A Modified Nested Sparse Grid Based Adaptive Stochastic Collocation Method for Statistical Static Timing Analysis

Xu LUO†, Student Member, Fan YANG†, Nonmember, Xuan ZENG†, Student Member, Jun TAO†, Nonmember, Hengliang ZHU†, Student Member, and Wei CAI†, Nonmember

SUMMARY In this paper, we propose a Modified nested sparse grid based Adaptive Stochastic Collocation Method (MASCSEM) for block-based Statistical Static Timing Analysis (SSTA). The proposed MASCSEM employs an improved adaptive strategy derived from the existing Adaptive Stochastic Collocation Method (ASCM) to approximate the key operator MAX during timing analysis. In contrast to ASCM which uses non-nested sparse grid and tensor product quadratures to approximate the MAX operator for weakly and strongly nonlinear conditions respectively, MASCSEM proposes a modified nested sparse grid quadrature to approximate the MAX operator for both weakly and strongly nonlinear conditions. In the modified nested sparse grid quadrature, we firstly construct the second order quadrature points based on extended Gauss-Hermite quadrature and nested sparse grid technique, and then discard those quadrature points that do not contribute significantly to the computation accuracy to enhance the efficiency of the MAX approximation. Compared with the non-nested sparse grid quadrature, the proposed modified nested sparse grid quadrature not only employs much fewer collocation points, but also offers much higher accuracy. Compared with the tensor product quadrature, the modified nested sparse grid quadrature greatly reduced the computational cost, while still maintains sufficient accuracy for the MAX operator approximation. As a result, the proposed MASCSEM provides comparable accuracy while remarkably reduces the computational cost compared with ASCM. The numerical results show that with comparable accuracy MASCSEM has 50% reduction in run time compared with ASCM.

key words: process variations, statistical static timing analysis, extended Gauss-Hermite quadrature, nested sparse grid, stochastic collocation method

1. Introduction

As manufacturing technology enters into nanometer era, process variations, especially within-die variations, render the delay of the circuit unpredictable, making sign-off ineffective in assuring against chip failure [1]. Traditional Deterministic Static Timing Analysis (DSTA), which computes the circuit delay for a specific process condition, is unable to cope with a large process corner space [1]. An alternative approach that overcomes these problems is Statistical Static Timing Analysis (SSTA) [2]–[9], which treats delays of gates and interconnects not as fixed numbers, but as probability density functions.

SSTA methods can be categorized into path-based approaches [2], [3] and block-based approaches [4]–[9]. Path-based SSTA methods require the enumeration of paths and suffer from the exponential number of paths with respect to the circuit size. Therefore, such methods are not realistic for practical applications. Block-based SSTA methods propagate the random signal arrival times using SUM and MAX operators similarly to propagating the deterministic arrival times by a DSTA. Since block-based SSTA methods have linear run-times with respect to the circuit size, they are more promising than path-based SSTA methods [10].

In order to propagate the random arrival times through the timing graph, the statistical operators SUM and MAX have to be approximated during the block-based SSTA [1]. SUM is a linear operator and not difficult to be approximated, while MAX is nonlinear and the key operator of the block-based SSTA [1]. To approximate the MAX operator, two kinds of approaches have been proposed, i.e., moment match based methods [4], [6] and stochastic collocation methods [7]–[9]. Most of the available moment match based methods adopt the first two order moments to approximate the statistical MAX operator. However, the first two order moments are not accurate enough to capture the nonlinear effect of MAX operator [8], [9]. In contrary to moment match based methods, stochastic collocation methods [7]–[9] approximate the MAX operator in the full random space by minimizing the mean square error, and offer much higher accuracy. In stochastic collocation methods [7]–[9], the random delays of gates and interconnects are modeled by quadratic polynomial chaos expansions. The MAX operator is approximated by a quadratic polynomial chaos expansion, and the unknown coefficients of polynomial chaos expansion are obtained by solving a set of multidimensional integrals over the full random space. In order to reduce the number of collocation points, a dimension reduction technique is adopted in [7] before numerically solving the multidimensional integrals. However, the accuracy of the method in [7] is limited by the dimension reduction technique, and the improvement on efficiency is also limited due to the extra time needed by several times of dimension reduction within one MAX approximation [8], [9].

In [8], [9], an Adaptive Stochastic Collocation Method (ASCM) is proposed for MAX operator approximation, which achieves significant improvements in accuracy and efficiency compared with [7]. The input conditions of the MAX operator in ASCM consist of linear, weakly and
strongly nonlinear conditions. For the linear condition, ASCM simply takes the input with the larger mean as the output of MAX operator. For the weakly nonlinear condition, ASCM employs the Gauss-Hermite quadrature based non-nested sparse grid quadrature to compute the unknown coefficients of polynomial chaos expansion. When the strongly nonlinear condition occurs, the non-nested sparse grid quadrature is not accurate enough for approximating the MAX operation. Therefore the tensor product quadrature is employed to deal with strongly nonlinear conditions. However, the number of collocation points required by the tensor product quadrature increases exponentially with respect to the number of random variables, which makes the tensor product quadrature prohibitive for practical applications.

In this paper, we aim to propose a modified nested sparse grid quadrature method which can handle both the weakly and strongly nonlinear conditions in the MAX operator with good enough accuracy and much cheap computational cost. The modified nested sparse grid quadrature can be derived from the one-dimensional extended Gauss-Hermite quadrature [11] and nested sparse grid technique [12]. The extended Gauss-Hermite quadrature has the nested property [11], i.e., the set of the extended Gauss-Hermite quadrature nodes with lower precision is contained in the one with higher precision. The one-dimensional extended Gauss-Hermite quadrature is extended to multi-dimensional case by nested sparse grid technique. Compared with the conventional Gauss-Hermite quadrature based non-nested sparse grid quadrature in [8],[9], the extended Gauss-Hermite quadrature nodes [11] based nested sparse grid quadrature [12] can achieve much higher computation accuracy due to the nested property. However, the number of the quadrature points employed by the nested sparse grid quadrature is slightly larger than the non-nested one.

Note that the unknown coefficient of the polynomial chaos expansion of MAX operator can be calculated by an integration in Eq. (3). The integrand in Eq. (3) is equal to the product of the MAX operator and a Hermite polynomial with the order at most 2. When the input condition of the MAX operator is weakly or strongly nonlinear, the MAX operator can be numerically approximated by Hermite polynomial of order 2 or at least 3. Therefore the integrand in Eq. (3) will have polynomial order of at least 5 if the strong nonlinear condition is considered. As a result, the integrand in Eq. (3) can be numerically integrated with acceptable accuracy by a quadrature with the polynomial exactness of degree 5. The polynomial exactness of degree 5 means that the quadrature is exact on the polynomial space containing all the polynomials with the order at most 5.

In the nested sparse grid quadrature [12], the one-dimensional extended Gauss-Hermite quadratures with polynomial exactness 1, 5 and 15 are employed to extend the one-dimensional quadrature to multidimensional case. Based on the above observation that the polynomial exactness of degree 5 is enough for the integration in Eq. (3), we propose the modified nested sparse grid quadrature by replacing the one-dimensional quadratures with polynomial exactness 15 with the lower order quadratures with polynomial exactness 5. Consequently, the collocation points with high polynomial exactness are discarded in the modified nested sparse grid quadrature, and the number of collocation points as well as the computational cost are significantly reduced, while the computation accuracy for the MAX operator approximation is still guaranteed even for the strongly nonlinear condition. Compared with the existing non-nested sparse grid quadrature for weakly nonlinear condition in ASCM [8],[9], the proposed modified nested sparse grid quadrature not only employs much fewer collocation points, but also offers much higher accuracy. Moreover, compared with tensor product quadrature for the strongly nonlinear condition in ASCM [8],[9], the proposed modified nested sparse grid quadrature greatly reduces the computational cost, while still maintains sufficient accuracy for the MAX operator approximation.

Finally, we propose a Modified nested sparse grid based Adaptive Stochastic Collocation Method (MASCM) for block-based SSTA. The proposed MASCM inherits the adaptive strategy from ASCM to approximate the MAX operator during timing analysis. In contrast to ASCM which uses the non-nested sparse grid and tensor product quadratures to approximate the MAX operator for weakly and strongly nonlinear conditions respectively, MASCM makes use of the proposed modified nested sparse grid quadrature to approximate the MAX operator for both weakly and strongly nonlinear conditions. The proposed MASCM achieves comparable accuracy while remarkably reduces the computational cost compared with ASCM.

The rest of this paper is organized as follows. The background of the problem formulation of MAX operator approximation and the existing Adaptive Stochastic Collocation Method are presented in Sect. 2. The Modified nested sparse grid based Adaptive Stochastic Collocation Method (MASCM) is proposed in Sect. 3. Experiment results are demonstrated in Sect. 4. In Sect. 5, we conclude the paper.

2. Background

We first briefly review the problem formulation of MAX operator approximation in Sect. 2.1. Then we give the basic concepts of numerical quadrature in Sect. 2.2, and finally introduce the existing Adaptive Stochastic Collocation Method (ASCM) for SSTA in Sect. 2.3.

2.1 Problem Formulation for MAX Operator Approximation

The process variations, e.g., variations of the lengths and widths of gates, are typically modeled as correlated Gaussian random variables [13]. By using the Principal Component Analysis (PCA) technique [13], the correlated Gaussian random variables can be expressed as a linear combination of independent and identically distributed normal ran-
dom variables \(\mathbf{\bar{\xi}} = \{\xi_1, \xi_2, \cdots\}\). Both gate and interconnect delays can be approximated by Hermite polynomial chaos expansion as follows [8], [9],

\[
A(\mathbf{\bar{\xi}}) = \sum_{i_1, i_2}^{2} a_{d_{i_1}, i_2} H_{d_{i_1}, i_2}(\mathbf{\bar{\xi}}),
\]

where \(d\) is the number of random variables, \(H_{d_{i_1}, i_2}(\mathbf{\bar{\xi}})\) denotes a \(d\)-dimensional Hermite polynomial, the superscript \(i_1, i_2, \cdots, d\) in \(H_{d_{i_1}, i_2}(\mathbf{\bar{\xi}})\) denotes the polynomial order on the \(k\)-th dimension, and \(\{a_{d_{i_1}, i_2}\}\) are the coefficients that can be obtained from the interconnect and gate models [14], [15].

Suppose we have \(N\) arrival times \(A_1(\mathbf{\bar{\xi}}), \cdots, A_N(\mathbf{\bar{\xi}})\) modeled by (1), then the MAX operator among the \(N\) arrival times can be approximated by a quadratic Hermite polynomial expansion as follows [7]–[9],

\[
M(\mathbf{\bar{\xi}}) \triangleq \text{MAX} \left\{A_1(\mathbf{\bar{\xi}}), \cdots, A_N(\mathbf{\bar{\xi}})\right\} \\
\approx C(\mathbf{\bar{\xi}}) \triangleq \sum_{i_1, i_2}^{2} c_{d_{i_1}, i_2} H_{d_{i_1}, i_2}(\mathbf{\bar{\xi}}),
\]

where \(\{c_{d_{i_1}, i_2}\}\) are the unknown coefficients to be determined.

Based on the orthogonality of Hermite polynomials, the unknown coefficients in (2) can be computed by the following equation [7]–[9],

\[
c_{d_{i_1}, i_2} = \int_{\Gamma} M(\mathbf{\bar{\xi}}) H_{d_{i_1}, i_2}(\mathbf{\bar{\xi}}) \rho(\mathbf{\bar{\xi}}) d\mathbf{\bar{\xi}},
\]

where \(\Gamma \in \mathbb{R}^{d}\) and \(\rho(\mathbf{\bar{\xi}}) = (2\pi)^{-d/2} \exp(-\mathbf{\bar{\xi}}^T \mathbf{\bar{\xi}}/2)\) are the probability space and joint Probability Density Function (PDF) of normal random variables \(\mathbf{\bar{\xi}}\), respectively.

The right-hand side of (3) is a multidimensional integral, and can be approximated by a numerical quadrature as follows,

\[
c_{d_{i_1}, i_2} \approx \int_{\mathbf{\bar{\xi}}} f(\mathbf{\bar{\xi}}) \rho(\mathbf{\bar{\xi}}) d\mathbf{\bar{\xi}} \approx Q_l(f) \triangleq \sum_{j=1}^{N(l,d)} f(\mathbf{\bar{\xi}}_j) w_j,
\]

where \(f(\mathbf{\bar{\xi}}) = M(\mathbf{\bar{\xi}}) H_{d_{i_1}, i_2}(\mathbf{\bar{\xi}}), Q_l(f)\) denotes a \(d\)-variable numerical quadrature with the \(l\)-th level precision, \(w_j\) is the quadrature weight at the quadrature node \(\mathbf{\bar{\xi}}_j\), and \(N(l, d)\) is the total amount of quadrature nodes \(\{\mathbf{\bar{\xi}}_j\}\).

### 2.2 Basic Concepts of Numerical Quadrature

The multidimensional quadrature can be constructed from one-dimensional quadratures via tensor product method or sparse grid technique [12]. We first introduce the basic concepts of one-dimensional quadrature in this subsection.

Generally, a one-dimensional quadrature \(Q_l(f)\) with the \(l\)-th level precision can be expressed as follows,

\[
\int_a^b f(\xi) \rho(\xi) d\xi \approx Q_l(f) \triangleq \sum_{j=1}^{N(l,1)} f(\xi_j) w_j,
\]

where \((a, b)\) is the integral interval, \(f(\xi)\) is the integrand, \(\rho(\xi)\) is the weight function, \(w_j\) is the quadrature weight at the quadrature node \(\xi_j\), and \(N(l, 1)\) is the total amount of quadrature nodes \(\{\xi_j\}\).

In order to analyze the quadrature accuracy, we introduce the concept of polynomial space as follows.

**Definition 1** Polynomial space \(\mathbb{P}(e)\) is defined as the space of all the polynomials in one variable of degree at most \(e\).

For example, the polynomial space \(\mathbb{P}(2)\) includes all the linear combination of the polynomials with the order at most 2, i.e., \(\mathbb{P}(2) = \{a_0 + a_1 x + a_2 x^2; a_0, a_1, a_2 \in \mathbb{R}\}\).

**Definition 2** Given that the integrand \(f(\xi)\) in (5) is chosen as any polynomials in the polynomial space \(\mathbb{P}(e)\), if the integral can be exactly calculated by the numerical quadrature in (5), we say this quadrature is exact on the polynomial space \(\mathbb{P}(e)\).

The conventional Gauss-Hermite quadrature is often used to approximate one-dimensional integrals and regarded as the best choice for the Gaussian weight function \(\rho(\xi) = (2\pi)^{-1/2} \exp(-\xi^2/2)\) [16]. The corresponding Gauss-Hermite quadrature nodal set \(\mathbb{V}_l^e (l \in \mathbb{N})\) comprises the zeros of the \(l\)-th order Hermite polynomial [16], namely,

\[
\begin{align*}
\mathbb{V}_1^1 &= \{0\}, \\
\mathbb{V}_2^1 &= \{-1, 1\}, \\
\mathbb{V}_3^1 &= \{-1.732, 0, 1.732\}, \\
&\quad \cdots
\end{align*}
\]

Since an \(n\)-node one-dimensional Gauss-Hermite quadrature is exact on the polynomial space \(\mathbb{P}(2n - 1)\) [16], the Gauss-Hermite quadratures with the nodal sets \(\mathbb{V}_1^1 = \{0\}, \mathbb{V}_2^1 = \{-1, 1\}, \mathbb{V}_3^1 = \{-1.732, 0, 1.732\}, \cdots\), are exact on the polynomial space \(\mathbb{P}(1), \mathbb{P}(3), \mathbb{P}(5), \cdots\), respectively.

### 2.3 Adaptive Stochastic Collocation Method (ASCM)

In [8], [9], an Adaptive Stochastic Collocation Method (ASCM) is proposed to approximate the MAX operator. According to the input conditions of the MAX operator, ASCM divides the MAX operator into three categories: 1) linear condition: the distributions of two inputs do not overlap significantly; 2) strongly nonlinear condition: the distributions of two inputs remarkably overlap with each other; and 3) weakly nonlinear condition: all conditions other than the previous two conditions. For linear condition, ASCM simply takes the input with the largest mean as the output of MAX operator. In such a case, it is no necessary to compute the polynomial approximation in (2) and the multidimensional integral in (4). While for strongly and weakly nonlinear conditions, different quadratures are employed for the
multidimensional integral in (4).

For the strongly nonlinear condition, ASCM utilizes Tensor Product (TP) quadrature to compute the multidimensional integral in (4). The quadrature nodes used by tensor product quadrature with $l$-th level precision is given by the Cartesian product of one-dimensional quadrature nodes as follows \cite{8, 9}

$$n_d^l = \mathcal{V}_1^{n_{1,l+1}} \times \cdots \times \mathcal{V}_{d}^{n_{d,l+1}}, \quad (7)$$

where $\mathcal{V}_i^{n_{i,l+1}}$ given in (6) denotes one-dimensional Gauss-Hermite quadrature nodes with the $(l+1)$-th quadrature precision. The total amount of the quadrature nodes in $\mathcal{V}_d^l$ is \cite{8, 9}

$$N_{TP}(l, d) = (l + 1)^d. \quad (8)$$

Obviously, $N_{TP}(l, d)$ increases exponentially with respect to the dimension $d$, which is prohibitive for practical applications where $d \gg 1$.

For weakly nonlinear condition, ASCM employs Sparse Grid (SG) quadrature to compute the multidimensional integral in (4). Sparse grid quadrature with the $l$-th level precision actually is a special combination of one-dimensional quadratures with different precisions in the following form \cite{8, 9}

$$Q^l_{j}(f) = \sum_{l+1 \leq |\vec{k}| \leq l+d} (-1)^{|\vec{k}|} \prod_{i=1}^{d} \left( d - 1 \right) \left( l + d - |k_i| \right) \left( Q^1_{k_1} \otimes \cdots \otimes Q^1_{k_d} \right) (f), \quad (9)$$

where $\otimes$ denotes a tensor operator, $|\vec{k}| = k_1 + \cdots + k_d$, and $Q^1_{k_i}$ denotes the one-dimensional quadrature using Gauss-Hermite quadrature nodes $\mathcal{V}_i^1$ in (6) along the $i$-th dimension. The nodal set of the quadrature nodes used in (9) in ASCM is given by

$$\mathcal{V}_d^l = \bigcup_{l+1 \leq |\vec{k}| \leq l+d} \mathcal{V}_{k_1}^1 \times \cdots \times \mathcal{V}_{k_d}^1. \quad (10)$$

The total amount of quadrature nodes in $\mathcal{V}_d^l$ is \cite{12}

$$N_{SG}(l, d) = \sum_{l+1 \leq |\vec{k}| \leq l+d} n_1^{k_1} n_2^{k_2} \cdots n_d^{k_d} \approx \frac{n!}{l^d}, \quad d \gg 1, \quad (11)$$

where $n_i^{k_i}$ denotes the number of one-dimensional Gauss-Hermite quadrature nodes in $\mathcal{V}_i^{k_i}$. The number of quadrature nodes shown in (11) avoids the exponential increase with respect to the dimension $d$ of the tensor product quadrature shown in (8). Especially for $l = 2$, Eq. (11) becomes

$$N_{SG}(2, d) = 2d^2 + 2d + 1. \quad (12)$$

by using the facts that $n_1^1 = 1$, $n_2^1 = 2$ and $n_3^1 = 3$.

However, the conventional Gauss-Hermite quadrature nodal sets do not have the nested property given in Definition 3, e.g., $\mathcal{V}_1^1 = \{0\} \not\subseteq \mathcal{V}_2^1 = \{-1, 1\} \not\subseteq \mathcal{V}_3^1 = \{-1.732, 0, 1.732\}$. It means that the quadrature nodes with lower precision, e.g., $\mathcal{V}_2^1 = \{-1, 1\}$, can not be reused in the quadrature nodal set with higher precision, e.g., $\mathcal{V}_3^1 = \{-1.732, 0, 1.732\}$.

The conventional Gauss-Hermite quadrature node based sparse grid quadrature is referred to as non-nested sparse grid quadrature in this paper. It will be shown in the next section that if the quadrature is nested, the quadrature nodes can be reused and the precision of the sparse grid quadrature can be remarkably improved. Although the nested sparse grid quadrature can achieve much higher computation accuracy, the number of quadrature points employed by the nested sparse grid quadrature is slightly larger than the non-nested one.

In the next section, we will propose a modified nested sparse grid quadrature. It provides higher computation accuracy than the non-nested sparse grid quadrature with even lower computational cost. Moreover, compared with the tensor product quadrature, it can achieve comparable accuracy with greatly reduced computational cost. The modified nested sparse grid quadrature can serve as a universal quadrature rule for both weakly and strongly nonlinear conditions, and significantly reduce the computational cost for MAX operator approximation with comparable accuracy in SSTA.

3. Modified Nested Sparse Grid Based Adaptive Stochastic Collocation Method for SSTA

In this section, we focus on the multi-dimensional numerical quadrature in (4), which is the key step of MAX operator approximation in SSTA. In Sect. 3.1, we first propose a nested sparse grid based multi-dimensional quadrature method. In order to further improve the efficiency of nested sparse grids quadrature method, we propose a modified nested sparse grid quadrature method in Sect. 3.2. Finally, the SSTA algorithm based on the Modified nested sparse grid based Adaptive Stochastic Collocation Method (MASCM) is presented in Sect. 3.3.

3.1 Nested Sparse Grids (NSGs) Based Multidimensional Quadrature

The nested sparse grids based multidimensional quadrature \cite{12} is constructed from the one-dimensional quadratures. We choose the one-dimensional extended Gauss-Hermite quadrature \cite{11}, other than the one-dimensional conventional Gauss-Hermite quadrature used in ASCM \cite{8, 9}, to construct the multidimensional nested sparse grid quadrature. The extended Gauss-Hermite quadrature aims to incrementally construct the new quadrature nodes based on the existing quadrature nodes, such that all the quadrature nodes have the highest possible degree of precision. Obviously, the extended Gauss-Hermite quadrature nodes naturally satisfy the nested property given in Definition 3.
**Definition 3** If the set of the quadrature nodes with lower precision is contained in the one with higher precision, we call this kind of quadrature formula and nodal set are nested.

For the Gauss-Hermite integral in (5), several families of extended quadrature nodes are explored by Genz in [11]. To save the number of quadrature nodes, we choose the following nodal sequence,

\[
\mathcal{V}_1^d = \{0\},
\]

\[
\mathcal{V}_2^d = \{0, \pm 1.732\},
\]

\[
\mathcal{V}_3^d = \{0, \pm 1.732, \pm 0.741, \pm 2.86, \pm 4.18\},
\]

\[\vdots\]

Among the families of extended quadrature nodes in [11], this nodal sequence has the slowest increasing rate as quadrature level advances. It can be observed that the nodal sets in (13) satisfy the nested property given in Definition 3, i.e., \(\mathcal{V}_1^d \subseteq \mathcal{V}_2^d \subseteq \mathcal{V}_3^d \subseteq \cdots\). According to [11], the numerical quadratures with the extended Gauss-Hermite quadrature nodes \(\mathcal{V}_1^d, \mathcal{V}_2^d\) and \(\mathcal{V}_3^d\) in (13) are exact on the polynomial spaces \(\mathbb{P}(1), \mathbb{P}(5)\) and \(\mathbb{P}(15)\), respectively. Compared with the conventional Gauss-Hermite quadratures with the nodal sets (6), the extended one with the nodal sets (13) is exact on a much larger polynomial space due to the reuse of quadrature nodes with lower levels.

Nested sparse grid technique [12] can be used to construct the multi-dimensional quadrature nodes based on one-dimensional extended Gauss-Hermite quadrature nodes. This kind of quadrature is referred to as nested sparse grid quadrature in this paper. The nested sparse grid quadrature formula has the same form as non-nested sparse grid quadrature formula in (9), of which the quadrature nodes and weights can be constructed in an alternative way as follows.

The nodal set used by a \(d\)-dimensional nested sparse grid quadrature with the \(l\)-th level precision is given as follows,

\[
\mathcal{V}_l^d = \bigcup_{|\vec{k}|_1 = l+d} \mathcal{W}_k^d \times \cdots \times \mathcal{W}_k^d, \tag{14}
\]

where \(\mathcal{W}_k^d (i = 1, \ldots, d)\) denotes the different set between the two adjacent extended Gauss-Hermite quadrature nodal sets \(\mathcal{V}_1^d, \mathcal{V}_2^d\), i.e.,

\[
\mathcal{W}_k^d = \mathcal{V}_k^d \setminus \mathcal{V}_1^d.
\]

The symbol \(\setminus\) means the difference of sets [12], i.e., \(A \setminus B = \{x : x \in A \text{ and } x \notin B\}\). For instance, \(\mathcal{W}_1^2 = \mathcal{V}_2^2 \setminus \mathcal{V}_1^2 = \{\pm 1.732\}\). Additionally, we define \(\mathcal{W}_1^1 = \mathcal{V}_1^1 = \{0\}\). Denoting the number of the elements in \(\mathcal{W}_k^d\) as \(\tilde{m}_k^d\), the total amount of the \(d\)-dimensional quadrature nodes in \(\mathcal{V}_l^d\) is [12]

\[
N_{NSG}(l, d) = \sum_{|\vec{k}|_1 = l+d} \tilde{m}_1^d \tilde{m}_2^d \cdots \tilde{m}_l^d. \tag{15}
\]

Especially for quadrature level \(l = 2\), the number of the \(d\)-dimensional quadrature nodes in \(\mathcal{V}_2^d\) is

\[
N_{NSG}(2, d) = 2d^2 + 6d + 1, \tag{16}
\]

by using the facts that \(\tilde{m}_1^1 = 1, \tilde{m}_2^1 = 2\) and \(\tilde{m}_3^1 = 6\).

In order to obtain the quadrature weights of nested sparse grid based multidimensional quadrature, we replace the integrand \(f(\vec{\xi})\) in (4) with Hermite polynomials \(H_{d-1}^{1-i_1}(\vec{\xi})\), and the following linear equation system can be derived from (4) by using the orthogonality of Hermite polynomials [16].

\[
\begin{bmatrix}
H_{d}^{0}\ldots0(\vec{\xi}_1) & \cdots & H_{d}^{0}\ldots0(\vec{\xi}_N) \\
H_{d-1}^{1}\ldots1(\vec{\xi}_1) & \cdots & H_{d-1}^{1}\ldots1(\vec{\xi}_N) \\
\vdots & \ddots & \vdots \\
H_{d-1}^{i_1-1}\ldots(i_1-1)(\vec{\xi}_1) & \cdots & H_{d-1}^{i_1-1}\ldots(i_1-1)(\vec{\xi}_N)
\end{bmatrix}
\begin{bmatrix}
w_1^d \\
w_2^d \\
\vdots \\
w_N^d
\end{bmatrix}
= 0 \quad \text{if} \quad i_1 \leq l \tag{17}
\]

In practical implementation, the polynomial of Hermite polynomials should be large enough such that the rank of the above equation is at least \(N(l, d)\). The numerical quadrature weights required in (4) can be obtained by solving this linear equation system (17).

The quadrature accuracy of the nested sparse grid quadrature obeys the following theorem.

**Theorem 1** Given that the extended Gauss-Hermite quadrature with the nodal set \(\mathcal{V}_1^d\) is exact on the polynomial space \(\mathbb{P}(\vec{k})\), then the nested sparse grid quadrature with the nodal set \(\mathcal{V}_d^l\) in (14) is exact on the following multidimensional polynomial space \(\mathbb{P}_d^l\) [17],

\[
\mathbb{P}_d^l = \sum_{|\vec{k}|_1 = l+d} \mathbb{P}(k_1) \otimes \cdots \otimes \mathbb{P}(k_d), \tag{18}
\]

where \(k_i = 1, 2, \ldots, i = 1, \ldots, d\).

For the non-nested sparse grids based multidimensional quadrature, we also have a similar theorem as follows.

**Theorem 2** Given that the conventional Gauss-Hermite quadrature with the nodal set \(\mathcal{V}_1^d\) is exact on the polynomial space \(\mathbb{P}(\vec{k})\), then the non-nested sparse grid quadrature with the nodal set \(\mathcal{V}_d^l\) in (10) is exact on the following multidimensional polynomial space \(\mathbb{P}_d^l\) [17],

\[
\mathbb{P}_d^l = \sum_{|\vec{k}|_1 = l+d} \mathbb{P}(k_1) \otimes \cdots \otimes \mathbb{P}(k_d), \tag{19}
\]

where \(k_i = 1, 2, \ldots, i = 1, \ldots, d\).

Typically, the nested or non-nested sparse grid based multidimensional quadrature with the quadrature level \(l = 2\) is employed for the MAX operator approximation [8],[9]. Based on Theorems 1 and 2, we have the following two corollaries.

**Corollary 1** For the quadrature level \(l = 2\), the nested
sparse grid quadrature with the nodal set \( \tilde{\mathcal{V}}^2_d \) in (14) is exact on

\[
P^2_d = \sum_{|\tilde{k}| = 2^d} \mathcal{P}(\tilde{e}^{\tilde{k}}) \otimes \cdots \otimes \mathcal{P}(\tilde{e}^{\tilde{k}}),
\]

where \( k_i = 1, 2, 3, i = 1, \ldots, d, \mathcal{P}(e^1) = \mathcal{P}(1), \mathcal{P}(e^2) = \mathcal{P}(5) \) and \( \mathcal{P}(e^3) = \mathcal{P}(15) \).

**Corollary 2** For the quadrature level \( l = 2 \), the non-nested sparse grid quadrature with the nodal set \( \mathcal{V}^2_d \) in (10) is exact on

\[
P^2_d = \sum_{|\tilde{k}| = 2^d} \mathcal{P}(e^{\tilde{k}}) \otimes \cdots \otimes \mathcal{P}(e^{\tilde{k}}),
\]

where \( k_i = 1, 2, 3, i = 1, \ldots, d, \mathcal{P}(e^1) = \mathcal{P}(1), \mathcal{P}(e^2) = \mathcal{P}(5) \) and \( \mathcal{P}(e^3) = \mathcal{P}(15) \).

Generally, if a quadrature is exact on a larger polynomial space, the quadrature precision will be higher [17]. It can be observed from Corollaries 1 and 2 that the nested sparse grid quadrature with the nodal set \( \mathcal{V}^2_d \) is exact on a much larger polynomial space than the non-nested sparse grid quadrature with the nodal set \( \mathcal{V}^2_d \), i.e., \( P^2_d \) is much larger than \( \tilde{P}^2_d \). Taking \( d = 2 \) as an example, the polynomial space on which the nested sparse grid quadrature is exact, i.e., \( P^2_2 = \mathcal{P}(1) \otimes \mathcal{P}(15) + \mathcal{P}(15) \otimes \mathcal{P}(1) + \mathcal{P}(5) \otimes \mathcal{P}(5) \), is much larger than that on which the non-nested sparse grid quadrature is exact, i.e., \( \tilde{P}^2_2 = \mathcal{P}(1) \otimes \mathcal{P}(5) + \mathcal{P}(5) \otimes \mathcal{P}(1) + \mathcal{P}(3) \otimes \mathcal{P}(3) \). As a result, the nested sparse grid quadrature has higher quadrature precision than the non-nested one.

Although the nested sparse grid quadrature offers much higher precision than the non-nested one, the number of the nested sparse grids increases a little bit faster than that of the non-nested ones. From (12) and (16), we can see that the number of the nested sparse grids with the quadrature level \( l = 2 \) is larger than that of the non-nested one, i.e., \( N_{NSG}(2, d) = 2d^2 + 6d + 1 > N_{SG}(2, d) = 2d^2 + 2d + 1 \). Furthermore, the nested sparse grid quadrature is exact for polynomial space containing some polynomials with the order up to 15, which is much higher for MAX operator approximation. It is possible to remove some quadrature points with remarkably high polynomial exactness while still maintaining adequate computation accuracy for MAX approximation, as will be discussed in the following subsection.

3.2 Modified Nested Sparse Grids (MNSGs) Based Multi-dimensional Quadrature

We know from Sect. 1 that the integrand in Eq. (3) only has polynomial order of at least 5 even the strongly nonlinear condition is considered. However, the polynomial space \( \mathcal{P}(15) \) in (20) contains all the polynomials with the order at most 15. Therefore, we can discard some quadrature points with remarkably high polynomial exactness, while the computation accuracy for the MAX operator approximation can still be guaranteed even for the strongly nonlinear condition. In the multidimensional nested sparse grid quadrature, we replace the corresponding one-dimensional quadrature that is exact on the polynomial space \( \mathcal{P}(15) \) with a quadrature that is exact on the polynomial space \( \mathcal{P}(5) \). We find that the accuracy of MAX operator approximation can be still guaranteed even for strongly nonlinear conditions. However, the number of quadrature points as well as the computational cost are remarkably reduced. The new family of one-dimensional quadrature nodes \( \mathcal{V}^1_d \) can be expressed as follows,

\[
\hat{\mathcal{V}}^1_1 = \{0\},
\]

\[
\hat{\mathcal{V}}^1_2 = \{0, \pm 1.732\},
\]

\[
\hat{\mathcal{V}}^1_3 = \{0, \pm 1.732\},
\]

\[
\cdots
\]

Compared with the original extended Gauss-Hermite nodal sequence (13), the new nodal sequence (22) replaces the third nodal set \( \mathcal{V}^1_1 \) with the nodal set \( \mathcal{V}^2_1 \), such that the quadrature that is exact on the polynomial space \( \mathcal{P}(15) \) is replaced with the quadrature that is exact on the polynomial space \( \mathcal{P}(5) \). The number of collocation points can be reduced since the new nodal set \( \mathcal{V}^3_1 \) has much fewer elements than the original nodal set \( \mathcal{V}^3_1 \).

The sparse grids based on the new family of one-dimensional quadrature nodes in (22) is named as Modified Nested Sparse Grids (MNSGs). The nodal set of MNSGs can be expressed in a similar form of (14) as follows,

\[
\hat{\mathcal{V}}^1_d = \bigcup_{|\tilde{k}| \leq 2^d} \hat{\mathcal{V}}^k_1 \times \cdots \times \hat{\mathcal{V}}^k_1,
\]

where \( \hat{\mathcal{V}}^k_1 = \mathcal{V}^k_1 \setminus \mathcal{V}^{k-1}_1 \). Similarly, denoting the number of the elements in \( \hat{\mathcal{V}}^k_1 \) as \( \tilde{m}^k_1 \), the total amount of quadrature nodes in \( \hat{\mathcal{V}}^1_d \) is

\[
N_{MNSG}(l, d) = \sum_{|\tilde{k}| \leq 2^d} \tilde{m}^1_1 \tilde{m}^2_1 \cdots \tilde{m}^k_1.
\]

Especially for the quadrature level \( l = 2 \), the number of the quadrature nodes in the nodal set \( \hat{\mathcal{V}}^2_d \) is

\[
N_{MNSG}(2, d) = 2^d + 1.
\]

by using the facts that \( \tilde{m}^1_1 = 1, \tilde{m}^2_1 = 2 \) and \( \tilde{m}^3_1 = 0 \). According to Theorem 1, for the modified nested sparse grid quadrature, we have the following corollary.

**Corollary 3** For the quadrature level \( l = 2 \), the modified nested sparse grid quadrature with the nodal set \( \hat{\mathcal{V}}^2_d \) in (23) is exact on

\[
\hat{P}^2_d = \sum_{|\tilde{k}| = 2^d} \mathcal{P}(e^{\tilde{k}}) \otimes \cdots \otimes \mathcal{P}(e^{\tilde{k}}),
\]

where \( k_i = 1, 2, 3, i = 1, \ldots, d, \mathcal{P}(e^1) = \mathcal{P}(5) \) and \( \mathcal{P}(e^2) = \mathcal{P}(5) \) and \( \mathcal{P}(e^3) = \mathcal{P}(15) \).
and \( P(\hat{e}^3) = P(5) \).

Compared with the original nested sparse grid quadrature, the modified nested sparse grid quadrature employs much fewer quadrature nodes, while still maintains comparable accuracy. It can be observed from (16) and (25) that the modified nested sparse grid quadrature with the nodal set \( V_d^3 \) employs fewer quadrature nodes than the original nested sparse grid quadrature with the nodal set \( V_d^2 \), i.e., \( N_{NSG}(2, d) = 2d^2 + 1 < N_{NSG}(3, d) = 2d^2 + 6d + 1 \). According to Corollaries 2 and 3, the polynomial space \( P_d^3 \) on which the modified nested sparse grid quadrature is exact is smaller than the polynomial space \( P_d^2 \) on which the original nested sparse grid quadrature is exact. However, from numerical experiments, we find that the polynomial space \( P_d^2 \), which contains all the \( d \)-variable polynomials with the order at most 5, is sufficiently large for the calculation of numerical quadrature in (4) even under strongly nonlinear conditions. The modified nested sparse grid quadrature has comparable accuracy with the original nested sparse grid quadrature.

Compared with the non-nested sparse grid quadrature used in ASCM [8], [9], the modified nested sparse grid quadrature not only has much higher accuracy, but also employs much fewer quadrature nodes. It can be found from Corollaries 1 and 3 that the modified nested sparse grid quadrature with the nodal set \( V_d^2 \) offers a much larger polynomial space than the non-nested one with the nodal set \( V_d^3 \), i.e., \( P_d^2 \) is much larger than \( P_d^3 \). Taking \( d = 2 \) as an example, the polynomial space on which the modified nested sparse grid quadrature is exact, i.e., \( P_2^2 = P(1) \otimes P(5) + P(5) \otimes P(1) + P(5) \otimes P(5) \), is much larger than that on which the non-nested sparse grid quadrature is exact, i.e., \( P^3_2 = P(1) \otimes P(5) + P(5) \otimes P(3) + P(3) \otimes P(3) \). As a result, the accuracy of the modified nested sparse grid quadrature is higher than that of the non-nested sparse grid quadrature. Moreover, it can be observed from (12) and (25) that the amount of the efficient nested grids in \( V_d^2 \) is smaller than that of the non-nested sparse grids in \( V_d^3 \), i.e., \( N_{NSG}(2, d) = 2d^2 + 1 < N_{NSG}(3, d) = 2d^2 + 2d + 1 \).

Compared with tensor product quadrature used in ASCM [8], [9], the modified nested sparse grid quadrature employs much fewer quadrature nodes, while still maintains sufficient accuracy for the MAX operator approximation, as will be shown in Sect. 4.

### 3.3 SSTA Based on the Proposed MASCM

The SSTA method based on the proposed modified nested sparse grid based MASCM is summarized in Algorithm 1. Algorithm 1 employs an adaptive strategy derived from ASCM to approximate the MAX operator. Algorithm 1 divides the inputs of the MAX operator into linear and nonlinear conditions, while ASCM further divides the nonlinear condition into weakly and strongly nonlinear conditions. For the linear condition, Algorithm 1 uses the same approximation scheme as ASCM. In contrast to ASCM which uses the non-nested sparse grid and tensor product quadratures to approximate the MAX operator for weakly and strongly nonlinear conditions respectively, MASCM employs the proposed modified nested sparse grid quadrature to approximate the MAX operator for both weakly and strongly nonlinear conditions.

According to the theoretical analysis in Sect. 3.2, the modified nested sparse grid quadrature employs much fewer collocation points while offers much higher accuracy than the non-nested sparse grid quadrature. Moreover, compared with tensor product quadrature, it can achieve comparable accuracy with greatly reduced computational cost for the MAX operator approximation. Therefore, the SSTA based on MASCM provides comparable accuracy while remarkably reduces the computational cost compared with ASCM.

### 4. Experiment Results

We have implemented the proposed MASCM for block-based SSTA in C++. We first use the MAX operator approximation experiment taken from ASCM [8], [9] to compare the efficiency and accuracy of the proposed modified nested sparse grid quadrature with tensor product quadrature, non-nested sparse grid quadrature and the original nested sparse grid quadrature. Then we test the proposed MASCM on ISCAS85 benchmark circuits [18], and compare the results with ASCM [8], [9]. All the experiments are run on a 3 GHz workstation with 8 GB RAM.

#### 4.1 Accuracy and Efficiency Comparisons of Collocation Methods for MAX Operator Approximation

ASCM [8], [9] divides the MAX operator into: linear condition, strongly nonlinear condition and weakly nonlinear condition. For linear condition, ASCM simply takes the input with the largest mean as the output of MAX operator. While for strongly and weakly nonlinear conditions, tensor product quadrature and non-nested sparse grid quadrature
are employed for the multidimensional integral in (4), respectively. In order to verify the accuracy and efficiency of the proposed nested sparse grid quadrature under the weakly and strongly input conditions, we consider the approximation of two input MAX operator \( MAX(A(\vec{\xi}), B(\vec{\xi})) \). Both the two inputs \( A(\vec{\xi}) \) and \( B(\vec{\xi}) \) are modeled as quadratic Hermite polynomial chaos expressions.

At the beginning, the mean of \( A(\vec{\xi}) \) is set as a small value compared with the mean of \( B(\vec{\xi}) \). Then the mean of \( A(\vec{\xi}) \) is swept to become bigger and bigger with a small step length each time. As the mean of \( A(\vec{\xi}) \) is sweeping, the five conditions: linear condition, weakly nonlinear condition, strongly nonlinear condition and linear condition, will appear one by one according to the classification of input conditions given in Sect. 2.3. For the weakly and strongly nonlinear conditions, the MAX operator is approximated by Monte Carlo method and Stochastic Collocation Methods (SCMs) with different quadrature rules: tensor product quadrature [8], [9], non-nested sparse grid quadrature [8], [9], the original nested sparse grid quadrature and the modified nested sparse grid quadrature.

The experiment results of MAX operator approximation with 10 random variables are given in Fig. 1. The y-axis represents the relative mean and standard deviation errors compared with 10,000 Monte Carlo simulations, and x-axis represents step number. It can be observed from Fig. 1 that the tensor product quadrature based SCM has the highest accuracy while the non-nested sparse grid quadrature based SCM has the worst accuracy. Both the original and modified nested sparse grid quadrature based SCMs have much higher accuracy than the non-nested sparse grid quadrature based SCM. They also provide comparable accuracy as the tensor product quadrature based SCM. Compared with the original nested sparse grid quadrature base SCM, the proposed modified nested sparse grid quadrature based SCM offers almost the same accuracy. After discarding the quadrature points with remarkably high polynomial exactness, the computation accuracy of the modified nested sparse grid is still guaranteed for the MAX approximation.

Table 1 and Fig. 2 give the number of collocation points and run time of five different methods for one time approximation of a MAX operator under the strongly nonlinear condition with different number of random variables, respectively. Among the non-nested, nested and modified nested sparse grid quadrature based SCMs, the proposed modified nested sparse grid quadrature based SCM has the least number of collocation points and run time. For the tensor prod-

---

**Table 1** Comparison of the number of collocation points with different dimension (d).

<table>
<thead>
<tr>
<th>Collocation methods</th>
<th>d = 3</th>
<th>d = 4</th>
<th>d = 5</th>
<th>d = 6</th>
<th>d = 7</th>
<th>d = 8</th>
<th>d = 9</th>
<th>d = 10</th>
<th>d = 11</th>
<th>d = 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monte Carlo method</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
<td>10000</td>
</tr>
<tr>
<td>Tensor product quadrature based SCM</td>
<td>21</td>
<td>81</td>
<td>243</td>
<td>729</td>
<td>2187</td>
<td>6561</td>
<td>19683</td>
<td>59049</td>
<td>177147</td>
<td>531441</td>
</tr>
<tr>
<td>Non-nested sparse grid quadrature based SCM</td>
<td>25</td>
<td>41</td>
<td>61</td>
<td>85</td>
<td>113</td>
<td>145</td>
<td>181</td>
<td>221</td>
<td>265</td>
<td>313</td>
</tr>
<tr>
<td>Nested sparse grid quadrature based SCM</td>
<td>37</td>
<td>57</td>
<td>81</td>
<td>109</td>
<td>141</td>
<td>177</td>
<td>217</td>
<td>261</td>
<td>309</td>
<td>361</td>
</tr>
<tr>
<td>Modified nested sparse grid quadrature based SCM</td>
<td>19</td>
<td>33</td>
<td>51</td>
<td>73</td>
<td>99</td>
<td>129</td>
<td>163</td>
<td>201</td>
<td>243</td>
<td>289</td>
</tr>
</tbody>
</table>

---

**Fig. 1** Comparison of MAX operator approximation.

**Fig. 2** Comparison of run time of different collocation methods.
uct quadrature based SCM, the number of collocation points and computation time increase exponentially with respect to the number of random variables, and are larger than Monte Carlo method with 10,000 sampling points when the number of random variables is larger than 8. Therefore, the high computational cost of tensor product quadrature prevent it from practical applications.

For the strongly nonlinear condition, compared with the tensor product quadrature, the proposed modified nested sparse grid quadrature greatly reduces the number of collocation points and computation time, while still maintains sufficient accuracy. For the weakly nonlinear condition, the proposed modified nested sparse grid quadrature not only has higher accuracy, but also takes less number of collocation points and run time than the non-nested sparse grid quadrature. We can use the proposed modified nested sparse grid quadrature as the universal quadrature for both weakly and strongly nonlinear input conditions.

4.2 Comparison between MASCM and ASCM on ISCAS85 Benchmark Circuits

In this subsection, ISCAS85 benchmark circuits [18] with the 65 nm technology in Predictive Technology Model [19] are used to compare the proposed Modified nested sparse grid based Adaptive Stochastic Collocation Method (MASCM) with the existing ASCM [8], [9].

The correlated random variables considered in the experiments are gate length $L$, gate width $W$ and threshold voltage $V_{th}$. We use the spatial correlation model in [13] to model parameter correlations. After performing PCA, we order the resulting eigenvalues $\{\lambda_i\}$ such that $\lambda_i > \lambda_{i+1}$, and truncate these eigenvalues by finding the smallest $M$ such that $\lambda_M(\sum_{i=1}^{M} \lambda_i)^{-1} \leq \epsilon$, where $\epsilon$ is a threshold decided by the designer. We set $\epsilon = 0.01$ and preserve six normal random variables for all the ISCAS85 benchmark circuits.

In the linear condition, both MASCM and ASCM will directly choose the MAX input with the larger mean as the MAX output. In the weakly nonlinear condition, MASCM and ASCM employ the modified nested sparse grid quadrature and non-nested sparse grid quadrature to approximate the MAX operator respectively, and the numbers of collocation points required by MASCM and ASCM for each MAX approximation are 73 and 85 respectively. In the strongly nonlinear condition, MASCM still employs the modified nested sparse grid quadrature to approximate the MAX operator, while ASCM uses tensor product quadrature, and the numbers of collocation points required by MASCM and ASCM for each MAX approximation are 73 and 729, respectively.

Table 2 gives the comparison of mean errors, standard deviation (SD) errors and run time between the proposed MASCM and the existing ASCM [8], [9] on ISCAS85 benchmark circuits. Table 2 also lists the number of three kinds of input conditions when approximating the MAX operator in each circuit. All data for accuracy comparison in Table 2 is the relative error compared with 10,000 Monte Carlo (MC) simulations. We also give the average errors and run time of MASCM and ASCM as well as the average percentage of three input conditions at the bottom of Table 2.

It can be observed from Table 2 that compared with Monte Carlo method, the proposed MASCM has less than 0.6% errors in mean and standard deviation, and at the same time about 30X improvement in run time. From Table 2, we can also see that the proposed MASCM has about 50% reduction in run time compared with ASCM, while both mean errors and standard deviation errors of MASCM are very small and comparable with ASCM. The time reduction of MASCM is mainly due to the replacement of tensor product quadrature by the modified nested sparse grid quadrature in the strongly nonlinear condition. Since there exist about 31% nonlinear conditions in the experiment, the modified nested sparse grid quadrature gives good balance of run time and accuracy. On the other hand, for the existing ASCM, if we replace tensor product quadrature by non-nested sparse grid quadrature in the strongly nonlinear condition, the computation time of ASCM can be reduced. However, the corresponding computation error will be very large. For instance, for the benchmark circuit c3540, the mean and standard deviation errors will be as high as 23.8% and 56.8%, respectively.

### Table 2: Results of SSTA with ISCAS85 benchmark circuits.

<table>
<thead>
<tr>
<th>Circuits</th>
<th>Mean error (%)</th>
<th>SD error (%)</th>
<th>Run time (seconds)</th>
<th>Number of input conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ASCM</td>
<td>MASCM</td>
<td>ASCM</td>
<td>MASCM</td>
</tr>
<tr>
<td>c432</td>
<td>0.022</td>
<td>0.045</td>
<td>0.445</td>
<td>0.451</td>
</tr>
<tr>
<td>c499</td>
<td>0.029</td>
<td>0.029</td>
<td>0.168</td>
<td>0.166</td>
</tr>
<tr>
<td>c880</td>
<td>0.074</td>
<td>0.059</td>
<td>0.323</td>
<td>0.240</td>
</tr>
<tr>
<td>c1908</td>
<td>0.043</td>
<td>0.047</td>
<td>0.394</td>
<td>0.377</td>
</tr>
<tr>
<td>c2670</td>
<td>0.081</td>
<td>0.004</td>
<td>0.055</td>
<td>0.448</td>
</tr>
<tr>
<td>c3540</td>
<td>0.095</td>
<td>0.082</td>
<td>0.273</td>
<td>0.427</td>
</tr>
<tr>
<td>c5315</td>
<td>0.089</td>
<td>0.090</td>
<td>0.413</td>
<td>0.410</td>
</tr>
<tr>
<td>c6288</td>
<td>0.061</td>
<td>0.062</td>
<td>0.239</td>
<td>0.243</td>
</tr>
<tr>
<td>c7552</td>
<td>0.130</td>
<td>0.110</td>
<td>0.551</td>
<td>0.517</td>
</tr>
<tr>
<td>Average</td>
<td>0.069</td>
<td>0.059</td>
<td>0.318</td>
<td>0.364</td>
</tr>
</tbody>
</table>
5. Conclusions and Future Work

In this paper, we propose a modified nested sparse grid quadrature method for the MAX approximation in SSTA. The modified nested sparse grid quadrature provides higher computation accuracy than the non-nested one with even lower computational cost. Moreover, compared with the tensor product quadrature, it can achieve comparable accuracy with greatly reduced computational cost. As a result, we propose a Modified nested sparse grid based Adaptive Stochastic Collocation Method (MASCM) for block-based statistical static timing analysis. The MASCM employs the modified nested sparse grid quadrature to approximate the MAX operator for both weakly and strongly nonlinear conditions. The MASCM has much higher computation efficiency than the existing Adaptive Stochastic Collocation Method [8], [9], while still maintains very high computation accuracy. Experiment results on ISCAS85 benchmark circuits with 65 nm technology have demonstrated the merits of the proposed MASCM. In this paper, we focus on process variations modeled by correlated random variables with Gaussian probability distributions. In the future work, we will generalize the proposed MASCM for random variables with arbitrary probability distributions.

Acknowledgments

This research is supported by National major Science and Technology special project 2008ZX01035-001-06 and 2009ZX02023-4-3 of China during the 11-th five-year plan period, NSFC research project 60976034, 60676018 and 60806013, the National Basic Research Program of China under the grant 2005CB321701, the doctoral program foundation of Ministry of Education of China 200802460068, the International Science and Technology Cooperation program foundation of Shanghai 08510700100, the program for Outstanding Academic Leader of Shanghai. W. Cai is supported partly by the US National Science Foundation (grant number: DEFG0205ER25678) and the US Department of Energy (grant number: DEF0205SER25678).

References


Xu Luo received the B.S. in Electronic Science and Technology from Xidian University, Xi’an, China, in 2004. He is currently working toward the Ph.D. degree at the State Key Lab of ASIC and System, Fudan University, Shanghai, China. His research interests include parameter extraction and variation-aware stochastic methods for the modeling and simulation of VLSI circuits.
Fan Yang received the B.E. degree in electronic engineering from Xi’an Jiaotong University, Xi’an, China, in 2003. He received the Ph.D. degree in microelectronics from Fudan University, Shanghai, China, in 2008. He is now an assistant professor of the microelectronics department, Fudan University. His research interests include model order reduction and circuit simulation.

Xuan Zeng received the B.Sc. and Ph.D. degrees in electrical engineering from Fudan University, Shanghai, China, in 1991 and 1997, respectively. She joined the Electrical Engineering Department, Fudan University in 1997 and became a full professor in Microelectronics Department in 2001. Now she serves as the Director of the State Key Lab of ASIC and System of Fudan University. She was a visiting professor in the Electrical Engineering Department, Texas A&M University, U.S.A. and Microelectronics Department of TU Delft, Netherlands in 2002 and 2003 respectively. Her research interests include DFM, analog and mixed signal design automation (behavioral modeling, circuit simulation and analog layout generation), high speed interconnect analysis and design and ASIC design. Dr. Zeng received the Cross-Century Outstanding Scholar Award from the Ministry of Education of China in 2002. She was selected into IT Top 10 in Shanghai China in 2003. She served in the technical program committee of IEEE/ACM ASP-DAC in 2000 and 2005.

Jun Tao received the B.S. and Ph.D. degrees in electronic engineering from Fudan University, Shanghai, China, in 2002 and 2007, respectively. She is currently an Assistant Professor of the Microelectronics department, Fudan University. Her research interests include ECP/CMP process modeling, circuit simulation and timing analysis.

Wei Cai received the Ph.D. degree in applied mathematics from Brown University, Providence, RI in 1989. He joined the Department of Mathematics, the University of North Carolina at Charlotte in 1989 as an assistant professor, later became an associate professor in 1995. In 1995, he joined the Department of Mathematics at the University of California at Santa Barbara as an Assistant Professor and later an Associate Professor. In 1996, he returned to the Department of Mathematics, the University of North Carolina at Charlotte and became a full Professor in 1999. His research interest includes numerical methods for model reduction, computational electromagnetics for parameter extraction for computer packaging and VLSI designs and photonic devices.

Hengliang Zhu received the B.S. in electronic engineering from University of Science and Technology of China, Hefei, China, in 2004. He is currently working toward the Ph.D. degree at the State Key Lab of ASIC and System of Fudan University, Shanghai, China. His research interests include variational-aware parameter extraction algorithms for VLSI circuits.