

# Efficient Warpage Simulation of Complex 2.5-D/3-D IC Structures with Novel Meshing Algorithm and Layerwise Plate Theory

Tianxiang Zhu<sup>1</sup>, Qipan Wang<sup>1,2</sup>, Yibo Lin<sup>1,3,4\*</sup>, Runsheng Wang<sup>1,3,4\*</sup>,

<sup>1</sup>School of Integrated Circuits, <sup>2</sup>Academy for Advanced Interdisciplinary Studies, Peking University, Beijing,

<sup>3</sup>Institute of Electronic Design Automation, Peking University, Wuxi,

<sup>4</sup>Beijing Advanced Innovation Center for Integrated Circuits  
{txzhu, qpwang, yibolin, r.wang}@pku.edu.cn

**Abstract**—Nowadays, warpage effect is becoming one of the main concerns in the manufacture of 2.5-D/3-D IC packages. Numerical simulation of warpage in the design stage by the finite element method (FEM) is required for manufacturability and reliability optimization. 2.5-D/3-D IC packages are generally composed of laminated thin plates with high aspect ratios and complex in-plane material boundaries, leading to intrinsic difficulties in obtaining high-quality hexahedral meshes essential for fast convergence and high-quality results. In this paper, we propose a novel meshing algorithm for efficient generation of sweep hexahedral meshes towards complex 2.5-D/3-D structures. On the basis of the sweep mesh, we utilize a modified 2-D layerwise plate theory to further improve the convergence of the solver. Compared with Ansys Workbench, our meshing algorithm can either reduce the meshing time ( $74.7\times$  to  $221\times$ ) and the number of mesh nodes ( $5.26\times$  to  $18.4\times$ ), or improve the mesh quality ( $3.45\times$  to  $9.75\times$ ) and reduce convergence time of the solver ( $1.48\times$  to  $4.50\times$ ), with  $< 0.5\%$  errors. A  $3.75\times$  to  $12.6\times$  reduction in convergence time is further achieved with the proposed 2-D layerwise plate theory compared to the 3-D formulation, while maintaining the errors within 3%.

## I. INTRODUCTION

Driven by the diminishing benefits brought by process scaling, chiplet-based integration technologies represented by 2.5-D/3-D IC packaging have prevailed in the post-Moore Era [1]. Despite its numerous advantages, 2.5-D/3-D packaging remains challenged by critical manufacturability and reliability issues, one of which is the warpage effect [2], [3]. Warpage is the non-uniform z-direction deformation of the package induced by the mismatch of the thermal expansion coefficients of different materials, together with the thermal load between fabrication temperature and room temperature. Severe warpage will cause de-bonding of interconnects, complicate alignment, and incur structural damage, making design-stage analysis crucial for manufacturability and reliability optimization [4]–[6].

In practice, analytical models, such as the Suhir model [7], are inadequate for warpage analysis of complex structures. Thus, structural mechanics simulation of the package is required to obtain the detailed warpage profile, which is widely performed by the finite element method (FEM) and in commercial FEM software like ANSYS Workbench [8].

Nonetheless, the unique structures of 2.5-D/3-D ICs introduce intrinsic challenges for finite element simulation. In FEM, preparation of a high-quality mesh is a critical

\*Corresponding authors.

This work was supported in part by the National Science Foundations of China (Grant No. 62125401), the 111 project (B18001), and the Beijing Outstanding Young Scientist Program (JWZQ20240101004).

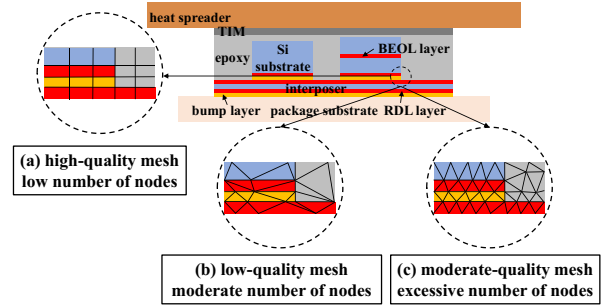


Fig. 1. Schematic of a hybrid 2.5-D/3-D IC package, which can be regarded as a lamination of thin layers with high width-to-thickness ratios. For this kind of structure, (a) sweep hexahedral meshes have high mesh qualities and low numbers of mesh nodes, while tetrahedral meshes either suffer from (b) low qualities for moderate numbers of mesh nodes, or (c) moderate qualities with excessive numbers of mesh nodes.

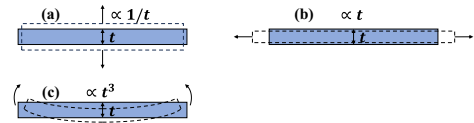


Fig. 2. Schematics of the (a) out-of-plane stiffness, (b) membrane stiffness, and (c) bending stiffness of a thin plate. They are scaled by  $1/t$ ,  $t$ , and  $t^3$ , respectively. The out-of-plane (z-direction) stiffness is the source of the high condition number of the system stiffness matrix.

procedure and consumes a substantial portion of time [9], [10]. Experts have often preferred hexahedral meshes over tetrahedral ones in structural mechanics simulation because they usually deliver faster convergence and better quality of results [11]–[14]. The advantages are more prominent in 2.5-D/3-D ICs, whose structures can be regarded as lamination of a series of thin layers with very high width-to-thickness ratios [15]. As shown in Fig. 1, for these thin layers, meshing with tetrahedral elements will easily result in low-quality elements with high skewness and high maximum corner angles, unless the lateral dimensions of the elements are in the same magnitude order as the vertical ones. However, the latter will lead to excessive mesh elements and nodes, significantly increasing the meshing time and the subsequent computational cost as a consequence.

In order to automatically generate high-quality hexahedral meshes for general 2.5-D/3-D IC structures, meshing a 2-D surface first and then sweeping the 2-D mesh along the z dimension is a preferable choice to meshing the whole 3-D structure directly, considering the high width-to-thickness ratios of the laminated layers. This ensures that the sides of the elements are perpendicular to their undersides to minimize skewness, and reduces meshing time by converting structures to be meshed from 3-D to 2-D. However, there

exist many in-plane material boundaries in each layer, such as the boundaries between chiplets and epoxy filler, and the boundaries introduced by material homogenization of local interconnects [3], [16]–[19]. The misalignment of these complex boundaries makes existing sweeping algorithms inapplicable to real-world 2.5-D/3-D IC structures.

Another critical numerical problem induced by the high width-to-thickness ratios is the huge difference among the out-of-plane stiffness ( $\propto 1/t$ ), the membrane stiffness ( $\propto t$ ), and the bending stiffness ( $\propto t^3$ ) when the thickness  $t$  is small (Fig. 2) [20]. This anisotropy can cause a significant difference in the magnitudes of the maximal and minimal singular values of the system stiffness matrix, thereby causing a sharp increase in the condition number of the matrix, deteriorating numerical stability, and slowing down convergence or even preventing it altogether.

In this work, we propose a novel meshing algorithm and further a modified layerwise plate theory to address the above-mentioned challenges. An integrated simulator based on the proposed algorithms is released on GitHub [21]. The main contributions of our work are as follows:

- We propose a novel meshing algorithm for fast generation of high-quality sweep hexahedral meshes towards complex 2.5-D/3-D IC structures, which employs an efficient projection method to divide the target structures.
- Based on the sweep mesh, we replace the full 3-D structural mechanics theory with a modified 2-D layerwise plate theory to eliminate the out-of-plane stiffness contribution and further improve the convergence of the iterative solver.
- We test our algorithms on several complex 2.5-D/3-D IC packages. Compared with Ansys Workbench, our meshing algorithm can either reduce the meshing time ( $74.7\times$  to  $221\times$ ) and the number of mesh nodes ( $5.26\times$  to  $18.4\times$ ), or improve the mesh quality ( $3.45\times$  to  $9.75\times$ ) and reduce the convergence time of the solver ( $1.48\times$  to  $4.50\times$ ), with errors  $< 0.5\%$ . A  $3.75\times$  to  $12.6\times$  speedup of convergence time is further achieved with the proposed 2-D layerwise plate theory compared to the 3-D formulation, with errors  $< 3\%$ .

## II. RELATED WORK

### A. Warpage modeling and simulation

Suhir proposed the first analytical warpage model based on the classical Kirchhoff-Love plate theory [7]. After that, many analytical models with various modifications to the Suhir model have been proposed [22]–[24]. However, analytical models are overly simplistic and difficult to apply to various complex structures, making detailed numerical simulation essential for warpage analysis of 2.5D/3D ICs, as the warping effect becomes increasingly serious.

Previous works on warpage simulation can be classified into two categories. The first category focuses on obtaining homogenized equivalent material parameters for local microstructures (such as redistribution metal traces, bumps, and through-silicon vias) to simplify models and reduce the number of degrees of freedom (DoFs) [3], [16]–[19]; the second category studies how to solve more efficiently given existing models and equivalent material parameters.

Fewer studies focused on the second category before, while the most recent and relevant study was performed

by Lo et al., who utilized a modified classical plate theory to perform dimension reduction and obtained a remarkable speedup compared with commercial software [15]. However, their method belongs to the equivalent single-layer plate theory, which is difficult to apply to complex, real-world 2.5D/3D IC structures containing many unaligned layers, and the authors did not consider how to mesh a complex structure in their original work, where the test cases only contain two layers. Our current work belongs to the second category, and we focus on the simulation acceleration of complex 2.5D/3D IC structures.

### B. Mesh generation

There are many recent works focusing on the generation of meshes for thermal simulation of 2.5D/3D ICs. For example, Chen et al. combined neural network and mathematical methods to accelerate the generation of tetrahedral meshes [10]. Wang et al. proposed a multi-level mesh generation algorithm based on the quad-tree structure [25]. However, structural mechanics simulation differs a lot from thermal simulation. It is more sensitive to qualities of meshes, where hexahedral ones are usually superior to tetrahedral ones in convergence and quality of results [14]. In our work, considering the unique layerwise structures of 2.5D/3D ICs, we utilize sweeping techniques to generate high-quality pure hexahedral meshes.

Previous studies on sweep mesh generation mainly focus on the division of the so-called “one-to-many” or “many-to-many” geometries or the adaptive sweeping of curved geometries [11]–[14]. These research scenarios are not consistent with ours, where many stacked layers with unaligned in-plane boundaries need to be converted into a sweepable configuration efficiently.

### C. Plate theories

Plate theories were developed in the last century to address the issues of laminated thin plates [20]. There are numerous variants [26]–[28], but all of them share the same principle, which is expanding the displacements in terms of the  $z$ -coordinate, thus converting the original 3-D problem into a 2-D problem. Lo et al. used a modified single-layer equivalent classical plate theory in their work, which regards different layers as one equivalent layer [15]. It is not suitable for complex structures where layers are not aligned. We combine a modified layerwise formulation and the first-order shear deformation (FSDT) theory in our work to overcome the above-mentioned disadvantages.

## III. PRELIMINARIES

### A. 3-D formulation for structural mechanics simulation

In 3-D scenarios, the  $x$ ,  $y$ ,  $z$  displacement components are denoted by  $u(x, y, z)$ ,  $v(x, y, z)$ ,  $w(x, y, z)$ , respectively. The distribution of  $w$  is the warpage we aim to obtain. In the Voigt notation, the strain vector  $\varepsilon$  is given by [20]:

$$\varepsilon = (\varepsilon_x, \varepsilon_y, \varepsilon_z, 2\varepsilon_{yz}, 2\varepsilon_{xz}, 2\varepsilon_{xy})^T \quad (1)$$

$$= \left( \frac{\partial u}{\partial x}, \frac{\partial v}{\partial y}, \frac{\partial w}{\partial z}, \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}, \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}, \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^T.$$

The stress vector  $\sigma = (\sigma_x, \sigma_y, \sigma_z, \sigma_{yz}, \sigma_{xz}, \sigma_{xy})^T$  can be derived by the constitutive relation of the materials:

$$\sigma = C \cdot (\varepsilon - \alpha \cdot \Delta T), \quad (2)$$

where  $C$  is the stiffness matrix of the materials,  $\alpha$  is the thermal expansion coefficient vector, and  $\Delta T$  is the constant thermal load ( $\Delta T = T_{room} - T_{fabrication}$ ).

After that, by expanding  $u$ ,  $v$ , and  $w$  with the basis functions (shape functions) attached to the DoFs of the mesh, and applying Eq. 1 and 2 to the virtual work principle

$$\delta E = \int_V \sigma \cdot \delta \varepsilon \, dx dy dz = 0. \quad (3)$$

The final system of linear equations can be obtained through an assembling procedure:

$$Ax = b, \quad (4)$$

where  $A$  is the system stiffness matrix,  $b$  is the load vector, and  $x$  is the displacement solution vector. Due to the large scale of the system and the large number of DoFs, iterative methods are generally used for solving Eq. 4.

### B. Definition of the 2.5-D/3-D IC structures

The structures that our current work can support are defined as follows. 2.5-D/3-D ICs are regarded as  $K$  stacked rectangle layers  $l_i$  with certain filling materials  $M_0^{l_i}$ , coordinates of the lower-left corners  $(X_0^{l_i}, Y_0^{l_i})$ , and dimensions  $(L_0^{l_i}, W_0^{l_i})$  where  $i = 1, 2, \dots, K$ . All of the layers form a layer set  $L$ . Inside each layer, there are  $N^{l_i}$  rectangle areas  $r_j^{l_i}$  with certain filling materials  $M_j^{l_i}$ , relative lower-left coordinates  $(X_j^{l_i}, Y_j^{l_i})$ , and dimensions  $(L_j^{l_i}, W_j^{l_i})$  where  $j = 1, 2, \dots, N^{l_i}$ .  $M_j^{l_i}$  overrides  $M_0^{l_i}$  if they are not the same. These rectangular areas may originate from chiplets, homogenization of local interconnects, and others. They, together with the layer boundary itself, form the rectangle set of  $l_i$ , denoted as  $R^{l_i}$ . In the  $z$ -direction, a layer and its owned rectangles share the same start  $z$ -coordinate  $z^{l_i}$  and thickness  $t^{l_i}$ .

The filling materials  $M_0^{l_i}$  and  $M_j^{l_i}$  may be assigned as empty, which means hollow spaces. For example, there may be no gap filler between chiplets in a 2.5-D IC. The start  $z$ -coordinates  $z^{l_i}$  and thickness  $t^{l_i}$  of each layer may not align with each other, as well. For example, chiplet A may consist of a different number of layers than chiplet B.

This definition is very flexible and can cover most of the 2.5-D, 3-D, and hybrid ICs because package components and homogenized areas are generally in the shape of rectangles [3], [16]–[18], [29]. Our meshing algorithm can be adapted for other geometries like circles, which will be supported in our future work.

## IV. EFFICIENT WARPAGE SIMULATION ALGORITHMS

In this section, we elaborate on our algorithms for efficient warpage simulation of 2.5D/3D ICs. Our sweep mesh generation algorithm is first explained. Subsequently, we introduce the modified layerwise plate theory and its application to the current problem based on the generated sweep mesh.

### A. Sweep hexahedral mesh generation

One direct way to generate a sweep hexahedral mesh is to project all of the rectangles onto a 2-D plane, mesh the 2-D plane with quadrilateral elements, and then sweep the 2-D mesh along the  $z$ -coordinate. However, this method equivalently introduces numerous unnecessary boundaries into the target structure. As shown in Fig. 3, if the interspace between the boundaries of two layers is empty, these

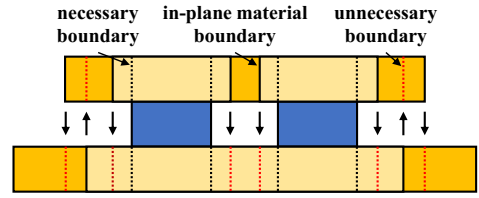


Fig. 3. Simply projecting all rectangles onto a 2-D plane to create a sweep mesh will equivalently introduce numerous unnecessary boundaries (dotted line) into the target structure.

boundaries should not be projected. Because the mesh has to be conformal to all of the boundaries in the structure, these unnecessary boundaries will result in a longer meshing time and a larger number of DoFs, and will deteriorate the quality of the mesh if the unnecessary boundaries are too close to the necessary boundaries.

The aim of our algorithm is to divide the target structure automatically and avoid introducing any unnecessary boundaries. We develop an efficient projection method to fulfill this task. It begins with gathering all of the start and end  $z$ -coordinates and splitting all layers at these  $z$ -coordinates. The generated  $K'$  sub-layers  $sl_i$ , where  $i = 1, 2, \dots, K'$ , now have aligned  $z$ -coordinates and thicknesses, and the rectangle sets of the original layers are also copied to their sub-layers. The sub-layers form a sub-layer set  $SL$  and are numbered from the top to the bottom.

Next, the algorithm generates the source plane  $\mathcal{P}^{sl_{K'}}$  to start sweeping from:

1) *Initialize*: A R-tree denoted by  $\mathcal{R}$  is first initialized to efficiently manage the spatial relationship of the geometries, and the rectangle set of the first sub-layer  $R^{sl_1}$  is inserted into it. An empty rectangle set  $OR_{sl_0}^{sl_1}$  is also initialized.

2) *The first and the second sub-layers*: Then, insert the rectangle set of the second sub-layer  $R^{sl_2}$  into  $\mathcal{R}$ . After that, the algorithm finds the overlapping set  $OR_{sl_1}^{sl_2} = R^{sl_2} \cap (R^{sl_1} \cup OR_{sl_0}^{sl_1})$ . This set contains all of the overlapping areas between  $r_j^{sl_1} \in (R^{sl_1} \cup OR_{sl_0}^{sl_1})$  and  $r_{j'}^{sl_2} \in R^{sl_2}$ . The overlapping set can be set up efficiently with the R-tree.

If two rectangles  $r_j^{sl_1}$  and  $r_{j'}^{sl_2}$  overlap, their overlapping part is also a rectangle, denoted as  $r_{j''}^{j'}$ . The minimal  $x$ -coordinate of the rectangle  $r_{j''}^{j'}$  is the larger one of the minimal  $x$ -coordinates of  $r_j^{sl_1}$  and  $r_{j'}^{sl_2}$ , while the maximal  $x$ -coordinate is the smaller one of the maximal  $x$ -coordinates of  $r_j^{sl_1}$  and  $r_{j'}^{sl_2}$ . The same goes for the  $y$ -coordinates.  $r_{j''}^{j'}$  is then added to the overlapping set.

After finding each overlapping rectangle, delete  $R^{sl_1}$  from  $\mathcal{R}$  and add  $OR_{sl_1}^{sl_2}$  into  $\mathcal{R}$ . Till now, we have fully selected the necessary geometry boundaries of sub-layer  $sl_1$  on sub-layer  $sl_2$ , and discarded the unnecessary boundaries that are not in the coverage area of  $sl_2$ .

3) *The  $i$ -th and the  $(i+1)$ -th sub-layers*: Following the above-mentioned procedures, the rectangle set of the  $i$ -th sub-layer  $R^{sl_i}$  together with the overlapping rectangle set between the  $i$ -th layer and  $(i-1)$ -th layer  $OR_{sl_{i-1}}^{sl_i}$  now includes the necessary boundaries from  $sl_1$  to  $sl_i$ . By repeating this set of procedures until the last sub-layer  $sl_{K'}$ , the rectangle set  $(R^{sl_{K'}} \cup OR_{sl_{K'-1}}^{sl_{K'}})$  with all necessary boundaries projected onto it and the unnecessary boundaries discarded can be obtained.

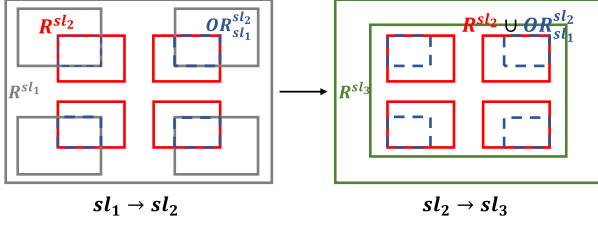


Fig. 4. Schematic of the proposed projection method, taking the projection process from the sub-layer  $sl_1$  to  $sl_2$  and  $sl_2$  to  $sl_3$  for example. This projection method efficiently and automatically includes all necessary boundaries and discards all unnecessary boundaries.

4) *2-D meshing and sweeping*: After obtaining  $(R^{sl_{K'}} \cup OR_{sl_{K'-1}}^{sl_{K'}})$ , we can perform the self-fragmentation of the rectangles in it and get the source plane  $\mathcal{P}^{sl_{K'}}$  for sweeping. A 2-D quadrilateral mesh is first generated on  $\mathcal{P}^{sl_{K'}}$ , and then the mesh is swept from the bottom to the top along the z-direction. Another R-tree  $\mathcal{R}'$  is maintained during this process to sweep the fragmented rectangles covered by  $OR_{sl_i}^{sl_{i+1}}$ , together with the parts which are discarded when previously projecting  $R^{sl_i}$  to  $R^{sl_{i+1}}$ , which is the difference set  $DR_{sl_i}^{sl_{i+1}} = R^{sl_i} - OR_{sl_i}^{sl_{i+1}}$ .

After completing all of the procedures, a pure sweep hexahedral mesh can be obtained for the target 2.5-D/3-D IC structure. The schematic of our algorithm is shown in Fig. 4 and its implementation is concluded in Algorithm 1.

### B. 2-D Layerwise plate theory

Although we have obtained hexahedral meshes with perpendicular sides and potential skewness which only originates from the 2-D meshing, the width-to-thickness ratios of the elements are rather high, resulting in an increase of the condition number of the system stiffness matrix A. This issue mainly stems from the out-of-plane stiffness term ( $\propto 1/t$ ), which will be excessively large when the thicknesses  $t$  of the layers are small. Adopting the plate theory, we can eliminate the out-of-plane stiffness and improve the condition number and convergence.

The first-order shear deformation (FSDT) theory is an improved version of the classical plate theory used in [15]. It can be used for plates with very small thicknesses to moderate thicknesses, and is adopted in our work. The formulation of the original FSDT is [20]:

$$\begin{aligned} u(x, y, z) &= u_0(x, y) + z\phi_x(x, y), \\ v(x, y, z) &= v_0(x, y) + z\phi_y(x, y), \\ w(x, y, z) &= w_0(x, y), \end{aligned} \quad (5)$$

where  $u_0(x, y)$ ,  $v_0(x, y)$ , and  $w_0(x, y)$  are the displacements of the mid-plane, and  $\phi_x(x, y)$  and  $\phi_y(x, y)$  are the rotations of a transverse normal about the y- and x-axes. This formulation eliminates the contribution of the out-of-plane stiffness, because the z-displacement is now invariant to z, and therefore the gradient term  $\frac{\partial w}{\partial z}$  vanishes.

For multi-layers, FSDT treats them as an equivalent single layer. However, this treatment is not suitable for the 2.5-D/3-D IC scenarios, where different layers may not be aligned in the x- and y-direction. To improve the flexibility of the original FSDT, we modify it into a layerwise formulation:

$$\begin{aligned} u_i(x, y, z) &= u_i^b(x, y)\varphi_i^b(z) + u_i^t(x, y)\varphi_i^t(z), \\ v_i(x, y, z) &= v_i^b(x, y)\varphi_i^b(z) + v_i^t(x, y)\varphi_i^t(z), \\ w_i(x, y, z) &= w_i(x, y), \end{aligned} \quad (6)$$

### Algorithm 1 Implementation of our meshing algorithm

**Input:** layer set  $L = \{l_i\}, i = 1, 2, \dots, K$ ; each layer  $l_i$  contains a rectangle set  $R^{l_i}$  with  $N^{l_i} + 1$  rectangles

**Output:** a sweep hexahedral mesh

- 1: split the layers at all  $z^{l_i}$ s and  $z^{l_i} + t^{l_i}$ s to form a sub-layer set  $SL = \{sl_i\}, i = 1, 2, \dots, K'$  numbering from the top to the bottom; copy  $R^{l_i}$  of  $l_i$  to its sub-layers
- 2: initialize a R-tree  $\mathcal{R}$ ; insert  $R^{sl_1}$  into  $\mathcal{R}$
- 3: initialize an empty rectangle set  $OR_{sl_0}^{sl_1}$
- 4: **for**  $i = 1, 2, \dots, K' - 1$  **do**
- 5:   insert  $R^{sl_{i+1}}$  into  $\mathcal{R}$
- 6:   find the overlapping rectangle set  $OR_{sl_i}^{sl_{i+1}} = R^{sl_{i+1}} \cap (R^{sl_i} \cup OR_{sl_{i-1}}^{sl_i})$
- 7:   delete  $R^{sl_i}$  and  $OR_{sl_{i-1}}^{sl_i}$  from  $\mathcal{R}$
- 8:   insert  $OR_{sl_i}^{sl_{i+1}}$  into  $\mathcal{R}$
- 9: **end for**
- 10: perform self-fragmentation of  $(R^{sl_{K'}} \cup OR_{sl_{K'-1}}^{sl_{K'}})$  and obtain the source plane  $\mathcal{P}^{sl_{K'}}$  for sweeping
- 11: generate a 2-D quadrilateral mesh on  $\mathcal{P}^{sl_{K'}}$
- 12: initialize a R-tree  $\mathcal{R}'$ ; insert  $R^{sl_{K'}}$  into  $\mathcal{R}'$
- 13: sweep  $\mathcal{P}^{sl_{K'}}$  along  $(0, 0, t^{sl_{K'}})$
- 14: **for**  $i = K' - 1, K' - 2, \dots, 1$  **do**
- 15:   insert  $R^{sl_i}$  to  $\mathcal{R}'$
- 16:   find the difference set  $DR_{sl_i}^{sl_{i+1}} = R^{sl_i} - OR_{sl_i}^{sl_{i+1}}$
- 17:   let  $\mathcal{P}^{sl_i} = DR_{sl_i}^{sl_{i+1}} + \{r \text{ covered by } OR_{sl_i}^{sl_{i+1}}\}$
- 18:   sweep  $\mathcal{P}^{sl_i}$  along  $(0, 0, t^{sl_i})$
- 19:   delete  $R^{sl_{i+1}}$  from  $\mathcal{R}'$
- 20: **end for**

where  $u_i$ ,  $v_i$ , and  $w_i$  are the displacements of the i-th sub-layer;  $u_i^b$  and  $v_i^b$  are the displacements of the bottom surface of the i-th sub-layer, while  $u_i^t$  and  $v_i^t$  are for the top surface; the z-displacements  $w_i$  are assumed to be constant through the thickness of each sub-layer as in FSDT.  $\varphi_i^b$  and  $\varphi_i^t$  are the z-direction basis functions:

$$\varphi_i^b(z) = 1 - \frac{z - z^{sl_i}}{t^{sl_i}}, \quad \varphi_i^t(z) = \frac{z - z^{sl_i}}{t^{sl_i}}. \quad (7)$$

Applying Eq. 6 and 7 to Eq. 1 and utilizing  $\varphi_i^b(z) + \varphi_i^t(z) \equiv 1$ , the strain vector  $\varepsilon_i$  of the i-th sub-layer is:

$$\begin{aligned} \varepsilon_i &= \varepsilon_i^b \varphi_i^b(z) + \varepsilon_i^t \varphi_i^t(z), \\ \varepsilon_i^b &= \left( \frac{\partial u_i^b}{\partial x}, \frac{\partial v_i^b}{\partial y}, \frac{v_i^t - v_i^b}{t_i} + \frac{\partial w_i}{\partial y}, \frac{u_i^t - u_i^b}{t_i} + \frac{\partial w_i}{\partial x}, \frac{\partial u_i^b}{\partial y} + \frac{\partial v_i^b}{\partial x} \right)^T, \\ \varepsilon_i^t &= \left( \frac{\partial u_i^t}{\partial x}, \frac{\partial v_i^t}{\partial y}, \frac{v_i^t - v_i^b}{t_i} + \frac{\partial w_i}{\partial y}, \frac{u_i^t - u_i^b}{t_i} + \frac{\partial w_i}{\partial x}, \frac{\partial u_i^t}{\partial y} + \frac{\partial v_i^t}{\partial x} \right)^T. \end{aligned} \quad (8)$$

Applying Eq. 8 into Eq. 2, the stress vector  $\sigma_i$  is:

$$\begin{aligned} \sigma_i &= \sigma_i^b \varphi_i^b(z) + \sigma_i^t \varphi_i^t(z), \\ \sigma_i^b &= \mathbf{C}_{2D} \cdot (\varepsilon_i^b - \alpha \cdot \Delta T), \\ \sigma_i^t &= \mathbf{C}_{2D} \cdot (\varepsilon_i^t - \alpha \cdot \Delta T), \end{aligned} \quad (9)$$

where  $\mathbf{C}_{2D}$  is the reduced material stiffness matrix with the out-of-plane normal term removed [20].

By applying Eq. 8 and 9 to Eq. 3, integrating over the z-dimension, and then summing over the sub-layers, we can derive the virtual work formulation of the proposed modified plate theory ( $b \leftrightarrow t$  means swapping the superscripts):

$$\delta E = \sum_i t_i \int_{S_i} \left( \frac{\sigma_i^b \cdot \delta \varepsilon_i^b}{3} + \frac{\sigma_i^t \cdot \delta \varepsilon_i^t}{6} \right) dx dy + b \leftrightarrow t. \quad (10)$$

Then, following the standard FEM procedures, the system of linear equations as Eq. 4 can be obtained.

Additionally, multi-point constraints are applied to the system to bind up the displacements of adjacent sub-layers. For sub-layers  $sl_i$  and  $sl_{i+1}$ , the multi-point constraints applied to the contact parts are:

$$u_i^b = u_{i+1}^t, v_i^b = v_{i+1}^t, w_i = w_{i+1}. \quad (11)$$

During implementation, we only take the 2-D mesh on the bottom surface of each sub-layer in the generated sweep mesh as the computational domain and apply multi-point constraints on vertically corresponding DoFs. A non-sweep mesh cannot support such operations. Therefore, our sweep mesh generation algorithm is the basis of the proposed 2-D layerwise formulation.

## V. EXPERIMENTAL SETUP AND RESULTS

### A. Implementation

Our meshing algorithm is implemented in Python, using the Python interface of Gmsh [30] to perform CAD operations, 2-D mesh generation, and sweeping. The plate-theory-based FEM solver is implemented with an open-sourced C++ FEM backend, DOLFINx [31]–[34]. Because the number of DoFs is large for complex systems, we implement an iterative solver using the Python interface of PETSc [35] to solve the systems of linear equations. Both the assembling and solving processes can be performed in parallel.

### B. Experimental setup

Because there are no open-sourced benchmarks for warpage simulation, we generate 6 test cases following previous works [15], [25], [36]. Lamination details of the 2.5-D/3-D IC packages are retained, and a certain amount of rectangular areas with different homogenized materials are added to the interposer layers, substrate layers, redistribution layers, and bump layers to mimic the in-plane heterogeneity in real scenarios [3]. Brief descriptions of the test cases are listed in Table I. The thermal load  $\Delta T$  is set to be  $-150$  K for all test cases [15], [24].

The ground truth simulations are performed in the commercial software Ansys Workbench [8] with tetrahedral meshes. Two types of meshes are obtained for each test case: one is a high-quality mesh with similar mesh quality metrics as the mesh generated by our algorithm, and the other is a low-quality mesh with a similar number of mesh nodes. The former one is used for node number comparison and ground truth simulation, and the latter one is used for quality comparison. Degree 2 Lagrange discretization is adopted for both tetrahedral and hexahedral meshes. Warpage distributions on the bottom of the structures are compared. We did not compare with [15] in this work because their method does not support the complex structures of our test cases.

The sampled minimal signed inverted condition number (minSICN) is chosen as the quality metric [30], [40], [41]:

$$\text{SICN} = \text{sign}(\det(J)) \cdot \frac{1}{\text{Condition Number of } J}, \quad (12)$$

where  $J$  is the Jacobian matrix of a mesh element. The minimal SICN at the sampling points in an element will be selected. The minSICN of an element ranges from  $-1$  to  $+1$ , and the closer it is to 1, the better the element's quality. Elements with  $\text{minSICN} < 0$  are invalid. We use the ratio of the number of elements with  $\text{minSICN} < 0.02$  to the total

TABLE I  
BRIEF DESCRIPTIONS OF THE TEST CASES.

Case	Package dimensions	Description
ATPlace 1 ~ 3 [36]	$57 \times 59 \text{ mm}^2$	3 chiplet placement results 11 chiplets in the system
WSE [37]	$215 \times 215 \text{ mm}^2$	Wafer-scale engine 12*7 chiplets in the system
GPU [38]	$44 \times 38 \text{ mm}^2$	Hybrid 2.5-D/3-D IC 4 HBM chiplets (3-D) 1 GPU chiplet (2-D)
PoP [39]	$50 \times 50 \text{ mm}^2$	An artificial package-on-package chip 4 chiplets on the bottom 1 chiplets on the top

number of elements as the quality indicator of the whole mesh, because the condition number of the system matrix is constrained by the worst elements in the mesh [9].

Rigidity motions are eliminated from the displacement results. For comparison of accuracy, we use the absolute percentage error to the maximal warpage  $\frac{100\%}{n} \sum_i \frac{|\hat{z}_i - z_i|}{z_{max} - z_{min}}$  as the reference point can be arbitrary, where  $\hat{z}_i$  is the warpage obtained by our algorithm and  $z_i$  is the ground truth.

The ground truth simulations run on an Intel Xeon Platinum 8582C server, and our algorithm runs on an Intel Xeon Silver 4210R server, both utilizing 16 threads.

### C. Results

1) *Summarized meshing results:* The comparison results of the sweep hexahedral meshes generated by our meshing algorithm with the tetrahedral meshes generated by Ansys are summarized in Table II. For all test cases, our algorithm demonstrates significant advantages in mesh generation time and number of mesh nodes (compared with high-quality tetrahedral meshes), or in mesh quality and final convergence time (compared with low-quality ones).

For the high-quality tetrahedral meshes generated by Ansys, the mesh quality metrics are kept close to our meshes. However, this leads to excessive numbers of mesh nodes for the tetrahedral meshes, and consequently, a much longer meshing time. The reduction in the number of mesh nodes gained by our algorithm ranges from  $5.26\times$  to  $18.4\times$ , while the reduction in meshing time ranges from  $74.7\times$  to  $221\times$ . Another main reason for the reduction in meshing time is that, in our algorithm only 2-D planes  $\mathcal{P}^{sl_i}$  needs meshing, while in the ground truth algorithm the whole 3-D structure needs meshing. Our algorithm also excels in the convergence time of the iterative solver, mainly because of fewer DoFs. Meanwhile, errors in the warpage results obtained by the combination of our meshing algorithm with the 3-D formulation are less than 0.5% for all of the cases. This indicates that our algorithm does not sacrifice accuracy for efficiency.

For the low-quality tetrahedral meshes generated by Ansys, the number of mesh nodes is kept close to ours. Consequently, the mesh qualities of the tetrahedral meshes are much inferior to those of our meshes. The improvement in mesh qualities obtained by our algorithm ranges from  $3.45\times$  to  $9.75\times$ . With similar number of mesh nodes, the convergence time of the tetrahedral mesh is clearly longer than that of our algorithm (from  $1.48\times$  to  $4.50\times$ ), implying a deterioration of the condition number. Besides, the meshing

TABLE II

SUMMARIZED RESULTS OF ANSYS [8] AND OUR MESHING ALGORITHM. WARPAGE DISTRIBUTIONS OBTAINED BY HIGH-QUALITY TETRAHEDRAL MESHES ARE THE GROUND TRUTH. "MESH QUALITY" IS DEFINED AS THE PROPORTION OF ELEMENTS WITH MINSICN < 0.02. "VS H" MEANS THE IMPROVEMENTS IN THE BRACKETS ARE COMPARED WITH THE HIGH-QUALITY TETRAHEDRAL MESHES; "VS L" INDICATES THE LOW-QUALITY ONES.

Case	High-quality tetrahedral mesh by Ansys					Low-quality tetrahedral mesh by Ansys					Ours				
	Mesh-ing time (s)	# of mesh nodes	Mesh quality (%)	Conver-gence time (s)	Error (%)	Mesh-ing time (s)	# of mesh nodes	Mesh quality (%)	Conver-gence time (s)	Error (%)	Mesh-ing time (s) (vs H)	# of mesh nodes (vs H)	Mesh quality (%) (vs L)	Conver-gence time (s) (vs L)	Error (%)
ATPlace 1	125.66	196,463	0.015	1718	-	10.02	19,820	0.123	490	2.79	1.17 (107×)	20,485 (9.59×)	0.016 (7.69×)	153 (3.20×)	0.37
ATPlace 2	135.54	195,987	0.0145	1521	-	9.66	19,555	0.073	352	3.29	1.06 (128×)	19,649 (9.97×)	0.013 (5.61×)	125 (2.82×)	0.36
ATPlace 3	133.65	195,199	0.015	1795	-	9.35	19,527	0.075	532	3.26	1.04 (128×)	19,329 (10.1×)	0.013 (5.77×)	136 (3.91×)	0.34
WSE	4427.8	7,552,081	0.007	28,293	-	615	432,792	0.039	5583	3.05	20.7 (214×)	409,327 (18.4×)	0.0042 (9.75×)	1241 (4.50×)	0.13
GPU	716.83	853,118	0.002	2317	-	41.0	81,334	0.0095	419	1.41	3.24 (221×)	98,395 (8.67×)	0.0019 (5.00×)	215 (1.95×)	0.08
PoP	106.87	188,295	0.0012	354	-	15.4	36,948	0.0038	86	2.55	1.43 (74.7×)	35,779 (5.26×)	0.0011 (3.45×)	58 (1.48×)	0.22

time of our algorithm remains to be far shorter, owing to the dimension reduction of the entities to be meshed from 3-D to 2-D. The results from our algorithm are much more precise as well, further demonstrating its superiority.

TABLE III

SUMMARIZED RESULTS OF THE COMBINATION OF OUR MESHING ALGORITHM WITH THE PROPOSED 2-D LAYERWISE PLATE FORMULATION. THE IMPROVEMENTS IN CONVERGENCE TIME OVER THE 3-D FORMULATION ARE PRESENTED AS WELL.

Case	Convergence time (s)	Improvement over 3-D	Error (%)
ATPlace 1	12.15	12.6×	1.36
ATPlace 2	12.60	9.92×	1.54
ATPlace 2	11.95	11.4×	0.99
WSE	248.4	5.00×	0.20
GPU	32.61	6.59×	0.67
PoP	15.47	3.75×	2.78

## 2) Combination with the 2-D layerwise plate theory:

We combine our meshing algorithm with the proposed 2-D layerwise plate formulation to further improve the convergence of the solver. The additional reduction in convergence time and the errors in results are concluded in Table III. A further speed-up from 3.75× to 12.6× is achieved compared with the 3-D formulation owing to the improvement of convergence. The errors of the 2-D theory are larger than those of the 3-D one, which is reasonable because the displacements are simplified in the 2-D theory. However, this kind of error is controllable, which is less than 3% for all test cases.

The warpage contours obtained from the ground truth simulation, our meshing algorithm, and its combination with the 2-D layerwise plate theory for the test cases are shown in Fig. 5. It is clear that there are only marginal differences in the results. The bottom surfaces of test cases ATPlace, WSE, and PoP are concave, because we add a Cu heat-spreader on top of the package for these cases.

## VI. CONCLUSION

In this paper, we propose a fast sweep hexahedral mesh generation algorithm for warpage simulation of complex 2.5-D/3-D IC structures composed of laminated thin layers. Our algorithm divides the structures automatically through an efficient projection method without introducing any unnecessary boundaries. Based on the generated high-quality

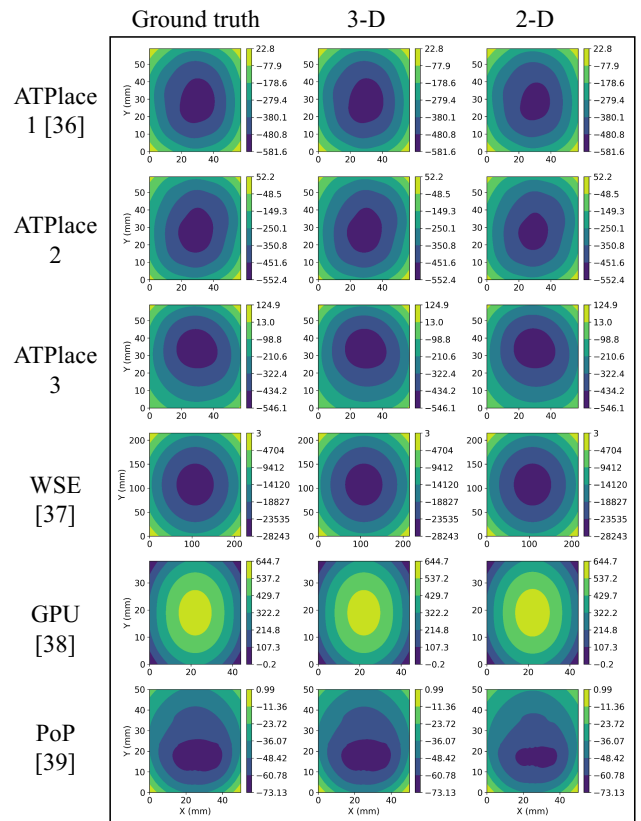


Fig. 5. Warpage contours obtained from the ground truth simulation, our meshing algorithm with the 3-D formulation, and our meshing algorithm with the 2-D formulation. The lower-left corner is chosen as the reference point ( $w = 0$ ) for comparison. (unit:  $\mu\text{m}$ )

sweep hexahedral mesh, we further propose a modified 2-D layerwise plate formulation to eliminate the out-of-plane stiffness terms and improve the condition number of the system stiffness matrix. Experimental results exhibit that, compared with Ansys Workbench, our meshing algorithm can either reduce the meshing time (74.7× to 221×) and the number of mesh nodes (5.26× to 18.4×), or improve the mesh quality (3.45× to 9.75×) and reduce the convergence time of the solver (1.48× to 4.50×), with errors < 0.5%. When using the proposed 2-D layerwise plate formulation instead of the 3-D one, a 3.75× to 12.6× speedup in convergence is further achieved with errors < 3%.

## REFERENCES

- [1] Y.-W. Chang, "Physical design challenges in modern heterogeneous integration;" in *Proceedings of the 2024 International Symposium on Physical Design*, 2024, pp. 125–134.
- [2] V. Moroz, X. Xu, A. Svizhenko, X.-W. Lin, S. Popov, H. Sheng, and K. Larsen, "3dic system-technology co-optimization with a focus on the interplay of thermal, power, timing, and stress effects," in *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2024, pp. 1–2.
- [3] H. Jang, B. S. Ma, S. Kim, J.-H. Lee, S. Myung, Y.-J. Lee, I. Huh, S. Kim, M. C. Park, N. Jeong *et al.*, "Design-aware full-chip warpage modeling for stco: Bridging reliability and design for a new era of advanced systems," in *2025 Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2025, pp. 1–3.
- [4] J. H. Lau, M. Li, L. Yang, M. Li, I. Xu, T. Chen, S. Chen, Q. X. Yong, J. P. Madhukumar, W. Kai *et al.*, "Warpage measurements and characterizations of fan-out wafer-level packaging with large chips and multiple redistributed layers," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 8, no. 10, pp. 1729–1737, 2018.
- [5] T.-C. Chiu and E.-Y. Yeh, "Warpage simulation for the reconstituted wafer used in fan-out wafer level packaging," *Microelectronics Reliability*, vol. 80, pp. 14–23, 2018.
- [6] T.-Y. Chen, K. Chuang, W. Hung, T.-S. Lin, and Y.-M. Chen, "Package-system thermal modeling and new material," in *2024 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2024, pp. 1–2.
- [7] E. Suhir, "Stresses in adhesively bonded bi-material assemblies used in electronic packaging," *MRS Online Proceedings Library*, vol. 72, no. 1, pp. 133–138, 1986.
- [8] "Ansys Workbench," <https://www.ansys.com/products/ansys-workbench>.
- [9] M. G. Larson and F. Bengzon, *The finite element method: theory, implementation, and applications*. Springer Science & Business Media, 2013, vol. 10.
- [10] P. Chen, D. Niu, D. Zhang, W. Wang, D. Xie, Z. Jin, W. W. Xing, and L. He, "Neuralmesh: Neural network for fem mesh generation in 2.5d/3d chiplet thermal simulation," in *Proceedings of the 62st ACM/IEEE Design Automation Conference*, 2025, pp. 1–7.
- [11] M. A. Scott, M. N. Earp, S. E. Benzley, and M. B. Stephenson, "Adaptive sweeping techniques," in *Proceedings of the 14th International Meshing Roundtable*. Springer, 2005, pp. 417–432.
- [12] M. Lai, S. Benzley, and D. White, "Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping," *International Journal for Numerical Methods in Engineering*, vol. 49, no. 1-2, pp. 261–275, 2000.
- [13] D. R. White, S. Saigal, and S. J. Owen, "Cesweep: automatic decomposition of multi-sweep volumes," *Engineering with computers*, vol. 20, no. 3, pp. 222–236, 2004.
- [14] N. Pietroni, M. Campen, A. Sheffer, G. Cherchi, D. Bommes, X. Gao, R. Scateni, F. Ledoux, J. Remacle, and M. Livesu, "Hex-mesh generation and processing: a survey," *ACM transactions on graphics*, vol. 42, no. 2, pp. 1–44, 2022.
- [15] S.-Y. Lo, M. Liu, and Y.-W. Chang, "Efficient high-fidelity two-dimensional warpage modeling for advanced packaging analysis," in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, 2024, pp. 1–9.
- [16] D.-H. Kim, S.-J. Joo, D.-O. Kwak, and H.-S. Kim, "Warpage simulation of a multilayer printed circuit board and microelectronic package using the anisotropic viscoelastic shell modeling technique that considers the initial warpage," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 6, no. 11, pp. 1667–1676, 2016.
- [17] M. Wang and B. Wells, "Substrate trace modeling for package warpage simulation," in *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*. IEEE, 2016, pp. 516–523.
- [18] V.-L. Pham, H. Wang, J. Xu, J. Wang, S. Park, and C. Singh, "A study of substrate models and its effect on package warpage prediction," in *2019 IEEE 69th Electronic Components and Technology Conference (ECTC)*. IEEE, 2019, pp. 1130–1139.
- [19] P. Chen, Z. Ji, Y. Liu, C. Wu, N. Ye, and H. Takiar, "Warpage prediction methodology of extremely thin package," in *2017 IEEE 67th Electronic Components and Technology Conference (ECTC)*. IEEE, 2017, pp. 2080–2085.
- [20] J. N. Reddy, *Mechanics of laminated composite plates and shells: theory and analysis*. CRC press, 2003.
- [21] T. Zhu, *et al.*, <https://github.com/txzhu2050/M2Mechanic>.
- [22] M. Tsai, C. Hsu, and C. Han, "A note on suhir's solution of thermal stresses for a die-substrate assembly," *Journal of Electronic Packaging*, vol. 126, no. 1, pp. 115–119, 2004.
- [23] B. Vandavelde, E. Beyne, D. Vandepitte, and M. Baelmans, "Analytical thermo-mechanical model for non-underfilled area array flip chip assemblies," *J. Electron. Packag.*, vol. 126, no. 3, pp. 351–358, 2004.
- [24] M.-Y. Tsai and Y.-W. Wang, "A theoretical solution for thermal warpage of flip-chip packages," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 10, no. 1, pp. 72–78, 2019.
- [25] Q. Wang, T. Zhu, Y. Lin, R. Wang, and R. Huang, "Atsim3. 5d: A multiscale thermal simulator for 3.5 d-ic systems based on nonlinear multigrid method," in *2025 International Symposium of Electronics Design Automation (ISED)*. IEEE, 2025, pp. 595–601.
- [26] K. Liew, Z. Pan, and L. Zhang, "An overview of layerwise theories for composite laminates and structures: Development, numerical implementation and application," *Composite Structures*, vol. 216, pp. 240–259, 2019.
- [27] L. Iurlaro, M. Gherlone, M. Di Sciuva, and A. Tessler, "Assessment of the refined zigzag theory for bending, vibration, and buckling of sandwich plates: a comparative study of different theories," *Composite Structures*, vol. 106, pp. 777–792, 2013.
- [28] F. Auricchio and E. Sacco, "Refined first-order shear deformation theory models for composite laminates," *Journal of applied mechanics*, vol. 70, no. 3, pp. 381–390, 2003.
- [29] T. Zhu, Q. Wang, Y. Lin, R. Wang, and R. Huang, "More-stress: Model order reduction based efficient numerical algorithm for thermal stress simulation of tsv arrays in 2.5 d/3d ic," in *2025 Design, Automation & Test in Europe Conference (DATE)*. IEEE, 2025, pp. 1–7.
- [30] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities," *International journal for numerical methods in engineering*, vol. 79, no. 11, pp. 1309–1331, 2009.
- [31] I. A. Baratta, J. P. Dean, J. S. Dokken, M. Habera, J. HALE, C. N. Richardson, M. E. Rognes, M. W. Scroggs, N. Sime, and G. N. Wells, "Dolfinx: the next generation fenics problem solving environment," 2023.
- [32] M. W. Scroggs, J. S. Dokken, C. N. Richardson, and G. N. Wells, "Construction of arbitrary order finite element degree-of-freedom maps on polygonal and polyhedral cell meshes," *ACM Transactions on Mathematical Software (TOMS)*, vol. 48, no. 2, pp. 1–23, 2022.
- [33] M. W. Scroggs, I. A. Baratta, C. N. Richardson, and G. N. Wells, "Basix: a runtime element basis evaluation library," *Journal of Open Source Software*, vol. 7, no. 73, p. 3982, 2022.
- [34] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells, "Unified form language: A domain-specific language for weak formulations of partial differential equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 40, no. 2, pp. 1–37, 2014.
- [35] L. D. Dalcin, R. R. Paz, P. A. Kler, and A. Cosimo, "Parallel distributed computing using python," *Advances in Water Resources*, vol. 34, no. 9, pp. 1124 – 1139, 2011, new Computational Methods and Software Tools.
- [36] Q. Wang, X. Li, T. Jia, Y. Lin, R. Wang, and R. Huang, "Atplace2. 5d: Analytical thermal-aware chiplet placement framework for large-scale 2.5 d-ic," in *Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*, 2024, pp. 1–9.
- [37] S. Lie, "Cerebras architecture deep dive: First look inside the hw/sw co-design for deep learning: Cerebras systems," in *2022 IEEE Hot Chips 34 Symposium (HCS)*. IEEE Computer Society, 2022, pp. 1–34.
- [38] T. NVIDIA, "Nvidia tesla v100 gpu architecture," *Tesla NVIDIA*, 2017.
- [39] A. Yoshida, J. Taniguchi, K. Murata, M. Kada, Y. Yamamoto, Y. Takagi, T. Notomi, and A. Fujita, "A study on package stacking process for package-on-package (pop)," in *56th Electronic Components and Technology Conference 2006*. IEEE, 2006, pp. 6–pp.
- [40] P. M. Knupp, "Algebraic mesh quality metrics," *SIAM journal on scientific computing*, vol. 23, no. 1, pp. 193–218, 2001.
- [41] T. Sorgente, S. Biasotti, G. Manzini, and M. Spagnuolo, "A survey of indicators for mesh quality assessment," in *Computer graphics forum*, vol. 42, no. 2. Wiley Online Library, 2023, pp. 461–483.