Local explicit interval fields for non-stationary uncertainty modelling in finite element models

Robin R.P. Callens*, Matthias G.R. Faes, David Moens

KU Leuven, Departement of Mechanical Engineering, Division LMSD, Campus De Nayer, Jan De Nayerlaan 5, St.-Katelijne-Waver 2860, Belgium

Abstract

Interval fields have been introduced to model spatial uncertainty in Finite Element Models when the stochastic resolution of available data is too limited to build representative probabilistic models. However, current interval fields modelling techniques are according to the state-of-the-art limited in potential, as they homogenise the uncertain parameters to globally defined parameters. Hence, these techniques are inherently unable to represent local uncertainty. In practice, local variations in the uncertain parameters (non-stationary uncertainty) often occur, e.g., through local effects in manufacturing processes. This paper presents a novel method to model local explicit interval fields, that is furthermore less computationally demanding and less conservative than global explicit interval fields. The method presented in this paper is based on the concept of explicit interval fields and develops an alternative approach for the commonly applied inverse distance weighting approach for the generation of the basis functions. The paper includes three case studies to compare the introduced local explicit interval

*Corresponding author

Email address: robin.callens@kuleuven.be (Robin R.P. Callens)

fields approach with the global explicit interval fields method. The obtained results are discussed from a numerical and application point of view to show the effectiveness and efficiency of the proposed methods.
Keywords: Interval analysis, Local explicit interval fields, Spatial uncertainty

1. Introduction

Interval analysis is becoming popular when limited or incomplete data are available of the true model parameter values,e.g., in civil engineering [1], in structural engineering [2, 3], in fatigue analysis [4]. Compared to probabilistic techniques, which require distributions of the uncertain parameters, intervals quantify the uncertainty on the actual parameter value by an upper and a lower bound [5]. Also imprecise probability models have been developed, such as probability-box (p-box) [6], which can be regarded as the combination of non-probabilistic and probabilistic models [7]. Interval analysis quantify the best and worst-case behaviour of the structure. By definition intervals are independent, and hence, the joint description of several interval-valued parameters is given by a hyper-rectangle [8]. Standard interval Finite Element (FE) analysis models spatial uncertainty by defining independent intervals on locations in the FE-model (e.g., element centres, element nodes, Gauss points) [9]. This leads to a high dimensional input space, which is not attractive from an application perspective, since it results in a high computational cost for propagation. In literature several techniques are proposed to reduce this computational cost,e.g., with Cauchy distributions [10, 11]. Furthermore, these independent intervals ne-

glect all correlation present in the physical quantities under consideration, which generally leads to over-conservative results of the interval analysis. Local and global variations (=non-stationary (NS) uncertainty) are often encountered in parts or components, which originates from their manufacturing process. On the one hand, local NS uncertainty is for instance encountered in manufactured components such as, e.g., deep-drawing where local defects as micro-cracks and voids introduce local uncertainty in the part's material properties. Manufacturing processes on the other hand that introduce global NS uncertainty are, for instance, casting where variations in cooling rates generate global variations. To obtain less over-conservative results, Moens et al. [12] proposed explicit interval fields to represent spatial uncertainty in FE analyses. This approach models spatial dependence using a series expansion of basis functions, weighted with interval-valued scalars. This results in a significant advantage compared to standard interval FE, since the input space dimension is reduced from the number of all locations in the FE-model to a few interval scalars. Recently, these interval fields were used for several applications such as, e.g., modelling of dynamic phenomena [13] and the effect of manufacturing-related uncertainty on the mechanical quality of additive manufactured plastic parts [14]. Faes et al. [15] introduced for the explicit interval field expansion the application of Inverse Distance Weighting (IDW) to construct basis functions. However, the IDW basis functions are by definition only applicable for modelling global NS uncertainty, and as such, this method is a global explicit interval field. This approach was recently extended to scaled IDW which allows more control on the global non-stationary interval fields [16]. Another approach to construct interval

field basis functions is to limit the gradients in the interval field realisations in combination with radial basis function interpolation [17]. However, this approach keeps a high number of independent intervals, as it defines them on locations in the FE-model (element nodes, element centres, Gauss integration points). Sofi et al. [18] and Sofi and Muscolino [19] presented an alternative approach for explicit interval field to limit the over-conservative results of standard interval FE, by applying an affine arithmetic representation of the interval uncertainty in combination with the Karhunen-Loeve expansion. Also, other expansions can be used as an Expansion Optimal Linear Estimator [20]. This approach is applied on several cases, e.g., structural dynamics [21, 22], electromagnetic [23] and a Euler-Bernoulli beam [24]. However, this approach as presented is only limited to stationary uncertainty modelling, whereas the global explicit interval fields can also model global non-stationary (NS) uncertainty. Furthermore, other interval field modelling techniques were also formulated by other scholars, such as e.g., an interval field for static plane stress analysis [25] and an interval field for spatial-time varying uncertainty [26, 27]. In engineering applications were only data are available on local NS uncertainty, e.g., for welding this could be data in the region of the heat effected zone or in case of casting this could be data in the region of local cooling effects. In this case with only data of local NS uncertainty, all presented interval field techniques are limited in their ability to model this local NS uncertainty without imposing a high number of additional interval scalars. This, in its turn significantly increases the computational cost of the propagation. Another option is that they homogenise the local NS uncertainty over the full model domain which in turn increases

the over-conservatism of the results.

The goal of this paper is to define an interval field modelling technique that models local NS uncertainty with a small number of control points and furthermore is capable of accurately representing local NS uncertainty. This is specifically obtained using the concept of explicit interval fields [12] and IDW to build basis functions as applied by Faes et al. [15]. In essence, this is achieved by applying IDW in combination with a local weighting function (e.g., a weight function of meshless methods [28]) defined on a local domain around its control point, that is (1) monotonically decreasing from the control point towards the edge of the local domain, (2) continuous on the edge of the local domain, (3) only non-negative in the local domain. The result is a local explicit interval field modelling technique that models local NS uncertainty with a limited number of control points and reduces the over-conservatism as compared to other more global explicit interval fields modelling techniques. The structure of the paper is as follows: section 2 presents interval field FE analysis, explicit interval fields that model the interval field with basis functions and a limited set of interval scalars and it describes the method of IDW. Section 3 proposes a method for modelling local explicit interval fields. The next sections 4, 5 and 6 handle case studies, respectively illustrating the modelling of one- two- and three-dimensional local explicit interval fields including their computational cost in comparison with the global explicit interval fields. At the end of this paper, section 7 summarises the conclusions.

Definitions:

x	Lower-case are real variables
$\boldsymbol{x}$	Boldface lower-case are vectors
$\boldsymbol{x}_i$	Entry of a vector
$\boldsymbol{X}$	Boldface upper-case are matrices
$\boldsymbol{X}_i$	Entry of a matrix
x^I	Lower-case with superscript I are interval scalars
$\boldsymbol{x}^I$	Boldface lower-case with superscript I are interval vectors
$\boldsymbol{y}^S$	Solution set of output vector $\boldsymbol{y}$
$\underline{x}$,$\overline{x}$	Respectively the lower and upper interval bound
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of positive real numbers
$\mathbb{R}^n$	Set of real numbers of dimension n
$\mathbb{IR}$	Set of real interval numbers
$\mathbb{IR}^n$	Set of real interval numbers of dimension n
$\mathbb{N}$	Set of natural numbers
$\mathbb{K}$	Set that is compact
$\mathcal{M}(\boldsymbol{x})$	Numerical model
$\mathcal{X}$, $\mathcal{Y}$	Admissible sets of input and output parameters
Ω	Domain of the finite element model
Ω_e or $\mathcal{K}_e$	Domain of one element in Ω or $\mathcal{K}$
Ω_i or $\mathcal{K}$	Domain of support of a weight function Ω_i
$d_{..}$	Dimension with subscript denoting the set or vector
${}^{IDW}..$	Denoting a IDW variable, vector or matrix
$..^{\mathcal{K}}$	Denoting a variable, vector or matrix in $\mathcal{K}$

2. Interval Field Finite Element analysis

In current engineering practice, the application of FE analysis is popular to approximate the solution of a partial differential equation (PDE). In this work, we consider FE models that are parametric, i.e., represented by a numerical model $\mathcal{M}(\boldsymbol{x})$ and parametrised by a vector $\boldsymbol{x}(\boldsymbol{r}) \in \mathcal{X} \subsetneq \mathbb{R}^{d_\mathcal{X}}$ with $\mathcal{X}$ the admissible set of input parameters and $d_\mathcal{X} \in \mathbb{N}$ the number of input parameters. For instance, $\boldsymbol{x}(\boldsymbol{r})$ contains material parameters as a function of the spatial coordinate $\boldsymbol{r} \in \Omega \subsetneq \mathbb{R}^{d_\Omega}$ with $d_\Omega \in \mathbb{N}, d_\Omega \leq 3$ the number of dimensions (max. 3 Cartesian dimensions). Consider a discretised FE-model domain Ω with N_e elements $\Omega_e \subset \Omega$ that contains d_d degrees of freedom (DOF) and as such approximates the PDE with the solution of a set of d_d linear equations $m_i, i = 1, \ldots, d_d$. The model $\mathcal{M}(\boldsymbol{x})$ provides a vector of model responses $\boldsymbol{y}(\boldsymbol{r}) \in \mathcal{Y} \subsetneq \mathbb{R}^{d_d}$ with $\mathcal{Y}$ the admissible set of output parameters and $d_d \in \mathbb{N}$ the number of output parameters. $\mathcal{M}(\boldsymbol{x})$ is defined as:

$$\mathcal{M}(\boldsymbol{x}) : y_i(\boldsymbol{r}) = m_i(\boldsymbol{x}(\boldsymbol{r})) \tag{1}$$

with $m_i : \mathbb{R}^{d_\mathcal{X}} \mapsto \mathbb{R}$ a function that maps the parametric input space to the individual outputs $y_i(\boldsymbol{r})$ of the FE analysis. Note that the output $\boldsymbol{y}(\boldsymbol{r})$ is generated on element or nodal level $\boldsymbol{r}$ (e.g., displacement, strains). When considering outputs on a global modal level (e.g., eigenfrequencies), the outputs $y_i(\boldsymbol{r})$ are not dependent on $\boldsymbol{r}$.

2.1. Interval Finite Elements

Consider an interval scalar $x^I \in \mathbb{IR}$, where $\mathbb{IR}$ is the domain of closed real-valued intervals. An interval scalar x^I is defined as:

$$x^I = [x_{min}\ x_{max}] = [\underline{x}\ \overline{x}] = \{x \in \mathbb{R} | \underline{x} \leq x \leq \overline{x}\}, \tag{2}$$

with x_{min} and x_{max} bounds of the uncertain parameter x and the midpoint of an interval defined as:

$$x_\mu = \frac{\underline{x} + \overline{x}}{2}. \tag{3}$$

An interval vector $\boldsymbol{x}^I \in \mathbb{IR}^{d_\mathcal{X}}$ contains $d_\mathcal{X}$ interval scalars $x_i^I,\ i = 1, \ldots, d_\mathcal{X}$ which are by definition independent. In general, the interval FE analysis [29] has an interval vector $\boldsymbol{x}^I \in \mathbb{IR}^{d_\mathcal{X}}$ as input, that is defined as:

$$\boldsymbol{x}^I = \left\{ x_1^I \quad x_2^I \quad \ldots \quad x_{d_\mathcal{X}}^I \right\}^T = \left\{ \boldsymbol{x} \in \mathbb{R}^{d_\mathcal{X}} | x_i \in x_i^I \right\}, \tag{4}$$

and the solution set $\boldsymbol{y}^S \in \mathbb{R}^{d_d}$ as output, which is defined as:

$$\boldsymbol{y}^S = \left\{ \boldsymbol{y} | \boldsymbol{y} = \mathcal{M}(\boldsymbol{x}), \boldsymbol{x} \in \boldsymbol{x}^I \right\}, \tag{5}$$

with $\mathcal{M}(\boldsymbol{x})$ containing the deterministic functions $m_1, \ldots, m_{d_d}$ mapping the input to the output space of the FE analysis and $\boldsymbol{y} \in \mathbb{R}^{d_d}$. The solution set is usually approximated by an interval $\boldsymbol{y}^I \in \mathbb{IR}^{d_d}$: $\boldsymbol{y}^I = \left\{ y_1^I \quad y_2^I \quad \ldots \quad y_{d_d}^I \right\}^T$. The components $y_i^I = \left[\underline{y}_i\ \overline{y}_i \right]$ of $\boldsymbol{y}^I$ are determined by means of (anti-)optimisation:

$$\underline{y}_i = \min_{\boldsymbol{x} \in \boldsymbol{x}^I} m_i(\boldsymbol{x}), \tag{6}$$

$$\overline{y}_i = \max_{\boldsymbol{x} \in \boldsymbol{x}^I} m_i(\boldsymbol{x}). \tag{7}$$

With this optimisation the interval of each component y_i^I of $\boldsymbol{y}^I$ is found independently, resulting in an approximation of the solution set $\boldsymbol{y}^S$ as a hyper-rectangle. Note that following this approach, a total of $2 * d_{\mathcal{X}}$ optimization problems have to be solved.

2.2. Explicit interval fields for finite elements

The interval FE method neglects all correlation present in physical quantities under consideration, which leads to over-conservative results. Also from an application perspective, when propagating the interval field through $\mathcal{M}(\boldsymbol{x})$, the number of required deterministic model evaluations is important, especially when using industrially sized FE models (containing up to millions of DOF). In the case of linear monotonic interval analysis, the number of deterministic model evaluations scale with $2^{d_{\mathcal{X}}}$. It is clear that when models containing many input parameters are considered, the computational cost will scale quickly to being intractable.

To reduce those shortcomings, explicit interval fields were introduced by Moens et al. [12]. This approach uses an interval vector $\boldsymbol{\alpha}^I \in \mathbb{IR}^{n_b}$ containing $n_b \in \mathbb{N}$ independent interval scalars, such that the input space is reduced from $\mathbb{IR}^{d_{\mathcal{X}}}$ to $\mathbb{IR}^{n_b}, d_{\mathcal{X}} \geq n_b$. The interval fields approach models the spatial dependency using basis functions $\boldsymbol{\psi}_i(\boldsymbol{r}) : \Omega \mapsto \mathbb{R}^{k_{FE}}$ with $i = 1, 2, \ldots, n_b$ and k_{FE} the total number of interval field discretisation points in Ω. An explicit interval field $\boldsymbol{x}^I(\boldsymbol{r}) : \Omega \times \mathbb{IR}^{n_b} \mapsto \mathbb{IR}^{k_{FE}}$ is then defined as a series expansion of n_b basis functions, multiplied with interval scalars:

$$\boldsymbol{x}^I(\boldsymbol{r}) = \sum_{i=1}^{n_b} \boldsymbol{\psi}_i(\boldsymbol{r}) \cdot \alpha_i^I. \tag{8}$$

Note that the maximal number of interval scalars n_b that are possible with explicit interval fields and all other technique discussed in this paper depends on the available computational resources, the computational cost of the numerical model $\mathcal{M}$, and the used propagation scheme (e.g., (anti-)optimisation, vertex analysis).

2.3. Global explicit interval fields with Inverse Distance Weighting

This work uses the global explicit interval fields expansion in combination with IDW, presented by Faes et al. [30] as a starting point. The IDW basis functions are generated to describe the spatial dependence between interval scalars that are defined on control points $\boldsymbol{r}_{n_i}$ in the domain Ω, using weight functions ${}^{IDW}\boldsymbol{w}_i(\boldsymbol{r})$ which are computed as:

$$\boldsymbol{\psi}_i(\boldsymbol{r}) = \frac{{}^{IDW}\boldsymbol{w}_i(\boldsymbol{r})}{\sum_{j=1}^{n_b} {}^{IDW}w_j(\boldsymbol{r})}, \tag{9}$$

with $\boldsymbol{r} \in \Omega$ and ${}^{IDW}\boldsymbol{w}_i(\boldsymbol{r}) : \Omega \mapsto \mathbb{R}^{k_{FE}}$ the weight functions defined as:

$${}^{IDW}\boldsymbol{w}_i(\boldsymbol{r}) = \frac{1}{[\boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i})]^p}, \tag{10}$$

where $\boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i})$ is the Euclidean distance between the locations $\boldsymbol{r} \in \Omega$ and locations of the control points $\boldsymbol{r}_{n_i} \in \Omega$. The power $p \in \mathbb{R}^+$ controls the gradient and the continuity of the weight functions. From equation 10 it is clear that the calculation of a set of distance measures over the interval field discretisation points is necessary to obtain the weight functions. The global explicit interval field is discretised over the full FE-model, resulting in a total number of distance measures $O(k_{FE} \times n_b)$, where k_{FE} are the global explicit interval field discretisation points and n_b are the control points. For

small FE models (DOF $< 10K$) the computational cost corresponding to the distance measures is acceptable. However, when using large industrial sized models (DOF>1M), this results in a non-negligible computational burden in terms of CPU and RAM usage. A potential solution hereto is to define the weight functions and corresponding global explicit interval field on a sparse grid and interpolate between the sparse grid and the FE model, as introduced by the authors in [31]. However, the adaptive fitting of an optimal sparse grid representation also causes computational overhead.

The global explicit interval field also does not allow for modelling local NS uncertainty, this follows from equation 9 and 10. Based on these definitions, the inverse distance basis functions are strictly positive on all locations of the model that do not coincide with control points, and they are unity at the defining control point location $\boldsymbol{r}_{n_i}$ and zero at all other control points. The global explicit interval field with IDW achieves global dependence through the weighting functions and the following properties:

1. $\dfrac{\partial\, {}^{IDW}\boldsymbol{w}_i}{\partial \boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i})} \leq 0,$

2. $\begin{cases} {}^{IDW}\boldsymbol{w}_i(\boldsymbol{r}) \geq 0 & \text{if} \quad \boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i}) \neq \infty, \\ {}^{IDW}\boldsymbol{w}_i(\boldsymbol{r}) = 0 & \text{if} \quad \boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i}) = \infty. \end{cases}$

for the element size $h_e \to 0$. As a result, this weighting technique with global dependence is not capable of describing local dependence, as, through the second property, all weight functions will contribute to non-negative uncertainty levels throughout the entire spatial domain.

3. Local explicit interval field for local non-stationary uncertainty

3.1. Local weight functions

This paper models local NS uncertainty using the explicit interval field method and the IDW technique. Hereto, the use of other weight functions is required. Such local weighting functions need to satisfy the following properties:

$$1. \quad \frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i})} \leq 0, \tag{11}$$

$$2. \quad \frac{\partial \boldsymbol{w}_i}{\partial \boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i})} = 0 \quad \text{if} \quad \boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i}) = \boldsymbol{B}_i, \tag{12}$$

$$3. \quad \begin{cases} \boldsymbol{w}_i(\boldsymbol{r}) \geq 0 \quad \text{if} \quad \boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i}) \neq \boldsymbol{B}_i, \\ \boldsymbol{w}_i(\boldsymbol{r}) = 0 \quad \text{if} \quad \boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i}) = \boldsymbol{B}_i, \end{cases} \tag{13}$$

for $h_e \to 0$ and with $\boldsymbol{B}_i$ the width of the support zone $\Omega_i \subseteq \Omega$ in each dimension $d_\Omega \leq 3$ around one control point r_{n_i}. The first property ensures that the weighting function is monotonically decreasing. The second property is such that the resulting interval field is continuous on the edge of the support zone Ω_i. Property three introduces the local description of the weight function, specifying that the weight function is zero starting from a distance $\boldsymbol{B}_i$ around the control point n_i. A possible weight function that satisfies these requirements is the quartic spline function, that is one of the possible weight functions in the meshless methods [28]. The support zone $\Omega_i \subseteq \Omega$ where $\boldsymbol{d}(\boldsymbol{r}, \boldsymbol{r}_{n_i}) < \boldsymbol{B}_i \to \boldsymbol{w}_i(\boldsymbol{r}) \neq 0$ is compact $\mathbb{K}$ as it is closed and bounded by the width $\boldsymbol{B}_i$. This reduces the calculation of the distance measures from the full domain Ω that is required for the global explicit interval field to the domain Ω_i, yielding a drastic gain in computational efficiency, namely

$O\left(\sum_{i=1}^{n_b} k_i\right)$ by reducing k_{FE} to a limited set of points k_i for each control point n_i. Further in this paper, the notation of the support zone Ω_i of one control point n_i is changed to $\mathcal{K}$, as this shortens the mathematical notations and equations. Weight functions are then $\boldsymbol{w}^{\mathcal{K}}(\boldsymbol{r}) : \mathcal{K} \mapsto \mathbb{R}^{k_i}$ and the basis functions are $\boldsymbol{\psi}^{\mathcal{K}}(\boldsymbol{r}) : \mathcal{K} \mapsto \mathbb{R}^{k_i}$. Paragraph 3.2 describes the construction of basis functions for one-dimensional domains ($d_\Omega = 1$) and paragraph 3.4 for higher dimensional domains ($d_\Omega \leq 3$).

3.2. Local basis functions for one-dimensional domains

This section extends equation 9 to build a local basis function $\boldsymbol{\psi}^{\mathcal{K}}(\boldsymbol{r}) : \mathcal{K} \mapsto \mathbb{R}^{k_i}$ from weight functions in 1D. Equation 9 normalises the weight functions to the weight functions of other control points. However, when doing so in this context, the equation normalises the single local weight function inside the domain $\mathcal{K}$ to itself, yielding a constant basis function $\boldsymbol{\psi}^{\mathcal{K}}(\boldsymbol{r})$ equal to one inside $\mathcal{K}$. This results in a discontinuous basis function: $\boldsymbol{\psi}^{\mathcal{K}}(\boldsymbol{r}) = 1$ for $\boldsymbol{r} \in \mathcal{K}$ and 0 otherwise. To overcome this limitation, it is proposed to add two virtual nodes $n^{\mathcal{K}}$ on the edges of each spatial dimension of the domain $\mathcal{K}$ to allow for defining a continuous basis function. These virtual nodes are based on the meshless method, where also only nodes with a region of support are defined. The virtual nodes are then the meshless nodes inside $\mathcal{K}$. For one-dimensional domains, the locations of the virtual nodes in $\mathcal{K}$ are:

1. the lower bound edge of $\mathcal{K}$: $\boldsymbol{r}^{\mathcal{K}}_{n=1} = \boldsymbol{r}_{n_i} - B_i$,
2. the midpoint of $\mathcal{K}$: $\boldsymbol{r}^{\mathcal{K}}_{n=2} = \boldsymbol{r}_{n_i}$,
3. the upper bound edge of $\mathcal{K}$: $\boldsymbol{r}^{\mathcal{K}}_{n=3} = \boldsymbol{r}_{n_i} + B_i$,

with the first and last virtual node on the edges of $\mathcal{K}$ and $\boldsymbol{r}_{n_i} \in \mathcal{K}$ the location

of the control point n_i. On each of these nodes, weighting functions $\boldsymbol{w}_i^{\mathcal{K}}(\boldsymbol{r})$ are defined, with $i = 1, 2, 3$ corresponding to each virtual node for 1D. Figure 1 visualises this and illustrates the construction of one basis function inside $\mathcal{K}$ from the weight functions (in this figure, quartic splines are used, see section 4.2). The result is a basis function which is continuous in $\mathcal{K}$ and has local spatial dependence.

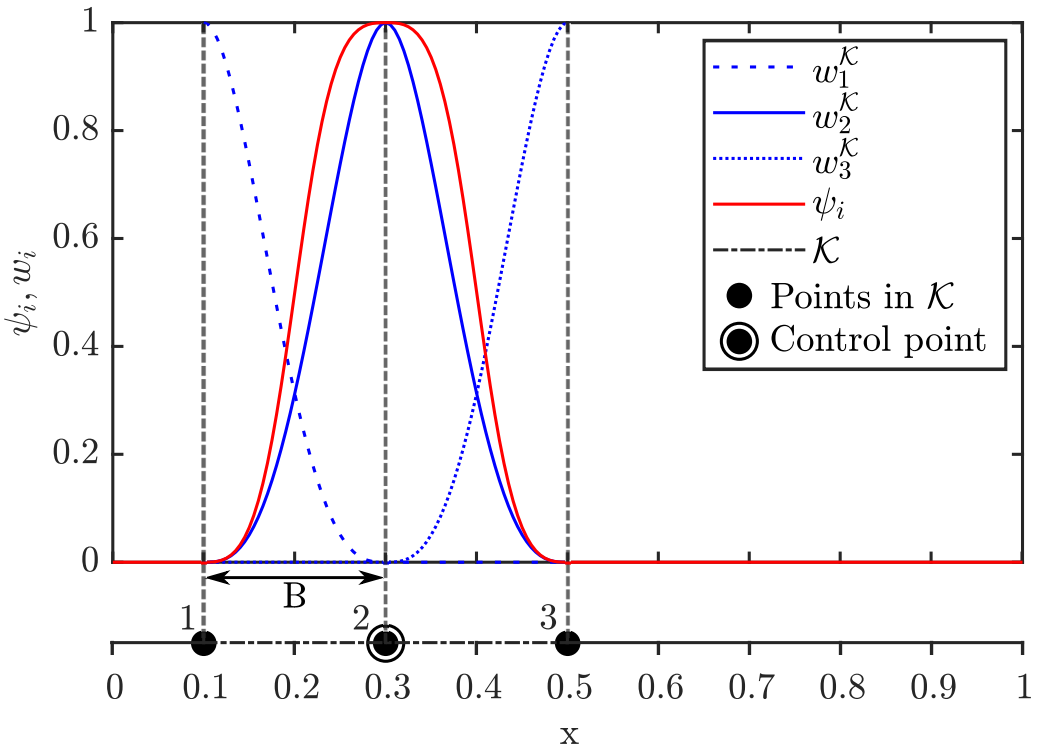


Figure 1: For an 1D domain, the location of the virtual nodes (small black dots) and control point (big black dot). The blue and red lines are respectively weight functions (quartic spline) and basis functions. B describes the width of the domain $\mathcal{K}$ around the control point.

With those virtual nodes and control point in $\mathcal{K}$, one could obtain the

basis functions with equation 9. However, this results in three basis functions, one for each virtual node, instead of one basis function for the control point n_i. Therefore, equation 9 is multiplied with a parameter $\boldsymbol{a}^{\mathcal{K}}$ to ensure that only the basis function of the control point n_i is non-negative. The basis functions are then calculated with:

$$\boldsymbol{\psi}^{\mathcal{K}}(\boldsymbol{r}^{\mathcal{K}}) = \boldsymbol{a}^{\mathcal{K}} \frac{\boldsymbol{w}_i^{\mathcal{K}}(\boldsymbol{r}^{\mathcal{K}})}{\sum_{j=1}^{3} w_j^{\mathcal{K}}(\boldsymbol{r}^{\mathcal{K}})}, \tag{14}$$

with $\boldsymbol{r}^{\mathcal{K}} \in \mathcal{K}$ and $\boldsymbol{a}^{\mathcal{K}}$ such that a single basis function that is non-negative is obtained as:

$$\boldsymbol{a}^{\mathcal{K}} = \begin{cases} 1 \text{ if } \boldsymbol{r}_n^{\mathcal{K}} = \boldsymbol{r}_{n_i} \\ 0 \text{ if } \boldsymbol{r}_n^{\mathcal{K}} \neq \boldsymbol{r}_{n_i}, \end{cases} \tag{15}$$

with $\boldsymbol{r}_n^{\mathcal{K}}$ the locations of the virtual nodes and control point in $\mathcal{K}$ and $\boldsymbol{r}_{n_i}$ the location of the control point n_i in Ω. The basis function $\boldsymbol{\psi}^{\mathcal{K}}(\boldsymbol{r}^{\mathcal{K}}) \in \mathcal{K}$ is then mapped from the domain $\mathcal{K}$ to Ω by the locations of, e.g., element nodes, element centres or Gauss points, and is explicitly given as:

$$\boldsymbol{\psi}_i(\boldsymbol{r}) = \begin{cases} \boldsymbol{\psi}^{\mathcal{K}}(\boldsymbol{r}^{\mathcal{K}}) & \text{if } \mathcal{K}_e = \Omega_e \\ 0 & \text{if } \mathcal{K}_e \neq \Omega_e, \end{cases} \tag{16}$$

This result in strict local dependence, as the basis function $\boldsymbol{\psi}_i(\boldsymbol{r}) \in \Omega$ is zero outside $\mathcal{K}$. The local explicit interval field can then finally be formulated as:

$$\boldsymbol{x}^{\boldsymbol{I}}(\boldsymbol{r}) = \boldsymbol{x}_\mu + \sum_{i=1}^{n_b} \boldsymbol{\psi}_i(\boldsymbol{r}) \cdot (\boldsymbol{\alpha}_i^I - \boldsymbol{x}_\mu), \tag{17}$$

where $\boldsymbol{x}_\mu$ is the midpoint of the field.

3.3. Local basis functions for two-dimensional domains

For problems which have $d_\Omega = 2$, the construction of a multidimensional weight function is necessary, which depends on the node structure in $\mathcal{K}$. In this paper, it is proposed to use a rectangular structure, since from a computational perspective a rectangular shape is the most efficient way. This is because a Cartesian coordinate system commonly defines the locations in FE models. Figure 2 visualises the node structure for $d_\Omega = 2$. This domain is only valid if the uncertainty under consideration is concentrated inside a rectangle that is characterised with a size of B for each dimension. Otherwise, it is necessary to define different structures, e.g., circles or ellipses. These different structures require a coordinate transformation of the FE-model to other coordinate systems this, however, falls outside the scope of this paper. The rectangular node structure in $\mathbb{K}$ allows to define a one-dimensional problem for each dimension of d_Ω. In the one-dimensional problem, there are 3 virtual nodes (incl. control point). This extends to higher-dimensional FE-models, by defining 3 virtual nodes (excl. control point) in each dimension. Where the midpoint virtual node in each domain is correspondingly on the x or y coordinate of the control point. For this two-dimensional domain a total of six virtual nodes are obtained. Next, the weight functions are calculated on all those virtual nodes in each dimension of $\mathcal{K}$ with a specific weight function that satisfies the three requirements (equations 11,12,13) as introduced in 3. This result for $d_\Omega = 2$ in 6 weight functions, 3 for each dimension: $\boldsymbol{w}_{x1}^{\mathcal{K}}$, $\boldsymbol{w}_{x2}^{\mathcal{K}}$, $\boldsymbol{w}_{x3}^{\mathcal{K}}$, $\boldsymbol{w}_{y1}^{\mathcal{K}}$, $\boldsymbol{w}_{y2}^{\mathcal{K}}$, $\boldsymbol{w}_{y3}^{\mathcal{K}}$. Combining these weight functions into weight matrices

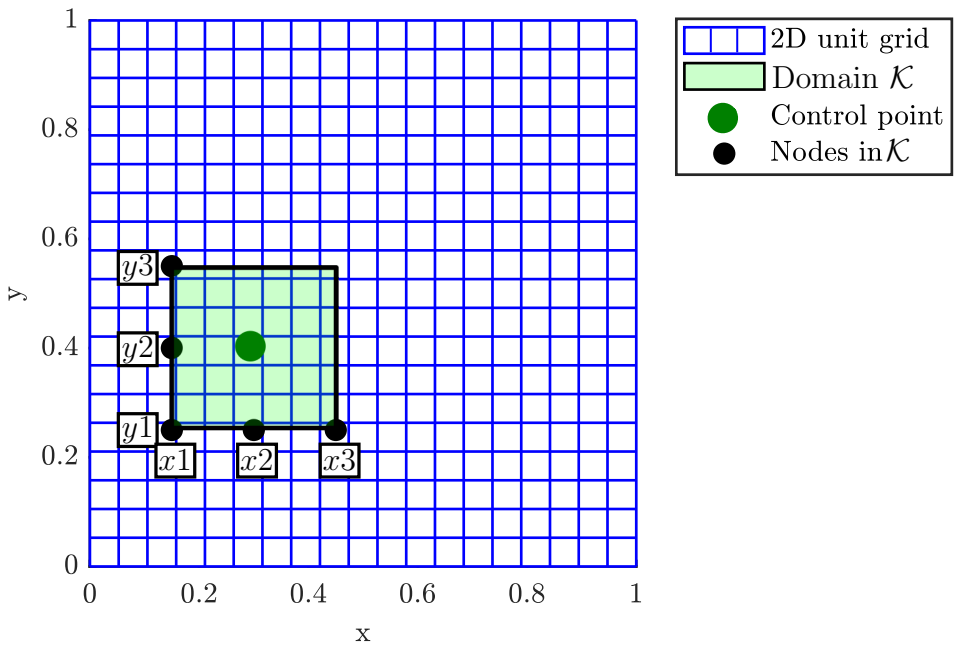


Figure 2: 2D unit grid of 20×20 in blue, where one control point in green is defined on the midpoint of domain $\mathcal{K}$: coordinate $(0.3; 0.4)$. The edges of the domain $\mathcal{K}$ show the width of the domain in black with dimensions: $B_x, B_y = 0.30$. The virtual nodes in $\mathcal{K}$ are the smaller black nodes.

for each dimension yields:

$$\boldsymbol{W}_{x}^{\mathcal{K}} = \begin{bmatrix} \boldsymbol{w}_{x1}^{\mathcal{K}} & \boldsymbol{w}_{x2}^{\mathcal{K}} & \boldsymbol{w}_{x3}^{\mathcal{K}} \end{bmatrix}, \tag{18}$$

$$\boldsymbol{W}_{y}^{\mathcal{K}} = \begin{bmatrix} \boldsymbol{w}_{y1}^{\mathcal{K}} & \boldsymbol{w}_{y2}^{\mathcal{K}} & \boldsymbol{w}_{y3}^{\mathcal{K}} \end{bmatrix}. \tag{19}$$

The two-dimensional weight functions are then obtained with the Kronecker product of these weight matrices as:

$$\begin{aligned} \text{For } d_{\mathcal{K}} = 2 : \boldsymbol{W}_{xy}^{\mathcal{K}} &= \boldsymbol{W}_{x}^{\mathcal{K}} \otimes \boldsymbol{W}_{y}^{\mathcal{K}} && (20) \\ &= \begin{bmatrix} \boldsymbol{w}_{x1}^{\mathcal{K}} \cdot \boldsymbol{W}_{y}^{\mathcal{K}} & \boldsymbol{w}_{x2}^{\mathcal{K}} \cdot \boldsymbol{W}_{y}^{\mathcal{K}} & \boldsymbol{w}_{x3}^{\mathcal{K}} \cdot \boldsymbol{W}_{y}^{\mathcal{K}} \end{bmatrix} && (21) \\ &= \begin{bmatrix} \boldsymbol{w}_{x1y1}^{\mathcal{K}} & \boldsymbol{w}_{x1y2}^{\mathcal{K}} & \boldsymbol{w}_{x1y3}^{\mathcal{K}} & \boldsymbol{w}_{x2y1}^{\mathcal{K}} & \cdots & \boldsymbol{w}_{x3y3}^{\mathcal{K}} \end{bmatrix}, && (22) \end{aligned}$$

where the discretised weight functions are columns of the weight matrix $\boldsymbol{W}_{xy}^{\mathcal{K}}$. Equation 14 then calculates the basis functions $\boldsymbol{\psi}_i^{\mathcal{K}}$ from these weight functions. Figures 3 and 4 visualise the weight and basis functions for $d_{\mathcal{K}} = 2$. This shows that local dependence is obtained as the basis function is only non-negative inside $\mathcal{K}$. The basis functions $\boldsymbol{\psi}^{\mathcal{K}}$ are subsequently mapped from the domain $\mathcal{K}$ to the domain Ω by the location of the nodes:

$$\boldsymbol{\psi}_i(\boldsymbol{r}) = \begin{cases} \boldsymbol{\psi}^{\mathcal{K}}(\boldsymbol{r}) & \text{if } \mathcal{K}_e = \Omega_e \\ 0 & \text{if } \mathcal{K}_e \neq \Omega_e. \end{cases} \tag{23}$$

Finally, the local explicit interval field can be computed as:

$$\boldsymbol{x}^{\boldsymbol{I}}(\boldsymbol{r}) = \boldsymbol{x}_{\mu} + \sum_{i=1}^{n_b} \boldsymbol{\psi}_i(\boldsymbol{r}) \cdot (\boldsymbol{\alpha}_i^I - \boldsymbol{x}_{\mu}), \tag{24}$$

where $\boldsymbol{x}_{\mu}$ is the midpoint of the field.

3.4. Local basis functions for three-dimensional domains

For a FE model that is defined on a three-dimensional domain $d_{\Omega} = 3$, the approach for two-dimensional domains applies (see subsection 3.3). In three-dimensional domains the rectangular domain $\mathbb{K}$ becomes a beam with width parameters B_x, B_y, B_z. Inside this domain nine virtual nodes are defined on

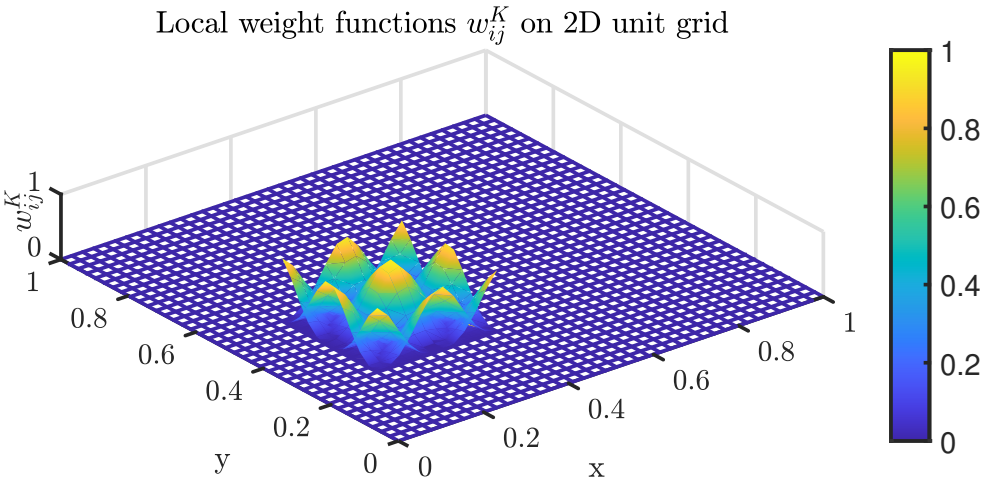


Figure 3: Weight functions on a 2D unit grid of 40×40 elements, on the domain $\mathcal{K}$ with virtual nodes as in figure 2

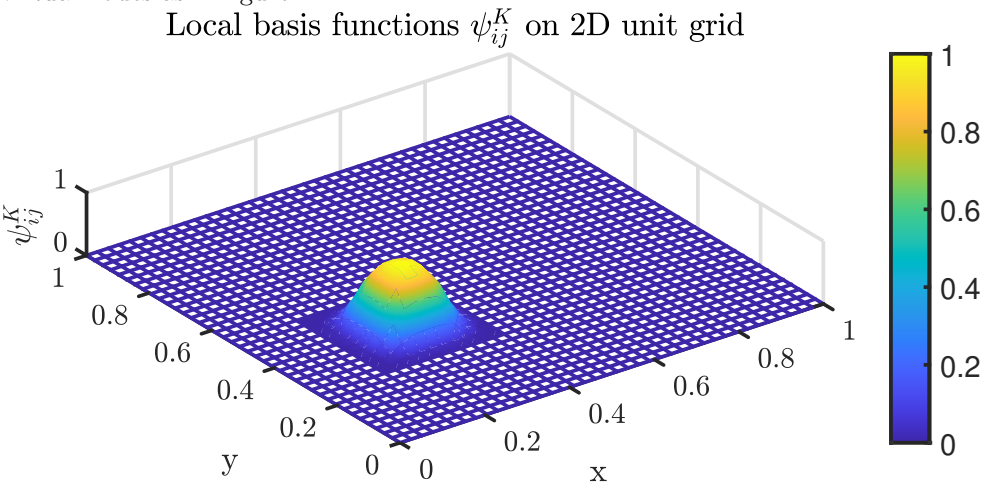


Figure 4: Basis functions on a 2D unit grid of 40×40 elements, on the domain $\mathcal{K}$ with virtual nodes as in figure 2

the edges, where in two-dimensions this is six. The corresponding weight functions defined on those virtual nodes are then: $\boldsymbol{w}^{\mathcal{K}}_{x1}$, $\boldsymbol{w}^{\mathcal{K}}_{x2}$, $\boldsymbol{w}^{\mathcal{K}}_{x3}$, $\boldsymbol{w}^{\mathcal{K}}_{y1}$, $\boldsymbol{w}^{\mathcal{K}}_{y2}$, $\boldsymbol{w}^{\mathcal{K}}_{y3}$, $\boldsymbol{w}^{\mathcal{K}}_{z1}$, $\boldsymbol{w}^{\mathcal{K}}_{z2}$, $\boldsymbol{w}^{\mathcal{K}}_{z3}$. Tree dimensional weight functions are then obtained with the Kronecker Product of the weight matrices:

$$\boldsymbol{W}^{\mathcal{K}}_{x} = \begin{bmatrix} \boldsymbol{w}^{\mathcal{K}}_{x1} & \boldsymbol{w}^{\mathcal{K}}_{x2} & \boldsymbol{w}^{\mathcal{K}}_{x3} \end{bmatrix}, \tag{25}$$

$$\boldsymbol{W}^{\mathcal{K}}_{y} = \begin{bmatrix} \boldsymbol{w}^{\mathcal{K}}_{y1} & \boldsymbol{w}^{\mathcal{K}}_{y2} & \boldsymbol{w}^{\mathcal{K}}_{y3} \end{bmatrix}, \tag{26}$$

$$\boldsymbol{W}^{\mathcal{K}}_{z} = \begin{bmatrix} \boldsymbol{w}^{\mathcal{K}}_{z1} & \boldsymbol{w}^{\mathcal{K}}_{z2} & \boldsymbol{w}^{\mathcal{K}}_{z3} \end{bmatrix}. \tag{27}$$

Those weight matrices contain the individual discretised weight functions.

$$\text{For } d_{\mathcal{K}} = 3 : \boldsymbol{W}^{\mathcal{K}}_{xyz} = \boldsymbol{W}^{\mathcal{K}}_{z} \otimes \left[\boldsymbol{W}^{\mathcal{K}}_{x} \otimes \boldsymbol{W}^{\mathcal{K}}_{y}\right] \tag{28}$$

$$= \boldsymbol{W}^{\mathcal{K}}_{z} \otimes \begin{bmatrix} \boldsymbol{w}^{\mathcal{K}}_{x1} \cdot \boldsymbol{W}^{\mathcal{K}}_{y} & \boldsymbol{w}^{\mathcal{K}}_{x2} \cdot \boldsymbol{W}^{\mathcal{K}}_{y} & \boldsymbol{w}^{\mathcal{K}}_{x3} \cdot \boldsymbol{W}^{\mathcal{K}}_{y} \end{bmatrix} \tag{29}$$

$$= \begin{bmatrix} \boldsymbol{w}^{\mathcal{K}}_{z1} \cdot \boldsymbol{W}^{\mathcal{K}}_{xy} & \boldsymbol{w}^{\mathcal{K}}_{z2} \cdot \boldsymbol{W}^{\mathcal{K}}_{xy} & \boldsymbol{w}^{\mathcal{K}}_{z3} \cdot \boldsymbol{W}^{\mathcal{K}}_{xy} \end{bmatrix}, \tag{30}$$

and $\boldsymbol{W}^{\mathcal{K}}_{xyz}$ With those three-dimensional weight functions, basis functions $\boldsymbol{\psi}^{\mathcal{K}}$ are calculated with equation 14. They are then subsequently mapped from the domain $\mathcal{K}$ to the domain Ω by the location of the nodes:

$$\boldsymbol{\psi}_i(\boldsymbol{r}) = \begin{cases} \boldsymbol{\psi}^{\mathcal{K}}(\boldsymbol{r}) & \text{if } \mathcal{K}_e = \Omega_e \\ 0 & \text{if } \mathcal{K}_e \neq \Omega_e. \end{cases} \tag{31}$$

Finally, the local explicit interval field can be computed as:

$$\boldsymbol{x}^{\boldsymbol{I}}(\boldsymbol{r}) = \boldsymbol{x}_{\mu} + \sum_{i=1}^{n_b} \boldsymbol{\psi}_i(\boldsymbol{r}) \cdot (\boldsymbol{\alpha}^{I}_{i} - \boldsymbol{x}_{\mu}), \tag{32}$$

where $\boldsymbol{x}_{\mu}$ is the midpoint of the field.

3.5. Defining the local domain in engineering applications

In practical engineering application were only limited spatial data are available of the spatial uncertainty, i.e., data are generated in a region of the full part such that there is only information of the spatial uncertainty in a local region. This local NS spatial uncertainty typically stems from the manufacturing process, for example: (1) with casting cooling effects could generate local porosity's, (2) with welding there is a heat effected zone and micro cracks could be introduced. In both examples the manufacturing process effects the material properties of the part or structure, in a local region of the part. As a result, they introduce NS spatial uncertainty. Also, the material properties could be effected on the full geometry yielding global NS spatial uncertainty.

The practical definition of the local explicit interval field starts with the midpoint of the local domain that equals the location of the midpoint of the spatial uncertainty, in both examples this is the location with the most variation on the material properties. The width is then selected based on the region of the spatial uncertainty, were bigger values of the width increase the conservatism. As a result, in the general case where the shape and size of the local spatial uncertainty is not perfectly quantifiable from the data, the analyst is recommended to use a domain that is too big as this increases the conservatism of the results. In case a perfect shape could be derived from the data and the shape is not a rectangular, it is recommended to use a width such that the rectangular domain over-sizes the size of the real shape as this will also be more conservative.

4. Case study 1: Explicit interval Field on an 1D bar

4.1. Problem description

The first case study concerns an explicit interval field on a 1D Finite Element model of a linear elastic bar that is $L = 1000\,\text{mm}$ long. This model is built and solved using Matlab. The FE mesh is uniformly discretised with an element size of $h = 2\,\text{mm}$, resulting in 500 linear first-order elements and converged results. The following boundary conditions apply on the bar: a Dirichlet boundary condition for $x = 0 : u = 0\,\text{mm}$ and a Neumann boundary condition for $x = 100$ mm of $F = 1000\,\text{N}$. The bar has a constant cross-sectional area of $A = 10\,\text{mm}^2$. Figure 5 visualises the bar with its boundary conditions. This academic case study defines the spatial uncertainty locally

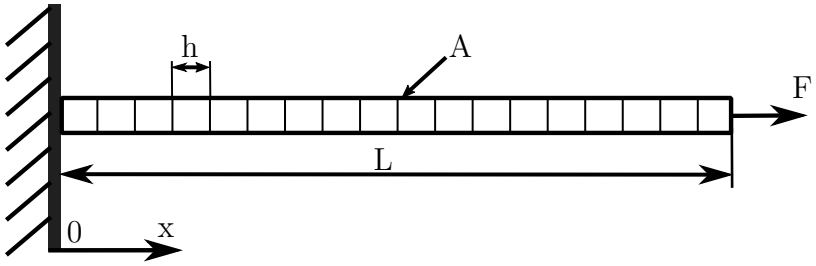


Figure 5: FE model of the 1D bar for this case study with indication of the used boundary conditions and physical properties

on Young's Modulus in three locations (control points). Table 1 gives the coordinates of those control points together with the corresponding interval scalars E_i^I and the width B_i of the domain $\mathcal{K}$ around each control point.

4.2. Modelling of local and global explicit interval fields

This case shows the difference between the local explicit interval field and the global explicit interval field with IDW. For the local explicit interval

Table 1: Uncertainty in 3 control points with location, intervals E_i^I and width of the local domain $\mathcal{K}$ around each control point i

Control point	Location (mm)	Intervals E_i^I (Gpa)	Width B_i (mm)
1	$x = 200$	$[50, 90]$	150
2	$x = 500$	$[50, 90]$	50
3	$x = 800$	$[50, 90]$	40

fields, the weighting functions are constructed using quartic splines as they satisfy the three necessary properties for the definition of a local explicit interval field. Section 2.3 formulates the global interval field with IDW, where equation 9 calculates the IDW functions ${}^{IDW}\boldsymbol{w}_i \in \Omega$ and equation 10 determines the basis functions $\boldsymbol{\psi}_i$. For this case, a power factor $p = 2$ is used. Figure 6 visualises the IDW basis functions and weight functions on the model domain Ω. A quartic spline function ${}^{Q}\boldsymbol{w}_i^{\mathcal{K}}$ from [28] is based on the Euclidean distance $\boldsymbol{d}$ between the location $\boldsymbol{r}_n^{\mathcal{K}}$ of the nodes $n^{\mathcal{K}}$ and the locations $\boldsymbol{r}^{\mathcal{K}}$. The quartic splines are then analytically expressed as:

$$ {}^{Q}\boldsymbol{w}_i^{\mathcal{K}}(\bar{\boldsymbol{d}}) = \begin{cases} 1 - 6\bar{\boldsymbol{d}}^2 + 8\bar{\boldsymbol{d}}^3 - 3\bar{\boldsymbol{d}}^4 & \bar{\boldsymbol{d}} \leq 1 \\ 0 & \bar{\boldsymbol{d}} > 1, \end{cases} \tag{33} $$

where $\bar{\boldsymbol{d}}$ is the normalised Euclidean distance with the support width B. That is defined as:

$$ \bar{\boldsymbol{d}}(\boldsymbol{r}_n^{\mathcal{K}}, \boldsymbol{r}^{\mathcal{K}}) = \frac{\boldsymbol{d}(\boldsymbol{r}_n^{\mathcal{K}}, \boldsymbol{r}^{\mathcal{K}})}{B}. \tag{34} $$

Figure 1 visualises the quartic spline function. Note that also other weighting functions can be applied, as long as they satisfy the three requirements postulated in section 3: (1) the weight function is monotonic decreasing with increasing distance from the control point, (2) the first derivative of the weight

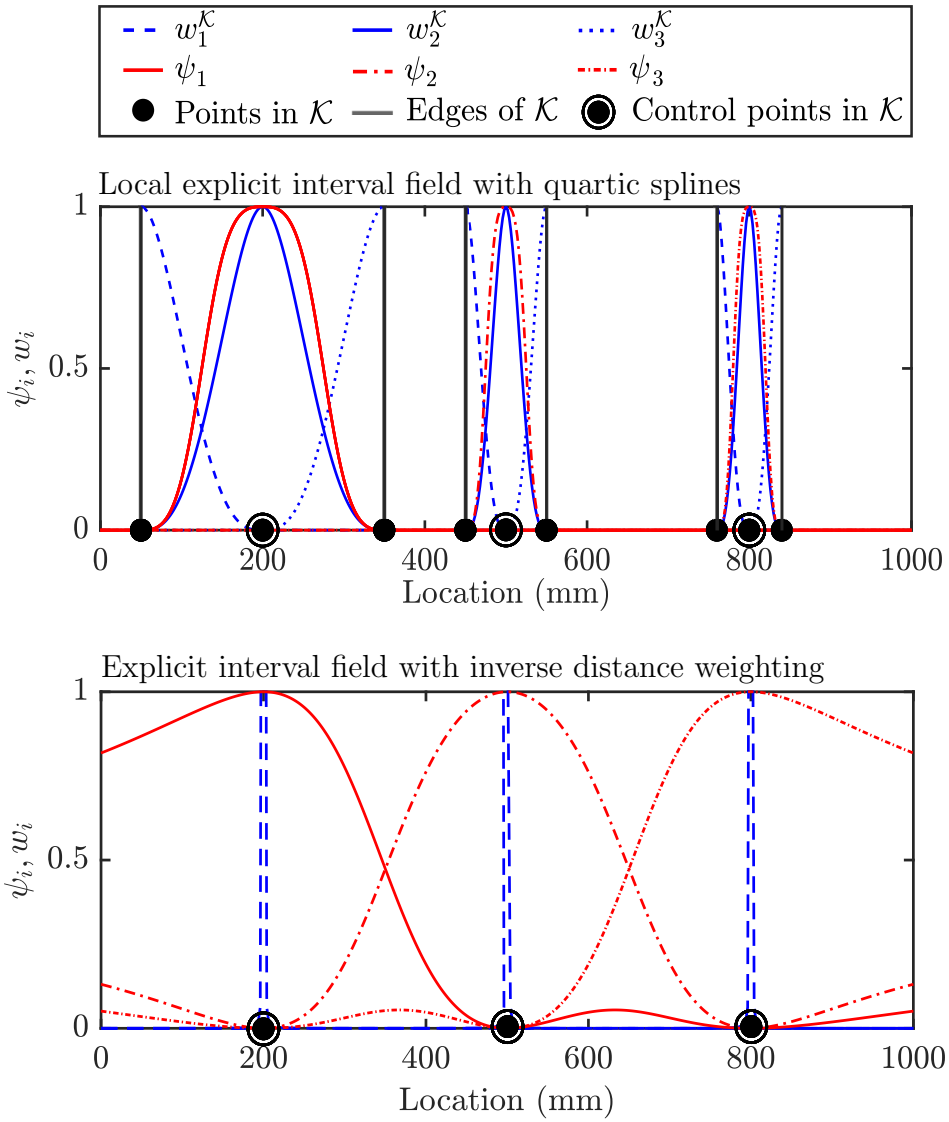


Figure 6: 1D bar with 3 control points (encircled by black dots) and the corresponding weight (blue) and basis function (red). Note that for the global explicit interval field $w_i = \infty$ and $\psi_i = 1$ when $\boldsymbol{r} = \boldsymbol{r}_{n_i}$. The local explicit interval field uses also virtual nodes, visualised as the black dots.

function is 0 on the edge of the local domain and (3) the weight function is non-negative in the local domain and zero outside that domain. Quartic splines satisfy those requirements, from equation 33 the quartic spline is monotonically decreasing and so satisfying requirement one and from equation 34 the weight function satisfies the second and third requirement. From these weight functions equation 14 calculates the basis functions $\boldsymbol{\psi}^{\mathcal{K}} \in \mathcal{K}$ and equation 16 maps the basis functions from the local domain $\mathcal{K}$ to the model domain Ω. Figure 6 visualises the basis functions and weight functions on the model domain Ω. This figure shows that local dependence is successfully obtained with the local explicit interval fields using quartic splines and the same number of control point as the global explicit interval field.

4.3. Interval field propagation

The goal of this propagation is to quantify the interval of the tip displacement y^I of the bar and the strains $\boldsymbol{\epsilon}(\boldsymbol{r})$, based on the spatially uncertain value of Young's modulus. Hereto, realisations of the local/global explicit interval field of equation 17 are propagated. This requires the defined basis functions (see subsection 4.2 for global explicit interval fields and section 3 for the local explicit interval field) and the interval scalars $\alpha_i^I = E_i^I$ and the interval field mean x_μ, as described in table 1. The bar is linear-elastic and so the problem is monotonic and especially strict negative monotonic as a high Young's modulus will cause a low displacement. As a result, the interval bounds on the displacement y^I is quantified with two FE-model realisations one with the upper bounds and one with the lower bounds of the interval scalars. Figures 7 and 8 visualise the difference in realisations between the local explicit interval field and the global explicit interval field. The black lines are the

vertex realisations of the interval hyper-rectangle and the blue lines are Sobol samples of the interval hyper-rectangle. In Figure 7, the vertex and Sobol realisations show that the local explicit interval field is successfully obtained as they are well within the locally defined regions. A big difference between global and local explicit interval fields is visible, the global explicit interval field has uncertainty on the full spatial domain where the local explicit interval field clearly confines the uncertainty on the local spatial domain. This clearly shows that with this approach, the objective of having better control on the spatial localisation of the uncertainty has been reached. The total computing time of the propagation for this interval analysis is $0,09$ seconds in Matlab on 1 CPU thread[1]. The result, being the interval on the tip displacement for local explicit interval field with quartic splines, is:

$$ {}^{Q}y^{I} = \left[\underline{y}, \overline{y}\right] = [1.35, 1.56]\,, \tag{35} $$

for global explicit interval fields the interval is:

$$ {}^{IDW}y^{I} = \left[\underline{y}, \overline{y}\right] = [1.11, 2]\,. \tag{36} $$

The interval for the local explicit interval field is smaller than the one obtained with the global explicit interval field. As such, local explicit interval fields give less conservative results than global explicit interval fields. Figure 9 visualises the strains $\boldsymbol{\epsilon}(\boldsymbol{r})$ in the bar, which shows the difference between local and global explicit interval fields. For the local explicit interval fields, local uncertainty in the strains is obtained, resulting in less conservative results for the local explicit interval field.

[1] 2x AMD EPYC 7601 32-Core @2.20Ghz

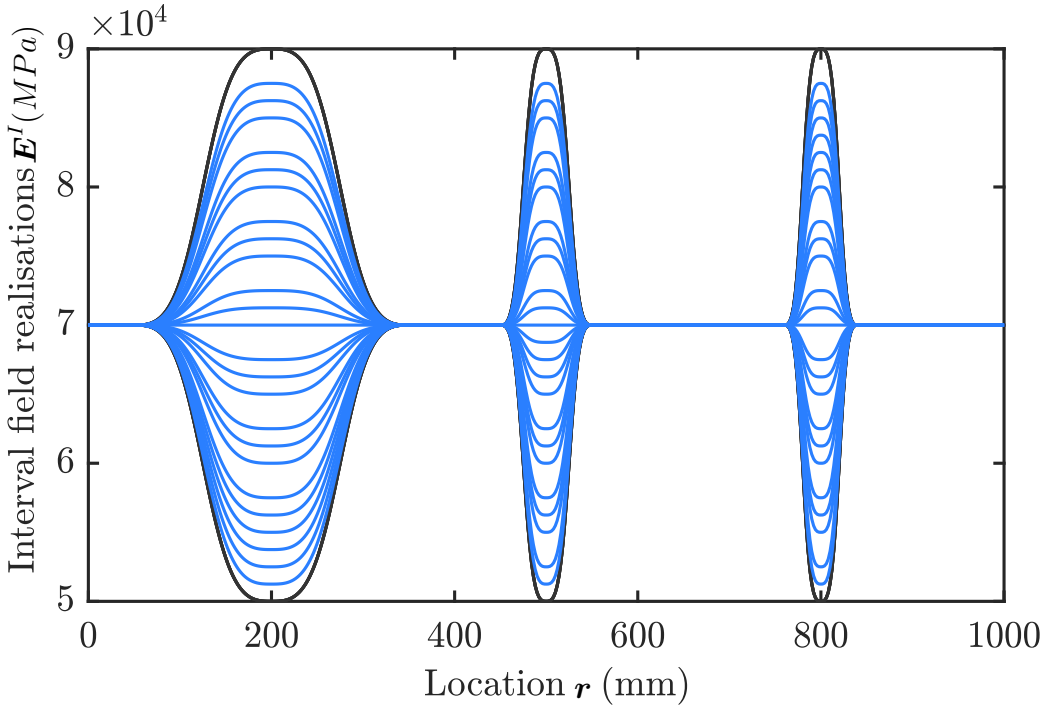


Figure 7: Interval field realisations for the local explicit interval field with quartic splines. The black lines show vertex realisations of input hypercube and the blue lines realisations of 25 Sobol samples.

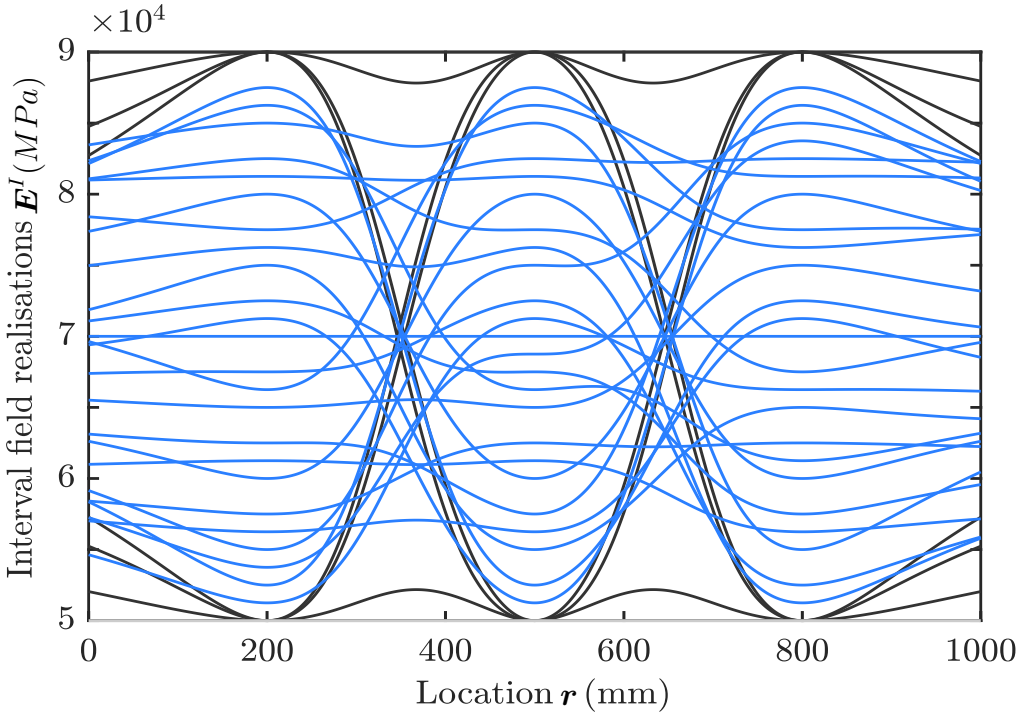


Figure 8: Interval field realisations for the explicit interval field with inverse distance weighting. The black lines show vertex realisations and the blue lines 25 sobol samples.

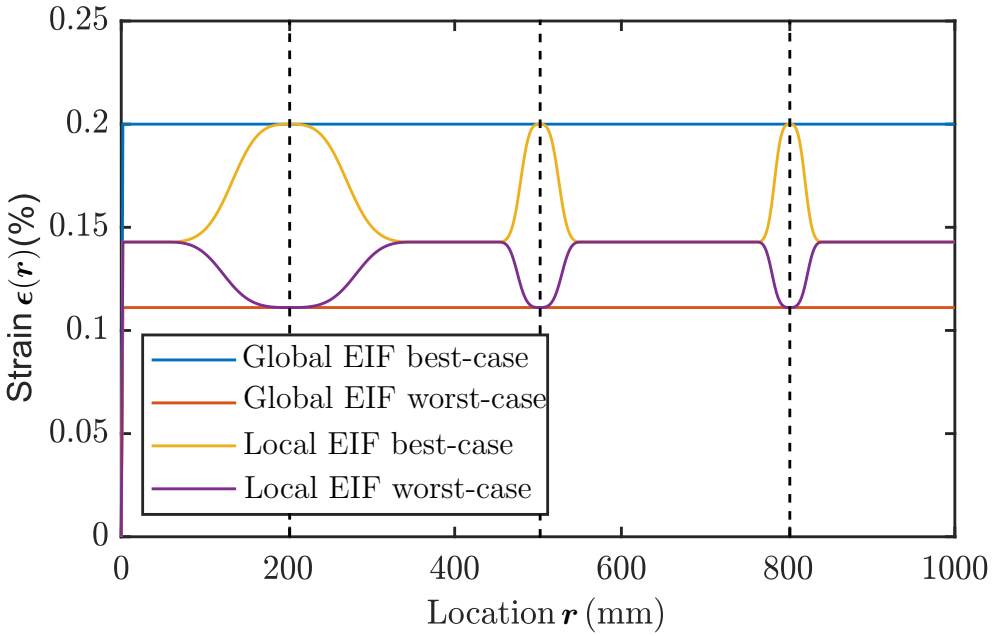


Figure 9: Strains $\boldsymbol{\epsilon}(\boldsymbol{r})$ for the worst and best-case, showing the global explicit interval field (EIF) and the local explicit interval field (EIF) with quartic splines, the black dashed line are the locations of the control points

4.4. Difference in computational cost

The computational cost in interval analysis is most often concentrated in propagating the uncertainty and less in modelling the field. However, when using the same interval scalars on the same locations for both techniques and especially for high dimensional FE-models (> 1M DOF), the computational cost of modelling the field becomes important. Modelling the field commonly requires the computation of distance measures to calculate the basis functions, namely $O(k_{fe} \times n_b)$ for global explicit interval fields and $O\left(\sum_{i=1}^{n_b} k_i\right)$ for the local explicit interval field. For this specific case, global explicit interval fields requires three weight functions on the full model domain Ω, resulting in:

$$\frac{3L}{h} = \frac{3 \cdot 1000}{2} = 1\,500 \text{ distance measures.} \tag{37}$$

The used RAM memory for this operation is 12 kBytes. The Local explicit interval field defines the three weighting functions on a local domain $\mathcal{K}$ around there control point, resulting in:

$$3\left(\frac{2B_1}{h} + \frac{2B_2}{h} + \frac{2B_3}{h}\right) = 720 \text{ distance measures.} \tag{38}$$

The used RAM memory for this operation is 5.8 kBytes. This reduces the number of distance measures and RAM usage with an order of magnitude 2.08 for calculating the field when using the local explicit interval field formulation.

5. Case study 2: Interval Field on a 2D plate

5.1. Problem description

This case study considers the definition of an interval field for a 2D FE linear elastic model of a square plate, with length and width $L = 1\,\text{m}$ and

uniformly discretised with linear square elements of size $h = 0.0125\,\text{mm}$, resulting in 6 400 elements and converged results. This model is built and solved using Matlab. A Dirichlet boundary condition constrains the plate for $y = 0 : u = 0\,\text{mm}$. The plate has a uniform thickness of $t = 25\,\text{mm}$. Figure 10 visualises the plate with its boundary conditions. This case defines

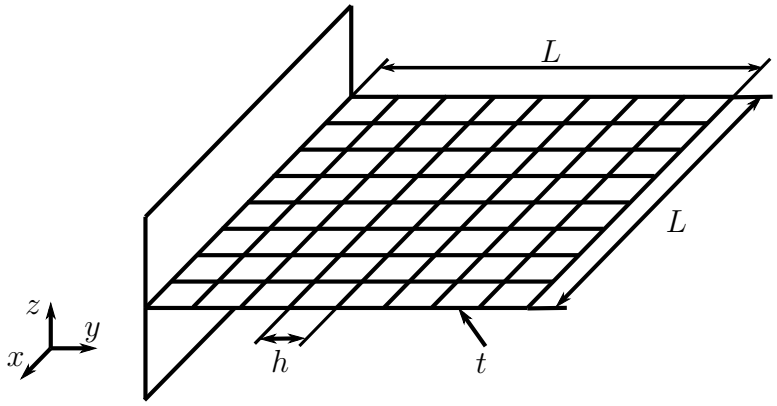


Figure 10: 2D case that is used for this case study with indication of the used boundary conditions and physical properties

uncertainty on Young's modulus and material density, using respectively two and three control points. Table 2 and 3 give the coordinates (x_i,y_i) of those control points, the interval values E_i^I and ρ_i^I, and the width (B_{xi}, B_{yi}) of the domain $\mathcal{K}$ around there control points.

Table 2: Uncertainty in 2 control points with location, intervals E_i^I and width of the local domain $\mathcal{K}$ around each control point i

Control point	Location (x, y)	Intervals E^I	Width (B_x, B_y)
1	$(0.2, 0.3)$ (m)	$[50, 90]$ (GPa)	$(0.1, 0.15)$ (m)
2	$(0.8, 0.1)$ (m)	$[50, 90]$ (GPa)	$(0.2, 0.1)$ (m)

Table 3: Uncertainty in 3 control points with location, intervals ρ_i^I and width of the local domain $\mathcal{K}$ around each control point i

Control point	Location (x, y)	Intervals ρ^I	Width (B_x, B_y)
1	$(0.5, 0.3)$ (m)	$[2600, 2725]$ (kg/m^3)	$(0.1, 0.15)$ (m)
2	$(0.27, 0.35)$ (m)	$[2725, 2900]$ (kg/m^3)	$(0.05, 0.20)$ (m)
3	$(0.8, 0.23)$ (m)	$[2600, 2900]$ (kg/m^3)	$(0.25, 0.1)$ (m)

5.2. Modelling of local and global explicit interval fields

Similar to the previous case, the goal of this case study is to show the difference between the modelling of global and local explicit interval fields in terms of applicability and computational cost in two dimensions. Similar to subsection 4.2 the global explicit interval fields using IDW are defined as described in 2.3. With for this case a power factor of $p = 2$. The local explicit interval fields in two dimensions calculates three local weight functions $\boldsymbol{w}_{x_i}^{\mathcal{K}}, \boldsymbol{w}_{y_i}^{\mathcal{K}}$ (quartic splines) in each dimension separately with equation 34 and equation 33, such that 6 weight functions are obtained. Two-dimensional weight functions are then calculated from those six one-dimensional weight functions with equation 20, and the basis functions $\boldsymbol{\psi}^{\mathcal{K}}$ with equation 14. The resulting weight and basis functions inside $\mathcal{K}$ are similar to the one visualised in respectively figure 3 and figure 4. These figures show that the dependency is limited to the local domain. The basis functions are then mapped with Equation 16 from the local domain $\mathcal{K}$ to the model domain Ω.

5.3. Interval field propagation

This case quantifies the interval of the first three eigenfrequencies f_1^I,f_2^I, f_3^I, by performing (anti-)optimisation. For this optimisation an genetic al-

gorithm is used for each output bound, resulting in 6 optimisation problems that require all several model evaluations, leading to a computing time for this case of multiple hours or days depending on the optimisation scheme and stopping criterion. For this case a max of 30 generations is used yielding a computing time of 1 day for each optimisation using Matlab on 16 CPU threads[2]. The genetic algorithm will search for a local minimum or maximum, which is here considered to be sufficient for giving an estimate of the interval bounds. Other optimisation strategies could also be applied without lose of generality of the proposed local explicit interval fields.

This results for the first three eigenfrequencies in the following intervals for local explicit interval fields:

$$^{Q}f_1^I = \left[\underline{f}_1, \overline{f}_1\right] = [20.8, 21.4]\,(Hz), \tag{39}$$

$$^{Q}f_2^I = \left[\underline{f}_2, \overline{f}_2\right] = [50.7, 51.4]\,(Hz), \tag{40}$$

$$^{Q}f_3^I = \left[\underline{f}_3, \overline{f}_3\right] = [129.7, 131.0]\,(Hz), \tag{41}$$

and for the global explicit interval fields to:

$$^{IDW}f_1^I = \left[\underline{f}_1, \overline{f}_1\right] = [17.7, 24.6]\,(Hz), \tag{42}$$

$$^{IDW}f_2^I = \left[\underline{f}_2, \overline{f}_2\right] = [43.2, 59.6]\,(Hz), \tag{43}$$

$$^{IDW}f_3^I = \left[\underline{f}_3, \overline{f}_3\right] = [109.7, 151.6]\,(Hz). \tag{44}$$

The resulting intervals are different for the two uncertainty modelling techniques. This originates from the different basis functions in the local explicit interval field and the global explicit interval field. The local explicit

[2]2x AMD EPYC 7601 32-Core @2.20Ghz

interval field has smaller bounds than the global explicit interval field, as a result when only local uncertainty is present in the component, the local uncertainty modelling technique will be less conservative than the global uncertainty modelling technique.

5.4. Difference in computational cost

For this case study two basis functions have to be computed. The global explicit interval field requires the calculation of two weighting functions on the full model domain Ω, such that the total number of distance measures equals:

$$2\left(\frac{L}{h}\right)^2 + 3\left(\frac{L}{h}\right)^2 = 5 \cdot \left(\frac{1}{0.0125}\right)^2 = 32\,000. \tag{45}$$

Local explicit interval fields require in two-dimensions six weighting functions (=three in each dimension) on a local domain $\mathcal{K}$ around their control point. So more weighting functions yet less distance measures are required due to the smaller domain. The distance measures for this modelling technique are:

$$\frac{3}{h}\left(B_i^E + B_i^\rho\right) = 660, \tag{46}$$

were B_i^E and B_i^ρ are the width of the local domain $\mathcal{K}$ in the dimensions x and y respectively for Young's modulus and material density as in table 2 and table 3. The memory used during the distance measures is for the local explicit interval fields in total 5 kBytes and for the global explicit interval fields 256 kBytes. As a result, using the local explicit interval field formulation again reduces the computational cost substantially.

6. Case study 3: Local explicit Interval Field on a 3D high dimensional FE-model

6.1. Problem description

This case study concerns the modelling of a local explicit interval field for a high dimensional (HD) 3D FE linear elastic model with 1 763 109 second order tetra elements, resulting in 16 345 542 DOFs and converged results. This model is solved with Nastran in combination with our in-house build uncertainty modelling and propagation toolbox. Inside this model a pressure of 1 500 kPa is applied. A local explicit interval field is used to model the

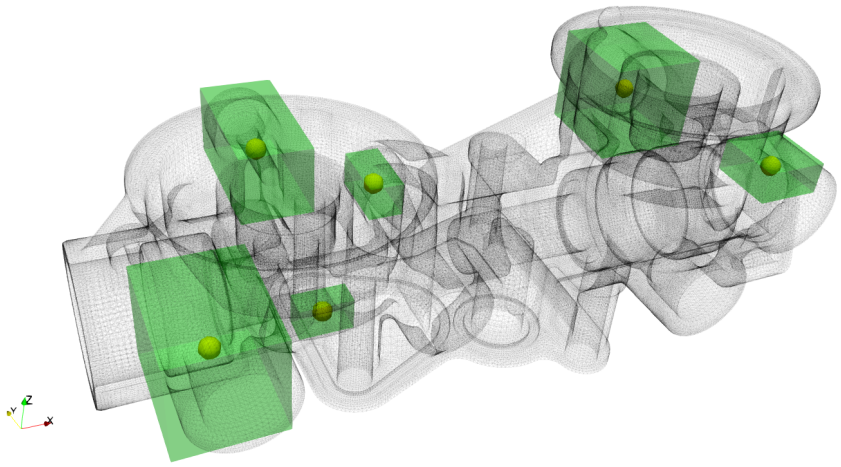


Figure 11: 3D HD FE linear elastic model used in this case study. The yellow points are the control points and the domain of local dependence is indicated in green.

uncertainty on Young's modulus with six control points. Table 4 gives the coordinates (x_i,y_i,z_i) of those control points, the interval values E_i^I and the width B_{xi}, B_{yi}, B_{zi} of the domain $\mathcal{K}$ around each control point. The midpoint

of this interval field is 70 GPa and constant over the full geometry of the model.

Table 4: Uncertainty in 6 control points with their location, intervals E_i^I and width of the local domain $\mathcal{K}$ around each control point i

Control point	Location (x, y, z)	Intervals E^I	Width (B_x, B_y, B_y)
1	$(90, 25, 53)$ mm	$[56, 84]$ GPa	$(15, 15, 15)$ mm
2	$(-30, 28, 49)$ mm	$[50, 80]$ GPa	$(10, 20, 10)$ mm
3	$(0, 15, 40)$ mm	$[50, 80]$ GPa	$(5, 10, 5)$ mm
4	$(-55, -15, 20)$ mm	$[50, 80]$ GPa	$(15, 20, 15)$ mm
5	$(110, -25, 50)$ mm	$[50, 80]$ GPa	$(10, 10, 5)$ mm
6	$(-20, 0, 10)$ mm	$[50, 80]$ GPa	$(8, 4, 5)$ mm

6.2. Computation of the interval field

The local explicit interval fields requires in three-dimensions, nine weighting functions (=three in each dimension) on a local domain $\mathcal{K}$ around one control point. The local explicit interval field requires distance measures within the local domains. For example, for the first two control points, the first and second domain contain respectively 265 533 and 198 926 nodes, resulting in 464 459 distance calculations when considering only two domains. In the case of the global explicit interval field approach the computational cost is higher as, for this model it would require $2 \times 1\,763\,109 = 3\,526\,218$ distance measures, a factor of 7.59 higher. The factor changes when the number of control points increases or decreases. Table 5 gives the required distance measures with increasing number of control point for both modelling

techniques, and shows that the required distance calculations depend on the size B_i of the local domain around the distinct control points.

Table 5: Number of distance calculations for global dependence versus local dependence with increasing number of control points, that are defined in table 4 with there location and width B_i

Control points	Global dependency	Local dependency	Factor
1	1 763 109	265 533	6.64
2	3 526 218	464 459	7.59
3	5 289 327	482 633	10.96
4	7 052 436	798 922	8.83
5	8 815 545	849 518	10.38
6	10 578 654	859 619	12.31

6.3. Propagation of the interval field

The case quantifies the interval width on the von Mises stress for the full geometry, by performing vertex analysis. This is to reduce the enormous computational cost of (anti-)optimisation as it requires a numerous amount of model evaluations for each interval bound. Also one model evaluation uses 253 GB of RAM and 1 h 32 min of computing time on 16 CPU threads[3]

[3] 2x AMD EPYC 7601 32-Core @2.20Ghz. For this case, vertex analysis is less accurate than (anti-)optimisation as the case is not monotonic, nevertheless it gives an estimate of the interval bounds while limiting the model evaluations to 2^8. Yielding an acceptable total computing time of 4 days. The interval width is then computed as:

$$\boldsymbol{\sigma}(\boldsymbol{r})_{vM,width} = \underline{\boldsymbol{\sigma}}(\boldsymbol{r})_{vM} - \overline{\boldsymbol{\sigma}}(\boldsymbol{r})_{vM}. \tag{47}$$

Figure 12 shows the interval width on the model, figure 13 visualises the right corner of section A. On that figure, 2 regions are highlighted to show the difference in interval width and bounds through the model. In the upper region a wider interval is noticed. This wider interval is obtained from the local non-stationary character of the input interval field, that has a wider interval in the upper region than in the lower region. This is also visible on figure 12 in the other regions were input uncertainty is defined. Hereto, we used our developed in-house code for visualising spatial uncertainty in high dimensional FE-models. With this case it is shown that the local explicit interval field is able to model local non-stationary uncertainty on high dimensional models with increased local control over global explicit interval fields.

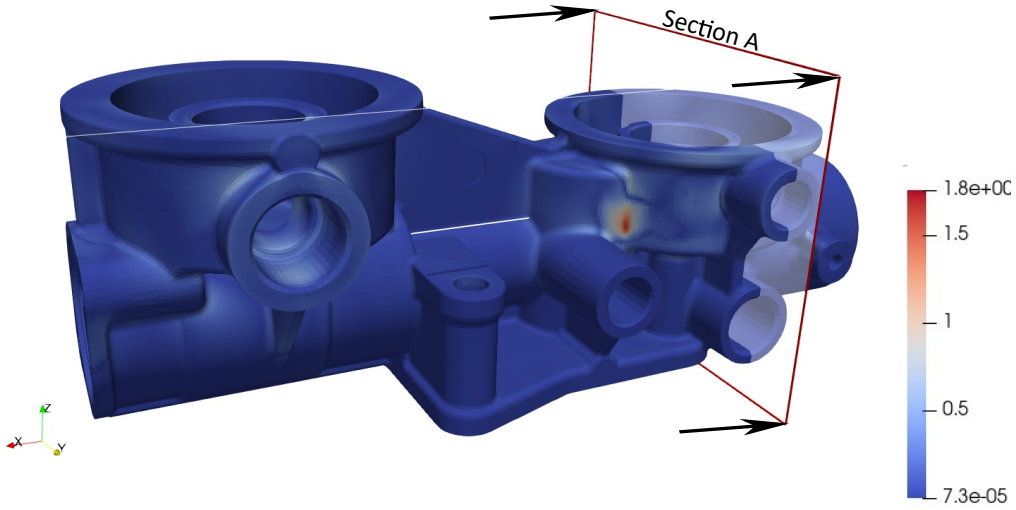


Figure 12: Interval width on the von Mises stress

where $\underline{\boldsymbol{\sigma}}(\boldsymbol{r})_{vM}$ and $\overline{\boldsymbol{\sigma}}(\boldsymbol{r})_{vM}$ are respectively the maximal responses of the model.

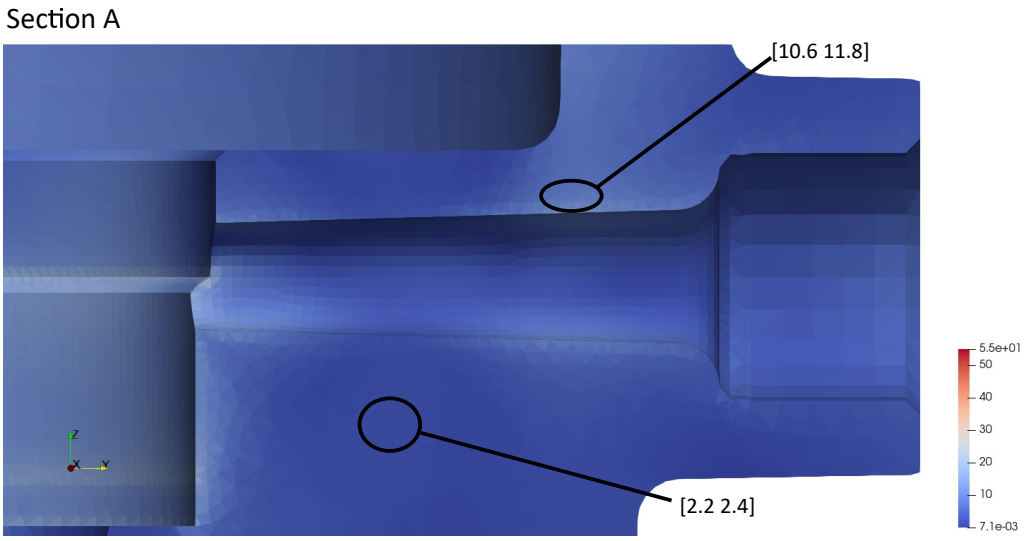


Figure 13: Interval width in the right corner of section A, with the interval bounds of two specific regions

7. Conclusion

This paper presents an approach to model a local explicit interval field to represent locally spatial uncertain parameters in Finite Element models. To characterise the local character of the interval field this paper proposes three requirements: (1) the weight function is a monotonically decreasing function with respect to the distance to the control point, (2) the weight function is continuous on the edge of the local domain up to its first derivative and (3) the weight function is strictly positive within a specific region around the control point, and zero outside that region. The paper proposes the use of quartic splines, as these function satisfy the proposed requirements. Next, two academic and one industrial relevant case studies are performed to illustrate the difference between local explicit interval fields and global explicit

interval fields using the inverse distance weighting technique. With these test cases, it is shown that the local explicit interval fields are able to model local dependence. The test cases also show the increased local control on the spatial localisation of the uncertainty in comparison with global explicit interval fields. The local explicit interval field are as such more applicable in realistic industrial applications, where there is local uncertainty. Other advantages of the local explicit interval fields are in reducing the computational cost (the number of distance measures) to model the interval field, as well as decreasing the over-conservatism when local uncertainty is present in the modelled part or structure. Please note that the technique, as presented in this paper, is limited to rectangular dependency regions in the model domain. The extension of the method to other, non-rectangular domains is left for future research. Also the combination of a global and local fields is left for future research, where the first could capture an overall trend and the second more local behaviour in the spatial uncertainty.

Acknowledgements

The authors would like to acknowledge the financial support of the Research Foundation - Flanders in the framework of the research project G0C2218N and the postdoctoral grant 12P359N of Matthias Faes.

References

[1] Q. Zou, L. Deng, C. Jiang, Predicting the bounds of vehicle-induced bridge responses using the interval analysis method, Journal of Bridge Engineering 21 (9) (2016) 04016046.

[2] X. Wang, L. Wang, Uncertainty quantification and propagation analysis of structures based on measurement data, Mathematical and Computer Modelling 54 (11) (2011) 2725 – 2735.

[3] X. Wang, L. Wang, Z. Qiu, A feasible implementation procedure for interval analysis method from measurement data, Applied Mathematical Modelling 38 (9) (2014) 2377 – 2397.

[4] X. Long, C. Jiang, K. Liu, X. Han, W. Gao, B. Li, An interval analysis method for fatigue crack growth life prediction with uncertainty, Computers & Structures 210 (2018) 1 – 11.

[5] M. Beer, V. Kreinovich, Interval or moments: which carry more information?, Soft Computing 17 (8) (2013) 1319–1327.

[6] M. M. Dannert, A. Fau, R. M. Fleury, M. Broggi, U. Nackenhorst, M. Beer, A probability-box approach on uncertain correlation lengths by stochastic finite element method, PAMM 18 (1) (2018) e201800114.

[7] J. Song, P. Wei, M. Valdebenito, S. Bi, M. Broggi, M. Beer, Z. Lei, Generalization of non-intrusive imprecise stochastic simulation for mixed uncertain variables, Mechanical Systems and Signal Processing 134 (2019) 106316.

[8] M. Faes, D. Moens, Recent trends in the modeling and quantification of non-probabilistic uncertainty, Archives of Computational Methods in Engineering (2019) 1–39.

[9] R. L. Muhanna, R. L. Mullen, Uncertainty in mechanics

problems—interval–based approach, Journal of Engineering Mechanics 127 (6) (2001) 557–566.

[10] V. Kreinovich, S. Ferson, A new cauchy-based black-box technique for uncertainty in risk analysis, Reliability Engineering & System Safety 85 (1) (2004) 267 – 279.

[11] A. C. Calder, M. M. Hoffman, D. E. Willcox, M. P. Katz, F. D. Swesty, S. Ferson, Quantification of incertitude in black box simulation codes, Journal of Physics: Conference Series 1031 (2018) 012016.

[12] D. Moens, M. De Munck, W. Desmet, D. Vandepitte, Numerical dynamic analysis of uncertain mechanical structures based on interval fields, in: IUTAM symposium on the vibration analysis of structures with uncertainties, Springer, 2011, pp. 71–83.

[13] M. Broggi, M. Faes, E. Patelli, Y. Govers, D. Moens, M. Beer, Comparison of bayesian and interval uncertainty quantification: Application to the airmod test structure, in: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2017, pp. 1–8.

[14] M. Faes, D. Moens, Identification and quantification of spatial variability in the elastostatic properties of additively manufactured components, in: 19th AIAA Non-Deterministic Approaches Conference, 2017, p. 1771.

[15] M. Faes, D. Moens, Identification and quantification of spatial interval uncertainty in numerical models, Computers & Structures 192 (2017) 16–33.

[16] C. van Mierlo, M. G. Faes, D. Moens, Inhomogeneous interval fields based on scaled inverse distance weighting interpolation, Computer Methods in Applied Mechanics and Engineering 373 (2020) 113542.

[17] M. Imholz, D. Vandepitte, D. Moens, Analysis of the effect of uncertain clamping stiffness on the dynamical behaviour of structures using interval field methods, in: Applied mechanics and materials, Vol. 807, Trans Tech Publ, 2015, pp. 195–204.

[18] A. Sofi, E. Romeo, A novel interval finite element method based on the improved interval analysis, Computer Methods in Applied Mechanics and Engineering 311 (2016) 671–697.

[19] A. Sofi, G. Muscolino, I. Elishakoff, Static response bounds of timoshenko beams with spatially varying interval uncertainties, Acta Mechanica 226 (11) (2015) 3737–3748.

[20] Y. Luo, J. Zhan, J. Xing, Z. Kang, Non-probabilistic uncertainty quantification and response analysis of structures with a bounded field model, Computer Methods in Applied Mechanics and Engineering 347 (2019) 663 – 678.

[21] B. Ni, C. Jiang, J. Li, W. Tian, Interval k-l expansion of interval process model for dynamic uncertainty analysis, Journal of Sound and Vibration 474 (2020) 115254.

[22] J. Feng, Q. Li, A. Sofi, G. Li, D. Wu, W. Gao, Uncertain structural free vibration analysis with non-probabilistic spatially varying parameters,

ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg 5 (2), 021005 (04 2019).

[23] Z. Wang, C. Jiang, B. Ni, C. Wang, J. Zhong, T. Fang, An interval finite element method for electromagnetic problems with spatially uncertain parameters, Science China Technological Sciences 63 (1) (2020) 25–43.

[24] A. Sofi, Structural response variability under spatially dependent uncertainty: stochastic versus interval model, Probabilistic Engineering Mechanics 42 (2015) 78–86.

[25] D. Wu, W. Gao, Uncertain static plane stress analysis with interval fields, International Journal for Numerical Methods in Engineering 110 (13) (2017) 1272–1300.

[26] B. Xia, L. Wang, Non-probabilistic interval process analysis of time-varying uncertain structures, Engineering Structures 175 (2018) 101–112.

[27] J. Li, B. Ni, C. Jiang, T. Fang, Dynamic response bound analysis for elastic beams under uncertain excitations, Journal of Sound and Vibration 422 (2018) 471–489.

[28] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: an overview and recent developments, Computer methods in applied mechanics and engineering 139 (1) (1996) 3–47.

[29] D. Moens, M. Hanss, Non-probabilistic finite element analysis for parametric uncertainty treatment in applied mechanics: Recent advances, Finite Elements in Analysis and Design 47 (1) (2011) 4–16.

[30] M. Faes, D. Moens, On auto-and cross-interdependence in interval field finite element analysis, International Journal for Numerical Methods in Engineering 121 (9) (2020) 2033–2050.

[31] M. Faes, R. Callens, D. Moens, Adaptive sparse grid approximation for high dimensional interval field construction, in: AIAA Scitech 2020 Forum, 2020, p. 1418.