Topological optimization is a mathematical method that optimizes the interior material distribution of a 3D model to meet specific demands on its physical properties, such as resistance to external loads. It has broad applications in high-tech industries, such as aerospace [1], automotive [2], architecture [3] and bio-engineering [4]. However, the manufacture of topologically optimized structures by using traditional techniques such as Computer Numerical Control (CNC) is challenging due to the complex geometry of the optimized structures. In the past decade, additive manufacturing (3D printing) has emerged as a revolutionary manufacturing technology that enables the convenient fabrication of complex structures. Thus there is a significant interest in integrating topological optimization and additive manufacturing to design and fabricate various optimized structures.

In this paper, we focus on the design and fabrication of porous structures with additive manufacturing [5]. Porous structures offer some desirable properties such as small weight-to-stiffness ratio [6], good buffering protection [7], and excellent thermal conductivity [8]. They are widely used in various fields such as modeling human bones and dentures in medical engineering, fibrous materials in civil engineering, metal foam, insulating materials, sound-absorbing coatings and heat dissipation materials in material engineering. Triply periodic minimal surfaces (TPMSs) [9] are often utilized as a foundational element in constructing porous structures, as they are mathematically defined as minimal surfaces. They are widely used in structural optimization due to their full connections, high smoothness, non-self-intersections and quasi-self-supporting properties [10].

Previous investigations of TPMSs have mainly focused on structure optimization for lightweight [10] or mechanical properties [11,12]. However, in many cases, the optimized structures cannot be directly printed by laser-based additive manufacturing techniques such as Stereolithography Apparatus (SLA) and Fused Deposition Modeling (FDM) due to overhang regions. Additional supporting material must be added to support these regions during the printing process. After the model is printed, then the supporting material should be removed. This process causes a waste of materials and produces artifacts on the surface of the printed model. A meticulous post-processing operation is crucial to refine the surface of the printed model. Additionally, the minimum thickness of the structure is a crucial consideration. If the structure is too thin, the printer may not be able to print it correctly, leading to a significant impact on the quality of the printed model.

The goal of this paper is to utilize topology optimization to design a self-supporting porous structure based on TPMS while ensuring its stiffness. A structure is considered to be self-supported...
if every point on the surface of the structure has an overhang angle smaller than a prescribed threshold [13]. To achieve this, we use TPMS-based geometry to provide an explicit expression for the normal of the boundary surface, and we build a novel global inequality constraint that guarantees self-supporting structures. We also use a weighting scheme to prevent isolated non-self-supporting points from contributing to a global constraint. Furthermore, we discretize the TPMS parameters to obtain design variables, resulting in smooth optimized designs without sharp features. A minimal thickness constraint is also imposed by restricting the design variables in some suitable intervals to ensure the printability of the models. By minimizing the compliance of the structure with the above constraints through a two-stage optimization strategy, our method generates printable and mechanically optimized porous structures effectively. We also minimize the number of design variables through collocation points, which further increases the efficiency of our approach. Fig. 1 illustrates some 3D printing models generated by our algorithm.

In summary, the specific contributions of the current paper are as follows:

• A topology optimization framework is developed to generate smooth porous structures with a high self-supporting ratio based on TPMS. We formulated a new global constraint while a weighting scheme is utilized to filter out small areas that can be printed.
• Establish the relationship between minimum length constraints and the shell thickness of TPMS to ensure the printability of porous structures.
• Apply several acceleration methods, including super element strategy, multigrid algorithm, and GPU-based implementation, have been applied to efficiently solve the topology optimization problem.

2. Related work

2.1. Topology optimization

Topology optimization (TO) is an algorithmic process that optimizes the material distribution in the design domain according to the predefined boundary conditions and constraints. Many methods have been developed to solve the TO problem, such as the homogenization method [14], solid isotropic material with penalization (SIMP) [15], level set method [16], topology derivative method [17], and moving morphable components (MMC) [18]. Among them, the SIMP method is a very simple and popular method that will be adopted in the current work.

2.2. TPMS-based structure optimization

TPMSs have been taken as an efficient representation of porous structures. Due to the implicit representations of TPMSs, the properties of porous structures such as porosity and volume fraction can be easily achieved by setting suitable parameters of TPMSs [19,20]. Complex operations such as boolean, modification, and convolutions can be applied to TPMSs [21,22].

In order to mimic natural porous structures and fulfill the demands of diverse applications, graded TPMS [23,24], heterogeneous TPMS [25,26], multi-scale TPMS [10,27] were designed. In addition, TPMSs have been widely used in the design of articular scaffolds due to their large surface areas and fully connected structures [23,28].

In recent years, TPMSs have gradually attracted the attention from the community of computer graphics and computer-aided design. Zhu et al. [29] designed a novel TPMS-based porous scaffold with optimized thickness to tune both the mechanical and biological properties. VISHWANATH et al. [30] applied 3D convolutional neural network (CNN) models for the topology optimization of a unit cell based on TPMS. Li et al. [31] proposed an optimization strategy for designing models of gyroid-based functionally graded cellular structures. Hu et al. [10] constructed a multi-scale porous structure by adding parameters to control the period and volume of the TPMSs. Later they proposed an efficient TPMS-based structural optimization method without remeshing operation in finite element computation. Wang et al. [32] adopted the optimization framework for the problem of steady-state heat conduction. However, these works do not take manufacturing constraints into consideration, and the resulting porous structures are hard to manufacture without additional supporting material.

2.3. Fabrication requirements

Structures generated by topology optimization are typically complex in geometry, with large overhang regions and small branches. To fabricate these structures using additive manufacturing, additional overhang constraints and minimal length constraints should be added. However, adding supporting material leads to material waste, longer fabrication time, and lower surface quality. Therefore, minimizing overhang regions and detecting and removing small branches has become an area of interest.

Many efforts have been proposed to reduce additional support materials in additive manufacturing, including optimizing the direction of fabrication [34,35] and modifying the geometries of the structures [36,37]. Paul and Anand [38] proposed a strategy to find an optimal printing direction that minimizes the volume
of support structures. Qian [39] introduced constraints on the boundary shape, such as undercut control and minimal overhang angle control, that are helpful in reducing the need for support structures. Wang and Qian [40] proposed a method that optimizes both the build orientation and density field to satisfy overhang angle constraints for self-supporting parts. Allaire et al. [41] presented a new mechanical constraint functional that mimics the layer-by-layer construction process used in additive manufacturing technologies. Wu et al. [13] proposed a method to generate infill structures that satisfy manufacturing requirements on overhang-angle and wall-thickness, but is limited to rhombic cells. Mizendehdel and Suresh [37] proposed a method for improving computational efficiency. In our paper, we establish topology optimization. Later Guest et al. [47] and Zhou et al. [48] reported that the topology of TPMSs is satisfactorily reproduced by truncating the series to the leading term, giving the following nodal approximations of P, G and D surfaces:

\[
\phi_0(r) = \cos(2\pi t_x x) + \cos(2\pi t_y y) + \cos(2\pi t_z z) = C, \\
\phi_1(r) = \sin(2\pi t_x x)\cos(2\pi t_y y) + \sin(2\pi t_z z) \cos(2\pi t_x x) + \sin(2\pi t_y y)\cos(2\pi t_z z) = C, \\
\phi_2(r) = \cos(2\pi t_x x)\cos(2\pi t_y y)\cos(2\pi t_z z) - \sin(2\pi t_x x)\sin(2\pi t_y y)\sin(2\pi t_z z) = C,
\]

where \( t_x, t_y, t_z \) represent periodic parameters along the x-axis, y-axis, and z-axis, respectively, and C is a nonzero constant.

3. Geometric representations of porous structures

3.1. Representations of TPMS

TPMSs are minimal surfaces that have a crystalline structure and are triply periodic in the sense that they repeat themselves in three independent directions. TPMSs have two main representations—Enneper-Weierstrass parametric representation and Fourier series representation.

According to the Enneper-Weierstrass parametric representation, TPMSs can be expressed analytically as [49,50]:

\[
\begin{align*}
x &= \text{Re} \left( e^{i\theta} \int_{s_0}^{s} (1 - t^2) R(t) \, dt \right), \\
y &= \text{Re} \left( e^{i\theta} \int_{s_0}^{s} i (1 + t^2) R(t) \, dt \right), \\
z &= \text{Re} \left( e^{i\theta} \int_{s_0}^{s} 2iR(t) \, dt \right),
\end{align*}
\]

where \( i \) is the imaginary unit, and \( R(t) \) is a complex function. Thus, the Cartesian coordinates of any point on a minimal surface are expressed as the real parts (Re) of some complex (cuvilinear) integrals, evaluated in the complex plane from a fixed point \( s_0 \) to a variable point \( s \). The Weierstrass function \( R(t) \) completely specifies the local differential geometry of the surface and guarantees that the described surface is a minimal surface.

However, only a few kinds of TPMSs can be generated from the parametric function (1). An alternative mathematical representation for TPMSs can be approximated by the periodic nodal surface defined in terms of Fourier series [19]:

\[
\psi(r) = \sum_{k=1}^{K} A_k \cos \left[ \frac{2\pi (\mathbf{h}_k \cdot \mathbf{r})}{\lambda_k} + P_k \right] = C,
\]

in which \( \mathbf{r} = (x, y, z) \in \mathbb{R}^3 \) is a location vector, \( A_k \) is the amplitude, \( \mathbf{h}_k \) is the kth reciprocal lattice vectors, \( \lambda_k \) is the period factor, \( P_k \) is the function phase and \( C \) is a constant. Earlier methods [51] reported that the topology of TPMSs is satisfactorily reproduced by truncating the series to the leading term, giving the following nodal approximations of P, G and D surfaces:

\[
\begin{align*}
\phi_0(r) &= \cos(2\pi t_x x) + \cos(2\pi t_y y) + \cos(2\pi t_z z) = C, \\
\phi_1(r) &= \sin(2\pi t_x x)\cos(2\pi t_y y) + \sin(2\pi t_z z) \cos(2\pi t_x x) + \sin(2\pi t_y y)\cos(2\pi t_z z) = C, \\
\phi_2(r) &= \cos(2\pi t_x x)\cos(2\pi t_y y)\cos(2\pi t_z z) - \sin(2\pi t_x x)\sin(2\pi t_y y)\sin(2\pi t_z z) = C,
\end{align*}
\]

where \( t_x, t_y, t_z \) represent periodic parameters along the x-axis, y-axis, and z-axis, respectively, and \( C \) is a nonzero constant.

3.2. TPMS-based porous structures

Signed distance fields are effective representations of porous structures. In this paper, we assume that positive and negative distances determine the inside and outside of a structure, respectively. A TPMS-based porous structure can be described by the intersection of two solids that are defined by two signed distance fields:

\[
\begin{align*}
\phi_1 &= \psi + C, \\
\phi_2 &= C - \psi, \\
\phi_3 &= \min(\phi_1, \phi_2),
\end{align*}
\]

where \( \phi_1 \) and \( \phi_2 \) represent two signed distance fields determined by the TPMS function \( \psi \), and \( C \) is the physical offset which measures the algebraic distance of two surfaces \( \phi_i = 0 \) (i = 1 or 2) and \( \phi = 0 \). The porous structure is defined by \( \phi_i > 0 \) which is the intersection of two solids \( \phi_1 > 0 \) and \( \phi_2 > 0 \). In order to calculate the derivatives of \( \phi_i(r) \) at any point \( r \), we introduce a smooth approximation of \( \phi_i(r) \) by:

\[
\phi_i(r) = \phi_1(r) + \phi_2(r) - \sqrt{\phi_1^2(r) + \phi_2^2(r)},
\]

\[
= 2C - \sqrt{2C^2 + 2\psi^2}.
\]

An example is illustrated in Fig. 2, where \( \psi \) is the P surface with \( t_x = t_y = t_z = 1 \) and \( C = 0.427 \). Fig. 3 show two sequences of porous structures for different parameter values of \( t_x, t_y, t_z \) and \( C \). Variations of porosity in one unit can be clearly observed.

In our work, a TPMS-based porous structure is optimized inside a given model \( \Omega_{\text{Model}} \):

\[
\phi_{\text{opt}} = \min(\phi_3, \phi_{\text{Model}}),
\]
where $\phi_{\text{Model}}$ is the signed distance field of the shape $\Omega_{\text{Model}}$. The final porous structure that fills in the interior of $\Omega_{\text{Model}}$ can be presented as $\phi_{\text{opt}} > 0$.

### 3.3. Design variable distribution function

In the minimal surface representations (3), $t_x(r)$, $t_y(r)$, $t_z(r)$ and $C(r)$ are free parameters that depend on the location vector $r$ and determine the shapes of the minimal surfaces. They are the design variables of our TO problem. In order to efficiently represent and compute these variables, we interpolate them using radial basis functions at a set of collocation points $\{r_i\}_{i=1}^n$ that are uniformly arranged in space [52]:

$$t_x(r) = \sum_{k=1}^n \alpha_k f(\|r - r_k\|) + \sum_{i=1}^q \beta_i p_i(r),$$

where $f(x) = x^2 \log(x)$ is the polyharmonic radial basis function, $n$ is the number of the collocation points, $q$ is the number of polynomial basis functions, $p_i(r)$, $l = 1, \ldots, q$ are polynomial basis functions.

There are $q$ additional degrees of freedom in (7), and they are fixed by $q$ additional homogeneous equations. This augmented system has a unique solution:

$$\sum_{k=1}^n \alpha_k f(\|r_i - r_k\|) + \sum_{i=1}^q \beta_i p_i(r_k) = t_x(r_i), \quad 1 \leq i \leq n;$$

$$\sum_{k=1}^n \alpha_k p_i(r_k) = 0, \quad 1 \leq l \leq q;$$

where $t_x(r_i)$ is the $x$-axis period variable at the $i$th collocation point. We solve the linear system of Eqs. (8) by the known $\{t_x(r_i)\}_{i=1}^n$ in the previous iteration to compute the unknown coefficients $\{\alpha_k\}_{k=1}^n$ and $\{\beta_i\}_{i=1}^q$. Thus the expression for $t_x(r)$ is obtained. The variable distribution functions of $t_x(r)$, $t_y(r)$ and $C(r)$ can be computed in a similar way.

### 4. Methods

#### 4.1. Problem and formulation

**Problem.** The goal of the current work is to find an interior material distribution $\phi_{\text{opt}}$ with self-supporting requirement and maximum stiffness under predefined volume fraction in a given design domain $\Omega_M$.

**Formulation.** This problem can be formulated as the following optimization problem:

$$\begin{align*}
(P) & \quad \min_{t_x(r), t_y(r), t_z(r), C(r)} \quad I = U^T F, \\
\text{s.t.} & \quad V(t_x(r), t_y(r), t_z(r), C(r)) \leq V_0, \\
& \quad \alpha \leq \alpha_0, \\
& \quad KU = F, \\
& \quad t_{\text{min}} \leq t_x(r), t_y(r), t_z(r) \leq t_{\text{max}}, \\
& \quad C_{\text{min}} \leq C(r) \leq C_{\text{max}}.
\end{align*}$$

where $t_x(r), t_y(r)$ and $t_z(r)$ denote the periodic variables along $x, y, z$ axes respectively, $C(r)$ represents the thickness variable which satisfies the printing minimal thickness constraint, and $V(t_x(r), t_y(r), t_z(r), C(r))$ and $V_0$ represent the volume fraction of the optimized structure and a predefined threshold. The inequality (11) is the overhang constraints defined on surface points by the printing tolerance angle $\alpha_0$, and (12) is the static equilibrium equation. The inequalities (13) and (14) are constraints for the design variables $t_x(r), t_y(r), t_z(r), C(r)$.

The value $\rho_i$ of the sign distance field $\phi_{\text{opt}}$ at $r_i \in \Omega_M$ is calculated by (5) as:

$$\rho_i = 2C - \sqrt{2C^2 + 2\phi^2(t_x(r_i), t_y(r_i), t_z(r_i))},$$

Thus, the density distribution $\rho_i$ can be truncated by $\rho_{i0}$:

$$\bar{\rho}_i = \begin{cases} 0, & \rho_i < 0; \\
1, & \rho_i \geq 0. 
\end{cases}$$

To obtain a differentiable parameter $\bar{\rho}_i$, (16) is replaced by its continuous approximation [18]:

$$\bar{\rho}_i = H_{\xi}(\rho_i) = \begin{cases}
1, & \rho_i > \xi; \\
\frac{1}{2} \left(1 + \frac{\rho_i}{\xi^2}\right) + \frac{1}{2}, & -\xi \leq \rho_i \leq \xi; \\
0, & \rho_i < -\xi,
\end{cases}$$

where $\xi$ is the parameter that controls the magnitude of regularization. Based on our experience, we typically recommend $\xi$ to be within the range of $[0.04, 0.15]$. Within this range, the optimized structures exhibit similar physical properties, and the choice of $\xi$ can be adjusted depending on the specific design requirements. In our implementation, we set $\xi = 0.15$.

#### 4.2. Fabrication constraints

4.2.1. Overhang constraints

The main goal of current work is to optimize the interior structure of an object such that it is self-supported as much as possible. Thus we need to handle overhang constraints.

Overhang regions in additive manufacturing are defined by the subtended angle [37] which is the angle between the printing direction $n_p$ and the normal vector of a boundary point, as shown in Fig. 4(a). A surface point is considered unprintable if the subtended angle $\alpha$ exceeds the user-defined threshold $\alpha_0 = \frac{\pi}{2}$. Boundary points with $\alpha \leq \alpha_0$ are considered self-supported (marked by blue), and otherwise the surface points (marked by red) are overhanging points which require material support, as illustrated in Fig. 4(b).

Explicit overhang constraints. In our case, TPMS is defined in the interior of an object and may stretch and deform due to varying local frequencies and amplitudes. This may yield overhang parts during fabrication, as illustrated in Fig. 4(c). In this work, the surface points $r^* = \{r_t^*\}_{t=1}^M$ of the TPMS-based porous structure are extracted by the marching cube method [53] from $\phi_{\text{opt}}$, as shown in Fig. 4(d). Where $M$ is the number of surface.
The normal of each point \( \mathbf{n}_g(r_{k}^*) \) can be easily computed according to the implicit representation \( \phi_{opt} \) in Eq. (5). In order to be self-supported, a surface point \( r_{k}^* \) must obey the rule:

\[
\alpha(r_{k}^*) \leq \alpha_0, \quad \forall k;
\]

or equivalently,

\[
A(r_{k}^*) = \frac{\cos(\alpha(r_{k}^*))}{\cos(\alpha_0)} = \frac{\mathbf{n}_g(r_{k}^*) \cdot \mathbf{n}_p}{||\mathbf{n}_g(r_{k}^*)|| \cdot ||\mathbf{n}_p|| \cdot \cos \alpha_0} \leq 1, \quad \forall k;
\]

where \( \mathbf{n}_p = (n_{px}, n_{py}, n_{pz}) \) is the normal vector of the boundary surface at point \( r_{k}^* \), \( \mathbf{n}_p \) is the printing direction, and we set it to be the positive direction of the \( z \)-axis \((0, 0, 1)\). Notice that Eq. (19) clearly holds at surface point whose \( z \)-component of the normal vector is non-negative, i.e., \( n_{pz} \geq 0 \). Thus, we only need to consider the constraints (19) at the surface points \( r_{k}^* = \{r_{k}^*\}_{k=1}^{N} \) with \( n_{pz} < 0 \), where \( N \) is the number of surface points with a negative \( z \)-component of the normal vector. However, it is very inefficient to directly impose the overhang constraints (19) during the optimization process. The reasons are twofold:

- The overhang constraints (19) are nonlinear. A large number of nonlinear constraints causes the optimization very inefficient.
- Imposing the overhang constraints on every point of the boundary surface is too strict. In fact, additional support structures are not needed in small overhang regions. Therefore, the overhang constraints corresponding to these points can be removed. Fig. 5 illustrates removable overhang regions in a model.

**Modified overhang constraints.** To overcome the first challenge, we integrate a large number of overhang constraints (19) into one global constraint:

\[
A_k = \frac{1}{N} \sum_{k=1}^{N} g \left( \frac{n_{pz}(r_{k}^*)}{||\mathbf{n}_g(r_{k}^*)|| \cdot \cos \alpha_0} \right) \leq 1 + \epsilon,
\]

where \( N \) is the number of surface points with a negative \( z \)-component of the normal vector, and \( \epsilon \) is a small positive value for the relaxation purpose. In our implementation, \( \epsilon = 0.05 \) is adopted. The function \( g(x) \) is employed to penalize the quantity \( A(r_{k}^*) \) when \( A(r_{k}^*) > 1 \) and is defined as:

\[
g(x) = \begin{cases} 0 & x \geq 1, \\ (x - 1)^2 + 1 & 1 < x < 1. \end{cases}
\]
balls show the corresponding neighborhood of points \( r^*_i \). Removable small overhang regions are colored green and actual overhang regions are colored orange as before. From the zoom views on the right, we can see that \( s(r^*_i) \) in the actual overhang region has a large value while \( s(r^*_i) \) is small in the green region.

Based on the definition of small area filtering function \( \tilde{w}(r^*) \), self-supporting points and overhang points on the removable overhang regions have a small weight. On one hand, this operation filters out small overhang areas that can be printed, and on the other hand, it helps to reduce the number of large overhang regions in the optimization process.

4.2.2. Minimal thickness constraints

Additive manufacturing requires a minimum thickness of printed objects. When the size of a part of the printed model is smaller than the printable size of a printer, the part of the model will be unprintable.

To incorporate minimal thickness constraints into the optimization process, Guest [47] suggested performing a thickness-dependent projection of the density field to prevent the occurrence of structures smaller than the prescribed minimal thickness. In our model, the thickness of a structure is determined by the period variables \( t_x(r), t_y(r) \) and the thickness variable \( C(r) \) while \( C(r) \) intuitively controls the thickness of the model as illustrated in Fig. 3. According to the homogenization method [14], we know that the stiffness of a periodic structure is scale-invariant in a unit volume. The effect of period variables is similar to scale transformation which also influences the thickness of the geometry. Fig. 8 shows that the minimal thickness varies with the variables \( C(r) \) and \( t_x(r) \) almost linearly. Therefore, we impose linear constraints (13) and (14) for the variables \( t_x(r), t_y(r) \) and \( C(r) \). The choice of the parameter bounds \( t_{\text{min}}, t_{\text{max}} \) and \( C_{\text{min}}, C_{\text{max}} \) will be discussed in Section 5.3.1.

4.3. Reformulation

To compute the optimization problem (\( P \)), we should change the variables \( t_x(r), t_y(r) \) and \( C(r) \) into discrete forms by using the technique in Section 3.3. Then the discrete optimization problem takes the form:

\[
\begin{align*}
\min_{t_x(r), t_y(r), C(r)} & \quad l = U^T F \\
\text{s.t.} & \quad V(t_x(r), t_y(r), C(r)) \leq V_0, \\
& \quad \hat{A}_x = \sum_{k=1}^{N} \tilde{w}(r^*_i) \left( \frac{n_{ax}(r^*_i)}{\|n_{ax}(r^*_i)\| \cos \alpha} \right) \leq 1 + \epsilon, \\
& \quad t_{\text{min}} \leq t_x(r), t_y(r), C(r) \leq t_{\text{max}}, \\
& \quad C_{\text{min}} \leq C(r) \leq C_{\text{max}},
\end{align*}
\]

where \( r^*_i \) are a set of collocation points that are evenly arranged in space. The numerical method to solve the optimization problem (\( Q \)) will be discussed in the next subsection.

4.4. Numerical implementation

The optimization problem (\( Q \)) can be solved by the Method of Moving Asymptotes (MMA) [54] which is widely used in topology optimization. MMA is an iterative approach to solve a general optimization problem. In each iteration step, a simple subproblem is generated to approximate the original problem, and then the subproblem is solved by the primal–dual interior point method. The iteration process terminates until the termination conditions are satisfied.

To apply MMA to the optimization problem (\( Q \)), we need to carry out finite element analysis of elasticity, i.e., to solve the static equilibrium Eq. (29) to update the variables in each iteration step. It involves computing the stiffness matrix \( K \) and solving a large and sparse linear system of equations \( KU = F \), which is quite time-consuming. In the following, we present strategies to solve the linear system more efficiently.

Computing stiffness matrix \( K \). To start the finite element analysis, we first generate a conservative voxelized model from the given model \( S_M \). This voxelized model can avoid remeshing in every iteration. We apply the super element strategy [55] by dividing the regular hexahedra elements (super elements) into smaller hexahedral elements (background elements) to improve the accuracy of FEM computation. The stiffness matrix is assembled on the background elements while the FEM computation is performed on the super element. As demonstrated in [55], the super element strategy provides better performance and convergence rates than traditional FEM calculations at the same resolution with background elements. To validate the effectiveness of our approach, we tested it on a cube model with
50 × 50 × 50 (4^3) elements (divide each super element into 4^3 background elements), as well as on models with resolutions of 200 × 200 × 200 elements. The results showed that this approach achieved a significant reduction in computation time, with runtimes of 10 s and 3560 s, respectively.

The stiffness matrix of super element i can be calculated as:

\[ K_i = \int_{\Omega_i} B^T D B dV = \sum_{j=1}^{n_{\Omega}} \int_{\Omega_j} B^T D B dV = \sum_{j=1}^{n_{\Omega}} E_{ij} B_j^T D_j B_j v_j = \sum_{j=1}^{n_{\Omega}} E_{ij} K_j^{ij}, \]

where \( \Omega_i \) denotes the i-th super element, \( \Omega_j \) denotes the j-th background element in the i-th super element, \( n_{\Omega} \) is the number of background elements in a super element, \( D \) is the strain matrix, \( B \) is the elastic matrix of \( \Omega_j \), \( D_0 \) is the elastic matrix of the solid material, \( v_j \) is the volume of \( \Omega_j \) and \( K_j^{ij} = B_j^T D_j B_j v_j \) is the approximate integral of \( B^T D B \) over \( \Omega_j \). In addition, \( E_{ij} \) is the Young’s modulus of the j-th background element in the i-th super element. In all the results presented in this paper, each super element is divided into 4^3 background elements.

The SIMP scheme [15] is adopted to model the Young’s modulus of the background element in the i-th super element, \( n_{\Omega} \) is the modulus of solid elements in a super element, and \( p \) is the penalization power. Based on our experiments, we observe that the optimized structures have similar physical properties in terms of compliance and self-supporting ratio. Furthermore, optimized structures using different penalty factors (3, 4, 5) show comparable results.

Note that \( K_j^{ij} \) is the same for different super elements, so we only need to compute it once in one super element and store them as a template \( K_j \). In each iteration, we just compute \( E_{ij} \) according to (33) to update \( K_j \).

**Acceleration.** Iterative methods such as Gauss–Seidel relaxation can be used to solve the linear system of (29) in principle. However, the convergence of such methods is generally slow, especially under high resolutions. Here we propose to employ the multigrid method [56] to accelerate the computation. But linear elasticity multigrid solvers for large problems are memory bound, meaning that they operate close to the theoretical memory bandwidth and further performance increases are difficult to achieve. To address this limitation, Dick et al. [57] proposed a GPU multigrid implementation which exploits the fast memory interface on such architectures. We adopt this multigrid technique based on GPU in our implementation. It has a parallelization scheme and matrix-free data structure, which significantly improves the performance compared to a CPU implementation. We have tested the effectiveness of the multigrid method in solving a set of fine element equations with a scale of 50^3. It appears that there is a significant reduction in the time required for each iteration, with the time decreasing from 1633 s to just a few seconds.

Our optimization process, which involves fine element analysis, sensitivity analysis, and MMA, is implemented on the GPU using the CUDA parallel programming API. Based on our experimental tests, GPU computing reduced the runtime by approximately 90% and the memory consumption by around 80% compared with CPU computing.

**Volume discretization.** In order to solve the optimization problem (27), the volume in the constraint (27) should be calculated. This can be done by discretizing the volume as:

\[ \frac{1}{8} \sum_{j=1}^{N_{\Omega}} \sum_{i=1}^{8} \nu_i^j \rho_i^j, \]

where \( \rho_i^j \) is the density at node point of the j-th background element and is computed according to the Eqs. (15)–(17), and \( N_{\Omega} \) is the total number of background elements.

**Two-stage optimization.** To expedite the optimization process for the problem (27) while using MMA, which typically exhibits slow convergence and longer optimization time, we divide it into two stages: period optimization and thickness optimization. During the period optimization stage, we keep the variables \( C(r_i) \) fixed and treat \( t_0(r_i) \), \( t_0(r_i) \), and \( t_0(r_i) \) as optimization variables. Conversely, during the thickness optimization stage, we fix \( t_0(r_i) \), \( t_0(r_i) \), and \( t_0(r_i) \) and treat \( C(r_i) \) as the optimization variables. This approach helps to achieve faster convergence and reduces the overall optimization time by half.

The input of our system is a model \( \Omega_m \) represented by a surface mesh. The system first computes a voxelized model. Then the voxel model is optimized with two-stage optimization. The termination criteria for each optimization stage is expressed as:

\[ \frac{\|I - I^\text{avg}\|}{I^\text{avg}} \leq 1 \times 10^{-3}, \]
\[ \frac{\|V - V^\text{avg}\|}{V^\text{avg}} \leq 1 \times 10^{-3}, \]
\[ \frac{\|\tilde{A}_g - \tilde{A}_g^\text{avg}\|}{\tilde{A}_g^\text{avg}} \leq 1 \times 10^{-3}, \]

where \( I \) and \( V \) are the compliance, the volume fraction and the function defined in (22) in the current iteration, respectively. \( I^\text{avg}, V^\text{avg} \) and \( \tilde{A}_g^\text{avg} \) are the average values of \( I \), \( V \) and \( \tilde{A}_g \) in the last ten iterations, respectively. Once the constraints have been satisfied and the two-stage optimization has converged, the final structure is generated.

4.5. Sensitivity analysis

In this subsection, we perform a sensitivity analysis of the objective function and the constraints with respect to the design variables.

**Period optimization.** In this process, we fix the thickness variables \( \{C_i\}_{i=1}^n \) and calculate the sensitivity analysis with respect to the period variables \( \{t_{si}\}_{i=1}^n \), \( \{t_{pi}\}_{i=1}^n \) and \( \{t_{2i}\}_{i=1}^n \).
of background elements in a super element, and $\rho_i^j$ represents density at the $i$th node point of the $j$th background element in the $j$th super element. $\rho_i^j$ is density at the $i$th node point of the $j$th background element in the model and $v_j$ is the volume of $j$th background element in the whole model. The sensitivity analysis with respect to $\{t_{x_i}\}_{i=1}^n$ and $\{t_{z_i}\}_{i=1}^n$ can be computed in a similar way.

The derivatives of $H_k$, $n_{g2}(r_i^k)$ and $||\hat{n}_g(r_i^k)||$ are computed by applying the chain rule, i.e.,

$$\frac{\partial H_k}{\partial t_{x_i}} = \frac{\partial H_k}{\partial \rho_i^j} \frac{\partial \rho_i^j}{\partial t_{x_i}},$$

$$\frac{\partial n_{g2}(r_i^k)}{\partial t_{x_i}} = \frac{\partial n_{g2}(r_i^k)}{\partial t_{x_i}},$$

$$\frac{\partial ||\hat{n}_g(r_i^k)||}{\partial t_{x_i}} = \frac{\partial ||\hat{n}_g(r_i^k)||}{\partial t_{x_i}},$$

where $\rho_i^j$ can be computed according to Eqs. (7) and (8).

5. Results and discussion

This section demonstrates the performance of our proposed structure optimization algorithm for different models. P surfaces are adopted as the TPMS representation in the porous structure optimization. The algorithm is implemented in C++ and runs on a PC with 3.20 GHz Intel(R) Core(TM)i7-8700 and 64 GB memory.

5.1. Pipeline

For a given model, we first generate a bounding box to enclose the input model and scale the bounding box such that the length of the longest edge of the bounding box is 1. The size of the bounding box will affect the selection of period parameters. In fact, if the bounding box is scaled by a factor $s$, then the period parameters will be scaled by $1/s$. After the bounding box is calculated, the system computes a voxelized model which is optimized with the two-stage optimization process introduced in the previous section. Once the constraints are satisfied and the two-stage optimization has converged, the final structure is generated. The overall optimization algorithm is described in Algorithm 1. Fig. 9 shows sliced views of a model and the physical properties (compliance, volume fraction and self-supporting ratio) along with the period and thickness optimization processes.

Algorithm 1 Fabrication friendly TPMS-based porous structure optimization

**Input:** A model $\Omega_d$, forces $F$, fixed area $F_T$  
**Output:** Period and thickness parameters

1. Generate a conservative voxelized domain from $\Omega_d$  
2. Generate a set of collocation points  
3. Initialize $\{t_{x_i}\}_{i=1}^n$, $\{t_{z_i}\}_{i=1}^n$, $\{t_{z_i}\}_{i=1}^n$, $\{C_i\}_{i=1}^n$, at the collocation points  
4. $h \leftarrow 1$  
5. while $h < max_iter$ & & !converged do  
   6. Update the combinatorial coefficients in (7)  
   7. Generate a set of surface points $\{r_i^m\}_{i=1}^M$ by the marching cube method  
   8. Solve the linear system $KU = F$ by the multigrid method to obtain $U$  
   9. Compute $I$, $V$, $\tilde{A}_g$ and perform sensitivity analysis $\partial I/\partial C_i$, $\partial V/\partial C_i$, $\partial \tilde{A}_g/\partial C_i$.  
   10. Update period variables $\{t_{x_i}\}_{i=1}^n$, $\{t_{z_i}\}_{i=1}^n$, $\{t_{z_i}\}_{i=1}^n$ by MMA  
   11. $h \leftarrow h + 1$  
   12. converged $\leftarrow$ check_converge_condition()  
   end while  
13. $h \leftarrow 1$  
14. while $h < max_iter$ & & !converged do  
15. Update the combinatorial coefficients in (7)  
16. Generate a set of surface points $\{r_i^m\}_{i=1}^M$ by the marching cube method  
17. Solve the linear system $KU = F$ by the multigrid method to obtain $U$  
18. Compute $I$, $V$, $\tilde{A}_g$ and perform sensitivity analysis $\partial I/\partial C_i$, $\partial V/\partial C_i$, $\partial \tilde{A}_g/\partial C_i$.  
19. Update the thickness variables $\{C_i\}_{i=1}^n$ by MMA  
20. converged $\leftarrow$ check_converge_condition()  
21. $h \leftarrow h + 1$  
22. end while

5.2. Performance

We choose four models–bunny, duck, kitten and molar as illustrated in Fig. 10(a) (the red arrows represent external forces imposed on the models) to test the performance of our structure optimization algorithm, and comparison results with two other models are also provided.

In order to show the effectiveness of our algorithm, we compare the results by our method with the results by two other models. One is the optimization model (Q) without the overhang constraint (28), which is similar to the model proposed in [32]. The other is the initial model we set for solving the problem (Q). We generate the initial model from the model with the same period variables $t_x = t_y = t_z = 4$, and then stretch the model in $z$ direction by setting $t_z \in [1, 1.4]$ in order to satisfy self-supporting constraints as much as possible. Finally, we adjust the thickness variable $C$ such that the volume constraint is satisfied. Fig. 10(b) shows the initial models generated by the above settings.

We compare the results by the following quantities: compliance, volume fraction, self-supporting ratio and computational time for four different models. All the models are tested with three different volume fractions: 35%, 50% and 65%. The statistics are summarized in Fig. 11. From the picture we can observe that our algorithm effectively reduces the compliance of the initial models, which means the optimized structures have better stiffness than the initial models. As for the comparison with the traditional topology optimization with only the volume constraint
Fig. 9. Physical properties (compliance, volume fraction and self-supporting ratio) of the molar model during the optimization process. Blue, green and red curves represent compliance, volume fraction and self-supporting ratio respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(e.g., the method in [32]), there is not much difference between the compliance. However, our method improves the average self-supporting ratio from 85% to 96%, which means that our models are more suitable for 3D printing. Furthermore, our method can also improve the printing accuracy, because 3D printers may change the structure a little bit when they encounter some unprintable areas. Figs. 10(d) and 10(f) show the results generated by the two methods respectively.

The computational performance of our algorithm is summarized in Table 1, where the second column lists the number of
elements used in the finite element simulation, the four column shows the number of iterations in the two-stage optimization, and the last column is the total computational time. As we can see from the table, our porous structure optimization algorithm costs a little more overhead than the optimization without the self-supporting constraint. Yet our algorithm improves the supporting ratio by ten percent. On the other hand, the computational cost of our algorithm depends largely on the number of finite elements used in the analysis, and for models discretized into millions of elements, the computational time is less than three hours.

5.3. Discussion

In this section, we discuss the initialization and bound selection of the design variables. The choice of the parameter in the small area filtering function is also discussed.

5.3.1. Design variables

Period and thickness variables initialization. For different choice of the period and thickness variables, the resulting TPMS-based porous structures are quite different, as illustrated in the previous picture. Fig. 3. Fig. 13 further depicts how the compliance and the volume fraction of the initial model depend on the change of the design variables. From the pictures we can see that, while the volume fraction is proportional to the design variables, the compliance monotonically decreases as the design variables increase.

According to our experiment, for a fixed volume fraction, the compliances and self-supporting ratios of the optimized structures have little difference for different initializations of the design variables. However, the optimized structures vary with different initializations. Generally, the larger the initial period variables, the more complex the internal structure of the optimized model, as shown in the middle column (with a volume fraction of 50%) and the right column (with a volume fraction of 65%) in Fig. 12. Table 2 gives the corresponding numerical results of compliances and self-supporting ratios of the optimized structures.

As we can observe, for a fixed volume fraction, the compliance and the self-supporting ratios of the optimized structures have little difference under different initializations of design variables. However, compliance depends mainly on volume fractions.

In [32], Hu et al. concluded that the minimal thickness of a TPMS is inversely proportional to the period variables and directly proportional to the thickness variable, and they gave an
Fig. 12. Comparison of different period and thickness variable initializations. The left column shows the initial models, the middle and right columns are the corresponding optimized structures by our algorithm with a volume fraction 50% and 65% respectively.

Table 2

<table>
<thead>
<tr>
<th>Initialization</th>
<th>Volume</th>
<th>Compliance</th>
<th>Self-supporting ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t₀, t₁, t₂, C)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top(2, 2, 1, 0.5)</td>
<td>50%</td>
<td>179.49</td>
<td>95.8%</td>
</tr>
<tr>
<td></td>
<td>65%</td>
<td>90.33</td>
<td>95.9%</td>
</tr>
<tr>
<td>Middle(4, 4, 1, 0.5)</td>
<td>50%</td>
<td>180.11</td>
<td>96.1%</td>
</tr>
<tr>
<td></td>
<td>65%</td>
<td>89.80</td>
<td>96.2%</td>
</tr>
<tr>
<td>Bottom(4, 4, 2, 0.5)</td>
<td>50%</td>
<td>181.01</td>
<td>95.9%</td>
</tr>
<tr>
<td></td>
<td>65%</td>
<td>90.76</td>
<td>95.7%</td>
</tr>
</tbody>
</table>

The left column shows the initial models, the middle and right columns are the corresponding optimized structures by our algorithm with a volume fraction 50% and 65% respectively. The volumetric compliance and self-supporting ratio are calculated for each initialization.

5.3.2. Small area filtering function

For an overhang region whose area is less than some tolerance \( A_{\text{tol}} \) (\( A_{\text{tol}} \) is set to be 16 mm \( \times \) 16 mm in our implementation), we do not have to add any supporting material in 3D printing, that is, the region is considered as self-supporting. For the overhang regions whose areas are larger than \( A_{\text{tol}} \), an overhang constraint must be imposed. For that purpose, a small area filtering function \( w(\mathbf{r}_k^*) \) defined in (24) is constructed. The weight function \( w(\mathbf{r}_k^*) \) has a relatively large value at overhang regions and a relatively small value at non-overhang regions.

To actually compute the function, we first generate a set of surface points of a porous structure by the marching cube method and label the overhang points. Then we compute the \( k \)-neighborhood points of every overhang point \( \mathbf{r}_k^* \) to get the overhang ratio \( s(\mathbf{r}_k^*) \) at this point. The radius \( \delta \) of the neighborhood ball is set to be 0.01 in our experiment. Finally the overhang function can be computed according to (24).

In the formula (24) of the small area filtering function \( w(\mathbf{r}_k^*) \), the parameter \( \eta \) is selected according to the threshold \( A_{\text{tol}} \). Fig. 14 shows the effect of the parameter \( \eta \) on the overhang regions for the same model. We find that \( \eta = 0.3 \) gives a better self-supporting result according to our numerical experiment. Fig. 14 depicts the over-hang regions of the Kitten model for different parameter values of \( \eta \).

5.4. Comparisons

Implementation details. Many previous works have considered overhang constraints in topology optimization for additive manufacturing such as [39,40,42]. We compare it with the representative work of Qian [39] in three aspects. Firstly, Qian et al. [39] took element density as design variables and density gradients to approximate boundary normals, while we used TPMS-based geometry by optimizing the period and thickness parameters. This allows us to generate complex and smooth geometries with high self-supporting ratios, and establish a correlation between print size and the thickness parameter in TPMS for better print quality. Secondly, we utilize acceleration techniques such as a super element strategy, a multigrid solver, and GPU computing.
Fig. 14. Parameter setting in the small area filtering function. The top row is the optimized TPMS-based porous structures. The models in the middle row show the removable overhang regions and the actual overhang regions colored in green and orange, respectively. The bottom row shows the final overhang regions in the models. (For interpretation of the reference to color in this figure legend, the reader is referred to the web version of this article.)

via the CUDA parallel programming API to handle millions of elements and achieve high-performance computing. This enables us to handle large-scale problems and obtain efficient solutions. Finally, our TPMS-based geometry generates smooth and connected porous structures instead of sharp features, which can occur in Qian [39]. This is essential for practical applications where smooth and connected geometries are preferred.

**Von mises stress.** There are numerous previous studies on optimizing infill structures using various methods, including support-free lattices [13,43], gyroids [31], and spheroids [44]. We have conducted a comparison of Von Mises stress, as shown in Fig. 15. The structures used for the comparison include honeycomb structures [58], uniform support-free infill structures [43], non-uniform support-free infill structures [13], as well as uniform TPMS-based support-free infill structures. The stress distribution analysis reveals that our model bears less stress under two volume fraction settings, and the maximum stress is the lowest among all the structures.

**Compression tests.** We conducted physical experiments to validate our results by fabricating the optimized structures using a Bambu X1-Carbon Combo 3D printer and subjecting it to compression testing. The models are printed using PLA, and we conduct compression testing on three sets of models: a kitten model, a duck model, and a bunny model with heights of 12 cm, 8 cm, and 8 cm, respectively. The MTS809 Axial/Torsional Test System is used to evaluate the strength of the printed models, as depicted in Fig. 16(a). The crosshead is moved at a constant speed of 2 mm/min, generating a consistent compressive force. The results shown in Fig. 16(b) demonstrate that our optimized structures can withstand significantly more force compared to the uniform support-free TPMS infill model used as the initialization.
Furthermore, we have conducted compression tests on our structures and others shown in Fig. 15. The tests are conducted under two volume fraction settings, and the statistical results are plotted in Figs. 17 and 18. The results in Fig. 17 demonstrate that our optimized structure can withstand higher force compared to honeycomb structure models, while the honeycomb model does not take overhang constraints into consideration. Our optimized model with a 54.5% volume fraction can withstand nearly 10kN of force, with significantly better stiffness than models generated by other methods, as shown in Fig. 18. The detailed structures after compression tests are shown in Fig. 19.

Fabrication quality. To compare the printing results of our method with and without the self-supporting constraint, we print four optimized models shown in Fig. 10 with two methods (ours shown on the left vs structures optimized without overhang constraint shown on the right). The results are illustrated in Fig. 20. We can observe that the models printed by our method are in good condition, while the models printed by the other method have many artifacts such as collapsed and stringy structures due to the lack of supporting material in the overhang regions.

5.5. Limitations

The proposed TPMS-based porous structure optimization algorithm has several limitations. Firstly, although it greatly reduces the overhang regions, it cannot generate completely self-supporting porous structures. This may be due to the limited representation ability of TPMS. Secondly, while our algorithm avoids the remeshing procedure in every iteration compared to traditional FEM-based computation, the computational accuracy depends on the sizes of the background elements used to calculate the integrals in the analysis. Large background elements may reduce the efficiency of solutions. This is a common problem in all FEM-based methods. Thirdly, there is significant room for improvement in the efficiency of the current algorithm. In fact, the algorithm spends most of the time in the sensitivity analysis, where the derivative calculation is computationally expensive, as shown in the formulas (36) and (38).

6. Conclusion

In this paper, we propose an algorithm to optimize porous structures with self-supporting constraints based on TPMS representations. The optimized structure has a function representation and inherits several good properties of TPMS, including high smoothness, full connectivity, good controllability, high surface-to-volume ratio, and good mechanical properties. It is also print-friendly in additive manufacturing since it generates less overhang regions and guarantees minimal thickness. Experimental results demonstrate the effectiveness of the proposed method.

Our framework utilizes period and thickness variables in TPMS-based representations to construct the geometry, rather than using traditional topology optimization with density variables. Additionally, we avoid the computationally expensive remeshing procedure during analysis, which is necessary in many other topology optimization methods. We achieve significant computational acceleration by employing a super element strategy, a multigrid solver, and GPU computing via the CUDA parallel programming API to handle millions of elements and achieve high-performance computing.

However, there are still several research directions that could improve our algorithm. Firstly, to obtain fully self-supporting structures, new representations should be explored. Secondly, the analysis step is still computationally expensive, and new strategies such as fast assembly of the stiffness matrix and efficient computation in sensitivity analysis could be explored. Finally, optimizing the tradeoff between efficiency and accuracy could also be discussed.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61972368), the Provincial Natural Science Foundation of Anhui, PR China (No. 2208085Q001) and the Fundamental Research Funds for the Central Universities, PR China (No. WK0010000075).
Fig. 19. Comparison of the detailed structures of several methods after compression tests.

Fig. 20. Comparison of printing results. The models on the left of each sub-figure are printed using our method and the models on the right are printed using the method without overhang constraint.

References


