Projecting interval uncertainty through the discrete Fourier transform: an application to time signals with poor precision

Marco Behrendt[a,b], Marco de Angelis[b], Liam Comerford[b], Yuanjin Zhang[c], Michael Beer[a,b,d]

[a]*Institute for Risk and Reliability, Leibniz Universität Hannover, Germany.*
[b]*Institute for Risk and Uncertainty, University of Liverpool, United Kingdom.*
[c]*School of Safety Science and Emergency Management, Wuhan University of Technology, China.*
[d]*International Joint Research Center for Engineering Reliability and Stochastic Mechanics, Tongji University, Shanghai, China.*

Abstract

The discrete Fourier transform (DFT) is often used to decompose a signal into a finite number of harmonic components. The efficient and rigorous propagation of the error present in a signal through the transform can be computationally challenging. Real data is always subject to imprecision because of measurement uncertainty. For example, such uncertainty may come from sensors whose precision is affected by degradation, or simply from digitisation. On many occasions, only error bounds on the signal may be known, thus it may be necessary to automatically propagate the error bounds without making additional artificial assumptions. This paper presents a method that can automatically propagate interval uncertainty through the DFT while yielding the exact bounds on the Fourier amplitude and on an estimation of the Power Spectral Density (PSD) function. The method allows technical analysts to project interval uncertainty–present in the time signals–to the Fourier amplitude and PSD function without making assumptions about the dependence and the distribution of the error over the time steps. Thus, it is possible to calculate and analyse system responses in the frequency domain without conducting extensive Monte Carlo simulations nor running expensive optimisation in the time domain. The applicability of this method in practice is demonstrated by a technical application. It is also shown that conventional Monte Carlo methods severely underestimate the uncertainty.

Keywords: Discrete Fourier transform, Complex intervals, Dependency tracking, Interval arithmetic, Power spectral density estimation, Uncertainty quantification.

1. Introduction

The discrete Fourier transform (DFT) is ubiquitous in signal processing and in engineering computing in general. The DFT allows signals to be decomposed into single harmonics and so facilitate their data compression and analysis. Its versatility resides in its mathematical property of being linear and fully invertible. The DFT is used in a range of different applications in science and engineering, such as in spectral analysis, random vibrations, differential equations, data compression, signal processing, image processing or probabilistic programming [1, 2]. There are various algorithms available for transforming a signal with the DFT, of which the best-known is probably the fast Fourier transform (FFT), presented by Cooley & Tukey [3]. Due to the increasing computational power, simulations and equivalent calculations can be carried out ever faster, which has also opened the possibility of conducting the FFT analysis on-the-fly on embedded systems. An overview of a variety of algorithms used can be found in abundance in the literature [4, 5].

The uncertainty in signals can arise, for example, from damaged sensors, equipment failures, measurement errors, operational range, inaccurately calibrated sensors, and incorrectly recorded extreme values. Moreover, data can be imprecise because of measurement uncertainty, e.g., due to degradation or digitisation. There are numerous cases and reasons why the signal can be imprecise, and yet still informative in sensing critical circumstances. In order to obtain accurate simulation results, e.g., in the context of reliability analysis, it is imperative to use suitable simulation methods and models. Since the simulation is only a representation of the real case, real data should be used by all means. Uncertain data increasingly complicates the task of obtaining meaningful results. Some approaches on how to deal with uncertain data can be found in [6, 7, 8].

Handling interval uncertainty is a broad problem at the intersection between engineering, mathematics and statistics [8, 9, 10]. For the reasons mentioned above, the signal can be of poor quality and the interpretation of the simulation results might lead to unrealistic or even dangerous outcomes, for example if an actually disastrous assumption is shifted into an acceptable

range by incorrect quantification. Such a scenario is particularly conceivable in risk analysis, as even smallest deviations in the input data can lead to a system failure that must be detected. Especially in the dynamic behaviour of systems, data with poor precision can lead to enormous problems, e.g., when the natural frequencies of the system under consideration are excited. The behaviour of the system can change significantly in magnitude if imprecise rather than precise data is used. Therefore, every attempt must be made to account for uncertainties.

Especially in the field of random vibrations [11, 12] and stochastic dynamics [13], where environmental processes, such as wind and earthquake loads or sea waves, are the governing excitations, a precise analysis of structures and buildings is indispensable. The DFT is particularly useful here to determine the dominant frequencies of excitation for vibration analysis. A versatile method to represent stochastic processes is the Power Spectral Density (PSD) function, which represents the transformation of signals in the frequency domain and is also calculated by means of the DFT [14]. For instance, it is used in earthquake engineering to display the amplitude of acceleration with respect to frequencies given a signal. Therefore, the amplitude of the DFT is of special interest in these fields, as it has an important physical meaning.

Some approaches have already been taken to estimate a reliable amplitude of the DFT from uncertain data. In the field of stochastic dynamics, missing data were reconstructed under the assumption that they are normally distributed. This probability density function of those data was then propagated through the Fourier transform [15, 16]. In other fields, the FFT and convolution were studied for signals with interval and fuzzy uncertainty [17].

In this work, signals with poor precision are represented by intervals whose width can either be constant or variable along the signal. The objective is to find the ranges of the absolute value of the DFT of such an interval signal, i.e. upper and lower bounds of the Fourier amplitude, and not merely the ranges for real and imaginary parts of the transformed signal. Interval arithmetic is sufficient to propagate interval signals through the DFT, but since the calculation of the amplitude contains repeated variables, this approach usually does not yield exact bounds [18, 19, 20]. An interval algorithm has been proposed in [21] that fully addresses the dependency problem caused by the repeated variables, leading to the exact bounds on the amplitude of the DFT. In this paper, the proposed algorithm is presented

in details and applied to an engineering example involving signals with poor precision.

This work is organised as follows: An overview about the theoretical background that outlines the rules of interval arithmetic used in this work, the interval extensions of the transform, and the theory of PSD function estimation, is given in Section 2. In Section 3, the novel algorithm for propagating interval uncertainty through the DFT is described. Section 4 presents an application for the utilisation of the proposed algorithm. The final conclusions are given in Section 5.

2. Preliminaries

In this section, some essential background that is relevant for this work is presented.

2.1. Interval arithmetic

An interval $\overline{\underline{x}}$ is a compact subset of $\mathbb{R}$ and is defined as $\overline{\underline{x}} = [\underline{x},\ \overline{x}] = \{\underline{x} \leq x \leq \overline{x}\}$.

A complex interval $\overline{\underline{z}} \subseteq \mathbb{C}$, consisting of intervals for the real component $\overline{\underline{z}}_{re} \subset \mathbb{R}$ and the imaginary component $\overline{\underline{z}}_{im} \subset \mathbb{R}$, is defined as

$$\overline{\underline{z}} = \overline{\underline{z}}_{\mathrm{re}} + i\ \overline{\underline{z}}_{\mathrm{im}} = \{z_{\mathrm{re}} + i\ z_{\mathrm{im}} \mid z_{\mathrm{re}} \in \overline{\underline{z}}_{\mathrm{re}}\ \wedge\ z_{\mathrm{im}} \in \overline{\underline{z}}_{\mathrm{im}}\}. \tag{1}$$

Using the rectangular interpretation of a complex interval, there are four endpoints: $z_1 = \underline{z}_{\mathrm{re}} + i\ \underline{z}_{\mathrm{im}},\ z_2 = \underline{z}_{\mathrm{re}} + i\ \overline{z}_{\mathrm{im}},\ z_3 = \overline{z}_{\mathrm{re}} + i\ \underline{z}_{\mathrm{im}},\ z_4 = \overline{z}_{\mathrm{re}} + i\ \overline{z}_{\mathrm{im}}$, each corresponding to a vertex of the rectangle. The addition between two complex intervals $\overline{\underline{z}}$ and $\overline{\underline{w}}$, can be defined in terms of such endpoints as:

$$\overline{\underline{z}} + \overline{\underline{w}} = [\min\{z_i + w_j\}, \max\{z_i + w_j\}],\ \text{for } i, j = 1, ..., 4. \tag{2}$$

For a singleton $a \in \mathbb{R}$, the singleton-interval multiplication is:

$$a\overline{\underline{z}} = [\min\{az_i\}, \max\{az_j\}],\ \text{for } i, j = 1, ..., 4. \tag{3}$$

The absolute value of a complex number is obtained by squaring real and imaginary components separately, followed by the square root of their sum, $|\overline{\underline{z}}| = \sqrt{\overline{\underline{z}}_{\mathrm{re}}^2 + \overline{\underline{z}}_{\mathrm{im}}^2}$. The square and square root of a real interval $\overline{\underline{x}}$ are

$$\overline{\underline{x}}^2 = \begin{cases} [\underline{x}^2,\ \overline{x}^2], & \overline{\underline{x}} > 0 \\ [\overline{x}^2,\ \underline{x}^2], & \overline{\underline{x}} < 0 \\ [0, \max(\underline{x}^2,\ \overline{x}^2)], & \overline{\underline{x}} \ni 0, \end{cases} \tag{4}$$

and

$$\sqrt{\overline{x}} = [\sqrt{\underline{x}},\ \sqrt{\overline{x}}], \quad x > 0. \tag{5}$$

2.2. Signal with interval uncertainty

Sensor data can be subject to experimental uncertainty for a variety of reasons, given that their acquisition can be faulty, intermittent due to maintenance, and of poor precision [8]. The reasons are complex and range from simple measurement errors to total failure of the sensors. Often the sensors are not accurate enough, are not calibrated correctly or are subject to certain technical limitations. For example, extreme values may not be recorded due to sensor threshold limitations. In some cases, the data can be disturbed by external influences, specifically if sensors are used for long-term recordings. It is also possible that the sensor is damaged by the event it is supposed to record, for example an earthquake, and makes incorrect recordings or stops recording completely. The placement of the sensors also has a significant impact on the quality of the signal. In addition, sensors may be temporarily unavailable due to maintenance. If the time interval of unavailability is sufficiently short, intervals could be used to bridge this gap.

In this work, the assumption is made that sensors work accurately only within certain tolerances. These tolerances are represented by intervals, i.e. each value of the data record is described by an upper and lower bound and every arbitrary value within these bounds is possible. The interval uncertainty can be specified as constant or variable along the signal. Two examples of constant and variable interval uncertainty are illustrated in Fig. 1. Intervals are the most conservative model of error, because no assumption is made about how the error is distributed nor how it depends on anything else. Statistics for experimental data carrying interval uncertainty are being extensively studied in the literature and just to mention a few, include [8, 9, 10, 22].

An interval signal can be defined in central notation as

$$\overline{x} = x + \xi\Delta, \tag{6}$$

where, $x \in \mathbb{R}^{\mathbb{N}}$ is the vector of midpoints, $\Delta = [-1, 1]$ is the unitary interval and ξ is the precision of the signal. When the precision is variable across the signal, $\xi \in \mathbb{R}^{\mathbb{N}}$ is a vector of differing scalars, otherwise it is a vector of identical scalars. This representation will be used to generate simulated interval signals from precise intervals given measurement error.

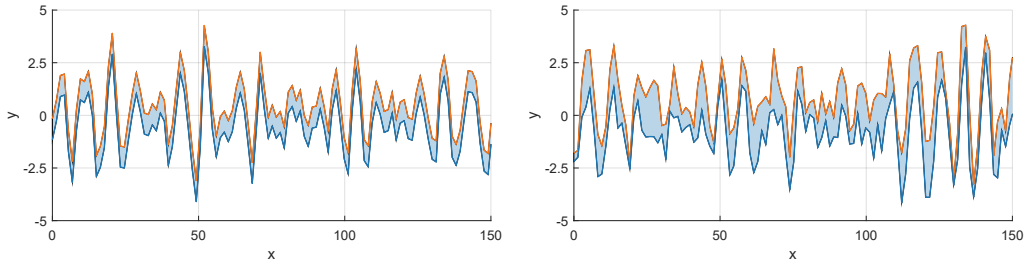

Figure 1: Two signals with poor precision: constant interval uncertainty $\xi = 0.5$ (left), variable interval uncertainty between $\xi_{min} = 0.1$ and $\xi_{max} = 1.5$ (right).

2.3. Interval extensions

The DFT converts a signal $x = x_0, x_1, ..., x_{N-1}$ to its Fourier sequence $z = z_0, z_1, ..., z_{N-1}$, for $k = 0, ..., N-1$.

The *interval extension* of the DFT is obtained by replacing the real signal with their interval values. The resulting interval Fourier transform at frequency number k is

$$\overline{z}_k = \sum_{n=0}^{N-1} \overline{x}_n \cdot e^{-\frac{i2\pi}{N}kn} = \sum_{n=0}^{N-1} \overline{x}_n \cdot \left[\cos\left(\frac{2\pi}{N}kn\right) - i \cdot \sin\left(\frac{2\pi}{N}kn\right)\right]. \quad (7)$$

The interval DFT amplitude is

$$\overline{A}_k = |\overline{z}_k| = \sqrt{\left[\sum_{n=0}^{N-1} \overline{x}_n \cdot \cos\left(\frac{2\pi}{N}kn\right)\right]^2 + \left[\sum_{n=0}^{N-1} \overline{x}_n \cdot \sin\left(\frac{2\pi}{N}kn\right)\right]^2}. \quad (8)$$

An alternative interval extension of the DFT can be obtained representing the interval signal as $\overline{x}_n = x_n + \xi\Delta$, i.e. separating interval from non-interval components, which results in

$$\overline{z}_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn} + \sum_{n=0}^{N-1} \xi\Delta \cdot e^{-\frac{i2\pi}{N}kn}, \quad (9)$$

where, $\Delta = [-1, 1]$ is the unitary interval and $\xi \in \mathbb{R}$ is the precision of the interval signal, expressed in the same units as the signal. This extension can also be used to derive algorithms for the interval propagation based on the

zonotope representation as shown in [23, 24], and to make comparisons with other methods, like Monte Carlo, for computing the bounds. Nonetheless, this representation will not be used for the derivation of the main algorithm.

Remark: An *interval extension* is obtained replacing the original expression's variables directly with interval variables. The fundamental theorem of interval analysis states that every interval extension–obtained combining the four rules of arithmetic–is *inclusion monotonic* [19, 25]. This translates into the renowned conservatism of interval analysis for computing the bounds. In other words, the bounds obtained by interval analysis on interval extensions are inclusive thus rigorous.

Remark: A *united extension* is defined as the union of all the images of a function evaluated on all the subsets of a given interval space. The image of a united extension is often not a box, but an arbitrarily-shaped set, which is referred to as the *united set*. In interval analysis, conservatism is often due to approximating the united set with its enclosing box. This kind of conservatism of interval computations arises in functions with repeated variables, or in back-calculation problems, and is also known as the wrapping effect.

2.4. Repeated variables problem

For readers not familiar with the *repeated variables problem* of interval computing, a brief recap is given as follows: Interval computations produce bounds that are *best possible*, i.e. without inflated uncertainty, when the interval variables appear only once in their mathematical expression. For example, the evaluation of a second-degree polynomial $a\overline{\underline{x}}^2 + b\overline{\underline{x}} + c$ with interval arithmetic can result in inflated bounds due to the interval $\overline{\underline{x}}$ repeating twice in the expression. For more details on this problem the reader can be referred to [26].

2.5. PSD function estimation

A stochastic (or random) process is influenced by random phenomena and fluctuations, so that it cannot be described completely deterministically. The value of the stochastic process at any point in time is determined by random variables [27]. An estimation of the stationary PSD function of a stochastic process can be obtained by the periodogram [14, 13], which can be defined by the squared amplitude of the DFT.

The periodogram for a non-interval discrete signal x_n is

$$\hat{S}_X(\omega_k) = \frac{\Delta t^2}{T}\left|\sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N}kn}\right|^2, \tag{10}$$

where Δt is the time step size, T is the total length of the record, n describes the data point index in the record and k is the frequency number of $\omega_k = \frac{2\pi k}{T}$. The interval extension of Eq. (10) in trigonometric form for an interval signal $\overline{\underline{x}}_n$ is

$$\overline{\underline{\hat{S}_X}}(\omega_k) = \frac{\Delta t^2}{T}\left(\left[\sum_{n=0}^{N-1} \overline{\underline{x}}_n \cdot \cos\left(\frac{2\pi}{N}kn\right)\right]^2 + \left[\sum_{n=0}^{N-1} \overline{\underline{x}}_n \cdot \sin\left(\frac{2\pi}{N}kn\right)\right]^2\right). \tag{11}$$

Because Eq. (11) can be expressed in terms of the square of the Fourier amplitude as follows: $\overline{\underline{\hat{S}_X}}(\omega_k) = \frac{\Delta t^2}{T}\overline{\underline{A}}_k^2$, and because the Fourier amplitude $\overline{\underline{A}}_k$, is a positive real interval, the interval PSD function $\overline{\underline{\hat{S}_X}}(\omega_k)$ can be computed without inflation, provided that its frequency components are not combined together in further computations.

3. The interval discrete Fourier transform algorithm

The objective of this work is to calculate the *exact* bounds on the interval extension of the Fourier amplitude of Eq. (8), and subsequently on the interval extension of an estimation of the PSD function of Eq. (11). Under no dependency assumption (non-interactivity) interval arithmetic suffices to obtain the exact bounds on the Fourier sequence.

To support this statement, it will suffice to note that Eq. (7) has no repeated variables. Conversely, repeated interval variables occur in the calculation of the interval amplitude spectrum of Eq.(8), thus additional computational investment is needed. In order to obtain the exact bounds, the finite set of complex pairs determining the boundary of the *united set* of the DFT is needed. Because the DFT is a linear map, its image under the interval constraints, is a compact set called united set. In this section, an algorithm that computes such united set, leading to the exact bounds on the amplitude of the DFT is presented. This algorithm is referred to as the *selective* algorithm.

Because the DFT is a linear map, the united set can be obtained by endpoints analysis on Eq. (7). Tracking all the endpoints in Eq. (7) allows the representation of the united extension in terms of its finite boundary, but this task has exponential complexity $O(2^N)$, with N being the cardinality of the set of intervals, which coincides with the length of the signal in this study. So, a better algorithm is needed to track down the endpoints whose image through the DFT lies on the boundary of the united set. Fig. 2 shows the united set (octagon), the interval extension (box), the image through the DFT of the endpoints (blue dots). Among the latter, those on the boundary of the united set are depicted with orange diamond-shaped markers. The interval extension is obtained applying the rules of interval arithmetic presented in Section 2 to Eq. (7).

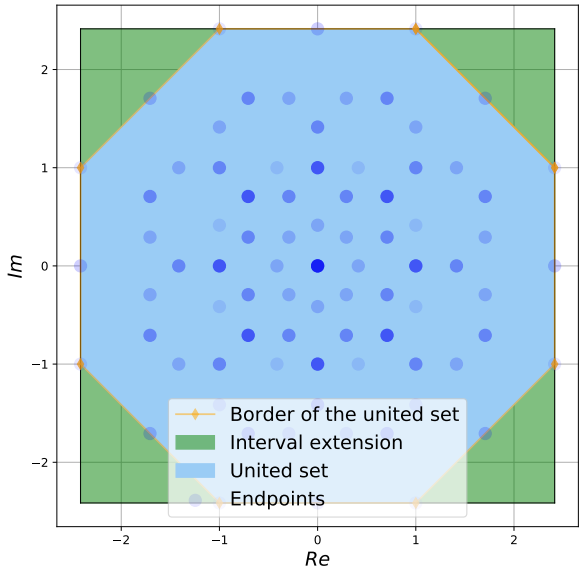

Figure 2: United set (light-blue octagon), interval extension (green box), endpoints image (blue dots), endpoints image on the boundary of the united set (orange diamonds) of a short signal of length $N = 8$.

The selective algorithm is tasked with the following three main steps:

1. Select the set of complex pairs on the boundary (vertices) of the united set, and track down the corresponding endpoints.
2. Select the two vertices that are farthest and nearest to the origin of the complex plane. These are named *anchoring points*. If the united set contains the origin of the complex plane, then return the origin as the nearest point.
3. Compute the absolute value of these two precise complex numbers to obtain the resulting interval amplitude. The absolute value of the farthest and nearest point is the upper and lower bound respectively.

These three steps are repeated for each frequency number k in (8), to construct the interval Fourier spectrum.

3.1. Obtaining the boundary of the united set

In this section, the procedure used by the *selective* algorithm to track down the endpoints whose image is on the boundary of the united set is presented. To understand the procedure, the Fourier sum of Eq. (7) can be conceptualised as a sequence of elementary additions. Each addend of this series, $\overline{\underline{x}}_n \cdot e^{-\frac{i2\pi}{N}kn}$, is an interval in the complex plane, whose real and imaginary components are perfectly dependent. In order to see that real and imaginary components are perfectly dependent, the $(n+1)_{\text{th}}$ addend for a given frequency number k in trigonometric form can be written as:

$$\overline{\underline{z}}_{k,n} = \overline{\underline{x}}_n \cos\left(\frac{2\pi}{N}kn\right) - i\ \overline{\underline{x}}_n \sin\left(\frac{2\pi}{N}kn\right).$$

From the trigonometric form, it appears evident that real and imaginary components are the same interval $\overline{\underline{x}}_n$ differing by just a multiplicative factor.

In the complex plane, the interval addend $\overline{\underline{z}}_{k,n}$ is better represented by an oriented segment rather than a box, as shown in Fig. 3. The Fourier series can be imagined as the sum of such dependent interval objects. What happens when two such objects are added together? Because a segment is a convex bounded set, the sum between two such sets is another convex set, whose vertices are obtained by summing the addends' vertices. Will all the vertices end up on the boundary of the resulting convex set? The answer is no, but why? Although there is an intuition as to which pairs of vertices will reach the boundary before the addition is done, the algorithm adds together all the vertices, and a posteriori discards the pairs of vertices that end up in the interior of the united set. An important theorem in computational

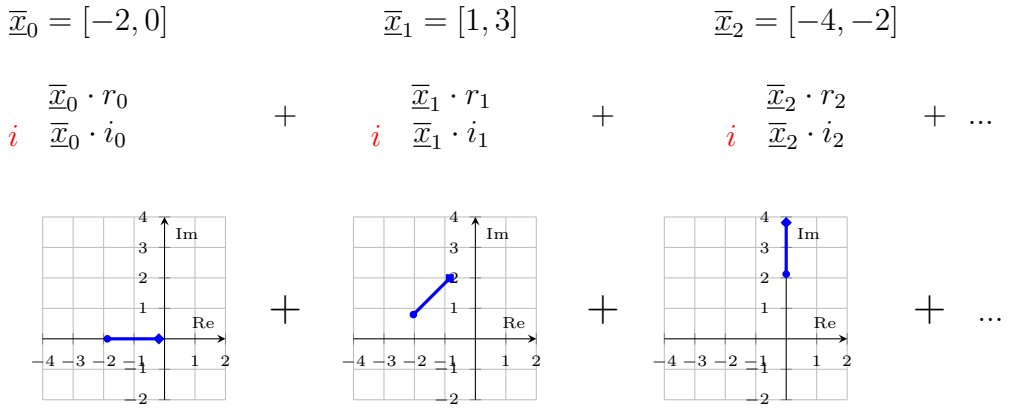

Figure 3: First three addends of the interval Fourier series seen as the sum of perfectly dependent intervals, where $r_n = \cos\left(\frac{2\pi}{N}kn\right)$ and $i_n = \sin\left(\frac{2\pi}{N}kn\right)$ for $N = 8$, and $k = 5$.

geometry, that has led to the Minkowski addition, guarantees that the points in the interior are necessarily mapped in the interior of the resulting polygon. In practice, the interior points are discarded by constructing the convex hull of the set of vertices mapped through each addition, followed by the indexing of only the points on the border, which will be subsequently kept in memory. The polygon resulting from adding two dependent intervals—shown in Fig. 4—is then added to the third dependent interval as shown in Fig. 5. This time, the addition generates two polygons, whose convex hull is the resulting united set. Once again, all the mapped vertices that end up in the interior are removed from memory, whilst the process continues until each addend of the Fourier sum has been processed.

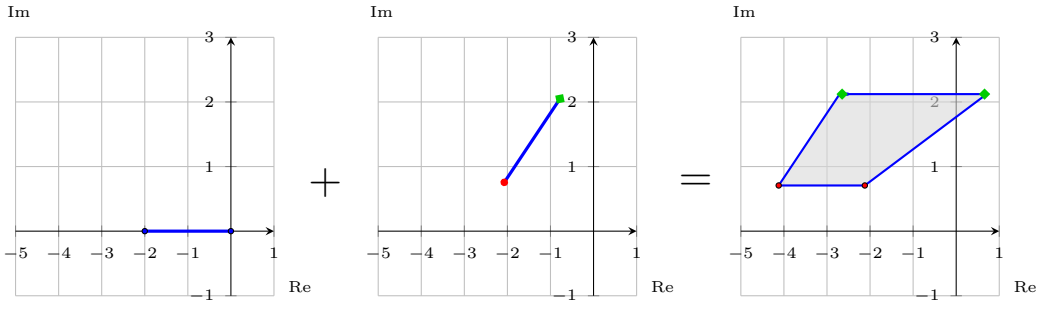

Figure 4: The first addition of the Fourier series results in a polygon with four vertices.

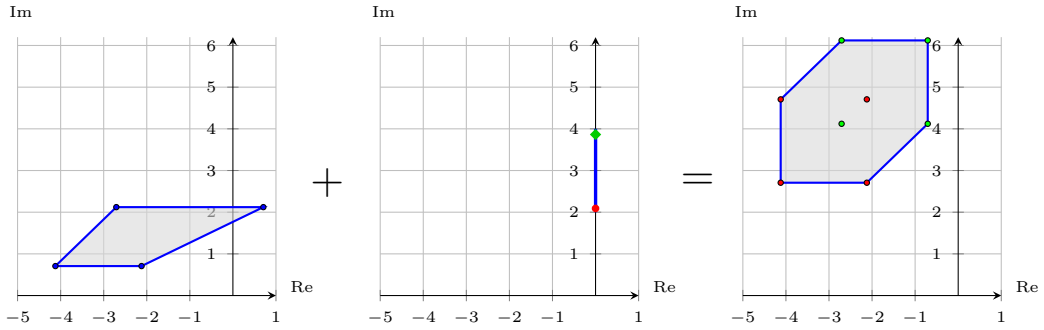

Figure 5: The second addition is the Minkowski sum of a polygon (resulting from the first addition) and the third dependent interval in the Fourier series. The two vertices mapped in the interior of the resulting polygon will be discarded.

The key steps performed by the *selective* algorithm are also presented in the Pseudocode (1) and are described in detail in Appendix A.

3.1.1. Verification of the selective algorithm

The interval extension of the DFT can be used as verification tool to ensure the correctness of the algorithm on arbitrary signals. This is possible thanks to the fact that the bounding box tightly circumscribe (see Fig. 2) the united set because there are no repeating variables in Eq. (7). In other words, for the verification of the algorithm, it is sufficient to show that the boundary of the united set is always resting (for some points) on the boundary of the interval extension box. This verification procedure can be done automatically and systematically within the code that implements the algorithm. This result can be appreciated in Fig. 6 where the boundary of the united set, the interval extension box and the endpoints are all depicted in the same plot for four different frequency numbers k. The interval extension algorithm, which is presented in the Pseudocode (2) in Appendix A, can also be used to obtain an outer approximation of the united set, when speed is a priority. This verification algorithm is much faster than the selective algorithm (Pseudocode 1), because no convex hull evaluation ought to take place. More details about the cost of the selective and interval algorithms are provided in Section 3.4.

Pseudocode 1 Selective algorithm

function ExactAmplitudeBounds($\underline{\overline{x}}$, k)
  $N \leftarrow l(\underline{\overline{x}})$ ▷ length of interval signal
  $v \leftarrow [e^{-i2\pi 0k/N}\underline{x}[0],\ e^{-i2\pi 0k/N}\overline{x}[0]]$ ▷ compute first two vertices
  $c_{hull} \leftarrow v$ ▷ initialise convex hull
  for $n \leftarrow 1$ **to** $N-1$ **do**
    $v \leftarrow [e^{-i2\pi nk/N}\underline{x}[n],\ e^{-i2\pi nk/N}\overline{x}[n]]$ ▷ compute subsequent vertices
    $ep \leftarrow$ add the two vertices v separately to the list c_{hull} to get a list of new endpoints
    $c_{hull} \leftarrow$ get convex hull of ep in $\mathbb{R}^2$ with real and imaginary components as coordinates
  end for
  if origin is inside c_{hull} **then**
    $\underline{\overline{A}}_k \leftarrow \text{Interval}(0,\ \max(|c_{hull}|))$
  else
    $\underline{\overline{A}}_k \leftarrow \text{Interval}\left(\min(|c_{hull}|),\ \max(|c_{hull}|)\right)$
  end if
  return $\underline{\overline{A}}_k$ ▷ bounds on the Fourier amplitude for frequency k
end function

3.2. Determining the two anchoring points

The two anchoring points are defined as the farthest and the nearest vertices of the united set from the origin of the complex plane. The absolute value of these two anchoring points yield the minimum and maximum amplitude at any given frequency. This holds under the interpretation that the absolute value, thus the amplitude of Eq. (8), can be seen as the Euclidean distance from the origin of an equivalent point in $\mathbb{R}^2$ whose coordinates are the real and imaginary components. In Appendix B, a simple proof is provided to show that the maximal distance is attained at one of the vertices of the united set.

If the united set at a given frequency number k is denoted by Z_k, and its border by ∂Z_k, the interval amplitude of the Fourier transform can be computed taking the minimum and maximum over all the vertices of the united set,

$$\underline{\overline{A}}_k = \begin{matrix}\max \\ \min\end{matrix} \{|z_k| : z_k \in \partial Z_k\}\,. \tag{12}$$

This is exemplified in Fig. 7 for four different frequencies. When the united

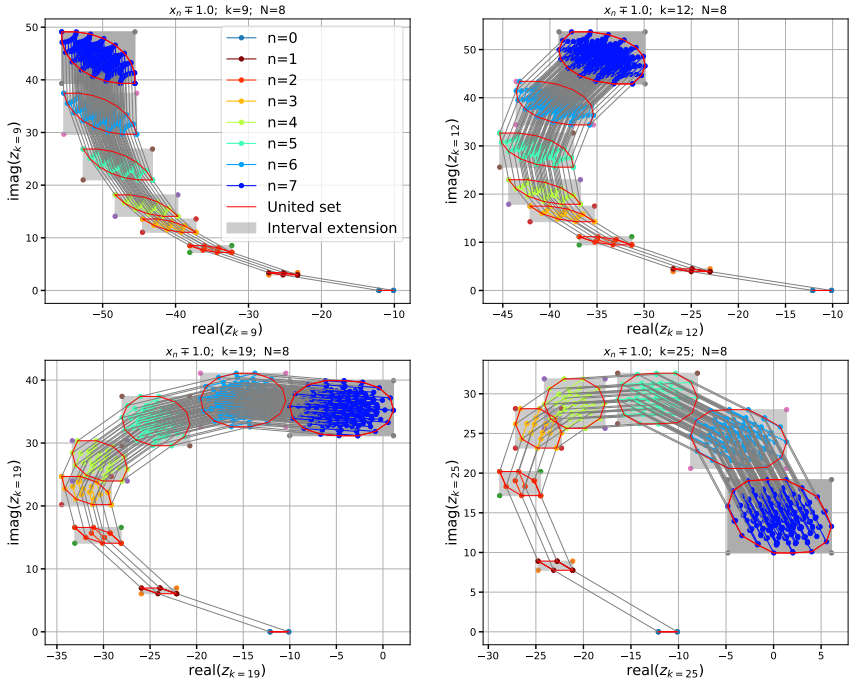

Figure 6: Boundary of the united set in red, image of the endpoints, and interval extension box, at each term of the Fourier series, up to $n = 7$, for frequencies $k \in \{9, 12, 19, 25\}$.

set contains the origin, the lower bound amplitude is zero, see Fig. 8. A function that checks if the origin of the complex plane is contained in the united set is needed to perform this task.

3.3. Obtaining the interval amplitude spectrum

The amplitude spectrum of an arbitrary interval signal is constructed by collecting minimum and maximum amplitude for all frequencies $k \in \{0, ..., N-1\}$, as shown in the previous section. A few examples of this interval spectrum are shown in Fig. 9 for the same simulated signal with different values of interval uncertainty. In Fig. 9 the interval amplitude spectrum obtained with the algorithm (united extension), displayed in dark orange,

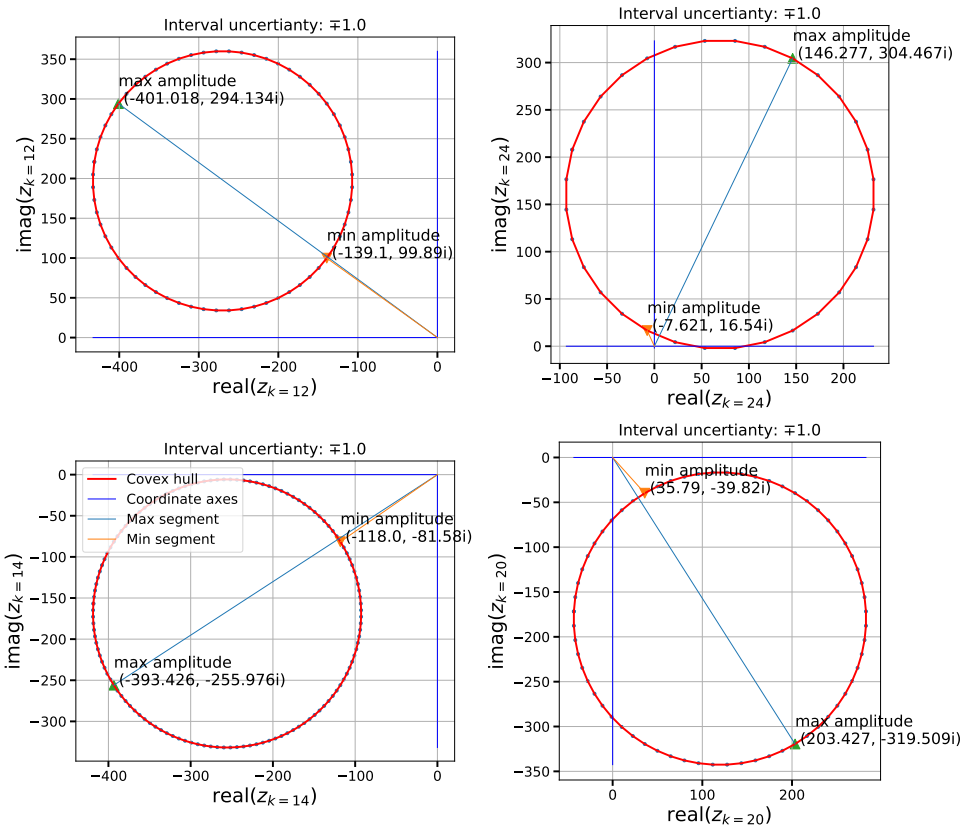

Figure 7: Anchoring points on the united set boundary at frequencies $k \in \{12, 14, 20, 24\}$, for a signal with ± 0.1 $[units]$ imprecision. The anchoring points corresponding to maximum and minimum amplitude are depicted with a triangle marker pointing *up* for the maximum and *down* for the minimum.

is superimposed on the interval amplitude spectrum obtained with interval arithmetic (interval extension), displayed in light blue. It can be seen from Fig. 7 that the number of vertices composing the boundary of the united set differs between frequency numbers. For example, frequency number $k = 24$ has less vertices than frequency number $k = 14$. While the algorithm should produce exactly 2 more vertices at each iteration, for certain frequencies the Fourier coefficient is such that some vertices get projected onto the same vertex. For example, frequency numbers that are a power of two display projections that are always parallel to the coordinate axes, making the resulting united set identified by four vertices only. This behaviour can be appreciated

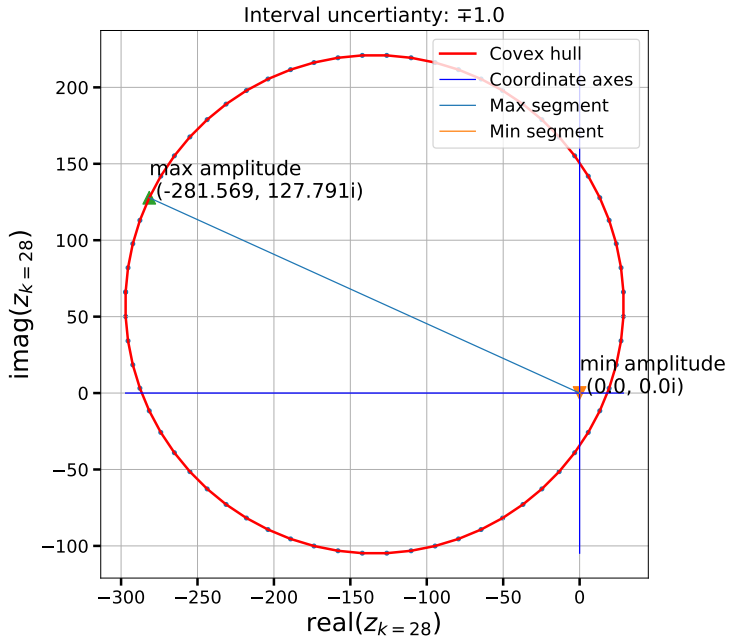

Figure 8: An example of a united set (at frequency $k = 28$) that contains the origin of the complex plane. In this case the minimum amplitude is zero. The anchoring points corresponding to maximum and minimum amplitude are depicted with a triangle marker pointing *up* for the maximum and *down* for the minimum.

in Fig. 9, where the united extension of the spectrum "peaks" at frequencies $k = \{32, 64, 96\}$. Therefore, these peaks are not due to a numerical artefact but are inherent to the process of propagating intervals through a discrete transform. For frequencies that are divisible by two, but have another cofactor like $k = 14 = 2 \cdot 7$ or $k = 24 = 2^3 \cdot 3$, some projections parallel to the coordinate axes make some of the vertices map onto the same one vertex, effectively making those aligned vertices collapse into one. When $N = 2^m$ with $m \in \mathbb{N}$, the Fourier coefficients $\frac{2\pi}{N}kn$ for frequency numbers k that are a power of two: $k = 2^p$, $p \in \mathbb{N}$, $p < m$, correspond to a rotation on the Euler circle with the following angle: $2\pi\frac{n2^p}{2^m} = 2\pi\frac{n}{2^{m-p}}$, which for $p = m - 1$ (or $k = N/2$) is a multiple of π. For such frequency numbers the united set is a rectangle, thus united extension interval extension coincide. This can be appreciated in Fig. 9, where, for frequency number $k = 64$, the interval and

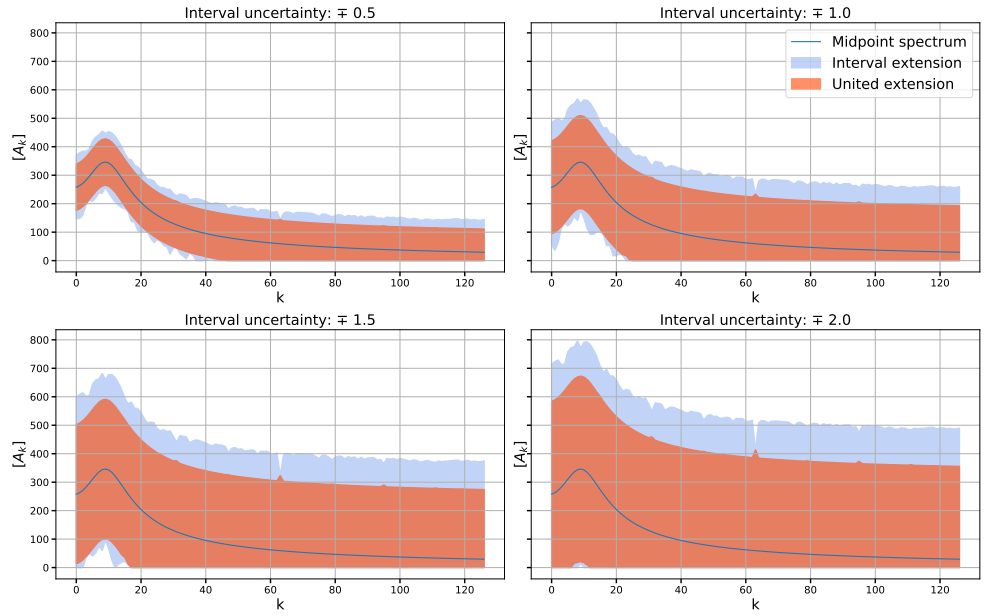

Figure 9: Interval Fourier amplitude spectrum obtained with the proposed algorithm (united extension) in dark orange versus interval spectrum obtained with interval arithmetic (interval extension), displayed in light blue, for signals with constant imprecision $\xi = \{0.5, 1.0, 1.5, 2.0\}$.

united extension meet (upper bound) in one point.

3.4. Computational cost

The selective algorithm (Pseudocode 1) tracks down the endpoints corresponding to the vertices of the united set for each frequency number. This process entails the addition of at most $4k$ complex numbers (twice $2k$ additions). Set aside the convex hull step, a total of at most $4N$ additions are performed at the end of the sum. At each k, however, the selective algorithm computes the convex hull of the set of vertices obtained by the polygon-segment addition depicted in Fig. 5, to eliminate the vertices that lie in the interior of the resulting polygon. Because of the convex hull, each addition step will cost at least $4k \ \log 4k$. In total, because $\sum_k^N k = \frac{N\ (N+1)}{2} \leq N^2, \quad N = 1, 2, ...,$ the cost of obtaining the boundary points is at most $(4N \ \log 4N)^2$ for each frequency number k. For a signal of length N–with $N = 2^m$ a power of two–the DFT outputs exactly $N/2$ frequencies, because the matrix of Fourier coefficients is symmetric and or-

thogonal. Hence, overall the cost of running the selective algorithm is at most $\frac{N}{2}\ (4N\ \log 4N)^2 = 8N^3 \log^2 4N$. Using the big O notation, the cost of the selective algorithm is $O(N^3\ \log^2 N)$, which means that the interval DFT can be run in polynomial time.

In contrast, an exhaustive search of all the endpoints of the interval signal (described in Pseudocode 3) would require 2^k complex additions, so a total of $N2^{2N}$ floating-point operations. In big O notation such a brute-force algorithm costs $O(2^N)$, which makes it unusable in practice.

The evaluation of the interval extension (described in Pseudocode 2), which yields the bounding box that circumscribes the united set, here used for verification, need not compute a convex hull at each addition, so it is much cheaper. The cost of obtaining the interval extension is thus equivalent to the standard DFT algorithm with the only exception that each operation is done within interval analysis. When interval multiplications (or divisions) are involved the cost of the intervalised algorithm is at most four times (eight times with complex intervals) the cost of the standard DFT algorithm. When there are only interval additions (or subtractions) and non-interval multiplications, the cost of an intervalised algorithm is at most two times (four times with complex intervals) the cost of the standard DFT algorithm. In summary, the cost of the intervalised DFT is: $\frac{1}{2}\ N\ 4N = 2N^2$, which in big O notation is $O(N^2)$. Given that the DFT algorithm runs in $O(N^2)$, there is no appreciable additional cost in running its intervalised version.

An overview of the computational costs for the individual algorithms is provided in Table 1.

Table 1: Summary of computational costs

Algorithm	Computational costs
Interval extension	$O(N^2)$
Selective	$O(N^3 \log^2 N)$
Brute-force tracker	$O(2^N)$

4. Technical application

In order to demonstrate the propagation of intervals through the DFT, the pile foundation of an offshore wind turbine excited by sea waves is considered. This model was chosen because it is easy to understand, especially

to demonstrate the proposed method, but still has the characteristics of a real structure. The offshore wind turbine model is depicted in Fig. 10. The total height of the structure is denoted by h_{pile}. Since the structure is hollow from the inside, the cross-section of the structure is specified. The outer radius R and the inner radius r are important parameters to describe the wind turbine. The structure is subjected to the continuous changes of the water height caused by the sea waves. When the water is at rest, the water height is h_{water}, which is indicated by the dashed line. The change in water height due to the sea waves is expressed by $h_{wave}(t)$, which is a time-dependent variable. The total water height including the sea waves is thus $H(t) = h_{water} + h_{wave}(t)$.

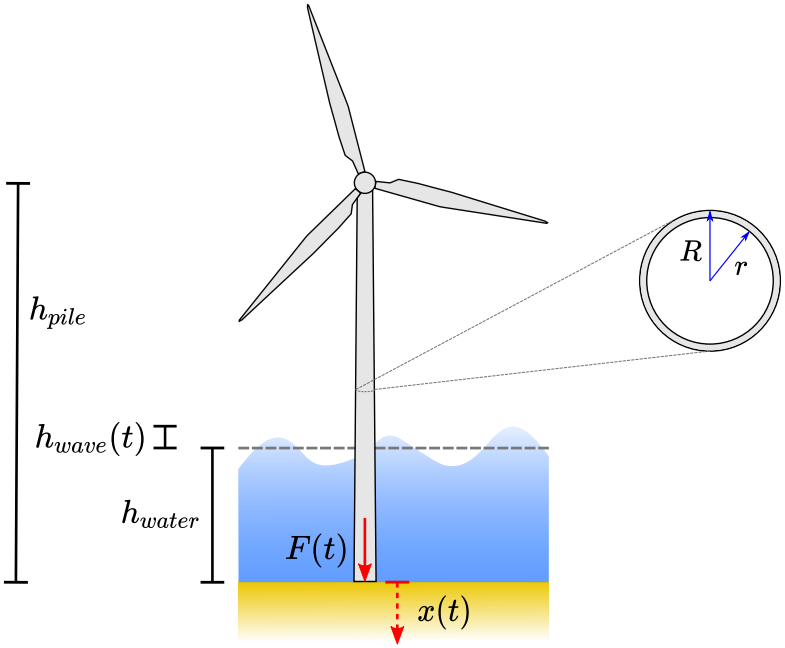

Figure 10: Exemplary illustration of the offshore wind turbine model used in this work.

This structure will be idealised as a Single-Degree-of-Freedom (SDOF) mass-spring-damper system and described by means of the following equation of motion

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t), \tag{13}$$

where m describes the mass of the system, c is the damping coefficient and k denotes the spring constant. x, $\dot{x}$ and $\ddot{x}$ describe the displacement, velocity

and acceleration of the system, respectively. The external excitation of the system is $F(t)$. The following structural parameters are assumed for the wind turbine. The mass m is defined as

$$m = \rho_{\text{steel}}\ h_{\text{pile}}(R^2 - r^2)\pi + \text{CONST} = 1.71 \cdot 10^6 + \text{CONST} \quad [\text{kg}], \tag{14}$$

with $\rho_{\text{steel}} = 7800$ kg/m^3 as the density of steel, $h_{\text{pile}} = 60$ m as the total height of the offshore pile, $R = 3$ m is the outer radius of the pile and $r = 2.8$ m is the inner radius of the hollow pile. The constant CONST in Eq. (14) denotes the weight of any other components of the offshore wind turbine, such as the turbine itself, the blades and other technical devices and equipment. Furthermore, the stiffness is $k = 10^6$ N/m and the damping coefficient is $c = 10^5$ Ns/m.

4.1. Modelling of the excitation by sea waves

The dynamic behaviour of the system is caused by the sea waves. The waves continuously change the water depth and thus the water pressure and the resulting buoyant force on the pile foundation. To model the dynamic behaviour of sea waves in the frequency domain, a PSD function derived within the Joint North Sea Wave Observation Project (JONSWAP) [28] is used, which describes an extension of the Pierson-Moskowitz PSD function [29] and reads as follows

$$S(\omega) = \frac{\alpha g^2}{\omega^5} \exp\left(-\frac{5}{4}\left(\frac{\omega_p}{\omega}\right)^2\right)\gamma^r \tag{15}$$

with

$$r = \exp\left(\frac{-(\omega - \omega_p)^2}{2\sigma^2\omega_p^2}\right). \tag{16}$$

In these equations α describes a spectral energy parameter, g is the gravity acceleration, ω_p describes the peak frequency, γ^r is the peak enhancement factor and σ the spectral width parameter. An example for the JONSWAP PSD function with $\alpha = 0.0081$, $w_p = 0.7$, $\gamma = 3.3$ and

$$\sigma = \begin{cases} 0.7 & \omega \leq \omega_p \\ 0.9 & \omega > \omega_p \end{cases}$$

is given in Fig. 11 (left).

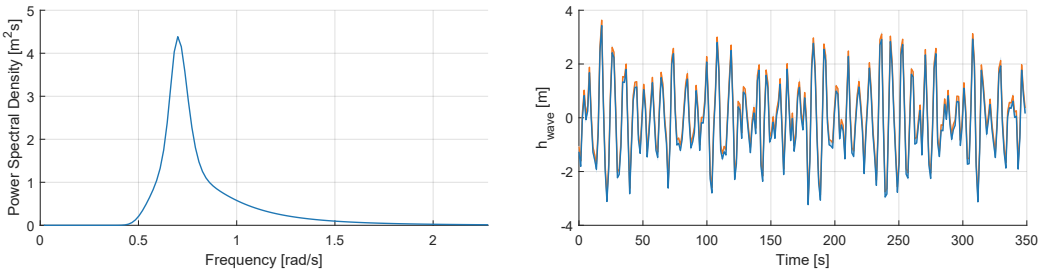

Figure 11: Example for the JONSWAP PSD function (left) and a generated signal of wave heights with interval uncertainty $\xi = 0.1$ m.

The JONSWAP PSD function (Eq. 15) is utilised to generate an artificial signal for the height of the sea waves, which will be the signal under consideration. The spectral representation method (SRM), given in Eq. (17), allows to generate stochastic processes from a given PSD function S_X [30]. It provides a suitable method for generating compatible time signals derived from and carrying the characteristics of an underlying PSD function. The method reads as follows

$$X_t = \sum_{n=0}^{N-1} \sqrt{4 S_X(\omega_n) \Delta\omega}\, \cos(\omega_n t + \varphi_n), \tag{17}$$

with

$$\omega_n = n\Delta\omega, \quad n = 0, 1, 2, \ldots, N-1 \tag{18}$$

where $N \to \infty$, t as time vector and φ_n as uniformly distributed random phase angles in the range $[0, 2\pi]$.

For generating the signal X_t with total length of $T = 350$ s, the time step size $\Delta t = 2\pi/(2\omega_u)$ and frequency step size $\Delta\omega = (2\pi)/T$ are determined according to [30]. The upper cut-off frequency is determined to be $\omega_u = 2.2975$ rad/s. Thus, $\Delta t = 1.367$ s and $\Delta\omega = 0.018$ rad/s. To emulate the poor precision of the resulting signal X_t, it is intervalised using Eq. (6) with interval uncertainty $\xi = 0.1$ m. The entire signal $\overline{X}_t$ is then represented by an interval at each point in time, see Fig. 11 (right). This step is performed here to simulate a real signal with poor precision.

4.2. Computing the interval PSD function and the interval system response

The intervalised signal $\overline{X}_t$ of the wave height, is transformed to the frequency domain as explained in Section 3 by means of the interval extension and the united extension using Eq. (11). Pushing the interval signal $\overline{X}_t$ through Eq. (11) using the algorithm described in Section 3 yields the exact bounds on $\hat{S}_X$,

$$\hat{S}_X(\omega_k) = \frac{\Delta t^2}{T} A_k^2 \tag{19}$$

because $A_k \geq 0$ is non-negative by definition, see Eq. (8), and show no dependence with T and Δt.

The bounds on the interval PSD function $\overline{\hat{S}_X}$ are shown on the left side of Fig. 12. For comparison, the midpoint spectrum is also given, which is calculated via the ordinary DFT from the signal without interval uncertainty. This procedure allows the uncertainties of the interval signal $\overline{X}_t$ to be quantified in the frequency domain and to obtain an upper and lower bound for the respective frequency components.

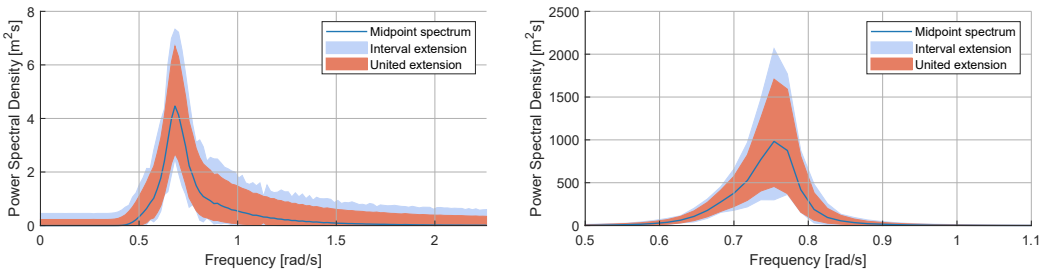

Figure 12: Bounds obtained by the interval extension and the united extension of the signal with interval uncertainty $\xi = 0.1$ m (left) and bounds on the resulting PSD function of the response (right) computed by Eq. (20)

. The bounds derived by interval extension are displayed in light blue, the exact bounds derived by the united extension in dark orange.

After the excitation of the wind turbine by the sea waves, a displacement of the pile foundation can be detected. The PSD function of the system response indicates the frequency components and their amplitude corresponding to the system displacement $x(t)$ of the pile foundation in the frequency domain. The system response is significantly dependent on the

excitation and the system parameters and can be utilised to detect critical system behaviour or system failure. Instead of a numerical solution obtained by extensive Monte-Carlo simulation in time domain, the response of the pile foundation in frequency domain can be directly obtained by the the frequency response function (e.g., [13, 31])

$$S_Y(\omega) = S_X(\omega)|H(\omega)|^2 \tag{20}$$

with $H(\omega)$ as transfer function

$$H(\omega) = \frac{1}{\omega_0^2 - \omega + 2\zeta\omega_0\omega i}. \tag{21}$$

In these equations $\omega_0 = \sqrt{k/m}$ describes the natural frequency of the system, $\zeta = c/(2\omega_0 m)$ is the damping ratio and i is the imaginary unit. The PSD function of the excitation is denoted by S_X and the PSD function of the response is S_Y.

The natural frequency of the system thus results in $\omega_0 = 0.7657$ rad/s and the damping ratio is $\zeta = 0.0383$. The constant CONST in Eq. (14) will be neglected for illustration purposes.

The upper and lower bounds of the interval PSD function of the response $\underline{\hat{S}_Y}$ can be determined separately by interval arithmetic using the singleton-interval multiplication (Eq. 3) for the frequency response function (Eq. 20) on the upper and lower bounds of the interval PSD function $\underline{\hat{S}_X}$. The uncertainty in the original signal of wave heights is thus propagated to the PSD function of the response $\underline{\hat{S}_Y}$ and determined by their upper and lower bounds, see Fig. 12 (right). The determination of a bounded system response in the frequency domain thus leads to a significant improvement in the evaluation of system responses in general, taking into account the uncertainties in the input signal. For each frequency component, an upper and lower bound is now identified in the response, which can be employed to determine whether possible responses are within an acceptable window or result in dangerous system behaviour. This procedure thus makes it possible to propagate the uncertainty of the input signal to the system response and consequently to detect critical or dangerous system behaviour taking these uncertainties into account. In particular, the range of natural frequencies of a system can thus be assessed more accurately in this way.

4.3. Comparison with Monte Carlo

To determine the uncertainty of an interval signal, random signals within the bounds of the interval signal are usually sampled using Monte Carlo (MC). In this example, a signal with constant interval uncertainty $\xi = 0.5$ m is used. Within the interval bounds of the signal $\overline{X}_t$, any arbitrary signal is possible due to the definition of intervals. For illustration, a set of 20 sampled signals within the bounds is shown in Fig. 13. Each of these realisations is transformed separately using Eq. (10) to obtain the PSD function. The maximum and minimum at each frequency can then be determined from all individual transforms to obtain the envelope. These are the extrema of the PSD function determined via MC. A comparison of these extrema and the bounds determined by the interval and united extension is depicted in Fig. 13. To obtain the MC extrema a total of 10^6 sample signals were generated and transformed.

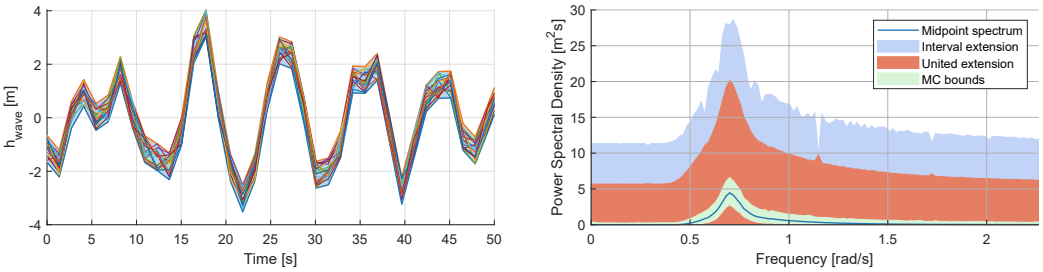

Figure 13: Signal with interval uncertainty $\xi = 0.5$ m and randomly sampled signals within the bounds (left), resulting PSD function bounds obtained by the interval extension in light blue, the united extension in dark orange and MC with 10^6 samples in light green (right).

It can be clearly seen that the minimum/maximum determined by MC are clearly suboptimal compared to the bounds determined by the united extension and interval extension. Since the united extension produces exact bounds, it can be concluded that the determination of the bounds with MC severely underestimates the uncertainty of the signal, which leads to major problems and potential hazards in reliability assessment.

The MC method is tasked to identify the local minimum and maximum while complying with the interval constraints. The number of MC samples determines how close these two extrema are to the exact bounds. In principle, the more MC samples are utilised the closer the minimum and maximum are to the exact bounds. However, the number of MC samples that provides

enough coverage can be prohibitively high, as shown in Fig. 13, where 10^6 MC samples were evaluated. In Fig. 14, the PSD functions were estimated for 3, 10, 100 and 500 sampled signals to show the progression of coverage as the number of MC samples increases. In Fig. 15 the envelopes of 10^1, 10^3 and 10^6 MC samples are compared against each other. This problem of coverage for the MC method has been presented for example in [32, 33]. A numerical optimization study has also been conducted using *fmincon* in Matlab. The minimum/maximum found by the optimizer are significantly better than those obtained with Monte Carlo, whilst also requiring a lot less DFT evaluations. Nevertheless, the minimum and maximum outputted by the optimiser are only known to be local optima, so they are not guaranteed to approximate the exact bounds to the desired accuracy in general.

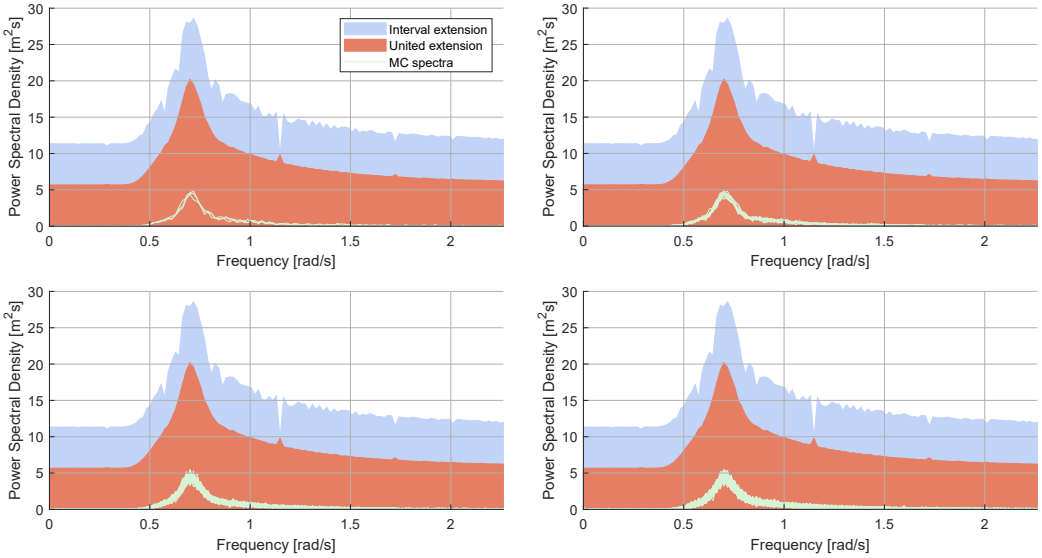

Figure 14: Estimated PSD functions for 3, 10, 100 and 500 sampled signals, against interval PSD function, for a signal with interval uncertainty $\xi = 0.5$ m.

4.4. Sensitivity analysis

To investigate the influence of the interval uncertainty ξ of the signal on the interval width of the PSD functions, a sensitivity analysis is carried out in this section. The interval uncertainty of the signal used in the previous sections was successively increased by 0.1 up to 5.0 and the Fourier transform

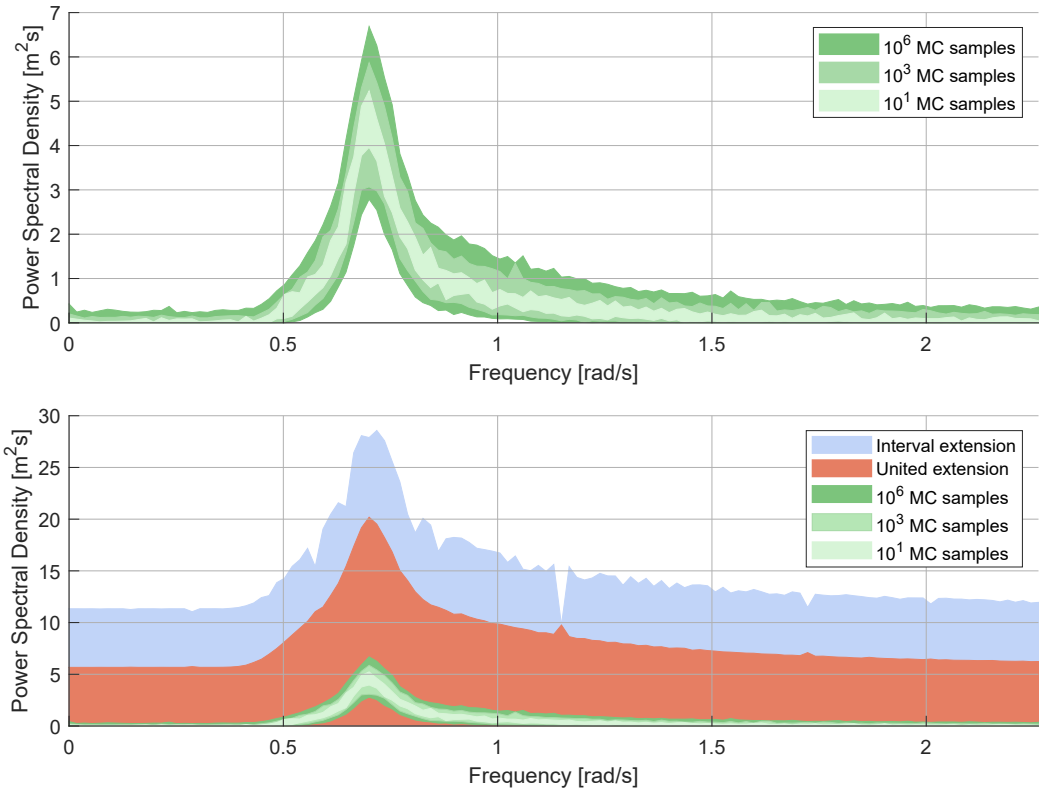

Figure 15: Envelopes of estimated PSD functions for 10^1, 10^3 and 10^6 sampled signals for a signal with interval uncertainty $\xi = 0.5$ m (top) compared against the bounds obtained by the interval extension in light blue and the united extension in dark orange (bottom).

with interval extension and united extension were calculated. For comparison, the interval width of the spectra at the peak frequency of $\omega_0 = 0.7657$ rad/s, which is the natural frequency of the system, is used. The results for both proposed methods are shown in Fig. 16 (left). It should be noted that for an interval uncertainty of $\xi \geq 0.4$, the lower bound is zero for the interval extension. This is shown in Table 2, where some selected values of the interval bounds are presented and is also visible in Fig. 16 (right). The same applies for an interval uncertainty of $\xi \geq 0.5$ for the united extension. Accordingly, the interval width grows only by the increase of the upper bounds. From Fig. 16 (right) a linear trend between widths can be appreciated up to $\xi \geq 0.5$. For values of imprecision greater than $\xi \geq 0.5$, for which the lower bound is

always zero, a non-linear trend between widths can be appreciated instead. Such uncertainty growth can have significant impact on the analysis, since there could be a potentially quadratic increase of the energy content in the PSD function as the uncertainty of the signal grows.

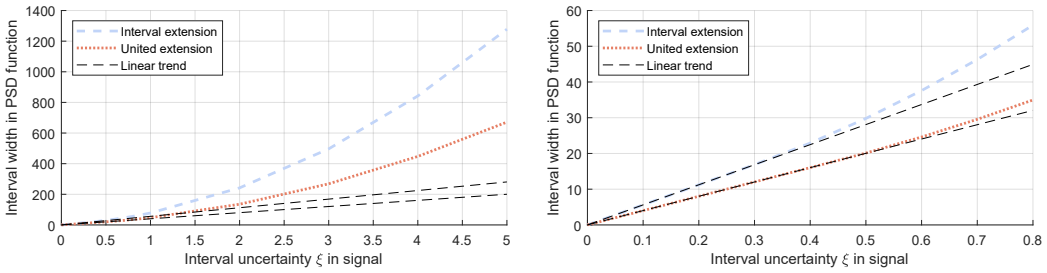

Figure 16: Influence of the interval uncertainty ξ of the signal on the interval width of the PSD function at the peak frequency $\omega_0 = 0.7657$ rad/s (left) and linear trend for smaller interval uncertainties in the signal (right).

Table 2: Interval width and bounds at the peak frequency $\omega_0 = 0.7657$ rad/s for the interval extension and the united extension of the PSD function $\underline{\hat{S}_X}$.

ξ	Interval extension		United extension	
	width($\overline{\underline{\hat{S}_X}}$)	$\overline{\underline{\hat{S}_X}}$	width($\overline{\underline{\hat{S}_X}}$)	$\overline{\underline{\hat{S}_X}}$
0	0	[4.4, 4.4]	0	[4.4, 4.4]
0.1	5.7	[2.0, 7.7]	4.1	[2.6, 6.7]
0.2	11.3	[0.6, 11.9]	8.1	[1.3, 9.4]
0.4	22.9	[0, 22.9]	16	[0, 16.0]
0.8	55.9	[0, 55.9]	35	[0, 35.0]
1	77.8	[0, 77.8]	47.1	[0, 47.1]
2	241.9	[0, 241.9]	135.2	[0, 135.2]
3	496.7	[0, 496.7]	268.6	[0, 268.6]
4	842.1	[0, 842.1]	447.3	[0, 447.3]
5	1278	[0, 1278]	671.3	[0, 671.3]

5. Conclusions

Especially in engineering, the quantification of uncertainties is a matter of paramount importance. In particular, it can be argued that real recorded data are hardly ever exact, being subject to uncertainties due to a wide variety of factors. The algorithm presented in this paper allows interval signals to be projected through the DFT to obtain an upper and lower bound on the Fourier amplitude, which has a significant impact on uncertainty quantification. The propagation of signals can be computed using interval arithmetic, which generally provides outer bounds, as opposed to exact. An algorithm that characterises the geometry of the united set of the DFT is presented. The algorithm fully addresses the repeated variables problem arising in the calculation of the amplitude and therefore yields the exact bounds. Both the interval and the united extension were compared against the MC method. It was shown that the extrema derived via MC severely underestimate the uncertainty. The proposed algorithm is thus more adequately able to account for the uncertainties of signals with poor precision in the Fourier domain. One advantage of this algorithm is that no assumptions have to be made about the error distribution. This leads to a significant improvement in the assessment of engineering problems regarding risk, reliability and uncertainty. The strengths and advantages of the algorithm were illustrated by means of an example involving dynamic structural analysis, in which the bounds of the PSD estimation of an interval signal were calculated. These bounds can be propagated through the system to quantify the response behaviour and the impact of the signal's inherent uncertainty on the quantity of interest.

6. Replicability

The software for computing the interval DFT can be accessed in a single instance via GitHub at: https://github.com/interval-fourier-transform/DFT (last accessed February 2022); whilst the code running the presented examples is available at: https://github.com/interval-fourier-transform/application-to-poor-precision-time-signals (last accessed February 2022). The code, examples and numerical results presented in this paper are therefore fully replicable.

Acknowledgement

This research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) with grant no. BE 2570/4-1 and CO 1849/1-1 and by the Engineering & Physical Sciences Research Council (EPSRC) with grant no. EP/R006768/1. DFG and EPSRC are greatly acknowledged for their funding and support.

References

[1] I. Sneddon, Fourier Transforms, Dover Books on Mathematics, Dover Publications, 1995.

[2] J. F. James, A Student's Guide to Fourier Transforms: With Applications in Physics and Engineering, Student's Guides, 3 ed., Cambridge University Press, 2011. doi:10.1017/CBO9780511762307.

[3] J. W. Cooley, J. W. Tukey, An Algorithm for the Machine Calculation of Complex Fourier Series, Mathematics of Computation 19 (1965) 297–301. URL: http://www.jstor.org/stable/2003354.

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to algorithms, MIT press, 2009.

[5] J. Stoer, R. Bulirsch, Introduction to numerical analysis, volume 12, Springer Science & Business Media, 2013.

[6] M. Beer, S. Ferson, V. Kreinovich, Imprecise probabilities in engineering analyses, Mechanical Systems and Signal Processing 37 (2013) 4 – 29. doi:https://doi.org/10.1016/j.ymssp.2013.01.024.

[7] G. I. Schuëller, On the treatment of uncertainties in structural mechanics and analysis, Computers & Structures 85 (2007) 235–243. doi:https://doi.org/10.1016/j.compstruc.2006.10.009.

[8] S. Ferson, V. Kreinovich, J. Hajagos, W. L. Oberkampf, L. Ginzburg, Experimental uncertainty estimation and statistics for data having interval uncertainty (2007). doi:10.2172/910198.

[9] A. Wiencierz, Regression analysis with imprecise data, Ph.D. thesis, Ludwig-Maximilians-Universität München, 2013. doi:10.5282/edoc.16678.

[10] T. Augustin, F. Coolen, G. de Cooman, M. Troffaes, Introduction to Imprecise Probabilities, Wiley Series in Probability and Statistics, Wiley, 2014.

[11] L. D. Lutes, S. Sarkani, Random Vibrations: Analysis of Structural and Mechanical Systems, Butterworth-Heinemann, 2004.

[12] A. Powell, S. Crandall, Random Vibration, The Technology Press of the Massachusetts Institute of Technology, Cambridge, 1958.

[13] J. Li, J. Chen, Stochastic Dynamics of Structures, John Wiley & Sons, 2009.

[14] D. Newland, An Introduction to Random Vibrations, Spectral & Wavelet Analysis: Third Edition, Dover Civil and Mechanical Engineering, Dover Publications, 2012.

[15] L. Comerford, I. A. Kougioumtzoglou, M. Beer, On quantifying the uncertainty of stochastic process power spectrum estimates subject to missing data, International Journal of Sustainable Materials and Structural Systems 2 (2015) 185–206. doi:https://doi.org/10.1504/IJSMSS.2015.078358.

[16] Y. Zhang, L. Comerford, I. A. Kougioumtzoglou, E. Patelli, M. Beer, Uncertainty quantification of power spectrum and spectral moments estimates subject to missing data, ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering 3 (2017) 04017020. doi:10.1061/AJRUA6.0000925.

[17] G. Liu, V. Kreinovich, Fast convolution and Fast Fourier Transform under interval and fuzzy uncertainty, Journal of Computer and System Sciences 76 (2010) 63–76. doi:https://doi.org/10.1016/j.jcss.2009.05.006, special Issue on Intelligent Data Analysis.

[18] G. Alefeld, J. Herzberger, Introduction to Interval Computation, Computer Science and Applied Mathematics, Elsevier Science, 2012.

[19] R. E. Moore, Interval analysis, volume 4, Prentice-Hall Englewood Cliffs, 1966.

[20] R. Moore, R. Kearfott, M. Cloud, Introduction to Interval Analysis, Cambridge University Press, 2009.

[21] M. De Angelis, M. Behrendt, L. Comerford, Y. Zhang, M. Beer, Forward interval propagation through the discrete Fourier transform, in: The 9th international workshop on Reliable Engineering Computing, 2020, pp. 39–52. arXiv:2012.09778.

[22] M. De Angelis, V. Ricciardi, E. Dalmau, Uncertainty estimation of road-dust emissions via interval statistics, Journal of Physics: Conference Series 1065 (2018) 212023. doi:10.1088/1742-6596/1065/21/212023.

[23] S. Schön, H. Kutterer, Using Zonotopes for Overestimation-Free Interval Least-Squares–Some Geodetic Applications, Reliable Computing 11 (2005) 137–155. doi:https://doi.org/10.1007/s11155-005-3034-4.

[24] O. Kosheleva, V. Kreinovich, Low-Complexity Zonotopes Can Enhance Uncertainty Quantification (UQ), in: Proceedings of the 4th International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2021), 2021, pp. 26–34. doi:https://doi.org/10.7712/120221.8019.18856.

[25] R. E. Moore, Methods and applications of interval analysis, SIAM, 1979.

[26] A. Gray, M. De Angelis, S. Ferson, E. Patelli, What's Z-X, when Z= X+Y? Dependency tracking in interval arithmetic with bivariate sets, in: The 9th international workshop on Reliable Engineering Computing, 2021, pp. 27–38.

[27] B. Priestley, Spectral Analysis and Time Series, Academic Press, 1982.

[28] K. F. Hasselmann, T. P. Barnett, E. Bouws, H. Carlson, D. E. Cartwright, K. Eake, J. Euring, A. Gicnapp, D. Hasselmann, P. Kruseman, et al., Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP), Ergaenzungsheft zur Deutschen Hydrographischen Zeitschrift, Reihe A (1973).

[29] W. J. Pierson Jr., L. Moskowitz, A proposed spectral form for fully developed wind seas based on the similarity theory of S. A. Kitaigorodskii, Journal of Geophysical Research (1896-1977) 69 (1964) 5181–5190. doi:https://doi.org/10.1029/JZ069i024p05181.

[30] M. Shinozuka, G. Deodatis, Simulation of stochastic processes by spectral representation, Applied Mechanics Reviews 44 (1991) 191–204. doi:10.1115/1.3119501.

[31] J. B. Roberts, P. D. Spanos, Random Vibration and Statistical Linearization, Courier Corporation, 2003.

[32] S. Ferson, What Monte Carlo methods cannot do, Human and Ecological Risk Assessment: An International Journal 2 (1996) 990–1007. doi:10.1080/10807039609383659.

[33] M. De Angelis, S. Ferson, E. Patelli, V. Kreinovich, Black-box propagation of failure probabilities under epistemic uncertainty, in: Proceedings of the 3rd International Conference on Uncertainty Quantification in Computational Sciences and Engineering (UNCECOMP 2019), 2019, pp. 713–723. doi:https://doi.org/10.7712/120219.6373.18699.

Appendix A. Pseudocodes

The provided Pseudocodes 1 and 2 describe the computation of the interval Fourier amplitude of an interval signal $\overline{\underline{x}}$ for a specific frequency k with the selective algorithm and with the interval extension using interval arithmetic. Pseudocode 3 describes the tracking of all endpoints for each iteration step. This code is mainly used for illustration purposes and as the foundation of the selective algorithm.

Pseudocode 1: This algorithm, also called the selective algorithm, calculates the Fourier amplitude of an interval signal $\overline{\underline{x}}$ for any frequency k by determining the convex hull of the endpoints. First, the length N of the interval signal is determined. The first two vertices v are calculated by computing the first coefficient for $n = 0$ of the DFT for the first segment of the signal for the upper and lower interval bounds. These are stored in c_{hull} and describe the first two vertices of the endpoint analysis.

Within the for loop, the next vertices v are iteratively computed by calculating the next coefficients for $n = 1, \ldots, N-1$. These two vertices v are added separately to the list c_{hull} to get a new list of endpoints, which are stored in the variable ep. The convex hull is calculated from ep, where the real and imaginary components represent the coordinates in $\mathbb{R}^2$. The vertices forming the convex hull are stored in c_{hull} and all other endpoints

are discarded. This procedure is repeated for each n until the last iteration $n = N - 1$ is reached, thus c_{hull} constitutes the convex hull after each iteration.

Since the minimum and maximum distance from the convex hull to the origin of the coordinate system determine the interval bounds, these are obtained from the absolute values of c_{hull}. If the origin is inside the convex hull, the lower bound is 0, otherwise it is defined by $\min(|c_{hull}|)$. The upper bound is always defined by $\max(|c_{hull}|)$.

The algorithm returns the upper and lower bound on the Fourier amplitude $\overline{\underline{A}}_k$ for the respective frequency k.

Pseudocode 2: In this algorithm interval arithmetic is utilised to achieve the bounds on the Fourier amplitude. This algorithm is mainly used as a verification algorithm for the algorithm presented in Pseudocode 1.

First, the interval extension of the DFT (Eq. 7) of the interval signal $\overline{\underline{x}}$ for the frequency k is determined and stored in $\overline{\underline{z}}_k$. From the complex interval $\overline{\underline{z}}_k$ the absolute values of the vertices of the bounding box are calculated. Those are defined by the 4 possible combinations of upper and lower bound for real and imaginary parts of $\overline{\underline{z}}_k$: $d_{ll} = |\mathrm{Re}(\underline{z}_k) + i\,\mathrm{Im}(\underline{z}_k)|$, $d_{lh} = |\mathrm{Re}(\underline{z}_k) + i\,\mathrm{Im}(\overline{z}_k)|$, $d_{hl} = |\mathrm{Re}(\overline{z}_k) + i\,\mathrm{Im}(\underline{z}_k)|$ and $d_{hh} = |\mathrm{Re}(\overline{z}_k) + i\,\mathrm{Im}(\overline{z}_k)|$.

Since the minimum and maximum distances to the origin of the coordinate system define the interval bounds, these will be determined: The upper bound is the maximum value of all distances to the origin described by $\overline{A}_k = \max(d_{ll},\ d_{lh},\ d_{hl},\ d_{hh})$. The lower bound is determined by considering different cases: If the origin of the coordinate system is inside $\overline{\underline{z}}_k$, 0 is determined as the lower bound. If $\mathrm{Re}(\underline{z}_k) \leq 0 \leq \mathrm{Re}(\overline{z}_k)$, then the lower bound will be defined as $\underline{A}_k = \min\left(|\mathrm{Im}(\underline{z}_k)|\,, |\mathrm{Im}(\overline{z}_k)|\right)$. If $\mathrm{Im}(\underline{z}_k) \leq 0 \leq \mathrm{Im}(\overline{z}_k)$, the lower bound will be $\underline{A}_k = \min\left(|\mathrm{Re}(\underline{z}_k)|\,, |\mathrm{Re}(\overline{z}_k)|\right)$, and otherwise the lower bound is determined to be $\underline{A}_k = \min\left(d_{ll},\ d_{lh},\ d_{hl},\ d_{hh}\right)$.

The algorithm returns an outer approximation of the interval bounds $\overline{\underline{A}}_k$ of the Fourier amplitude for a given frequency k.

Pseudocode 3: This algorithm describes the tracking of all endpoints using a brute-force method. The idea behind the code is the same as in Pseudocode (1). The difference is that the convex hull is not determined in each iteration and therefore no endpoints are discarded. Instead, all endpoints ep are stored in a binary *tree* for all N and are passed on to the next iteration. During this procedure the number of endpoints becomes significantly larger than in the selective algorithm, as the number of endpoints increases exponentially in base two and with exponent given by the iteration number.

The total number of iterations is determined by length N of the interval signal. The resulting bounds for the Fourier amplitude are the same as in Pseudocode (1), but for practical simulations this code is not feasible as it is too inefficient to be used for long signals. Therefore, this code is herein mainly used for illustration purposes and as the foundation of the selective algorithm.

Pseudocode 2 Verification algorithm with rigorous interval arithmetic

function IntervalAmplitudeBounds($\underline{\overline{x}}$, k)
  $\underline{\overline{z}}_k \leftarrow \mathrm{DFT}(\underline{\overline{x}},\ k)$ ▷ interval extension of the discrete Fourier transform (Eq. 7) for frequency k
  $d_{ll} \leftarrow |\mathrm{Re}(\underline{z}_k) + i\,\mathrm{Im}(\underline{z}_k)|$
  $d_{lh} \leftarrow |\mathrm{Re}(\underline{z}_k) + i\,\mathrm{Im}(\overline{z}_k)|$
  $d_{hl} \leftarrow |\mathrm{Re}(\overline{z}_k) + i\,\mathrm{Im}(\underline{z}_k)|$
  $d_{hh} \leftarrow |\mathrm{Re}(\overline{z}_k) + i\,\mathrm{Im}(\overline{z}_k)|$ ▷ d_{ll}, d_{lh},d_{hl} and d_{hh} define distances of the vertices of the bounding box to the origin
  $\overline{A}_k \leftarrow \max(d_{ll},\ d_{lh},\ d_{hl},\ d_{hh})$ ▷ get maximum distance to origin
  if origin is inside $\underline{\overline{z}}_k$ **then** ▷ get minimum distance to origin depending on different cases
    $\underline{A}_k \leftarrow 0$
  else if $\mathrm{Re}(\underline{z}_k) \leq 0 \leq \mathrm{Re}(\overline{z}_k)$ **then**
    $\underline{A}_k \leftarrow \min\left(|\mathrm{Im}(\underline{z}_k)|\,, |\mathrm{Im}(\overline{z}_k)|\right)$
  else if $\mathrm{Im}(\underline{z}_k) \leq 0 \leq \mathrm{Im}(\overline{z}_k)$ **then**
    $\underline{A}_k \leftarrow \min\left(|\mathrm{Re}(\underline{z}_k)|\,, |\mathrm{Re}(\overline{z}_k)|\right)$
  else
    $\underline{A}_k \leftarrow \min\left(d_{ll},\ d_{lh},\ d_{hl},\ d_{hh}\right)$
  end if
  $\underline{\overline{A}}_k \leftarrow \mathrm{Interval}(\underline{A}_k,\ \overline{A}_k)$
  return $\underline{\overline{A}}_k$ ▷ bounds on the Fourier amplitude for frequency k
end function

Pseudocode 3 Tracking of all the endpoints

function EndpointsTreeTracker($\underline{x}$, k)
  $N \leftarrow l(\underline{x})$ ▷ length of interval signal
  $v \leftarrow [e^{-i2\pi 0k/N}\underline{x}[0],\ e^{-i2\pi 0k/N}\overline{x}[0]]$ ▷ compute first two vertices
  $ep \leftarrow v$ ▷ initialise endpoints
  $tree[0] \leftarrow ep$
  for $n \leftarrow 1$ **to** $N-1$ **do**
    $v \leftarrow [e^{-i2\pi nk/N}\underline{x}[n],\ e^{-i2\pi nk/N}\overline{x}[n]]$ ▷ compute subsequent vertices
    $ep \leftarrow$ add the two vertices v separately to the list ep to get a list of new endpoints
    $tree[n] \leftarrow ep$ ▷ store n_{th} level of endpoints ep in $tree$
  end for
  return $tree$ ▷ endpoints of the Fourier sum for each iteration
end function

Appendix B. Anchoring points

The maximum amplitude of the DFT is the maximum distance from the boundary of the united set to the origin of the complex plane. This holds under the interpretation that a complex number is a point in $\mathbb{R}^2$, whose coordinates are the real and imaginary components, and its complex absolute value is the Euclidean distance of that point from the origin. Because the boundary of the united set consists of a finite number of vertices, in this Appendix, it will be sufficient to show that the farthest point to the origin of the plane is attained at the vertices and not on any of the connecting edges.

Proposition 1. *Let $A, B \in \mathbb{R}^2$ two points on the real plane, O the origin of the plane, and $\overrightarrow{AB}$ the segment connecting the two points. For every point $P \in \overrightarrow{AB}$, it holds that*

$$\left\|\overrightarrow{OP}\right\| \leq \max\left(\left\|\overrightarrow{OA}\right\|, \left\|\overrightarrow{OB}\right\|\right). \tag{B.1}$$

Proof 1. *It is well known in Linear Algebra that the difference between two points yields the oriented segment between the two points:*

$$\overrightarrow{OB} - \overrightarrow{OA} = \overrightarrow{AB}. \tag{B.2}$$

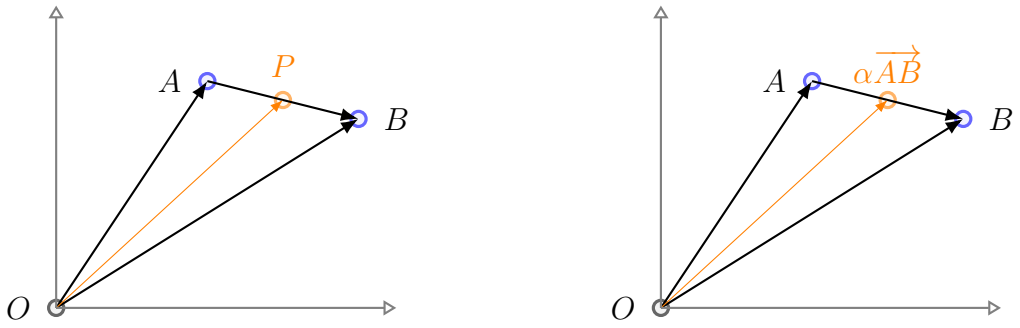

Figure B.17: Two vertices on the boundary of the united set. On the right, the point P in the interior is obtained multiplying the vector $\overrightarrow{AB}$ times a scalar $\alpha \in [0, 1]$.

Any point $P = \overrightarrow{OP}$ *in the interior of the segment* $\overrightarrow{AB}$*, see Fig. B.17, can be expressed as:*

$$\overrightarrow{OP} = \overrightarrow{OA} + \alpha\overrightarrow{AB}, \quad \text{where } \alpha \in [0, 1]. \tag{B.3}$$

Therefore, any point in the interior of the segment is

$$P = \overrightarrow{OP} = \overrightarrow{OA} + \alpha\left(\overrightarrow{OB} - \overrightarrow{OA}\right), \tag{B.4}$$

Assuming that $\left\|\overrightarrow{OB}\right\| \geq \left\|\overrightarrow{OA}\right\|$*, so that* $\left\|\overrightarrow{OB}\right\|$ *is maximum. Then, primarily because* $\alpha \in [0, 1]$*, it holds*

$$(1-\alpha)\left\|\overrightarrow{OA}\right\| + \alpha\left\|\overrightarrow{OB}\right\| \leq \left\|\overrightarrow{OB}\right\|. \tag{B.5}$$

The left-hand side of (B.5) *satisfies the Triangle Inequality:*

$$\left\|(1-\alpha)\overrightarrow{OA} + \alpha\ \overrightarrow{OB}\right\| \leq (1-\alpha)\left\|\overrightarrow{OA}\right\| + \alpha\left\|\overrightarrow{OB}\right\|. \tag{B.6}$$

Therefore, it holds

$$\left\|(1-\alpha)\overrightarrow{OA} + \alpha\ \overrightarrow{OB}\right\| \leq \left\|\overrightarrow{OB}\right\|. \tag{B.7}$$

If the left-hand side of (B.7) *is rearranged as follows, the original statement is proven:*

$$\left\|\overrightarrow{OA} + \alpha\left(\overrightarrow{OB} - \overrightarrow{OA}\right)\right\| = \left\|\overrightarrow{OP}\right\| \leq \left\|\overrightarrow{OB}\right\|. \quad \blacksquare \tag{B.8}$$