Universität
Stuttgart

Uncertainty modeling using efficient fuzzy arithmetic based on sparse grids: applications to dynamic systems

Andreas Klimke, Kai Willner, Barbara Wohlmuth

Berichte aus dem Institut für Angewandte Analysis und Numerische Simulation

Preprint 2004/003

Universität Stuttgart

Uncertainty modeling using efficient fuzzy arithmetic based on sparse grids: applications to dynamic systems

Andreas Klimke, Kai Willner, Barbara Wohlmuth

Berichte aus dem Institut für Angewandte Analysis und Numerische Simulation

Preprint 2004/003

Institut für Angewandte Analysis und Numerische Simulation (IANS)
Fakultät Mathematik und Physik
Fachbereich Mathematik
Pfaffenwaldring 57
D-70 569 Stuttgart

E-Mail: ians-preprints@mathematik.uni-stuttgart.de
WWW: http://preprints.ians.uni-stuttgart.de

ISSN **1611-4176**

Uncertainty modeling using efficient fuzzy arithmetic based on sparse grids: applications to dynamic systems

Andreas Klimke, Kai Willner, Barbara Wohlmuth

Abstract

Fuzzy arithmetic provides a powerful tool to introduce uncertainty into mathematical models. With Zadeh's extension principle, one can obtain a fuzzy-valued extension of any objective function. An efficient and accurate approach to compute expensive multivariate functions of fuzzy numbers is given by fuzzy arithmetic based on sparse grids. In this paper, we illustrate the general applicability of this new method by computing two dynamic systems subjected to uncertain parameters as well as uncertain initial conditions. The first model consists of a system of delay differential equations simulating the periodic outbreak of a disease. In the second model, we consider a multibody mechanism described by an algebraic differential equation system.

Keywords: fuzzy numbers; extension principle; uncertainty modeling; multibody mechanism; differential equations

---

1 Introduction

Fuzzy sets [1] provide a widely appreciated tool to introduce uncertain parameters into mathematical models. Especially in engineering applications (e.g. uncertain model parameters, manufacturing tolerances), a great interest in modeling uncertain input data and the effects on the output of the model can be observed. Zadeh's well-known extension principle [2] forms the theoretical basis of fuzzy arithmetic, extending real-valued functions to functions of fuzzy numbers. In practice, however, the implementation of fuzzy arithmetic turns out to be a problem of nonlinear programming, which is commonly difficult to solve and also computationally expensive.

When dealing with expensive multivariate nonlinear models, such as complex dynamic systems, classical methods, such as LR-fuzzy numbers according to Dubois and Prade [3], fuzzy arithmetic based on interval arithmetic according to Kaufmann and Gupta [4], or similar variations (e.g. [5]), often fail to produce satisfying results. In many cases, these methods are not even applicable, since the required decomposition of the model into its elementary operations is hardly possible. In other cases, the classical methods may severely overestimate the actual fuzzy results, as shown in [6, 7, 8]. As a consequence, modern algorithms for fuzzy arithmetic based on interval-based branch-and-bound codes [9, 10, 11], and point-based methods, such as the vertex method [6] (only applicable to monotonic functions) or the transformation method

[7], were developed. In case of non-monotonic functions, point-based methods may suffer from very high computational complexity, despite some possibilities for improvement [12, 13]. Interval branch-and-bound methods are only feasible if the model can be extended to permit interval-valued evaluations, which turns out to be either impractical or at least inconvenient, since it requires a significant amount of additional programming effort compared to models using only real-valued variables, operators, solvers, and so forth.

These problems have lead to the development of a new point-based approach, which we call sparse grid fuzzy arithmetic [14]. This technique can be applied directly to most nonlinear models and requires only real-valued function evaluations. The pre-defined structure of the sparse grid permits to construct interpolants for multiple output arguments at once without requiring additional function evaluations. As the only prerequisite, the objective model must satisfy some general smoothness criteria. The method can be summarized as follows: Using sparse grids, we construct a surrogate function with a low number of function evaluations, even in higher dimensions. Then, by applying constrained fuzzy arithmetic to the surrogate function (i.e. determining the accurate range of the result for each subsequent set of α-cuts given by the fuzzy input parameters), we compute a good approximation of the actual fuzzy result. In Section 2, we briefly review this new approach to fuzzy arithmetic. In Section 3, we apply the sparse-grid-based fuzzy arithmetic algorithm to two dynamic systems computed with uncertain input data, a delay differential equation system simulating the periodic outbreak and progression of an epidemic based on a Kermack-McKendrick model [15, 16], and an algebraic differential equation system (Andrews' squeezer mechanism [17, 18], a well-known planar benchmark problem for a multibody mechanism).

2 Fuzzy arithmetic using sparse grids

The main idea of the method is to compute a sparse grid interpolant $A_{q,d}(f)$ of the objective function f with sufficient accuracy for the base box Ω of the fuzzy input parameters (see Section 2.2.1), using only a low number of real-valued function evaluations. This holds even in higher dimensions due to the very moderate growth of the number of sparse grid points with increasing problem dimension. The hierarchical structure of the sparse grid interpolation scheme permits to subsequently increase the interpolation depth until a sufficient estimated relative or absolute accuracy is reached. One can then replace all interval- or real-valued function evaluations that become necessary with the application of an accurate implementation of Zadeh's extension principle by evaluations of the interpolant $A_{q,d}(f)$. A brief summary on how this can be done is given in Section 2.2.2.

The accuracy of the obtained fuzzy result depends on both parts of the algorithm. The initial contributor to the quality of the fuzzy result is of course the accuracy of the interpolant, which can be easily monitored during its hierarchical construction. The order of convergence in the maximum norm is well known, as stated in Eq. (11). We thus have a solid basis for the second part of the algorithm. The quality of the final fuzzy result, however, can still vary depending on the actual fuzzy arithmetic algorithm chosen to implement Zadeh's extension principle.

2.1 Part 1: Sparse grid interpolation

The interpolation problem considered with sparse grid interpolation is an optimal recovery problem (i.e. the selection of points such that a smooth function can be approximated with a suitable interpolation formula). Here, well-known univariate interpolation formulas are extended to the multivariate case by using tensor products in a special way. As a result, one obtains a powerful interpolation method that requires significantly less support nodes than conventional interpolation on a full grid. The difference in the number of required points compared to the full grid quickly amounts to several orders of magnitude with increasing problem dimension. The points comprising the multidimensional sparse grid are hereby selected in a predefined fashion. The most important property of the method constitutes the fact that the asymptotic quadratic error decay of full grid interpolation with increasing grid resolution is preserved up to a logarithmic factor. An additional benefit of the sparse grid interpolant is its hierarchical structure, which can be used to obtain an estimate of the current approximation error. Thus, one can formulate an algorithm that aborts automatically when a desired (estimated) accuracy is reached.

2.1.1 Smolyak's algorithm

In the following, we briefly review Smolyak's construction for multivariate interpolation, which forms the basis of sparse grid interpolation. The notation adheres to [19]. We would like to approximate functions $f : [0,1]^d \to \mathbb{R}$ using a finite number of support nodes. For simplicity, we restrict the domain to the d-dimensional unit cube, however, we can easily handle arbitrary axis-aligned boxes by rescaling later on. In the one-dimensional case, an interpolation formula is given by

$$U^i(f) = \sum_{j=1}^{m_i} a_j^i \cdot f(x_j^i), \tag{1}$$

with $i \in \mathbb{N}$, the basis functions $a_j^i \in C([0,1])$, $a_j^i(x_l^i) = \delta_{jl}$, $l \in \mathbb{N}$, and $x_j^i \in X^i = \{x_1^i, \ldots, x_{m_i}^i\}$, $x_k^i \in [0,1], 1 \leq k \leq m_i$. To obtain an interpolation formula for the multivariate case, one can use the tensor product formula

$$(U^{i_1} \otimes \cdots \otimes U^{i_d})(f) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_d=1}^{m_{i_d}} (a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}) \cdot f(x_{j_1}^{i_1}, \ldots, x_{j_d}^{i_d}). \tag{2}$$

However, the above formula requires a very high number of $m_{i_1} \cdots m_{i_d}$ support nodes, which are sampled on the full grid. To reduce the number of support nodes while maintaining the properties of the interpolation formula for $d = 1$, Smolyak's construction is used. With $U^0 = 0$, $\Delta^i = U^i - U^{i-1}$, $|\mathbf{i}| = i_1 + \cdots + i_d$ for $\mathbf{i} \in \mathbb{N}^d$, and $q \geq d, q \in \mathbb{N}$, the Smolyak algorithm is given by

$$A_{q,d}(f) = \sum_{|\mathbf{i}| \leq q} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d})(f) \tag{3}$$

$$= \sum_{k=0}^{n:=q-d} \underbrace{\sum_{|\mathbf{i}|=k+d} (\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d})(f)}_{\Delta A_{k+d,d}(f)}, \tag{4}$$

with

$$\Delta A_{k+d,d}(f) = \sum_{|\mathbf{i}|=k+d} \sum_{\mathbf{j}} (a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}) \left(f(\mathbf{x}_\mathbf{j}^\mathbf{i}) - A_{k+d-1,d}(f)(\mathbf{x}_\mathbf{j}^\mathbf{i}) \right), \tag{5}$$

$\Delta A_{d-1,d} = 0$, and $\mathbf{j}$ denoting the multi-index $(j_1, \ldots, j_d)$, $j_l = 1, \ldots, m_{i_l}^{\Delta}$, $l = 1, \ldots, d$, and the points $\mathbf{x}_\mathbf{j}^\mathbf{i} = (x_{j_1}^{i_1}, \ldots, x_{j_d}^{i_d})$ with $x_{j_l}^{i_l}$ denoting the j_lth element of $X_\Delta^{i_l} = X^{i_l} \setminus X^{i_l - 1}$, $X^0 = \emptyset$, and $m_{i_l}^{\Delta} = \# X_\Delta^{i_l}$. Furthermore, it is useful to define

$$w_\mathbf{j}^{k,\mathbf{i}} = f(\mathbf{x}_\mathbf{j}^\mathbf{i}) - A_{k+d-1,d}(f)(\mathbf{x}_\mathbf{j}^\mathbf{i}) \tag{6}$$

as the *hierarchical surpluses* [20] at level k with $k = |\mathbf{i}| - d$, since $\Delta A_{k+d,d}$ corrects $A_{k+d-1,d}$ at the points $\mathbf{x}_\mathbf{j}^\mathbf{i}$ to the actual value of $f(\mathbf{x}_\mathbf{j}^\mathbf{i})$. For continuous functions, the hierarchical surpluses tend to zero as the level k tends to infinity. This makes the hierarchical surpluses a natural candidate for error estimation and control. Specifically, we define

$$w_{\text{abs}}^n = \max w_\mathbf{j}^{n,\mathbf{i}} \quad \text{and} \quad w_{\text{rel}}^n = \max w_\mathbf{j}^{n,\mathbf{i}} / (\max f(\mathbf{x}_\mathbf{j}^\mathbf{i}) - \min f(\mathbf{x}_\mathbf{j}^\mathbf{i})) \tag{7}$$

as the estimated absolute and relative interpolation error of $A_{q,d}(f)$, respectively, recalling $n = q - d$.

From (3), we observe that we can increase the accuracy of the interpolation based on Smolyak's construction without having to discard previous results. Most importantly, to compute $A_{q,d}(f)$, only the function values at the sparse grid are needed. One should select the sets X^i in a nested fashion such that $X^i \subset X^{i+1}$ to obtain many recurring points with increasing q. With $X^0 = \emptyset$, $X_\Delta^i = X^i \setminus X^{i-1}$, $H_{q-1,d} = \emptyset$, one can construct the sparse grid according to

$$H_{q,d} = H_{(d-1,d)} \cup \underbrace{\bigcup_{|\mathbf{i}|=q} (X_\Delta^{i_1} \times \cdots \times X_\Delta^{i_d})}_{\Delta H_{q,d}}. \tag{8}$$

2.1.2 Accuracy of sparse grid interpolation

To guarantee a good approximation quality, the function f to recover must satisfy smoothness properties. In case of piecewise multilinear basis functions, an a priori error estimate can be obtained for a d-variate function f if bounded mixed derivatives

$$D^\beta f = \frac{\partial^{|\beta|} f}{\partial x_1^{\beta_1} \cdots x_d^{\beta_d}}, \tag{9}$$

with $\beta \in \mathbb{N}_0^d$, $|\beta| = \sum_{i=1}^{d} \beta_i$, and $\beta_1, \ldots, \beta_d \leq 2$, exist, i.e. $f \in F$ with

$$F := \left\{ f : [0,1]^d \to \mathbb{R},\ D^\beta f \in C^0\big([0,1]^d\big),\ \beta_1, \ldots, \beta_d \leq 2) \right\}. \tag{10}$$

According to [19], the order of the interpolation error in the maximum norm is then given by

$$\|f - A_{q,d}(f)\|_\infty = \mathcal{O}(N^{-2} \cdot (\log_2 N)^{3 \cdot (d-1)}), \tag{11}$$

with N denoting the number of sparse grid points of $H_{q,d}$. Piecewise multilinear approximation on a full grid with $\hat{N}$ grid points is much less efficient, i.e. the error in the maximum norm is of order $\mathcal{O}(\hat{N}^{-\frac{2}{d}})$.

Numerical tests [21] indicate that even for continuous functions, a satisfactory approximation is achieved, although the convergence rate is slower. Only discontinuous functions cannot be successfully approximated.

2.1.3 Definition of the grid and the basis functions

To implement sparse grid interpolation, it is required to select suitable basis functions and an according sparse grid structure. We use piecewise linear basis functions with sets X^i of equidistant nodes, as defined below.

$$m_i = \begin{cases} 1 & \text{if } i = 1, \\ 2^{i-1}+1 & \text{if } i > 1. \end{cases} \tag{12}$$

$$x_j^i = \begin{cases} (j-1)/(m_i-1) & \text{for } j = 1,\dots,m_i \text{ if } m_i > 1, \\ 0.5 & \text{for } j = 1 \text{ if } m_i = 1. \end{cases} \tag{13}$$

The basis functions are given by

$$a_1^1(x) = 1 \qquad \text{for } i = 1, \text{ and} \tag{14}$$

$$a_j^i(x) = \begin{cases} 1 - (m_i - 1)\left|x - x_j^i\right|, & \text{if } \left|x - x_j^i\right| < \frac{1}{(m_i-1)}, \\ 0, & \text{otherwise,} \end{cases} \tag{15}$$

for $i > 1$ and $j = 1, \dots, m_i$.

Note: We have written an easy-to-use MATLAB package for multilinear sparse grid interpolation [21], which is available for free at [22].

2.2 Part 2: Performing constrained fuzzy arithmetic

2.2.1 Discretization of the fuzzy numbers

Let $\tilde{p}_i$, $i = 1, \dots, d$ denote the fuzzy-valued input parameters with the membership functions $\mu_{\tilde{p}_i}(x_i)$, $\mu_i : \mathbb{R} \to [0,1]$. We obtain an arbitrarily accurate discrete formulation of the fuzzy numbers with $m \to \infty$ by decomposing them into sets of $m+1$ intervals [7]

$$P_i = \left\{ I_i^{(0)}, I_i^{(1)}, \dots, I_i^{(m)} \right\} \tag{16}$$

with the intervals of confidence

$$I_i^{(j)} = \left[a_i^{(j)}, b_i^{(j)} \right], \quad a_i^{(j)} \le b_i^{(j)}, \quad j = 0, \dots, m. \tag{17}$$

The intervals of confidence [4] are often called *α-cuts* with the α-level $\alpha = j/m \in [0,1]$. Thus, this discretization is usually referred to as α-cut representation. Note that by definition, a fuzzy number is convex, i.e. any fuzzy number can be decomposed into its α-cuts with the property

$$I_i^{(j+1)} \subseteq I_i^{(j)} \ \forall\, j \in 0, \dots, m-1. \tag{18}$$

In the following, we refer to $I_i^{(0)}$ as the *base* of the fuzzy number $\tilde{p}_i$. $I_i^{(0)}$ must be bounded, i.e. must be a closed interval. We define the Cartesian product of the bases as the *base box*

$$\Omega = I_1^{(0)} \times I_2^{(0)} \times \cdots \times I_d^{(0)}, \tag{19}$$

which is a d-dimensional interval vector.

2.2.2 Performing constrained fuzzy arithmetic on the interpolant

It is well known that the implementation of the extension principle leads to a global optimization problem that must be solved for each considered α-cut, i.e. in Step 2 of the algorithm, by replacing the objective function f with the sparse grid interpolant $A_{q,d}(f)$, one must solve

$$c^{(j)} = \min_{\mathbf{x} \in I_1^{(j)} \times \cdots \times I_d^{(j)}} A_{q,d}(f)(\mathbf{x}), \tag{20}$$

$$d^{(j)} = \max_{\mathbf{x} \in I_1^{(j)} \times \cdots \times I_d^{(j)}} A_{q,d}(f)(\mathbf{x}), \tag{21}$$

where $I_1^{(j)} \times \cdots \times I_d^{(j)}$ are d-dimensional boxes formed by the Cartesian product of the intervals of confidence $I_i^{(j)}$ of each α-cut with index j of the fuzzy input parameters $\tilde{p}_i$, $i = 1, \dots, d$, $j = 0, \dots, m$. We denote $J^{(j)} = [c^{(j)}, d^{(j)}]$ with the resulting intervals of confidence resulting in a set of intervals forming the discrete representation of the fuzzy result. Thus, if one uses a method that computes the global minimum and the global maximum of the surrogate function for each α cut exactly, one can eliminate the error resulting from Part 2. On the other hand, if one uses an approximative method to solve the global optimization problem, a second error source is introduced.

In general, one is free to select any type of fuzzy arithmetic algorithm for Part 2. Since the evaluation of $A_{q,d}(f)$ is explicitly given for single points in Eq. (3), fuzzy arithmetic algorithms requiring only real-valued function evaluations are applicable directly. We have successfully used the transformation method and a modified coordinate search algorithm for this approach. But it is also possible to implement interval arithmetic-based extensions of $A_{q,d}(f)$, i.e. implement inclusion functions $[A_{q,d}(f)]$ that take interval vectors (also called boxes) instead of single points as arguments. By doing so, we can apply branch-and-bound optimization algorithms based on interval arithmetic. The advantage of this approach is that the search of the interpolant is complete, i.e. it can be guaranteed that the global optimum is found.

Please refer to [14] for a detailed discussion of the topics of this section. In [14], we also give the entire procedure of sparse grid fuzzy arithmetic as a precise algorithm.

3 Applications to dynamic systems

3.1 Infectious disease modeling under uncertainty

The periodic outbreak of a disease can be modeled with delay differential equations [15]. Delay differential equations are equations with so-called time-lags τ of the form

$$y'(t) = f(t, y(t), y(t - \tau)), \tag{22}$$

i.e. the derivative of the solution depends also on the solution at a previous time $t - \tau$. We consider an enhanced Kermack-McKendrick model for epidemic modeling, as given in [15], where $y_1(t)$ is the susceptible, $y_2(t)$ is the infected, and $y_3(t)$ is the immunized portion of the population at a time t:

$$y_1'(t) = -y_1(t)y_2(t), \qquad y_2'(t) = y_1(t)y_2(t) - y_2(t), \qquad y_3'(t) = y_2(t). \tag{23}$$

By introducing two time lags τ_1 and τ_2, the classical model becomes capable of modeling the periodic outbreak of a disease. The enhanced model is given by

$$\begin{aligned} y_1'(t) &= -y_1(t)y_2(t - \tau_2) + y_2(t - \tau_1) \\ y_2'(t) &= y_1(t)y_2(t - \tau_2) - y_2(t) \\ y_3'(t) &= y_2(t) - y_2(t - \tau_1), \end{aligned} \tag{24}$$

with τ_2 denoting the incubation period, and τ_1 denoting the time after which the immunized ones of the population become susceptible again. In this simple model, all rate constants are equal to one. The solution is given in [15, Figure 17.6] for the following time-lags and initial conditions: $\tau_1 = 10$, $\tau_2 = 1$, $y_1(t_0) = 5$, $y_2(t_0) = 0.1$, and $y_3(t_0) = 1$ for $t_0 \le 0$. The considered time span was $t \in [0, 40]$. Since y_1 and y_2 do not depend on y_3, we will neglect y_3 in the following. Alternatively to integrating the differential equation given in Eq. (24), y_3 can be computed by

$$y_3(t) = y_{\text{total}} - (y_2(t) + y_1(t)), \tag{25}$$

where y_{total} is the total population ($y_{\text{total}} = y_1(t_0) + y_2(t_0) + y_3(t_0)$), which remains constant.

We can interpret the above model (24) as a nonlinear function with five input parameters and two output parameters (omitting y_3), i.e.

$$(y_1(t), y_2(t)) = f_{\text{dde}}(t, \tau_1, \tau_2, y_1(t_0), y_2(t_0)), \tag{26}$$

$t \in [0, t_1]$, since its behavior is uniquely determined for a given set of time-lags and initial conditions. To compute this function, a numerical algorithm is applied that discretizes and integrates the model equations with respect to the time t . Using a predefined number s of equally-spaced discretization steps $h = t_1/s$, a start time $t_0 = 0$, and an end time t_1, we arrive at the time-discrete function

$$(\hat{y}_{1,1}, \dots, \hat{y}_{1,s}, \hat{y}_{2,1}, \dots, \hat{y}_{2,s}) = f^h_{\text{dde}}(\tau_1, \tau_2, \hat{y}_{1,0}, \hat{y}_{2,0}, t_1, s), \tag{27}$$

where $\hat{y}_{i,k} = y_i(hk)$, $k \in \{0, \dots, s\}$, $i \in \{1, 2\}$.

We now introduce uncertainty to the model. We assume the time-lags in Eq. (27) to be uncertain, i.e. we replace τ_1 and τ_2 by the fuzzy-valued parameters $\tilde{\tau}_1$ and $\tilde{\tau}_2$, respectively.

Furthermore, we replace the initial conditions in Eq. (27) by uncertain parameters, i.e. the initial susceptible portion of the population $\hat{y}_{1,0}$ by the fuzzy-valued parameter $\tilde{y}_{1,0}$ and the initial infected portion of the population $\hat{y}_{2,0}$ by $\tilde{y}_{2,0}$. Consequently, the output parameters become fuzzy as well. The resulting fuzzy-parameterized model function is given by

$$(\tilde{y}_{1,1}, \ldots, \tilde{y}_{1,s}, \tilde{y}_{2,1}, \ldots, \tilde{y}_{2,s}) = f^h_{\text{dde}}(\tilde{\tau}_1, \tilde{\tau}_2, \tilde{y}_{1,0}, \tilde{y}_{2,0}, t_1, s). \tag{28}$$

In our numerical example, we use fuzzy numbers with membership functions of quasi-Gaussian shape [7] according to

$$\mu(x) = \begin{cases} \exp\left(-\frac{(x-\overline{m})^2}{2\sigma^2}\right), & \text{if } |x - \overline{m}| \leq 3\sigma, \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

The base variations are $\pm 10\%$ for the time-lags, and $\pm 5\%$ for the initial conditions. The peak values of the fuzzy numbers correspond to the values from [15]. The membership functions $\mu_1, \ldots, \mu_4$ of the fuzzy parameters $\tilde{p}_1, \ldots, \tilde{p}_4$ are thus given the parameters

$$\overline{m}_1 = 10 \qquad \overline{m}_2 = 1 \qquad \overline{m}_3 = 5 \qquad \overline{m}_4 = 0.1 \tag{30}$$

$$\sigma_1 = 3^{-1} \qquad \sigma_2 = 30^{-1} \qquad \sigma_3 = 12^{-1} \qquad \sigma_4 = 600^{-1}. \tag{31}$$

Since the objective function has multiple output arguments, the sparse grid surrogate function $\mathbf{A}_{q,d}(f^h_{\text{dde}})$ must have an according number of outputs as well (indicated by the bold-faced capital $\mathbf{A}$). Formally, the sparse grid interpolation algorithm is applied separately with respect to each output parameter, but since the same sparse grid points are used, the number of function evaluations remains constant (it is not affected). The memory and time complexity to compute the interpolant grows linearly with each additional output parameter. Our MATLAB implementation of piecewise multilinear sparse grid interpolation [23] is capable of handling multiple function output arguments. With the given base variations, the objective function f^h_{dde} is evaluated at the sparse grid points for the base box $\Omega = [9, 11] \times [0.9, 1.1] \times [4.75, 5.25] \times [0.095, 0.105]$. We have used the standard solver `dde23` by Shampine and Thompson [16] to do this, with the solver error tolerances $\epsilon_{\text{rel}} = 10^{-5}$ and $\epsilon_{\text{rel}} = 10^{-9}$. With `dde23`, the discretization of the time is not equally-spaced by default, however, with the `deval` function provided with `dde23`, the solution vectors can be made equidistant. This is necessary, since we require the different function evaluations to produce solution vectors of equal length and time step-size, such that we can apply the sparse grid algorithm for each component of the solution vectors.

We have used $s = 80$ discretization steps, and the sparse grid interpolation depths $n = 1, 3, 5$ (recalling $n = q - d$) to compute three sparse grid interpolants $A_{q,d}(f^h_{\text{dde}})$ of increasing accuracy in Part 1 of the algorithm for fuzzy arithmetic on sparse grids, and $m + 1 = 51$ α-cuts to generate the plots. The results of the simulation are summarized in Figure 1. A maximum relative error of the fuzzy results of less than 2% is achieved for the interpolation depth $n = 5$ (1105 function evaluations), but even for $n = 3$ (137 evaluations), an average error of less than 3% already provides a very good approximation of the solution.

To assess the accuracy of the results in greater detail, we present convergence histories for the maximum relative error $\epsilon_{\max}$ and the average relative error ϵ_{avg} of the fuzzy-valued results $\tilde{y}_1$ and $\tilde{y}_2$ over all time steps $0, \ldots, s$ in Figure 2. Since the exact results are not known, we have used the solutions obtained for $n = 8$ as references to compute the relative errors. In the plots, $w^n_{\max,\text{rel}}$ represents the maximum over the estimated relative errors of the interpolant $\mathbf{A}_{q,d}$ according to Eq. (7) with respect to the outputs $\tilde{y}_1$ and $\tilde{y}_2$, again over all time steps. One can observe that the estimated interpolation error gives a good indication of the accuracy of the fuzzy-valued results, provided that in Part 2 of the algorithm, the global optimization problems (20) are solved accurately.

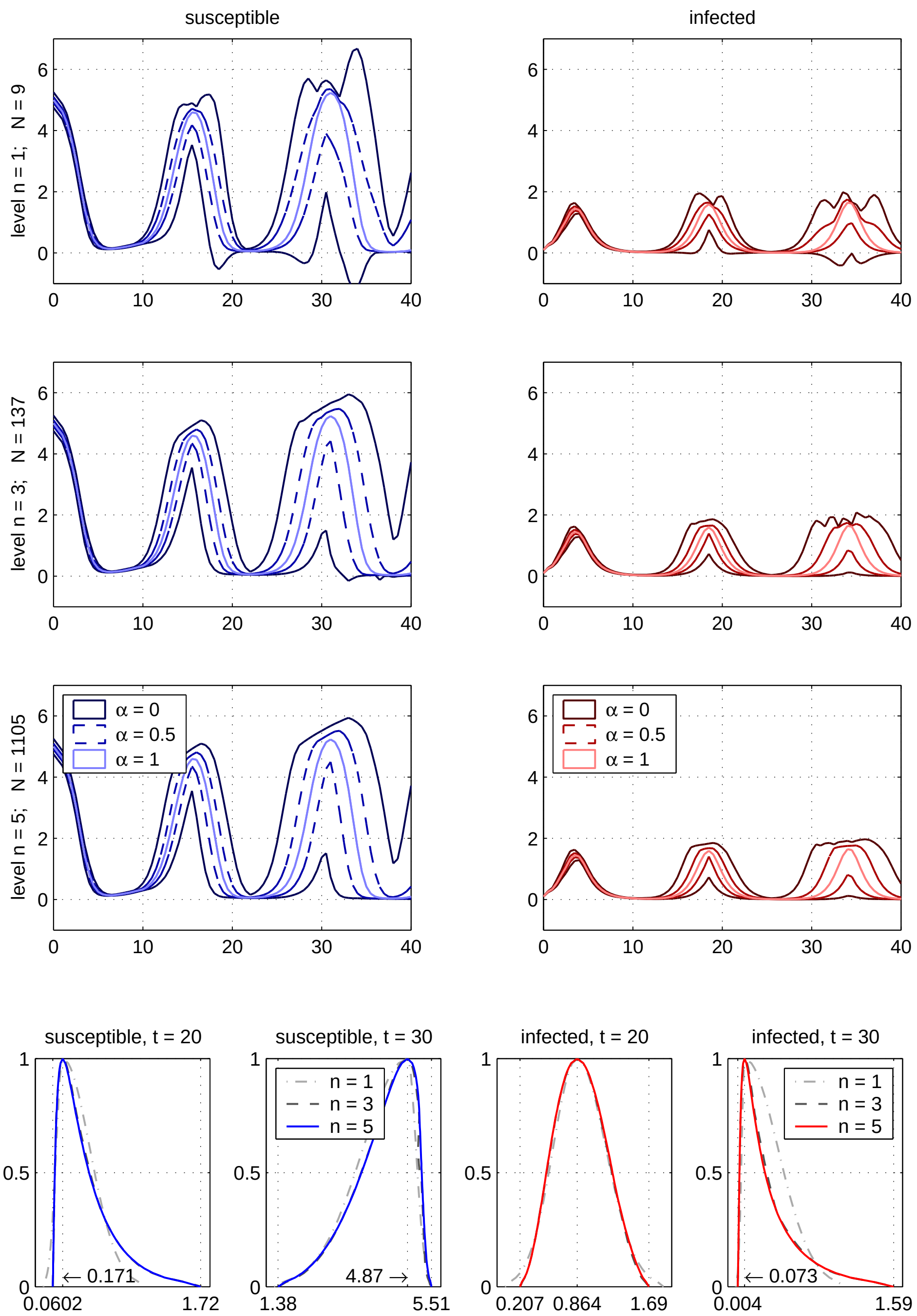

Figure 1: Fuzzy-valued results $\tilde{y}_1$ (susceptible) and $\tilde{y}_2$ (infected) for the solution of the Kermack-McKendrick model.

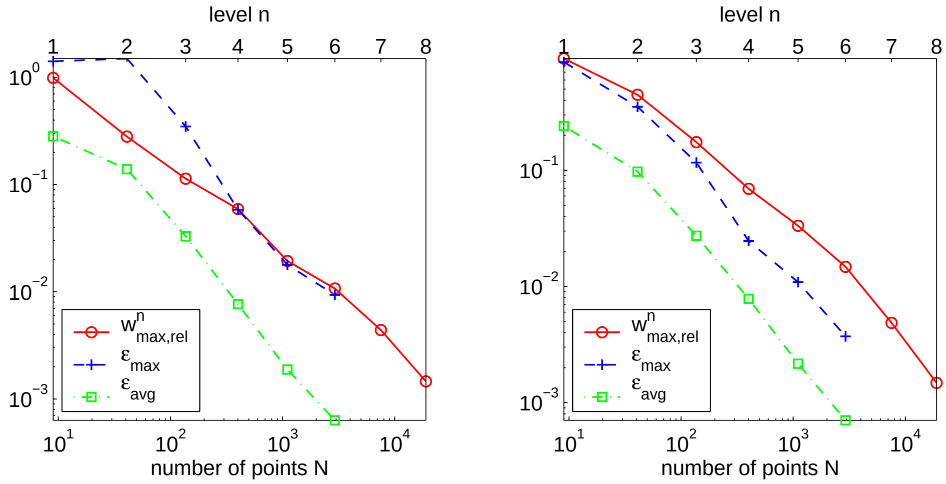

Figure 2: Relative error plots, left: $\tilde{y}_1$ (susceptible), right: $\tilde{y}_2$ (infected).

3.2 Computation of a multibody mechanism under uncertainty

Andrews' squeezer mechanism, which is shown in Fig. 3, is a popular test example for multibody dynamics. The plane mechanism consists of seven rigid bodies $B_1, \ldots, B_7$ connected by frictionless joints and a spring c. The mechanism is fixed at the origin of the coordinate system O and at the points A, B, and C. The positions of the bodies are described by the seven angles β, Θ, γ, Φ, δ, Ω, and ε as shown in the figure. Since there are three kinematic loops and correspondingly six algebraic constraints, the problem is actually a single degree of freedom mechanism. However, for the numerical solution it is more convenient to keep the equations of motion for the seven angles and the six algebraic constraints, and to treat the problem as a system of differential algebraic equations. A complete description including geometrical, inertia and stiffness properties can be found in [17].

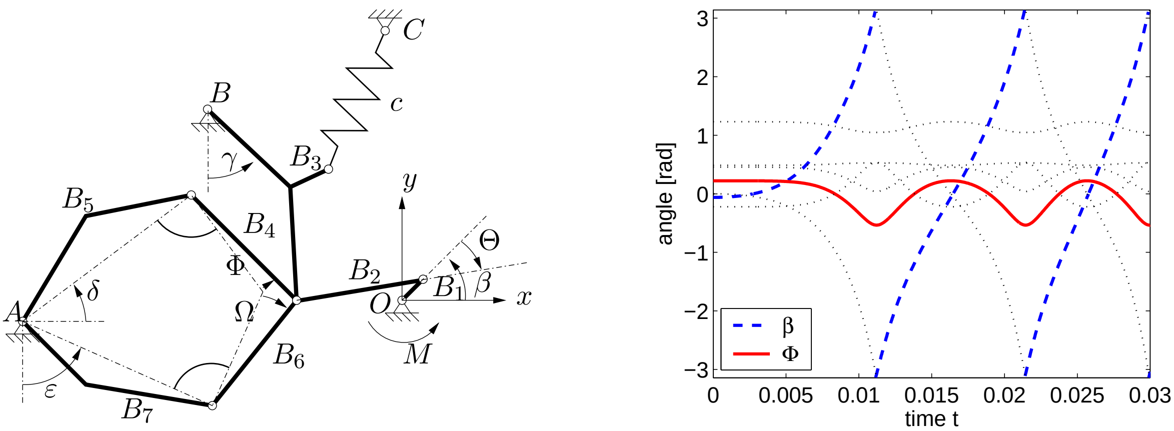

Figure 3: Andrews' squeezer mechanism, left: system, right: solution.

The mechanism is driven by a constant torque M at point O starting the system from rest at $\Theta = 0$, from which condition the complete initial state of the system can be calculated by a Newton iteration using just the geometrical data. Time integration of the DAE system in the interval $t \in [0, 0.03]$ is then performed using the standard MATLAB solver `ode45`, where a relative error tolerance of $\epsilon_{\mathrm{rel}} = 10^{-6}$ has been enforced. Output has been requested at $s = 60$ equal time steps with $h = 5 \cdot 10^{-4}$. The solution for the data given in [17] is shown (mod 2π) in Fig. 3, where the solutions for β and Φ have been highlighted for later reference.

We now consider uncertain values for the positions of the three fixed points A at (x_A, y_A), B at (x_B, y_B) and O at (x_O, y_O), while keeping the other data crisp. This assumption renders the initial state as well as the right hand side of the DAE system uncertain. The six coordinate values are assumed to be symmetric fuzzy numbers with triangular membership functions [7] with peak values taken from [17] and absolute base variations of ± 0.001.

If the general transformation method [7] is used to evaluate the fuzzy problem, the total number of system evaluations is given by $N = \sum_{k=1}^{m+1} k^d$, where d is the number of fuzzy parameters and $m + 1$ is the number of α-cuts. Thus, a direct fuzzy simulation with $d = 6$ fuzzy parameters and 11 α-cuts requires $N = 3\,749\,966$ evaluations of the system, which is a prohibitively high number. However, the same problem can be easily solved using sparse grid interpolation, where an estimated maximum relative interpolation error (over both fuzzy results for β and Φ at all time steps) of $\epsilon_{\mathrm{est}} < 0.01$ is reached at an interpolation depth of $n = 5$ at a cost of only $N = 4\,865$ evaluations. Figure 5 shows the fuzzy result for the angles β and Φ and interpolation depths $n = 1, 3, 5$. The results are shown as contour plots over the time history and as membership function plots for $t = 0.02$ and $t = 0.03$, respectively. The fuzzy results were obtained using 11 α-cuts and the coordinate search algorithm described in [14], although in very few cases, we had to manually adjust the optimization part to correctly compute the ranges of the interpolant (by using the more expensive interval-based complete search). Nevertheless, this did not affect the number of required function evaluations.

As expected, the peak values of the fuzzy results are equal to the corresponding results of the crisp solution shown in Fig. 3. The non-linear input-output relationship of the model becomes manifest in the non-symmetric membership functions of the outputs for symmetric inputs. Analogously to the previously discussed model, we have also included the convergence histories of the interpolation error and the maximum and average error of the fuzzy-valued results over all time steps in Fig. 4.

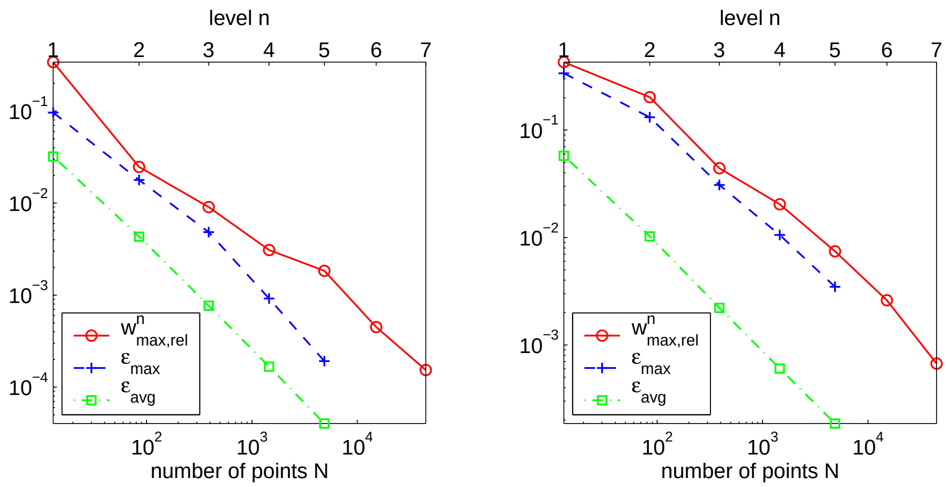

Figure 4: Relative error plots, left: $\tilde{\beta}$, right: $\tilde{\Phi}$.

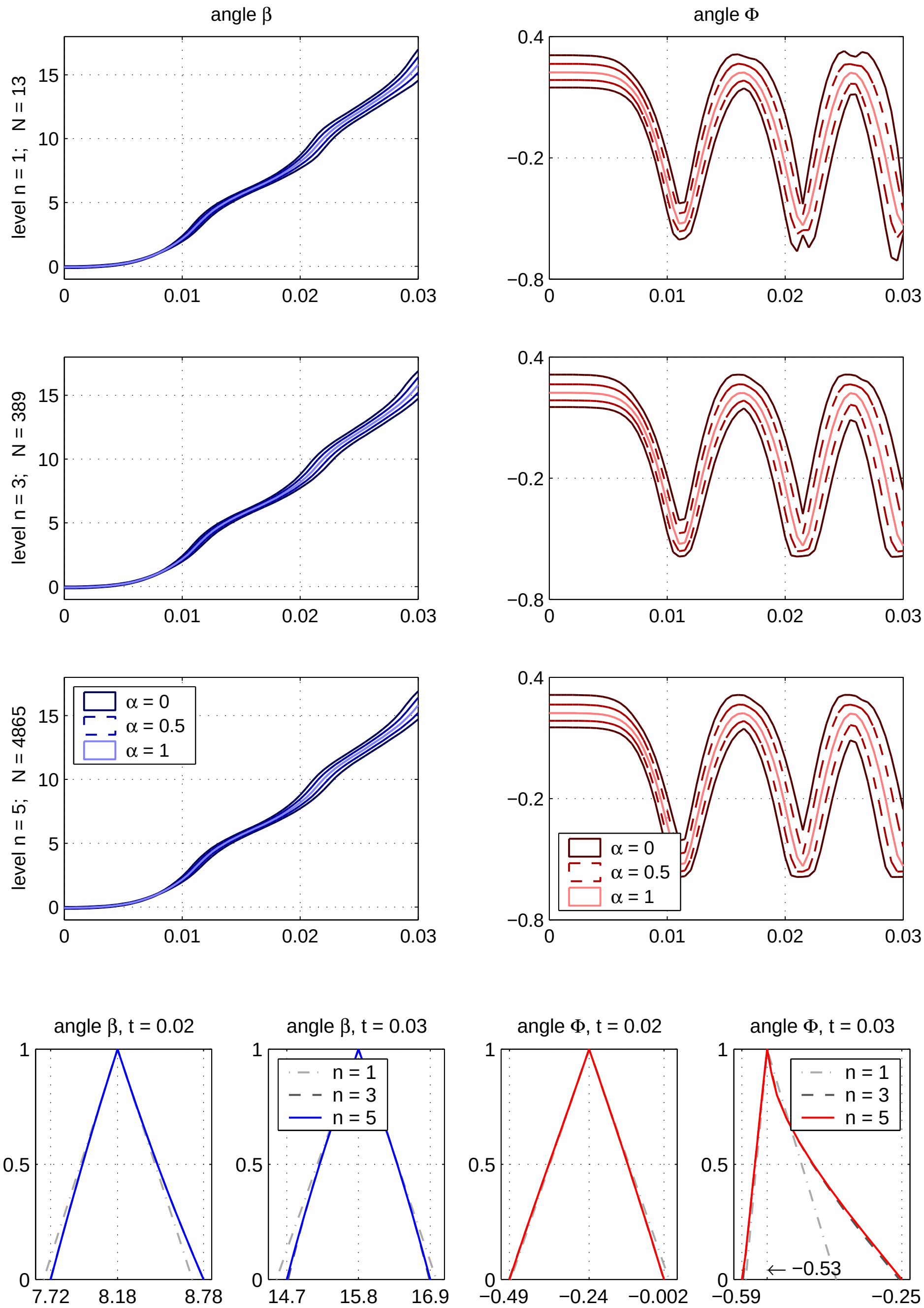

Figure 5: Fuzzy-valued results $\tilde{\beta}$ and $\tilde{\Phi}$.

4 Conclusions

Sparse grid fuzzy arithmetic provides a powerful tool to introduce uncertainty into dynamic systems provided that the underlying model is sufficiently smooth. By using a sparse grid interpolant as a surrogate function to the objective model, the number of function evaluations is significantly reduced compared to a direct method operating on the full grid, such as the transformation method. Compared to fuzzy arithmetic based on interval arithmetic, we emphasize that only mere real-valued function evaluations are needed for the sparse grid approach. Thus, the method is applicable in general without any need to modify the original, crisp model. The high-order convergence rate of sparse grid interpolation guarantees a quick convergence to the actual solution. Furthermore, the hierarchical structure of the sparse grid interpolation algorithm permits to construct increasingly accurate solutions without wasting previous function evaluations, while the actual estimated accuracy can be monitored by the hierarchical surpluses.

An additional difficulty when dealing with dynamic systems is the high number of outputs when multiple time steps are considered. Since the grid node coordinates are known a priori, on can construct the interpolant for many output variables at once without requiring additional function evaluations.

References

[1] L. A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.

[2] L. A. Zadeh, Fuzzy logic and approximate reasoning, Synthese 30 (1965) 407–428.

[3] D. Dubois, H. Prade, Fuzzy Sets and Systems, Theory and Applications, Vol. 144 of Mathematics in Science and Engineering, Academic Press, Inc., New York, 1980.

[4] A. Kaufmann, M. M. Gupta, Introduction to Fuzzy Arithmetic, Van Nostrand Reinhold Co., New York, 1991.

[5] R. E. Giachetti, R. E. Young, A parametric representation of fuzzy numbers and their arithmetic operators, Fuzzy Sets and Systems 91 (2) (1997) 185–202.

[6] W. Dong, H. C. Shah, Vertex method for computing functions of fuzzy variables, Fuzzy Sets and Systems 24 (1987) 65–78.

[7] M. Hanss, The transformation method for the simulation and analysis of systems with uncertain parameters, Fuzzy Sets and Systems 130 (3) (2002) 277–289.

[8] G. J. Klir, Fuzzy arithmetic with requisite constraints, Fuzzy Sets and Systems 91 (1997) 165–175.

[9] E. H. Hansen, Global Optimization Using Interval Analysis, Marcel Dekker, New York, NY, 1992.

[10] L. Jaulin, M. Kieffer, O. Didrit, É. Walter, Applied Interval Analysis, Springer, London, Great Britain, 2001.

[11] R. E. Moore, Interval Analysis, Prentice Hall, Englewood Cliffs, NJ, 1966.

[12] M. Hanss, The extended transformation method for the simulation and analysis of fuzzy-parameterized models, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 11 (6) (2003) 711–727.

[13] A. Klimke, An efficient implementation of the transformation method of fuzzy arithmetic, in: E. Walker (Ed.), Proceedings of NAFIPS 2003, Chicago, IL, 2003, pp. 468–473.

[14] A. Klimke, B. Wohlmuth, Computing expensive multivariate functions of fuzzy numbers using sparse grids, IANS preprint 2004/002, Tech. rep., University of Stuttgart (2004).
URL `http://preprints.ians.uni-stuttgart.de`

[15] E. Hairer, S. P. Nørsett, G. Wanner, Solving Ordinary Differential Equations I, Vol. 8 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1993.

[16] L. F. Shampine, S. Thompson, Solving DDEs in MATLAB, Appl. Numer. Math. 37 (4) (2001) 441–458.

[17] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II, Vol. 14 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1996.

[18] W. Schiehlen (Ed.), Multibody Systems Handbook, Springer, Berlin, 1990.

[19] V. Barthelmann, E. Novak, K. Ritter, High dimensional polynomial interpolation on sparse grids, Adv. Comput. Math. 12 (4) (2000) 273–288.

[20] H.-J. Bungartz, Finite Elements of Higher Order on Sparse Grids, Shaker Verlag, Aachen, 1998.

[21] A. Klimke, Piecewise multilinear sparse grid interpolation in MATLAB, IANS report 2003/019, Tech. rep., University of Stuttgart (2003).
URL `http://preprints.ians.uni-stuttgart.de`

[22] Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Scientific/Educational Matlab Database (2002).
URL `http://matlabdb.mathematik.uni-stuttgart.de`

[23] A. Klimke, An efficient implementation of the transformation method of fuzzy arithmetic (extended preprint version), IANS preprint 2003/009, Tech. rep., University of Stuttgart (2003).
URL `http://preprints.ians.uni-stuttgart.de`

Andreas Klimke

Institute of Applied Analysis and Numerical Simulation
University of Stuttgart
Pfaffenwaldring 57
70569 Stuttgart, Germany

E-Mail: `klimke@ians.uni-stuttgart.de`

Kai Willner

Institute of Applied Mechanics
University of Erlangen-Nuernberg
Egerlandstrasse 5
91058 Erlangen, Germany

E-Mail: `willner@ltm.uni-erlangen.de`

Barbara Wohlmuth

Institute of Applied Analysis and Numerical Simulation
University of Stuttgart
Pfaffenwaldring 57
70569 Stuttgart, Germany

E-Mail: `wohlmuth@ians.uni-stuttgart.de`

Erschienene Preprints ab Nummer 2004/001

Komplette Liste: `http://preprints.ians.uni-stuttgart.de`

2004/001 *Geis, W., Mishuris, G., Sändig, A.-M.*: 3D and 2D asymptotic models for piezoelectric stack actuators with thin metal inclusions

2004/002 *Klimke, A., Wohlmuth, B.*: Computing expensive multivariate functions of fuzzy numbers using sparse grids

2004/003 *Klimke, A., Willner, K., Wohlmuth, B.*: Uncertainty modeling using efficient fuzzy arithmetic based on sparse grids: applications to dynamic systems