

Article

Computation of Implicit Representation of Volumetric Shells with Predefined Thickness

Martin Geier *  and Hussein Alihussein 

TU Braunschweig, Institute for Computational Modeling in Civil Engineering (iRMB), Pockelsstr. 3, 38106 Braunschweig, Germany; hussein@irmb.tu-bs.de

* Correspondence: geier@irmb.tu-bs.de

Abstract: We propose and validate a method to find an implicit representation of a surface placed at a distance h from another implicit surface. With two such surfaces on either side of the original surface, a volumetric shell of predefined thickness can be obtained. The usability of the proposed method is demonstrated through providing solid models of triply periodic minimal surface (TPMS) geometries with a predefined constant and variable thickness. The method has an adjustable order of convergence. If applied to surfaces with spatially varying thickness, the convergence order is limited to second order. This accuracy is still substantially higher than the accuracy of any contemporary 3D printer that could benefit from the function as an infill volume for shells with predefined thicknesses.

Keywords: implicit surface; gyroid; Schwarz P; neovius; offsetting; Richardson extrapolation



Citation: Geier, M.; Alihussein, H. Computation of Implicit Representation of Volumetric Shells with Predefined Thickness.

Algorithms **2021**, *14*, 125. <https://doi.org/10.3390/a14040125>

Academic Editors: Andres Iglesias Prieto and Abel J.P. Gomes

Received: 3 March 2021

Accepted: 15 April 2021

Published: 18 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Implicit surfaces are of considerable interest in engineering as they provide an efficient means for the modeling of complex shapes. However, if transferred to the physical world by any means of production, any surface must have a finite thickness and hence, becomes a volume. This can be done by tools for computer aided design by finding surface points and extruding them in the normal direction. For example, Payne and Toga [1] proposed an algorithm to compute a distance field for a discretized surface such that the offset surface could be extracted as an isosurface of the distance field. Venkatesh applied the marching cube algorithm to generate volume models from triply periodic minimal surfaces [2]. Held et al. [3] used generalized Voronoi diagrams to compute offset functions of variable thickness. Pham presented an overview on offsetting methods [4], which was later extended by Maekawa [5]. However, these reviews do not address the analytic offsetting of implicit surfaces.

Fayolle et al. [6] proposed a method for offsetting implicit functions by computing level sets of the distance function. The distance function is obtained by normalizing the original implicit surface by Rvachev's normalization [7]. Fayolle's method is based on numerical procedures that require sampling of the initial implicit surface. They apply the Euler forward method to propagate the offset into the direction of the normal to the surface.

Sethian [8] provides a good overview over fast marching methods used for offsetting implicit surfaces. He also discusses how the accuracy of the fast marching method can be improved by discretizing the required normal derivatives with finite differences of higher order. However, this by itself does not guarantee an improvement of the order of convergence of the method.

In the mentioned methods, the resulting new surface is either a discretized surface with an explicit surface description or a discretized form of the implicit surface. In both cases, many of the appealing properties of the original implicit surface are lost.

Explicit surface representations usually require much more data than implicit representations, especially if the resolution requirements are high. Applications for implicit surfaces are in 3D printing [9–11] and simulation software. In both domains, tessellated

surfaces are currently the state of the art. However, simulation software can suffer from artifacts when curved surfaces are approximated by triangles. For example, the accurate simulation of flow around a sphere at a high Reynolds number can be qualitatively wrong if the sphere is approximated by triangles due to spurious flow separation at the edges of triangles, while accurate results can be obtained with the sphere defined as geometric primitive [12], which also requires less memory.

The purpose of the current paper is to propose a method to obtain a new implicit surface at a predefined distance to a given implicit surface.

The reminder of the paper is organized as follows. In Section 2, we introduce a low-order approximation of the offset function to the implicit surface. In Section 3, it is shown how the approximation of the low-order scheme can be improved by Richardson extrapolation in order to obtain a second-order convergent method. In Section 4, it is shown how this methodology is extended to obtain offset functions at any order of convergence. Section 5 demonstrates empirically that the predicted convergence orders are in fact obtained by computing the error norm for offset functions for three different triply periodic minimal surfaces. In Section 6, we discuss the case of a variable thickness of the shell and demonstrate second-order of convergence. Conclusions are given in Section 7.

2. First-Order Offset Function

Let $f(\vec{x}) = 0$ be an implicit surface Γ . Our aim is to find an implicit surface Γ_h as the solution of the offset function $\phi(\vec{x}, h) = 0$ such that the minimal Euclidean distance between any point on Γ to Γ_h is h . Further, we require that the distance between Γ_h and Γ_{-h} is $2h$. An exact solution to this problem is difficult to find in general and might not even exist, but we can start from a low-order approximation by computing the Taylor expansion of $f(\vec{x})$:

$$\phi(\vec{x}, h) = f(\vec{x}) - h\partial_n f(\vec{x}) + \mathcal{O}(h^2), \quad (1)$$

where the index n denotes the normal to Γ , which is approximated to lowest order by the direction of the gradient of $f(\vec{x})$ such that Equation (1) becomes

$$\phi(\vec{x}, h) = f(\vec{x}) - h\sqrt{(\nabla f(\vec{x})) \cdot (\nabla f(\vec{x}))} + \mathcal{O}(h^2). \quad (2)$$

Omitting the error term and rewriting $(\nabla f(\vec{x})) \cdot (\nabla f(\vec{x})) = (\nabla f(\vec{x}))^2$ gives us the first-order offset function $\phi^{(1)}(\vec{x}, h)$, which can be interpreted as an eikonal equation:

$$\phi^{(1)}(\vec{x}, h) = f(\vec{x}) - h\sqrt{(\nabla f(\vec{x}))^2}. \quad (3)$$

From $\phi^{(1)}(\vec{x}, h) = 0$, a first-order accurate approximation of the offset surface denoted by $\Gamma_h^{(1)}$ can be computed.

We note here that computing an offset function via the propagation of a level set is far from new [13]. In the following, we demonstrate that this simple method can be improved to any specified order of accuracy.

3. Second-Order Offset Function

A first-order accurate solution to the offset problem is only acceptable if h is very small. In what follows, we present a systematic approach to rise the order of accuracy starting from $\phi^{(1)}(\vec{x}, h) = 0$.

Let us first note that since $\phi^{(1)}(\vec{x}, h)$ is an implicit surface itself, we are able to apply the same methodology again in order to obtain an offset function of $\phi^{(1)}(\vec{x}, h)$. In the following recursive computation of the offset function $\phi_n^{(m)}(\vec{x}, h)$, we will indicate the level of the recursion by subscript n and the order of approximation by superscript (m) . With this, it is, for example, possible to cover the distance h in two steps of equal size. To achieve this, we recursively apply Equation (3) in itself to obtain

$$\phi_2^{(1)}(\vec{x}, h) = \phi_1^{(1)}(\vec{x}, h/2) - \frac{h}{2} \sqrt{(\nabla \phi_1^{(1)}(\vec{x}, h/2))^2}. \quad (4)$$

This second iteration $\phi_2^{(1)}(\vec{x}, h)$ has the same order of accuracy as $\phi^{(1)}(\vec{x}, h)$, as can be seen by substituting $\phi_1^{(1)}(\vec{x}, h/2) = \phi(\vec{x}, h/h) + \mathcal{O}(h^2)$ and applying the triangle inequality while realizing that h is not a function of \vec{x} , i.e., $\sqrt{(\nabla(\phi(\vec{x}, h/h) + \mathcal{O}(h^2)))^2} \leq \sqrt{(\nabla(\phi(\vec{x}, h/h))^2} + \mathcal{O}(h^2)$.

While $\phi_2^{(1)}(\vec{x}, h)$ has the same order of accuracy as $\phi^{(1)}(\vec{x}, h)$, it is expected to be more accurate as it uses half the step size. Adding more steps would raise the accuracy further but the methodology would quickly become impractical as the algebraic complexity of the implicit surface function grows with each step. Instead, it appears more promising to combine the obtained solutions through Richardson extrapolation [14]. Richardson extrapolation is a convergence acceleration method for sequences of approximations obtained with different smallness parameters (e.g., step sizes). It is applicable if the order of convergence with respect to the smallness parameter is either known or can be obtained directly from the series of solutions. For this, it must be possible to express the error of the method as a Taylor series in the smallness parameter. It is also necessary that the error does not depend on terms not related to the smallness parameter. Therefore, only results obtained by the same deterministic method can be combined in a Richardson extrapolation.

We recall the general equation for Richardson extrapolation given a solution $g^{(q)}(h)$ obtained with smallness parameter h and a solution $g^{(q)}(\beta h)$ obtained with a smallness parameter βh , both with the same method of order q :

$$g^{(q+1)} = \frac{g^{(q)}(h) - \beta^{-q} g^{(q)}(\beta h)}{1 - \beta^{-q}}. \quad (5)$$

In the current case, the refinement factor is $\beta = 1/2$ and the convergence order is $q = 1$. Note that both methods have to be evaluated at the same final location, which in the current case, is after a distance h . However, the solution $g^{(q)}(h) = \phi_1^{(1)}(\vec{x}, h)$ is obtained using a single step while the second solution $g^{(q)}(h/2) = \phi_2^{(1)}(\vec{x}, h)$ uses two steps of size $h/2$. After some algebra, this yields

$$\phi^{(2)}(\vec{x}, h) = 2\phi_2^{(1)}(\vec{x}, h) - \phi_1^{(1)}(\vec{x}, h) = f(\vec{x}) - h \sqrt{(\nabla \phi_1^{(1)}(\vec{x}, h/2))^2}. \quad (6)$$

A second-order approximation of the surface $\Gamma_h^{(2)}$ is obtained from $\phi^{(2)}(\vec{x}, h) = 0$.

4. Arbitrary-Order Offset Function

There are various systematic ways to rise the convergence order further. We propose here using an extended Richardson extrapolation strategy. For this, we briefly return to the theory behind the Richardson extrapolation and explain how the same method can be used to obtain results at arbitrary order.

The general assumption behind the Richardson extrapolation is that the approximating function $\phi_n^{(1)}(\vec{x}, h)$ can be written as

$$\phi_n^{(1)}(\vec{x}, h) = \phi(\vec{x}, h) + \sum_{i=1}^{\infty} \mathcal{E}_i \left(\frac{h}{n}\right)^i. \quad (7)$$

Here, n is the number of equidistant steps or iterations to reach a distance h such that the step size is h/n . The constants \mathcal{E}_i are independent of the step size. Richardson extrapolation ignores all but the leading-order error to get a better approximation of $\phi(\vec{x}, h)$ from the weighted combination:

$$\phi^{(2)}(\vec{x}, h) = w_1 \phi_{n1}^{(1)}(\vec{x}, h) + w_2 \phi_{n2}^{(1)}(\vec{x}, h). \quad (8)$$

The weights have to be recovered by inserting Equation (7) into Equation (8) and equating coefficients with regards to $\phi(\vec{x}, h)$ and \mathcal{E}_1 , where $\phi(\vec{x}, h)$ is to be recovered while the error term is to be eliminated, i.e.,

$$\begin{aligned}\phi(\vec{x}, h) : \quad 1 &= w_1 + w_2 \\ \mathcal{E}_1 : \quad 0 &= w_1 \frac{1}{n_1} + w_2 \frac{1}{n_2}.\end{aligned}\quad (9)$$

Solving this set of equations gives rise to the usual Richardson extrapolation formula.

While Richardson extrapolation usually only considers the elimination of the leading order error term, the formula can be extended to cover several error terms at the same time. The conditions for the extended Richardson extrapolation can be written simply by

$$\begin{aligned}\phi(\vec{x}, h) : \quad 1 &= \sum_{j=1}^m w_j \\ \mathcal{E}_i : \quad 0 &= \sum_{j=1}^m \frac{w_j}{(n_j)^i},\end{aligned}\quad (10)$$

where $i \leq m$. By solving Equation (10), the weights for an extended Richardson extrapolation at arbitrary order can be determined. Table 1 lists the weight coefficients for extended Richardson extrapolations up to order four.

Table 1. Table of the weights for extended Richardson extrapolation up to order four.

Order	$\phi_1^{(1)}(\vec{x}, h)$	$\phi_2^{(1)}(\vec{x}, h)$	$\phi_3^{(1)}(\vec{x}, h)$	$\phi_4^{(1)}(\vec{x}, h)$
1	1	-	-	-
2	-1	2	-	-
3	$\frac{1}{2}$	-4	$\frac{9}{2}$	-
4	$-\frac{1}{6}$	4	$-\frac{27}{2}$	$\frac{32}{3}$

As higher derivatives are successively eliminated from the Taylor expansion through the extended Richardson extrapolation, our method can also be interpreted as a way to obtain successively accurate approximations of $\sqrt{(\nabla f(\vec{x}))^2}$:

$$\sqrt{(\nabla f(\vec{x}))^2} = \frac{f(\vec{x}) - \phi^{(n)}(\vec{x}, h)}{h} + \mathcal{O}(h^n). \quad (11)$$

In this way, our method can be seen as a semianalytic way to construct successively more accurate approximations to the eikonal equation.

4.1. Implicit Infill Volumes

An implicit function for the infill volume can be derived from the implicit offset functions in positive and negative directions indicated by $h < 0$ and $h > 0$. We define the implicit volume $\Phi(\vec{x}, h)$ as the shell of the zero contour of $f(\vec{x})$ with thickness h by

$$\Phi(\vec{x}, h) = \phi(\vec{x}, h/2)\phi(\vec{x}, -h/2). \quad (12)$$

The point \vec{x} is said to be inside the shell if $\Phi(\vec{x}, h) < 0$ and outside the shell if $\Phi(\vec{x}, h) > 0$.

4.2. Limitations

The problem to supply an arbitrary surface with a finite thickness is not well posed in general. In particular, it is difficult to unambiguously define what is meant by *thickness* of a curved surface. For example, if the local radius of curvature of a surface is small

compared to the desired thickness of the shell, the resulting body could be self-intersecting. In the proposed method, the thickness is defined through Equation (1) as the path length of a curve following the normal of the implicit function $f(\vec{x})$. For shells with local radii much larger than the thickness h , this path is approximately straight, but for increasing thicknesses, the path itself can be curved and the distance of the end points of a curved path is always shorter than the length of the path. Hence, the method proposed here is only valid for shells with a thickness much smaller than their local radii of curvature. It is important to note that this limitation is intrinsic to the definition of thickness in the method and this limitation is not affected by the order of the Richardson extrapolation chosen. Further, we note that the method does not check for self-intersection.

5. Examples

In this section, we will investigate the performance of the proposed methodology considering some examples of varying complexity.

5.1. Circle

A circle can be defined in a Cartesian coordinate system by the implicit function

$$f(x, y) = x^2 + y^2 - R^2, \quad (13)$$

with R being the radius. For the zero contour of $f(x, y)$, it is straightforward to compute the two contours with offset h by replacing the radius with $(R - h)$ and $(R + h)$, respectively.

In order to validate the proposed method, we compute the first- and second-order offset functions from Equation (13):

$$\phi^{(1)}(x, y, h) = x^2 + y^2 - R^2 - 2h\sqrt{x^2 + y^2} \quad (14)$$

$$\phi^{(2)}(x, y, h) = x^2 + y^2 - R^2 - 2h\sqrt{x^2 + y^2 + \frac{h^2}{4}} - h\sqrt{x^2 + y^2}. \quad (15)$$

Rewriting Equations (14) and (15) in terms of polar coordinates gives

$$\phi^{(1)}(\rho, \theta, h) = \rho^2 - R^2 - 2h\rho \quad (16)$$

$$\phi^{(2)}(\rho, \theta, h) = \rho^2 - R^2 - 2h\sqrt{\rho^2 + \frac{h^2}{4}} - h\rho. \quad (17)$$

At the specified location of the new surface, the offset functions are

$$\phi^{(1)}(R + h, \theta, h) = -h^2 \quad (18)$$

$$\phi^{(2)}(R + h, \theta, h) = 0, \quad (19)$$

which means that the second-order offset function provides the exact solution while the error of the first-order offset function is within the predicted range.

5.2. Triply Periodic Minimal Surfaces (TPMSs)

Periodic minimal surfaces have many potential applications in engineering. There are several ways to define a minimal surface [15]. It can be defined as the surface that locally has the smallest area. Another definition is that a minimal surface has a zero mean curvature. In the following examples, we assume that the TPMS is given by the zero contour of the function $\mathfrak{F}(\vec{x})$. Without loss of generality, we assume the TPMS to be periodic in a cubic box with side length L .

The offset functions up to fourth order of convergence for a TPMS are obtained following the proposed procedure.

We compute four different first-order approximations of the offset function according to

$$\phi_1^{(1)}(\vec{x}, h) = \mathfrak{F}(\vec{x}) - h |\nabla \mathfrak{F}(\vec{x})| \quad (20)$$

$$\phi_2^{(1)}(\vec{x}, h/2) = \phi_1^{(1)}(\vec{x}, h/2) - \frac{h}{2} |\nabla \phi_1^{(1)}(\vec{x}, h/2)| \quad (21)$$

$$\phi_3^{(1)}(\vec{x}, h/3) = \phi_2^{(1)}(\vec{x}, h/3) - \frac{h}{3} |\nabla \phi_2^{(1)}(\vec{x}, h/3)| \quad (22)$$

$$\phi_4^{(1)}(\vec{x}, h/4) = \phi_3^{(1)}(\vec{x}, h/4) - \frac{h}{4} |\nabla \phi_3^{(1)}(\vec{x}, h/4)|. \quad (23)$$

From this, we obtain different orders of approximation using the extended Richardson extrapolation:

$$\phi^{(1)}(\vec{x}, h) = \mathfrak{F}(\vec{x}) - h |\nabla \mathfrak{F}(\vec{x})| \quad (24)$$

$$\phi^{(2)}(\vec{x}, h) = 2\phi_2^{(1)}(\vec{x}, h/2) - \phi_1^{(1)}(\vec{x}, h) \quad (25)$$

$$\phi^{(3)}(\vec{x}, h) = \frac{9}{2}\phi_3^{(1)}(\vec{x}, h/3) - 4\phi_2^{(1)}(\vec{x}, h/2) + \frac{1}{2}\phi_1^{(1)}(\vec{x}, h) \quad (26)$$

$$\phi^{(4)}(\vec{x}, h) = \frac{32}{3}\phi_4^{(1)}(\vec{x}, h/4) - \frac{27}{2}\phi_3^{(1)}(\vec{x}, h/3) + 4\phi_2^{(1)}(\vec{x}, h/2) - \frac{1}{6}\phi_1^{(1)}(\vec{x}, h). \quad (27)$$

The derivation of $\phi^{(n)}(\vec{x}, h)$ is easily automated in a computer algebra system such like Mathematica, Maple, or SymPy.

Mathematica source code for the generation of offset functions at different orders of convergence is presented in the Supplementary Material published alongside this paper.

5.2.1. Error Measurement Procedure

To measure the accuracy of the proposed method, we employ a sampling technique. We want to measure the distance between the surface of the TPMS and the zero contour of its offset function. The computer algebra software Mathematica is used to find a large sample of points N on one of the surfaces. The sampled points are uniformly distributed in x and y directions. The z coordinate is determined by solving for z given x and y on the surfaces. The procedure for collecting the points is described in Algorithm 1.

Algorithm 1: Points sampling from an implicit surface $F(x, y, z) = 0$

input :implicit surface function $F(x, y, z)$
output:list of 3-tuple *SampledPoints*
 // create two list of coordinates in x and y directions
 $A := \{a \mid a \in [0, 0.05, 0.1, \dots, 0.95, 1]\}$
 $B := \{b \mid b \in [0, 0.05, 0.1, \dots, 0.95, 1]\}$
 // substitute each a and b into F and solve for the c
 $SampledPoints := \{(a, b, \text{Solve}(F(a, b, c) = 0, c)) \mid a \in A \text{ and } b \in B\}$

From each point, a ray is shot in the normal direction to the surface. Then, the intersection distance $h_m^{(n)}$ with the other surface is measured and compared to the predefined distance h . The error is measured in the following norm:

$$Error = \|h^{(n)} - h\|_2 = \frac{1}{N} \sum_{i=1}^N \sqrt{\left(\frac{h_m^{(n)}(i) - h}{h}\right)^2}, \quad (28)$$

where N is the number of points used and n is the order of the approximation used to represent the offset surface. The procedure for calculating the error is described in Algorithm 2. The error measurement procedure, including sampling of the points from the surface and finding the intersection distance to the next surface, is done on both surfaces, i.e., the original zero contour of the TPMS and the offset function.

Algorithm 2: Calculation of the L^2 error in distance h from points in the surface $F_1(\vec{x}) = 0$ to a distanced surface $F_2(\vec{x}) = 0$

```

input :list of 3-tuple SampledPoints
input :implicit surface functions  $F_1(x, y, z)$  and  $F_2(x, y, z)$ 
input :prescribed distance  $h$ 
output:real Error
list pointError := {}
foreach 3-tuple  $\vec{p}$  in SampledPoints do
    // calculate the normal  $\hat{n}$  at  $\vec{p}$ 
     $\hat{n} := \frac{\nabla F_1(\vec{p})}{|\nabla F_1(\vec{p})|}$ 
    // shoot a ray from  $\vec{p}$  a distance  $h_m$  and calculate the position
    vector  $\vec{r}(h_m)$ 
     $\vec{r}(h_m) := \vec{p} + h_m \hat{n}$ 
    // solve for the distance  $h_m$ 
     $h_m := \text{Solve}(F_2(\vec{r}(h_m)) = 0, h_m)$ 
    // calculate the  $L^2$  error
    pointError := Append(pointError,  $\sqrt{\left(\frac{h_m - h}{h}\right)^2}$ )
end
// calculate the error mean
Error := Mean(pointError)

```

5.2.2. Gyroid

A Gyroid is a TPMS that is described by the implicit equation

$$\begin{aligned} \mathfrak{F}_G(\vec{x}) = & \sin\left(\frac{2\pi x}{L}\right) \cos\left(\frac{2\pi y}{L}\right) + \sin\left(\frac{2\pi y}{L}\right) \cos\left(\frac{2\pi z}{L}\right) \\ & + \sin\left(\frac{2\pi z}{L}\right) \cos\left(\frac{2\pi x}{L}\right) = 0, \end{aligned} \quad (29)$$

where L is the edge length of a cube circumscribing a unit cell of the Gyroid shown in Figure 1 together with the other surfaces investigated in this paper. With the emergence of 3D printing, Gyroids are of great interest since Gyroid-based patterns can be used to fill space while providing sufficient strength and low weight to the 3D-printed parts. Figure 2 shows the bodies generated with the third-order offset function of the Gyroid for various thicknesses.

We employ the error measurement from Section 5.2.1 to validate the proposed method. Figure 3 shows the convergence of the different offset functions for varying thickness. It is observed that the predicted order of convergence is recovered. The first-order offset function appears to be converging with second-order for large enough thicknesses but approaches the predicted asymptote for thin shells.

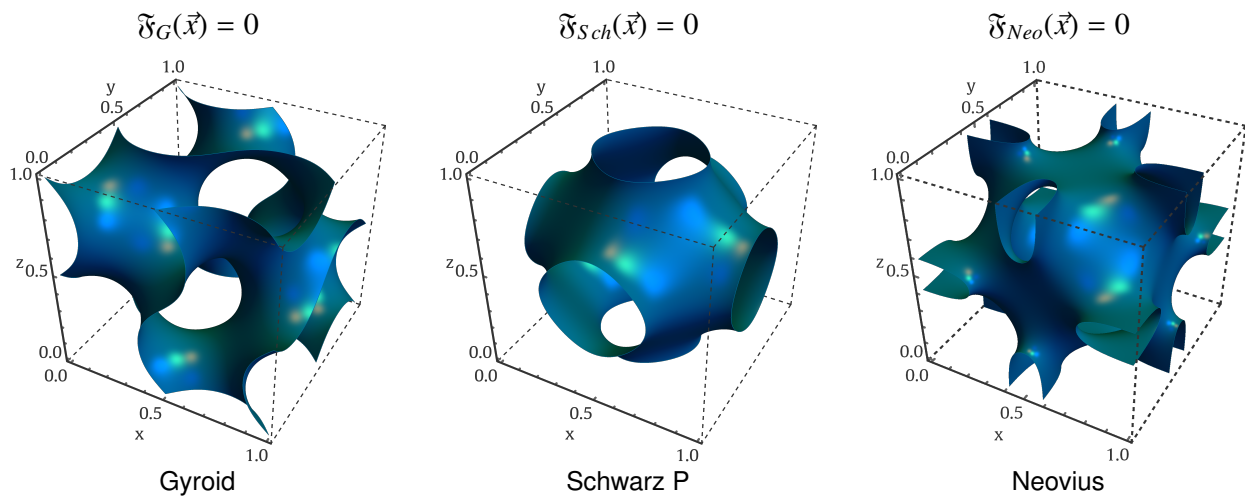


Figure 1. The original Gyroid, Schwarz P, and Neovius zero thickness surfaces, respectively, $\mathfrak{F}_G(\vec{x}) = 0$, $\mathfrak{F}_{Sch}(\vec{x}) = 0$, and $\mathfrak{F}_{Neo}(\vec{x}) = 0$. Only one cell is considered here: $\vec{x} \in [0, 1] \times [0, 1] \times [0, 1]$.

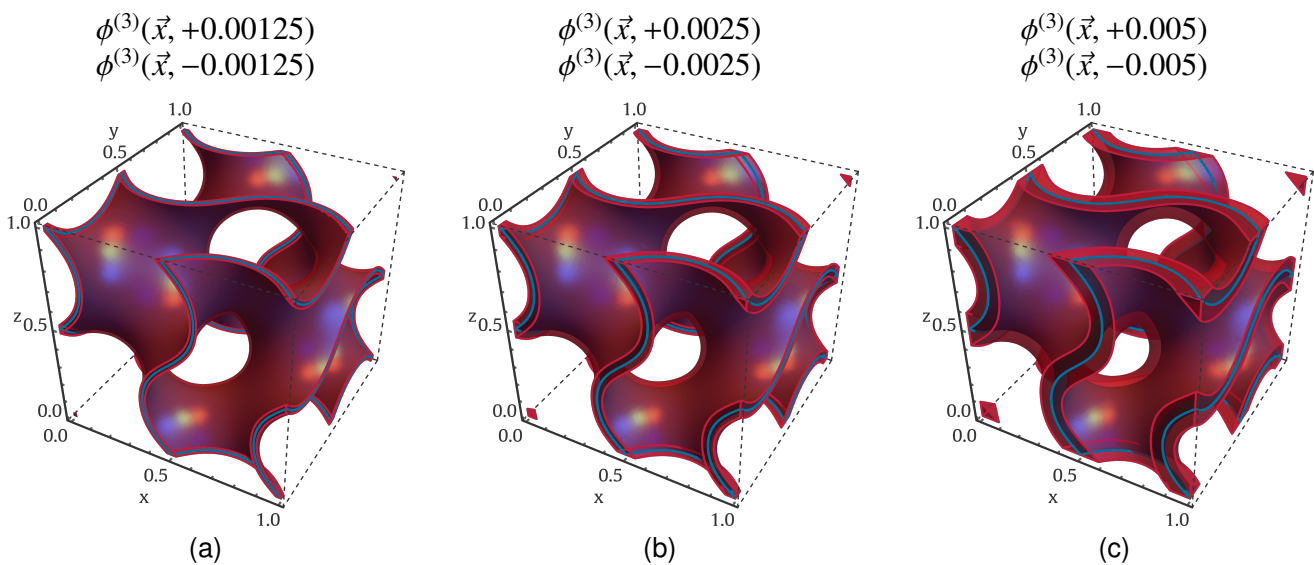


Figure 2. Infill volume enclosed between two offset contours from the original Gyroid interface $\mathfrak{F}_G(\vec{x}) = 0$. For a unit cell, the offset distance is 0.00125 in (a), 0.0025 in (b), and 0.005 in (c). The third-order offset function was used.

5.2.3. Schwarz P

The next example is the Schwarz P TPMS, defined as

$$\mathfrak{F}_{Sch}(\vec{x}) = \cos\left(\frac{2\pi x}{L}\right) + \cos\left(\frac{2\pi y}{L}\right) + \cos\left(\frac{2\pi z}{L}\right) = 0. \quad (30)$$

The resulting bodies obtained from the third-order offset function are shown in Figure 4. Again, we measure the convergence of the proposed method following Section 5.2.1 and display them in Figure 5. In this case, the asymptotic behavior of the four tested variants follows the theoretical prediction very closely.

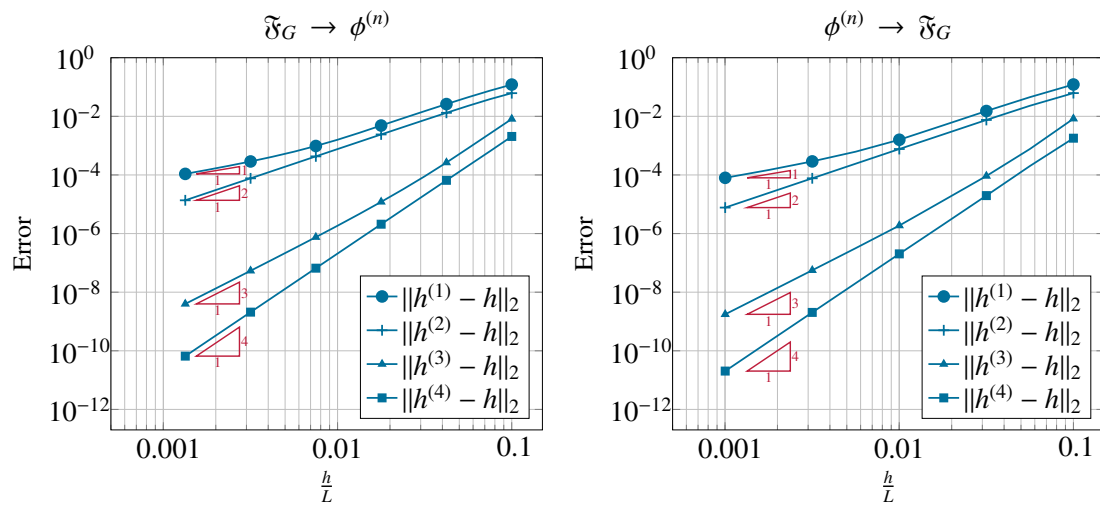


Figure 3. Gyroid convergence study. In both panels, the error in the measured offset h_m is plotted against different predefined normalized offsets. In the left figure, the offsets were measured starting from points on $\mathfrak{F}_G(\vec{x}) = 0$ to points on each surface $\phi^{(n)}(\vec{x}, h) = 0$, respectively. In the right figure, the measurement direction is reversed. The triangles represent the corresponding slope for each order n .

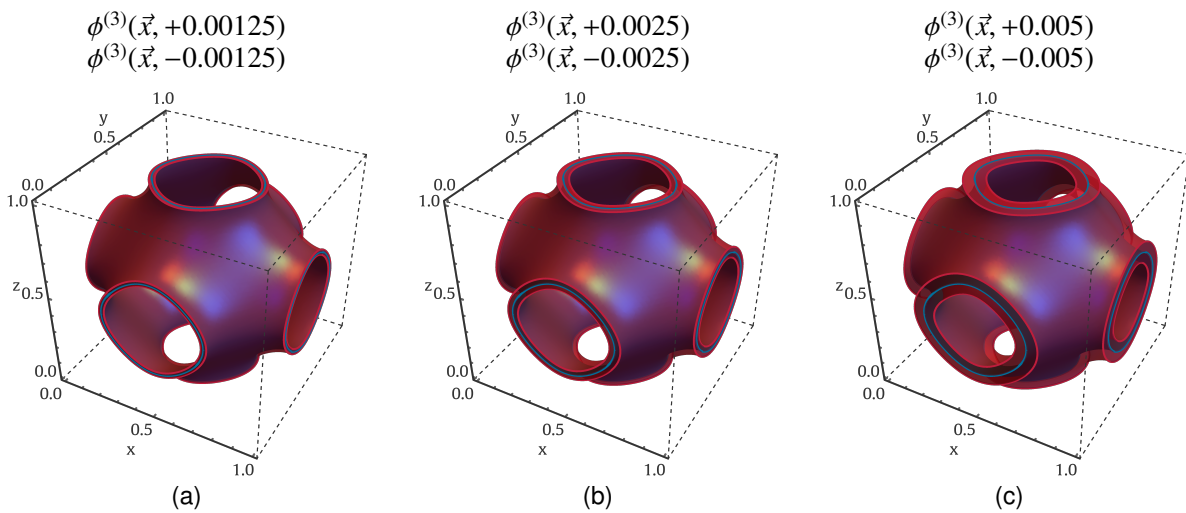


Figure 4. Infill volume enclosed between two offset contours from the original Schwarz P interface $\mathfrak{F}_{Sch}(\vec{x}) = 0$. For a unit cell, the offset distance is 0.00125 in (a), 0.0025 in (b), and 0.005 in (c). The third-order offset function was used.

5.3. Neovius

The previous examples had very homogeneous curvatures. For the next example, we chose a surface for which the curvature changes. The Neovius surface is defined as

$$\begin{aligned} \mathfrak{F}_{Neo}(\vec{x}) = & 3 \left[\cos \left(\frac{2\pi x}{L} \right) + \cos \left(\frac{2\pi y}{L} \right) + \cos \left(\frac{2\pi z}{L} \right) \right] \\ & + 4 \left[\cos \left(\frac{2\pi x}{L} \right) \cos \left(\frac{2\pi y}{L} \right) \cos \left(\frac{2\pi z}{L} \right) \right] = 0. \end{aligned} \quad (31)$$

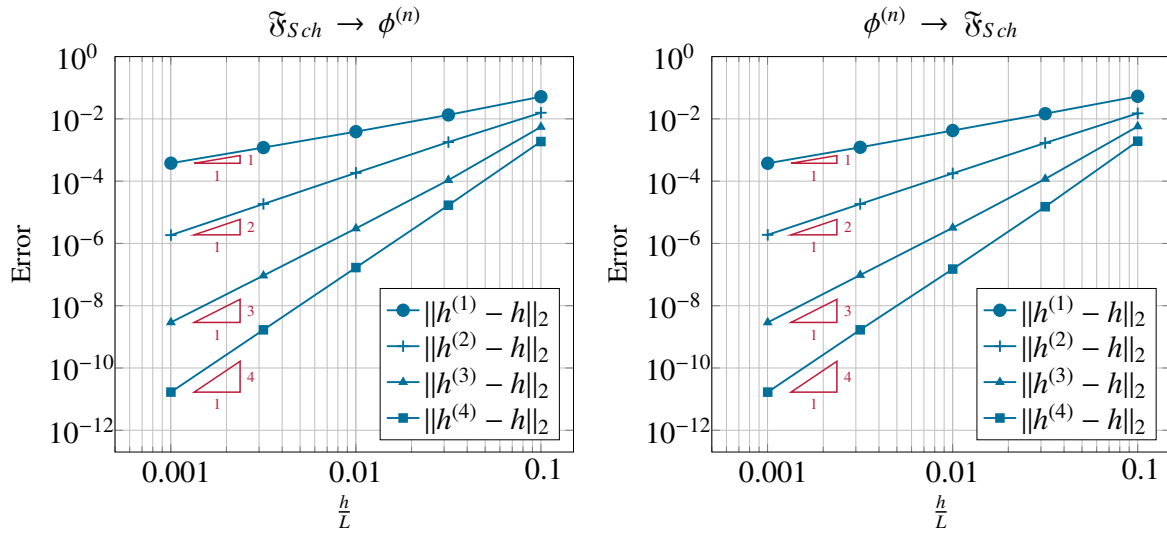


Figure 5. Schwarz P convergence study. In both panels, the error in the measured offset h_m is plotted against different predefined normalized offsets. In the left figure, the offsets were measured starting from points on $\mathfrak{F}_{Sch}(\vec{x}) = 0$ to points on each surface $\phi^{(n)}(\vec{x}, h) = 0$, respectively. In the right figure, the measurement direction is reversed. The triangles represent the corresponding slope for each order n .

The bodies generated from the third-order offset function are depicted in Figure 6. It is seen that the intersection of the Neovius surface with the axes plains has an elliptical shape. Figure 6c shows how the hollow connections between unit cells are filled up. It is obviously difficult to guarantee a constant thickness of the shell in these connections once the thickness approaches the width of the openings. This is evident from the convergence analysis, executed in the same way as for the other two bodies and displayed in Figure 7. For small enough thicknesses, the theoretically predicted convergence is well captured. However, for thicknesses of $0.1 L$, the error deteriorates as the resulting shape can no longer be described as a shell of the Neovius surface. One observation is that this deterioration is less severe when the thickness is measured from the offset function to the Neovius surface than when it is measured the other way around. The reason for this is most likely seen in the fact that the proportion of the offset function contour with inaccurate distance towards the Neovius surfaces shrinks when the connections are closed as these closed areas are no longer part of the offset function.

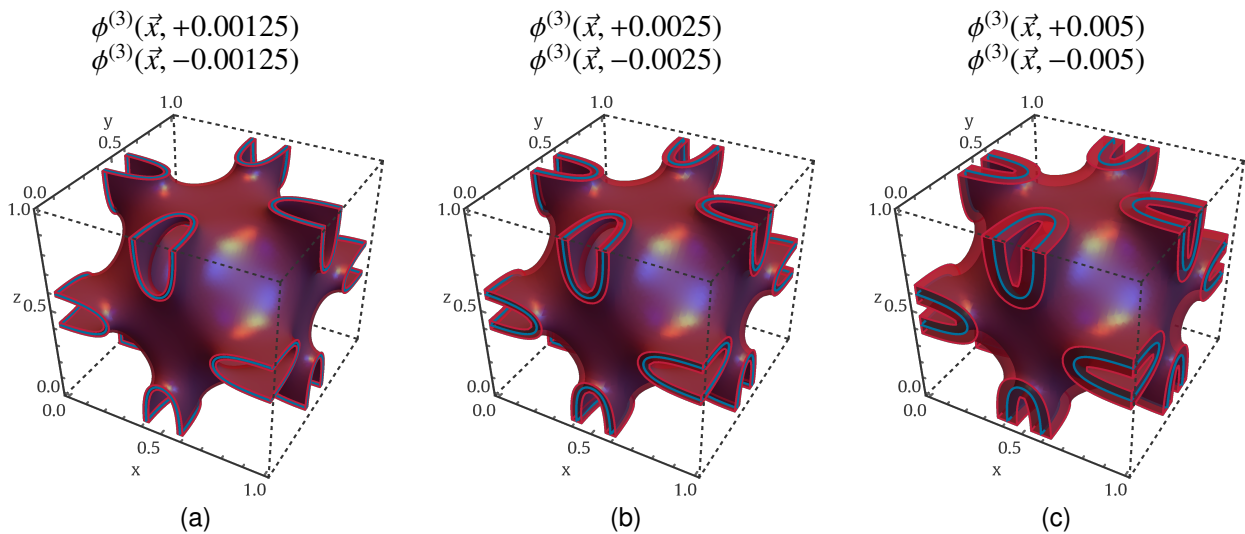


Figure 6. Infill volume enclosed between two offset contours from the original Neovius interface $\mathfrak{F}_{Neo}(\vec{x}) = 0$. For a unit cell, the offset distance is 0.00125 in (a), 0.0025 in (b), and 0.005 in (c). The third-order offset function was used.

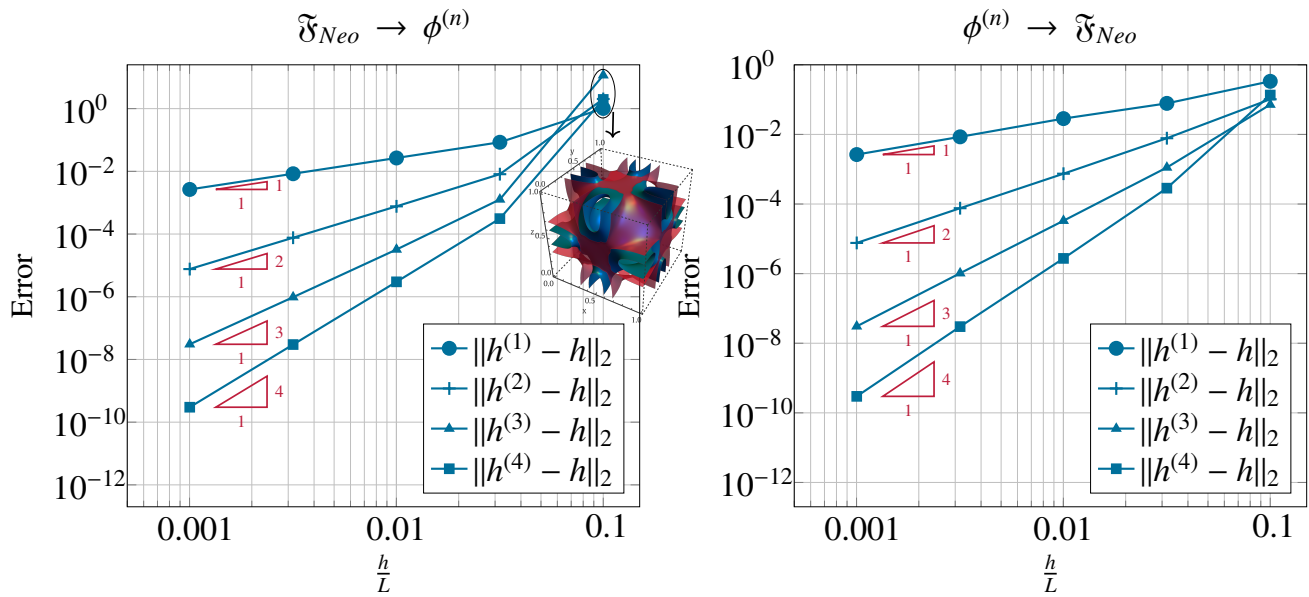


Figure 7. Neovius convergence study. In both panels, the error in the measured offset h_m is plotted against different predefined normalized offsets. In the left figure, the offsets were measured starting from points on $\mathfrak{F}_{Neo}(\vec{x}) = 0$ to points on each surface $\phi^{(n)}(\vec{x}, h) = 0$, respectively. In the right figure, the measurement direction is reversed. The triangles represent the corresponding slope for each order n . The nonasymptotic behavior shown at offsets of larger values (e.g., $h/L = 0.1$) is attributed to the high curvature of the original surface.

6. Shells with Variable Thickness

It can sometimes be of interest to allow for a variable thickness of the shell [16,17]. Applying the method outlined in this paper to shells with variable thickness would imply substituting h by the function $h(\vec{x})$ in all stages of $\phi^{(n)}(\vec{x}, h(\vec{x}))$ before taking the derivative. The implicit dependence of h on \vec{x} would then automatically be taken into account. However, this approach would not produce the desired result, which can be explained as follows: The proposed method approximates the thickness of the shell by moving a predefined distance along the normal of $\phi(\vec{x}, h) = 0$ for increasing h . This procedure is only accurate under the assumption that the normal of $\phi(\vec{x}, h) = 0$ does not deviate substantially from the normal of $f(\vec{x}) = 0$, such that the procedure follows a straight line. However, if h is not constant, the normal becomes, through the chain rule of differentiation:

$$\vec{n} = \frac{\nabla \phi(\vec{x}, h(\vec{x}))}{|\nabla \phi(\vec{x}, h(\vec{x}))|} = \frac{\nabla \phi(\vec{x}, h) + \frac{\partial \phi(\vec{x}, h)}{\partial h} \nabla h(\vec{x})}{|\nabla \phi(\vec{x}, h) + \frac{\partial \phi(\vec{x}, h)}{\partial h} \nabla h(\vec{x})|}. \quad (32)$$

The direction of the normal will therefore be governed by how h changes with \vec{x} , which is another way of saying that if h is not constant, $\phi(\vec{x}, h(\vec{x})) = 0$ is naturally not parallel to $f(\vec{x}) = 0$.

Another and more straightforward approach to obtain variable thickness is to assume h to be constant in the derivation of $\phi(\vec{x}, h)$ and only replace it when evaluating $\phi(\vec{x}, h) = 0$. This also has limited accuracy because if h is a function of \vec{x} , it changes by a value $\sim h \partial_n h$ over the thickness h . For smooth variations of the thickness, it is reasonable to assume that $\partial_n h = \mathcal{O}(h)$ such that $h \partial_n h = \mathcal{O}(h^2)$. Hence, the accuracy of the proposed method for variable thickness is bound to second order. This is still sufficiently accurate for many applications.

Figure 8 shows the convergence for shells with a linearly increasing thickness along the x axis for the different TPMS. It is observed that the first-order method remains to be first-order accurate, albeit it performs better than expected for the Gyroid at larger

thicknesses. The third-order method typically shows better results than the second-order method but it too converges only with second order. Since the intrinsic error in the thickness is already of order $\mathcal{O}(h^2)$, further increasing the order of $\phi^{(n)}(\vec{x}, h)$ would not improve the asymptotic nature of the result. For a shell thickness of 0.01 L , the second-order method already obtains errors smaller than 0.03% of the target thickness in all three cases. Figure 9 shows the resulting bodies with variable thickness.

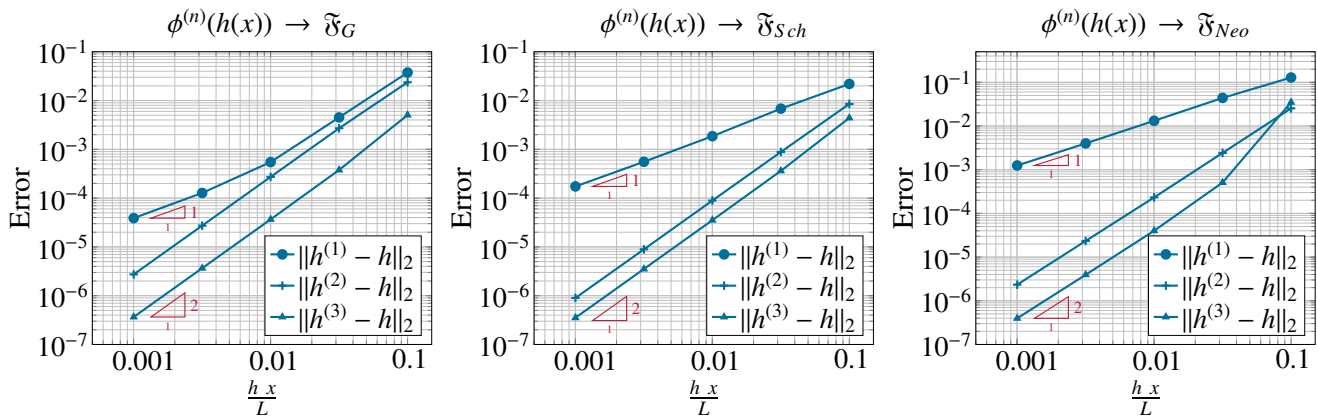


Figure 8. Convergence study for varied offset in the x direction. In each figure, the error in the measured offset h_m is plotted against different predefined normalized offsets governed by a linear function $h(x)$. The offsets at $x = 0$ are set to zero, while the offsets at $x = L$ vary from zero to a maximal value of 0.1 L . The offsets were measured starting from points on different offset surfaces $\phi^{(n)}(\vec{x}, h) = 0$ obtained with approximations up to order three. No further improvement is observed when using an approximation method with an order higher than two.

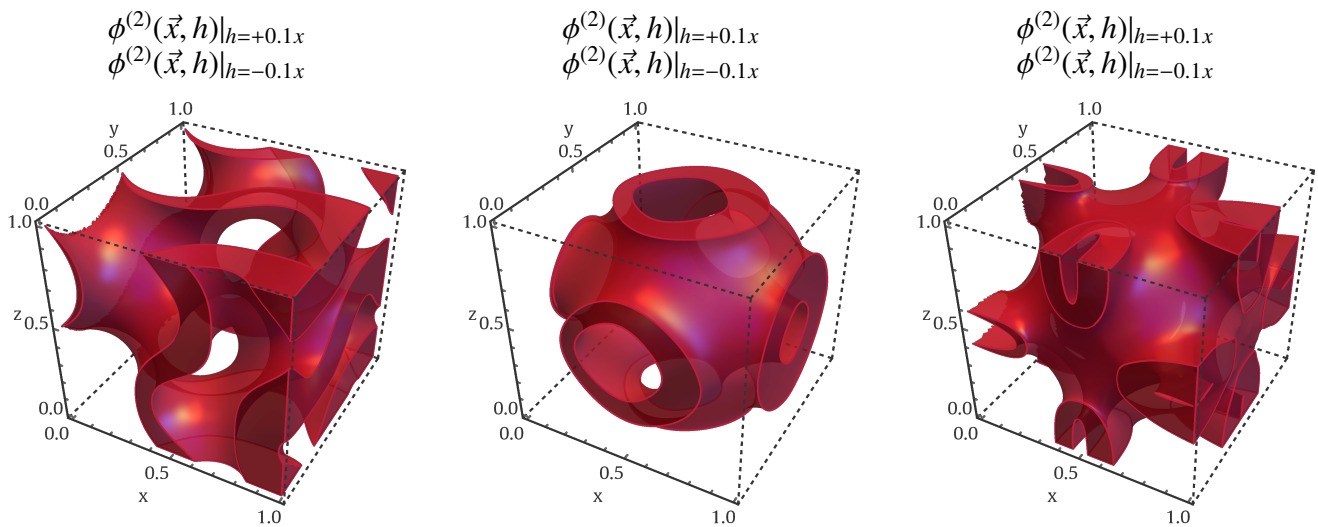


Figure 9. Infill volume enclosed between two offset contours from the original interface $\mathfrak{F}(\vec{x}) = 0$ of the Gyroid, Schwarz P, and Neovius, respectively. For a unit cell, the offset distance linearly increases from 0 up to 0.1 in the x direction. In the above figures, the method used to obtain the implicit representation for each of the offset interfaces is the second-order method.

7. Conclusions

In this paper, we proposed a systematic methodology for deriving analytic implicit body shells of predefined thickness from implicit surfaces. The convergence order and thereby the accuracy of the thickness of the shell can be freely adjusted through the extended Richardson extrapolation. The motivation for our method comes from engineering applications of additive manufacturing, where such infill volumes are required for the actual 3D-printing process on the one hand and for numerical simulations of such structures

on the other hand. For constant thicknesses, our method reaches a precision of smaller than 10^{-6} for wall thicknesses measuring 1% of the wave length of the periodic structure. This precision by far exceeds the accuracy of contemporary 3D-printing technology. By opting for offset functions of lower-order computational time can be reduced. The memory requirements for storing the offset function of a TPMS is negligible compared to a discretized surface mesh. Being analytic, the offset function does not lose any quality when scaled. At the time of this writing, it is not straightforward to use the offset function directly as input for commercial 3D printers as they typically expect a tessellated mesh representation of the surface, which can, of course, easily be obtained from discretizing the zero contour of the offset function. Our method can be used to identify, for any point in the domain, whether it lies in the shell or outside the shell. As the shell can be expressed as a solid body in this way, the question of water tightness of the surface never arises. However, the slicing of the object for G-code generation and 3D printing was not addressed in this theoretical work. In the presented semianalytic form, our method can only be applied to surfaces provided as differentiable implicit functions. Such surfaces are always closed as they provide an inner and an outer side. In 3D printing, it is often of interest to include open surfaces [18]. In our approach, this is possible by cutting the resultant shell with an additional solid in a postprocessing step.

Besides being useful for the definition of infill volumes for the printing process itself, the proposed method has other applications, for example, in the analysis and simulation of 3D-printed objects. In general, methods of adaptive manufacturing come with a finite smallest line width. In many applications, the line width of the printer is designed to coincide with the thickness of the shell structure under construction. This is, for example, common for 3D printing of slender concrete structures in applications of civil engineering and architecture [19,20]. But the same is also true for wire arc additive manufacturing [21] and the various types of powder bed methods like selective laser melting [22] and selective binding processes [23,24]. For structures built with these technologies, the thickness of the shell is often a simple result of the line width of the printer and is hence not specified as an infill volume. Here, the designer is interested in the resulting shell geometry actually manufactured by the printer which can be predicted with our method provided that the line width is known. Such knowledge is of particular interest for the modeling and simulation of 3D-printed structures as the simulation software requires accurate volume geometries. In simulation software, the substantial reduction of the memory requirements due to the use of an analytic function instead of a high-resolution surface mesh is a substantial advantage as simulation software is typically memory bound and preprocessing the geometry often takes more time than the actual simulation. This is actually the authors' main motivation as we are concerned with the simulation of air flow through 3D-printed TPMS with our lattice Boltzmann solver VirtualFluids [25–28]. Surface meshes are particularly ill-suited for large-scale periodic structures such that the search for a more economical representation of such structures was inevitable.

We note here that the proposed method is markedly different from discretized numerical methods for solving the eikonal equation such as the fast marching method described by Sethian [8]. The fast marching method depends on numerical differentiation, which is done on a grid. Even though Sethian points out that the fast marching method can be improved by using finite differences of higher order, this does not necessarily enhance the convergence order of the method. In fact, the convergence order is not directly related to the quality of the normal derivative as is seen from the fact that our method always builds on exact analytic derivations. Due to this utilization of exact derivatives, no discretization and hence no grid is required in the proposed method. This does not mean that our method is generally more efficient than a fast marching method, as discretization on a grid has its advantages. The analytically derived offset function can become extremely complicated. The proposed method is hence not to be seen as a replacement for the fast marching method but rather, as an additional, very precise tool for cases where the original implicit surface is efficiently expressed as an analytic function.

When the thickness of the structure is not constant but varies in space, our method reduces to second-order accuracy, which is still much more accurate than required by any perceivable application in additive manufacturing at the time.

Finally, we note that the extended Richardson extrapolation is by no means restricted to the method derived in this paper. The extended Richardson extrapolation provides a systematic procedure to derive methods of arbitrary accuracy starting from a poor approximation, provided that this approximation fulfills the requirement that the error can be expanded in series.

Supplementary Materials: The following are available online at <https://www.mdpi.com/1999-4893/14/4/125/s1>.

Author Contributions: Conceptualization, M.G.; methodology, M.G. and H.A.; software, H.A.; validation, H.A. and M.G.; formal analysis, M.G. and H.A.; investigation, M.G. and H.A.; writing—original draft preparation, M.G. and H.A.; writing—review and editing, M.G. and H.A.; visualization, H.A. and M.G.; supervision, M.G. and H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the German Research Foundation (DFG) project number 414265976-TRR 277: Additive Manufacturing in Construction and project number 390881007-EXC 2163: Sustainable and Energy-Efficient Aviation SE²A. The publication of this paper was funded by the Open Access Publication Funds of Technische Universität Braunschweig.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: A Mathematica notebook for the generation of the offset function is available as a supplement to this paper. The Mathematica code was generated with Wolfram Mathematica Version 11.3 and might not be fully compatible with other versions of Mathematica.

Acknowledgments: M.G. acknowledges financial support by the German Research Foundation (DFG) project number 414265976-TRR 277. H.A. acknowledges financial support by the German Research Foundation in the Project EXC 2163: Sustainable and Energy-Efficient Aviation SE²A. The publication of this article supported by the Open Access Publication Funds of Technische Universität Braunschweig.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Payne, B.A.; Toga, A.W. Distance field manipulation of surface models. *IEEE Comput. Graph. Appl.* **1992**, *9*, 65–71. [\[CrossRef\]](#)
2. Venkatesh, V.; Sreekanth, E. Design of mathematically defined heterogeneous porous Scaffold architecture for tissue engineering. *Int. J. Appl. Eng. Res.* **2014**, *10*, 2015.
3. Held, M.; Huber, S.; Palfrader, P. Generalized offsetting of planar structures using skeletons. *Comput.-Aided Des. Appl.* **2016**, *13*, 712–721. [\[CrossRef\]](#)
4. Pham, B. Offset curves and surfaces: a brief survey. *Comput.-Aided Des.* **1992**, *24*, 223–229. [\[CrossRef\]](#)
5. Maekawa, T. An overview of offset curves and surfaces. *Comput.-Aided Des.* **1999**, *31*, 165–173. [\[CrossRef\]](#)
6. Fayolle, P.A.; Fryazinov, O.; Pasko, A. Rounding, filleting and smoothing of implicit surfaces. *Comput.-Aided Des. Appl.* **2018**, *15*, 399–408. [\[CrossRef\]](#)
7. Shapiro, V. Semi-analytic geometry. *Acta Num.* **2007**, *16*, 239–303. [\[CrossRef\]](#)
8. Shapiro, V. Fast marching methods. *SIAM Rev.* **1999**, *41*, 199–235.
9. Huang, P.; Wang, C.C.; Chen, Y. Intersection-free and topologically faithful slicing of implicit solid. *J. Comput. Inf. Sci. Eng.* **2013**, *13*, 021009. [\[CrossRef\]](#)
10. Strano, G.; Hao, L.; Everson, R.; Evans, K. A new approach to the design and optimisation of support structures in additive manufacturing. *Int. J. Adv. Manuf. Technol.* **2013**, *66*, 1247–1254. [\[CrossRef\]](#)
11. Li, Q.; Hong, Q.; Qi, Q.; Ma, X.; Han, X.; Tian, J. Towards additive manufacturing oriented geometric modeling using implicit functions. *Vis. Comput. Ind. Biomed. Art* **2018**, *1*, 1–16. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Geier, M.; Pasquali, A.; Schönherr, M. Parametrization of the cumulant lattice Boltzmann method for fourth order accurate diffusion Part II: Application to flow around a sphere at drag crisis. *J. Comput. Phys.* **2017**, *348*, 889–898. [\[CrossRef\]](#)
13. Kimmel, R.; Bruckstein, A.M. Shape offsets via level sets. *Comput.-Aided Des.* **1993**, *25*, 154–162. [\[CrossRef\]](#)

14. Richardson, L. F. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philos. Trans. R. Soc. Lond. Ser. A* **1911**, *210*, 307–357.
15. Meeks III, W.; Pérez, J. The classical theory of minimal surfaces. *Bull. Am. Math. Soc.* **2011**, *48*, 325–407. [[CrossRef](#)]
16. Zhang, X.; Le, X.; Wu, Z.; Whiting, E.; Wang, C.C. Data-driven bending elasticity design by shell thickness. *Comput. Graph. For.* **2016**, *35*, 157–166. [[CrossRef](#)]
17. Zhang, X.; Fang, G.; Dai, C.; Verlinden, J.; Wu, J.; Whiting, E.; Wang, C.C. Thermal-comfort design of personalized casts. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, Quebec City, QC, Canada, 22–25 October 2017; pp. 243–254.
18. Wang, C. C.; Chen, Y. Thickening freeform surfaces for solid fabrication. *Rapid Prototyp. J.* **2013**, *19/6*, 395–406. [[CrossRef](#)]
19. Neudecker, S.; Bruns, C.; Gerbers, R.; Heyn, J.; Dietrich, F.; Dröder, K.; Raatz, A.; Kloft, H. A new robotic spray technology for generative manufacturing of complex concrete structures without formwork. *Procedia CIRP* **2016**, *43*, 333–338. [[CrossRef](#)]
20. Hack, N.; Kloft, H. Shotcrete 3D Printing Technology for the Fabrication of Slender Fully Reinforced Freeform Concrete Elements with High Surface Quality: A Real-Scale Demonstrator. In Proceedings of the Second RILEM International Conference on Concrete and Digital Fabrication, Eindhoven, The Netherlands, 6–8 July 2020; Bos, F.P., Lucas, S.S., Wolfs, R.J., Salet, T.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1128–1137.
21. Müller, J.; Grabowski, M.; Müller, C.; Hensel, J.; Unglaub, J.; Thiele, K.; Kloft, H.; Dilger, K. Design and parameter identification of wire and arc additively manufactured (WAAM) steel bars for use in construction. *Metals* **2019**, *9*, 725. [[CrossRef](#)]
22. Yap, C.Y.; Chua, C.K.; Dong, Z.L.; Liu, Z.H.; Zhang, D.Q.; Loh, L.E.; Sing, S.L. Review of selective laser melting: Materials and applications. *Appl. Phys. Rev.* **2015**, *2*, 041101. [[CrossRef](#)]
23. Henke, K.; Tremel, S. Wood based bulk material in 3D printing processes for applications in construction. *Eur. J. Wood Wood Prod.* **2013**, *71*, 139–141. [[CrossRef](#)]
24. Weger, D.; Lowke, D.; Gehlen, C. 3D printing of concrete structures with calcium silicate based cements using the selective binding method—effects of concrete technology on penetration depth of cement paste. In Proceedings of the Ultra-High Performance Concrete and High Performance Construction Materials, HiPerMat, Kassel, Germany, 9–11 March 2016; Volume 193.
25. Kutscher, K.; Geier, M.; Krafczyk, M. Multiscale simulation of turbulent flow interacting with porous media based on a massively parallel implementation of the cumulant lattice Boltzmann method. *Comput. Fluids* **2019**, *193*, 103733. [[CrossRef](#)]
26. Kutscher, K.; Geier, M.; Krafczyk, M. Massively Parallel Lattice Boltzmann Simulations of Turbulent Flow over and Inside Porous Media. In *Fundamentals of High Lift for Future Civil Aircraft*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 513–527.
27. Far, E.K.; Geier, M.; Kutscher, K.; Krafczyk, M. Simulation of micro aggregate breakage in turbulent flows by the cumulant lattice Boltzmann method. *Comput. Fluids* **2016**, *140*, 222–231. [[CrossRef](#)]
28. VirtualFluids. Available online: <https://www.tu-braunschweig.de/irmb/forschung/virtualfluids> (accessed on 3 November 2020).