# 4

# Interval Arithmetic

## 4.1 Introduction

The fundamental idea behind the interval arithmetic (IA) is that the values of a variable can be expressed as ranging over a certain interval. If one computes a number $A$ as an approximation to some unknown number $X$ such that $|X - A| \leq B$, where $B$ is a precise bound on the overall error in $A$, we will know for sure that $X$ lies in the interval $[A-B, A+B]$, no matter how A and B are computed. The idea behind IA was to investigate computations with intervals, instead of simple numbers.

In fact, when we use a computer to make real number computations, we are limited to a finite set of floating-point numbers imposed by the hardware. In these circumstances, there are two main options for approximating a real number. One is to use a simple floating point approximation of the number and to propagate the error of this approximation whenever the number is used in a calculation. The other is to bind the number in an interval (whose ends may also be floating point values) within which the number is guaranteed to lie. In the latter case, any calculation that uses the number can just as well use its interval approximation instead. This chapter deals with computations involving two floating-point numbers as intervals—the subject covered by interval arithmetic. Approximations carried out with a single floating-point number are studied in the next chapter.

Interval arithmetic, also known as interval mathematics, interval analysis, or interval computation, has been developed by mathematicians and computer scientists since the late 1950s and early 1960s as an approach to putting bounds on rounding errors in arithmetic computations. In this respect, Ramon Moore's PhD thesis [278], as well as his book [279] and other papers published *a posteriori*, played an important role in the development of interval arithmetic. Interval analysis is now a field of study in itself, widely used in numerical analysis and geometric modelling, as well as many other computation processes which require some guarantee in the results of calculations.

Other relevant references in interval arithmetic can be found in the literature. For example, Alefeld and Herzberger [4] propose using intervals in Newton-like methods for finding the roots of univariate functions and also in solving systems of equations. Neumaier [299] takes the concept of intervals further and develops distance definitions and topological properties for intervals. Methods for finding enclosures for the range of a function are given, as well as interval-based methods for solving systems of equations.

Apart from the classical way of looking at intervals, other approaches exist whereby the interval is regarded as an approximation of its centre. Ratschek and Rockne [335], as well as Neumaier [299], discuss the use of *centred-form intervals*. Comba and Stolfi [90] and Andrade et al. [17] take this approach even further in their *affine arithmetic*. Affine arithmetic (AA) still regards the interval as an approximation of the number at its centre, but at the same time keeps track of the various levels of error affecting the computed quantity at different steps of the evaluation of an expression. Their results are quite encouraging, in that they are tighter than the ones produced by the traditional IA. But as expected, there is a tradeoff between accuracy and computation time cost.

IA and AA are also used in research areas such as computer graphics and geometric modelling. At our best knowledge, Suffern and Fackerell [379] and Snyder [370] were who first introduced interval arithmetic in these research areas. For example, Snyder [370] explains the advantages of using IA in geometric modelling as opposed to approaching global problems by finding roots of polynomials. The main point he makes is that IA controls the approximation errors during the floating-point computation by computing bounds rather than exact values. The other major advantage of using interval methods is that they are exhaustive and can give information about the whole region of interest. Other references in scientific computing, computer graphics and geometric modelling include de Figueiredo and Stolfi [102], Heidrich et al. [181], Cusatis et al. [99], Voiculescu [401], Martin et al. [259], Bowyer et al. [62, 63], Bülher and Barth [72], Michelucci [270], Bülher [71], Shou et al. [367], Figueiredo et al. [103], Fang et al. [131], Paiva et al. [315], and Miyajima and Kashiwagi [275].

In this chapter, we look at the IA and AA rules, and describe how they can be used in geometric modelling. This is important because geometric modelling not only involves high precision calculations, but also uses intervals in order to denote and study regions of space, regardless of whether they contain implicit curves, surfaces or solids. For example, a point can be approximated by the intervals that give bounds for its coordinates. Hence a neighbourhood in the shape of a box describes the region of space where that point is guaranteed to lie. Evaluating the function at that point (or some similar potential value for the whole box of coordinate ranges) has geometrical meaning: it is a measure of how far away the point (or the box) is from the surface represented by the function. This measurement is only relative, as the function value merely indicates which of several points is closer to a given surface but it does not actually help evaluate the distance from a point to the surface.

## 4.2 Interval Arithmetic Operations

The execution of an automatic computation usually involves the propagation of inaccuracies and rounding errors, because floating point values are merely rational approximations of real numbers. If interval ranges are used instead of a single approximation, then an automatic computation results in a range of possible values for the final solution. This solution is generally described by means of an interval. Once again, this is only one intuitive motivation for using intervals and introducing arithmetic operations on the set of intervals. The exact way in which intervals are used in geometric modelling is explained later.

### 4.2.1 The Interval Number

Owing to Moore's work, the mathematical concept of number has been generalised to the ordered pair of its approximations—*the interval number*. An interval number $\mathbf{x}$ is denoted as the ordered pair of reals $[\underline{x}, \overline{x}]$, $\underline{x} \leq \overline{x}$, which defines the set of real numbers

$$[\underline{x}, \overline{x}] = \{x \mid \underline{x} \leq x \leq \overline{x}\}$$

When one of the extremities of the interval needs to be excluded from the interval set, variations of the following notation are used: $]\underline{x}, \overline{x}] = \{x \mid \underline{x} < x \leq \overline{x}\}$. Either or both extremities of an interval can be excluded from the set by using the appropriate inequalities. This particular notation has the advantage of distinguishing between the open interval $]\underline{x}, \overline{x}[$ and the pair of numbers $(\underline{x}, \overline{x})$.

### 4.2.2 The Interval Operations

The rules of arithmetic can be redefined so that they apply to interval numbers. If $\mathbf{x} = [\underline{x}, \overline{x}]$ and $\mathbf{y} = [\underline{y}, \overline{y}]$, and the operator $\odot \in \{+, -, \times, /\}$ then the four elementary arithmetic operations will follow the scheme:

$$\mathbf{x} \odot \mathbf{y} = \{x \ \odot \ y \ : \ x \in \mathbf{x}, \quad y \in \mathbf{y}\}$$

An interval operation must produce a new interval containing all the possible results that can be obtained by performing the operation in question on any element of the argument intervals. This template produces simpler specific rules for each of the arithmetic operators (see also Higham [186]):

**Addition:**

$$\mathbf{x}+\mathbf{y}=[\underline{x}+\underline{y},\overline{x}+\overline{y}] \tag{4.1}$$

**Subtraction:**

$$\mathbf{x}-\mathbf{y}=[\underline{x}-\overline{y},\overline{x}-\underline{y}] \tag{4.2}$$

**Multiplication:**

$$\mathbf{x}\times\mathbf{y}=[\min\{\underline{x}\,\underline{y},\underline{x}\overline{y},\overline{x}\,\underline{y},\overline{x}\overline{y}\},\max\{\underline{x}\,\underline{y},\underline{x}\overline{y},\overline{x}\,\underline{y},\overline{x}\overline{y}\}] \tag{4.3}$$

**Division:**

$$\mathbf{x}/\mathbf{y}=[\min\{\underline{x}/\underline{y},\underline{x}/\overline{y},\overline{x}/\underline{y},\overline{x}/\overline{y}\},\max\{\underline{x}/\underline{y},\underline{x}/\overline{y},\overline{x}/\underline{y},\overline{x}/\overline{y}\}] \tag{4.4}$$

Depending on the circumstances in which interval division is used, it may be appropriate to declare division by an interval containing zero as *undefined* or to express it as a union of two semi-infinite intervals.

Interval division can also be written as follows:

$$\mathbf{x}/\mathbf{y}=\mathbf{x}\times\frac{1}{\mathbf{y}} \tag{4.5}$$

where

$$\frac{1}{\mathbf{y}}=\begin{cases}\left[\frac{1}{\overline{y}},\frac{1}{\underline{y}}\right] & \text{if } \underline{y}>0 \text{ or } \overline{y}<0\\ \left]-\infty,\frac{1}{\underline{y}}\right] \cup \left[\frac{1}{\overline{y}},\infty\right[ & \text{if } \underline{y}\le 0\le\overline{y}\end{cases}$$

The addition and multiplication operations are commutative, associative and subdistributive. The subdistributivity property comes from that fact that the set $\mathbf{x}(\mathbf{y}+\mathbf{z})$ is a subset of $\mathbf{xy}+\mathbf{xz}$.

An additional operation is the exponentiation of an interval. Interestingly, it is defined differently from number exponentiation as follows:

**Exponentiation:**

$$[\underline{x},\overline{x}]^{2n+1}=\left[\underline{x}^{2n+1},\overline{x}^{2n+1}\right] \tag{4.6}$$

$$[\underline{x},\overline{x}]^{2n}=\begin{cases}\left[\underline{x}^{2n},\overline{x}^{2n}\right] & if\ \ 0\le\underline{x}<\overline{x}\\ \left[0,M^{2n}\right] & if\ \ \underline{x}<0\le\overline{x}\\ \left[\overline{x}^{2n},\underline{x}^{2n}\right] & if\ \ \underline{x}<\overline{x}<0\end{cases} \tag{4.7}$$

where $n$ is any natural number and $M = \max\{|\underline{x}|, |\overline{x}|\}$.

In particular, for even values of $k$,

$$[\underline{x}, \overline{x}]^k \neq \underbrace{[\underline{x}, \overline{x}] \times \cdots \times [\underline{x}, \overline{x}]}_{k}$$

which is proved by a simple counterexample:

$$\begin{aligned} [-1, 2] \times [-1, 2] &= [-2, 4] \\ [-1, 2]^2 &= [0, 4] \end{aligned}$$

The interval resulting from an even power exponentiation is always *entirely positive* (even when the interval which is being raised to the even power contains negative numbers).

## 4.3 Interval Arithmetic-driven Space Partitionings

Interval arithmetic is especially useful in geometric modelling when objects (e.g. points, curves, surfaces, and solids) are represented by implicit functions and are categorised by means of space partitioning. As seen above, an interval can be regarded as an entity which gives lower and upper approximations of a number. Since a point in Euclidean space is a pair of real coordinates in 2D (respectively, a triplet in 3D), it can be naturally approximated by a pair (respectively, triplet) of intervals, i.e. an axially aligned *box*. Thus, the classical point membership testing method used in geometric modelling can be extended to a *box testing method*.

We are then able to combine interval arithmetic with axially aligned space partitionings to locate objects defined implicitly. This is illustrated in Figure 4.1, where combining interval arithmetic and a 2-d tree space partitioning allows us to locate the following curve defined implicitly as follows:

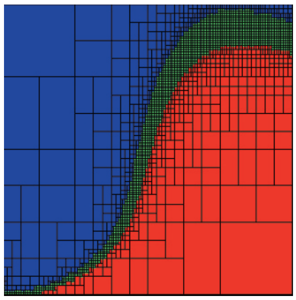


**Fig. 4.1.** An implicit curve specified by the power-form polynomial $p(x, y)$ defined below, in the ambient space $[0, 1] \times [0, 1]$ and using a minimum box size of $\frac{1}{2^7} \times \frac{1}{2^7}$.

$$p(x,y) = \frac{9446}{10{,}000}\,x\,y - \frac{700{,}443{,}214}{100{,}000{,}000}\,x^3\,y^2 + \frac{764{,}554}{100{,}000}\,x^4\,y^3 + \frac{564}{1000}\,y^4 - x^3$$

The curve is somewhere in the region of the green boxes, i.e. those boxes in which $p$ evaluates to an interval that straddles zero. The red boxes denote entirely negative boxes, i.e. boxes in which $p$ evaluates negative everywhere. The blue boxes identify entirely positive boxes, i.e. boxes in which $p$ evaluates positive everywhere.

Here is an example that illustrates this box classification. Given the axially aligned box $[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}]$, the two variables of the curve expression $x$ and $y$ are replaced by the two interval coordinates $[\underline{x}, \overline{x}]$ and $[\underline{y}, \overline{y}]$, respectively. This substitution produces an interval expression which is then evaluated by applying IA rules. This evaluation results in an interval. For example, let us consider the box $[\frac{1}{2}, \frac{5}{8}] \times [0, \frac{1}{8}]$ in Figure 4.1. Substituting $x$ and $y$ by $[\frac{1}{2}, \frac{5}{8}]$ and $[0, \frac{1}{8}]$, respectively, in the expression of $p$, we obtain

$$\begin{aligned} p_{rat}\left(\left[\frac{1}{2}, \frac{5}{8}\right], \left[0, \frac{1}{8}\right]\right) &= \frac{9446}{10{,}000}\left[\frac{1}{2}, \frac{5}{8}\right]\left[0, \frac{1}{8}\right] - \frac{700{,}443{,}214}{100{,}000{,}000}\left[\frac{1}{2}, \frac{5}{8}\right]^3\left[0, \frac{1}{8}\right]^2 \\ &\quad + \frac{764{,}554}{100{,}000}\left[\frac{1}{2}, \frac{5}{8}\right]^4\left[0, \frac{1}{8}\right]^3 + \frac{564}{1000}\left[0, \frac{1}{8}\right]^4 - \left[\frac{1}{2}, \frac{5}{8}\right]^3 \\ &= \left[-\frac{27{,}086{,}029{,}053}{10^{11}}, -\frac{4{,}878{,}689{,}313}{10^{11}}\right] \end{aligned}$$

which is an entirely negative interval; this confirms that the box $[\frac{1}{2}, \frac{5}{8}] \times [0, \frac{1}{8}]$ in Figure 4.1 is correctly depicted red.

## 4.3.1 The Correct Classification of Negative and Positive Boxes

As seen above, there are three types of boxes output by interval arithmetic: negative boxes, positive boxes and zero boxes (i.e. those that depict a region where the function evaluates to an interval that straddles zero). As will be shown later, not all zero boxes contain segments of the curve, i.e. not all boxes classified as zero boxes are genuine zero boxes. The prediction that the box contains at least a curve segment is reasonably accurate only for low-degree polynomials, but problems become manifest when the curve expression is of high degree.

However, we can prove that when a box is labelled as negative or positive it is indeed correctly classified. The proof will be carried out in the one-dimensional case, but can be easily generalised to any number of dimensions.

Given an implicit polynomial equation $f : \mathbb{R} \to \mathbb{R}$, $f(x) = 0$ and a 'box' $[\underline{x}, \overline{x}]$, we will first prove that if the box is labelled as positive then all the points in the box have a positive function value. In other words,

$$f([\underline{x}, \overline{x}]) > 0 \quad \overset{?}{\Longrightarrow} \quad f(x) > 0,\ \forall x \in [\underline{x}, \overline{x}].$$

Since $f(x,y)$ is chosen as a polynomial function, its expression is an algebraic combination of entities involving additions, subtractions, multiplications and exponentiations. Hence its corresponding interval expression will involve similar combinations. All that remains to be proved is that any arithmetic combination that yields a positive interval will yield a positive quantity when the calculation is performed with numbers instead of intervals.

**Addition:**

$$[\underline{x},\overline{x}] + [\underline{y},\overline{y}] = [\underline{x}+\underline{y},\overline{x}+\overline{y}] > 0 \quad \overset{?}{\Longrightarrow} \quad x+y>0,\ \forall x \in [\underline{x},\overline{x}], y \in [\underline{y},\overline{y}]$$

*Proof.*

$$x+y \geq \underline{x}+\underline{y} > 0$$

□

**Subtraction:**

$$[\underline{x},\overline{x}] - [\underline{y},\overline{y}] = [\underline{x}-\overline{y},\overline{x}-\underline{y}] > 0 \quad \overset{?}{\Longrightarrow} \quad x-y>0,\ \forall x \in [\underline{x},\overline{x}], y \in [\underline{y},\overline{y}]$$

*Proof.*

$$x-y \geq \underline{x}-\overline{y} > 0$$

□

**Multiplication:**

$$[\underline{x},\overline{x}] \times [\underline{y},\overline{y}] > 0 \quad \overset{?}{\Longrightarrow} \quad xy>0,\ \forall x \in [\underline{x},\overline{x}], y \in [\underline{y},\overline{y}]$$

*Proof.*

$$\begin{aligned}&[\underline{x},\overline{x}] \times [\underline{y},\overline{y}] = [\min\{\underline{x}\,\underline{y},\underline{x}\overline{y},\overline{x}\,\underline{y},\overline{x}\overline{y}\},\max\{\underline{x}\,\underline{y},\underline{x}\overline{y},\overline{x}\,\underline{y},\overline{x}\overline{y}\}]\\ &xy \geq \min\{\underline{x}\,\underline{y},\underline{x}\overline{y},\overline{x}\,\underline{y},\overline{x}\overline{y}\} > 0\end{aligned}$$

□

**Exponentiation:**

For any natural number $n$, let us first consider the exponentation operator for odd powers:

$$\left[\underline{x}^{2n+1},\overline{x}^{2n+1}\right] > 0 \quad \overset{?}{\Longrightarrow} \quad x^{2n+1}>0,\ \forall x \in [\underline{x},\overline{x}]$$

*Proof.*

$$x^{2n+1} \geq \underline{x}^{2n+1} > 0$$

□

Now, let us do the same for even powers:

$$[\underline{x},\overline{x}]^{2n} > 0 \quad \overset{?}{\Longrightarrow} \quad x^{2n} > 0,\ \forall x \in [\underline{x},\overline{x}]$$

*Proof.* If $0 \leq \underline{x} < \overline{x}$ then $x^{2n} \geq \underline{x}^{2n} > 0$. If $\underline{x} < \overline{x} \leq 0$ then $x^{2n} \geq \overline{x}^{2n} > 0$. The case $\underline{x} < 0 \leq \overline{x}$ cannot be achieved because this would mean that $[\underline{x},\overline{x}]^{2n} = \left[0, M^{2n}\right]$ (where $M = \max\{|\underline{x}|, |\overline{x}|\}$), which would contradict the strict inequality $[\underline{x},\overline{x}]^{2n} > 0$. $\square$

So, for any function $f$ involving a combination of the arithmetic operations above, we have proved that

$$f([\underline{x},\overline{x}]) > 0 \quad \Longrightarrow \quad f(x) > 0,\ \forall x \in [\underline{x},\overline{x}]$$

There is another half to this proof, stating an analogous result for negative boxes.

$$f([\underline{x},\overline{x}]) < 0 \quad \Longrightarrow \quad f(x) < 0,\ \forall x \in [\underline{x},\overline{x}]$$

This result is based on the symmetry of the IA rules. Its proof is analogous to the one of the first part.

This theory can be easily extended to include rational functions, as interval division is expressed in terms of multiplication. Another important generalisation can be done to include more than one dimension. In fact, the one-dimensional case has been used merely for clarity of the argument, but since multidimensional IA rules are expressed componentwise, there is no reason why the result should not hold in any number of dimensions.

### 4.3.2 The Inaccurate Classification of Zero Boxes

Let us now examine the case where the resulting interval of the substitution straddles zero. At first sight this may seem to correspond to a situation where the box contains some curve segment or surface patch, independently of whether it belongs to the frontier of a solid or not. This section will illustrate a one-dimensional counterexample. We will show it is possible for the interval to straddle zero despite the box being an positive box indeed. Again, the phenomenon described can be easily observed and generalised to any number of dimensions.

Consider the following four real polynomial functions $f, g, h, k : [0,1] \to \mathbb{R}$ given by

$$f(x) = 4x^2 - 12x + 9 \qquad (\textit{power form})$$

$$g(x) = (4x - 12)x + 9 \qquad (\textit{Horner form})$$

$$h(x) = 9(x-1)^2 - 6x(x-1) + x^2 \qquad (\textit{Bernstein form})$$

$$k(x) = (2x - 3)^2 \qquad (\textit{factored form})$$

Although they appear in different forms,[1] their definitions are chosen such that $f(x) = g(x) = h(x) = k(x)$. Despite the fact that they take only positive values over the interval $[0, 1]$, in some cases the membership test outputs intervals straddling zero, though of course they all contain the image of the function.

The functions $f$, $g$, $h$ and $k$ take the same values everywhere and have equivalent implicit expressions, so they must have the same image in the range—namely the interval $[1, 9]$. Depending on the form of the polynomial expression, the interval arithmetic method may give predictions for the image which are wider intervals *including* it. This phenomenon is known as *interval swell* or *interval over-estimation* and is responsible for the appearance of false zero boxes. Let us illustrate this with the previous four real-valued functions by replacing $x$ by $[0, 1]$ in their expressions:

$$\begin{aligned} f([0,1]) &= 4([0,1])^2 - 12[0,1] + 9 \\ &= [-3, 13] \\ &\supset [1,9] = \text{Image}\, f \end{aligned}$$

$$\begin{aligned} g([0,1]) &= (4\ [0,1] - 12)\ [0,1] + 9 \\ &= [-3, 9] \\ &\supset [1,9] = \text{Image}\, g \end{aligned}$$

$$\begin{aligned} h([0,1]) &= 9([0,1] - 1)^2 - 6[0,1]([0,1] - 1) + [0,1]^2 \\ &= [0, 16] \\ &\supset [1,9] = \text{Image}\, h \end{aligned}$$

$$\begin{aligned} k([0,1]) &= (2\ [0,1] - 3)^2 \\ &= [1, 9] \\ &= [1,9] = \text{Image}\, k \end{aligned}$$

After applying interval arithmetic to the functions $f$, $g$, $h$ and $k$, we observe that only the prediction given by $k(x)$ gives an exact answer: the prediction in this case equals the exact image $[1, 9]$ of the function. The other examples illustrate the typical situation where the resulting interval straddles zero but the corresponding box is a false zero box because the box itself lies entirely in the positive half-space.

The boxes that interval arithmetic does label as negative or positive are always properly identified. However, not all zero boxes are correctly identified. But this is only a cautious box classification as the interval arithmetic technique cannot determine correctly the type of *all* the boxes in a given region of interest. In this scenario, the box classification is said to be *conservative.*

[1] For the definition of *Horner's scheme* (also known as nested multiplication), the reader is referred to the original article [194] as well as any good textbook on algebra or geometric algorithms [132]. A good *splines* textbook [132] will contain a definition and usage of the Bernstein polynomial basis.

Conservativeness is the main weakness of IA. Often the intervals produced are much wider than the true range of the computed quantities. This problem is particularly severe in long computational chains where the intervals computed at one stage are input into the next stage of the computation. The more variable occurrences there are in the algebraic expression, the wider the prediction and the larger the interval swell. However, this is not a general rule, because there are other aspects (such as the presence of even exponents and the order of the arithmetic operations) which may influence the final result. Several conservativeness examples and a suggested approach to this problem can be found in [40], [62] and [401].

## 4.4 The Influence of the Polynomial Form on IA

There is a wide variety of ways of writing and rewriting a polynomial. In the previous section, we have briefly approached four polynomial forms: power form, Horner form, Bernstein form and factored form. The reader is referred to de Boor [98] for other polynomial forms. Unfortunately, there is no known method to determine what is the best form to express a given polynomial function in order to get the sharpest possible bounds. This is so because the optimal way of representing and storing a polynomial is crucially determined by the kind of operations the user might want to perform on it afterwards. Studies and comparisons are given in Martin et al. [258, 259].

This section shows that the Bernstein form is the most stable numerically by comparing it to the power form as input to interval arithmetic. As suggested in the previous section for the equivalent functions $f(x)$, $g(x)$, $h(x)$ and $k(x)$, the resulting intervals obtained by replacing $x$ by $[0,1]$ may differ from one to another. This means that applying interval arithmetic to two equivalent functions has as a result two distinct space partitionings (Figure 4.2).

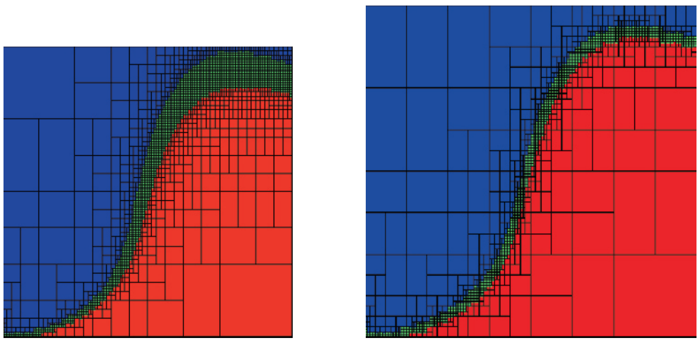

(a) (b)

**Fig. 4.2.** The influence of the polynomial form on interval arithmetic applied to locate a curve: (a) the power-form polynomial $p(x,y)$ defined in Section 4.3 and (b) its equivalent Bernstein-form polynomial.

As illustrated in Figure 4.2, it is visible the differences between the power form and Bernstein form of a polynomial, namely:

- *Number of sub-boxes.* Their corresponding 2-d tree space partitionings have a different number of sub-boxes. The power form polynomial on the left-hand side leads to a bigger number of space subdivisions than the Bernstein form polynomial on the right-hand side.
- *Box classification.* The zero boxes (in green) provide a better approximation to the curve when the function is in Bernstein form, so that there are fewer false zero boxes.

### 4.4.1 Power and Bernstein Form Polynomials

For brevity, we review univariate and multivariate polynomials in this section.

#### Univariate

A power form polynomial of degree $n \in \mathbb{N}$ in the variable $x$ is defined by:

$$f(x) = \sum_{i=0}^{n} a_i x^i, \tag{4.8}$$

where $a_i \in \mathbb{R}$. The equation $f(x) = 0$ is the *implicit equation* corresponding to the polynomial $f(x)$.

We have shown in the previous section that the form of the implicit expression supplied as input to interval arithmetic is crucial for the accuracy of the box classification. Since any input expression can be written in a number of equivalent forms, it makes sense to choose a transformation which will generate a more numerically stable polynomial form. If a base other than the power base is used in order to express the same polynomial, the interval arithmetic classification method will, in general, produce different results. The results which follow below encourage the use of the Bernstein base especially in the case of high-degree polynomials.

As seen in Section 3.1.3, the univariate Bernstein basis functions of degree $n$ on the interval $[\underline{x}, \overline{x}]$ (see also Lorentz [248]) are defined by:

$$B_i^n(x) = \binom{n}{i} \frac{(x-\underline{x})^i(\overline{x}-x)^{n-i}}{(\overline{x}-\underline{x})^n}, \quad \forall x \in [\underline{x}, \overline{x}], \quad i = 0, 1, \ldots, n. \tag{4.9}$$

For a given $n \in \mathbb{N}$, these $n+1$ univariate degree-$n$ Bernstein polynomials $(B_i^n)_{i=0,n}$ forms a basis for the ring of degree-$n$ polynomials. This means that any univariate power form polynomial can be represented on the interval $[\underline{x}, \overline{x}]$ using its equivalent Bernstein form as follows:

$$f(x) = \underbrace{\sum_{i=0}^{n} a_i x^i}_{\text{power form } p(x)} = \underbrace{\sum_{i=0}^{n} b_i^n B_i^n(x)}_{\text{Bernstein form } B(x)}$$

where $b_i^n$ are the Bernstein coefficients corresponding to the degree-$n$ base. The two univariate representations $p(x)$ and $B(x)$ are equivalent on the interval $[\underline{x}, \overline{x}]$. For example, on the unit interval $[0, 1]$, determining $B(x)$ from $p(x)$ requires the computation of the univariate Bernstein coefficients in terms of the power coefficients:

$$b_i^n = \sum_{j=0}^{i} \frac{\binom{i}{j}}{\binom{n}{j}} a_j \tag{4.10}$$

As referred in Section 3.1.3, Formula (4.10) can be used to design an algorithm of conversion between the power form and the Bernstein form of an univariate polynomial [133, 134].

## Multivariate

The generalisation of Bernstein bases to multivariate polynomials is not immediate. The power form of a polynomial in $d$ variables is written in terms of $x_1, \ldots, x_d$ like this:

$$f(x_1, \ldots, x_d) = \sum_{0 \le k_1 + \cdots + k_d \le n} a_{(k_1, \ldots, k_d)} x_1^{k_1} \cdots x_d^{k_d}$$

where the coefficients $a_{(k_1, \ldots, k_d)} \in \mathbb{R}$. Again, the equation $f(x_1, \ldots, x_d) = 0$ is the *implicit equation* corresponding to the *implicit polynomial* $f(x_1, \ldots, x_d)$. By convention, the degree of each term is $k_1 + \cdots + k_d$, and the degree of the polynomial is the maximum of all degrees of its terms.

The multivariate Bernstein form is defined recursively as a polynomial whose main variable is $x_d$ and whose coefficients are multivariate Bernstein-form polynomials in $x_1, \ldots, x_{d-1}$.

Formula (4.10) can be generalised to more variables. Conversion between the power and the Bernstein representation is possible regardless of the number of variables (see Geisow [158] and Garloff [155, 425]). In [40, 41] Berchtold et al. give formulae and algorithms for the computation of the Bernstein form of bi- and trivariate polynomials, as needed for locating implicit curves in 2D and surfaces and solids in 3D, respectively. The following example makes use of this particular conversion method.

*Example 4.1.* Let us look again at Figure 4.2. The power-form polynomial appears on the left-hand side and is given by the polynomial defined in Section 4.3 and written now, for convenience, with $\mathbb{R}$-style coefficients, though under the understanding that the calculations are exact:

$$p(x, y) = 0.9446\, x\, y - 7.0044\, x^3\, y^2 + 7.6455\, x^4\, y^3 + 0.5640\, y^4 - x^3$$

The corresponding Bernstein form in $[0, 1] \times [0, 1]$ appears on the righ-hand side of Figure 4.2 and is as follows:

$$
\begin{aligned}
B(x,y) = {} & \Big( -x^3(1-x) - x^4 \Big)(1-y)^4 + \\
& 4\Big(0.2361x(1-x)^3 + 0.7084x^2(1-x)^2 - 0.2915x^3(1-x) - 0.7638x^4\Big) \cdot \\
& \quad y(1-y)^3 + \\
& 6\Big(0.4723x(1-x)^3 + 1.4169x^2(1-x)^2 - 0.7505x^3(1-x) - 1.6951x^4\Big) \cdot \\
& \quad y^2(1-y)^2 + \\
& 4\Big(0.7084x(1-x)^3 + 2.1253x^2(1-x)^2 - 2.3768x^3(1-x) - 1.8823x^4\Big) \cdot \\
& \quad y^3(1-y) + \\
& \Big(0.564(1-x)^4 + 3.2006x(1-x)^3 + 6.2178x^2(1-x)^2 - 2.9146x^3(1-x) + \\
& \quad 1.1497x^4\Big) y^4
\end{aligned}
$$

The power representation in Figure 4.2(a) is less effective in areas which are further away from the origin, whereas the Bernstein representation in Figure 4.2(b) starts classifying correctly boxes which are roughly at a constant distance away from the function.

For low-degree polynomials the advantage of using the Bernstein form is not immediately obvious. However, in rectangular areas which are further away from the origin, high-degree polynomials in the power form usually operate with large powers of large numbers. Any small errors in the coordinates can cause a significant change in the value of the polynomial. Thus, the Bernstein base is more numerically stable than the power base, which means that minor perturbations introduced in the coefficients tend not to affect the value of the polynomial.

Floating point errors can also be a reason for interval swell. Very small numbers on the "wrong" side of the origin are decisive in the classification procedure. Numerical stability helps correct this problem, though the Bernstein representation is not entirely error-free.

### 4.4.2 Canonical Forms of Degrees One and Two Polynomials

The standard form polynomial for three-dimensional quadrics is, as for any degree-two polynomial, written in the following manner:

$$A + 2Bx + 2Cy + 2Dz + 2Exy + 2Fxz + 2Gyz + Hx^2 + Iy^2 + Jz^2 = 0 \quad (4.11)$$

This is also known as the general *expanded* equation of a quadric. Quadric surfaces are always a special category of surfaces in geometric modelling because of various nice geometric properties they possess (see, for example, Sarraga [348]). Their importance comes from the fact that they are able to describe the geometry of most engineering mechanical parts designed by current CAD/CAM systems. This explains why CSG geometric kernels were designed and implemented from quadrics. For further details, the reader is referred to the sVLIs set-theoretic kernel geometric modeller [61].

Quadrics are more commonly known by their respective *canonical form* equations, where terms are grouped together in a symmetrical manner. By *canonical form* we mean the best-known implicit form in which quadrics are normally defined and studied (as shown in Figure 4.3):

$$\pm\frac{x^2}{a^2} \pm \frac{y^2}{b^2} \pm \frac{z^2}{c^2} = 1 \tag{4.12}$$

that is, the normalised equation for a 3D quadric centred at the origin $(0,0,0)$. According to the sign of the coefficients of the expanded form (4.11) or the canonical form (4.12), the quadrics can be of different types. It can be easily proved (see, for example, Bronstein and Semendjajew [67]) that there are only a finite number of types of quadric surfaces.

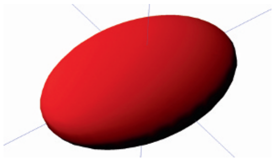
Furthermore, empirical tests carried out by the geometric modelling research group at Bath suggest it is probably the case that IA yields perfect classifications of *all* the sub-boxes of a region, provided they are tested against the equation of a *plane* or a *quadric surface* in the canonical form. Otherwise, the classifications are only conservative.

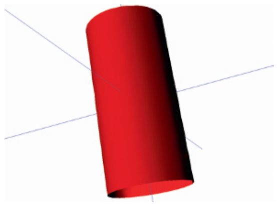
The multiplication of an interval by a constant and the addition and subtraction of two intervals are all exact operations. Hence, when an interval is substituted into a linear equation of the type $Ax + By + Cz + D = 0$ the arithmetic is expected to be well-behaved. The immediate geometrical consequence is that it is always possible to determine precisely whether a plane in space cuts a given box. The 'perfect' results are due not only to the linearity of the polynomial form but also to the fact that each variable occurs in the expression of the polynomial exactly once and independently from other variables. Thus no interference occurs between the different sources of noise. The coefficients $A$, $B$, $C$ and $D$ in the linear form $Ax+By+Cz+D=0$ are assumed to be obtained after all the reductions possible have been performed. Otherwise the swelling phenomenon reappears.

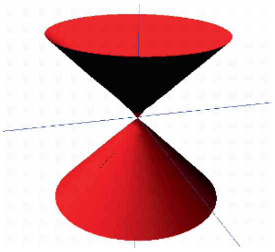
As an illustration, consider the polynomial $p(x) = 2x - x$. When studied over the unit interval, it yields a swollen result, despite its linearity:

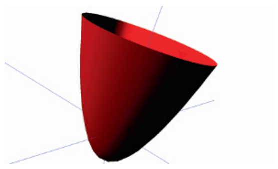
$$2\ [0,1] - [0,1] = [0,2] - [0,1] = [-1,2] \quad \supset \quad [0,1]$$

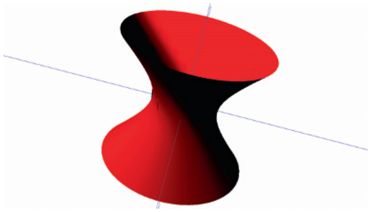
With the exception of the plane equation and quadrics in their canonical form, these "perfect" results *cannot* be obtained for equations of degree two or higher. The functions $f(x)$, $g(x)$ and $h(x)$ given in Section 4.3.2 have already illustrated a counterexample. That is, they all had degree-two equations but the intervals which resulted after applying interval arithmetic were not the

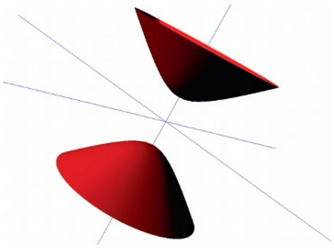

(a) Ellipsoid: $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$

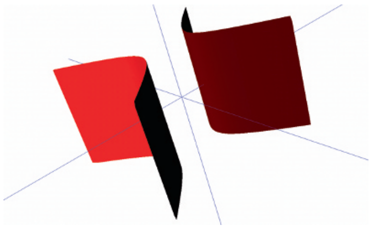

(b) Elliptic cylinder: $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$


(c) Elliptic cone: $\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0$


(d) Elliptic paraboloid: $z = \frac{x^2}{a^2} + \frac{y^2}{b^2}$


(e) Hyperboloid of 1 sheet:
$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1$


(f) Hyperboloid of 2 sheets:
$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = -1$


(h) Hyperbolic cylinder: $\frac{x^2}{a^2} - \frac{y^2}{b^2} = -1$

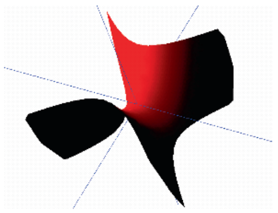

(i) Hyperbolic paraboloid: $z = \frac{x^2}{a^2} - \frac{y^2}{b^2}$

**Fig. 4.3.** Canonical forms for quadrics.

results of the exact calculations. When comparing the general expanded form with the canonical form of the same quadric, it is customarily the case that the former is the expansion of the latter and has degree-one terms as well as square terms. Most of the canonical forms of the quadrics have only degree-two terms, which constitutes an advantage for the application of interval arithmetic. This is due to the exponentiation rule stated in Equation (4.7), which causes the tightest positive intervals to be generated as results.

In general, the interval arithmetic technique can be used successfully for the location of the familiar quadrics, in that all the boxes of the spatial subdivision are classified correctly. The canonical form of the conic section surfaces, each of the variables occurs exactly once, independently from the others and with an exponent of one or two; thus it is expected that the resulting interval will coincide with the exact range. The technique starts suffering from conservativeness in the case of surfaces of an arbitrary representation, or of higher degree.

### 4.4.3 Nonpolynomial Implicits

One reason for extensively using polynomials is that the most important curves, surfaces and solids in geometric modelling can be expressed by means of polynomials. Perhaps the only significant exception is the helix. The helix is useful to represent as it is widely used in practice for such things as screw threads, but its formulation requires transcendental functions. Another reason why polynomials have been preferred is that algebraic theories provide extensive studies of polynomials. The findings concerning general algebraic functions cannot always be extended to transcendental functions.

As expected, conservativeness remains a problem for transcendental implicits. Whilst performing correctly for quite a large number of negative and positive boxes, the interval arithmetic technique still outputs some regions of space as zero boxes, although in reality they are purely negative or purely positive. Similarly to the polynomial case, the result is usable but not satisfactory.

*Example 4.2.* The expression $\sin(x)$ can legitimately be assumed to take values in the range $[-1, 1]$, but this may be quite a gross estimate. In the particular case where $x \in [\frac{1}{2}, \frac{7}{3}]$ the function's image is only $[\sin(\frac{1}{2}), 1] \subseteq [0.47, 1]$. When the $\sin(x)$ function is incorporated in further calculations, an initial range approximation as gross as $[-1, 1]$ will propagate the interval swell throughout the computation chain, affecting the final result.

An alternative evaluation method for periodic trigonometric functions (like $\sin(x)$ and $\cos(x)$) would be to calculate the range as a result of a circumstantial study of the domain. If the length of the domain interval is larger than the function period ($2\pi$ in the case of $\sin(x)$ or $\cos(x)$), then the function takes values over the whole of the range $[-1, 1]$. If not, then a detailed study of the relative positions of the ends of the interval and multiples of the values $0$, $\frac{\pi}{2}$, $\pi$, $\frac{3\pi}{2}$ and $2\pi$ will help establish the exact range. This is the case with the tangent and cotangent functions as well, with the further complication that these are not defined for certain values of their argument.

Other transcendental functions, like the logarithmic and exponential functions are slightly better behaved. Because they are monotone, an exact range can be obtained by evaluating the function at both ends of the interval. Problems may occur, however, when the function is not defined for the whole domain interval (e.g. $\log(x)$ is not defined for negative numbers or zero).

## 4.5 Affine Arithmetic Operations

As seen above, the conservativeness of algebraic methods that rely on interval arithmetic depends on the polynomial form used to represent implicit curves, surfaces and solids. We have also seen that the conservativeness is reduced when the input is provided in the Bernstein form. Furthermore, in the particular case of planes or quadrics represented by canonical form polynomials, the conservativeness vanishes.

As described in Section 4.3, the box classification method relies on substituting the interval coordinates of a box for the variables of an implicit function expression, performing interval arithmetic calculations, and studying the relative positions of the resulting interval and zero. It might be thought that the interval swell during the interval arithmetic evaluation depends merely on the number of occurrences of a variable in the implicit expression. There are other aspects (such as the presence of even exponents or the order of the arithmetic operations) which contradict this assumption. It is known that the Bernstein form of a polynomial has many more variable occurrences than the power form; despite this, the former behaves better with IA than the latter.

Still, whenever interval calculations are performed, no account is taken of the fact that each occurrence of any variable, such as $x$, always represents the same quantity. That is to say that each variable introduces the same error in all the terms of the polynomial. The method, called *affine arithmetic*, described in the rest of this chapter makes use of this observation and correlates the sources of error in the interval classification (see also Martin et al. [258] or Shou et al. [367]). And, more importantly, it does *not* depend on the polynomial form used to represent an implicit object. Thus, affine arithmetic can be viewed as a more sophisticated version of interval arithmetic.

### 4.5.1 The Affine Form Number

Affine arithmetic was proposed by Comba, Stolfi and others [90] in the early 1990s with a view to tackle the conservativeness problem caused by standard interval arithmetic. Like interval arithmetic, affine arithmetic can be used to manipulate imprecise values and to evaluate functions over intervals. While, like interval arithmetic, it provides guaranteed bounds for computed results, affine arithmetic also takes into account the dependencies between the sources of error. In this way it is able to produce much tighter and more accurate intervals than interval arithmetic, especially in long chains of computations.

In affine arithmetic an uncertain quantity $x$ is represented by an affine form $\hat{x}$ that is a first-degree polynomial of a set of noise symbols $\varepsilon_i$.

$$\hat{x} = x_0 + x_1\varepsilon_1 + \cdots + \ x_m\varepsilon_m = x_0 + \sum_{i=1}^{m} x_i\varepsilon_i$$

Here the value of each noise symbol $\varepsilon_i$ is unknown but defined to lie in the interval $[-1, 1]$. The corresponding coefficient $x_i$ is a real number that determines the magnitude of the impact of the product $x_i\varepsilon_i$. Each product $x_i\varepsilon_i$ stands for an independent source of error or uncertainty which contributes to the total uncertainty in the quantity $x$. The number $m$ may be chosen as large as necessary in order to represent all the sources of error. These may well be input data uncertainty, formula truncation errors, arithmetic rounding errors, and so on.

This piece of reasoning is not restricted to the univariate case. On the contrary, given a polynomial expression in any number of variables, the dependencies between them can be easily expressed by using the same noise symbol $\varepsilon_i$ wherever necessary. If the same noise symbol $\varepsilon_i$ appears in two or more affine forms (e.g. in both $\hat{x}$ and $\hat{y}$) it indicates the interdependencies and correlations that exist between the underlying quantities $x$ and $y$. For example, in the bivariate case, computing with the affine forms is a matter of replacing $x$ and $y$ by $\hat{x}$ and $\hat{y}$ in $f(x, y)$, respectively, and each operation in $f(x, y)$ with the corresponding affine operation on $\hat{x}$ and $\hat{y}$. Of course, each affine operation must take into account the relationships between the noise symbols in $x$ and $y$.

The rules for arithmetic operations on affine forms are explained below. The important thing to notice about the way affine arithmetic works is that algebraic expressions take into account the fact that the same variable may appear in them more than once. Thus using affine arithmetic, similar terms get cancelled when they appear in an expression (e.g. $2\hat{x} + \hat{y} - \hat{x} = \hat{x} + \hat{y}$). This is *not* the case with interval arithmetic.

### 4.5.2 Conversions between Affine Forms and Intervals

Conversions between affine forms and intervals are defined in various papers by Comba and Stolfi [90], Figueiredo [100] and Figueiredo and Stolfi [102].

Given an interval $[\underline{x}, \overline{x}]$ representing a quantity $x$, its affine form can be written as

$$\hat{x} = x_0 + x_1\varepsilon_x, \qquad \text{where } \ x_0 = \frac{\underline{x} + \overline{x}}{2}, \ \ x_1 = \frac{\overline{x} - \underline{x}}{2}. \tag{4.13}$$

Conversely, given an affine form $\hat{x} = x_0 + x_1\varepsilon_1 + \cdots + \ x_m\varepsilon_m$, the range of possible values of its corresponding interval is

$$[\underline{x}, \overline{x}] = [x_0 - \xi, x_0 + \xi]\,, \qquad \text{where } \ \xi = \sum_{i=1}^{m} |x_i|. \tag{4.14}$$

### 4.5.3 The Affine Operations

The affine arithmetic rules are fully defined in Comba and Stolfi [90]. Those that are relevant to the location of curves and surfaces are addition and multiplication, both of a scalar to an affine form, and of (two or more) affine forms to each other. Given the affine forms $\hat{x}$ and $\hat{y}$, and the real number $\alpha \in \mathbb{R}$ the simple arithmetic operations are carried out thus:

**Addition:**

$$\alpha + \hat{x} = (\alpha + x_0) + x_1\varepsilon_1 + \cdots + x_m\varepsilon_m \tag{4.15}$$
$$\hat{x} + \hat{y} = (x_0 + y_0) + (x_1 + y_1)\varepsilon_1 + \cdots + (x_m + y_m)\varepsilon_m \tag{4.16}$$

**Subtraction:**

$$\alpha - \hat{x} = (\alpha - x_0) + x_1\varepsilon_1 + \cdots + x_m\varepsilon_m \tag{4.17}$$
$$\hat{x} - \hat{y} = (x_0 - y_0) + (x_1 - y_1)\varepsilon_1 + \cdots + (x_m - y_m)\varepsilon_m \tag{4.18}$$

**Multiplication:**

$$\alpha\hat{x} = (\alpha x_0) + (\alpha x_1)\varepsilon_1 + \cdots + (\alpha x_m)\varepsilon_m \tag{4.19}$$
$$\hat{x}\hat{y} = (x_0 + x_1\varepsilon_1 + \cdots + x_m\varepsilon_m)(y_0 + y_1\varepsilon_1 + \cdots + y_m\varepsilon_m) \tag{4.20}$$
$$= \left(x_0 + \sum_{i=1}^{m} x_i\varepsilon_i\right)\left(y_0 + \sum_{i=1}^{m} y_i\varepsilon_i\right)$$
$$= \underbrace{x_0y_0 + \sum_{i=1}^{m}(x_0y_i + x_iy_0)\varepsilon_i}_{\mathcal{L}(\varepsilon_1,\ldots,\varepsilon_m)} + \underbrace{\left(\sum_{i=1}^{m} x_i\varepsilon_i\right)\left(\sum_{i=1}^{m} y_i\varepsilon_i\right)}_{\mathcal{Q}(\varepsilon_1,\ldots,\varepsilon_m)} \tag{4.21}$$

Now, $\mathcal{L}(\varepsilon_1,\ldots,\varepsilon_m)$ is an affine form in which the noise symbols $\varepsilon_i$ occur only with degree 1, whereas $\mathcal{Q}(\varepsilon_1,\ldots,\varepsilon_m)$ is quadratic in the noise symbols. The quadratic term can be handled so that it becomes linear itself, at the expense of introducing a new noise symbol $\varepsilon_k \in [-1, 1]$, with coefficient $\mu\nu$, where $\mu = \sum_{i=1}^{m} |x_i|$ and $\nu = \sum_{i=1}^{m} |y_i|$. So $\hat{x}\hat{y}$ can be expressed as an affine combination of first-degree polynomials in the noise symbols:

$$\hat{x}\hat{y} = x_0y_0 + \sum_{i=1}^{m}(x_0y_i + x_iy_0)\varepsilon_i + \mu\nu\varepsilon_k$$
$$= x_0y_0 + (x_0y_1 + x_1y_0)\varepsilon_1 + \cdots + (x_0y_m + x_my_0)\varepsilon_m + \mu\nu\varepsilon_k$$

The index $k$ can be chosen as $m + 1$.

**Division:**

Division can be defined via inversion and multiplication in the same style as shown in Formula (4.5) for intervals. This is rarely used in calculations, as there is little scope for simplifying the polynomial expansions obtained.

**Exponentiation:**

$$\hat{x}^a = (x_0 + x_1\varepsilon_x)^a = x_0^a + \sum_{i=1}^{a} \binom{a}{i} x_0^{a-i} x_1^i \varepsilon_x^i, \quad a \in \mathbb{Z}. \tag{4.22}$$

Unlike interval arithmetic, the affine exponentiation is a particular case of the affine multiplication because

$$\hat{x}^2 = \hat{x}.\hat{x}$$

and, consequently, there is no interval swell caused by exponentiation.

It is immediately apparent from the rules above that the affine arithmetic operations are commutative, associative and distributive. This was not the case with interval arithmetic, whose misbehaviour with the distributivity law caused the interval swell.

Practical experience with polynomials other than those of lowest degree, shows that simply using the rules of affine arithmetic directly gives relatively little advantage over ordinary interval arithmetic when localising polynomials (e.g. curves and surfaces), which are basically defined by additions, subtractions, multiplications and exponentiations. This is due to rapid introduction of many new error symbols. Much better results can be obtained by taking more care, in particular in handling exponentiations.

### 4.5.4 Affine Arithmetic Evaluation Algorithms

Various affine arithmetic schemes have been proposed for use in geometric modelling. One of the earlier ones (see Zhang and Martin [426] or Voiculescu et al. [402]) proposes to simplify exponentiations in a way that separates odd exponent terms from even exponent terms, and express any (univariate) polynomial with a degree-one polynomial of three terms and just two noise symbols:

$$\hat{x}^a = x_0^a + x_{odd}\varepsilon_{xodd} + x_{even}\varepsilon_{xeven}.$$

Whilst this yields results very efficiently and leads to reasonably narrow result intervals, it unfortunately does so at the expense of the loss of conservativeness. This comes from trying to share the noise symbols $\varepsilon_{xodd}$ and $\varepsilon_{xeven}$ between the computations of two distinct powers [373].

A more complete yet more expensive scheme is proposed in a related paper [258] where Martin et al. give a matrix-form evaluation of the affine interval polynomials that leads to a conservative interval result.

## 4.6 Affine Arithmetic-driven Space Partitionings

When applying affine arithmetic to algebraic surface location, the polynomial representing the implicit surface needs to be evaluated on the intervals over which its variables range. In particular, in order to locate a planar curve a polynomial $f(x, y)$ needs to be evaluated over the ranges in $x$ and $y$ representing a box. These are $[\underline{x}, \overline{x}]$ and $[\underline{y}, \overline{y}]$ or their affine equivalents $\hat{x}$ and $\hat{y}$ respectively.

Because the affine arithmetic form can be converted back into an interval, it can easily be used as an alternative to producing box classifications for power- or Bernstein-form polynomials using direct interval arithmetic rules.

To compare the relative merits of interval arithmetic and carefully evaluated affine arithmetic for curve drawing, we now present a practical example.

*Example 4.3.* Let us consider the following bivariate polynomial function $p(x, y)$ in the power form:

$$p(x,y) = \frac{945}{1000}\, x\, y - \frac{94{,}3214}{100{,}000}\, x^2\, y^3 + \frac{74{,}554}{10{,}000}\, x^3\, y^2 + y^4 - x^3$$

and then in its Bernstein form in the unit box $[0, 1] \times [0, 1]$:

$$\begin{aligned}
B(x,y) = &- \ x^3\,(1-y)^4 + \\
&4\left(\frac{23{,}625}{100{,}000}\, x\,(1-x)^2 + \frac{4725}{10{,}000}\, x^2\,(1-x) - \frac{76{,}375}{100{,}000}\, x^3\right) y\,(1-y)^3 + \\
&6\left(\frac{4725}{10{,}000}\, x\,(1-x)^2 + \frac{945}{1000}\, x^2\,(1-x) + \frac{715{,}066{,}667}{1{,}000{,}000{,}000}\, x^3\right) y^2\,(1-y)^2 + \\
&4\left(\frac{70{,}875}{100{,}000} x(1-x)^2 - \frac{9{,}405{,}350{,}004}{10{,}000{,}000{,}000} x^2(1-x) + \frac{1{,}078{,}415}{1{,}000{,}000} x^3\right) y^3(1-y) + \\
&\left((1-x)^3 + \frac{3945}{1000}\, x\,(1-x)^2 - \frac{4{,}542{,}140{,}001}{1{,}000{,}000{,}000}\, x^2\,(1-x) - \frac{103{,}174}{100{,}000}\, x^3\right) y^4
\end{aligned}$$

The left-hand side of Figure 4.4 represents the interval arithmetic classification of the Bernstein form (i.e. the best polynomial form for IA). The right-hand side illustrates the result of applying affine arithmetic (AA) to the power-form polynomial $p(x, y)$. Both have been drawn using a minimum box size of $\frac{1}{2^7} \times \frac{1}{2^7}$. As apparent from Figure 4.4, AA definitely classifies a larger area, and in bigger chunks at a time, than either case of IA. The Table 4.1 gives the respective box percentages for $p(x, y)$ at a resolution $\Delta = \frac{1}{2^{10}} \times \frac{1}{2^{10}}$.

The complexity of each algorithm depends on the type of arithmetic used (i.e. standard interval arithmetic or affine arithmetic), as well as on the form of the input. Tables 4.2 and 4.3 summarise the running times and the number of subdivisions in each case. (Note that the times are interesting to compare, but not relevant in absolute terms, as the implementation depends on the interval package and hardware used.)

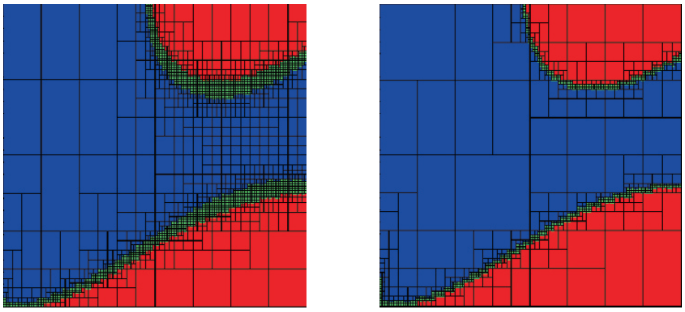

(a) IA on Bernstein form (b) AA on power form

**Fig. 4.4.** Interval- and affine arithmetic box classification for $p(x, y)$ and $B(x, y)$ in the unit box $[0, 1] \times [0, 1]$.

**Table 4.1.** Box percentages for $p(x, y)$ at a resolution $\Delta = \frac{1}{2^{10}} \times \frac{1}{2^{10}}$.

| | Negative boxes [−,−] | Zero boxes [−,+] | Positive boxes [+,+] |
|---|---|---|---|
| IA on power form | 0.3171 | 0.0231 | 0.6597 |
| IA on Bernstein form | 0.3241 | 0.0088 | 0.6670 |
| AA on power form | 0.3266 | 0.0037 | 0.6695 |

**Table 4.2.** Running times and number of subdivisions for $p(x, y)$ at $\Delta = \frac{1}{2^{10}} \times \frac{1}{2^{10}}$.

| | time (*sec*) | subdivisions |
|---|---|---|
| IA on power form | 2338.121 | 39834 |
| IA on Bernstein form | 2783.140 | 15568 |
| AA on power form | 194.339 | 6447 |

**Table 4.3.** Running times and number of subdivisions for $p(x, y)$ at $\Delta = \frac{1}{2^7} \times \frac{1}{2^7}$.

| | time (*sec*) | subdivisions |
|---|---|---|
| IA on power form | 20.94 | 1854 |
| IA on Bernstein form | 51.52 | 947 |
| AA on power form | 10.07 | 433 |

The results in Table 4.2 and Table 4.3 have been obtained also using different minimum box sizes, $\frac{1}{2^{10}} \times \frac{1}{2^{10}}$ and $\frac{1}{2^7} \times \frac{1}{2^7}$, respectively.

For the example given above the affine arithmetic method produces results more quickly (and accurately) than either interval arithmetic method. The former involves slightly more calculations per box, but classifies big boxes in a very efficient manner. When interval arithmetic is applied there are fewer calculations per box than for affine arithmetic. Still, the Bernstein polynomial form is so much more complicated that the program runs much slower.

Regarding the number of subdivisions, interval arithmetic needs much finer subdivision of boxes for the power form than for the Bernstein form and ends up with a less accurate result. Affine arithmetic needs comparatively fewer subdivisions to reach a very accurate result.

In principle, rather than the interval arithmetic, one could also study the Bernstein form using the affine arithmetic approach. However, as it has been shown that affine arithmetic operations are associative, commutative and distributive, it is expected that different polynomial representations would produce the same results. This is because the various ways of expressing a polynomial function using different bases does nothing other than rearranging the terms. This rearrangement does not affect the arithmetic of the polynomial, and hence does not affect the result of applying affine arithmetic to an equivalent polynomial form. Therefore, when studying affine arithmetic, it is only the power basis that needs to be considered. The proof of this final statement has been published in [259].

As a final remark in this section, it is worth noting that when interval arithmetic produces a correct estimate of the range of values, then affine arithmetic is expected to produce an exact range too. For example, in the case of function $k(x) = (2x - 3)^2$ studied in Section 4.3.2, interval arithmetic gives the correct range $[1, 9]$, and so does affine arithmetic.

## 4.7 Floating Point Errors

Recursive subdivision using interval arithmetic relies fundamentally on the arithmetic operations carried out on the end values of the intervals being accurate. This is why the examples given so far have involved polynomials with rational coefficients and subdivisions of boxes stored as rational intervals. Implementations in languages without a *rational number* data type will compromise the precision of the calculations by storing the numbers as *floating point* values.

The current section illustrates the extent to which floating point errors propagate through the evaluation process, often making the classification process impracticable. Let us recall the polynomial $p$, defined in Section 4.3, in its rational and floating point power forms:

$$p_{rat}(x, y) = \frac{9446}{10{,}000}\, x\, y - \frac{700{,}443{,}214}{100{,}000{,}000}\, x^3\, y^2 + \frac{764{,}554}{100{,}000}\, x^4\, y^3 + \frac{564}{1000}\, y^4 - x^3$$

$$p_{flt}(x, y) = 0.9446\, y\, x - 7.00443214\, y^2\, x^3 + 7.64554\, x^4\, y^3 + 0.564\, y^4 - x^3$$

This was originally defined in the unit box, and had the zero set illustrated in Figure 4.1.

We now aim to translate $p$ so that it has the same zero set in a general box, say $[9.62, 10.62] \times [7.31, 8.31]$. This can be achieved in several ways, all involving the substitution of $x$ by $x$-minus-some-quantity, and $y$ by $y$-minus-some-quantity, in either $p_{rat}$ or $p_{flt}$:

- substitute $x := x - \frac{962}{100}$ and $y := y - \frac{731}{100}$ in $p_{rat}$, yielding $p_1$;
- substitute $x := x - \frac{962}{100}$ and $y := y - \frac{731}{100}$ in $p_{flt}$, yielding $p_2$;
- substitute $x := x - 9.62$ and $y := y - 7.31$ in $p_{rat}$, yielding $p_3$;
- substitute $x := x - 9.62$ and $y := y - 7.31$ in $p_{flt}$, yielding $p_4$.

Floating point errors already start occurring at the stage where brackets are multiplied out. In the particular case of $p(x, y)$, when using a precision of 10 significant digits, $p_3 = p_4$. The three zero sets (corresponding to $p_1$, $p_2$ and $p_3$ respectively) in the box $[9.62, 10.62] \times [7.31, 8.31]$ are plotted in Figure 4.5, in the order *cyan, magenta, yellow.*

The affine arithmetic method necessarily complies to one of the four schemes above. Our study uses two schemes in parallel: all the way through the subdivision process described above. Any subdivision decisions are taken using $p_1$ and a "totally rational" scheme. At the same time, the subboxes are also converted to their floating point equivalents and subjected to the sign test against the floating point polynomial $p_4$. Thus it is certain that subdivision is carried out correctly. The respective ranges (given by the two different approximations) can be compared.

When the signs of the two ranges agree (in that they both indicate a negative or a positive box), the same conventions for colours as before has been used—that is, red for negative and blue for positive. However, when the rational arithmetic predicts a negative box and the floating point arithmetic calculation disagrees, the box is coloured *magenta*. Similarly, when the rational arithmetic predicts a positive box but the floating point arithmetic calculation disagrees, the box is coloured *cyan*. Zero boxes are still coloured green. The result is illustrated in Figure 4.6(a).

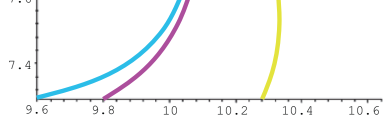


**Fig. 4.5.** Zero sets of $p_1$ (cyan), $p_2$ (magenta) and $p_3$ (yellow).

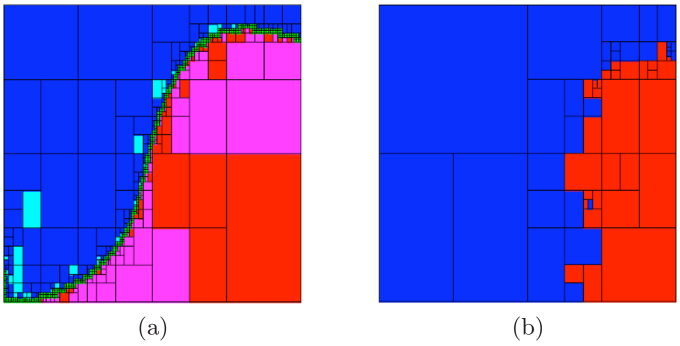


**Fig. 4.6.** (a) Affine arithmetic classification of floating-point polynomial using rational evaluations in the box $[9.62, 10.62] \times [7.31, 8.31]$; (b) Affine arithmetic (mis)classification using only floating-point evaluations in the box $[9.62, 10.62] \times [7.31, 8.31]$.

The frequent occurrence of magenta and cyan boxes indicates to what extent floating point errors can influence the affine arithmetic calculations. Had there been only floating point evaluations, the classification would have been totally irrelevant, as decisions for further subdivision would have been taken in completely the wrong places. Indeed, when running such a test it simply returns an inconsistent collection of negative and positive boxes, which only vaguely evokes the shape of the initial curve (Figure 4.6(b)).

This is a typical illustration of the propagation of floating point errors. Let us now consider a single magenta box and examine the way in which the four possibilities there are for approximating either the coefficients or the box can influence the final range given as a result.

Take the rational box:

$$\left[\frac{8471}{800}, \frac{33{,}909}{3200}\right] \times \left[\frac{1637}{200}, \frac{13{,}121}{1600}\right] =$$
$$[10.58875000, 10.59656250] \times [8.185000000, 8.200625000]$$

This is one of the boxes coloured magenta in Figure 4.6(a). Its corresponding affine forms are:

$$\hat{x} = \frac{67{,}793}{6400} + \frac{1}{256}\varepsilon_x = 10.59265625 + 0.003906250000\,\varepsilon_x$$
$$\hat{y} = \frac{26{,}217}{3200} + \frac{1}{128}\varepsilon_y = 8.192812500 + 0.007812500000\,\varepsilon_y$$

Let us evaluate the results returned by affine arithmetic when classifying these affine forms and/or their floating point equivalents against the various forms of $p$. The results differ according to the amount of floating point approximation carried out:

$$\texttt{affine\_eval}(p_1(x,y), \hat{x}_{rat}, \hat{y}_{rat}) = \left[\frac{\text{-8,180,237,644,390,479,080,447}}{\text{56,294,995,342,131,200,000,000}}, \frac{\text{-2,383,608,804,974,363}}{\text{140,737,488,355,328,000}}\right] = [-0.1453102109, -0.01693655921]$$

$$\texttt{affine\_eval}(p_1(x,y), \hat{x}_{\text{float}}, \hat{y}_{\text{float}}) = [0.09670367511, 0.2243380429]$$

$$\texttt{affine\_eval}(p_4(x,y), \hat{x}_{\text{rat}}, \hat{y}_{\text{rat}}) = [-0.1038870246, 0.02492879264]$$

$$\texttt{affine\_eval}(p_4(x,y), \hat{x}_{\text{float}}, \hat{y}_{\text{float}}) = [0.09641367500, 0.2246280630]$$

To summarise, the intervals generated as answers vary in their signs and positions relative to zero. The results are not conservative anymore; on the contrary, some of them have completely misclassified the box type, as shown in Table 4.4:

**Table 4.4.** Box classification for $p_1$ and $p_4$.

| $p(x,y)$ | $\hat{x} \times \hat{y}$ | interval type | box classification |
|---|---|---|---|
| rat | rat | $[-,-]$ | negative |
| rat | float | $[+,+]$ | positive |
| float | rat | $[-,+]$ | zero |
| float | float | $[+,+]$ | positive |

Of course, "mixed" forms of the polynomial (such as $p_2$) could have been used in the experiments as well, generating a potentially wider variety of answers. Nevertheless the study outlined above illustrates the point being made in this section, which is that floating point errors are not negligible.

All the floating point calculations in this section have been carried out using a precision of 10 significant digits. Increasing the precision of the calculations may eliminate the problem for particular cases. Indeed, in the case of $p(x,y)$ a precision of 40 significant digits seems to be enough for a box classification comparable to the one where rational arithmetic had been used. However this is not a general solution, as the result depends thoroughly on the precision with which the polynomial coefficients and the edges of the box are being calculated in the first place.

## 4.8 Final Remarks

There are, of course, a variety of ways in which the polynomial can be input, such as storing it in some canonical form, using a planar basis [61], or using an implicitisation of some Bernstein form [41], a Taylor expansion [368], etcetera. Overall, we conclude that the conservativeness problem which occurs in surface location can be reduced in at least two major ways: either the input is given in Bernstein form instead of power form and interval arithmetic is used, or the calculations are carried out on the power form, but a careful strategy based on affine arithmetic is used instead of interval arithmetic.

When the Bernstein form is used the improvement is significant: boxes can be located much more accurately in a given region of interest. The shape of the surface is outlined in enough detail for it to be located.

When affine arithmetic is used as shown above, our results demonstrate that curves can be located even more closely. This is because the intervals produced during polynomial evaluation are tighter.

Affine arithmetic calculations are more complicated than interval arithmetic ones. This is why the method is more error-prone when using floating point calculations. This is also why, in some cases, we have found it to be perhaps twice as slow as simple interval arithmetic, although this is strongly dependent on the implementation.

However, affine arithmetic has a speed advantage in some cases when interval arithmetic performs particularly badly. This advantage arises in the subdivision method because fewer boxes need to be considered, even though the amount of computation for any single box is greater.

All in all, it is fully expected that the benefits shown in curve drawing are also applicable to other uses of solutions to implicit equations, such as surface intersection, surface location, etc. Although the examples shown here have used polynomials, similar approaches could also be used if non-polynomial functions are needed for modelling. Different suitable basis functions and affine evaluation methods will need to be found for such cases.

There is also a need to express the operations defined here in a more compact form (perhaps using matrices). This would facilitate generalisations and would help study the operations and properties at a higher level of abstraction.