance matrices $\mathbf{\Sigma}_{fc_1}, \mathbf{\Sigma}_{fc_2}, \mathbf{\Sigma}_{c_1c_1}, \mathbf{\Sigma}_{c_1c_2}$, and $\mathbf{\Sigma}_{c_2c_2}$ from (8) are explicitly defined.

2.3.2 | Neumann constraints

The operator to impose Neumann BCs ($\kappa = 3$) is described as follows. For the definition of C_3, it is convenient to roll out the Jacobian of the generic function g in a vector Dg, that is,

$$\text{Dg} = \begin{bmatrix} \nabla[g]_1 \\ \vdots \\ \nabla[g]_m \end{bmatrix} \quad \text{with} \quad \nabla[g]_i = \begin{bmatrix} \frac{\partial[g]_i}{\partial[x]_1} \\ \vdots \\ \frac{\partial[g]_i}{\partial[x]_d} \end{bmatrix}, \tag{25}$$

where $[g]_i$ indicates the ith component of $g(\cdot)$. Introducing the input dependent matrix of coefficients $L : \Omega_3 \subset \mathbb{R}^d \to \mathbb{R}^{m_3 \times d \cdot m}, x \mapsto L(x)$, where m_3 denotes the dimension of this third constraint type, the Neumann constraint operator is

$$C_3(g(x);x) = L(x)\text{Dg}(x). \tag{26}$$

The input dependency of L arises from the fact that expressions for Neumann BCs involve spatially varying surface normal vectors. The covariance expressions involved in the second constraint are

$$[\mathbb{C}\text{ov}(\bar{f}(x), \bar{c}_3^{(a)})]_{ij} = \sum_{p=1}^{d\cdot m} [L(x)]_{jp}\mathbb{C}\text{ov}([\bar{f}(x)]_i, [\text{D}\bar{f}(x_3^{(a)})]_p) \tag{27}$$

$$[\mathbb{C}\text{ov}(\bar{c}_\kappa^{(a)}, \bar{c}_3^{(b)})]_{ij} = \sum_{p=1}^{d\cdot m} [L(x)]_{jp}\mathbb{C}\text{ov}([\bar{f}(x_\kappa^{(a)})]_i, [\text{D}\bar{f}(x_3^{(b)})]_p) \quad \text{with} \quad \kappa \in \{1,2\} \tag{28}$$

$$[\mathbb{C}\text{ov}(\bar{c}_3^{(a)}, \bar{c}_3^{(b)})]_{ij} = \sum_{p=1}^{d\cdot m}\sum_{q=1}^{d\cdot m} [L(x)]_{ip}[L(x)]_{jq}\mathbb{C}\text{ov}([\text{D}\bar{f}(x_3^{(a)})]_p, [\text{D}\bar{f}(x_3^{(b)})]_q). \tag{29}$$

From the prior assumptions in (3) and (4) it can be inferred[1,24] that the covariance between a value of the stochastic process at location x and the partial derivative of the stochastic process at location x', and the covariance between partial derivatives at locations x and x' are

$$\mathbb{C}\text{ov}\left([\bar{f}(x)]_i, \frac{\partial[\bar{f}(x')]_j}{\partial[x']_p}\right) = \frac{\partial k(x,x')}{\partial[x']_p}\delta_{ij} \tag{30}$$

$$\mathbb{C}\text{ov}\left(\frac{\partial[\bar{f}(x)]_i}{\partial[x]_p}, \frac{\partial[\bar{f}(x')]_j}{\partial[x']_q}\right) = \frac{\partial^2 k(x,x')}{\partial[x]_p\partial[x']_q}\delta_{ij}. \tag{31}$$

By means of the above expressions, the covariance matrices $\mathbf{\Sigma}_{fc_3}, \mathbf{\Sigma}_{c_1c_3}, \mathbf{\Sigma}_{c_2c_3}$, and $\mathbf{\Sigma}_{c_3c_3}$ from (8) are explicitly defined.

2.3.3 | Higher order derivative constraints

It is, of course, also possible to enforce higher order derivative constraints. This possibility can for instance be exploited in order to construct basis functions which constitute a solution to the PDE at certain locations prior to solving a global system of equations ($\kappa = 4$). In a MKC context, the motivation for introducing PDE-constraints arises from the fact that with standard MK, the numerical solution fails to satisfy the PDE in the vicinity of Dirichlet nodes. As will be demonstrated in Section 6 by means of an example, this circumstance causes the numerical solution to exhibit less plausible boundary fluxes at the Dirichlet boundary. The gravity of this problem is alleviated if the localized MK approximation functions are constrained to satisfy the differential equation at said Dirichlet nodes.

From now on, functions which, by construction, constitute a solution to the PDE at certain locations are referred to as *PDE-constrained*. A formal description as was given for the previous constraint operators is refrained from since this procedure is exhaustive and in principle identical to the one employed in Section 2.3.2. Further details can be found in the literature[17-20] in which higher order derivative constraints for Gaussian Processes have been studied in the context of fully probabilistic numerical schemes for solving PDEs.

It is important to distinguish these approaches from the present one. In the fully probabilistic schemes, a linear system of equations needs to be solved in order to statistically condition a Gaussian Process on a PDE. The corresponding system matrix is a symmetric positive definite covariance matrix; it is nonsparse and its rank is approximately equal to the dimensions k of the PDE times the number of discretization nodes n. Note that these approaches do not require the solution of a system of equations in the sense of (48). However, due to the size and nonsparsity of the system of equations that needs to be solved in the course of the statistical conditioning, this method is unsuitable for large scale simulations.

3 | MOVING KRIGING INTERPOLATION

The interpolation scheme that was derived in the previous section necessitates the inversion of a covariance matrix of rank $\sum_{\kappa=1}^{K} n_\kappa m_\kappa$. It must be emphasized that, as will become apparent in Section 4, an actual inversion of the covariance matrix is required in contrast to a single solve of a linear system of equations. Such an inversion would undeniably entail overwhelming computational costs. In order to circumvent this obstacle the following strategy, which aims at a localization of the interpolant, is employed. This localization approach is akin to a moving least squares scheme[25] and is therefore referred to as moving or local Kriging.[5,6] In the following, this moving Kriging (MK) concept is embedded in the notation employed in this article and is extended in a way that allows for the utilization of generalized observations.

In MK, the interpolant $\hat{f}_{x'}$ centered at x' is constructed solely based on generalized observations at locations in the vicinity of x'. Hence, the demands from (32) are relaxed and now given by

$$C_\kappa(\hat{f}(x);x) \begin{cases} = c_\kappa^{(a)} & \text{if } x = x_\kappa^{(a)} \wedge\ x \in \mathcal{H}(x') \\ \approx c_\kappa(x) & \text{if } x \in \Omega_\kappa \end{cases}, \tag{32}$$

where $\mathcal{H}(x') = \{x : \|x - x'\| \leq \delta\}$ is a spherical region with radius δ (interaction radius) and centred at x. The localized interpolant is thus given by

$$\hat{f}_{x'}^{\mathrm{LOC}}(x) = \boldsymbol{A}^{\mathrm{loc}\,x'}(x)\boldsymbol{c}^{\mathrm{loc}\,x'} \quad \text{with} \quad \boldsymbol{c}^{\mathrm{loc}\,x'} = \begin{bmatrix} \boldsymbol{c}_1^{\mathrm{loc}x'} \\ \vdots \\ \boldsymbol{c}_K^{\mathrm{loc}x'} \end{bmatrix}, \tag{33}$$

where the local linear interpolation operator $\boldsymbol{A}^{\mathrm{loc}x'}(x)$ is constructed analogously to (19). It is however adapted so as to act only on a local dataset, that is, only on the generalized observations in the vicinity $\mathcal{H}(x')$. Each block component of the local vector of generalized observations $\boldsymbol{c}_\kappa^{\mathrm{loc}x'}$ can be written as

$$\boldsymbol{c}_\kappa^{\mathrm{loc}x'} = \begin{bmatrix} c_\kappa^{(a_1)} \\ \vdots \\ c_\kappa^{(n_\kappa[x'])} \end{bmatrix} \quad \text{with} \quad c_\kappa^{(a_i)} \in \{c_\kappa^{(a)} = C_\kappa(f(x^{(a)});x^{(a)}) : x^{(a)} \in \mathcal{H}(x')\}, \tag{34}$$

and where $n_\kappa[x']$ denotes the number of generalized observations of type κ in $\mathcal{H}(x')$. The radius δ of the spherical region $\mathcal{H}(x')$ is to be selected such that $n_\kappa[x'] \ll n$. A globally operating interpolant $\hat{f}^{\mathrm{LOC}}(x)$ is obtained from (33) as

$$\hat{f}^{\mathrm{LOC}}(x) = \hat{f}^{\mathrm{LOC}}_{x'=x}(x). \tag{35}$$

Note that the interpolant $\hat{f}^{\mathrm{LOC}}(\cdot)$ is discontinuous since the local dataset upon which the local interpolation operators act varies discretely across the domain. Such discontinuities are practically irrelevant for approximation schemes derived in the sequel. The discontinuities will only be recognizable in interpolated approximations with too coarse nodal arrangements.

4 | MOVING KRIGING COLLOCATION

In the following, an MKC scheme for the numerical solution of PDEs is explained in general terms. In the previous sections, the objective consisted in finding a data interpolating function with certain additional properties. This function was referred to as *interpolant*. In the context of numerical solutions of PDEs, the function of interest shall be called *approximant*. A Kriging interpolant can be reinterpreted as an approximant if a subset of generalized observations is understood as DOFs as opposed to empirical data. This DOF-subset is denoted by $\boldsymbol{f}^{\mathrm{DOF}}$ and is identified with the (unknown) values of the solution function of a PDE at discrete locations which are distributed across the entire domain Ω.

Let a generic linear PDE and the corresponding Neumann and Dirichlet BCs be given by

$$\mathcal{D}f(x) = b(x) \quad \text{in } \Omega \tag{36}$$

$$\mathcal{G}(x)f(x) = t(x) \quad \text{on } \Gamma_N \tag{37}$$

$$f(x) = f^D(x) \quad \text{on } \Gamma_D, \tag{38}$$

where $\mathcal{D}$ and $\mathcal{G}$ are linear differential operators acting on the solution function f. The symbols b, t and f^D denote known, vector-valued functions. Let the domain of interest Ω including its boundary $\Gamma = \Gamma_N \cup \Gamma_D$ be represented by a set of discrete locations $x^{(a)} \in \mathbb{R}^d$ with $a \in \{1, \dots, n\}$ with $n = \sum_{\kappa=1}^{4} n_\kappa$.

In common collocation-type methods,[13] one seeks to approximate the solution function f by demanding that the approximant $\hat{f}$ satisfies the differential equation (36) at discrete locations that belong to the interior of the domain. At discrete locations that lie on the domain surface, these approximants are solely required to satisfy the BCs (37) and (38). This strategy shall in the following be referred to as *conventional*. In our approach, the differential equation is enforced at all discrete locations, that is, also at locations on the boundary of the domain. A pointwise exact imposition of BCs is enabled by constructing the localized approximants such that they satisfy the BCs at discrete locations on the domain surface a priori, that is, prior to solving the global system of equations. In other words, the function space spanned by the interpolant comprises exclusively functions that comply with the BCs. Hence, in contrast to the conventional approach, our approach to imposing BCs shall from now on be referred to as *consistent*.

In order to achieve this consistency, the approximant $\hat{f}$ is constructed such that it satisfies the BCs (37) and (38) irrespective of the values of its discrete DOFs. This requirement can be complied with elegantly by resorting to the (moving) Kriging framework. There are essentially two properties of a Kriging approximant that make it appealing: (1) it interpolates its DOFs and (2) it can be constructed so as to satisfy constraints that are formulated in terms of linear combinations of partial derivatives. A suitable Kriging approximant is obtained by reinterpreting the differential operators in (36) to (38) as constraint operators, that is,

$$C_1(f(x);x) = f(x), \quad C_2(f(x);x) = f(x), \quad C_3(f(x);x) = \mathcal{G}(x)f(x) \quad \text{and } C_4(f(x);x) = \mathcal{D}f(x). \tag{39}$$

Note that, as was explained in 2.3.1, the operators C_1 and C_2 are identical. For better readability, the constraint sets Ω_1 and Ω_4 corresponding to constraint types 1 and 4 are renamed as Ω_{DOF} and Γ_{PDE}, respectively. Also, the constraint sets Ω_2 and Ω_3 can be identified with, respectively, the Dirichlet boundary Γ_D and the Neumann boundary Γ_N (cf. Figure 1).

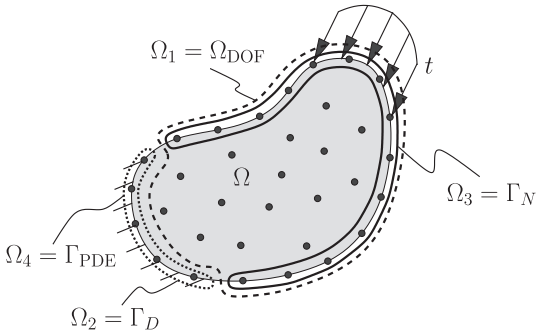


FIGURE 1 Schematic of a computational domain Ω and the subdomains $\{\Omega_\kappa\}_{\kappa=1,\dots,4}$ used in MKC. The black dots represent discretization nodes, t symbolizes the boundary flux

Note that in the examples to come and also in the visualization in Figure 1, all nodes in $\{x_3^{(a)}\}_{a=1,\dots,n_3}$ coincide with nodes in $\{x_1^{(a)}\}_{a=1,\dots,n_1}$ and all nodes in $\{x_4^{(a)}\}_{a=1,\dots,n_4}$ coincide with nodes in $\{x_2^{(a)}\}_{a=1,\dots,n_2}$.

The Kriging approximant has the form $\hat{f}(x) = \tilde{\boldsymbol{A}}(x)\boldsymbol{c}$, where

$$\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \boldsymbol{c}_3 \\ \boldsymbol{c}_4 \end{bmatrix} \quad \text{and} \quad \boldsymbol{c}_1 = \boldsymbol{f}^{\text{DOF}},\ \boldsymbol{c}_2 = \boldsymbol{f}^{D},\ \boldsymbol{c}_3 = \boldsymbol{t} \text{ and } \boldsymbol{c}_4 = \boldsymbol{b}^{D}. \tag{40}$$

The tilde in $\tilde{\boldsymbol{A}}$ indicates that this is a generic interpolation operator in lieu of which any of the previously introduced operator versions can be used. In particular, it can be selected to ensure first order completeness of the approximant. As already pointed out above, the DOFs of the approximant, which at the same time are the values of the approximant at discrete locations in Ω_{DOF}, are denoted by $\boldsymbol{f}^{\text{DOF}}$. Note that $\boldsymbol{f}^{\text{DOF}}$ also comprises DOFs that belong to nodes that lie on the Neumann boundary Γ_N. The prescribed values at nodes on the Dirichlet boundary Γ_D are contained in the vector $\boldsymbol{f}^{D}$. The vector $\boldsymbol{t}$ is composed of evaluations of the function $t(\cdot)$ at the n_3 Neumann boundary nodes in Γ_N, that is, $\boldsymbol{t} = [t(x_3^{(1)})^\top, \dots, t(x_3^{(n_3)})^\top]^\top$. Accordingly, the vector $\boldsymbol{b}$ is composed of evaluations of the function $b(\cdot)$ at the n_4 nodes contained in the set Γ_{PDE}, that is, $\boldsymbol{b} = [b(x_4^{(1)})^\top, \dots, b(x_4^{(n_4)})^\top]^\top$. Utilizing the techniques presented in Sections 2.1 and 2.2, the linear interpolation operator $\tilde{\boldsymbol{A}}_x$ is constructed such that

$$\mathcal{C}_2(\hat{f}(x_2^{(a)}); x_2^{(a)}) = f^D(x_2^{(a)}) \quad \forall\, x_2^{(a)} \in \Omega_2 = \Gamma_D \tag{41}$$

$$\mathcal{C}_3(\hat{f}(x_3^{(a)}); x_3^{(a)}) = t(x_3^{(a)}) \quad \forall\, x_3^{(a)} \in \Omega_3 = \Gamma_N \tag{42}$$

$$\mathcal{C}_4(\hat{f}(x_4^{(a)}); x_4^{(a)}) = b(x_4^{(a)}) \quad \forall\, x_4^{(a)} \in \Omega_4 = \Omega_{\text{PDE}}, \tag{43}$$

holds for all values of $\boldsymbol{f}^{\text{DOF}}$. The reason for including the constraint operator $\mathcal{C}_4$, which represents PDE-constraints, will become apparent at the end of this section. Due to the linearity of the differential operator $\mathcal{D}$, one may write

$$\mathcal{D}\hat{f}(x) = \mathcal{D}\tilde{\boldsymbol{A}}(x)\boldsymbol{c} = b(x). \tag{44}$$

Within vector $\boldsymbol{c}$, the values of $\boldsymbol{f}^{D}$, $\boldsymbol{t}$, and $\boldsymbol{b}^{D}$ are known, whereas the values of $\boldsymbol{f}^{\text{DOF}}$ are unknown and to be determined. The unknown values in $\boldsymbol{f}^{\text{DOF}}$ are obtained by demanding that (44) be satisfied at all discrete locations $x_1^{(a)}$, that is:

$$\mathcal{D}\boldsymbol{A}(x = x^{(a)})\boldsymbol{c} \stackrel{!}{=} b(x_1^{(a)}) \quad \forall\, a \in \{1, \dots, n_1\}. \tag{45}$$

This constitutes a linear system of equations. Equivalently, by introducing the system matrix $\boldsymbol{K}$ and right hand side vector $\boldsymbol{b} = [b(x_1^{(1)})^\top, \dots, b(x_1^{(n)})^\top]^\top$, (45) can be rewritten as

$$\boldsymbol{K}\boldsymbol{c} = \boldsymbol{b}. \tag{46}$$

The system matrix $\boldsymbol{K}$ can be decomposed into block components as follows:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{\Omega\Omega} & \boldsymbol{K}_{\Omega\Gamma} \end{bmatrix}, \tag{47}$$

with $\boldsymbol{K}_{\Omega\Omega} \in \mathbb{R}^{n_1 \times n_1}$ and $\boldsymbol{K}_{\Omega\Gamma} \in \mathbb{R}^{n_1 \times (n_2+n_3+n_4)}$. The system to be solved then reads

$$\boldsymbol{K}_{\Omega\Omega}\boldsymbol{f}^{*\mathrm{DOF}} = \boldsymbol{b} - \boldsymbol{K}_{\Omega\Gamma}\begin{bmatrix} \boldsymbol{f}^D \\ \boldsymbol{t} \\ \boldsymbol{b}^D \end{bmatrix}, \tag{48}$$

with $\boldsymbol{f}^{*\mathrm{DOF}}$ denoting the solution of this system of equations. The resulting approximant

$$\hat{f}^*(x) = \tilde{\boldsymbol{A}}(x)\begin{bmatrix} \boldsymbol{f}^{*\mathrm{DOF}} \\ \boldsymbol{f}^D \\ \boldsymbol{t} \\ \boldsymbol{b}^D \end{bmatrix}, \tag{49}$$

will satisfy the differential equation exactly at all nodes in Ω^{DOF}. Moreover, the approximant $\hat{f}^*$ fulfils, by construction (cf. (41) to (43)), the constraints at all nodes to which these were applied. Further, one recognizes now the purpose of the PDE-constraints: First, assume that the sets Γ_D and Ω_{DOF} are disjoint. Then, if the PDE-constraints are omitted, the approximant $\hat{f}^*$ will not satisfy the PDE at Dirichlet nodes since this demand is not contained in the set of equations (45). If, alternatively, the set Ω_{DOF} includes the set Γ_D and thus $\{x_2^{(a)} : a \in \{1, \dots, n_2\}\} \subset \{x_1^{(a)} : a \in \{1, \dots, n_1\}\}$ one runs into a contradiction: On the one hand, function values at locations in Γ_D are fixed to a certain value and, on the other, they must be variable in order to satisfy (45). This contradiction can be circumvented by constructing the approximant $\hat{f}$ such that it satisfies the PDE at discrete locations in Γ_D irrespective of the values it takes in this region. This is achieved by means of the higher order derivative constraints.

Note that the construction of the matrix $\boldsymbol{K}$ requires n evaluations of the linear interpolation operator $\tilde{\boldsymbol{A}}(x)$ (cf. (19) and (45)). By means of localization, the size of the systems of equation that need to be solved in the course of the evaluation of $\tilde{\boldsymbol{A}}(x)$ is drastically reduced.

5 | IMPLEMENTATION ASPECTS

In this section, practical aspects regarding the implementation are outlined. An overview of the various constituents of the MKC method is given in Algorithm 1. Section 5.1 deals with the computational costs and in Section 5.2, a guideline for an appropriate selection of the characteristic length scale parameter of the covariance function is provided.

5.1 | Discussion of costs

In this section, it is assumed that the approximant is localized, Pth order complete and satisfies all constraints. The costs are assessed in the context of an m-dimensional linear PDE in d variables that is discretized with n_1 DOF-nodes. The interaction radius is given by $\delta = \gamma\Delta x \propto \gamma n^{-1/d}$, with $\gamma > 1$ and Δx being the characteristic discretization length scale. The two major sources of computational cost are the composition of the system matrix and the solution of the corresponding linear system of equations.

Composing and assembling the global system matrix involves the following tasks: First, one needs to create neighbor lists that indicate, for each node $x_1^{(a)}$ with $a \in \{1, \dots, n_1\}$, which other nodes carrying constraints of type κ lie within the spherical neighborhood $\mathcal{H}(x_1^{(a)})$ of that node (Algorithm 1, lines 4 to 6). Second, partial derivatives of the linear interpolation operators $\boldsymbol{A}^{\mathrm{loc}\ x_1^{(a)}}$ centred at the discrete locations $x_1^{(a)}$ with $a \in \{1, \dots, n_1\}$ need to be evaluated (Algorithm 1 lines 7 to 13). In the course of these evaluations, small linear systems of equations need to be solved (Algorithm 1 lines 10 to

13) for various right hand sides. The corresponding system matrix is symmetric positive definite and has, if $x_1^{(a)}$ is sufficiently far away from the boundary, rank $m_1 \cdot n_1[x_1^{(a)}]$. As defined in Section 3, $n_1[x_1^{(a)}]$ denotes the number of nodes carrying DOF-constraints in the vicinity $\mathcal{H}(x_1^{(a)})$. For each node of the discretization, such a system of equations needs to be solved for $\binom{k+P}{P} + d - 1$ different right hand sides. This number is equal to the dimension d of the PDE plus the number of basis functions contained in a k-dimensional Pth order complete polynomial basis. For example, in the 2D linear elasticity problem that will be discussed in Section 6.2 ($d=2$, $k=2$, $\gamma \approx 3$ and $P=2$), a total of $7 \cdot n$ systems of equations of rank $\approx k \cdot (2 \cdot \gamma)^d = 72$ would need to be solved.

Algorithm 1. Pseudo code of the MKC scheme

// Assembly of under-determined system of equations $\boldsymbol{K}\left[\boldsymbol{f}^{\mathrm{DOF}\top} \quad \boldsymbol{f}^{D\top} \quad \boldsymbol{t}^{\top} \quad \boldsymbol{b}^{D\top}\right]^{\top} = \boldsymbol{b}$ (cf. (46))
for $a \in \{1, \dots, n_1\}$ **do**
  // Neighbor search
  for $\kappa \in \{1,2,3,4\}$ **do**
    $X_\kappa^{\mathrm{loc}\,x_1^{(a)}} = \{x_\kappa^{(b)} \mid x_\kappa^{(b)} \in \mathcal{H}(x_1^{(a)})\} \leftarrow$ Neighbors of nodes of node $x_1^{(a)}$ that carry constraints of type κ
  end for
  // Evaluate local quantities
  $\mathcal{D}\boldsymbol{\Sigma}_{fc}(x_1^{(a)})$, $\boldsymbol{\Sigma}_{cc}^{-1}$, $\boldsymbol{H}_c$, $\mathcal{D}h^{\top}(x_1^{(a)}) \leftarrow$ Evaluate covariances and explicit basis functions for nodes in $\{X_\kappa^{\mathrm{loc}\,x_1^{(a)}}\}_{\kappa=1,\dots,4}$
  // Solve local systems of equations
  $\boldsymbol{C}_\Sigma = \mathcal{D}\boldsymbol{\Sigma}_{fc}(x_1^{(a)})\boldsymbol{\Sigma}_{cc}^{-1}$
  $\boldsymbol{C}_H = \boldsymbol{\Sigma}_{cc}^{-1}\boldsymbol{H}_c$
  // Evaluate partial derivatives of local interpolation operator $\boldsymbol{A}_D^{\mathrm{loc}\,x_1^{(a)}} = \mathcal{D}\boldsymbol{A}^{\mathrm{loc}\,x_1^{(a)}}$ at location $x_1^{(a)}$
  $\boldsymbol{A}_D^{\mathrm{loc}\,x_1^{(a)}} = \boldsymbol{C}_\Sigma + (\mathcal{D}h^{\top}(x_1^{(a)}) - \mathcal{D}\boldsymbol{\Sigma}_{fc}(x_1^{(a)})\boldsymbol{C}_H)(\boldsymbol{H}_c^{\top}\boldsymbol{C}_H)^{-1}\boldsymbol{C}_H^{\top}$
  // Assemble
  $\boldsymbol{K} \leftarrow$ Consistently insert rows of $\boldsymbol{A}_D^{\mathrm{loc}\,x^{(a)}}$ into the global matrix
end for
// Solve
$\boldsymbol{f}^{*\mathrm{DOF}} \leftarrow$ Rearrange and solve the global system of equations (cf. (48))

In summary, in order to arrive at the global system matrix within the MKC framework, many relatively small linear systems of equations need to be solved. This is the price in lieu of numerical integration which is required in other strategies. The computation of the solutions to these linear systems of equations is massively parallelizable.

Subsequent to the assembly of the global system matix, the corresponding global linear system of equations needs to be solved. By construction, a system matrix that arises from a MKC approach is sparse. In contrast to some other methods, the MKC system matrix is not necessarily symmetric for elliptic problems.

5.2 | Characteristic length scale parameter

The characteristic length scale λ controls the complexity of the Kriging function. The larger the value of λ, the more complex the Kriging function becomes. Complexity in the present sense describes the capacity to recover different functions. In the context of parametric, linear regression models for instance, complexity is usually associated with the number of basis functions used.

Collocation methods are based on the assumption that a certain target function can be represented by a finite set of discrete function values. These discrete function values are provided at so-called collocation points or nodes. The spatial arrangement of these nodes needs to be sufficiently dense in order to ensure that the main characteristics of the target function are captured.

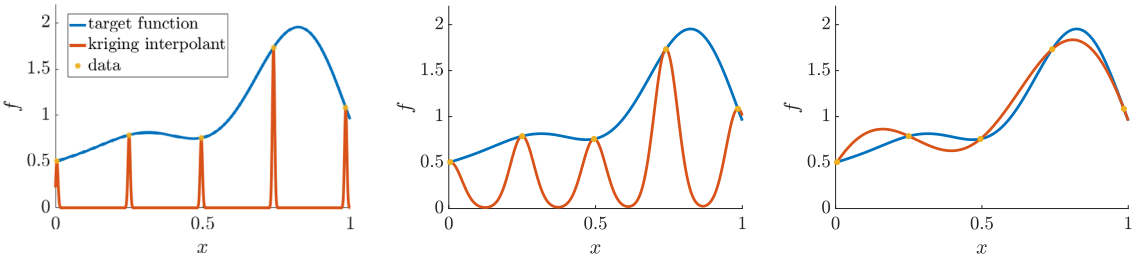


FIGURE 2 Illustration of trade-off between prior belief and data: Data generating target function, data points and MK interpolant with $\lambda = 0.004$ in (A), $\lambda = 0.04$ in (B), and $\lambda = 0.4$ in (C)

A Kriging function is a Bayesian model and as such it represents a trade-off between data and prior beliefs. In order to make this clearer, consider the following example: Let there be an unknown target function f that one would like to recover based on a finite number of (noise free) function evaluations $f^{(a)}$ with $f^{(a)} = f(x^{(a)})$ and $a \in \{1, \dots, n\}$. This inference task can only be achieved if certain assumptions regarding the target function are put in place. These assumptions are referred to as prior beliefs in Bayesian terminology. In the present case, the most elementary prior belief consists in the assumption that the discrete function values $f^{(a)}$ provide sufficient information for the recovery of the target function f. There are in principle infinitely many possibilities to formulate such a prior belief concretely. For instance, one could postulate that the function f is the constant zero function almost everywhere except at those locations where empirical data is available. A compliant a posteriori interpolant would then read

$$\hat{f}(x) = \begin{cases} f^{(a)} & \text{if } x = x^{(a)} \\ 0 & \text{otherwise} \end{cases}. \tag{50}$$

Obviously, this function is discontinuous and the influence of the data on the interpolant is degenerate. In order to come up with a more "plausible" interpolant, one has to increase the influence of the data and reduce the impact of the prior belief. One possibility to achieve this would be to place smooth, locally supported functions at the collocation points and weight them with the corresponding function values, that is:

$$\hat{f}(x) = \begin{cases} f^{(a)} W(||x - x^{(a)}||) & \text{if } x \text{ near } x^{(a)} \\ 0 & \text{otherwise} \end{cases}, \tag{51}$$

where $W(.)$ is some bell-shaped, nonnegative function with $W(0) = 1$.

In a Kriging framework, the relative weighting of data and prior beliefs is controlled by the length scale parameter λ. An illustration of this is provided in Figure 2. For larger values of λ, the impact of the data is increased. We argue that the prior knowledge regarding the solution of a PDE tends to zero. The collocation points are thus the only source of information for the construction of an approximant. Consequently, the value of the length scale parameter λ should be as large as possible.

In the examples given in Section 6, the greatest possible value for the length scale parameter was selected which still ensured a good conditioning of the covariance matrices. This principle implies that the value of λ is to depend on the characteristic discretization length Δx. The smaller Δx the smaller λ needs to be. Specific choices for the exact relations between λ and Δx will be given in the respective examples.

6 | NUMERICAL EXPERIMENTS

In order to demonstrate the efficacy and behavior of the proposed MKC scheme, it is applied to 1D and 2D PDEs. The investigation of the 1D PDE example serves as a simple demonstrator. It demonstrates the elementary functionality of the method and the improvements achieved by means of the consistent approach for imposing natural BCs. It shall be

pointed out once more that the *consistent* approach differs from *conventional* strategies in that the PDE is enforced not only at nodes that belong to the interior of the domain but also at those nodes that lie on the domain's surface. Conversely, the localized approximants that are centered at nodes that belong to the interior of the domain also have to satisfy the BC.

As a 2D example, the 2D elasticity equations are employed. In particular, a 2D beam under axial and transverse loading conditions is considered. The former configuration corresponds to what is often referred to as a patch test. Herein, the linear displacement field solution is known analytically. In order to quantify errors in the transverse loading configuration, a numerical reference solution based on the finite element (FE) method is utilized.

6.1 | MKC for 1D PDE

The differential equation under consideration in this section is given by

$$\partial_{xx}^2 f = -cf \quad \text{in } \Omega = (0,1) \tag{52}$$

$$f = f_0 \quad \text{on } \Gamma_D \text{ at } x = 0 \tag{53}$$

$$\partial_x f = t_1 \quad \text{on } \Gamma_N \text{ at } x = 1, \tag{54}$$

where $c = 40$ is a constant and $f_0 = 1$ and $t_1 = 3.33$ are the Dirichlet and Neumann values prescribed at the boundary locations $x = 0$ and $x = 1$, respectively. The analytical solution to this differential equation with the given BCs is

$$f(x) = f_0 \cos(\sqrt{c}\, x) + \frac{t_1 \frac{1}{\sqrt{c}} + f_0 \sin(\sqrt{c})}{\cos(\sqrt{c})} \sin(\sqrt{c}x). \tag{55}$$

In this example, the relation between the characteristic length scale λ of the Kriging approximant and the interaction radius δ is given by $\lambda = 10\ \delta$. The interaction radius δ is selected such that each node interacts with at least two other nodes to its left and to its right (except of those nodes, of course, that are close to the boundary). The explicit basis function (cf. (12) from Section 2.2) is first oder complete and has the form $h(x)^\top = [1\ x]$.

In order to quantify the deviation between the Kriging approximation and the analytical reference, the following absolute error measure is used:

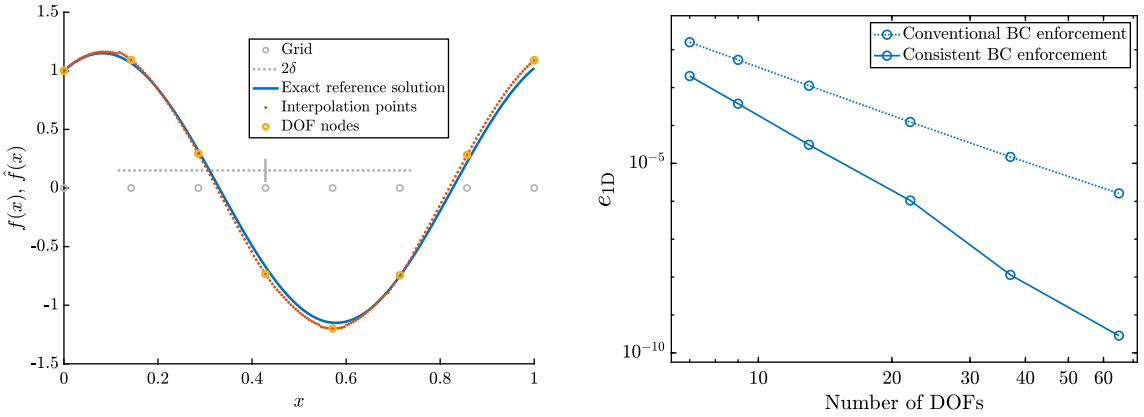
$$e_{1\mathrm{D}} = \frac{1}{n_{err}} \sum_{a=1}^{n_{err}} (\hat{f}(x_{err}^{(a)}) - f(x_{err}^{(a)}))^2, \tag{56}$$

where $\{x_{err}^{(a)}\}_{a=1,\dots,n_{err}}$ is a set of uniformly distributed error evaluation locations. The MK approximation $\hat{f}$ of the solution to (52)-(54) is visualized in Figure 3A. In this graph, the exact reference solution is depicted in blue, yellow circles represent the MKC approximation at nodes that carry the DOFs $\boldsymbol{f}^{\mathrm{DOF}}$ or the Dirichlet information $\boldsymbol{f}^D$. Nodes which carry DOFs are referred to as *DOF nodes*. The locations in between the DOF nodes, at which the approximant is also evaluated, are called *interpolation points*. The values of the MKC approximation at these interpolation points are depicted in red. The small discontinuities, which are present in the MKC approximation, are an expected consequence of the localization of the approximant.

In Figure 3B, the conventional method for imposing Neumann BCs in collocation methods is compared to the consistent strategy proposed in this work. In the example at hand, the consistent strategy improves the accuracy of the approximation by one to three orders of magnitude.

6.2 | MKC for 2D linear elasticity

We consider as an example for the procedure of constructing an MKC discretization of a PDE, the equilibrium equations of 2D (plane stress) linear, isotropic elasticity. These can be written as



(A) (B)

FIGURE 3 1D example differential equation: (A) MKC approximation of (52) to (54) with $n = 8$ DOF nodes. (B) Comparison between convergence curves of the proposed, consistent strategy for BC enforcement (consistent BC-enforcemnt) and the conventional strategy (Conventional BC-enforcement). The displayed values are based on the e_{1D}-error measure defined in (56)

$$\begin{cases} \mathcal{P}u = b & \text{in } \Omega \\ \mathcal{B}(x)u = t & \text{on } \Gamma_N \\ u = u_0 & \text{on } \Gamma_D \end{cases} . \tag{57}$$

The linear differential operators $\mathcal{P}$ and $\mathcal{B}$ are

$$\mathcal{P} = \frac{E}{1-\nu^2} \begin{bmatrix} \partial_{xx} + \frac{1-\nu}{2}\partial_{yy} & \frac{1+\nu}{2}\partial_{xy} \\ \frac{1+\nu}{2}\partial_{xy} & \frac{1-\nu}{2}\partial_{xx} + \partial_{yy} \end{bmatrix}, \tag{58}$$

$$\mathcal{B} = \frac{E}{1-\nu^2} \begin{bmatrix} n_x\partial_x + n_y\frac{1-\nu}{2}\partial_y & n_y\frac{1-\nu}{2}\partial_x + n_x\nu\partial_y \\ n_y\nu\partial_x + n_x\frac{1-\nu}{2}\partial_y & n_x\frac{1-\nu}{2}\partial_x + n_y\partial_y \end{bmatrix}, \tag{59}$$

with E and ν denoting Young's modulus and Poisson's ratio, respectively. For the sake of improved readability, the notation used in (58) to (59) uses letters x and y instead of numbers as indices of vector valued quantities. In particular this entails the possibility to rewrite the generic linear interpolation operator as

$$\tilde{\boldsymbol{A}}_x \rightarrow \begin{bmatrix} \tilde{\boldsymbol{A}}_x \\ \tilde{\boldsymbol{A}}_y \end{bmatrix}. \tag{60}$$

As outlined in Section 2.1, the construction of a suitable linear interpolation operator requires the knowledge of the constraint operators C_1, C_2, C_3, and C_4. The first two operators are trivial. The third operator C_3, which is used to enforce Neumann BCs, was introduced in general terms in Section 2.3.2 and involves the matrix L. In the present example, this matrix is obtained by casting the differential operator $\mathcal{B}$ from (59) into a format that is compatible with the rolled out Jacobian defined in (25). It then reads

$$\boldsymbol{L}(x) = \frac{E}{1-\nu^2} \begin{bmatrix} n_x & n_y\frac{1-\nu}{2} & n_y\frac{1-\nu}{2} & n_x\nu \\ n_y\nu & n_x\frac{1-\nu}{2} & n_x\frac{1-\nu}{2} & n_y \end{bmatrix}. \tag{61}$$

As pointed out in Section 2.3.3, the definition of the operator C_4 which ensures the fulfilment of the PDE at discrete locations on the Dirichlet boundary is analogous to that of C_3 and therefore not stated explicitly. Once an interpolant

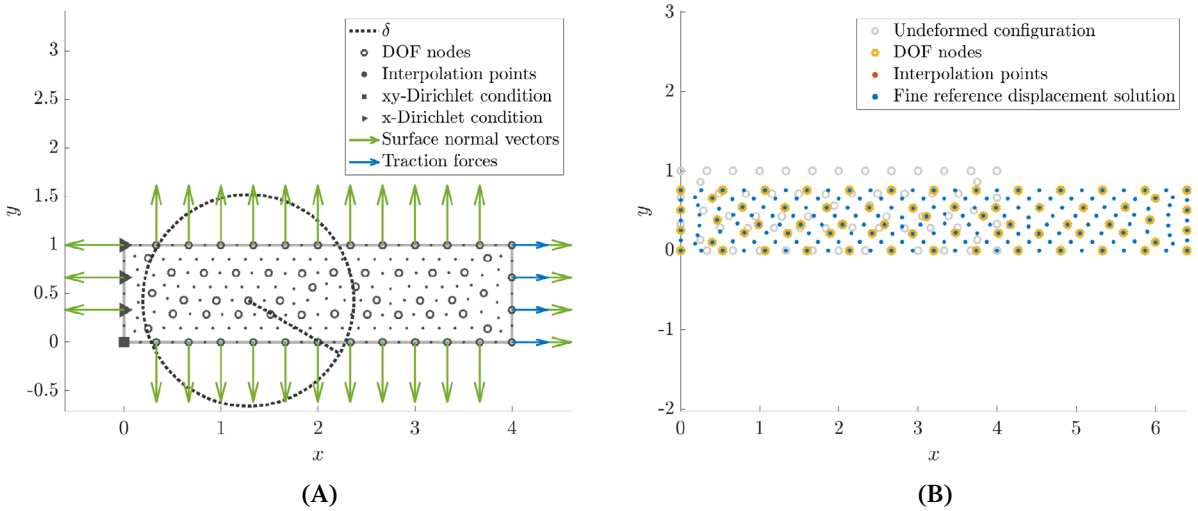


FIGURE 4 Patch test configuration discretized with $n = 65$ DOF nodes: (A) Numerical set-up. (B) Visualization of the MKC displacement approximation that recovers the analytical reference solution exactly. The red interpolation points cannot be seen in the plot since they coincide perfectly in size an position with the blue markers of the reference solution

has been constructed that satisfies all constraints, the differential operator $\mathcal{P}$ is applied to it. Evaluating the resulting expression at discrete locations yields a linear system of equations.

The explicit basis function that was used in all subsequently discussed examples is a complete second order polynomial. The corresponding linear model (cf. (12) from Section 2.2) then is

$$h(x) = \begin{bmatrix} h_x(x,y) & 0 \\ 0 & h_y(x,y) \end{bmatrix}^{\mathsf{T}}, \tag{62}$$

where $h_x(x,y) = h_y(x,y) = \begin{bmatrix} 1 & x & y & x^2 & y^2 & xy \end{bmatrix}^{\mathsf{T}}$.

Error measures and overview

In this section, the performance of the proposed method is assessed by means of a 2D rectangular beam under axial (patch test) and transverse loading conditions. The geometry of the beam is depicted in Figure 4A. Its material parameters are $E = 1$ and $\nu = 0.4$. In order to quantify the deviation between the Kriging approximation and the exact reference, two relative error measures are used. The first one, based on v. Mises stresses, is

$$e_{\mathrm{vM}} = \frac{\sum_{a=1}^{n_{err}} |\sigma_{\mathrm{vM}}(x_{err}^{(a)}) - \hat{\sigma}_{\mathrm{vM}}(x_{err}^{(a)})|}{\sum_{a=1}^{n_{err}} \sigma_{\mathrm{vM}}(x_{err}^{(a)})}, \tag{63}$$

where $\sigma(\cdot)$ and $\hat{\sigma}(\cdot)$ denote the reference v. Mises stress function and its MK approximation, respectively. The second error measure is displacement based. It reads

$$e_{\mathrm{u}} = \frac{1}{n_{err}} \sum_{a=1}^{n_{err}} \frac{\|u(x_{err}^{(a)}) - \hat{u}((x_{err}^{(a)})\|}{\|u(x_{err}^{(a)})\|}, \tag{64}$$

where $u(\cdot)$ and $\hat{u}(\cdot)$ denote the reference displacement function and its MK approximation, respectively. As (63) and (64) suggest, the e_{vM}- and the e_{u}-error measures are averages over spatially sampled, local errors. In order to avoid too low sampling rates in coarser discretizations, the local errors are not evaluated at DOF nodes but at uniformly distributed error evaluation locations $\{x_{err}^{(a)}\}_{a=1,\dots,n_{err}}$.

The exact reference solution is known in closed form only for the patch test configuration. In transverse load cases where no analytical solution is available, a very fine FE reference solution based on a regular grid comprising about 3.5×10^5 elements and about 1.7×10^5 nodes was generated with the standard Matlab PDE-tool.* In order to relate the performance of the MKC scheme to that of an established method, FE approximations based on coarser discretizations were also computed.

Before presenting the results, it is worthwhile elucidating the choice of specific examples. To this end, the following structured overview is provided:

1. *Patch test configuration*: To demonstrate MKC's ability to reproduce constant stress solutions, it is subjected to the patch test.
2. *Bending test configuration*: In order to show how the solution accuracy improves under refinement of the discretization in a scenario for which no analytical solution is available, the MKC method is subjected to a bending test. This problem is challenging since its solution comprises singularities at domain corners.
3. *Bending test configuration (Impact of irregularity of nodal arrangement)*: Here, the methods (in)sensitivity with respect to the degree of irregularity of the discretization is investigated.
4. *Bending test configuration (Consistent vs conventional imposition of v. Neumann BCs)*: The improvements that are achieved solely by imposing natural BCs in the proposed, consistent manner are established here.
5. *Bending test configuration (PDE-constrained Dirichlet nodes)*: This part is dedicated to the demonstration of the effects of applying PDE constraints at Dirichlet nodes in combination with a consistent imposition of natural BCs.
6. *Cook's membrane with hole*: All set-ups described above involve a simple, rectangular geometry. In order to demonstrate the method's efficacy in the case of a more complicated geometry, a variation of Cook's membrane is used as an example.
7. *Comparison with Peridynamics*: In order to embed MKC into the context of other truly mesh free discretization schemes, a comparison between MKC and Peridynamics is made.

6.2.1 | Patch test configuration

In Figure 4A, the numerical set-up for the patch test is visualized. The green arrows indicate the surface normal vectors at nodes on the boundary of the domain. Blue arrows indicate the traction forces $t = [0.6\ 0]^\top$. The grey circles represent nodes that carry the DOFs (DOF nodes), whereas the grey dots mark interpolation points. The latter are arbitrary discrete locations at which the approximant is evaluated in order to assess the approximation quality beyond the DOF node locations. The DOF nodes are irregularly distributed over the domain. In the patch test configuration, the horizontal component of the deformation at the nodes on the left edge of the rectangular domain is constrained to be zero. At the node in the lower left corner, both the vertical and the horizontal displacements are constrained to be zero. The dotted line marks one exemplary circular region $\mathcal{H}(x^{(a)})$ with radius δ centred at $x^{(a)}$ that comprises all nodes that interact with $x^{(a)}$ in the localized scheme.

The deformed configuration is depicted in Figure 4B. Blue dots mark the displacements of the reference solution at the DOF nodes of the MKC scheme and at the interpolation points. Yellow circles and red dots represent, respectively, the displaced DOF nodes and the displacements at interpolation points. The interpolated displacement markers cannot be seen because they coincide in size and location with the markers of the reference displacement. The displacement-based and the stress-based relative errors are computed to $e_\mathrm{u} \approx 7 \times 10^{-12}$ and $e_\mathrm{vM} \approx 9 \times 10^{-12}$. In summary, it can be asserted that even with a rather coarse and irregular arrangement of discretization nodes the patch test is passed. This result is expected since the explicit basis h was selected to be second order complete and thus the solution space comprises the exact solution.

It is important to remark that this result was obtained for the specific choice of surface normal vectors that is depicted in Figure 4A. The choice of the surface normal vectors at nodes that lie at corners of the domain is, however, not unique. Alternatively the surface normal vectors at corners could be defined as scaled vector sums of the normal vectors of the adjacent straight edges. Naturally, such a modification in the numerical set-up would slightly alter the solution. We argue that in scenarios, in which the proposed method could be applied, there are mostly smooth surfaces in which case the described nonuniqueness becomes negligible.

*MATLAB and Partial Differential Equation Toolbox Release 2019a, The MathWorks, Inc., Natick, Massachusetts, United States.

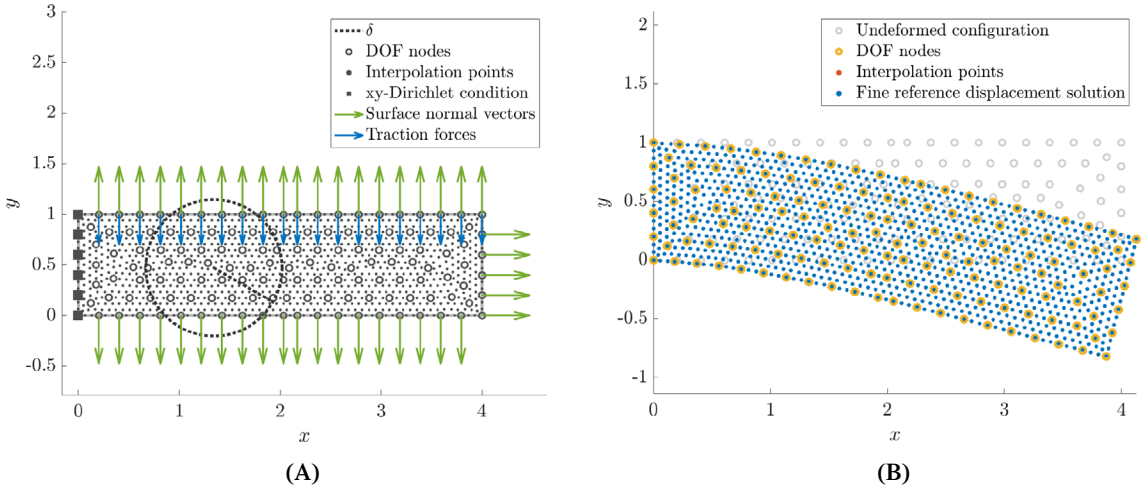


FIGURE 5 Bending test configuration discretized with $n = 252$ DOF nodes: (A) Numerical set-up. (B) Visualization of MKC displacement approximation that coincides well with the reference solution. The red interpolation points cannot be seen in the plot since they coincide perfectly in size an position with the blue markers of the reference solution

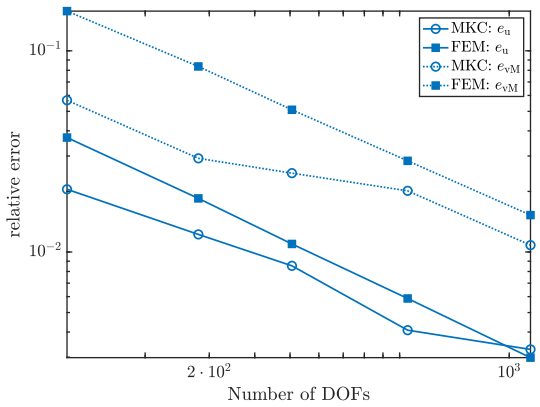


FIGURE 6 Bending test configuration: Graphs of displacement- and stress errors over number of DOFs for MKC and FE. The FE-results are intended to serve as a reference frame and not as a competitive benchmark. The accuracy of the MKC approximations improves under refinement

6.2.2 | Bending test configuration

In Figure 5A, the numerical set-up for the bending test is visualized. The meaning of the components of the plot is analogous to Figure 4A. The discretization nodes are distributed uniformly but irregularly over the domain. In the bending test, both the horizontal and the vertical displacements at all nodes on the left edge of the domain are constrained to be zero. Note that in this example the Dirichlet boundary nodes are PDE-constrained in the sense of Section 2.3.3 and the Neumann BCs are imposed in the proposed consistent way. The impact of either omitting the PDE constraints at the Dirichlet nodes or imposing Neumann BCs in the conventional way is discussed below. The tractions $t = [0 \; -0.002]^\mathsf{T}$ act on the top edge of the domain.

The deformed configuration is depicted in Figure 5B. By visual judgement, the MK approximation and the reference solution coincide very well. The results of a quantitative assessment are displayed in Figure 6 in which both the e_{vM} and the e_u errors are displayed as a function of the number of DOFs. The relation between the length scale parameter λ, the interaction radius δ, and the characteristic discretization width is prescribed as $\lambda = \frac{2}{5}\delta \approx \frac{4}{3}\Delta x$. For orientation, the errors of a FE approximation with linear, triangular elements and as many DOFs as were used in the generation of the MKC approximations are also displayed. The MK scheme performs well. A peculiarity that can be observed in Figure 6 are the kinks in the relative error curves. These kinks can be explained by the fact that our domain discretization algorithm does not produce perfectly uniform and regular nodal arrangements. As will be discussed in more detail below, the accuracy of the approximation exhibits some sensitivity with respect to the nodal configuration.

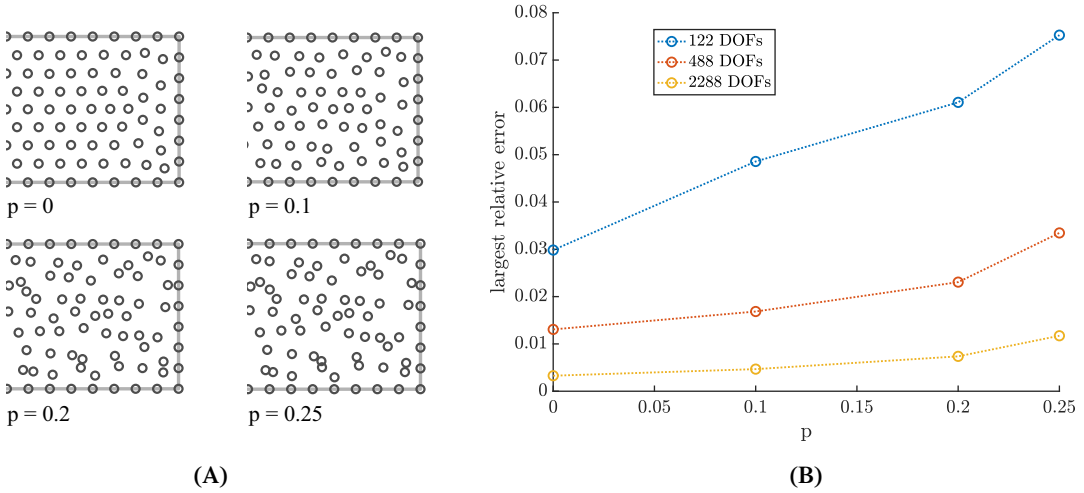


FIGURE 7 Bending test configuration with irregular nodal arrangement: (A) Illustrations of nodal arrangements generated with various values for p. (B) Displacement errors of the MKC approximations over value of perturbation parameter p (cf. (65)) for various levels of refinement

6.2.3 | Bending test configuration (impact of irregularity of nodal arrangement)

Next, the impact of the degree of irregularity of the nodal arrangement on the accuracy of the MK approximant is investigated. To this end, an irregular nodal arrangement is generated by adding noise to the coordinates of the uniformly arranged nodes. Formally, this is achieved as follows:

$$x^{(a),\text{pert}} = x^{(a)} + p\ \Delta x\ \xi^{(a)}, \quad \text{with} \quad \xi^{(a)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{65}$$

and where p is a perturbation parameter that controls the degree of irregularity. Note that only nodes that do not lie on the boundary of the domain are perturbed. Note further that the perturbation is proportional to the discretization width Δx. Figure 7B displays the relation between the degree of irregularity of the nodal arrangement (value of the perturbation parameter) and the corresponding sensitivity of the results. The sensitivity of the results with regards to irregularity is determined as follows: First, MKC solutions are generated for 15 different realizations of $\xi^{(a)}s$. Second, the errors of all 15 solutions are computed. Finally, the largest of these errors is used as a representation of the sensitivity with regards to the irregularity of the nodal arrangement and plotted as a function of the perturbation parameter. Figure 7A shows illustrations of nodal arrangements generated with various values for the perturbation parameter p. In our opinion, these graphs reveal that the accuracy of the MKC approximations deteriorates only moderately as the degree of irregularity increases. Moreover, it can be observed that MKC approximations are less sensitive with regards to the degree of irregularity as the discretization becomes more refined.

6.2.4 | Bending test configuration (consistent vs conventional imposition of Neumann boundary conditions)

In Section 6.1, the consistent strategy for imposing Neumann BCs was demonstrated by means of a 1D differential equation. Improvements of similar conspicuousness can also be observed in the case of the 2D linear elasticity example. The stress- and displacement-errors (63) and (64) are plotted over the number of DOFs in Figure 8. By comparison of red and blue graphs, it can be asserted that employing the consistent strategy improves the accuracy of the MK approximation significantly. What is more, the increase in error for increasing numbers of nodes for certain levels of discretization refinements that can be observed in Figure 8 is eliminated by employing the consistent strategy. Note that all results shown in Figure 8 were obtained without applying PDE-constraints to Dirichlet nodes. The positive effects of this are discussed below.

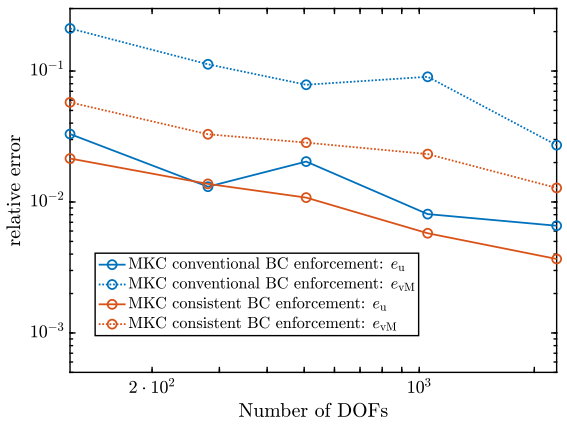


FIGURE 8 Bending test configuration without PDE-constraints at Dirichlet nodes: Displacement and stress errors over number of DOFs for MKC and FE. The consistent strategy for imposing BCs clearly effectuates an error reduction. Again, the FE-results represent an uncompetitive frame of reference

6.2.5 | Bending test configuration (PDE-constrained Dirichlet nodes)

In Section 2.3.3, it was elucidated how conditioning an approximant on higher order derivative information can be of use in a collocation context. In particular, if the approximant is PDE-constrained at the locations of Dirichlet boundary nodes, the approximation accuracy in these regions is significantly improved. Figure 9A visualizes the vertical tractions at the left edge of the beam. The solid gray line and the dashed gray line represent the traction values of the fine and coarse FE-reference solution, respectively. The coarse FE-reference approximations are based on meshes with as many DOFs as were used in the generation of the MKC approximations. The MKC approximations for an exemplary discretization with and without PDE-constraints are depicted by a red and a blue line, respectively. The larger dots in the graphs represent the tractions at DOF nodes.

As Figure 9A shows, omitting the PDE-constraints in the MKC scheme can give rise to an implausible, in this case oscillatory traction approximation. By introducing the PDE-constraints this circumstance is significantly alleviated. In Figure 10, the traction approximations for other levels of discretization refinement are displayed. It can be seen that the imposition of PDE-constraints gives rise to approximations that are qualitatively more plausible than those produced without these constraints. In order to quantify the improvements induced by the PDE-constraints, the following traction error measure is defined:

$$e_{\mathrm{t,y,dir}} = \frac{1}{n_{err,\Gamma}} \sum_{a}^{n_{err,\Gamma}} \frac{|\hat{t}_y^{(a)} - t_y^{(a)}|}{|t_y^{(a)}|}, \tag{66}$$

where in the present case the index a is meant to run over all $n_{err,\Gamma}$ error evaluation locations on the Dirichlet boundary. The y-component of the fine FE reference traction and its MKC approximation are denoted by $t_y^{(a)}$ and $\hat{t}_y^{(a)}$, respectively. The results of the error analysis shown in Figure 9B confirm that for most discretizations the accuracy of the traction approximation is significantly improved if PDE-constraints are included. As can be inferred from Figure 9C, also the global errors are clearly reduced as a consequence of the PDE-constraints.

The local residuals $\|\mathcal{P}(\hat{u}(x^{(a)})) - b(x^{(a)})\|$ of the MKC approximation for an exemplary discretization are depicted in Figure 11. The examination of the local residuals reveals the reason for which the consistent imposition of natural BCs and the application of PDE-constraints at the Dirichlet nodes are advantageous: By construction of the standard method, the local residuals vanish at all DOF nodes that do not lie on the boundary. At DOF nodes that do lie on the boundary, the approximant is not required to satisfy the differential equation. Hence, the local residuals at these nodes do not vanish which in turn leads to increased local residual values in the vicinity of boundaries (cf. Figure 11A). As one sees by comparison of Figure 11A and B, the consistent strategy for imposing natural BCs causes local residuals at the corresponding boundary nodes to vanish. By additionally introducing PDE constraints, the approximant also satisfies the differential equation at Dirichlet nodes regardless of the values that are prescribed at these locations. The local residuals in between the DOF nodes are thus reduced as well (cf. Figure 11C) and the accuracy of the approximation is improved.

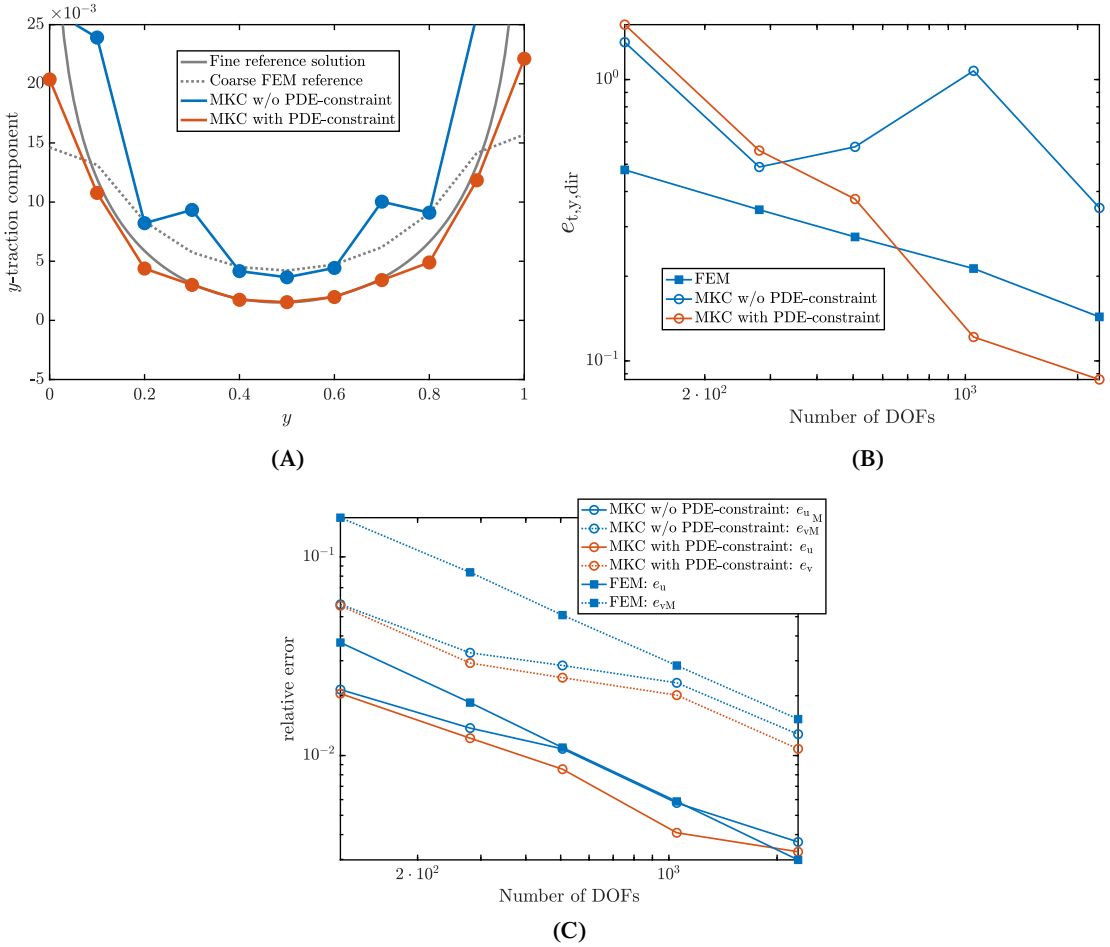


FIGURE 9 Bending test configuration: (A) Fine reference and various approximations of vertical tractions at the left edge of the beam. The approximations are based on discretizations comprising 537 DOF nodes. By applying PDE constraints at Dirichlet nodes in the MKC scheme, the plausibility of the boundary fluxes is significantly improved. (B) Graph of relative traction error $e_{\mathrm{t,y,dir}}$ of the left edge over number of DOF nodes for MKC and FE confirming the improvements effectuated by the PDE constraints. (C) Displacement and stress errors over number of DOFs for MKC and FE. The PDE-constraints clearly effectuate an error reduction

6.2.6 | Cook's membrane with hole

The 2D numerical examples discussed above are all based on a simple rectangular geometry. In this section, an example involving a more complicated geometry is investigated. The geometry that is selected to this end is a variation of Cook's membrane (cf. Figure 12A). This geometry has acute and obtuse angles as well as a curved surface. Figure 12B shows the deformation that results from the traction force $t = [0 \; -0.02]^{\mathsf{T}}$ acting on the top edge of the domain. Stress and displacement errors are shown in Figure 12C. Albeit being larger than those of the FE results, the errors of the MKC solution reduce consistently as the number of DOFs is increased. The v. Mises stresses as obtained by the MKC method are visualized in Figure 13A. The stresses are visualized as colored circles in order to reflect the mesh-free nature of MKC. The MKC approximation of the v. Mises stress field corresponds well to the fine reference solution which is depicted in Figure 13B.

6.2.7 | Comparison with Peridynamics

The motivation for developing a mesh-free method that enables a consistent and robust imposition of BCs arose in the context of an investigation of established mesh-free methods. In particular, Peridynamics and fully Lagrangean

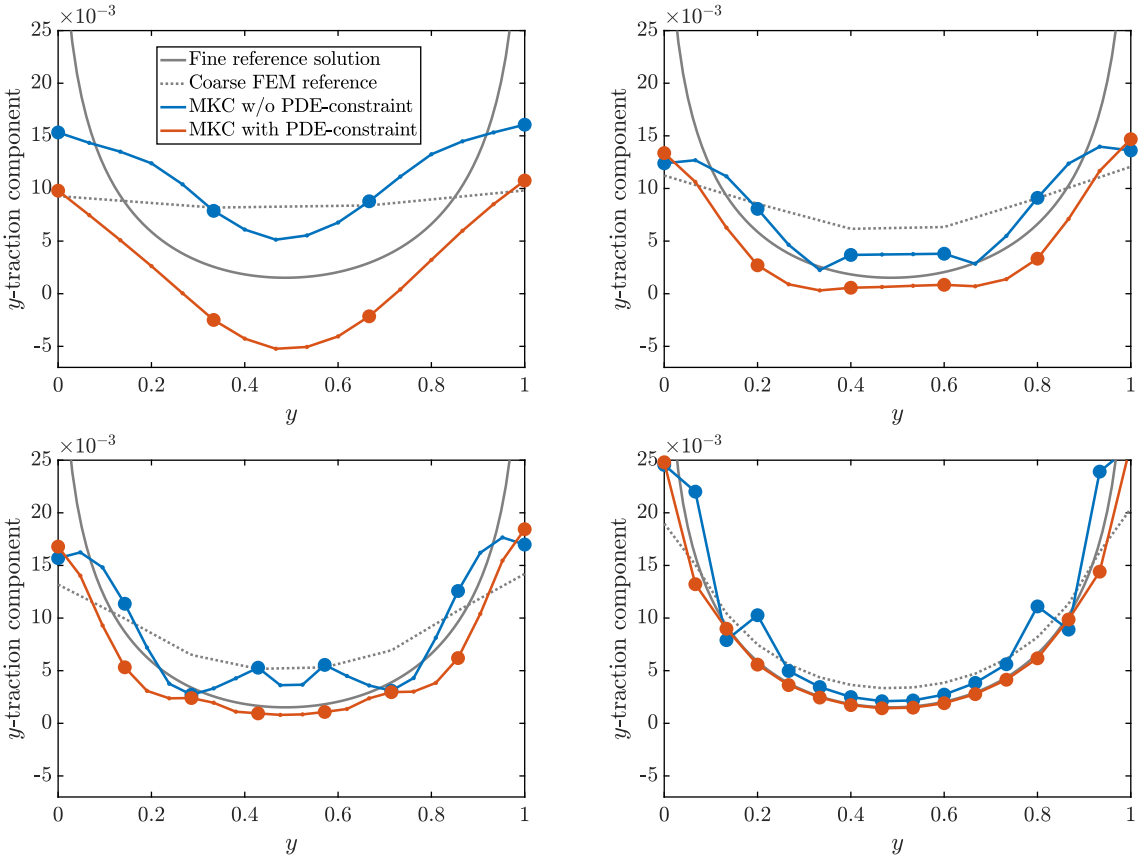


FIGURE 10 Bending test configuration: Fine reference and various approximations of vertical tractions at the left edge of the beam for various levels of discretization refinement. The discretizations comprise, from left to right, 65, 146, 261, 537, and 1160 DOF nodes. Larger dots represent tractions at DOF nodes, smaller dots (only present in the first three plots), represent tractions at interpolation points. The results based on an MKC scheme using PDE constraints appear more plausible throughout

Smoothed Particle Hydrodynamics (SPH) were considered. As shown in Reference 12, Peridynamics and SPH are equivalent approaches if Peridynamics is used with classical material models. These methods seem particularly appealing since they are truly mesh-free in that no background mesh is required for numerical integration. On the downside, however, these methods do not offer a consistent and reliable strategy for imposing BCs.

In this section, the results of a Peridynamics-based implementation of the beam are presented. As a material-model, the linear Peridynamic solid was used. In Figure 14A, the Peridynamics numerical set-up for the patch test configuration is visualized. Note that in Peridynamics traction forces cannot be applied directly. Instead, they have to be represented by volume forces that act on nodes belonging to a boundary region. Here, these boundary nodes are depicted in red. Dirichlet-type BCs are enforced by prescribing the deformations of those nodes that are closest to the corresponding boundary. Figure 14B shows the deformed configuration. This configuration reveals clearly that the technique conventionally used for imposing BCs in Peridynamics gives rise to unphysical artifacts in the deformation: At the left edge, a displacement discontinuity can be observed. At the right edge two things happen that are a consequence of the fact that traction forces have to be represented by volume forces: First, the beam widens in this region. Second, the beam bends slightly downwards. The second artifact occurs because the nodal arrangement is not perfectly regular. Thus, the resultant of the volume forces applied to each boundary node exerts a torque with respect to the centerline of the beam.

Figure 15A and B depicts the previously defined relative deformation error as a function of the number of DOFs for the bending and the patch test configurations, respectively. In the latter case, the FEM and MKC errors are not displayed since these methods pass the patch test, that is, they reproduce the analytical reference solution exactly. Note that the error comparison between Peridynamics and the other methods is to be interpreted with caution since the exact Peridynamics solution is identical to the solution in the classical theory only in the limit of $\delta \to 0$. These results are, nevertheless, suitable to illustrate the limitations of the strategies for imposing BCs in establishes mesh-free methods.

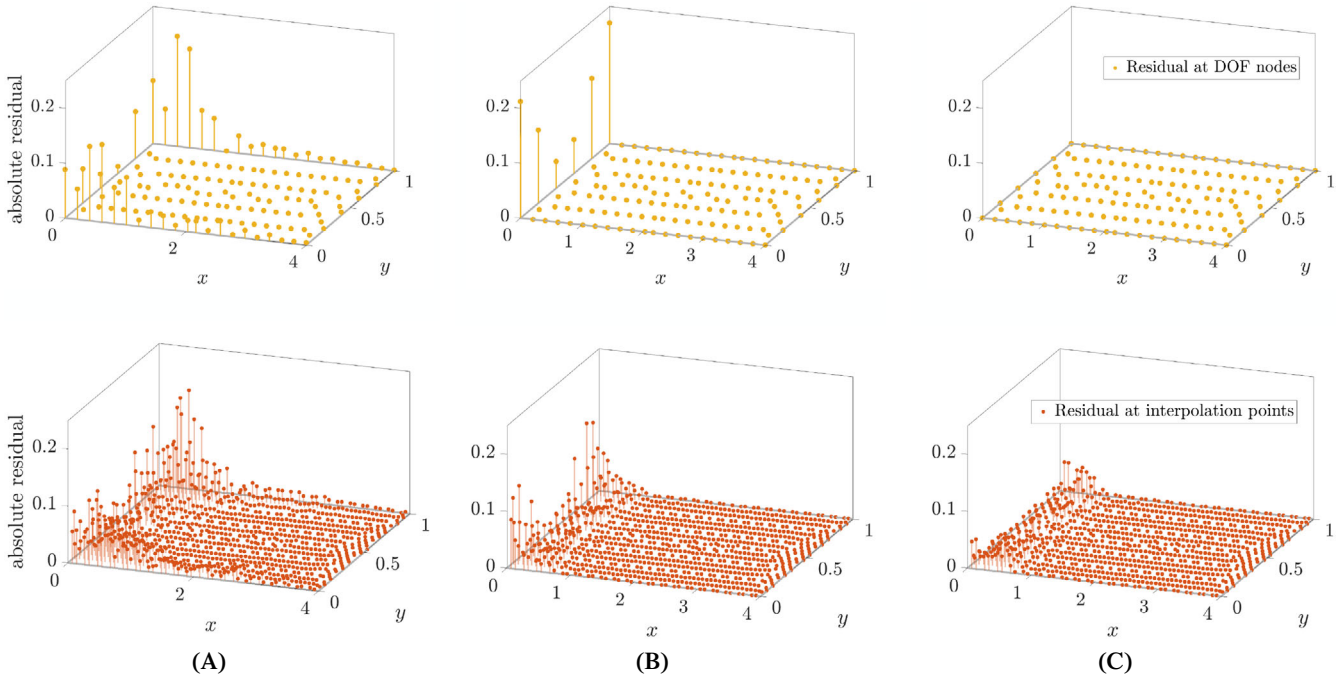


FIGURE 11 Bending test configuration: Local residuals of the MKC approximant based on a discretization comprising 261 DOF nodes. Top row: residuals at DOF nodes, bottom row: residuals at interpolation points. (A) *Without* PDE constraints at Dirichlet nodes, *conventional* imposition of BCs. (B) *Without* PDE constraints at Dirichlet nodes, *consistent* imposition of BCs. (C) *With* PDE constraints at Dirichlet nodes, *consistent* imposition of BCs

7 | POSSIBLE GENERALIZATION TO NONLINEAR PDES

In future work, the MKC method is to be explored further in the context of nonlinear PDEs. In many established, implicit strategies, a nonlinear problem is solved by decomposing it into a series of linear problems.[26] The same idea can be employed in MKC. Consider the generic nonlinear differential equation

$$\mathcal{D}^{\text{nonlin}}(f(x)) = b(x) \quad \text{in } \Omega \tag{67}$$

$$\mathcal{G}^{\text{nonlin}}(f(x)) = t(x) \quad \text{on } \Gamma_N \tag{68}$$

$$f(x) = f^D(x) \quad \text{on } \Gamma_D. \tag{69}$$

In order to approximate the solution to such a system of differential equations, the presented collocation approach would demand that the approximant satisfies the differential equation (67) at discrete locations within and on the boundary of the domain. Moreover, the approximant needs to be constructed such that it satisfies the BCs a priori, that is, independent of the values of its discrete DOFs. In particular, it has to satisfy the derivative constraints in (68). However, due to the nonlinearity of $\mathcal{G}^{\text{nonlin}}(\cdot)$, such an approximant cannot directly be realized by means of the methods presented above. An adequate approximant can, however, be constructed for a linearized version of $\mathcal{G}^{\text{nonlin}}(\cdot)$. Based on this insight, a solution strategy for nonlinear problems is developed and outlined in the remainder of this section.

In a nonlinear setting, the approximant $\hat{f}$ to the exact solution function f could for instance be obtained by means of the Newton-Raphson (NR) method. In order to illustrate how this algorithm could be realized within the MK framework, one NR iteration step is presented. To this end, let $\hat{f}|_i$ denote the MK approximant at the end of the ith Newton iteration. In the course of the subsequent iteration the NR algorithm yields the function increment $\Delta\hat{f}_{i+1}$. As compared to $\hat{f}|_i$, the updated approximant $\hat{f}|_{i+1} = \hat{f}|_i + \Delta\hat{f}_{i+1}$ is (in case of convergence) closer to the exact solution f. In order to determine $\Delta\hat{f}_{i+1}$, the following steps need to be undertaken:

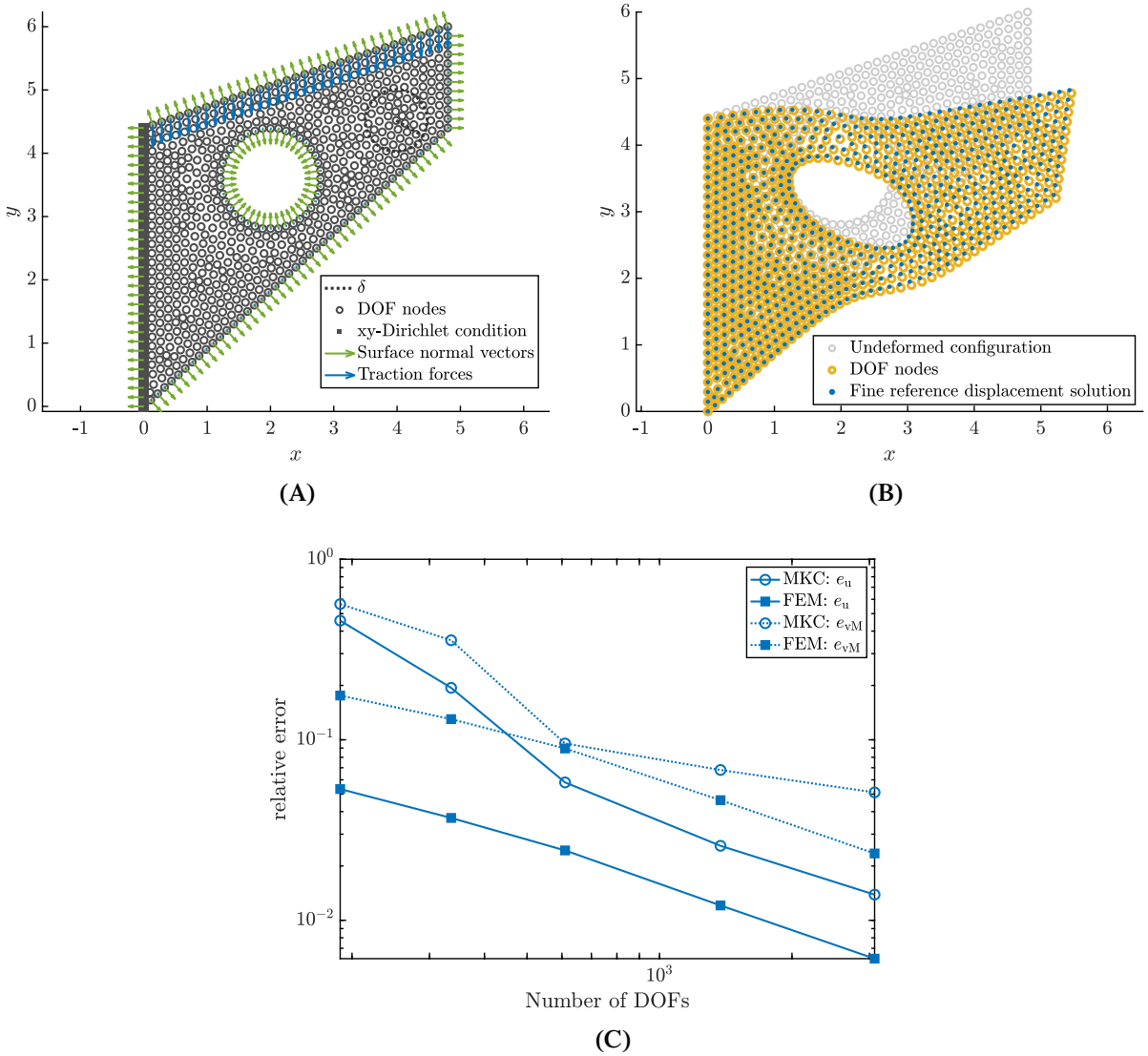


FIGURE 12 Configuration with complicated geometry: (A) Numerical set-up. (B) Visualization of MKC approximation and fine reference solution. (C) Graphs of displacement and stress errors over number of DOFs for MKC and FE

1. Linearize the derivative constraints $\mathcal{G}^{\text{nonlin}}(\cdot)$ and the differential operator $\mathcal{D}^{\text{nonlin}}(\cdot)$ around the current approximant $\hat{f}|_i$:

$$\mathcal{G}^{\text{nonlin}}(\hat{f}|_i + \Delta\hat{f}_{i+1}) \approx \mathcal{G}^{\text{nonlin}}(\hat{f}|_i) + \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \mathcal{G}^{\text{nonlin}}(\hat{f}|_i + \epsilon\Delta\hat{f}_{i+1}) \tag{70}$$

$$= \mathcal{G}^{\text{nonlin}}(\hat{f}|_i) + \mathcal{G}^{\text{lin}}\Delta\hat{f}_{i+1}, \tag{71}$$

$$\mathcal{D}^{\text{nonlin}}(\hat{f}|_i + \Delta\hat{f}_{i+1}) \approx \mathcal{D}^{\text{nonlin}}(\hat{f}|_i) + \frac{d}{d\epsilon}\bigg|_{\epsilon=0} \mathcal{D}^{\text{nonlin}}(\hat{f}|_i + \epsilon\Delta\hat{f}_{i+1}) \tag{72}$$

$$= \mathcal{D}^{\text{nonlin}}(\hat{f}|_i) + \mathcal{D}^{\text{lin}}\Delta\hat{f}_{i+1}, \tag{73}$$

where the linear differential operators $\mathcal{G}^{\text{lin}}$ and $\mathcal{D}^{\text{lin}}$ were introduced.

2. Construct a MK-representation of the incermental function $\Delta\hat{f}_{i+1}$, that is:

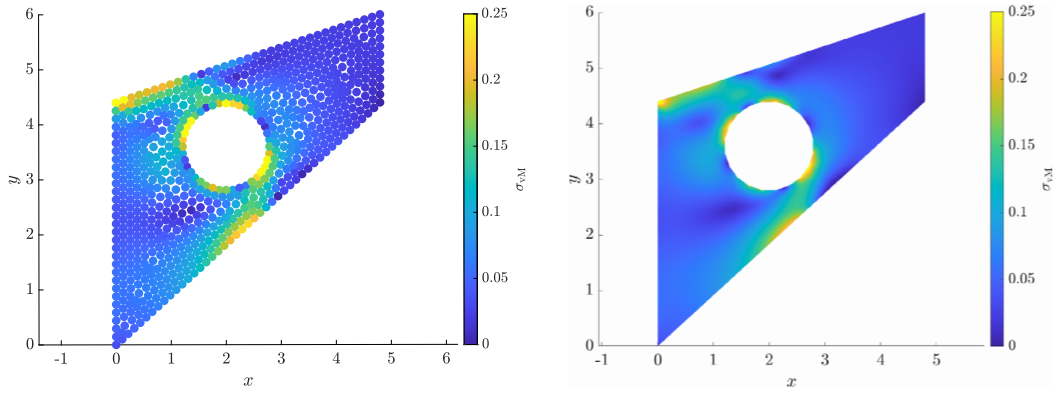


FIGURE 13 Visualization of v. Mises stress field for complicated geometry: (A) MKC approximation. (B) Fine reference solution

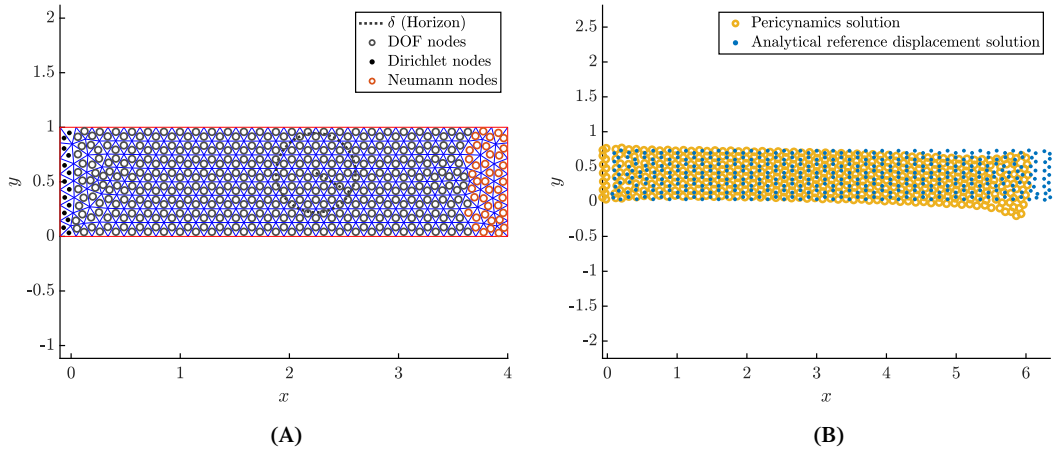


FIGURE 14 Peridynamics patch test configuration: (A) Numerical set-up. (B) Visualization of Peridynamics approximation and fine reference solution

$$\Delta\hat{f}_{i+1}(x) = \tilde{\boldsymbol{A}}(x)\begin{bmatrix} \Delta\boldsymbol{f}^{\mathrm{DOF}}_{i+1} \\ \Delta\boldsymbol{f}^{D}_{i+1} \\ \Delta\boldsymbol{t}_{i+1} \\ \Delta\boldsymbol{b}^{D}_{i+1} \end{bmatrix}. \tag{74}$$

The linear interpolation operator $\tilde{\boldsymbol{A}}(x)$ is to be constructed such that

$$\mathcal{G}^{\mathrm{lin}}\Delta\hat{f}_{i+1} = t(x) - \mathcal{G}^{\mathrm{nonlin}}(\hat{f}|_i) \tag{75}$$

$$\mathcal{D}^{\mathrm{lin}}\Delta\hat{f}_{i+1} = b(x) - \mathcal{D}^{\mathrm{nonlin}}(\hat{f}|_i), \tag{76}$$

is fulfilled for all x that coincide with the discretization nodes corresponding to the respective constraints. This implies that

$$\Delta\boldsymbol{t}_{i+1} = \begin{bmatrix} t(x_3^{(1)})^{\top} - \mathcal{G}^{\mathrm{nonlin}}(\hat{f}|_i(x_3^{(1)})) \\ \vdots \\ t(x_3^{(n_3)})^{\top} - \mathcal{G}^{\mathrm{nonlin}}(\hat{f}|_i(x_3^{(n_3)})) \end{bmatrix} \quad \text{and} \quad \Delta\boldsymbol{b}^{D}_{i+1} = \begin{bmatrix} b(x_4^{(1)})^{\top} - \mathcal{D}^{\mathrm{nonlin}}(\hat{f}|_i(x_4^{(1)})) \\ \vdots \\ b(x_4^{(n_4)})^{\top} - \mathcal{D}^{\mathrm{nonlin}}(\hat{f}|_i(x_4^{(n_4)})) \end{bmatrix}. \tag{77}$$

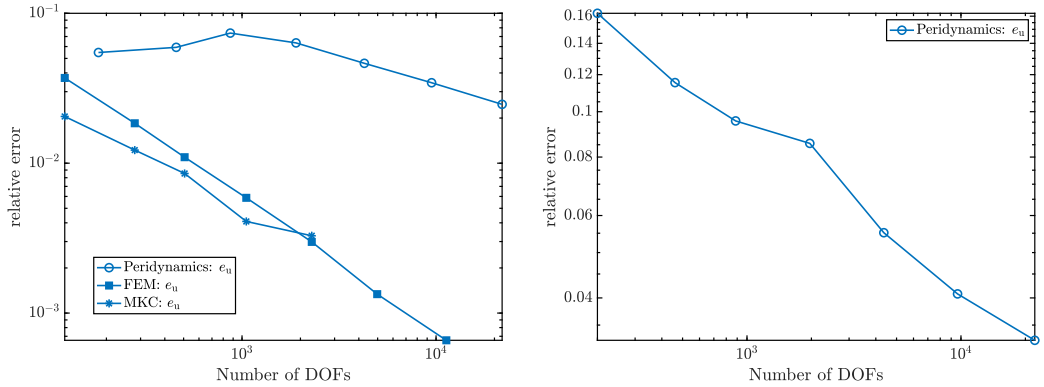


FIGURE 15 Displacement errors over number of DOFs for FEM, MKC, and Peridynamics: (A) Bending test configuration. (b) Patch test configuration

With this, there now is a MK representation of $\Delta\hat{f}_{i+1}$ which satisfies (75) and (76) pointwise for arbitrary values of the incemental discrete DOFs $\Delta\boldsymbol{f}_{i+1}^{\text{DOF}}$.

3. Analogous to the procedure presented in Section 4, solve the linearized and discretized system

$$\mathcal{D}^{\text{lin}}\boldsymbol{A}(x_1^{(a)})\begin{bmatrix}\Delta\boldsymbol{f}_{i+1}^{\text{DOF}}\\ \Delta\boldsymbol{f}_{i+1}^{D}\\ \Delta\boldsymbol{t}_{i+1}\\ \Delta\boldsymbol{b}_{i+1}^{D}\end{bmatrix} \overset{!}{=} b(x^{(a)}) - \mathcal{D}^{\text{nonlin}}(\hat{f}|_i(x_1^{(a)})) \quad \forall\, a \in \{1, \dots, n_1\}, \tag{78}$$

where $\Delta\boldsymbol{f}_{i+1}^{D}$, $\Delta\boldsymbol{t}_{i+1}$, and $\boldsymbol{b}_{i+1}^{D}$ are known quantities.

4. Update the solution: $\hat{f}|_{i+1} = \hat{f}|_i + \Delta\hat{f}_{i+1}$.

Note that it is possible, of course, to embedd the NR method in an incremental loading scheme. As indicated in the beginning of this section, the testing of the described strategy for solving nonlinear problems is still to be done and will form the subject of future investigations.

8 | CONCLUSION

In this article, a MK based, truly mesh free algorithm for the numerical solution of linear PDEs is developed. The proposed method allows for a consistent imposition of BCs by constructing a solution space that comprises solely functions that comply with the boundary constraints at a finite number of locations. In addition to satisfying BCs, functions from the solution space satisfy the PDE at Dirichlet nodes irrespective of the values of the DOFs, thus improving the approximation accuracy in these regions. The entire procedure is strong form based and therefore no integrals need to be evaluated numerically. The global linear system of equations that is obtained as a result of the MK discretization is not symmetric which can be seen as a disadvantage as compared to some other methods such as FEs. Due to the first order completeness which the approximant possesses with an appropriate choice of explicit basis functions, the 2D patch test of linear elasticity is passed. It is demonstrated that the approximation accuracy of MKC improves under discretization refinement, also for analytically intractable cases. The graphs describing the error as a function of the number of DOFs, however, exhibit slight irregularities which are supposedly caused by the irregularities in the nodal arrangements that are produced by our domain discretization algorithm. In future research, the issue of translating the MKC concept to nonlinear and three-dimensional solid mechanics problems will be addressed.

ORCID

Michael W. Gee https://orcid.org/0000-0001-9293-7201

REFERENCES

1. Williams CKI, Rasmussen CE. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press; 2006.
2. Journel AG, Huijbregts CJ. *Mining Geostatistics*. London, UK: Academic Press; 1978.
3. Lam KY, Wang QX, Li H. A novel meshless approach–local kriging (lokriging) method with two-dimensional structural analysis. *Comput Mech*. 2004;33(3):235-244.
4. Thai CH, Nguyen-Xuan H. A moving kriging interpolation meshfree method based on naturally stabilized nodal integration scheme for plate analysis. *Int J Comput Methods*. 2018;16(4):1850100.
5. Gu YT, Wang QX, Lam KY. A meshless local kriging method for large deformation analyses. *Comput Methods Appl Mech Eng*. 2007;196(9-12):1673-1684.
6. Gu L. Moving kriging interpolation and element-free Galerkin method. *Int J Numer Methods Eng*. 2003;56(1):1-11.
7. Dehghan M, Abbaszadeh M. Two meshless procedures: moving kriging interpolation and element-free Galerkin for fractional PDEs. *Appl Anal*. 2017;96(6):936-969.
8. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Mon Notices Royal Astron Soc*. 1977;181(3):375-389.
9. Monaghan JJ. Smoothed particle hydrodynamics. *Rep Progr Phys*. 2005;68(8):1703-1759.
10. Silling SA, Lehoucq RB. *Peridynamic Theory of Solid Mechanics*. Amsterdam, Netherlands: Elsevier; 2010:73-168.
11. Silling SA, Epton M, Weckner O, Xu J, Askari E. Peridynamic states and constitutive modeling. *J Elasticity*. 2007;88(2):151-184.
12. Ganzenmüller GC, Hiermaier S, May M. On the similarity of meshless discretizations of peridynamics and smooth-particle hydrodynamics; 2014. arXiv:1401.8268 [physics].
13. Huerta A, Belytschko T, Fernández-Méndez S, Rabczuk T, Zhuang X, Arroyo M. Meshfree methods. *Encyclopedia of Computational Mechanics*. Vol 2017. 2nd ed. Hoboken, NJ: John Wiley; 2018:1-38.
14. Ganzenmüller GC. An hourglass control algorithm for Lagrangian smooth particle hydrodynamics. *Comput Methods Appl Mech Eng*. 2015;286:87-106.
15. Bui TQ, Nguyen TN, Nguyen-Dang H. A moving kriging interpolation-based meshless method for numerical simulation of kirchhoff plate problems. *Int J Numer Methods Eng*. 2009;77(10):1371-1395. https://doi.org/10.1002/nme.2462.
16. Bui TQ, Zhang C. Moving kriging interpolation-based meshfree method for dynamic analysis of structures: meshfree dynamic of structures. *PAMM*. 2011;11(1):197-198. https://doi.org/10.1002/pamm.201110090.
17. Fasshauer GE, Ye Q. *A Kernel-Based Collocation Method for Elliptic Partial Differential Equations with Random Coefficients*. New York, NY: Springer; 2013:331-347.
18. Cockayne J, Oates C, Sullivan T, Girolami M. *Probabilistic Meshless Methods for Partial Differential Equations and Bayesian Inverse Problems*. Berlin, Germany: Zuse Institute; 2016:16-31.
19. Raissi M, Karniadakis GE. Machine learning of linear differential equations using Gaussian processes. *J Comput Phys*. 2017;348:683-693.
20. Bilionis I. Probabilistic solvers for partial differential equations; 2016. arXiv:1607.03526 [math].
21. Matheron G. The intrinsic random functions and their applications. *Adv Appl Probab*. 1973;5(3):439-468.
22. Blight BJN, Ott L. A Bayesian approach to model inadequacy for polynomial regression. *Biometrika*. 1975;62(1):79-88.
23. O'Hagan A, Kingman JFC. Curve fitting and optimal design for prediction. *J Royal Stat Soc Ser B (Methodol)*. 1978;40(1):1-42.
24. Papoulis A, Unnikrishna-Pillai S. *Probability, Random Variables, and Stochastic Processes*. New York, NY: Tata McGraw-Hill Education; 2002.
25. Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. *Math Comput*. 1981;37(155):141-158.
26. Bonet J, Wood RD. *Nonlinear Continuum Mechanics for Finite Element Analysis*. 2nd ed. Cambridge, MA: Cambridge University Press; 1997.

How to cite this article: Bott FS, Gee MW. A strong form based moving Kriging collocation method for the numerical solution of partial differential equations with mixed boundary conditions. *Int J Numer Methods Eng*. 2021;122:657–683. https://doi.org/10.1002/nme.6553