

Article

Sparse Grid Adaptive Interpolation in Problems of Modeling Dynamic Systems with Interval Parameters

Alexander Yu Morozov , Andrey A. Zhuravlev and Dmitry L. Reviznikov

Federal Research Center “Computer Science and Control” of Russian Academy of Sciences (FRC CSC RAS), 119333 Moscow, Russia; zhuravlyow.andrei@yandex.ru (A.A.Z.); reviznikov@mai.ru (D.L.R.)

* Correspondence: morozov@infway.ru

Abstract: The paper is concerned with the issues of modeling dynamic systems with interval parameters. In previous works, the authors proposed an adaptive interpolation algorithm for solving interval problems; the essence of the algorithm is the dynamic construction of a piecewise polynomial function that interpolates the solution of the problem with a given accuracy. The main problem of applying the algorithm is related to the curse of dimension, i.e., exponential complexity relative to the number of interval uncertainties in parameters. The main objective of this work is to apply the previously proposed adaptive interpolation algorithm to dynamic systems with a large number of interval parameters. In order to reduce the computational complexity of the algorithm, the authors propose using adaptive sparse grids. This article introduces a novelty approach of applying sparse grids to problems with interval uncertainties. The efficiency of the proposed approach has been demonstrated on representative interval problems of nonlinear dynamics and computational materials science.

Keywords: adaptive interpolation algorithm; interval ordinary differential equations (ODEs); sparse grids; hierarchical basis; multidimensional interpolation; high dimensions; molecular dynamics modeling



Citation: Morozov, A.Y.; Zhuravlev, A.A.; Reviznikov, D.L. Sparse Grid Adaptive Interpolation in Problems of Modeling Dynamic Systems with Interval Parameters. *Mathematics* **2021**, *9*, 298. <https://doi.org/10.3390/math9040298>

Academic Editor: Mikhail Posypkin

Received: 18 December 2020

Accepted: 28 January 2021

Published: 3 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Problems related to inaccurately specified data arise in many modern fields of science and technology. When applied to non-stationary processes, they are often formulated as dynamic systems with interval parameters. The result of solving such problems is an interval estimate of the set of possible system states depending on the uncertainties in the parameters. Basic methods of interval analysis are presented in books [1–5]. There are known methods based on the representation of a set of solutions through geometric primitives: parallelepipeds and ellipses [6,7], methods based on symbolic computation [8,9], as well as stochastic methods [10], such as Monte-Carlo methods. Methods based on classical interval arithmetic are subject to the so-called wrap effect [1], which manifests itself in an unlimited increase in the width of the obtained interval estimates of solutions. This effect arises due to the replacement of the exact form of the set of solutions by a simpler form, and for iterative methods, the divergence of intervals' boundaries is often exponential. Existing methods that are not subject to this effect, or weakly susceptible to it, often have exponential complexity with respect to the number of interval parameters. It concerns symbolic methods operating in series, Monte-Carlo methods, and the adaptive interpolation algorithm [11]. Therefore, there is a need for efficient approaches to reduce the computational complexity of methods that are not affected by the wrapping effect.

While solving a considered class of problems, the main idea is to construct an explicit dependence of the solution to the corresponding non-interval problem on the point values of the interval parameters. If such dependence is available, finding an interval estimate would be reduced to solving a certain number of constrained optimization problems for explicitly given functions.

Papers [11–14] describe the adaptive interpolation algorithm in detail. The essence of the algorithm is the dynamic construction of a piecewise polynomial function that interpolates the solution of the problem with a given accuracy. The theoretical basis of the algorithm is given in References [11,12]. The algorithm has a number of essential features: it is not subject to the wrapping effect [11]; efficiently parallelizes on GPUs; able to simulate rigid systems [13]; determines the presence of bifurcations and chaos in the system [14]. It has been tested on applied problems of chemical kinetics [13], gas dynamics and celestial mechanics, and complex dynamics with bifurcations and chaos [14]. Nevertheless, there is some drawback. The algorithm uses multidimensional interpolation on a regular grid, which requires $(p + 1)^d$ nodes, where p —a polynomial degree for each dimension and d —the number of dimensions. With a large number of interval parameters, the application of the algorithm becomes difficult. However, a typical situation is when the degree of influence of different parameters and their combinations on a solution can differ significantly; therefore, it naturally follows to use approaches that take into account these features and, as a consequence, reduce the computational complexity.

Within the framework of modeling methods for dynamical systems with interval parameters, it is worth noting the work [15], which describes a method based on the polynomial approximation of the solution, which requires points in the sample less than $(p + 1)^d$. This method has been successfully applied to the problem of modeling a rotating system with both random and interval variables [16]. This class of problems is significant from an applied point of view.

The curse of dimension, that is, the exponential growth of the number of calculations, is a critical problem. Typically, this situation arises when studying multidimensional functions presented in the form of a black box. The general tactic for reducing computational complexity is to determine and take into account the features of the function under consideration.

Sparse grids [17] are numerical methods for representing, integrating, or interpolating multidimensional functions based on a hierarchical basis [18,19] and reducing the curse of dimension. This approach was first presented by the Russian mathematician Smolyak in 1963. Classic sparse grids result from computational cost optimization for approximating functions with bounded mixed derivatives [20]. This fact is important since it imposes certain restrictions on the solution's dependence on interval parameters. Interpolation using sparse grids requires significantly fewer nodes than standard full grid interpolation.

There are many works devoted to sparse grids [21–24]. Reference [21] gives an initial introduction to sparse grids and the technique of combining them. It provides a program code in the Python programming language. In Reference [22], some parallelization issues are considered; Reference [23] provides an overview of the foundations and applications of sparse grids, with particular attention to the solution of partial differential equations.

The behavior of the solution to the ordinary differential equations (ODE) system can differ significantly depending on the parameters and initial conditions. Adaptive grids can drastically reduce computational costs by condensing nodes in regions with strong solution dependence on parameters and rarefaction in areas with weak dependence. Besides such adaptation, additional properties of the solution can be taken into account using sparse grids. This approach is effective when the interpolated function has a weak dependence on subsets of variables. For example, if the solution to an ODE system can be represented as a linear combination of functions from certain subsets of parameters and initial conditions, then it is sufficient to consider only the corresponding subsets and construct a grid only from them. Sparse grids are especially effective in multidimensional problems and can significantly reduce computational costs.

The main problem is the high computational costs when solving problems with uncertainties. The main goal of this work is to apply the previously proposed adaptive interpolation algorithm to the case of dynamical systems with a large number of interval parameters. The novelty lies in the application of sparse grids to problems with interval uncertainties, including problems of molecular dynamics.

The research methodology is based on methods of mathematical modeling, computational mathematics, and differential calculus. The statement of the problem is formulated in the form of the Cauchy problem for a system of ordinary differential equations with interval parameters. The method is tested on a representative set of problems.

The following sections give a description of the adaptive interpolation algorithm on sparse grids, present the results of testing the algorithm on a number of model problems of nonlinear dynamics, and solve an important problem of computational materials science, namely the determination of an interval stress tensor based on molecular dynamics modeling.

2. Algorithm for Adaptive Interpolation Using Sparse Grids

Dynamic systems with uncertainties in parameters arise in many practical areas. Traditionally, interval problems for dynamic systems are formulated in the form of the Cauchy problem for a system of ordinary differential equations (ODE) with interval initial conditions or parameters. It is necessary to obtain an interval estimate of the solution based on interval values of the parameters.

Consider the Cauchy problem with m interval initial conditions:

$$\begin{cases} \frac{dy_i(t)}{dt} = f_i(y_1(t), y_2(t), \dots, y_n(t)), & 1 \leq i \leq n, \\ y_i(t_0) \in [\underline{y}_i^0, \overline{y}_i^0], & 1 \leq i \leq m, \\ y_i(t_0) = y_i^0, & m+1 \leq i \leq n, \\ t \in [t_0, t_N]. \end{cases} \quad (1)$$

Hereinafter, the underline denotes the lower bound of the interval, and the overline—the upper bound of the interval.

If the ODE system is not autonomous or contains interval parameters, then fictitious equations are added to the system so that it would take the form of system Equation (1). A vector function $\mathbf{f} = (f_1, f_2, \dots, f_n)^T$ meets all conditions ensuring the uniqueness and existence of a solution for all $y_i(t_0) \in [\underline{y}_i^0, \overline{y}_i^0]$, $1 \leq i \leq m$.

The goal is, for each moment of time t_k , to construct a piecewise polynomial vector function $\mathbf{P}^k(y_1^0, y_2^0, \dots, y_m^0)$, where $y_i(t_0) \in [\underline{y}_i^0, \overline{y}_i^0]$, $1 \leq i \leq m$, which interpolates the dependence of the solution on the interval parameters with controlled accuracy. If the function \mathbf{P}^k is available, finding the interval estimate of the solution (finding the left and right boundaries of the intervals) should be reduced to solving constrained optimization problems for an explicitly given function.

Suppose that the solution to $\mathbf{y}^k(y_1^0, y_2^0, \dots, y_m^0)$ is known at the moment of time t_k , where $y_i^0 \in [\underline{y}_i^0, \overline{y}_i^0]$, $1 \leq i \leq m$. An adaptive sparse grid is constructed for the set formed by the interval initial conditions. Each grid point has a corresponding solution to the noninterval system (1) at pointwise values of interval parameters that correspond to the position of a node. To obtain an interval solution at the moment of time t_{k+1} , the transfer of all non-interval solutions contained in the grid nodes to the time layer $(k+1)$ is performed, followed by the adaptation of the grid and the construction of an interpolation polynomial \mathbf{P}^{k+1} .

A short description of sparse grid interpolation according to the works [20,21] is given below.

Consider the interpolation of a smooth function $f(x)$ of one variable on the unit interval $[0, 1]$. For the sake of simplicity, it is assumed that the function is equal to zero at the boundary points: $f(0) = f(1) = 0$.

The interpolation is performed on a piecewise linear hierarchical basis using the hat function:

$$\varphi(x) = \begin{cases} 1 - |x|, & x \in [-1, 1] \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

Define a set of grids G_l on a unit interval $[0, 1]$, where l is the level that determines the grid width $h_l = 2^{-l}$. The grid points $x_{l,i}$ are given as:

$$x_{l,i} = i \cdot h_l, 0 \leq i \leq 2^l.$$

Families of basis functions $\varphi_{l,i}(x)$ are generated based on the obtained sets of points, using the stretching and transfer of the hat Equation (2):

$$\varphi_{l,i}(x) = \varphi\left(\frac{x - i \cdot h_l}{h_l}\right). \quad (3)$$

A nodal basis is formed for each given l of Equation (3). Here, the common piecewise linear interpolation (Figure 1) is applied, and the corresponding polynomial is written as follows:

$$P(x) = \sum_{i=1}^{2^l-1} a_{l,i} \varphi_{l,i}(x), \quad a_{l,i} = f(x_{l,i}). \quad (4)$$

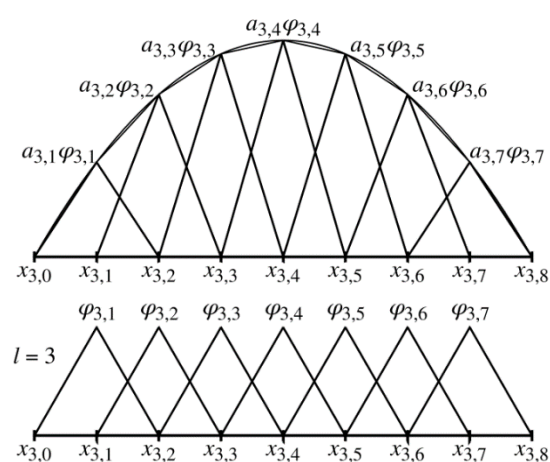


Figure 1. Interpolation on a nodal basis.

Let us make the transition to a hierarchical basis (Figure 2). The basis functions given by Equation (3) is expressed with even level indices k in terms of the basis level functions $(k-1)$:

$$\varphi_{k,2i}(x) = \varphi_{k-1,i}(x) - \frac{1}{2}(\varphi_{k,2i-1}(x) + \varphi_{k,2i+1}(x)), \quad 1 \leq i \leq 2^{k-1} - 1.$$

In this case, the interpolation polynomial given by Equation (4) takes the following form:

$$P(x) = \sum_{k=1}^l \sum_{\substack{i=1, \\ i \text{ odd}}}^{2^k-1} a_{k,i} \varphi_{k,i}(x), \quad a_{k,i} = f(x_{k,i}) - \frac{1}{2}(f(x_{k,i-1}) + f(x_{k,i+1})) \quad (5)$$

Next, consider the multidimensional interpolation of a smooth function $f(x_1, x_2, \dots, x_d)$ using d —dimensional unit cube $\Omega = [0, 1]^d$, provided that $f|_{\partial\Omega} = 0$. A multidimensional basis is constructed by the direct product of hierarchical one-dimensional bases:

$$\varphi_{1,i}(\mathbf{x}) = \prod_{j=1}^d \varphi_{l_j,i_j}(x_j), \quad 1 \leq i_j \leq 2^{l_j} - 1, \quad i_j \text{ odd},$$

where $\mathbf{l} = (l_1, l_2, \dots, l_d)$ are the levels of the corresponding one-dimensional grids, $\mathbf{i} = (i_1, i_2, \dots, i_d)$ is the basis function multi-index, $\mathbf{x} = (x_1, x_2, \dots, x_d)$. If $\sum_{j=1}^d l_j \leq n + d - 1$, there is a sparse grid of n level; if $\max_{j=1,d} (l_j) \leq n$, there exists a complete grid (Figure 3). The number of nodes in a sparse grid is estimated as $O(p(\log_2 p)^{d-1})$, and the interpolation error is estimated as $O(h_n^2(\log_2 p)^{d-1})$; for a full grid the respective number of nodes is $O(p^d)$, and the error is $O(h_n^2)$, where $p = 2^n - 1$ is the number of nodes in each dimension [20].

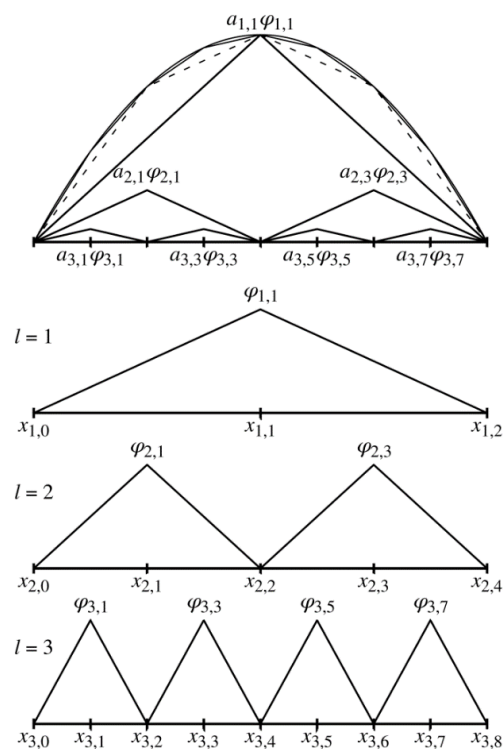


Figure 2. Interpolation on a hierarchical basis.

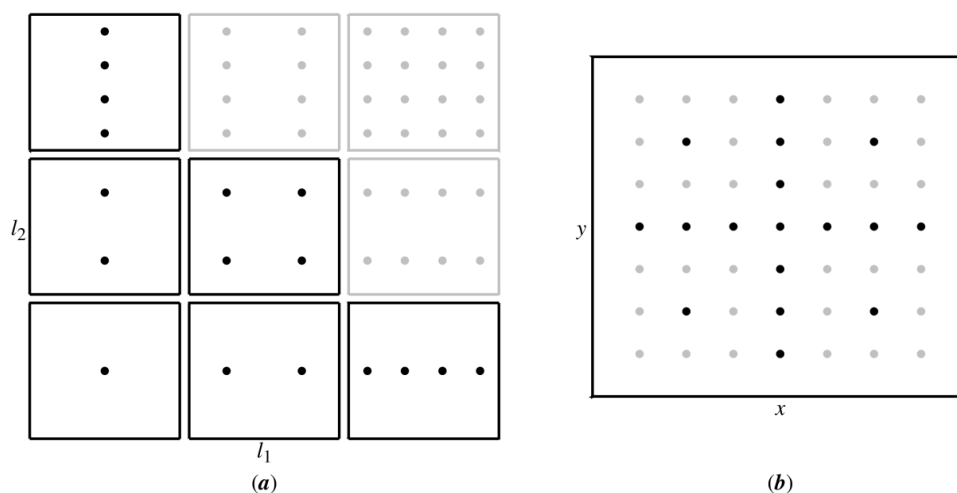


Figure 3. Sparse grid of the third level: black dots—sparse grid; black and grey dots—full grid: (a) sets of points corresponding to basis functions of the same level; (b) combining all points into one grid.

The interpolation polynomial is written as follows:

$$P(\mathbf{x}) = \sum_{\mathbf{l}, \mathbf{i}} a_{\mathbf{l}, \mathbf{i}} \varphi_{\mathbf{l}, \mathbf{i}}(\mathbf{x}), \|\mathbf{l}\|_1 \leq n + d - 1, 1 \leq i_j \leq 2^{l_j} - 1, i_j \text{ odd} \quad (6)$$

where

$$a_{\mathbf{l}, \mathbf{i}} = \sum_{\Delta_1, \dots, \Delta_d} \left(-\frac{1}{2} \right)^{\sum_{j=1}^d |\Delta_j|} f(x_{l_1, i_1 + \Delta_1}, x_{l_2, i_2 + \Delta_2}, \dots, x_{l_d, i_d + \Delta_d}), -1 \leq \Delta_j \leq 1, 1 \leq j \leq d. \quad (7)$$

In the case when the interpolated function has a nonzero value at the boundary, the one-dimensional basis is supplemented by two additional functions: $\varphi_{0,0}(x)$ and $\varphi_{0,1}(x)$ (Figure 4). Two values are added to the polynomial given by Equation (5): $a_{0,0}\varphi_{0,0}(x)$ and $a_{0,1}\varphi_{0,1}(x)$, where $a_{0,0} = f(0)$, $a_{0,1} = f(1)$. By analogy, for multidimensional interpolation, it follows that if $l_j = 0$, then $i_j = 0, 1$ in Equation (6) and $\Delta_j = 0$ in Equation (7). Allowance for boundary values in the multidimensional case can be considered as the construction of sparse grids for all faces of lower dimensions. Figure 5 shows a sparse grid, which takes into account the boundary values.

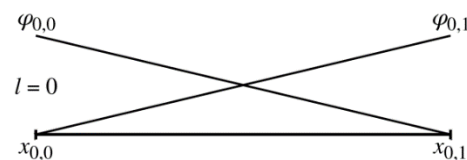


Figure 4. Additional basis functions taking into account boundary values.

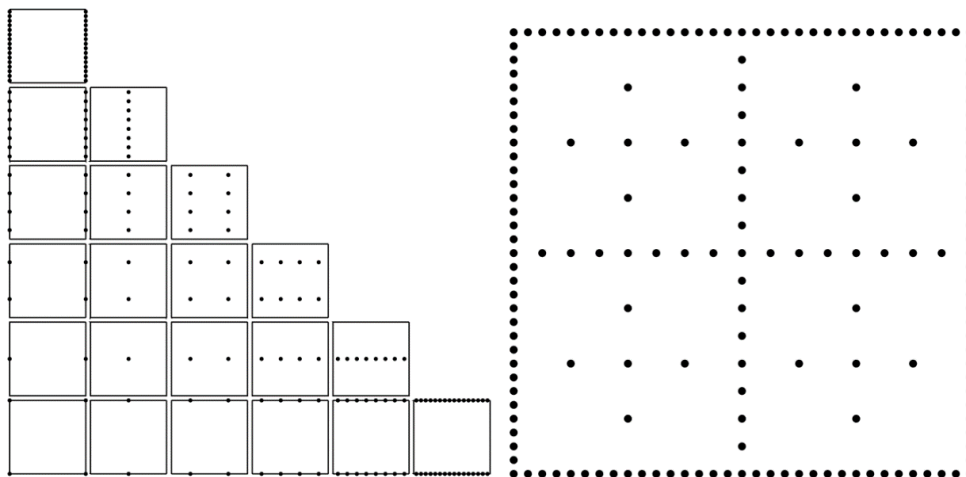


Figure 5. Sparse grid of the level $n = 4$, which takes into account boundary values.

In addition, there are adaptive sparse grids for which a general tree can be used to perform structuring. Each vertex of the tree corresponds to a certain basis function $\varphi_{\mathbf{l}, \mathbf{i}}$. If the value of the corresponding coefficient $a_{\mathbf{l}, \mathbf{i}} / \max(f(x_{\mathbf{l}, \mathbf{i}}), 1) > \varepsilon$, where ε is some predetermined value, then each vertex creates $2d$ descendants, which correspond to the basis functions of the next level. This process continues recursively until the values $a_{\mathbf{l}, \mathbf{i}}$ at all leaf vertices become less than ε . With this approach, it is important to make sure that there is no duplication of vertices.

Consider some examples. Figure 6 shows several functions $\mathbb{R}^2 \rightarrow \mathbb{R}$ and the resulting adaptive grid, Figure 7 shows grids for functions $\mathbb{R}^3 \rightarrow \mathbb{R}$.

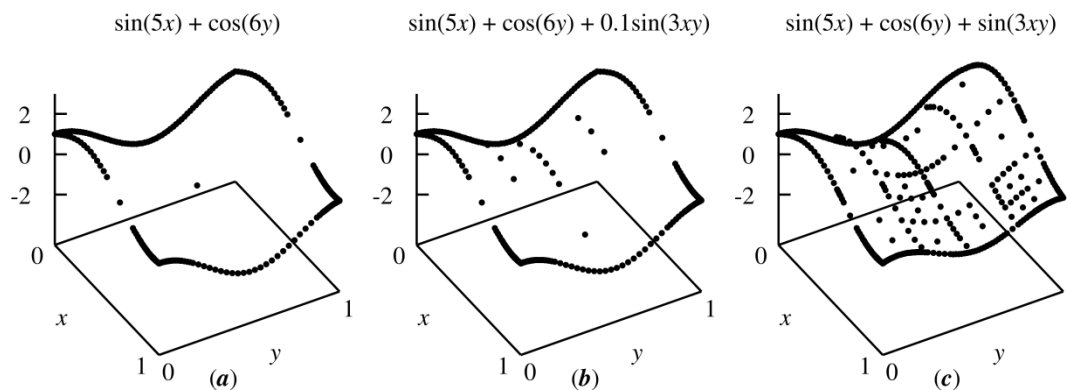


Figure 6. Examples showing interpolation using functions of two variables. (a) Linear combination of univariate functions. (b) Linear combination of univariate functions and a function of two variables x and y with a small coefficient. (c) Linear combination of univariate and a function of two variables x and y .

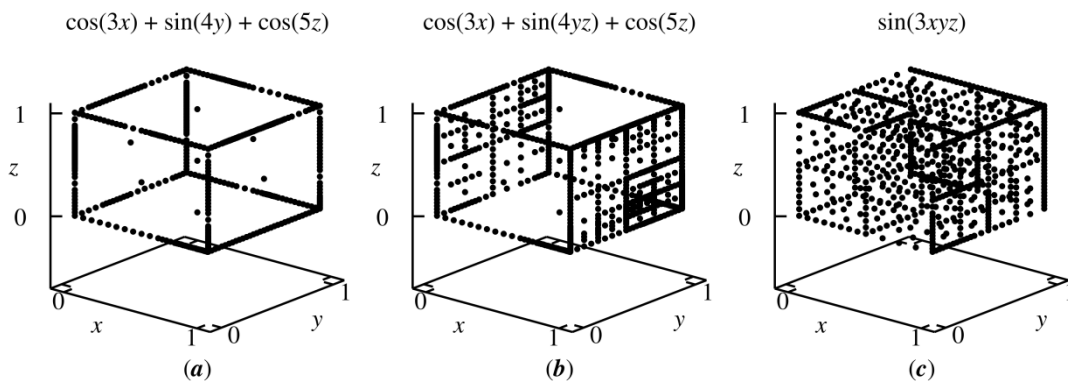


Figure 7. Examples of grids for functions of three variables. (a) Linear combination of univariate functions. (b) Linear combination of univariate functions and a function of two variables y and z . (c) Nonlinear function of three variables.

It can be seen from the figures that if the initial dependence is a linear combination of functions determined by certain subgroups of variables, then the adaptive sparse grid will become more dense not in the entire set (Figures 6c and 7c), but only in subsets of lower dimension that correspond to these subgroups (Figures 6a and 7a,b). The subsets for grid construction are determined by those subgroups of parameters, for which the mixed derivatives are nonzero, and the grid density directly depends on the values of these derivatives (Figure 6b,c).

To build a solution for the system given by Equation (1), the uncertainty area $y_i^0 \in [y_i^0, \bar{y}_i^0]$, $1 \leq i \leq m$ is transformed with the help of displacement and stretching into a m -dimensional unit cube. Taking into account that solving the problem requires interpolating n functions at once (n is the number of phase variables of the system), Equations (6) and (7) will take the following form:

$$P^k(y^0) = \sum_{l,i} a_{l,i}^k \varphi_{l,i}(y^0),$$

where

$$a_{l,i}^k = \sum_{\Delta_1, \dots, \Delta_m} \left(-\frac{1}{2} \right)^{\sum_{j=1}^m |\Delta_j|} y^k(y_{1,l_1,i_1+\Delta_1}^0, y_{2,l_2,i_2+\Delta_2}^0, \dots, y_{m,l_m,i_m+\Delta_m}^0), \quad -1 \leq \Delta_j \leq 1, \quad 1 \leq j \leq m \quad (8)$$

The vector norm $a_{l,i}^k$ (for example, the maximum one) can be used as a criterion for adapting the grid.

Construct an interpolation polynomial $\mathbf{P}^{k+1}(\mathbf{y}^0)$. All the solutions that participated in the calculation of the coefficients given by Equation (8) are transferred to the $(k+1)$ -th time layer using some numerical integration method, after which a new set of $\mathbf{a}_{i,i}^{k+1}$ coefficients is calculated and the adaptation of the grid is performed. When compacting the grid, the addition of new basis functions occurs at the k -th time layer and the solutions involved in computing the corresponding weight coefficients are transferred to the next layer.

The efficiency of the considered approach will be noticeable when many mixed derivatives of the solution with respect to the parameters $\frac{\partial^{\sum \alpha_i} \mathbf{y}^k(y_1^0, y_2^0, \dots, y_m^0)}{(\partial y_1^0)^{\alpha_1} (\partial y_2^0)^{\alpha_2} \dots (\partial y_m^0)^{\alpha_m}}, \max_{1 \leq i \leq m} \alpha_i \leq 2, y_i^0 \in [\underline{y}_i^0, \overline{y}_i^0], 1 \leq i \leq m$ are negligible or equal to zero. Particularly, this takes place, if the solution to the ODE system can be represented as a linear combination of functions determined by a certain subset of interval parameters.

Thus, the scope of application of the proposed approach is rather wide and includes various dynamic systems. In the next section, it is demonstrated how the method is applied to some representative problems.

3. Approximation of the Algorithm for Nonlinear Dynamics Problems

To characterize computational costs, a criterion is determined, which is equal to the average number of integrated non-interval ODE systems at a time step in the computational process:

$$I = \frac{1}{N} \sum_{k=1}^N C_k,$$

where C_k is the number of nodes at the k step. A similar criterion exists for the classical adaptive interpolation algorithm [11]. The I value is equivalent to the number of sampling points from the original region of uncertainty.

To estimate the posterior interpolation error at the initial moment of time, n_{check} points are randomly generated:

$$y_i^j(t_0) = rand[\underline{y}_i^0, \overline{y}_i^0], 1 \leq i \leq m, 1 \leq j \leq n_{check}.$$

For the initial conditions obtained, with the help of a numerical integration method, solutions are constructed at the final moment of time t_N . The relative posterior global estimate of the error is written as follows:

$$error = \max_{\substack{1 \leq j \leq n_{check}, \\ 1 \leq i \leq n}} \frac{|P_i^N(y_1^j(t_0), y_2^j(t_0), \dots, y_m^j(t_0)) - y_i^j(t_N)|}{\max(|y_i^j(t_N)|, 1)}.$$

Let us integrate several ODE interval systems using the described approach. The calculation is performed for two values of $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-5}$ (ε imposes a restriction on the values of the weight coefficients of the basic functions when constructing an adaptive sparse grid). First, let us take into account an ordinary differential system with two interval initial conditions, which describes a conservative oscillator:

$$\begin{cases} x' = y, y' = -\sin(x), \\ x(0) = x_0 \in [-1, 1], y(0) = y_0 \in [0, 1], t \in [0, 25]. \end{cases} \quad (9)$$

Figure 8 shows a set of solutions for the system given by Equation (9) at different moments of time; it twists into a spiral structure during the integration. Figure 9 shows the grid resulting from applying the algorithm. The points in these two figures correspond to each other.

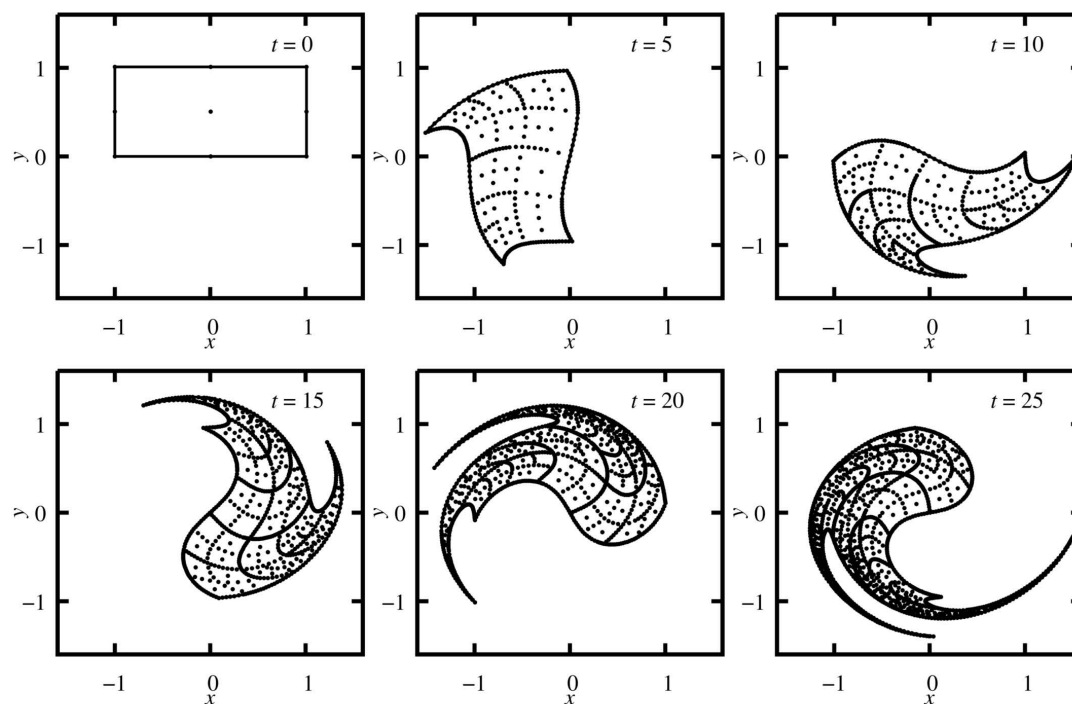


Figure 8. The interval solution of system given by Equation (9) at different moments of time.

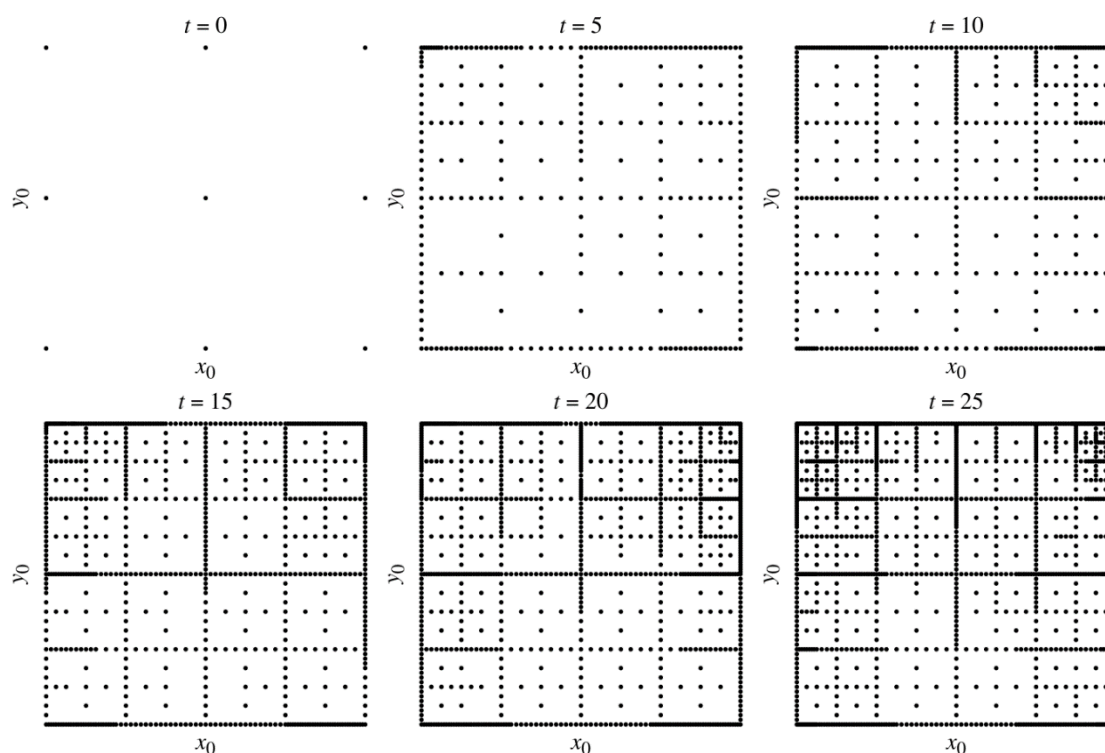


Figure 9. The grids obtained in the process of solving system given by Equation (9).

Table 1 shows a comparison of computational costs and error estimates for different approaches. When set to a low precision ($\varepsilon = 10^{-3}$), adaptive sparse grids work a little faster than the classical adaptive interpolation algorithm, and twice as fast as conventional sparse grids. However, for $\varepsilon = 10^{-5}$, the classical algorithm wins due to the application

of an interpolation polynomial of a high degree. The levels of the grids were adjusted to obtain approximately the same error as in other approaches.

Table 1. Comparison of approaches for system given by Equation (9).

| Methods | $\varepsilon=10^{-3}$ | | $\varepsilon=10^{-5}$ | |
|---|-----------------------|----------------------|-----------------------|----------------------|
| | <i>I</i> | <i>error</i> | <i>I</i> | <i>error</i> |
| full grid (level 6 and level 8) | 4225 | 3.6×10^{-3} | 66,049 | 2.2×10^{-4} |
| sparse grid (level 7 and level 10) | 1793 | 7.7×10^{-3} | 36,865 | 6.9×10^{-5} |
| adaptive sparse grid | 689 | 6.6×10^{-3} | 9455 | 1.1×10^{-4} |
| adaptive interpolation algorithm, $p = 2$ | 3877 | 6.3×10^{-3} | 52,195 | 1.8×10^{-4} |
| adaptive interpolation algorithm, $p = 4$ | 990 | 6.8×10^{-3} | 4752 | 1.5×10^{-4} |

Next, the Volterra-Lotka model with interval initial conditions and one interval coefficient is considered. The Cauchy problem in the case has the form:

$$\begin{cases} x' = 4x - \frac{5}{4}xy - \alpha x^2, & y' = -2y + \frac{1}{2}xy - \frac{1}{20}y^2, \\ x(0) = x_0 \in [4, 5], & y(0) = y_0 \in [2.8, 3.2], \quad t \in [0, 25], \end{cases} \quad (10)$$

where $\alpha \in [-0.05, 0.05]$.

This model describes predator–prey interactions. A feature of the system is the fact that at $\alpha < 0$ there is an unstable focus and the amplitude of fluctuations in the population of species grows, and at $\alpha > 0$ the focus is stable and the state of the system tends to be stationary over time.

Figure 10 shows the set of solutions for the system at different points in time. The following picture is clearly observed here: some part of the set converges to a point, which corresponds to a stable focus, and another part of the set increases in its size, which corresponds to an unstable focus. Figure 11 shows the resulting grid. Due to the fact that uncertainty is present in the parameters, the set of solutions on the phase plane is only a projection of the three-dimensional set onto the two-dimensional phase space. The additional dimension corresponds to the interval parameter α .

Table 2 shows a comparison of the different approaches. Similar to the previous task, adaptive sparse grids are effective with lower accuracy ε .

Table 2. Comparison of approaches on system given by Equation (10).

| Methods | $\varepsilon=10^{-3}$ | | $\varepsilon=10^{-5}$ | |
|---|-----------------------|----------------------|-----------------------|----------------------|
| | <i>I</i> | <i>error</i> | <i>I</i> | <i>error</i> |
| full grid (level 4 and level 6) | 4913 | 3.3×10^{-3} | 274,625 | 2.0×10^{-4} |
| sparse grid (level 3 and level 7) | 705 | 2.4×10^{-3} | 19,713 | 4.3×10^{-5} |
| adaptive sparse grid | 193 | 3.1×10^{-3} | 3170 | 4.8×10^{-5} |
| adaptive interpolation algorithm, $p = 2$ | 544 | 4.6×10^{-3} | 48,013 | 6.8×10^{-5} |
| adaptive interpolation algorithm, $p = 4$ | 369 | 1.6×10^{-3} | 3978 | 5.2×10^{-5} |

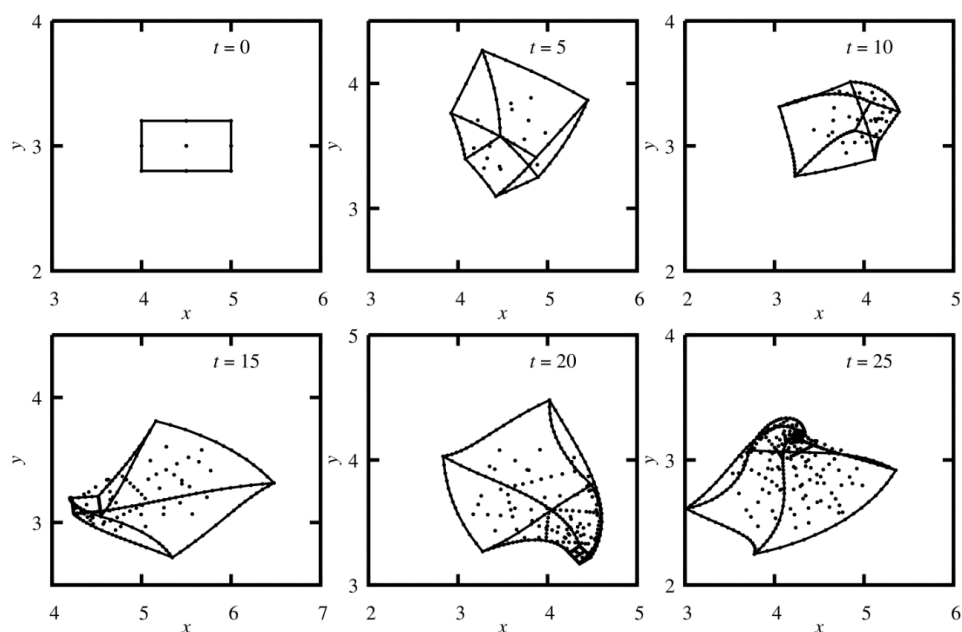


Figure 10. The interval solution of system given by Equation (10) at different times.

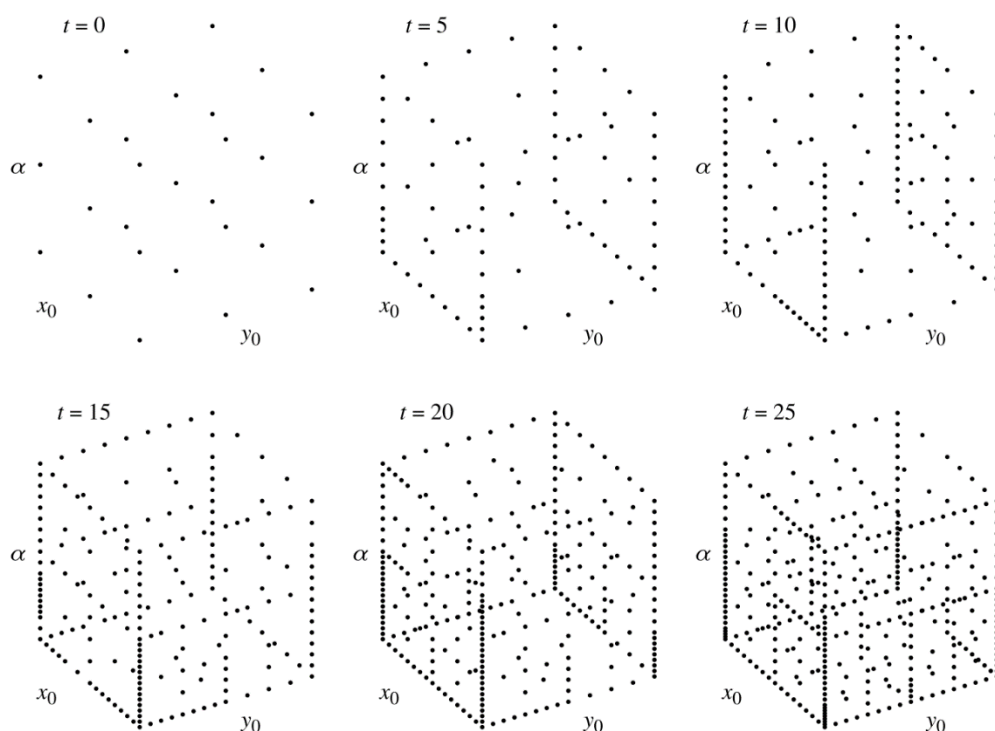


Figure 11. The grid obtained in the process of solving the system given by Equation (10).

Consider an ordinary differential system presenting the expanded Volterra-Lotka model with three interval initial conditions and seven interval parameters:

$$\left\{ \begin{array}{l} x' = x(\delta_1 - y - \varepsilon x), \\ y' = -\gamma_1 y(\delta_2 - x + z) - \varphi y^2, \\ z' = -\gamma_2 z(\alpha - y), \end{array} \right. \quad \left| \begin{array}{l} x(0), y(0), z(0), \delta_1, \delta_2, \gamma_1, \gamma_2 \in [1.0, 1.01], \\ \varepsilon, \varphi \in [-0.0005, 0.0005], \\ \alpha \in [0.9, 0.91]. \end{array} \right. \quad (11)$$

Figure 12 shows the dependences of the interval estimates of solutions on time.

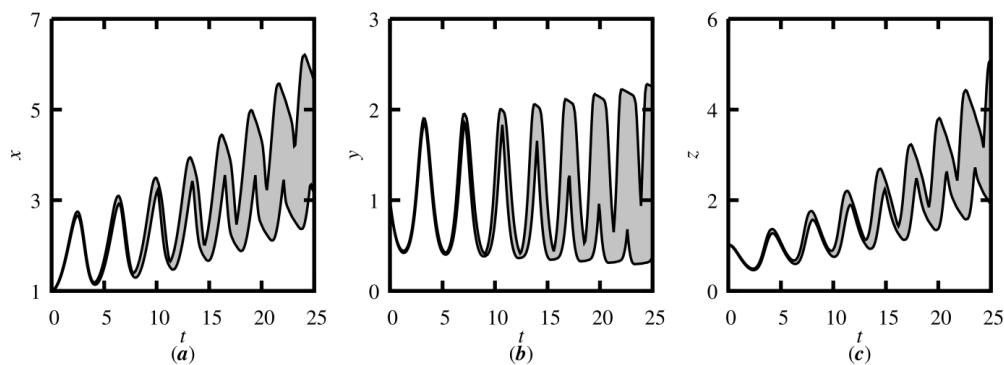


Figure 12. Time dependence of the upper and lower bounds for the solution of system given by Equation (11): (a) $x(t)$; (b) $y(t)$; (c) $z(t)$.

For a reasonable time, the solution was obtained using only adaptive sparse grids. For $\varepsilon = 10^{-3}$, the obtained result was $I = 81,566.1$ and $error = 1.2 \times 10^{-2}$.

Consider a model describing the motion of interacting bodies. The problem can be formulated as a dynamic system with interval initial velocities. The system of ordinary differential equations in dimensionless variables is as follows:

$$\begin{cases} (v_i^x)' = \sum_{j=1, j \neq i}^4 m_j (x_j - x_i) r_{ij}^{-3}, & (v_i^y)' = \sum_{j=1, j \neq i}^4 m_j (y_j - y_i) r_{ij}^{-3}, & (v_i^z)' = \sum_{j=1, j \neq i}^4 m_j (z_j - z_i) r_{ij}^{-3}, \\ x_i' = v_i^x, & y_i' = v_i^y, & z_i' = v_i^z, & 2 \leq i \leq 4, \\ x_1(0) = y_1(0) = z_1(0) = v_1^x(0) = v_1^y(0) = v_1^z(0) = 0, \\ x_{2,3}(0) = \pm 1, & y_{2,3}(0) = z_{2,3}(0) = 0, & v_{2,3}(0) = \begin{pmatrix} 0 & \pm v & 0 \end{pmatrix}^T + \Delta v_{2,3}^T, \\ y_4(0) = 1, & x_4(0) = z_4(0) = 0, & v_4(0) = \begin{pmatrix} 0 & 0 & v \end{pmatrix}^T + \Delta v_4^T, \\ t \in [0.0, 0.02] \end{cases} \quad (12)$$

where $r_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$ is the distance between two bodies, $v = 316.23$ is the initial velocity of bodies, $m_1 = 10^5$, $m_{2,3,4} = 10^{-5}$ are body masses, $\Delta v_{2,3,4} = ([-2, 2], [-2, 2], [-2, 2])$ are the interval uncertainties in body velocities.

Figure 13 shows graphs for the dependence of the interval estimates of the 2nd body coordinates and velocities on time. Similar to the previous problem, the solution was calculated only using adaptive sparse grids. For $\varepsilon = 10^{-3}$, the obtained result was $I = 133830.9$ and $error = 2.6 \times 10^{-2}$.

This system is demonstrative because the uncertainty in the speed of a particular body mainly affects the position and speed of that particular body and has little effect on other bodies. In this regard, the solution of the system will have a specific form, as most of the mixed derivatives will be close to zero.

Note that the classical adaptive interpolation algorithm for systems given by Equations (6) and (7) constructs sets of solutions with fewer integrations of the corresponding non-interval ordinary differential systems since it uses nonlinear interpolation. However, when the number of interval parameters increases (systems given by Equations (8) and (9)), the use of adaptive sparse grids becomes more efficient. When increasing the dimension of the problem, it is practically impossible to increase the degree of the interpolation polynomial in the adaptive interpolation algorithm to obtain higher accuracy due to the exponential growth of the number of nodes in the grid. Therefore, for high dimensional-problems, it is suitable to use methods that have lower accuracy, but at the same time allow reasonable computational costs; in particular, adaptive sparse grids.

The examples above demonstrate that by using sparse grids it is possible to simulate dynamic systems with ten interval uncertainties in a reasonable time. When solving system given by Equation (11), the equivalent number of sampling points was about 80 thousand, and in the case of using classical interpolation with the degree of polynomial equal to 4, the value would be of order 10^7 . A lower estimate of the computational cost can be obtained. It

follows from Equation (8) that the number of solved non-interval ODE systems cannot be less than 3^m . The upper estimate of the computational costs, in the general case, essentially depends on the features of the ODE system being solved, in particular on the values of the mixed derivatives of the solution with respect to the point values of the interval parameters. For comparison, the classical adaptive interpolation algorithm requires at least $(p+1)^m$ points, and the method proposed in Reference [15] requires $\frac{(m+p)!}{m!p!}$ points, where p is the degree of the interpolation polynomial.

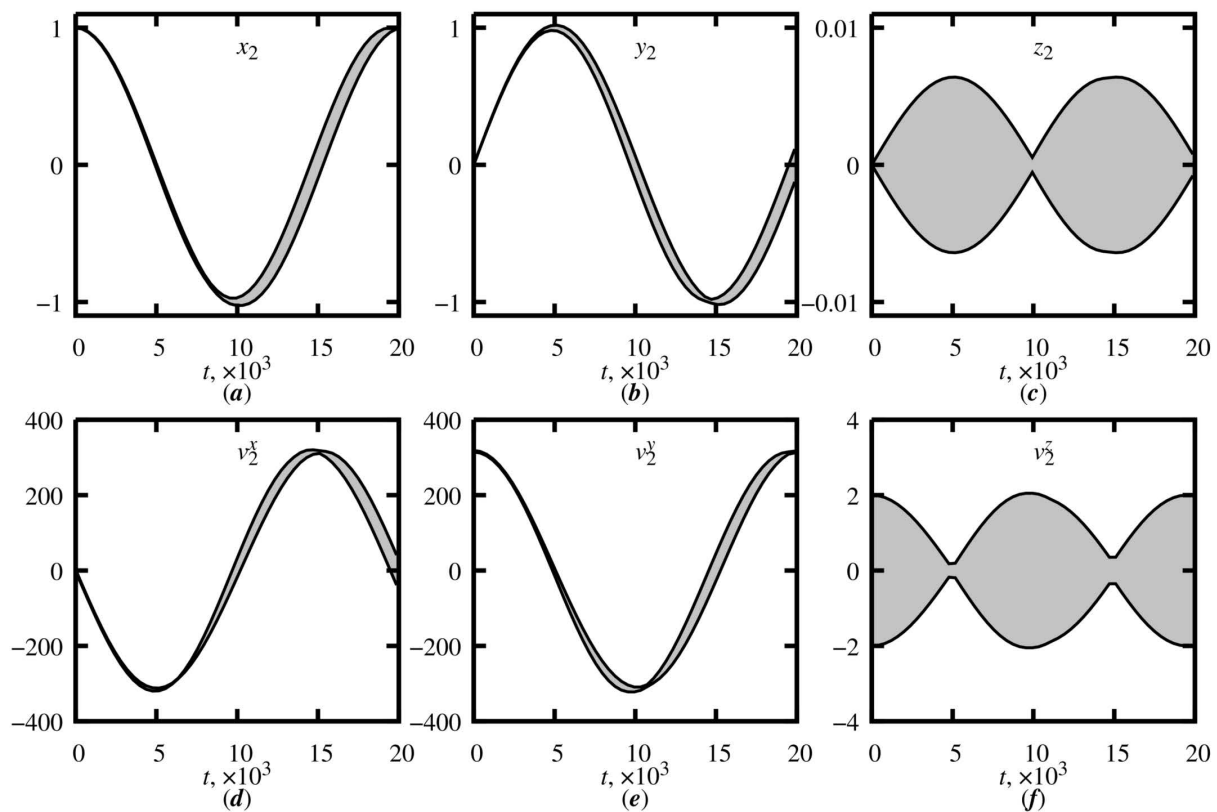


Figure 13. Time dependencies of upper and lower estimates of the 2nd body coordinates and velocities: (a) $x_2(t)$; (b) $y_2(t)$; (c) $z_2(t)$; (d) $v_2^x(t)$; (e) $v_2^y(t)$; (f) $v_2^z(t)$.

4. Computation of an Interval Stress Tensor for Materials with a Covalent Chemical Bond

Let us consider an applied problem of computational materials science, within the framework of which the stresses arising during the deformation of an ideal crystal are calculated [12]. Different angles are possible between the orientation of the crystal lattice and the direction of stretching with a fixed stretch value. The stress tensor thus becomes interval. This problem is solved using molecular dynamics simulation. The motion of atoms is described by the classical equations of dynamics:

$$\begin{cases} \mathbf{r}_i' = \mathbf{v}_i, \\ \mathbf{v}_i' = \frac{1}{m_i} \mathbf{F}_i, \end{cases}$$

where \mathbf{r}_i is the radius vector of the atom with the number i , \mathbf{v}_i is its velocity, m_i is its mass, and \mathbf{F}_i is the force acting on the atom, in this case $\mathbf{F}_i = -\nabla_i E$, where E is the total energy of the system, and ∇_i is the gradient along the position of the atom with the number i .

In this problem, materials with a covalent interatomic bond are considered. The total energy of interaction between atoms of such materials is well described using the Tersoff potential [25]:

$$\left\{ \begin{array}{l} E = \frac{1}{2} \sum_i \sum_{j \neq i} V_{ij}, \quad V_{ij} = f_C(r_{ij}) (f_R(r_{ij}) + b_{ij} f_A(r_{ij})), \\ f_C(r) = \begin{cases} 1, & r < R - D, \\ \frac{1}{2} \left(1 - \sin\left(\frac{\pi}{2} \frac{r-R}{D}\right) \right), & R - D \leq r < R + D, \\ 0, & R + D \leq r, \end{cases} \\ f_R(r) = A \exp(-\lambda_1 r), \quad f_A(r) = -B \exp(-\lambda_2 r), \\ b_{ij} = \left(1 + \beta^n \zeta_{ij}^n \right)^{-\frac{1}{2n}}, \\ \zeta_{ij} = \sum_{k \neq i, j} f_C(r_{ik}) g(\theta_{ijk}) \exp\left(\lambda_3^m (r_{ij} - r_{ik})^m \right), \\ g(\theta) = \gamma \left(1 + \frac{c^2}{d^2} - \frac{c^2}{d^2 + (\cos(\theta) - \cos(\theta_0))^2} \right), \end{array} \right.$$

where E is total system energy, V_{ij} is the contribution to the interaction energy of atoms with numbers i and j , r_{ij} is the distance between atoms with numbers i and j , $f_C(r)$ is a cut-off function, $f_R(r)$ and $f_A(r)$ are the repulsion and attraction potentials, respectively, and R , D , A , B , n , m , λ_1 , λ_2 , λ_3 , β , γ , c , d and $\cos(\theta_0)$ are potential parameters that are selected in order to reproduce the properties of the simulated material. Methods of parametric identification of the Tersoff potential parameters are considered in papers [26,27].

The initial positions of atoms and their number are determined by the structure of the crystal lattice and the restriction on the minimum size of the simulated space is specified by the structure of the potential.

Consider crystalline silicon as a typical material. The simulated sample is represented by eight unit cells of a diamond crystal lattice making up a cube of $2 \times 2 \times 2$ in size, with periodic boundary conditions; each unit cell contains eight unique atoms (Figure 14), so a total of 64 atoms are involved in the simulation. Initial speeds are considered to be zero. The initial conditions for a dynamical system can be represented as follows:

$$\left\{ \begin{array}{l} \mathbf{r}_i(0) = \left((x, y, z)^T + (dx, dy, dz)^T \right) a, \quad (x, y, z) \in \text{Base}, \quad dx, dy, dz \in \{0, 1\}, \quad \mathbf{v}_i(0) = (0, 0, 0)^T, \\ \text{Base} = \left\{ \begin{array}{l} (0, 0, 0), (0, 0.5, 0.5), (0.5, 0, 0.5), (0.5, 0.5, 0), (0.75, 0.75, 0.75), \\ (0.75, 0.25, 0.25), (0.25, 0.75, 0.25), (0.25, 0.25, 0.75) \end{array} \right\}, \end{array} \right.$$

where $a = 5.429 \times 10^{-10}$ m is the linear size of a unit cell of a silicon crystal.

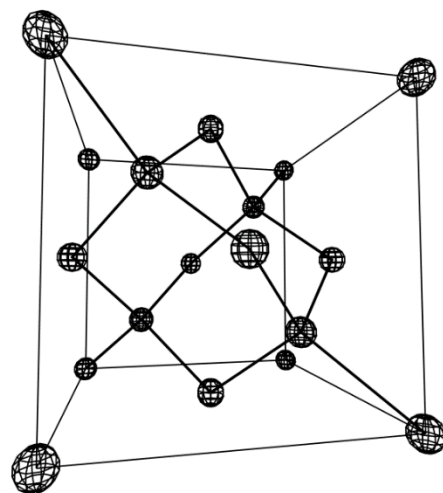


Figure 14. Unit cell of a silicon crystal.

To take into account deformation, 4 additional variables are introduced: one of them reflects the elongation value, and three more are responsible for the angle between the orientation of the lattice and the direction of stretching. In this case, the elongation is set to be fixed, and the variables responsible for the rotation are taken as interval. Rotation is generated evenly using quaternions [28].

The final system looks like this:

$$\left\{ \begin{array}{l} \mathbf{r}_i' = \mathbf{v}_i, \mathbf{v}_i' = -\frac{1}{2m_i} \sum_i \sum_{j \neq i} \nabla_i (f_C(r_{ij}) (f_R(r_{ij}) + b_{ij} f_A(r_{ij}))), \\ r_{ij}(s, \mu_1, \mu_2, \mu_3) = \|\mathbf{S}(s) \mathbf{R}(\mu_1, \mu_2, \mu_3) \mathbf{r}_{ij}\|, \\ \mathbf{S}(s) = \text{diag}(1 + s, 1, 1), \\ \mathbf{R}(\mu_1, \mu_2, \mu_3) = \begin{pmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1 q_2 - q_3 q_0) & 2(q_1 q_3 + q_2 q_0) \\ 2(q_1 q_2 + q_3 q_0) & 1 - 2(q_1^2 + q_3^2) & 2(q_2 q_3 - q_1 q_0) \\ 2(q_1 q_3 - q_2 q_0) & 2(q_2 q_3 + q_1 q_0) & 1 - 2(q_1^2 + q_2^2) \end{pmatrix}, \\ (q_0, q_1, q_2, q_3) = (\sqrt{1 - \mu_1} \sin(2\pi\mu_2), \sqrt{1 - \mu_1} \cos(2\pi\mu_2), \sqrt{\mu_1} \sin(2\pi\mu_3), \sqrt{\mu_1} \sin(2\pi\mu_3)), \\ t \in [0, 10^{-12}], \end{array} \right. \quad (13)$$

where $m = 4.65 \times 10^{-26}$ kg is the mass of atoms, $s = 0.1$ is the relative elongation of the sample, $\mu_1 \in [0.1, 0.9]$, $\mu_2 \in [0.1, 0.9]$, and $\mu_3 \in [0.1, 0.9]$ are stretching direction parameters, $R = 2.85 \times 10^{-10}$ m, $D = 0.15 \times 10^{-10}$ m, $A = 6.12 \times 10^{-16}$ J, $B = 1.81 \times 10^{-17}$ J, $c = 9.69$, $d = 2.35$, $n = 4.16$, $\beta = 0.132$, $\lambda_1 = 3.36 \times 10^{10} \text{ m}^{-1}$, $\lambda_2 = 1.27 \times 10^{10} \text{ m}^{-1}$, $\lambda_3 = 1.19 \times 10^{10} \text{ m}^{-1}$, $\gamma = 5.71$, and $\cos(\theta_0) = -0.408$ are the parameters of the potential.

Integration of the resulting ordinary differential system (13) was carried out using the Verlet velocity method with a constant integration step of 10^{-15} s. As a result, the interval stress tensor was obtained (values are given in Pascals):

$$\begin{pmatrix} [-1.58 \times 10^{10}, -1.43 \times 10^{10}] & [-1.35 \times 10^9, 1.35 \times 10^9] & [-1.35 \times 10^9, 1.35 \times 10^9] \\ [-1.35 \times 10^9, 1.35 \times 10^9] & [-4.79 \times 10^9, -1.46 \times 10^9] & [-1.42 \times 10^9, 1.42 \times 10^9] \\ [-1.35 \times 10^9, 1.35 \times 10^9] & [-1.42 \times 10^9, 1.42 \times 10^9] & [-4.79 \times 10^9, -1.46 \times 10^9] \end{pmatrix}$$

For $\varepsilon = 10^{-2}$, the obtained result was $I = 1079132.3$ and $error = 2 \times 10^{-1}$.

Note that the possibilities of the proposed approach are not limited to a specific type of interatomic interaction potential in a material. The method can be applied to the simulation of the stress-strain state of materials with various types of chemical bonds, including the modeling of composite materials.

5. Discussion

In the previous sections, the proposed approach was tested on representative interval problems of nonlinear dynamics and computational materials science. It is found that, thanks to sparse grids, it is possible to integrate ODE systems with a large number of interval uncertainties in a reasonable time. To estimate the computational costs, a criterion was used that is equal to the equivalent number of sampling points from the original uncertainty region.

Table 1 shows estimates of the computational costs for the ODE system given by Equation (9) describing a conservative oscillator with two interval initial conditions for two values ε . For $\varepsilon = 10^{-3}$, the approach proposed in the paper works 1.5 to 5 times faster than the classical adaptive interpolation algorithm.

Table 2 shows the computational costs when integrating the ODE system given by Equation (10), describing the Lotka-Volterra model with two interval initial conditions and one interval parameter. Here, for $\varepsilon = 10^{-3}$, the use of adaptive sparse grids gives an acceleration of 1.9–2.8 times compared to the classical algorithm, and for $\varepsilon = 10^{-5}$, the acceleration is from 1.25 time to 15 time.

For ODE systems given by Equations (11) and (12), it was possible to obtain a solution in a reasonable time only using the approach proposed in the paper since the number of interval uncertainties is quite large. To solve system given by Equation (11), the equivalent number of sampling points was about 80 thousand, and in the case of using the classical

algorithm with a degree of polynomial 4, the value would be about 10 million. Thus, using sparse grids in this problem gives an acceleration of at least 125 times.

The tables show that adaptive sparse grids work faster than regular sparse grids, and even faster than full grids. This fact is in line with the theoretical foundations. The classical adaptive interpolation algorithm in example (9) with two interval uncertainties showed itself slightly better only when $\varepsilon = 10^{-5}$ and $p = 4$. This is primarily due to the chosen value of p . It is known that the greater the degree of the interpolation polynomial, the faster the error decreases with increasing mesh density. Therefore, it seems promising to use sparse grids on a nonlinear basis. We should also note the possibilities of applying adaptive grids to ODE systems not only with interval uncertainties but also with stochastic uncertainties, including applied nonlinear systems.

6. Conclusions

The adaptive interpolation algorithm allows simulating dynamic systems with interval parameters. In the course of the algorithm operation, a regular grid is constructed in the parameter space. The number of grid nodes depends exponentially on the number of interval parameters, which limits the scope of the algorithm. A typical situation is when the degree of influence of different parameters and their combinations on the solution can differ significantly. This can be used in adaptive interpolation. The paper presents an adaptive interpolation algorithm on sparse grids, which allows for reducing the exponential complexity when solving multidimensional problems in parameter space. The efficiency of the proposed approach has been demonstrated on representative interval problems of nonlinear dynamics and computational materials science. It is shown that, for most variants, adaptive sparse grids are more efficient than the classical adaptive interpolation algorithm in terms of computational costs. With the suggested method, it was possible to solve problems with up to 10 interval parameters in a reasonable amount of time. At the same time, the classical algorithm of adaptive interpolation failed to do so. Taking into account that an increase in the degree of the interpolation polynomial in the classical adaptive interpolation algorithm leads to higher accuracy and lower computational costs, we can outline the use of sparse grids with a nonlinear basis as a direction for further research.

Author Contributions: Methodology, software, investigation, A.Y.M.; investigation, software, validation, A.A.Z.; conceptualization, methodology, supervision, D.L.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Ministry of Science and Higher Education of the Russian Federation, project No 075-15-2020-799.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Richtmyer, R.D.; Moore, R.E. Interval Analysis. *Math. Comput.* **1968**, *22*, 219. [[CrossRef](#)]
2. Moore, R.E.; Kearfott, R.B.; Cloud, M.J. *Introduction to Interval Analysis*; SIAM: Philadelphia, PA, USA, 2009.
3. Eijgenraam, P. The Solution of Initial Value Problems Using Interval Arithmetic Formulation and Analysis of an Algorithm. *Math. Comput.* **1984**, *42*, 343. [[CrossRef](#)]
4. Dobronets, B.S. *Interval Mathematics*; Krasnoyarsk State University: Krasnoyarsk, Russia, 2007. (In Russian)
5. Shary, S.P. *Finite Dimensional Interval Analysis*; Institute of Computational Technologies SB RAS. XYZ Publisher: Novosibirsk, Russia, 2019; p. 634. (In Russian)
6. Chernousko, F.L. *Evaluation of Phase States of Dynamic Systems. The Method of Ellipsoids*; Science: Moscow, Russia, 1988; p. 319. (In Russian)
7. Lohner, R.J. Enclosing the Solutions of Ordinary Initial and Boundary Value Problems. In *Computer Arithmetic: Scientific Computation and Programming Languages*; Wiley-Teubner Series in Computer Science; Wiley-Teubner: Stuttgart, Germany, 1987; pp. 255–286.

8. Makino, K.; Berz, M. Models and Their Applications. Numerical Software Verification Computations. In *Proceedings of the Numerical Software Verification 2017 Conference; Heidelberg, Germany, 22–23 July 2017*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 3–13.
9. Rogalev, A.N. Guaranteed Methods of Ordinary Differential Equations Solution on the Basis of Transformation of Analytical Formulas. *Vychisl. Tekhnol.* **2003**, *8*, 102–116. (In Russian)
10. Ermakov, S.M.; Mikhailov, G.A. *Statistical Modeling*; Science: Moscow, Russia, 1982. (In Russian)
11. Morozov, A.; Reviznikov, D.L. Adaptive Interpolation Algorithm Based on a kd-Tree for Numerical Integration of Systems of Ordinary Differential Equations with Interval Initial Conditions. *Differ. Equ.* **2018**, *54*, 945–956. [[CrossRef](#)]
12. Morozov, A.; Zhuravlev, A.A.; Reviznikov, D.L. Analysis and Optimization of an Adaptive Interpolation Algorithm for the Numerical Solution of a System of Ordinary Differential Equations with Interval Parameters. *Differ. Equ.* **2020**, *56*, 935–949. [[CrossRef](#)]
13. Morozov, A.; Reviznikov, D.L.; Gidasov, V.Y. Adaptive Interpolation Algorithm Based on a kd-Tree for the Problems of Chemical Kinetics with Interval Parameters. *Math. Model. Comput. Simul.* **2019**, *11*, 622–633. [[CrossRef](#)]
14. Morozov, A.; Federal Research Center Computer Science and Control of the Russian Academy of Sciences; Reviznikov, D.L. Moscow aviation Institute (national research University) Modeling of Dynamic Systems with Interval Parameters in the Presence of Singularities. *Nelineinaya Din.* **2020**, *16*, 479–490. [[CrossRef](#)]
15. Fu, C.; Ren, X.; Yang, Y.-F.; Lu, K.; Qin, W. Steady-state response analysis of cracked rotors with uncertain-but-bounded parameters using a polynomial surrogate method. *Commun. Nonlinear Sci. Numer. Simul.* **2019**, *68*, 240–256. [[CrossRef](#)]
16. Fu, C.; Xu, Y.; Yang, Y.; Lu, K.; Gu, F.; Ball, A. Response analysis of an accelerating unbalanced rotating system with both random and interval variables. *J. Sound Vib.* **2020**, *466*, 115047. [[CrossRef](#)]
17. Smoljak, S.A. Quadrature and Interpolation Formulae on Tensor Products of Certain Classes of Functions. *Dokl. Akad. Nauk. Sssr* **1963**, *148*, 1042. (In Russian)
18. Faber, G. Über stetige Funktionen. (Mit 2 Figuren im Text). *Math. Ann.* **1909**, *66*, 81–94. [[CrossRef](#)]
19. Yserentant, H. Hierarchical bases. In *ICIAM 91: Proceedings of the Second International Conference on Industrial and Applied Mathematics, Washington, DC, 8–12 July 1991*; SIAM: Philadelphia, PA, USA, 1992; pp. 256–276.
20. Gerstner, T.; Griebel, M. Sparse Grids. In *Encyclopedia of Quantitative Finance*; Wiley: Hoboken, NJ, USA, 2010.
21. Garcke, J. *Sparse Grids in a Nutshell*; Sparse Grids and Applications; Lecture Notes in Computational Science and Engineering; Springer: Berlin/Heidelberg, Germany, 2013; Volume 88, pp. 57–80.
22. Brumm, J.; Scheidegger, S. Using Adaptive Sparse Grids to Solve High-Dimensional Dynamic Models. *Econometrica* **2017**, *85*, 1575–1612. [[CrossRef](#)]
23. Bungartz, H.-J.; Griebel, M. Sparse grids. *Acta Numerica.* **2004**, *13*, 147–269.
24. Zhang, Z.; Tretyakov, M.V.; Rozovskii, B.; Karniadakis, G.E. A recursive sparse grid collocation method for differential equations with white noise. *SIAM J. Sci. Comput.* **2014**, *36*, A1652–A1677. [[CrossRef](#)]
25. Tersoff, J. New empirical approach for the structure and energy of covalent systems. *Phys. Rev. B* **1988**, *37*, 6991–7000. [[CrossRef](#)] [[PubMed](#)]
26. Abgaryan, K.K.; Posypkin, M. Optimization methods as applied to parametric identification of interatomic potentials. *Comput. Math. Math. Phys.* **2014**, *54*, 1929–1935. [[CrossRef](#)]
27. Abgaryan, K.K.; Grevtsev, A.V. Parametric Identification of Tersoff Potential for Two-Component Materials. *Agent Multi-Agent Syst. Technol. Appl.* **2020**, *173*, 257–268. [[CrossRef](#)]
28. Shoemake, K. Uniform random rotations. *Graph. Gems Iii (Ibm Version)* **1992**, 124–132. [[CrossRef](#)]