A level set based method for fixing overhangs in 3D printing

Cacace, Simone; Cristiani, Emiliano; Rocchi, Leonardo

[Link to publication on Research at Birmingham portal](#)

Accepted Manuscript

A Level Set Based Method for Fixing Overhangs in 3D Printing

Simone Cacace, Emiliano Cristiani, Leonardo Rocchi

Please cite this article as: Simone Cacace, Emiliano Cristiani, Leonardo Rocchi, A Level Set Based Method for Fixing Overhangs in 3D Printing, *Applied Mathematical Modelling* (2017), doi: 10.1016/j.apm.2017.02.004

Highlights

- A short review of math-oriented open problems in 3D printing in presented.

- A level-set method for fixing overhangs and avoid support structures is proposed.

- The new method is particularly efficient if used with soluble filaments.

1

A Level Set Based Method for Fixing Overhangs in 3D Printing

Simone Cacace[a], Emiliano Cristiani[b], Leonardo Rocchi[c]

[a]*Dipartimento di Matematica, Sapienza – Università di Roma, Rome, Italy*
[b]*Istituto per le Applicazioni del Calcolo, Consiglio Nazionale delle Ricerche, Rome, Italy*
(corresponding author) `e.cristiani@iac.cnr.it`
[c]*School of Mathematics, University of Birmingham, Birmingham, UK*

Abstract

3D printers based on the Additive Manufacturing technology create objects layer-by-layer dropping fused material. As a consequence, strong overhangs cannot be printed because the new-come material does not find a suitable support over the last deposed layer. In these cases, one can add support structures (scaffolds) which make the object printable, to be removed at the end. In this paper we propose a level set based method to create object-dependent support structures, specifically conceived to reduce both the amount of additional material and the printing time. We also review some open problems about 3D printing which can be of interests for the mathematical community.

Keywords: level set method, Hamilton-Jacobi equations, evolving interface, support structure, scaffolding, CAD software, additive manufacturing, fused deposition modelling, digital fabrication
2010 MSC: 65D17, 35F21

1. Introduction

Is a new industrial revolution coming? Many people think so: 3D printers are able to create almost any solid object one can image and replicate existing ones. Nowadays, the price of a 3D printer is small enough to allow many people to have one at home, and create their own plastic objects. Within a decade, some products may be downloaded from the Internet for printing at home, causing a revolution in the market of such a small objects. Most important, the number of printable materials is growing and it is already possible printing an object mixing different materials. We leave to futurists

the comments about the time when 3D printers will be able to fully replicate themselves.

While the computer science literature about 3D printing is already rich in algorithms, optimization techniques and applications, the mathematical literature is basically null. This means that advanced mathematical tools based on PDEs, optimal control theory and variational methods are, so far, little explored. In order to fill the gap and promote the solution of the engineering issues related to CAD 3D printer software, in the next section we propose a bird-eye view over typical open problems encountered by practitioners who use 3D printers based on Fused Decomposition Modeling (FDM).

Main goal. The core of the paper is devoted to the solution to a particular problem, namely fixing the overhangs. When FDM technology is employed, the solid object is created layer by layer, starting from the lowest one. As a consequence, each layer can only be deposited *on top* of an existing surface, otherwise the print material falls and solidifies "in the air". In doing this, little exceptions can be handled, i.e. the upper layer can protrude over the lower layer within a certain limit. The more the material cools down rapidly and the extruder moves slowly, the more the limit can be increased. If the overhang exceeds the hardware limit, an additional support must be necessarily added, in order to make the object printable. Note that the support structures are meant to be removed at the end of the process, and thus they represent wasted material. Even more important, they represent an additional source of printing time.

Related work. The overhang problem was already investigated in the computer science and engineering literature, and some solutions were proposed [1, 7, 11, 17, 19, 21]. In most cases, support structures fill either densely or sparsely the free space encountered when a part is projected downward in its build orientation, see left object in Fig. 1(b). The difference between the methods is in how much material is used, the reliability of the supports, and the type of material which can be used. The paper [1] proposes two support geometry algorithms particularly suitable for weak support materials. The paper [7] proposes an algorithm for the automatic generation of horizontal bridges and vertical pillars, connected in such a way to create a hierarchical structure. The paper [17] uses a cone-based scan to detect the closest points which can serve as a support base (upon the model itself or the build plate) for any overhanging point. The paper [11] uses instead slant hourglass-like pillars. The paper [19] proposes to create cellular supports, riddling dense structures with holes. The paper [21] proposes an algorithm which creates

3

thin tree-like hierarchical support structures, similar (but more efficient) to the ones generated by the software Autodesk® Meshmixer® v2.9[1].

In this paper we propose to "enlarge" the object in such a way that supports are no longer needed. In particular, we avoid the creation of pillars which touch the build plate by means of optimally shaped chamfers, suitably placed below hanging parts, see right object in Fig. 1(b).

2. What every mathematician should know about 3D printing

In the context of 3D printers there exist several open problems and modelling needs. Generally speaking, the main issues come from the fact that software solutions are not object-dependent, and not change during the printing time, whereas each object (and each layer!) has its own peculiarities. An exhaustive bibliography is out of the scope of the paper, therefore for each problem we point out just a few significant references.

Infill. Printing fully solid objects is often not convenient because of the large quantity of material to be used. Shape optimization tools can give the optimal way to hollow out the object, reducing the overall material volume and keeping at the same time the desired rigidity and printable features. The problem reduces to finding the optimal inner structure supporting the whole object from the inside [23] or partitioning the object to print hollow parts [24].

Orientation & supports. In some cases the object is not 3D-printable due to the presence of hanging parts. In this case one should find the orientation of the solid which minimizes the hanging parts [8] and then the minimal amount of additional material needed to support the hanging parts. The latter problem is the one we consider in this paper.

Balancing. It is important to ensure that during the printing process (and once it is finished), the object can lie in equilibrium without falling down. This problem can be solved by trying to balancing in an appropriate way the mass of the object and by creating cavities in the inner structure so that it stands in its natural pose without requiring any glue or pedestal [4, 16].

Partitioning. Sometimes it is necessary to divide a 3D model into multiple printable pieces, so as to save the space, to reduce the printing time,

---

[1]http://www.meshmixer.com/

4

81 or to make a large model printable by small printers. This problem was at-
82 tacked by means of a level set based approach similar to the one proposed
83 here in [25].

84 **Slicing & toolpath generation.** Creating layers from a 3D model is
85 a crucial step in 3D printing. Usually one computes the intersection curves
86 between the model represented by polygonal meshes and a sequence of paral-
87 lel planes. However, this procedure is not trivial in case of very complicated
88 (self-intersecting, overlapped) objects. Moreover, once the layers are created,
89 the exact trajectory of the nozzle must be defined. The infill pattern must
90 be travelled in the shortest way, continuously, without halting the manufac-
91 turing process, and minimizing the jump from the end of one sub-path to the
92 starting point of another sub-path. Interestingly, this problem can be seen
93 as a generalized travelling salesman problem [3, 6, 9, 10, 12].

94 **Shape or shading?** 3D-printed objects replicating real objects are usu-
95 ally made of a different (and cheaper) material with respect to the original
96 ones. As a consequence, it is expected that the replicated object reflects light
97 in a different manner (different albedo, different degree of Lambertianity),
98 thus resulting in an unsatisfactory product. In some cases it can be better
99 creating an object *with different shape but which appears as the original one.*
100 In other words, one aims at replicating the reflectance properties of an object,
101 not its original shape [13].

102 **Oozing.** It can happen that the machine deposits too much material in
103 some parts of the object, or the material oozes, especially when the nozzle
104 changes direction or stays on the same point for a long time. This issue is
105 mainly related to the temperature of the nozzle's hot end and the pressure
106 drop because of the filament. The nozzle's temperature, the retraction of
107 the filament and the speed of the extruder should be related to each other
108 and optimized with respect to the printing and travelling time (i.e. extruder
109 movements with and without emission of material, respectively).

110 **Multi-material printing.** Let us also mention the possibility to print
111 objects with different materials simultaneously, alternating them while print-
112 ing. Materials can have different reflectance properties and transparency,
113 and, consequently, endless combinations are possible, as well as related op-
114 timization processes. Similarly, one can coat the surface with paint, thus
115 altering the reflectance properties [22].

5

116 **3. The level set method**

117   The level set method was introduced in [15] and since then it was success-
118 fully applied in many contexts, see e.g., [14, 18]. It allows to track Eulerianly
119 the evolution of a $(d-1)$-dimensional surface embedded in $\mathbb{R}^d$ transported
120 by a given velocity vector field $\mathbf{v} : \mathbb{R}^d \to \mathbb{R}^d$. Let us briefly recall the method
121 in the case of $d = 3$ which is of interest for our problem.

122 *3.1. The level set function and the Hamilton–Jacobi equation*

    It is given a bounded closed surface $\Sigma_0 : U \subset \mathbb{R}^2 \to \mathbb{R}^3$ at initial time
$t = 0$. We denote by Σ_t its (unknown) evolution under the action of $\mathbf{v}$ at
time t and by Ω_t the 3D domain strictly contained in Σ_t so that $\Sigma_t = \partial\Omega_t$,
for all $t \geq 0$. The main idea of the level set method stems on the definition
of a level set function $\varphi(t, x, y, z) : \mathbb{R}^+ \times \mathbb{R}^3 \to \mathbb{R}$ such that

$$\Sigma_t = \{(x, y, z) \ : \ \varphi(t, x, y, z) = 0\}, \quad \forall t \geq 0. \tag{1}$$

In this way the surface is recovered as the zero level set of φ at any time.
Initially, the function φ is chosen in such a way that

$$\varphi(0, x, y, z) \begin{cases} > 0, & \text{if } (x, y, z) \notin \overline{\Omega_0}, \\ = 0, & \text{if } (x, y, z) \in \Sigma_0, \\ < 0, & \text{if } (x, y, z) \in \Omega_0. \end{cases} \tag{2}$$

A typical choice for $\varphi(0, x, y, z)$ is the signed distance function from Σ_0,
although this choice does not lead to a smooth function. It is easy to prove
[18] that the level set function φ at any later time satisfies the following
Hamilton–Jacobi equation

$$\partial_t \varphi + \mathbf{v} \cdot \nabla\varphi = 0, \qquad t \in \mathbb{R}^+, \ (x, y, z) \in \mathbb{R}^3, \tag{3}$$

with a suitable initial condition $\varphi(0, x, y, z) = \varphi_0(x, y, z)$ satisfying (2). Here
$\nabla = (\partial_x, \partial_y, \partial_z)$ denotes the gradient with respect to the space variables.
One of the most appealing features of the level set method is that several
geometrical properties of the evolving surface can be described by means of
its level set function φ. For example, it possible to write the unit exterior
normal $\hat{\mathbf{n}}$ and the (mean) curvature κ in terms of φ and its derivatives. More
precisely, we have

$$\hat{\mathbf{n}} = \frac{\nabla\varphi}{|\nabla\varphi|} \quad \text{and} \quad \kappa = \nabla \cdot \hat{\mathbf{n}}.$$

6

If the vector field has the form $\mathbf{v} = v\hat{\mathbf{n}}$ for some scalar function v, the equation (3) turns into

$$\partial_t \varphi + v|\nabla\varphi| = 0, \qquad t \in \mathbb{R}^+, \ (x,y,z) \in \mathbb{R}^3. \tag{4}$$

₁₂₃ *3.2. Computation of the signed distance function*

₁₂₄   The computation of the signed distance function φ_0 is a problem *per se*.
₁₂₅ In our case, we can assume that the surface Σ_0 of the object to be printed
₁₂₆ is watertight and that it is given by means of a triangulation (typically in
₁₂₇ the form of a .STL file). Each triangle (facet) f is characterized by the 3D
₁₂₈ coordinates of its three vertices. Moreover, vertices are oriented in order to
₁₂₉ distinguish the internal and the external side of the facet (right-hand rule).

     Given a point $(x,y,z) \in \mathbb{R}^3$, it is easy to find the distance $d((x,y,z),f)$
between the point and the facet, so that the unsigned distance from the
surface is given by

$$d((x,y,z),\Sigma_0) = \min_f d((x,y,z),f).$$

₁₃₀ The computation of the distance's sign is more tricky since one has to check
₁₃₁ if the point is internal or external to the surface. Several methods can be
₁₃₂ employed here. For example, one can note that the solid angle subtended by
₁₃₃ the whole surface at a given point is maximal and equal to 4π iff the point is
₁₃₄ internal. Then, one can sum all the solid angles subtended by the facets at
₁₃₅ the point and check if it equals 4π. If this is the case, the point is internal
₁₃₆ to the surface, otherwise it is external.

₁₃₇   Note that the solid angle itself should be *signed*, in the sense that it
₁₃₈ must be positive if the point looks at the internal part of the facet, negative
₁₃₉ otherwise. A nice algorithm to compute the signed solid angle between a
₁₄₀ point and a triangle was given by van Oosterom and Strackee [20].

₁₄₁ **4. Fixing overhangs**

₁₄₂   In this section we propose a solution for fixing the overhang issue in 3D
₁₄₃ printing and, in most cases, getting rid of long support structures extended
₁₄₄ until the build plate. We want to use the level set method by considering the
₁₄₅ domain Ω as the object to be printed and its surface Σ as an evolving front.
₁₄₆ Therefore, the idea is to modify the initial unprintable object Ω_0 letting it
₁₄₇ evolve by an *ad hoc* vector field $\mathbf{v}$ until it becomes fully printable, meaning

7

148 that there are no more unprintable hanging parts. The final object Ω_* is
149 then actually printed and the difference $\Omega_* \backslash \Omega_0$ is finally removed. Note that
150 the difference $\Omega_* \backslash \Omega_0$ can be easily identified by standard techniques and
151 consequently printed with a different material (e.g., a soluble filament) or
152 with a different printing resolution.

It is useful to divide the surface Σ of the object Ω in three subsets, on
the basis of their *printability*. To this end, we denote by $\hat{\mathbf{g}} = (0, 0, -1)$ the
unit gravity vector, and again by $\hat{\mathbf{n}}(x, y, z)$ the exterior unit normal to the
surface Σ of the object Ω at the point (x, y, z). Moreover, let

$$\theta(\hat{\mathbf{n}}) := \arccos\left(\hat{\mathbf{g}} \cdot \hat{\mathbf{n}}\right) \tag{5}$$

153 be the angle between $\hat{\mathbf{g}}$ and $\hat{\mathbf{n}}$.

154 **Definition 4.1.** *A point (x, y, z) of the surface Σ is said to be*

$$
\begin{array}{ll}
\textit{unprintable,} & \textit{if } \theta \in [0, \bar{\alpha}) \cup (2\pi - \bar{\alpha}, 2\pi], \\
\textit{safe,} & \textit{if } \theta \in [\pi/2, 3\pi/2], \\
\textit{modifiable,} & \textit{otherwise,}
\end{array}
$$

156 *where $\bar{\alpha}$ is a given limit angle[2], see Fig. 1(a).*

157 While the first two definitions are immediately clear, it is worth to spend
158 some words on the third one. Modifiable points are indeed printable since
159 the overhang is sufficiently small. On the other hand, it could be convenient
160 to move those points as well in order to make printable the unprintable ones.
161 This guarantees a sufficient flexibility to shape the object conveniently and
162 not to create long supports like the one depicted on the left of Fig. 1(b). We
163 can extend Definition 4.1 by saying that the set of both modifiable and safe
164 point constitute the overall printable points.

165 The rest of the section will be devoted to the construction of the vector
166 field $\mathbf{v}$. A suitable choice is the one used in equation (4) where $\mathbf{v} = v\hat{\mathbf{n}}$ for
167 some scalar function v, possibly depending on $\hat{\mathbf{n}}$ and κ.

In the following we denote by

$$P(\omega) := \omega^+ \quad \text{and} \quad M(\omega) := \omega^-, \qquad \omega \in \mathbb{R},$$

---

[2]Typically $\bar{\alpha} = \frac{\pi}{4}$, because of the so-called 45 degree rule, though it actually depends
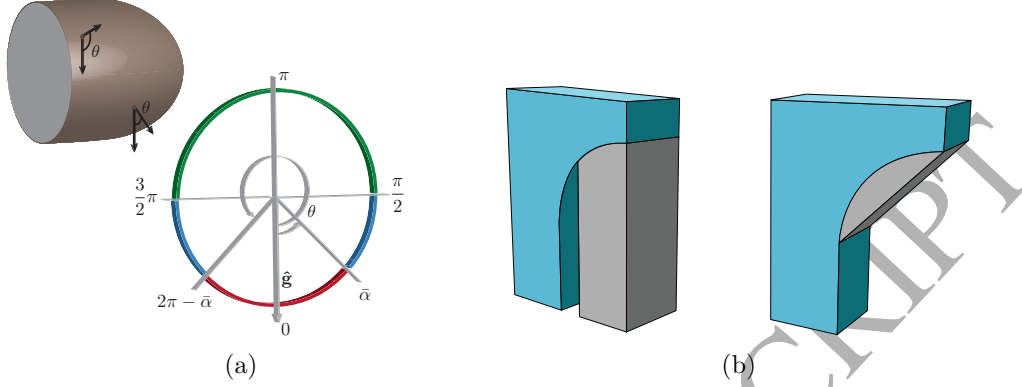on the 3D printer settings, print material, cooling, etc.

Figure 1: (a) Unprintable (red), modifiable (blue) and safe (green) points with respect to the counter-clockwise angle θ between the gravity $\hat{\mathbf{g}}$ and the normal $\hat{\mathbf{n}}$. Modifiable and safe points are printable. (b) The left grey support wastes a lot of material contrary to the chamfer on the right that saves more material and keeps the printability of the overhang as well.

¹⁶⁸ the positive and negative part, respectively.

¹⁶⁹ **Positivity and build plate.** We need to be guaranteed that $\Omega_0 \subseteq \Omega_t \subseteq \Omega_*$,
¹⁷⁰ for all $t \geq 0$, since once the object is printed we can remove material but not
¹⁷¹ add new one. This is why we need $v \geq 0$, i.e. the movement of each point of
¹⁷² the surface Σ has to be along the normal exterior direction $\hat{\mathbf{n}}$. Furthermore,
¹⁷³ the object cannot move under the build plate, supposed at a fixed $z = z_{\min} \in$
¹⁷⁴ $\mathbb{R}$. Then we impose $v = 0$ if $z \leq z_{\min}$.

Movement of unprintable points. We introduce the term

$$v_1(\hat{\mathbf{n}}; \bar{\alpha}) := P(\cos\theta(\hat{\mathbf{n}}) - \cos\bar{\alpha}), \tag{6}$$

¹⁷⁵ which lets the unprintable points move outward. The speed is higher when-
¹⁷⁶ ever θ is close to 0, which represents the (hardest) case of a horizontal hanging
¹⁷⁷ part.

¹⁷⁸ **Rotation.** It is convenient introducing a rotational effect in the evolution
¹⁷⁹ which avoids the unprintable regions to evolve "as it is" until they touch the
¹⁸⁰ build plate. To this end we introduce the term $(z_{\max} - z)$, where $z_{\max} \in \mathbb{R}$
¹⁸¹ is the maximal height reached by the object. This term simply increases the
¹⁸² speed of lower points with respect to higher ones. This makes the lower parts
¹⁸³ be resolved (or eventually touch the built plate) before the higher parts, thus
¹⁸⁴ saving material.

9

Movement of modifiable points. Modifiable points are moved, if necessary, by means of the following term in the vector field

$$v_2(\kappa) := M(\kappa). \tag{7}$$

185 It moves outward the points with negative curvature until it vanishes, i.e.
186 the surface is locally flat. In particular, it moves concave corners and let
187 modifiable points become a suitable support for the still unprintable points
188 above.

189 **Blockage of safe points.** Finally, it is necessary to exclude from the evolu-
190 tion the safe points of the object. In order to identify them, we use the sign
191 of the third component n_3 of the unit exterior normal vector $\hat{\mathbf{n}}$.

By putting together all the terms we end up with

$$v(x, y, z, \hat{\mathbf{n}}, \kappa; \bar{\alpha}) :=$$
$$\begin{cases} C_1 (z_{\max} - z)v_1(\hat{\mathbf{n}}; \bar{\alpha}) + C_2 \, v_2(\kappa), & \text{if } n_3 < 0 \text{ and } z > z_{\min}, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

192 with $C_1, C_2 > 0$ positive constants (model parameters). The result expected
193 from a such vector field is an evolution similar to the one depicted on the
194 right in Fig. 1(b), corresponding to a support whereby the angle θ in each of
195 its point is less or equal to $\bar{\alpha}$.

196 The surface evolution relative to equation (4) must be stopped at some
197 final time $T > 0$. Rather than waiting that the velocity field vanishes com-
198 pletely, it is convenient to check directly (at every time $t < T$) whether the
199 overall surface is printable or not, according to Definition 4.1. More precisely,
200 we stop the evolution when all the points belonging to the zero level set are
201 safe or modifiable, i.e., printable.

202 **Remark 4.1.** (Optimality of the final surface) By construction, the surface
203 always evolves towards a printable object. Indeed, any non-printable part
204 of the surface is forced to move downward, and the surface has to stop once
205 the build plate is reached. Nevertheless, we have no guarantee that the final
206 object is "optimal" in terms of additional printing material. In the worst-
207 case scenario the surface evolves until it touches the build plate, obtaining
208 something similar to the results depicted on the left in Fig. 1(b). For in-
209 stance, this is the case of a perfectly symmetric bridge-shaped object, unless
210 some symmetry-breaking terms are added in the evolution model. Likely, the
211 method works fine in most cases, as one can see in section 6, where several
212 objects are tested.

10

213 5. Theoretical analysis

214 In this section we show that a slightly regularised version of the Hamilton–
215 Jacobi equation (4) with velocity field (8) fits the classical theory of viscosity
216 solutions and it is then well-posed.

Consider a general second order PDE of the form

$$\varphi_t + F(t, \mathbf{x}, \varphi, \nabla\varphi, \mathbf{H}\varphi) = 0, \qquad t > 0, \quad \mathbf{x} \in \mathbb{R}^n, \tag{9}$$

where $\mathbf{H}\varphi$ is the Hessian matrix of φ and $F : \mathbb{R}^+ \times \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n \times \mathcal{S}_n \to \mathbb{R}$ is continuous and $\mathcal{S}_n$ is the set of symmetric $n \times n$ matrices. Resorting to classical results [5], we can say that the IVP for (9) is well-posed if the function F is *proper* for any fixed $t \in [0, T]$, i.e.

$$\forall t \quad F(t, \mathbf{x}, r, \mathbf{p}, \mathbf{X}) \le F(t, \mathbf{x}, s, \mathbf{p}, \mathbf{Y}) \qquad \text{whenever } r \le s \text{ and } \mathbf{Y} \le \mathbf{X}. \tag{10}$$

217 Before writing our equation in the form (9), let us note that $M(\omega) = P(-\omega)$
218 for all $\omega \in \mathbb{R}$, and $M(c\omega) = cM(\omega)$, for all $\omega \in \mathbb{R}$ and $c > 0$. Moreover, let
219 $H : \mathbb{R} \to \{0, 1\}$ be the Heaviside function and state the following equality:

Lemma 5.1. Given any function $u \in C^2(\mathbb{R}^n; \mathbb{R})$, we have

$$\operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right)|\nabla u| = \operatorname{trace}\left(\left(\mathbf{I} - \frac{\nabla u \otimes \nabla u}{|\nabla u|^2}\right)\mathbf{H}u\right),$$

220 where $\mathbf{I}$ is the $n \times n$ identity matrix and $(\mathbf{a} \otimes \mathbf{b})_{i,j} = a_i b_j$ for all $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$
221 and $i, j = 1, \ldots, n$.

222 The proof of the Lemma is postponed in the Appendix.

Making explicit the dependence on φ, we can rewrite the speed as

$$v(z, \nabla\varphi, \mathbf{H}\varphi; \bar{\alpha}) =$$
$$\left[C_1(z_{\max} - z)M\left(\frac{\partial_z\varphi}{|\nabla\varphi|} + \cos\bar{\alpha}\right) + C_2M\left(\operatorname{div}\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right)\right)\right]H(-\partial_z\varphi)H(z - z_{\min}).$$

Note that in our case the function v does not depend explicitly on t and φ, but depends implicitly on $\mathbf{H}\varphi$ by means of the divergence operator. In our

11

case we have $F = v|\nabla\varphi|$, and then, using Lemma 5.1, we have

$$
F(z, \nabla\varphi, \mathbf{H}\varphi; \bar{\alpha}) =
$$
$$
\left[C_1(z_{\max} - z)M\left(\partial_z\varphi + |\nabla\varphi|\cos\bar{\alpha}\right) + \right.
$$
$$
\left. C_2 M\left(\mathrm{div}\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right)|\nabla\varphi|\right) \right] H(-\partial_z\varphi)H(z - z_{\min}) =
$$
$$
\left[C_1(z_{\max} - z)M\left(\partial_z\varphi + |\nabla\varphi|\cos\bar{\alpha}\right) + \right.
$$
$$
\left. C_2 M\left(\mathrm{trace}\left(\left(\mathbf{I} - \frac{\nabla\varphi \otimes \nabla\varphi}{|\nabla\varphi|^2}\right)\mathbf{H}\varphi\right)\right) \right] H(-\partial_z\varphi)H(z - z_{\min}).
$$

Following again [5] (Example 1.2), and considering the sign of the function M, we are left to prove that the matrix $\left(\mathbf{I} - \frac{\mathbf{p} \otimes \mathbf{p}}{|\mathbf{p}|^2}\right)$, $\mathbf{p} \in \mathbb{R}^3$, is positive semi-definite. A straightforward computation shows that, for all vectors $(x_1, x_2, x_3) \neq (0, 0, 0)$,

$$
(x_1\ x_2\ x_3)\ \left(\mathbf{I} - \frac{\mathbf{p} \otimes \mathbf{p}}{|\mathbf{p}|^2}\right)\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} =
$$
$$
\frac{1}{|\mathbf{p}|}\left[(x_1 p_2 - x_2 p_1)^2 + (x_1 p_3 - x_3 p_1)^2 + (x_2 p_3 - x_3 p_2)^2\right] \geq 0.
$$

This proves that F is proper. In order to entirely fit the theoretical framework we should guarantee the continuity of the function F, although this is not expected to be a crucial point from the numerical point of view, since continuity cannot be actually satisfied a discrete level. An easy solution is the mollification of the Heaviside function by convolution, which makes F be continuous.

6. Numerical tests

We solve equation (4) with velocity (8) by using a monotone upwind scheme based on finite differences as described in [18, Sect. 6.4], with an adaptive time step in order to strictly satisfy the CFL condition.

12

²³³ As a preliminary test, we solved a dimension-reduced problem by con-
²³⁴ sidering a 2D interface Σ with two hanging parts as the zero level set of a
²³⁵ specific level set function $\varphi : \mathbb{R}^+ \times \mathbb{R}^2 \to \mathbb{R}$. The computational domain is
²³⁶ $[0,6] \times [0,10]$, divided in 120×200 regular grid nodes. Parameters are $C_1 = 6$
²³⁷ and $C_2 = 0.4$. Initial and final shapes of the interface are shown in Fig. 2(a).
²³⁸ Moreover, by "extruding" the 2D domain Ω, as it was a section of a real 3D
²³⁹ object, we printed it out with the supports created from our method (Fig.
²⁴⁰ 2(b)), and keeping the scaffolding structure created by the commercial soft-
²⁴¹ ware Cura v15.04.2 (Fig. 2(c)). Finally, Fig. 2(d) shows the support structure
generated by the commercial software Autodesk® Meshmixer® v2.9.
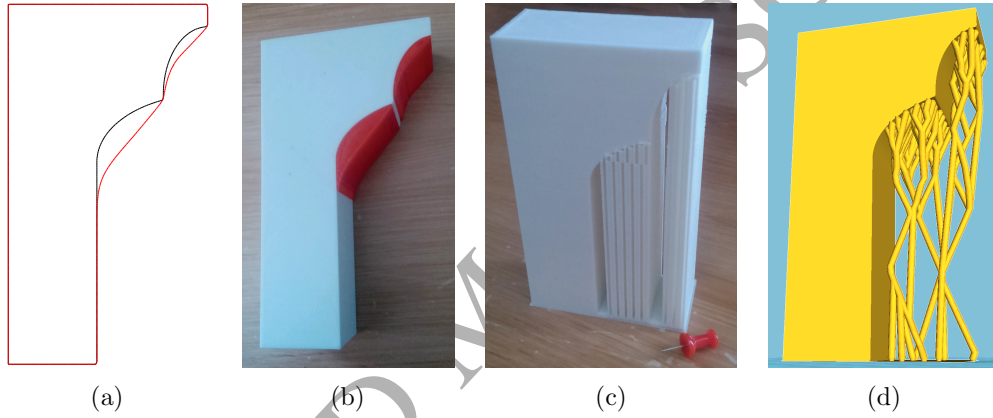

(a)   (b)   (c)   (d)

Figure 2: 2D test. (a) Initial contour Σ_0 (black) and the optimized one Σ_* (red) after
the evolution. (b) Printed model with proposed support. (c) Printed model with sup-
port structure generated by free software Cura v15.04.2. (d) Tree-like support structure
generated by Autodesk® Meshmixer® v2.9.

²⁴²
²⁴³ Moving to real 3D problems, we tested eight objects. In all cases the
²⁴⁴ computational domain $[-2,2]^3$ is divided in 100^3 regular grid nodes. The
²⁴⁵ first two examples, a sphere and a cross, are shown in Fig. 3(a,b). They have
²⁴⁶ been easily obtained as the zero level set of a corresponding hyper-surface
²⁴⁷ embedded in $\mathbb{R}^4$ and no .STL files have been required. The parameters used
²⁴⁸ for the evolution are $C_1 = 0.7$, $C_2 = 0.3$ for the sphere and $C_1 = 1.5$, $C_2 = 0.5$
²⁴⁹ for the cross.
²⁵⁰ These simple numerical tests clearly show the advantage of the proposed
²⁵¹ approach: the additional material used to make the object printable is rather
²⁵² minimal, being concentrated in the critical zones. No evident waste of mate-
²⁵³ rial is visible. More precisely, we see that the additional material is limited
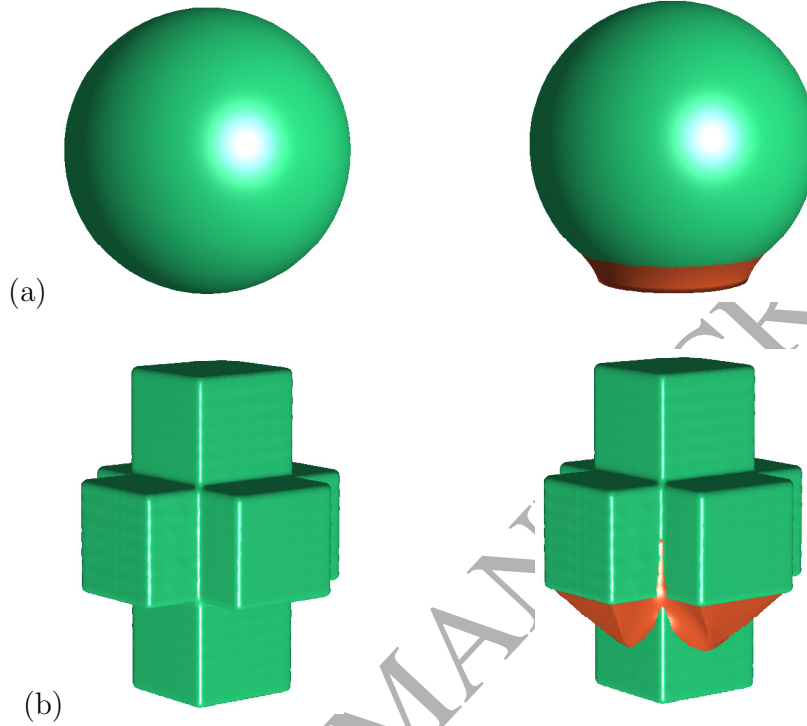
13

(a)

(b)

Figure 3: Two simple objects. Initial surface (left) and final result (right).

²⁵⁴ to the quantity needed to support overhangs within the maximum allowed
²⁵⁵ slope.

²⁵⁶ The following three examples are shown in Fig. 4(a,b,c). In this case, we
²⁵⁷ obtained the objects as the zero level set of the distance function (see section
²⁵⁸ 3.2) for the corresponding surface given as .STL file. The parameters used
²⁵⁹ for the evolution are (a) $C_1 = 0.8$, $C_2 = 0.8$, (b) $C_1 = 1.8$, $C_2 = 0.6$, and (c)
²⁶⁰ $C_1 = 0.6$, $C_2 = 1.2$.

²⁶¹ Again we see that the additional material is rather minimal and concen-
²⁶² trated in the critical zones.

²⁶³ The last three examples are shown in Fig. 5(a,b,c). In this case we tried
²⁶⁴ to fix overhangs of some mechanical components, starting again from the
²⁶⁵ corresponding .STL files. The parameters used for the evolution are (a)
²⁶⁶ $C_1 = 3.5$, $C_2 = 0.9$, (b) $C_1 = 1.2$, $C_2 = 1.6$, and (c) $C_1 = 5.6$, $C_2 = 0.3$.

²⁶⁷ In the first two cases the evolution is close to the optimal one since no
²⁶⁸ evident waste of material is visible. In the last case instead, the horizontal
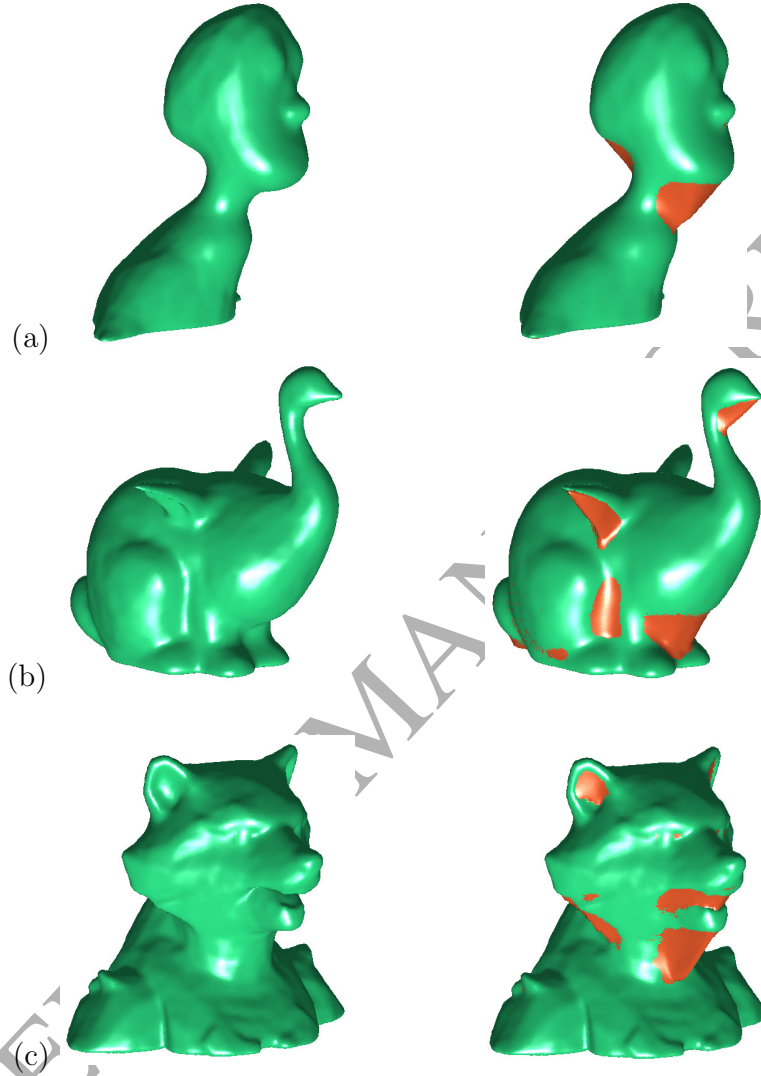
14

Figure 4: Three objects. Initial surface (left) and final result (right).

₂₆₉ holes on the overhanging part are a challenge for the proposed method. They
₂₇₀ hinder the rotation and make the surface evolve vertically until the build plate
₂₇₁ is reached, thus realizing the worst-case scenario discussed in Remark 4.1.
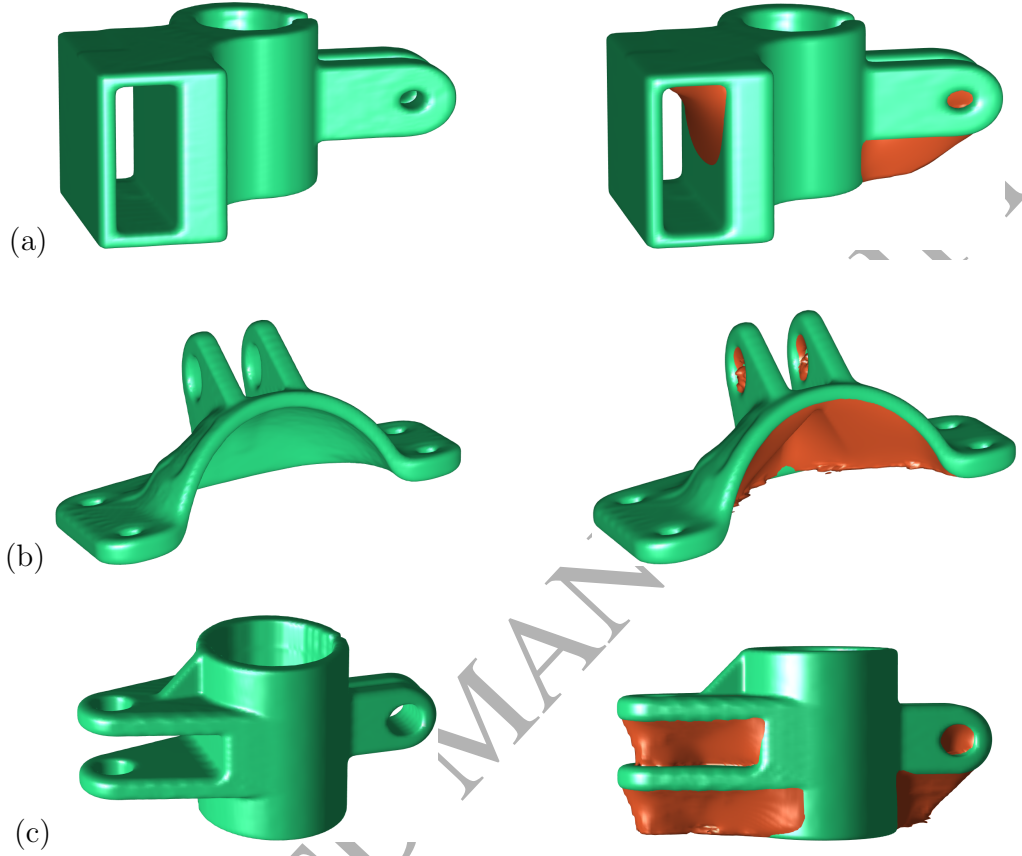
15

Figure 5: Three brackets. Initial surface (left) and final result (right).

Conclusions and future work

We have introduced a level set based method to create *ad hoc* chamfers in additive manufacturing, avoiding in most cases the use of classical vertical support structures. Moreover, in the worst-case scenario the evolved surface will not be worse than the one obtained with commercial software.

The main drawback of the proposed approach is that objects with small or sharp details clearly require a quite fine computational grid, thus rising the CPU time. Note that this is not a limitation of the evolution model, rather a limitation of the level set method itself.

Let us also stress here that we consider only the overhanging issue in printability, although the use of the support structure is not just for overhangs. As recalled in section 2, it also keeps the whole model from falling

16

²⁸⁴ because of the gravity. Moreover, for printers that use the same material for
²⁸⁵ both the model and the support structure, removing the support structure
²⁸⁶ from the model can become difficult too. For all these reasons the proposed
²⁸⁷ method is more suitable for printers that use different materials for the sup-
²⁸⁸ port structure, such as polyjet printers.

²⁸⁹ We hope that this study can pave the way to shape optimization methods
²⁹⁰ based on the coupling of the level set method and the shape derivatives [2].
²⁹¹ In that context one could minimize directly the printing time and at the same
²⁹² time penalize the contact between the desired object and the removable parts,
²⁹³ in order to simplify the final detaching operations.

²⁹⁷ Appendix A. Proof of Lemma 5.1

Proof. First of all, it is useful to recall a basic property of the trace operator. Let us consider two $n \times n$ matrices $\mathbf{A}$ and $\mathbf{B}$, and define $\mathbf{C} := \mathbf{AB}$. Denote by $\mathbf{a}_{i,\cdot}$ the i-th row of $\mathbf{A}$ and by $\mathbf{b}_{\cdot,j}$ the j-th column of $\mathbf{B}$. We have

$$\text{trace}(\mathbf{C}) \overset{\text{def}}{=} \sum_i c_{i,i} = \sum_i \mathbf{a}_{i,\cdot} \cdot \mathbf{b}_{\cdot,i}^{\intercal} = \sum_i \sum_j a_{i,j}\, b_{j,i}. \tag{A.1}$$

17

The Lemma is proved as follows:

$$|\nabla u|\text{div}\left(\frac{\nabla u}{|\nabla u|}\right) = |\nabla u|\sum_{i=1}^{n}\partial_i\left(\frac{\partial_i u}{|\nabla u|}\right) =$$

$$|\nabla u|\frac{1}{|\nabla u|^2}\sum_{i=1}^{n}\left(\partial_i^2 u|\nabla u| - \partial_i u\frac{1}{|\nabla u|}\sum_{j=1}^{n}\partial_j u\ \partial_i\partial_j u\right) =$$

$$\triangle u - \frac{1}{|\nabla u|^2}\sum_{i=1}^{n}\partial_i u\sum_{j=1}^{n}\partial_j u\ \partial_i\partial_j u =$$

$$\triangle u - \frac{1}{|\nabla u|^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\nabla u\otimes\nabla u)_{i,j}\ \partial_i\partial_j u =$$

$$\text{trace}(\mathbf{H}u) - \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{(\nabla u\otimes\nabla u)_{i,j}}{|\nabla u|^2}\ \partial_j\partial_i u \stackrel{\text{(A.1)}}{=}$$

$$\text{trace}(\mathbf{H}u) - \text{trace}\left(\frac{\nabla u\otimes\nabla u}{|\nabla u|^2}\mathbf{H}u\right) =$$

$$\text{trace}\left(\left(I - \frac{\nabla u\otimes\nabla u}{|\nabla u|^2}\right)\mathbf{H}u\right).\quad \square$$

References

[1] E. Barnett and C. Gosselin. Weak support material techniques for alternative additive manufacturing materials. *Additive Manufacturing*, 8:95–104, 2015.

[2] M. Burger and S. J. Osher. A survey on level set methods for inverse problems and optimal design. *Eur. J. Appl. Math.*, pages 263–301, 2005.

[3] K. Castelino, R. D'Souza, and P. K. Wright. Toolpath optimization for minimizing airtime during machining. *Journal of Manufacturing Systems*, 22(3):173–180, 2003.

[4] A. N. Christiansen, R. Schmidt, and J. A. Bærentzen. Automatic balancing of 3D models. *Computer-Aided Design*, 58:236–241, 2015.

[5] M. G. Crandall, H. Ishii, and P.-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc.*, 27(1):1–67, 1992.

18

[6] S. Dhanik and P. Xirouchakis. Contour parallel milling tool path generation for arbitrary pocket shape using a fast marching method. *Int. J. Adv. Manuf. Technol.*, 50(9-12):1101–1111, 2010.

[7] J. Dumas, J. Hergel, and S. Lefebvre. Bridging the gap: automated steady scaffoldings for 3D printing. *ACM Trans. Graph.*, 33(4):Article No.98(10 pages), 2014.

[8] B. Ezair, F. Massarwi, and G. Elber. Orientation analysis of 3D objects toward minimal support volume in 3D-printing. *Computers & Graphics*, 51:117–124, 2015.

[9] K. Hildebrand, B. Bickel, and M. Alexa. Orthogonal slicing for additive manufacturing. *Computers & Graphics*, 37(6):669–675, 2013.

[10] P. Huang, C. C. L. Wang, and Y. Chen. Intersection-free and topologically faithful slicing of implicit solid. *J. Comput. Inf. Sci. Eng.*, 13(2):021009, 2013.

[11] X. Huang, C. Ye, S. Wu, K. Guo, and J. Mo. Sloping wall structure support generation for fused deposition modeling. *Int. J. Adv. Manuf. Technol.*, 42:1074–1081, 2009.

[12] Y.-A. Jin, Y. He, J.-Z. Fu, W.-F. Gan, and Z.-W. Lin. Optimization of tool-path generation for material extrusion-based additive manufacturing technology. *Additive Manufacturing*, 1-4:32–47, 2014.

[13] Y. Lan, Y. Dong, F. Pellacini, and X. Tong. Bi-scale appearance fabrication. *ACM Trans. Graph.*, 32(4):Article No.145(12 pages), 2013.

[14] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2003.

[15] S. Osher and J. A. Sethian. Front propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[16] R. Prévost, E. Whiting, S. Lefebvre, and O. Sorkine-Hornung. Make it stand: balancing shapes for 3D fabrication. *ACM Trans. Graph.*, 32(4):Article No.81(10 pages), 2013.

[17] J. Qiu, L. Wu, and Y. Mao. A novel supporting structure generation scheme to 3D printing. In *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, ICIMCS '15, pages 69:1–69:4, New York, NY, USA, 2015. ACM.

[18] J. A. Sethian. *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and material science.* Cambridge University Press, New York, 1999.

[19] G. Strano, L. Hao, R. M. Everson, and K. E. Evans. A new approach to the design and optimisation of support structures in additive manufacturing. *Int. J. Adv. Manuf. Technol.*, 66:1247–1254, 2013.

[20] A. van Oosterom and J. Strackee. The solid angle of a plane triangle. *IEEE Transactions on Biomedical Engineering*, BME-30(2):125–126, 1983.

[21] J. Vanek, J. A. G. Galicia, and B. Benes. Clever support: efficient support structure generation for digital fabrication. *Comput. Graph. Forum*, 33(5):117–125, 2014.

[22] K. Vidimče, S.-P. Wang, J. Ragan-Kelley, and W. Matusik. OpenFab: a programmable pipeline for multi-material fabrication. *ACM Trans. Graph.*, 32(4):Article No.136(12 pages), 2013.

[23] W. Wang, T. Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, and X. Liu. Cost-effective printing of 3D objects with skin-frame structures. *ACM Trans. Graph.*, 32(6):Article No.177(10 pages), 2013.

[24] X.-R. Wei, Y.-H. Zhang, and G.-H. Geng. No-infill 3d printing. *3D Res.*, 7(24), 2016.

[25] M. Yao, Z. Chen, L. Luo, R. Wang, and H. Wang. Level-set-based partitioning and packing optimization of a printable model. *ACM Trans. Graph.*, 34(6):Article No.214(11 pages), 2015.

20