

RESEARCH

Open Access

# A refined affine approximation method of multiplication for range analysis in word-length optimization

Ruiyi Sun, Yan Zhang\* and Aijiao Cui

## Abstract

Affine arithmetic (AA) is widely used in range analysis in word-length optimization of hardware designs. To reduce the uncertainty in the AA and achieve efficient and accurate range analysis of multiplication, this paper presents a novel refined affine approximation method, Approximation Affine based on Space Extreme Estimation (AASEE). The affine form of multiplication is divided into two parts. The first part is the approximate affine form of the operation. In the second part, the equivalent affine form of the estimated range of the difference, which is introduced by the approximation, is represented by an extra noise symbol. In AASEE, it is proven that the proposed approximate affine form is the closest to the result of multiplication based on linear geometry. The proposed equivalent affine form of AASEE is more accurate since the extreme value theory of multivariable functions is used to minimize the difference between the result of multiplication and the approximate affine form. The computational complexity of AASEE is the same as that of trivial range estimation (AATRE) and lower than that of Chebyshev approximation (AACHA). The proposed affine form of multiplication is demonstrated with polynomial approximation, B-splines, and multivariate polynomial functions. In experiments, the average of the ranges derived by AASEE is 59% and 89% of that by AATRE and AACHA, respectively. The integer bits derived by AASEE are 2 and 1 b less than that by AATRE and AACHA at most, respectively.

**Keywords:** Word-length optimization; Range analysis; Affine approximation; Accuracy; Uncertainty analysis

## 1 Introduction

As a method of representing real numbers, floating point can support a wide dynamic range and high precision of values. It has been thus commonly used in signal processing, such as image processing, speech processing, and digital signals processing, to represent signals. When these applications are implemented on hardware for high speed and stability, the signals need to be represented in fixed point to optimize the performance of area, power, and speed of the hardware. Hence, the values in floating-point need to be converted to those in fixed point. This process is named as word-length optimization. Its goal is to achieve optimal system performance while satisfying the specification on the system output precision. Word-length optimization involves range analysis and precision

analysis. The former one is to find the minimum word length of the integer part of the value, while the latter one focuses on the optimization of the fractional part of the word length.

Word-length optimization has been proven to be an NP-hard problem [1]. It can be usually classified into dynamic analysis [2-7] and static analysis [8-20]. By analyzing a large set of stimuli signals, dynamic analysis is applicable to all types of systems. However, it will take long time on simulation to provide sufficient confidence. Also, the precision for the signals without simulation cannot be guaranteed. Comparatively, the static analysis is an automated and efficient word-length optimization method and more applicable to large designs when compared to dynamic analysis. The static analysis mainly uses the characteristics of the input signals to estimate the word length conservatively, which can result in overestimation [12] to some extent. As a part of word-length optimization, the range analysis can also be classified in the same way.

\*Correspondence: ianzh@foxmail.com

Key Laboratory of Network Oriented Intelligent Computation, Harbin Institute of Technology Shenzhen Graduate School, Xili, Shenzhen 518055, China

Affine arithmetic (AA) [21] is often used for range analysis in static analysis. In AA, every signal must be represented in an affine form, which is a first-degree polynomial. As AA tracks the correlations among range intervals of signals, it can provide more accurate word-length range. This makes it suitable for range analysis of the result of linear operations. It is noted that besides linear operations, nonlinear operations, such as multiplication, are also involved in hardware operations, typically in linear time invariant (LTI) systems. AA cannot provide an exact affine form for nonlinear operations. To solve this problem, Stolfi and de Figueiredo [22] proposed affine approximation methods for multiplication, which include trivial range estimation (AATRE) and Chebyshev approximation (AACHA). AATRE is efficient for computation, but the range produced by it can be four times of real range at most. The accumulation of the uncertainty of all signals in the computational chain may result in an error explosion, which is unacceptable in application. Such overestimation obviously cannot satisfy the accuracy requirement of the system, which limits the application of AATRE in large systems. The uncertainty of AACHA is less than AATRE, however, it is too complex to be used in large systems. Since LTI operations are accurately covered by AA, the proposed method is applied in the field of the range analysis of word-length optimization in this paper.

A novel affine approximation method, Approximation Affine based on Space Extreme Estimation (AASEE), is proposed to reduce the uncertainty of multiplication and achieve an accurate and efficient range analysis of multiplication in this paper. To analyze the uncertainty conveniently, we use two parts to divide the different parts of all the approximation methods for multiplication, which include AATRE, AACHA, and AASEE. The first part is named as approximate affine form, which is approximated to the nonlinear operation. The second part is named as equivalent affine form, which is the equivalent affine form of the estimated range of the difference between the result of multiplication and the approximate affine form. The more accurate the two parts are, the more accurate the approximation method is. Based on linear geometry [23], it is proven that the proposed approximate affine form is the closest to the result of multiplication. To derive the equivalent affine form, we use the extreme value theory of multivariable functions [24] to estimate the upper and lower bounds of the difference in space, and the difference is introduced by the approximation of the first part. The uncertainty of the proposed method is minimized. The accuracy of the resulting affine form by AASEE is higher than that by AATRE and averagely higher than that by AACHA. Meanwhile, the computational complexity of AASEE is equivalent to that of AATRE and lower than that of AACHA.

The rest of this paper is organized as follows. Background of range analysis for multiplication is presented in Section 2. Section 3 presents the method of derivation of the two parts for multiplication. The refined affine form of multiplication, AASEE, is presented in next section. In Section 5, we compare the computational complexity and the accuracy among AASEE to AATRE and AACHA. The case studies and experimental results are demonstrated in Section 6. Section 7 concludes the paper.

## 2 Background

### 2.1 Related work

Interval arithmetic (IA) and affine arithmetic (AA) have been widely used in range analysis in word-length optimization.

IA [25] is a range arithmetic theory which is firstly presented by Moore in 1962. Cmar [2] employs it for range analysis of digital signal processing (DSP) systems. Carreras [20] presents a method based on IA. To reduce the oversized word length, the method provides the probability density functions that can be used when some truncation must be performed due to constraints in the specification. IA is not suitable for most real-world applications, since it could lead to drastic overestimation of the true range.

AA [21] is proposed to overcome the weakness of IA by Stolfi in 1993. In [8,9], Fang uses AA to analyze word-length optimization. Both range and precision are represented by the same affine form, which limits the optimization. Pu and Ha [10] also use AA for word-length optimization. Simultaneously, they use two different affine forms for range analysis and precision analysis, respectively, and achieve more refined result of word-length optimization. Similarly, Lee et al. [11] develop an automatic optimization approach, which is called MiniBit, to produce accuracy-guaranteed solutions, and area is minimized while meeting an error constraint. Osborne [12] uses both IA and AA for range analysis for different situations. Computation using either of the two methods in the design is time-consuming. The problem of overestimation is serious due to the approximation of the nonlinear operations.

Since AA cannot be used in the systems with infinite number of loops, an improved approach, quantized AA (QAA), has been proposed in [13] for linear time-invariant systems with feedback loops. This method can provide fast and tight estimation of the evolution of large sets of numerical inputs, using only an affine-based simulation, but it does not provide the exact bounds.

AATRE [22] is adopted for multiplication in most of the works for the low computational complexity. But the uncertainty of the range by AATRE is very large. To adjust the trade-off between the accuracy of approximation and computational complexity, Zhang [14] introduces a new

parameter  $N$  in the  $N$ -level simplified affine approximation ( $N$ -SAA). This method is faster than AACHA and more accurate than AATRE, but it is more complex than AATRE. Furthermore, it is troublesome to choose a suitable  $N$ . A method of range analysis is proposed by Pang [26]. This method combines methods of IA, AATRE, and arithmetic transform (AT); and the result of the method is more accurate than AATRE, while the CPU implementation time is longer than AATRE. To deal with applications from the scientific computing domain, Kinsman [17,18] uses the computational methods based on Satisfiability Modulo Theory. Search efficiency of this method is improved leading to tighter bounds and thus smaller word length.

For all the existing methods, the accuracy of approximation is improved at the expense of the computational complexity. This paper presents an affine approximation method for multiplication, which achieves better trade-off between accuracy and computational complexity.

## 2.2 Range analysis

Range analysis involves studying the data range of every signal and minimizing the integer word lengths for signals on the premise that the signals in the design have enough bits to accommodate this range. The range of signal  $x$  is represented by  $x = [x_{\min}, x_{\max}]$ , where the two real numbers,  $x_{\min}$  and  $x_{\max}$ , denote the lower and upper bounds of  $x$ , respectively. The required integer part of the word length for signal  $x$ , which is represented as  $IWL_x$ , can be derived by:

$$IWL_x = \begin{cases} \lceil \log_2(|x|_{\max}) \rceil + \alpha, & |x|_{\max} \geq 1 \\ 1, & |x|_{\max} < 1 \end{cases}$$

where  $|x|_{\max} = \max(|x_{\min}|, |x_{\max}|)$  and

$$\alpha = \begin{cases} 1, & \text{mod}(\log_2(x_{\max}), 1) \neq 0 \\ 2, & \text{mod}(\log_2(x_{\max}), 1) = 0 \end{cases}$$

(1)

In (1), all the signals in the design are assumed to be expressed as signed numbers, and the sign bit is taken into account in  $IWL_x$ . According to (1), once the range of a signal is decided, the integer part of word length of the signal can be derived.

## 2.3 Affine arithmetic

AA is widely applied for range analysis. In AA, an uncertain signal  $x$  is represented by an affine form as a first-degree polynomial [22]:

$$\hat{x} = x_0 + x_1\varepsilon_1 + x_2\varepsilon_2 + \dots + x_n\varepsilon_n, \text{ where } \varepsilon_i = [-1, 1]. \quad (2)$$

For the signal  $x$ ,  $x_0$  is the central value, and  $\varepsilon_i$  is the  $i$ th noise symbol.  $\varepsilon_i$  denotes an independent uncertainty

source that contributes to the total uncertainty of the signal  $x$ , and  $x_i$  is its coefficient.

The upper and lower bounds for the range of  $x$  can be represented as

$$x_{\max} = x_0 + \sum_{i=1}^n |x_i|, \quad x_{\min} = x_0 - \sum_{i=1}^n |x_i|. \quad (3)$$

With  $x_{\min}$  and  $x_{\max}$ , the input interval  $\bar{x} = [x_{\min}, x_{\max}]$  can be converted into an equivalent affine form as (4), using only one independent noise symbol.

$$\hat{x} = x_0 + x_1\varepsilon_1, \quad (4)$$

with  $x_0 = \frac{x_{\max} + x_{\min}}{2}$ ,  $x_1 = \frac{x_{\max} - x_{\min}}{2}$ .

AA can keep correlations among the signals of the computational chain by contributing the sample noise symbol  $\varepsilon_i$  to each signal [22].

For multiplication, AATRE and AACHA are typical approximation methods.

The affine form of AATRE is

$$\hat{x}\hat{y} = x_0y_0 + \sum_{i=1}^n (x_0y_i + y_0x_i)\varepsilon_i + \sum_{i=1}^n |x_i| \sum_{i=1}^n |y_i|\varepsilon_{n+1}. \quad (5)$$

Suppose  $M_1 = \max(n_1, n_2)$ , in which  $n_1$  and  $n_2$  denote the number of the noise symbol, whose coefficient is nonzero, of  $\hat{x}$  and  $\hat{y}$ , respectively. The computational complexity of AATRE is  $O(M_1)$ .

AACHA provides a better approximation result, but it is more complex. The affine form of AACHA is

$$\hat{x}\hat{y} = x_0y_0 + \sum_{i=1}^n (x_0y_i + y_0x_i)\varepsilon_i + \frac{a+b}{2} + \frac{b-a}{2}\varepsilon_{n+1}, \quad (6)$$

where  $a$  and  $b$  denote the minimum and the maximum of the range of  $\left(\sum_{i=1}^n x_i\varepsilon_i\right)\left(\sum_{i=1}^n y_i\varepsilon_i\right)$ . Suppose  $M_2 = n_1 + n_2$ .

The complexity of computing the both extremal values,  $a$  and  $b$ , is  $O(M_2 \log M_2)$ . As  $M_1 \leq M_2$ , the computational complexity of AATRE is lower than that of AACHA [22].

## 2.4 Extreme value theory

The proposed approximation is based on the extreme value theory of multivariable functions [24].

According to the extreme value theory of multivariable functions, the Hessian matrix of the function,  $\mathbf{H}$ , and Jacobian matrix of the function,  $\mathbf{J}$ , can be used to find

the local maxima and the local minima. Hessian matrix of function  $f(x_1, x_2, \dots, x_n)$  is

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \dots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}. \quad (7)$$

Here we use  $H_{f^\alpha}$  to represent  $H$  at a point  $f^\alpha = (x_1^\alpha, x_2^\alpha, \dots, x_n^\alpha)$  and  $J_{f^\alpha}$  to represent  $J$  at a point  $f^\alpha$ .

A stationary point of  $f, f^\alpha$ , is a point where  $J_{f^\alpha} = 0$ .  $H_{f^\alpha}$  is indefinite when  $H_{f^\alpha}$  is neither positive semidefinite nor negative semidefinite. If  $H_{f^\alpha}$  is positive definite, then  $f^\alpha$  is a local minimum point. If  $H_{f^\alpha}$  is negative definite, then  $f^\alpha$  is a local maximum point. If  $H_{f^\alpha}$  is indefinite, then  $f^\alpha$  is neither a local maximum nor a local minimum. It is a saddle point. Otherwise,  $f^\alpha$  is not utilized in this paper.

The principal minor determinants are used to determine if a matrix is positive or negative definite or semidefinite.

It is necessary and sufficient for a positive semidefinite matrix that all the principal minor determinants of the matrix are nonnegative real numbers.

It is necessary and sufficient for a negative semidefinite matrix that all the odd order principal minor determinants of the matrix are non-positive real numbers and all the even order principal minor determinants of the matrix are nonnegative real numbers.

### 3 Derivation of the two parts for multiplication

A generic nonlinear operation  $z \leftarrow f(\hat{x}, \hat{y})$  proposed in [22] can be described by (8):

$$\begin{aligned} z &= f(x_0 + x_1 \varepsilon_1 + \dots + x_n \varepsilon_n, y_0 + y_1 \varepsilon_1 + \dots + y_n \varepsilon_n) \\ &= f^*(\varepsilon_1, \dots, \varepsilon_n). \end{aligned} \quad (8)$$

Since the operation  $f$  is nonlinear,  $f^*(\varepsilon_1, \dots, \varepsilon_n)$  cannot be expressed exactly as an affine combination of the noise symbols,  $\varepsilon_i$ . Under this case, an approximate affine form of the operation, which is represented as  $f_z$ , must be used to approximate  $f^*(\varepsilon_1, \dots, \varepsilon_n)$ . The difference introduced by this approximation,  $d_f = f^* - f_z$ , can be expressed by an equivalent affine form of the estimated range of the difference, which is represented as  $\hat{d}$ . Hence, the affine form of  $z$  can be expressed as

$$\hat{z} = f_z + \hat{d}. \quad (9)$$

In (9),  $f_z$  is a first-degree function of  $\varepsilon_i$  and can be expressed as (10)

$$f_z(\varepsilon_1, \dots, \varepsilon_n) = z_0 + \sum_{i=1}^n z_i \varepsilon_i. \quad (10)$$

The computational complexity of computing the true range of  $d_f$  is very high in a practical application. The estimated range of  $d_f$  is utilized instead of the true range. Suppose  $d_{\max}$  and  $d_{\min}$  denote the upper and lower bounds of the estimated range of  $d_f$ , respectively. According to (4), the  $\hat{d}$  can be expressed as (11)

$$\hat{d} = z' + z_{n+1} \varepsilon_{n+1} = \frac{d_{\max} + d_{\min}}{2} + \frac{d_{\max} - d_{\min}}{2} \varepsilon_{n+1}. \quad (11)$$

With (10) and (11), the affine form of  $z$  can be represented as

$$\hat{z} = f_z + \hat{d} = z_0 + \sum_{i=1}^n z_i \varepsilon_i + z' + z_{n+1} \varepsilon_{n+1}. \quad (12)$$

For multiplication,  $z$  can be expressed as

$$z = x_0 y_0 + x_0 \sum_{i=1}^n y_i \varepsilon_i + y_0 \sum_{i=1}^n x_i \varepsilon_i + \left( \sum_{i=1}^n x_i \varepsilon_i \right) \left( \sum_{i=1}^n y_i \varepsilon_i \right). \quad (13)$$

The first three items of (13) form an affine form and the last term is a quadratic term. Its affine form can also be represented as (12).

According to the definition of  $f_z$  in (10) and  $\hat{d}$  in (11), AATRE and AACHA can also be represented by  $f_z$  and  $\hat{d}$ . For AATRE in (5), the  $f_z$  and  $\hat{d}$  are defined as

$$f_z = x_0 y_0 + \sum_{i=1}^n (x_0 y_i + y_0 x_i) \varepsilon_i, \quad (14)$$

$$\hat{d} = \sum_{i=1}^n |x_i| \sum_{i=1}^n |y_i| \varepsilon_{n+1}. \quad (15)$$

For AACHA in (6), the  $f_z$  and  $\hat{d}$  are defined as

$$f_z = x_0 y_0 + \sum_{i=1}^n (x_0 y_i + y_0 x_i) \varepsilon_i, \quad (16)$$

$$\hat{d} = \frac{a+b}{2} + \frac{b-a}{2} \varepsilon_{n+1}. \quad (17)$$

In the existing affine approximation methods of AATRE and AACHA,  $d_{\max}$  and  $d_{\min}$  are estimated in the  $XY$  plane. In these methods, the same noise symbol of different variables is considered to be independent. Hence, the range of  $\hat{d}$  is much larger than that of  $d_f$ . The difference between  $\hat{d}$  and  $d_f$  will propagate to  $\hat{z}$  and result in uncertainty.

To describe the multiplication accurately, we use  $\varepsilon_i$  as the input arguments and estimate the range of  $z$  in the  $(n+1)$ -dimensional space  $\mathbf{E}^{n+1}$ . The  $(n+1)$ -dimensional space  $\mathbf{E}^{n+1}$  is labeled as  $(\varepsilon_1, \dots, \varepsilon_n, z)$ . In space  $\mathbf{E}^{n+1}$ , a first-degree polynomial function can be expressed as a

$(n + 1)$ -dimensional hyperplane and a nonlinear polynomial function denotes a  $(n + 1)$ -dimensional space curved surface. The approximate affine form in (10) denotes a  $(n + 1)$ -dimensional hyperplane in  $\mathbf{E}^{n+1}$ . Each hyperplane in  $\mathbf{E}^{n+1}$  can be viewed as a parallel translation of a tangent hyperplane at a certain point of  $(n + 1)$ -dimensional space curved surface. Hence, all possible approximate affine forms for  $z$  can be regarded as the  $(n + 1)$ -dimensional tangent hyperplanes at all points of  $(n + 1)$ -dimensional space curved surface in  $\mathbf{E}^{n+1}$ . The translation amount is taken into account in  $d_f$ , which is approximated by  $\hat{d}$ . In space  $\mathbf{E}^{n+1}$ ,  $d_f$  can be viewed as the function of the distance between the points of space curved surface and the tangent hyperplane.

Figure 1 shows an example of  $\hat{x} = 1 + \varepsilon_1 + 5\varepsilon_2$  and  $\hat{y} = 3 - 6\varepsilon_1 + \varepsilon_2$ . The space is labeled as  $(\varepsilon_1, \varepsilon_2, z)$ . The red mesh surface represents the function  $z = \hat{xy} = (1 + \varepsilon_1 + 5\varepsilon_2)(3 - 6\varepsilon_1 + \varepsilon_2)$ . The blue plane represents the tangent plane  $f_z$ ,  $z = 3 - 3\varepsilon_1 + 16\varepsilon_2$ , at the point  $z^\alpha = (0, 0, 3)$ . All the possible approximate affine forms for  $z$  are the tangent planes of all the points.  $d_f$  is a function of distance between  $z$  and  $f_z$ .

Here we use  $f_{z^\alpha}$  in (18) to represent the tangent hyperplane at the point  $z^\alpha = (\varepsilon_1^\alpha, \varepsilon_2^\alpha, \dots, \varepsilon_n^\alpha)$ . Then, the possible approximate affine form can be represented as  $f_{z^\alpha}$ , too.

$$f_{z^\alpha} = z^\alpha + z'_{\varepsilon_1}(\varepsilon_1 - \varepsilon_1^\alpha) + z'_{\varepsilon_2}(\varepsilon_2 - \varepsilon_2^\alpha) + \dots + z'_{\varepsilon_n}(\varepsilon_n - \varepsilon_n^\alpha). \quad (18)$$

In (18),  $z'_{\varepsilon_n}$  are the partial derivatives of  $z$  with respect to the variables  $\varepsilon_n$  at the point  $z^\alpha$ .

With the estimated range of  $d_f$ , the maximum absolute error of  $d_f$  can be expressed as

$$e_a = \max(|d_{\max}|, |d_{\min}|). \quad (19)$$

To reduce the uncertainty,  $f_z$  must be the most closed to the result of multiplication. Hence,  $f_z$  is the tangent hyperplane whose maximum absolute error is minimum among that of all the possible affine form  $f_{z^\alpha}$ , that is,

$$e_a(f_z) = \min(e_a(f_{z^\alpha})). \quad (20)$$

The geometrical meaning of  $f_z$  denotes the tangent hyperplane whose maximum absolute error is minimized.

$f_z$  is derived by the range of  $d_f$ , while  $\hat{d}$  is the equivalent affine form of  $d_f$ . It is very complex to compute the true range of  $d_f$ . With  $\hat{d}$  in (11), the uncertainty in AA for nonlinear operations is generated due to the difference between the true range of  $d_f$  and the estimated range of  $d_f$ .

It is much tighter and easier to estimate range of  $d_f$  in  $\mathbf{E}^{n+1}$  space than in the  $XY$  plane. Based on the extreme value theory of multivariable functions, the estimated range of  $d_f$  in AASEE is derived.

With more accurate  $d_{\max}$  and  $d_{\min}$ ,  $f_z$  and  $\hat{d}$  can be calculated more precisely, and AASEE can achieve a refined affine approximation result.

In the next sections, the estimated range of  $d_f$  will be derived firstly, and the two parts will be derived later.

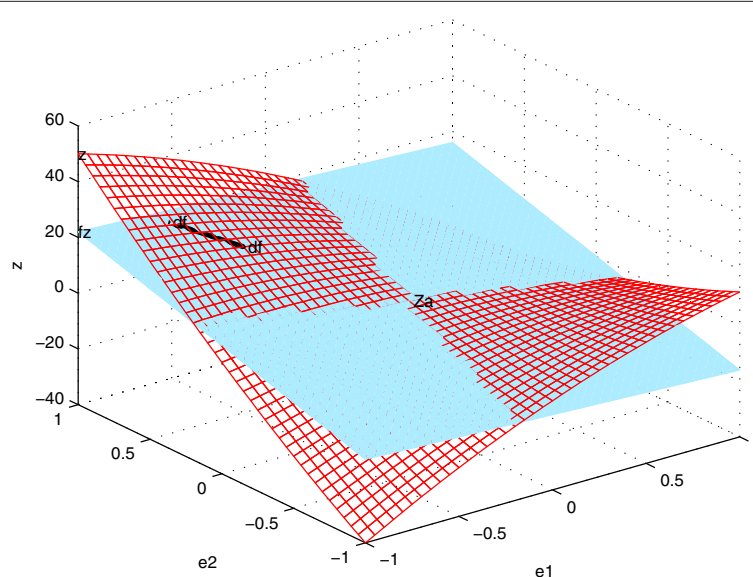


Figure 1 Example of multiplication in  $(n + 1)$ -dimensional space  $\mathbf{E}^{n+1}$ .

#### 4 AAASE for multiplication

##### 4.1 Estimated range of the difference

For multiplication, which is expressed as (13), the value of  $z$  at the point  $z^\alpha$  is

$$z^\alpha = \left( x_0 + \sum_{i=1}^n x_i \varepsilon_i^\alpha \right) \left( y_0 + \sum_{i=1}^n y_i \varepsilon_i^\alpha \right). \quad (21)$$

The partial derivatives of  $z$  with respect to the variable  $\varepsilon_i$  at the point  $z^\alpha$  are

$$z'_{\varepsilon_i} = \left( x_i \left( y_0 + \sum_{j=1}^n y_j \varepsilon_j^\alpha \right) + y_i \left( x_0 + \sum_{j=1}^n x_j \varepsilon_j^\alpha \right) \right). \quad (22)$$

Upon substitution for  $z^\alpha$  and  $z'_{\varepsilon_i}$ , the tangent hyperplane  $f_{z^\alpha}$  can be expressed as

$$\begin{aligned} f_{z^\alpha} = & \left( x_0 + \sum_{i=1}^n x_i \varepsilon_i^\alpha \right) \left( y_0 + \sum_{i=1}^n y_i \varepsilon_i^\alpha \right) + \left( x_1 \left( y_0 + \sum_{i=1}^n y_i \varepsilon_i^\alpha \right) \right. \\ & + y_1 \left( x_0 + \sum_{i=1}^n x_i \varepsilon_i^\alpha \right) \left. \right) (\varepsilon_1 - \varepsilon_1^\alpha) + \dots \\ & + \left( x_n \left( y_0 + \sum_{i=1}^n y_i \varepsilon_i^\alpha \right) + y_n \left( x_0 + \sum_{i=1}^n x_i \varepsilon_i^\alpha \right) \right) (\varepsilon_n - \varepsilon_n^\alpha). \end{aligned} \quad (23)$$

The difference between the tangent hyperplane  $f_{z^\alpha}$  and  $(n + 1)$ -dimensional quadratic surface  $z$  is

$$d_f = z - f_{z^\alpha} = \sum_{i,j=1}^n x_i y_j (\varepsilon_i - \varepsilon_i^\alpha) (\varepsilon_j - \varepsilon_j^\alpha), \quad (24)$$

$$\text{where } \varepsilon_i, \varepsilon_j, \varepsilon_i^\alpha, \varepsilon_j^\alpha = [-1, 1].$$

Suppose  $d_{\text{emax}}$  and  $d_{\text{emin}}$  denote the estimated maximum and minimum of the function value at the domain boundary respectively, and  $d_{\text{fimax}}$  and  $d_{\text{fimin}}$  denote the local maxima and the local minima, respectively. The estimated maximum and minimum of multivariable function  $d_f$ ,  $d_{\text{max}}$  and  $d_{\text{min}}$ , can be expressed as

$$d_{\text{max}} = \max(d_{\text{emax}}, d_{\text{fimax}}), \quad (25)$$

$$d_{\text{min}} = \min(d_{\text{emin}}, d_{\text{fimin}}). \quad (26)$$

According to (24), the function value at the domain boundary,  $d_{f_e}$ , is represented by

$$d_{f_e} = \sum_{i,j=1}^n x_i y_j [\varepsilon_i \varepsilon_j - \varepsilon_j \varepsilon_i^\alpha - \varepsilon_i \varepsilon_j^\alpha + \varepsilon_i^\alpha \varepsilon_j^\alpha] \quad (27)$$

$$\text{where } \exists \varepsilon_i = \pm 1, \forall i = 1, 2, \dots, n.$$

To simplify, we observe the extreme case of  $\forall \varepsilon_i = \pm 1$ . Under this case, for the first item, it is always positive

when  $i = j$ . Hence, the estimated function value at the domain boundary,  $d_e$ , is expressed as

$$\begin{aligned} d_e = & \sum_{i,j=1, i=j}^n x_i y_j + \sum_{i,j=1}^n x_i y_j \varepsilon_i^\alpha \varepsilon_j^\alpha + \sum_{i,j=1, i \neq j}^n x_i y_j \varepsilon_i \varepsilon_j \\ & - \sum_{i,j=1}^n x_i y_j \varepsilon_j \varepsilon_i^\alpha - \sum_{i,j=1}^n x_i y_j \varepsilon_i \varepsilon_j^\alpha \end{aligned} \quad (28)$$

$$\text{where } \forall \varepsilon_i = \pm 1.$$

Hence, the maximum and minimum of  $d_e$ ,  $d_{\text{emax}}$  and  $d_{\text{emin}}$  are derived as

$$\begin{aligned} d_{\text{emax}} = & \sum_{i=1}^n x_i y_i + \sum_{i,j=1}^n x_i y_j \varepsilon_i^\alpha \varepsilon_j^\alpha + \sum_{i,j=1, i \neq j}^n |x_i y_j| \\ & + \sum_{i,j=1}^n |x_i y_j \varepsilon_i^\alpha| + \sum_{i,j=1}^n |x_i y_j \varepsilon_j^\alpha| \end{aligned} \quad (29)$$

$$\begin{aligned} d_{\text{emin}} = & \sum_{i=1}^n x_i y_i + \sum_{i,j=1}^n x_i y_j \varepsilon_i^\alpha \varepsilon_j^\alpha - \sum_{i,j=1, i \neq j}^n |x_i y_j| \\ & - \sum_{i,j=1}^n |x_i y_j \varepsilon_i^\alpha| - \sum_{i,j=1}^n |x_i y_j \varepsilon_j^\alpha|. \end{aligned} \quad (30)$$

To simply compare,  $d_{\text{fimax}}$  and  $d_{\text{fimin}}$  in (25) and (26) can be expressed as

$$\begin{aligned} d_{\text{fimax}} = & \sum_{i=1}^n x_i y_i \varepsilon_i^2 + \sum_{i,j=1, i \neq j}^n x_i y_j \varepsilon_i \varepsilon_j + \sum_{i,j=1}^n x_i y_j \varepsilon_i \varepsilon_j^\alpha \\ & + \sum_{i,j=1}^n x_i y_j \varepsilon_j \varepsilon_i^\alpha + \sum_{i,j=1}^n x_i y_j \varepsilon_i^\alpha \varepsilon_j^\alpha, \end{aligned} \quad (31)$$

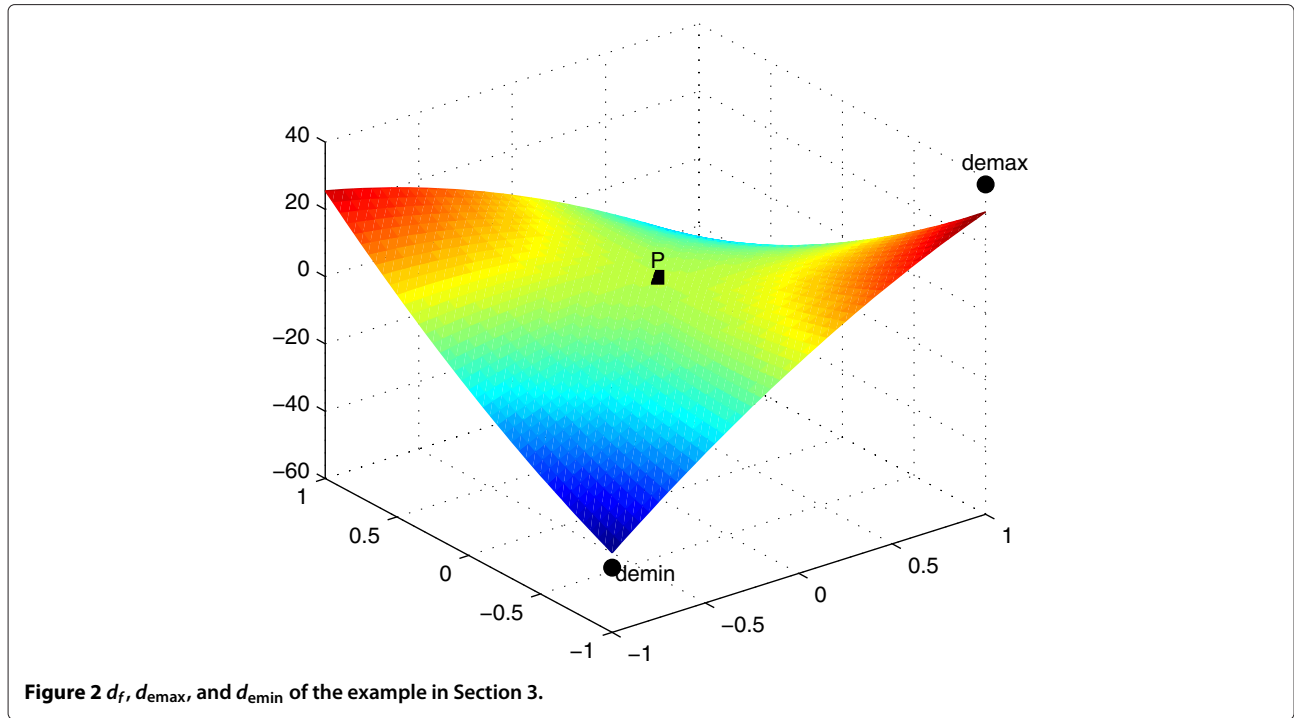
$$\begin{aligned} d_{\text{fimin}} = & \sum_{i=1}^n x_i y_i \varepsilon_i^2 + \sum_{i,j=1, i \neq j}^n x_i y_j \varepsilon_i \varepsilon_j + \sum_{i=1}^n x_i y_i \varepsilon_i (\varepsilon_i^\alpha + \varepsilon_j^\alpha) \\ & + \sum_{i,j=1, i \neq j}^n x_i y_j \varepsilon_i \varepsilon_j^\alpha + \sum_{i,j=1, i \neq j}^n x_i y_j \varepsilon_j \varepsilon_i^\alpha + \sum_{i,j=1}^n x_i y_i \varepsilon_i^\alpha \varepsilon_j^\alpha, \end{aligned}$$

$$\text{where } \varepsilon_i, \varepsilon_j = (-1, 1), \text{ and } \varepsilon_i^\alpha, \varepsilon_j^\alpha = [-1, 1].$$

$$(32)$$

As the example in Section 3, Figure 2 shows the function of  $d_f = -6(\varepsilon_1 - 0.1)^2 - 29(\varepsilon_1 - 0.1)(\varepsilon_2 - 0.1) + 5(\varepsilon_2 - 0.1)^2$  when  $\varepsilon_1^\alpha = 0.1$  and  $\varepsilon_2^\alpha = 0.1$ . The estimated maximum and minimum of  $d_f$  at the domain boundary,  $d_{\text{emax}}$  and  $d_{\text{emin}}$ , are also marked in the figure. Since the value of  $\varepsilon_i$  in (27) are substituted by  $\forall \varepsilon_i = \pm 1$ ,  $d_{\text{emax}}$  is larger than the maximum of  $d_f$  and  $d_{\text{emin}}$  is smaller than the minimum.

The extreme value theory of multivariable functions is used to compare  $d_{\text{emax}}$ ,  $d_{\text{fimax}}$ ,  $d_{\text{emin}}$ , and  $d_{\text{fimin}}$ .



Hessian matrix of function  $d_f = \sum_{i,j=1}^n x_i y_j (\varepsilon_i - \varepsilon_i^\alpha) (\varepsilon_j - \varepsilon_j^\alpha)$

is

$$\begin{aligned}
 \mathbf{H} &= \begin{bmatrix} \frac{\partial^2 d_f}{\partial \varepsilon_1^2} & \frac{\partial^2 d_f}{\partial \varepsilon_1 \partial \varepsilon_2} & \cdots & \frac{\partial^2 d_f}{\partial \varepsilon_1 \partial \varepsilon_n} \\ \frac{\partial^2 d_f}{\partial \varepsilon_2 \partial \varepsilon_1} & \frac{\partial^2 d_f}{\partial \varepsilon_2^2} & \cdots & \frac{\partial^2 d_f}{\partial \varepsilon_2 \partial \varepsilon_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 d_f}{\partial \varepsilon_n \partial \varepsilon_1} & \frac{\partial^2 d_f}{\partial \varepsilon_n \partial \varepsilon_2} & \cdots & \frac{\partial^2 d_f}{\partial \varepsilon_n^2} \end{bmatrix} \\
 &= \begin{bmatrix} 2x_1 y_1 & x_1 y_2 + x_2 y_1 & x_1 y_3 + x_3 y_1 & \cdots \\ x_1 y_2 + x_2 y_1 & 2x_2 y_2 & x_2 y_3 + x_3 y_2 & \cdots \\ x_1 y_3 + x_3 y_1 & x_2 y_3 + x_3 y_2 & 2x_3 y_3 & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ x_1 y_n + x_n y_1 & x_2 y_n + x_n y_2 & x_3 y_n + x_n y_3 & \cdots \end{bmatrix}. \quad (33)
 \end{aligned}$$

From (33), we can see that  $\mathbf{H}$  is independent of  $\varepsilon_i$ . It is a expression of  $x_i$  and  $y_i$ . This means that  $\mathbf{H}$  is same for all the points in the domain.

To determine if  $\mathbf{H}$  is positive or negative definite or semidefinite, its principal minor determinants are derived as

$$D_0 = 2x_i y_i \quad (34)$$

$$D_1 = \begin{vmatrix} 2x_i y_i & x_i y_j + x_j y_i \\ x_i y_j + x_j y_i & 2x_j y_j \end{vmatrix} = -(x_i y_j - x_j y_i)^2 \quad (35)$$

$$D_2 = D_3 = \cdots = D_n = 0, \quad (36)$$

where  $1 \leq i < j \leq n$ .

As introduced in Section 2.4,  $\mathbf{H}$  is a positive semidefinite matrix, iff it satisfies

$$\forall x_i y_i \geq 0, \forall x_i y_j = x_j y_i, \quad \text{for } 1 \leq i < j \leq n. \quad (37)$$

$\mathbf{H}$  is a negative semidefinite matrix, iff it satisfies

$$\forall x_i y_i \geq 0, \exists x_i y_j \neq x_j y_i, \quad \text{for } 1 \leq i < j \leq n. \quad (38)$$

If it satisfies neither (37) nor (38), which means it satisfies (39),  $\mathbf{H}$  is an indefinite matrix as

$$\exists x_i y_i < 0, \quad \text{for } 1 \leq i \leq n. \quad (39)$$

According to (37), (38), and (39), we can compare  $d_{\max}$ ,  $d_{\min}$ ,  $d_{\text{fimax}}$ , and  $d_{\text{fimin}}$ , which are expressed as (29), (30), (31), and (32), respectively. Based on (25) and (26),  $d_{\max}$  and  $d_{\min}$  can be identified.

**Lemma 1.** *The estimated maximum of function  $d_f$ ,  $d_{\max}$  equals to the estimated maximum of the function value at the domain boundary, and the estimated minimum of function  $d_f$ ,  $d_{\min}$  equals to the estimated minimum of the function value at the domain boundary. This can be expressed as*

$$d_{\max} = d_{\text{emax}} \quad d_{\min} = d_{\text{emin}}. \quad (40)$$

*Proof.* There are two cases to consider, as  $\exists x_i y_i < 0$  and  $\forall x_i y_i \geq 0$ .

For  $\exists x_i y_i < 0$ , (39) is satisfied and  $\mathbf{H}$  is indefinite. The stationary point is a saddle point, such as the point  $\mathbf{P}$  in Figure 2. Neither  $d_{\text{fimax}}$  nor  $d_{\text{fimin}}$  exists in  $d_f$ , that is,

$$d_{\text{max}} = d_{\text{emax}} \quad d_{\text{min}} = d_{\text{emin}}. \quad (41)$$

According to (41), Lemma 1 can be proven in this case.

For  $\forall x_i y_i \geq 0$ ,  $\mathbf{H}$  may be positive semidefinite or negative semidefinite.  $d_f$  may have local minima or local maxima under this condition.

As  $\varepsilon_i = [-1, 1]$ , the following inequalities are established:

$$\sum_{i,j=1, i \neq j}^n |x_i y_j| \geq \pm \sum_{i,j=1, i \neq j}^n x_i y_j \varepsilon_i \varepsilon_j, \quad (42)$$

$$\sum_{i,j=1}^n |x_i y_j \varepsilon_i^\alpha| \geq \pm \sum_{i,j=1}^n x_i y_j \varepsilon_i \varepsilon_j^\alpha, \quad (43)$$

$$\sum_{i,j=1}^n |x_i y_j \varepsilon_j^\alpha| \geq \pm \sum_{i,j=1}^n x_i y_j \varepsilon_j \varepsilon_i^\alpha. \quad (44)$$

If a local maximum lies at  $z^\alpha$ , the difference between  $d_{\text{emax}}$  and  $d_{\text{fimax}}$  is

$$d_{\text{emax}} - d_{\text{fimax}} \geq \sum_{i=1}^n x_i y_i (1 - \varepsilon_i^2). \quad (45)$$

$\forall x_i y_i \geq 0$ , there exists

$$d_{\text{emax}} \geq d_{\text{fimax}}. \quad (46)$$

According to (25) and (46), we can prove that

$$d_{\text{max}} = d_{\text{emax}}. \quad (47)$$

Similarly, if a local minimum lies at  $z^\alpha$ , the difference between  $d_{\text{emin}}$  and  $d_{\text{fimin}}$  is

$$\begin{aligned} d_{\text{emin}} - d_{\text{fimin}} &\leq - \sum_{i=1}^n x_i y_i (\varepsilon_i^2 + \varepsilon_i (\varepsilon_i^\alpha + \varepsilon_j^\alpha) + 1) \\ &\leq - \sum_{i=1}^n x_i y_i (\varepsilon_i + 1)^2. \end{aligned} \quad (48)$$

As  $\forall x_i y_i \geq 0$  in (48), the inequality (49) can be proven:

$$d_{\text{emin}} \leq d_{\text{fimin}}. \quad (49)$$

According to (26) and (49), we can prove that

$$d_{\text{min}} = d_{\text{emin}}. \quad (50)$$

As (47) and (50) are established, Lemma 1 can be proven in the case of  $\forall x_1 y_1 \geq 0$ .

Combining these two cases, Lemma 1 is proven.  $\square$

According to Lemma 1,  $d_{\text{max}}$  and  $d_{\text{min}}$  at a point  $z^\alpha$  can be computed as  $d_{\text{emax}}$  and  $d_{\text{emin}}$  in (29) and (30).

## 4.2 Expression of the approximate affine form in AASEE

**Lemma 2.** When  $f_z$  represents a tangent hyperplane at the point  $z^0 = z_0 = (0, 0, \dots, 0)$ , it satisfies (20).

*Proof.* According to Lemma 1, (29), and (30), the maximum absolute error of  $d_f$  is

$$\begin{aligned} e_a &= \left| \sum_{i=1}^n x_i y_i \right| + \sum_{i,j=1, i \neq j}^n |x_i y_j| + \sum_{i,j=1}^n |x_i y_j \varepsilon_i^\alpha| \\ &\quad + \sum_{i,j=1}^n |x_i y_j \varepsilon_j^\alpha| + \left| \sum_{i,j=1}^n x_i y_j \varepsilon_i^\alpha \varepsilon_j^\alpha \right|. \end{aligned} \quad (51)$$

So the maximum absolute error between the tangent hyperplane  $f_{z^0}$  at the point  $z^0 = z_0 = (0, 0, \dots, 0)$  and  $(n+1)$ -dimensional quadratic surface  $z$  is

$$e_a(z^0) = \left| \sum_{i=1}^n x_i y_i \right| + \sum_{i,j=1, i \neq j}^n |x_i y_j|. \quad (52)$$

Suppose that there is another point  $z^\alpha \neq z^0$ , which is typically represented by  $z^\alpha = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$ , where  $\varepsilon_i = [-1, 1]$ , and  $\varepsilon_i$  cannot be equal to 0 for all  $i, i = 1 \dots n$ . The maximum absolute error between the tangent hyperplane  $f_{z^\alpha}$  at point  $z^\alpha$  and  $(n+1)$ -dimensional quadratic surface  $\hat{x}\hat{y}$  is

$$\begin{aligned} e_a(z^\alpha) &= \left| \sum_{i=1}^n x_i y_i \right| + \sum_{i,j=1, i \neq j}^n |x_i y_j| + \sum_{i,j=1}^n |x_i y_j \varepsilon_i^\alpha| \\ &\quad + \sum_{i,j=1}^n |x_i y_j \varepsilon_j^\alpha| + \left| \sum_{i,j=1}^n x_i y_j \varepsilon_i^\alpha \varepsilon_j^\alpha \right|. \end{aligned} \quad (53)$$

$e_a(z^\alpha)$  and  $e_a(z^0)$  can be compared by

$$\begin{aligned} e_a(z^0) - e_a(z^\alpha) &= - \sum_{i,j=1}^n |x_i y_j \varepsilon_i^\alpha| \\ &\quad - \sum_{i,j=1}^n |x_i y_j \varepsilon_j^\alpha| - \left| \sum_{i,j=1}^n x_i y_j \varepsilon_i^\alpha \varepsilon_j^\alpha \right| \leq 0. \end{aligned} \quad (54)$$

Because  $e_a(z^0) \leq e_a(z^\alpha)$ , the tangent hyperplane  $f_{z^0}$  at the point  $z^0 = z_0 = (0, 0, \dots, 0)$  is the tangent hyperplane whose maximum absolute error is minimized.

It is proven that the chosen  $f_z$  is a tangent hyperplane at the point  $z^0 = z_0 = (0, 0, \dots, 0)$ .  $\square$

According to Lemma 2,  $f_z$  of AASEE denotes the tangent hyperplane at the point  $z_0 = (0, 0, \dots, 0)$  and can be expressed as

$$f_z = x_0 y_0 + x_0 \sum_{i=1}^n y_i \varepsilon_i + y_0 \sum_{i=1}^n x_i \varepsilon_i. \quad (55)$$

This  $f_z$  is the same as the  $f_z$ s in AATRE and AACHA.



### 4.3 Expression of the equivalent affine form in AASEE

According to (55), the  $d_f$  between the tangent hyperplane  $f_{z_0}$  and the quadratic surface is

$$d_f = \sum_{i,j=1}^n x_i y_j \varepsilon_i \varepsilon_j. \quad (56)$$

According to Lemma 1, (29), and (30), the estimated maximum and estimated minimum of  $d_f$ ,  $d_{\max}$  and  $d_{\min}$  can be expressed as

$$d_{\max} = d_{\max} = \sum_{i=1}^n x_i y_i + \sum_{i,j=1, i \neq j}^n |x_i y_j| \quad (57)$$

$$d_{\min} = d_{\min} = \sum_{i=1}^n x_i y_i - \sum_{i,j=1, i \neq j}^n |x_i y_j|.$$

$n = 1$  is a special case and  $d_{\max}$  and  $d_{\min}$  can be optimized as

$$d_{\max} = \begin{cases} x_1 y_1, & \text{for } n = 1, x_1 y_1 \geq 0 \\ 0, & \text{for } n = 1, x_1 y_1 \leq 0 \end{cases} \quad (58)$$

$$d_{\min} = \begin{cases} 0, & \text{for } n = 1, x_1 y_1 \geq 0 \\ x_1 y_1, & \text{for } n = 1, x_1 y_1 \leq 0. \end{cases} \quad (59)$$

By combining the two cases,  $d_{\max}$  and  $d_{\min}$  are rewritten as

$$d_{\max} = \begin{cases} \sum_{i=1}^n x_i y_i + \sum_{i,j=1, i \neq j}^n |x_i y_j|, & \text{for } n > 1 \\ x_1 y_1, & \text{for } n = 1, x_1 y_1 \geq 0 \\ 0, & \text{for } n = 1, x_1 y_1 < 0 \end{cases} \quad (60)$$

$$d_{\min} = \begin{cases} \sum_{i=1}^n x_i y_i - \sum_{i,j=1, i \neq j}^n |x_i y_j|, & \text{for } n > 1 \\ 0, & \text{for } n = 1, x_1 y_1 \geq 0 \\ x_1 y_1, & \text{for } n = 1, x_1 y_1 < 0. \end{cases} \quad (61)$$

When  $n > 1$ , the range of  $\hat{d}$  can be expressed as

$$\left[ \sum_{i=1}^n x_i y_i - \sum_{i,j=1, i \neq j}^n |x_i y_j|, \sum_{i=1}^n x_i y_i + \sum_{i,j=1, i \neq j}^n |x_i y_j| \right]. \quad (62)$$

According to (11), the affine form of  $\hat{d}$  can be expressed as

$$\hat{d} = \sum_{i=1}^n x_i y_i + \sum_{i,j=1, i \neq j}^n |x_i y_j| \varepsilon_{n+1}. \quad (63)$$

When  $n = 1$ , the range of  $\hat{d}$  can be expressed as

$$[x_1 y_1, 0] \quad \text{or} \quad [0, x_1 y_1]. \quad (64)$$

The affine form of  $\hat{d}$  can be expressed as

$$\hat{d} = \frac{1}{2} x_1 y_1 + \frac{1}{2} |x_1 y_1| \varepsilon_2. \quad (65)$$

### 4.4 Formulary of AASEE

According to (12), the affine form of AASEE for multiplication is

$$\hat{z} = f_z + \hat{d} = x_0 y_0 + x_0 \sum_{i=1}^n y_i \varepsilon_i + y_0 \sum_{i=1}^n x_i \varepsilon_i \quad (66)$$

$$+ \sum_{i=1}^n x_i y_i + \sum_{i,j=1, i \neq j}^n |x_i y_j| \varepsilon_{n+1} \quad \text{for } n > 1,$$

$$\hat{z} = f_z + \hat{d} = x_0 y_0 + (x_0 y_1 + y_0 x_1) \varepsilon_1 + \frac{1}{2} x_1 y_1 + \frac{1}{2} |x_1 y_1| \varepsilon_2 \quad \text{for } n = 1. \quad (67)$$

It is impossible to obtain the exact affine form for multiplication in AA. The result of multiplication must be approximated to an affine form. Using  $\varepsilon_i$  as the input arguments, the uncertainty of multiplication in AASEE is reduced. The proposed  $f_z$  is the most closed to the result of multiplication among all the possible approximate affine forms, and the upper and lower bounds of  $\hat{d}$  in AASEE are much closer to true bounds of  $d_f$ . Hence, the uncertainty in AASEE is smaller than that in AATRE and AACHA. Formed by such  $f_z$  and  $\hat{d}$ , AASEE creates a refined affine form of multiplication.

## 5 Comparison of AASEE to AATRE and AACHA

### 5.1 Computational complexity

The computational complexity of an expression is determined by its most complex item. For  $n > 1$ , the most complex item is the coefficient of  $\varepsilon_{n+1}$ . To make the analysis convenient, we transform this coefficient:

$$\begin{aligned} \sum_{i,j=1, i \neq j}^n |x_i y_j| &= \sum_{i,j=1}^n |x_i y_j| - \sum_{i=1}^n |x_i y_i| \\ &= \sum_{i,j=1}^n |x_i| \sum_{i,j=1}^n |y_j| - \sum_{i=1}^n |x_i y_i|. \end{aligned} \quad (68)$$

The computational complexity of the minuend is  $O(M_1)$ , where  $M_1$  is defined in Section 2.3, while the computational complexity of the subtrahend is less than  $O(M_1)$ .

Hence, the computational complexity of AASEE is  $O(M_1)$ . We can see that it is the same as that of AATRE and is lower than that of AACHA.

### 5.2 Accuracy

The accuracy of  $\hat{d}$  is influential to the accuracy of the affine approximation methods of multiplication. The more accurate  $\hat{d}$  will lead to a more accurate the affine approximation result.

For AATRE,  $\hat{d} = \sum_{i=1}^n |x_i| \sum_{i=1}^n |y_i| \varepsilon_{n+1}$ . In this method, the same noise symbol of different variables is considered to be independent. The range of this  $\hat{d}$  is

$$\left[ -\sum_{i=1}^n |x_i| \sum_{i=1}^n |y_i|, \sum_{i=1}^n |x_i| \sum_{i=1}^n |y_i| \right]. \quad (69)$$

It is much larger than the range of  $\hat{d}$  by AASEE, which is expressed in (62) and (64).

In AACHA,  $\hat{d} = \frac{a+b}{2} + \frac{b-a}{2} \varepsilon_{n+1}$ , where  $a$  and  $b$  are represented the estimated range of  $\hat{d}$ . In this method, a polygon in  $XY$  plane is used to find  $a$  and  $b$ . The domain of  $\hat{x}\hat{y}$  is bounded by the polygon. However, the polygon is larger than the true domain, and all the same noise symbols of different variables are not taken into account together.

All the same noise symbols of different variables are considered together by  $\hat{d}$  of AASEE. It is more accurate than  $\hat{d}$  of AATRE. In the most cases, it is more accurate than  $\hat{d}$  of AACHA, too.

## 6 Case studies

The following nonlinear system cases are used to demonstrate the efficiency of the proposed refined affine form of multiplication. These cases are commonly used in signal processing. The first two cases are univariate cases and come from [11]. The rest of cases are multivariate polynomial functions and come from [27-29].

### 6.1 Introduction of the cases

*Case 1. Polynomial approximation.* The first case study is that degree-four polynomial for the approximation of  $y = \ln(1+x)$ , where  $x = [0, 1]$ . Horner's rule evaluates the polynomial

$$y = (((-0.0550x+0.2168)x-0.4645)x+0.9956)x+0.0001,$$

where the coefficients are obtained by polynomial curve fitting technique.

*Case 2. B-splines* Uniform cubic B-splines are commonly used for image warping [30]. Basic functions  $B_0$ ,  $B_1$ ,  $B_2$ , and  $B_3$  in B-spline are defined as

$$B_0(u) = \frac{1}{6}(1-u)^3, \quad B_1(u) = \frac{1}{6}(3u^3 - 6u^2 + 4),$$

$$B_2(u) = \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1), \quad B_3(u) = \frac{-u^3}{6},$$

where  $u = [0, 1]$ .

*Case 3. Multivariate polynomial functions.* In the third case, eight multivariate polynomial functions are examined. They are as follows:

1. Savitzky-Golay filter:

$$f_1(X) = 7x_1^3 - 984x_2^3 - 76x_1^2x_2 + 92x_1x_2^2 + 7x_1^2$$

$$- 39x_1x_2 - 46x_2^2 + 7x_1 - 46x_2 - 75$$

where the input range:  $X = [-2, 2]^2$

2. Image rejection unit:

$$f_2(X) = 16384(x_1^4 + x_2^4) + 64767(x_1^2 - x_2^2) + x_1 - x_2$$

$$+ 57344x_1x_2(x_1 - x_2)$$

where the input range:  $X = [0, 1]^2$

3. A random function:

$$f_3(X) = (x_1 - 1)(x_1 + 2)(x_2 + 1)(x_2 - 2)x_3^2$$

where the input range:  $X = [-2, 2]^3$

4. Mitchell function:

$$f_4(X) = 4 \left[ x_1^4 + (x_2^2 + x_3^2)^2 \right] + 17x_1^2(x_2^2 + x_3^2)$$

$$- 20(x_1^2 + x_2^2 + x_3^2) + 17$$

where the input range:  $X = [-2, 2]^3$

5. Matyas function:

$$f_5(X) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$$

where the input range:  $X = [-100, 100]^2$

6. Three-hump function:

$$f_6(X) = 12x_1^2 - 6.3x_1^4 + x_1^6 + 6x_2(x_2 - x_1)$$

where the input range:  $X = [-10, 10]^2$

7. Goldstein-Price function:

$$f_7(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2$$

$$+ 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2$$

$$\times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

where the input range:  $X = [-2, 2]^2$

8. Ratscheck function:

$$f_8(X) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

where the input range:  $X = [-100, 100]^2$

### 6.2 Analysis of case 1

For the input range  $x = [0, 1]$ , equivalent affine form is  $\hat{x} = 0.5 + 0.5\varepsilon_1$ . For case 1, the intermediate and output signals are defined as

$$y_1 = -0.0550x + 0.2168, \quad y_2 = y_1x - 0.4645,$$

$$y_3 = y_2x + 0.9956, \quad y = y_3x + 0.0001. \quad (70)$$

Using AATRE, the affine forms of intermediate and output are

$$\begin{aligned} y_1 &= 0.1893 - 0.0275\varepsilon_1, \\ y_2 &= -0.36985 + 0.0809\varepsilon_1 + 0.01375\varepsilon_2, \\ y_3 &= 0.81068 - 0.14448\varepsilon_1 + 0.00688\varepsilon_2 + 0.04733\varepsilon_3, \\ y &= 0.4054 + 0.3331\varepsilon_1 + 0.0034\varepsilon_2 + 0.0237\varepsilon_3 + 0.0993\varepsilon_4. \end{aligned}$$

Using AACHA, the affine forms of intermediate and output are

$$\begin{aligned} y_1 &= 0.1893 - 0.0275\varepsilon_1, \\ y_2 &= -0.3768 + 0.0809\varepsilon_1 + 0.0069\varepsilon_2, \\ y_3 &= 0.8291 - 0.1479\varepsilon_1 + 0.0034\varepsilon_2 + 0.0220\varepsilon_3, \\ y &= 0.3761 + 0.3406\varepsilon_1 + 0.0017\varepsilon_2 + 0.0110\varepsilon_3 + 0.0436\varepsilon_4. \end{aligned}$$

Using AASEE, the affine forms of intermediate and output are

$$\begin{aligned} y_1 &= 0.1893 - 0.0275\varepsilon_1, \\ y_2 &= -0.37673 + 0.0809\varepsilon_1 + 0.00688\varepsilon_2, \\ y_3 &= 0.84769 - 0.14791\varepsilon_1 + 0.00344\varepsilon_2 + 0.00344\varepsilon_3, \\ y &= 0.34999 + 0.34989\varepsilon_1 + 0.00172(\varepsilon_2 + \varepsilon_3) + 0.00344\varepsilon_4. \end{aligned}$$

Table 1 shows the variable ranges and the range intervals,  $(y_{\max} - y_{\min})$ , of intermediates and output by the three methods. The true range of  $y$  lies in  $[0, 0.6931]$ , and the range interval of output is 0.6931. Suppose  $R(T)$ ,  $R(C)$ , and  $R(A)$  are represented as the ratios of range interval obtained by AATRE, AACHA, and AASEE to the true range interval, respectively. The closer this ratio converges to 1, the more accurate the method is. In this case, as  $R(T) = 1.33$ ,  $R(C) = 1.15$ , and  $R(A) = 1.03$ , we can see the range by AASEE is closer to the true range than AATRE and AACHA.

### 6.3 Comparison of range and computational complexity by the three cases

The output ranges by the three methods of case 2 and case 3 can be obtained according to the process of case 1.

Table 2 demonstrates the ranges and the integer word lengths by AASEE and comparison among AATRE, AACHA and AASEE. Column *c<sub>f</sub>fun* shows the case study and the function of the row. The true output ranges, which are used as reference values, are obtained by numerical

method or nonlinear programming technique, which are time-consuming and are not practical to solve the true bounds for large number of signals. From the table, we can see that the ranges, which are derived by AASEE, cover the true ranges and they are smaller than those by AATRE, for all the functions. For these thirteen functions, the ranges, which are derived by AASEE, are smaller than those by AACHA for nine functions, and equal to those by AACHA for two functions. According to (1), the integer word length can be decided by the range. The integer word-length, which is derived by AASEE, is 2 b less than that by AATRE and 1 b less than that by AACHA, at most. Comparing with AATRE, AASEE and AACHA can save 0.54 b on average.

To calculate the estimated range of  $d_f$ , the values of  $\exists \varepsilon_i = \pm 1, \forall i = 1, 2, \dots, n$  in (27) are substituted by  $\forall \varepsilon_i = \pm 1$  in AASEE. The difference between the estimated range and the true range of  $d_f$  is introduced by this approximation. In most of the applications, the estimated ranges, which are computed by AASEE, are closer than those by AACHA. However, the estimated minimum and maximum of  $\hat{x}\hat{y}$  on the boundary of the polygon are independent of the value of  $\varepsilon_i$ . In some applications such as functions  $f_2$  and  $f_8$  in Table 2, the results by AASEE are almost the same as those by AACHA.

In Table 3, ratios of range intervals and the computational complexity are compared among AATRE, AACHA, and AASEE. The computational complexity is calculated from the numbers of multiplications and additions. For AACHA, the extreme value of a quadratic function in one variable on a bounded interval needs to be calculated.  $N_m$ ,  $N_a$ , and  $N_e$  denote the numbers of multiplications, additions and the extreme value computations of each case, respectively. Table 3 shows that  $R(T)$  values are from 1.04 to 281.2,  $R(C)$  are from 1.03 to 233.7, and  $R(A)$  are from 1.03 to 192.9. The ratios of  $R(A)$  to  $R(T)$  and  $R(C)$  show the accuracy of AASEE compared to AATRE and AACHA, respectively. The average ratios can be used to evaluate the accuracy of the affine approximation methods. The ratios of  $R(A)$  to  $R(T)$  are from 0.18 to 0.99, and the average of these ratios is 0.59. The ratios of  $R(A)$  to  $R(C)$  are from 0.33 to 1.17, and the average of these ratios is 0.89. For these 13 cases, on average, the accuracy of AASEE is 1.69 times than that of AATRE and 1.12

**Table 1 Comparison of ranges and range intervals for every variable of the three methods for case 1**

Variable	AATRE		AACHA		AASEE	
	Range	Interval	Range	Interval	Range	Interval
$y_1$	[0.1618, 0.2168]	0.055	[0.1618, 0.2168]	0.055	[0.1618, 0.2168]	0.055
$y_2$	[-0.4645, -0.2752]	0.1893	[-0.4645, -0.2890]	0.1755	[-0.4645, -0.2890]	0.1755
$y_3$	[0.6120, 1.0094]	0.3974	[0.6558, 1.0025]	0.3467	[0.6929, 1.0025]	0.3096
$y$	[-0.0541, 0.8650]	0.9191	[-0.0253, 0.7685]	0.7938	[-0.0068, 0.7068]	0.7136

**Table 2 Comparison of analytical ranges and bits by the three methods**

c.fun	True output		AATRE		AACHA		AASEE	
	Range	Bits	Range	Bits	Range	Bits	Range	Bits
$c_1.Y$	[0, 0.6931]	1	[-0.0541, 0.8650]	1	[-0.0253, 0.7685]	1	[-0.0068, 0.7068]	1
$c_2.B_0$	[0, 0.17]	1	[-0.13, 0.17]	1	[-0.05, 0.17]	1	[-0.02, 0.17]	1
$c_2.B_1$	[0.17, 0.67]	1	[-0.33, 1.29]	2	[-0.05, 0.98]	1	[0.10, 0.92]	1
$c_2.B_2$	[0.17, 0.67]	1	[-0.21, 1.17]	2	[-0.02, 0.89]	1	[0.04, 0.73]	1
$c_2.B_3$	[-0.17, 0]	1	[-0.17, 0.13]	1	[-0.17, 0.05]	1	[-0.17, 0.02]	1
$c_3.f_1$	[-9453, 9303]	15	[-9821, 9671]	15	[-9793, 9487]	15	[-9793, 9487]	15
$c_3.f_2$	[-5.51e4, 8.79e4]	18	[-1.75e5, 1.79e5]	19	[-0.95e5, 1.28e5]	18	[-1.15e5, 1.41e5]	19
$c_3.f_3$	[-36, 64]	8	[-256, 256]	10	[-192, 192]	9	[-64, 64]	8
$c_3.f_4$	[-8, 641]	11	[-1087, 1121]	12	[-223, 881]	11	[-335, 641]	11
$c_3.f_5$	[0, 10 <sup>4</sup> ]	15	[-10 <sup>4</sup> , 10 <sup>4</sup> ]	15	[-4800, 10 <sup>4</sup> ]	15	[-4800, 10 <sup>4</sup> ]	15
$c_3.f_6$	[0, 0.94e6]	21	[-1.07e6, 1.07e6]	22	[-0.06e6, 1.00e6]	21	[-0.11e6, 0.94e6]	21
$c_3.f_7$	[3, 1.01e6]	21	[-1.42e8, 1.42e8]	29	[-1.23e8, 1.13e8]	28	[-9.87e7, 9.61e7]	28
$c_3.f_8$	[-1.03, 3.3e11]	40	[-3.3e11, 3.3e11]	40	[-2.1e8, 3.3e11]	40	[-4.2e10, 3.3e11]	40

c.fun, case study and the function of the row.

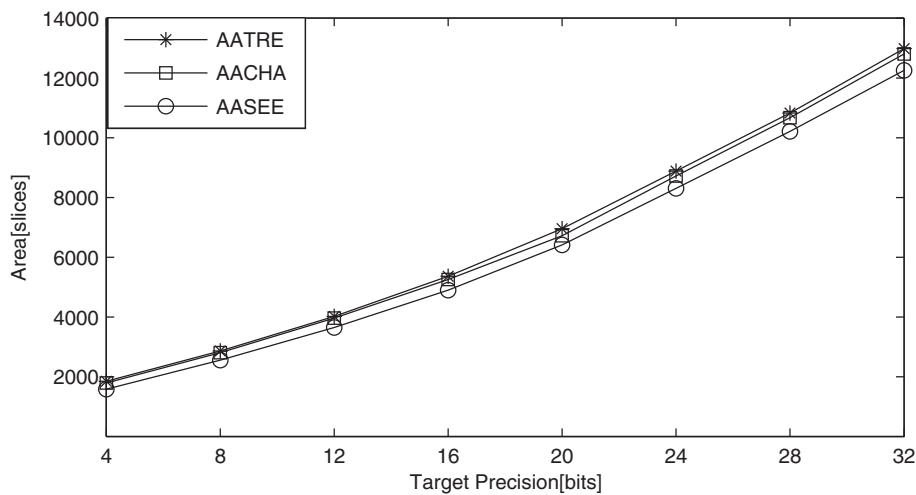
times than that of AACHA. The extreme value computation, which is only necessary for AACHA, of the quadratic function is the most complex and time-consuming among the operations. Hence, the computational complexity of AACHA is much higher than that of AATRE and AASEE. The increase rate of the number of multiplications,  $N_m$ , by AASEE to AATRE is from 0.091 to 1.75, and the average

is 0.450. The increase rate of the number of multiplications,  $N_m$ , by AASEE to AACHA is from 0.2 to 1.833, and the average is 0.567. The increase rate of the number of additions,  $N_a$ , by AASEE to AATRE is from 0.05 to 3.4, and the average is 0.944. The increase rate of the number of additions,  $N_a$ , by AASEE to AACHA is from 0 to 0.985, and the average is 0.157. The numbers of multipli-

**Table 3 Comparison of range ratios and computational complexity by the three methods**

c.fun	AATRE			AACHA				AASEE		
	R(T)	$N_m$	$N_a$	R(C)	$N_e$	$N_m$	$N_a$	R(A)	$N_m$	$N_a$
$c_1.Y$	1.33	17	20	1.15	18	14	17	1.03	20	21
$c_2.B_0$	1.76	13	8	1.29	7	11	10	1.12	15	10
$c_2.B_1$	3.24	20	11	2.06	7	18	13	1.64	22	13
$c_2.B_2$	2.76	22	13	1.82	7	20	15	1.38	24	15
$c_2.B_3$	1.76	13	7	1.29	7	17	9	1.12	15	9
$c_3.f_1$	1.04	16	9	1.03	22	13	15	1.03	22	21
$c_3.f_2$	2.48	42	54	1.56	16	36	51	1.79	46	59
$c_3.f_3$	5.12	17	19	3.84	21	19	23	1.28	23	27
$c_3.f_4$	3.4	12	12	1.7	9	22	34	1.5	33	37
$c_3.f_5$	2	6	2	1.48	6	6	4	1.48	9	4
$c_3.f_6$	2.28	8	5	1.13	12	6	16	1.12	17	16
$c_3.f_7$	281.2	48	72	233.7	36	40	66	192.9	67	131
$c_3.f_8$	2	12	5	1	14	9	22	1.39	19	22

c.fun, case study and the function of the row.



**Figure 3** Area variation for  $c_3.f_3$  with increasing target precision.

cations and additions of AASEE are increased a few. As shown in Table 3, AACHA is slightly more accurate for functions  $c_3.f_2$  and  $c_3.f_8$ , but the computational complexity of AACHA is much higher than that of AASEE.

#### 6.4 Comparison of the design cost by the three methods

To compare the design cost, the system area by the three methods, the fractional word lengths are obtained by the precise analysis in [11]. Typically, we select the case of a random function of case 3,  $c_3.f_3$ , for this section. The design of  $c_3.f_3$  is synthesized on Xilinx Xc2vp30-7ff896 FPGA device (Xilinx, San Jose, CA, USA).

Figure 3 shows the area variation for  $c_3.f_3$  with increasing target precision. It can be seen that the area, which is calculated by AASEE, is less than that by AATRE and AACHA, and the area difference between them is increasing with the target precision. This difference is from 265 to

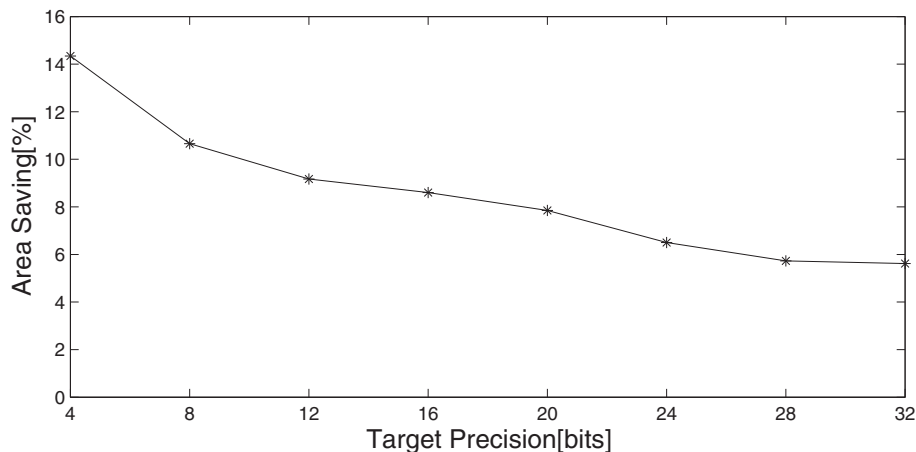
729 with the target precision increased. Such optimization of integer word length can save area.

Figure 4 shows the percentage area saving of AASEE over AATRE at different target precision for  $c_3.f_3$ . The percentage area saving is from 14.34% to 5.62% with the target precision increased. Generally, we obtain increased relative saving for lower precision.

#### 7 Conclusions

This paper presents a novel affine approximation method for multiplication, Approximation Affine based on Space Extreme Estimation. In this method, an extra noise symbol is added to an approximated affine form.

To reduce the uncertainty in AA, we derive this method in the  $(n + 1)$ -dimensional space  $E^{n+1}$ . In space  $E^{n+1}$ , approximate affine form can be regarded as the tangent hyperplane at a certain point of  $(n + 1)$ -dimensional space



**Figure 4** Percentage area saving of AASEE over AATRE at different target precision for  $c_3.f_3$ .

curved surface. Using the linear geometry, it is proven that the  $f_z$  of AASEE is the closest to the result of multiplication among all the possible approximate affine forms. Taking  $\varepsilon_i$  as the input arguments, all the same noise symbols of different variables are taken into account together. Hence, the uncertainty of  $\hat{d}$  of AASEE is reduced. Based on the extreme value theory of multivariable functions, we can prove that the range of this  $\hat{d}$  covers the true range of the difference introduced by approximation and much tighter than that by AATRE and AACHA.

The uncertainty in AASEE is much smaller than that in AATRE and AACHA on average. At the same time, the computational complexity of AASEE is the same as that of AATRE and lower than that of AACHA.

In the case studies, the accuracy of AASEE is 1.69 times than that of AATRE and 1.12 times than that of AACHA on average. The integer word length, which is derived by AASEE, is 2 b less than that by AATRE and 1 b less than that by AACHA, at most. For the case of  $c_3f_3$ , the area, which is computed by AASEE, is less than that by AATRE and AACHA, and the percentage area saving of AASEE over AATRE is from 14.34% to 5.62% with the target precision increased.

#### Competing interests

The authors declare that they have no competing interests.

Received: 8 June 2013 Accepted: 10 March 2014  
Published: 22 March 2014

#### References

1. G Constantinides, G Woeginger, The complexity of multiple wordlength assignment. *Appl. Math. Lett.* **15**(2), 137–140 (2002)
2. R Cmar, L Rijnders, P Schaumont, S Vernalde, I Bolsens, A methodology and design environment for DSP ASIC fixed point refinement, in *Proceedings of Design, Automation and Test in Europe* (IEEE Computer Society, Munich, 09–12 March 1999), pp. 271–276
3. K Kum, W Sung, Combined word-length optimization and high level synthesis of digital signal processing systems. *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.* **20**(8), 921–930 (2001)
4. S Roy, P Banerjee, An algorithm for trading off quantization error with hardware resources for MATLAB-based FPGA design. *IEEE Trans. Comput.* **54**(7), 886–896 (2005)
5. A Mallik, D Sinha, H Zhou, Low-power optimization by smart bit-width allocation in a SystemC-based ASIC design environment. *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.* **26**(3), 447–455 (2007)
6. G Caffarena, C Carreras, JA Lopez, SQNR estimation of fixed-point DSP algorithms. *Eurasip J. Adv. Signal Process.* **21**, 1–12 (2010)
7. A Banciu, E Casseau, D Menard, Stochastic modeling for floating-point to fixed-point conversion, in *Proceedings of IEEE Workshop on Signal Processing Systems (SiPS)* (IEEE Computer Society, Beirut, 4–7 October 2011), pp. 180–185
8. CF Fang, R Rutenbar, M Puschel, T Chen, Toward efficient static analysis of finite-precision effects in DSP applications via affine arithmetic modeling, in *Proceedings of Design Automation Conference, Institute of Electrical and Electronics Engineers Inc.* (Anaheim, 2–6 June 2003), pp. 496–501
9. CF Fang, R Rutenbar, Fast, accurate static analysis for fixed-point finite-precision effects in DSP designs, in *Proceedings of International Conference on Computer-Aided Design, Institute of Electrical and Electronics Engineers Inc.* (San Jose, 9–13 November 2003), pp. 275–282
10. Y Pu, Y Ha, An automated, efficient and static bit-width optimization methodology towards maximum bit-width-to-error tradeoff with affine arithmetic model, in *Proceedings of Asia and South Pacific Design Automation Conference, Institute of Electrical and Electronics Engineers Inc.* (Yokohama, 24–27 January 2006), pp. 886–891
11. DU Lee, AA Gaffar, RC Cheung, O Mencer, W Luk, GA Constantinides, Accuracy guaranteed bit-width optimisation. *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.* **25**(10), 1990–2000 (2006)
12. WG Osborne, JGF Coutinho, W Luk, O Mencer, Instrumented multi-stage word-length optimization, in *Proceedings of IEEE International Conference on Field-Programmable Technology, Institute of Electrical and Electronics Engineers Inc.* (Kitakyushu, 12–14 December 2007), pp. 89–96
13. JA Lopez, C Carreras, O Nieto-Taladriz, Improved interval-based characterization of fixed-point LTI systems with feedback loops. *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.* **2**(11), 1923–1933 (2007)
14. L Zhang, Y Zhang, W Zhou, Tradeoff between approximation accuracy and complexity for range analysis using affine arithmetic. *J. Signal Process. Syst.* **61**(3), 279–291 (2010)
15. O Sarbishei, K Radecka, Z Zilic, Analytical optimization of bit-widths in fixed-point LTI systems. *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.* **31**(3), 343–355 (2012)
16. R Rocher, D Menard, P Scalart, Analytical approach for numerical accuracy estimation of fixed-point systems based on smooth operations. *IEEE Trans. Circuits Syst. I, Reg. Papers.* **59**(10), 2326–2339 (2012)
17. AB Kinsman, N Nicolici, Bit-width allocation for hardware accelerators for scientific computing using SAT-modulo theory. *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.* **29**(3), 406–413 (2010)
18. AB Kinsman, N Nicolici, Computational vector-magnitude-based range determination for scientific abstract data types. *IEEE Trans. Comput.* **60**(11), 1652–1663 (2011)
19. SA Wadekar, AC Parker, Accuracy sensitive word-length selection for algorithm optimization, in *Proceedings of the International Conference on Computer Design: VLSI in Computers and Processors, 1998. ICCD '98, Institute of Electrical and Electronics Engineers Inc.* (Austin, 5–7 October 1998), pp. 54–61
20. C Carreras, JA Lopez, O Nieto-Taladriz, Bit-width selection for data-path implementations, in *Proceedings of the 12th International Symposium on System Synthesis, 1999* (IEEE Computer Society, Boca Raton, 1–4 November 1999), pp. 114–119
21. JLD Comba, J Stolfi, Affine arithmetic and its applications to computer graphics, in *Proceedings of SIBGRAP'93 - VI Simposio Brasileiro de Computacao Grafica e Processamento de Imagens* (IEEE Computer Society, Recife, 20–22 October 1993), pp. 9–18
22. J Stolfi, LH de Figueiredo (eds), Affine arithmetic, in *Self-Validated Numerical Methods and Applications* (Monograph for 21st Brazilian Mathematics Colloquium, IMPA, Rio de Janeiro, Brazil, 1997), pp. 70–74
23. K Huang, H Yee, Improved tangent hyperplane method for transient stability studies [of power systems], in *Proceedings of APSCOM-91 Conference, Institution of Electrical Engineers* (Hong Kong, 5–8 November 1991), pp. 363–366
24. E Eivind, TS Gustavsen, *GRA6035 Mathematics* (BI Norwegian Business School, Oslo, 2010)
25. R Moore, *Interval Analysis* (Prentice-Hall, New Jersey, 1966)
26. Y Pang, K Radecka, An efficient algorithm of performing range analysis for fixed-point arithmetic circuits based on SAT checking, in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)* (IEEE Computer Society, Rio de Janeiro, 15–18 May 2011), pp. 1736–1739
27. N Shekhar, P Kalla, F Enescu, Equivalence verification of arithmetic datapaths with multiple word-length operands, in *Proceedings of Design, Automation and Test in Europe* (IEEE Computer Society, Munich, 6–10 March 2006), pp. 824–829
28. S Gopalakrishnan, P Kalla, MB Meredith, F Enescu, Finding linear building-blocks for RTL synthesis of polynomial datapaths with fixed-size bit-vectors, in *Proceedings of International Conference on Computer-Aided Design, Institute of Electrical and Electronics Engineers Inc.* (San Jose, 5–8 November 2007), pp. 143–148
29. H Shou, W Song, J Shen, R Martind, G Wang, A recursive Taylor method for ray-casting algebraic surfaces, in *Proceedings of International*

*Conference on Computer Graphics and Virtual Reality* (CSREA Press, Las Vegas, 26–29 June 2006), pp. 196–204

30. J Jiang, W Luk, D Rueckert, FPGA-based computation of free-form deformations in medical image registration, in *Proceedings of IEEE International Conference on Field-Programmable Technology 2003* (IEEE Computer Society, Tokyo, 15–17 December 2003), pp. 234–241

doi:10.1186/1687-6180-2014-36

**Cite this article as:** Sun *et al.*: A refined affine approximation method of multiplication for range analysis in word-length optimization. *EURASIP Journal on Advances in Signal Processing* 2014 **2014**:36.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Immediate publication on acceptance
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](http://springeropen.com)

---