Article

CAD/CAM System for Additive Manufacturing with a Robust and Efficient Topology Optimization Algorithm Based on the Function Representation

Dmitry Popov , Yulia Kuzminova, Evgenii Maltsev , Stanislav Evlashin , Alexander Safonov , Iskander Akhatov and Alexander Pasko

Center for Design, Manufacturing and Materials, Skolkovo Institute of Science and Technology, 30/1 Bolshoi Boulevard, 121205 Moscow, Russia; Yulia.Kuzminova@skoltech.ru (Y.K.); E.Maltsev@skoltech.ru (E.M.); S.Evlashin@skoltech.ru (S.E.); a.safonov@skoltech.ru (A.S.); I.Akhatov@skoltech.ru (I.A.); A.Pasko@skoltech.ru (A.P.)

* Correspondence: dmitry.popov@skoltech.ru

Abstract: Additive manufacturing erases the distance between design ideas and finished parts. However, designers must use several software tools to use these advantages. Moreover, these tools operate with different representations of geometry. This paper describes the architecture of a new CAD/CAM system that uses only the function representation of the geometry (FRep). It provides all widely used design operations and allows an engineer to employ robust and efficient topology optimization algorithms. The developed CAD/CAM system consists of 3D modeling, simulation, topology optimization, and direct manufacturing modules. We successfully printed designed parts and performed mechanical tests of printed parts. The results of tests show good agreement with simulation data. The system makes it possible to create structures with the desired properties in a fast and flexible way. The proposed approach significantly helps in designing additive manufacturing process and saves time for its users.

Keywords: function representation; level-set methods; topology optimization; CAD/CAM; additive manufacturing; 3D printing

Citation: Popov, D.; Kuzminova, Y.; Maltsev, E.; Evlashin, S.; Safonov, A.; Akhatov, I.; Pasko, A. CAD/CAM System for Additive Manufacturing with a Robust and Efficient Topology Optimization Algorithm Based on the Function Representation. *Appl. Sci.* **2021**, *11*, 7409. <https://doi.org/10.3390/app11167409>

Academic Editor: Ana M. Camacho

Received: 20 July 2021

Accepted: 9 August 2021

Published: 12 August 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

The design stage in the manufacturing process has become more complex in recent decades [1]. Designers try to create details that fully correspond to their intended functionality. This translates into certain mass limits and stiffness requirements to designed parts. Simulation and topology optimization algorithms help obtain the geometry satisfying these constraints. However, topology optimization algorithms can produce geometries that are impossible with traditional production methods but suitable for additive manufacturing (AM) [2]. These technologies impose new requirements to computer-aided design (CAD) and computer-aided manufacturing (CAM) software. Moreover, they significantly increase the complexity of CAD models that can be manufactured, and clearly show the limitations of boundary representation (BRep) used in traditional CAD/CAM systems. Application of multiple set-theoretic and other operations, varying the parameters of BRep models, leads to corruption of meshes and significantly increases the size of files with resulting geometry for models in high precision applications [3]. The use of BRep leads to occurrence of incorrect faces in exported STL files or errors in the slicing process. Moreover, when a design includes the analysis or topology optimization steps, the traditional approach uses the following geometry transformations. First, the BRep CAD geometry is transformed to the voxel representation, and then analysis and topology optimization is performed. Second, the BRep is reconstructed with possible smoothing. Finally, the resulting geometry is sliced to create a computer numerical control (CNC) program in a specific language such as G-code [4]. Several software packages usually serve these purposes.

The use of the FRep-based CAD/CAM system instead resolves all previously mentioned problems of the BRep format. In contrast to BRep, function representation (FRep) guarantees the correctness of resulting geometry regardless of the number of set-theoretic and other operations and the number of parameters in the model [5]. The paper [6] presents the fundamentals of the FRep. It supports many modeling operations. The basis of set-theoretic operations are R-functions [7]. Operations on FRep objects have become the subject of several studies, including blending, bending, offsetting, sweeping [8], bounded blending, trimming [9], metamorphosis, etc. [10]. The mentioned theory of R-functions was initially developed as a tool for domains modeling in partial differential equations (PDE) [7]. Later publications elaborated this approach [11] and proposed an algorithm for topology optimization of FRep models [12]. Many algorithms employ the FRep for domain description in various simulation and structural optimization tasks [13,14]. These algorithms form a group called level-set methods for topology optimization. The advantage of such optimization algorithms is their ability to directly provide the optimal geometry in contrast to 0–1 density methods such as the solid isotropic material with penalization (SIMP) method [14]. FRep-based optimization algorithms provide ready-to-print shapes in contrast to BRep-based methods requiring post-processing steps, e.g., smoothing the geometry, leading to remodeling of the body. Level-set methods for multi-material topology optimization exist [15]. For example, they can be applied for designing of continuous fiber-reinforced structures instead of existing topology optimization methods [16,17]. Thus, FRep can be applied to solve various tasks; however, no complete FRep-based CAD/CAM system has yet been proposed.

The presented work has two contributions. The first is the aggregation of FRep and level-set methods in one CAD/CAM system. Both FRep and level-set methods operate with continuous functions for describing domains. It allowed us to develop the first framework that provides tools for topology optimization and uses its results as modeling primitives for further design. The second contribution is the modification of the level-set-based topology optimization algorithm. These modifications increased the speed of the algorithm up to 90 times and improved its stability. This paper describes a CAD/CAM system that uses FRep for all modeling, analysis, topology optimization, and direct manufacturing operations. The system was tested by 3D printing of a designed part. The part was obtained by structural optimization of a rectangular domain with the simulation of its mechanical properties. The simulated properties were experimentally validated.

2. Software Structure and Methods

We propose the FRep-based CAD/CAM system that makes it possible to rapidly design and print optimized parts. Figure 1 shows the principal components of the FRep system. The key part of the developed system is an FRep kernel connecting all modules of the system. An FRep defining function sets up the classification rule to obtain boundary segments of the body with the given precision. The FRep kernel supports the import and export of models from other CAD systems. Data from this component are used both in direct manufacturing routines and in visualization. Moreover, the defining function is processed as a level-set function in optimization and analysis algorithms.

The next essential component of the system is the FRep CAM module. This component is responsible for interactions with additive manufacturing equipment. Its main routine is slicing of FRep models. However, this procedure can take hours in high precision applications where the slice thickness is 10,000 times smaller than the size of a model. Techniques described in [18] can speed up the contouring step but they do not work well in the case of models with high porosity and defining functions with loops. The post-processing follows the contouring step. The post-processor generates G-code to control AM equipment [19]. Let us consider all mentioned components of the system in more details.

Figure 1. The principal components of the FRep system.

2.1. Topology Optimization

The developed system uses the modified topology optimization algorithm presented in [20]. For the 2D topology optimization case the compliance minimization problem used in the system can be expressed as follows:

$$\begin{aligned}
 &\text{minimize(over } f) J_0(u) = \int_D E_{ijkl} \varepsilon_{ij}(u) \varepsilon_{kl}(u) H(f) d\Omega, \\
 &\text{subject to: } a(u, v, f) = l(v, f), \quad \forall v \in U, \\
 &\quad u|_{\Gamma_1} = u_0, \\
 &\quad E_{ijkl} \varepsilon_{ij}(u)|_{\Gamma_2} = p, \\
 &\quad \int_D H(f) d\Omega \leq V_0,
 \end{aligned}$$

where

$$\begin{aligned}
 a(u, v, f) &= \int_D E_{ijkl} \varepsilon_{ij}(u) \varepsilon_{kl}(v) H(f) d\Omega, \\
 l(v, f) &= \int_D g v + \nabla(p v n) H(f) d\Omega, \\
 H(f) &= \begin{cases} 1 & \text{if } f \geq 0, \\ 0 & \text{if } f < 0, \end{cases}
 \end{aligned} \tag{1}$$

$J_0(u)$ is an elastic energy, the objective function; u —the displacement field; $\varepsilon_{ij}(u)$ and $\varepsilon_{kl}(u)$ —the linearized strain tensors; E_{ijkl} —the Hook elasticity tensor; u_0 , p and g —the constant values that represent the given displacement, traction and body force respectively; $a(u, v, f)$ is the linear elastic equilibrium equation written in its weak, variational form and $l(v, f)$ is the load linear form, with v denoting a virtual displacement field in the space U of kinematically admissible displacement fields; V_0 is the maximum allowable volume in the design domain D ; $H(f)$ is the Heaviside function that represents the presence of material at a point of the level-set function value f :

$$H(f) = \begin{cases} 1 & \text{if } f \geq 0 \\ 0 & \text{if } f < 0 \end{cases} \tag{2}$$

This problem statement uses the level-set function f which, at the same time, is an FRep defining function $f(x_1, x_2, \dots, x_n)$ (see Section 2.2):

$$f(x_1, x_2, \dots, x_n) \begin{cases} > 0, & \text{if } (x_1, x_2, \dots, x_n) \in \Omega \\ = 0, & \text{if } (x_1, x_2, \dots, x_n) \in \partial\Omega \\ < 0, & \text{if } (x_1, x_2, \dots, x_n) \notin \overline{\Omega} \end{cases} \quad (3)$$

where f is a real continuous function defined on E^n , n equals 2 for 2D, and 3 for 3D topology optimization, respectively.

Let us consider the 2D optimization. The proposed topology optimization algorithm incorporates a free-form defining function with bilinear basis functions:

$$\phi(x, y) = \begin{cases} (1 - |x|)(1 - |y|) & \text{if } |x| \leq 1 \text{ and } |y| \leq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

This function defines a surface drawn in Figure 2a. Moreover, in general case, we need to use this equation with constants h_1 and h_2 to define a rectangular non-zero region of the spline basis function, as follows:

$$\phi'(x, y) = \begin{cases} (1 - \frac{1}{h_1}|x|)(1 - \frac{1}{h_2}|y|) & \text{if } |x| \leq h_1 \text{ and } |y| \leq h_2 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Figure 2. Bilinear spline construction: (a) the surface defined by the bilinear spline basis function (4); (b) the surface produced by two spline basis functions and its zero-level set (boundary); (c) rectangular domain D with 90×25 elements and 91×26 knots.

However, the form (4) can be used here without the loss of generality. This form of the spline basis function is well suited for the further discussion. Space mapping is used to locate spline basis function at some portion of the domain. The bilinear spline basis function with the center of the non-zero region at (x_i, y_i) is:

$$\phi_i(x, y) = \begin{cases} (1 - |x - x_i|)(1 - |y - y_i|) & \text{if } |x - x_i| \leq 1 \\ & \text{and } |y - y_i| \leq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Therefore, the free-form defining function (3) used here can be expressed as follows:

$$f(x, y) = \sum_{i=1}^N \alpha_i \phi_i(x, y) \quad (7)$$

with constants α_i .

The main steps of the proposed modified algorithm are shown in Algorithm 1.

Algorithm 1: The proposed topology optimization algorithm.

- Step 1. Define the number of grid elements in the rectangular domain $nex \times nely$;
 - Step 2. Initialize coefficients of the bilinear spline;
 - Step 3. Define boundary conditions for FEM analysis;
 - Step 4. Initialize parameters of the optimization loop;
 - Step 5. Perform FEM analysis of the domain;
 - Step 6. If the algorithm converges, then quit, else update α_i and go to Step 5.
-

The following equation is used to update the free-form function (3):

$$\begin{aligned} V_n &= V \cdot \left(-\frac{\nabla f}{|\nabla f|} \right), \\ \frac{\partial f}{\partial t} - V_n |\nabla f| &= 0, \end{aligned} \quad (8)$$

where t is the pseudo time representing the evolution of the free-form function, and V_n is the outward normal velocity chosen based on the shape derivative of the optimization problem. The equation for updating α_i is as follows:

$$\alpha_i(t_{j+1}) = \bar{\alpha}_i(t_j) + \Delta t \Phi^{-1} B(\bar{\alpha}_i(t_j), t_j), \quad (9)$$

where

$$\begin{aligned} \bar{\alpha}_i(t_j) &= \frac{\alpha_i(t_j)}{\text{mean}(|f(x_1^1, x_2^1, \dots, x_n^1, t_j)|, |f(x_1^2, x_2^2, \dots, x_n^2, t_j)|, \dots, |f(x_1^r, x_2^r, \dots, x_n^r, t_j)|)}, \\ \Phi &= \begin{pmatrix} \phi_1(x_1^1, x_2^1, \dots, x_n^1) & \cdots & \phi_N(x_1^1, x_2^1, \dots, x_n^1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_1^N, x_2^N, \dots, x_n^N) & \cdots & \phi_N(x_1^N, x_2^N, \dots, x_n^N) \end{pmatrix}, \end{aligned} \quad (10)$$

$(x_1^1, x_2^1, \dots, x_n^1), \dots, (x_1^r, x_2^r, \dots, x_n^r)$ are the points near the solid boundary, $(x_1^1, x_2^1, \dots, x_n^1), \dots, (x_1^N, x_2^N, \dots, x_n^N)$ are the knot points of the bilinear spline (3), $B(\bar{\alpha}_i(t_j), t_j)$ is the velocity vector at the time step t_j :

$$B(\bar{\alpha}_i(t_j), t_j) = \begin{pmatrix} V_n(x_1^1, x_2^1, \dots, x_n^1, t_j) \delta(f(x_1^1, x_2^1, \dots, x_n^1, \bar{\alpha}_i(t_j))) \\ \vdots \\ V_n(x_1^N, x_2^N, \dots, x_n^N, t_j) \delta(f(x_1^N, x_2^N, \dots, x_n^N, \bar{\alpha}_i(t_j))) \end{pmatrix}. \quad (11)$$

For the compliance minimization problem (1), the normal velocity V_n can be derived as follows:

$$V_n = E_{ijkl} \varepsilon_{ij}(u) \varepsilon_{kl}(u) - \lambda \quad (12)$$

where λ is the Lagrange multiplier to deal with the constraint of the volume fraction, which is calculated using the following augmented Lagrangian updating scheme:

$$\lambda^{k+1} = \begin{cases} \mu G^k & k \leq n_R \\ \lambda^k + \gamma^k G & k > n_R \end{cases}, \quad (13)$$

where μ and γ^k are parameters in the k -th iteration of the optimization, and γ^k is updated using the following scheme:

$$\gamma^{k+1} = \min(\gamma^k + \Delta\gamma, \gamma_{\max}) \quad k > n_R. \quad (14)$$

where $\Delta\gamma$ is the increment and $\gamma_{\max}$ is the upper limit of the parameter γ . Since the volume fraction of initial design usually does not meet the prescribed volume fraction, the volume constraint is relaxed in the first n_R iterations as below:

$$G^k = \int_D H(f) d\Omega - \left(V_0 - (V_0 - V_{\max}) \frac{k}{n_R} \right) \quad k \leq n_R. \quad (15)$$

To avoid the unbounded growth of the free-form function, an approximate $\delta(f)$ function is inserted into the velocity vector of (11), defined as follows:

$$\delta(f) = \begin{cases} 0 & f < -\Delta \\ \frac{3}{4}(1 - \frac{f^2}{\Delta^2}) & -\Delta \leq f \leq \Delta \\ 0 & f > \Delta \end{cases} \quad (16)$$

where the parameter Δ is used to control the magnitude of the upper and lower bounds of the free-form function. The ersatz material model is used in the finite element analysis to obtain the naturally extended velocity field from the strain energy density in the whole design domain. The MATLAB implementation of the proposed optimization algorithm is given in the Appendix A.

2.2. FRep Geometric Kernel

The core concept of an FRep-based geometric kernel is a solid body. Its formal definition is a closed subset of an n -dimensional Euclidean space E^n . The defining function (3) describes a solid body in the FRep format. A simple example of defining function is:

$$f(x, y) = \frac{1}{R}(R^2 - (x - x_0)^2 - (y - y_0)^2) \quad (17)$$

which describes a disk in E^2 , with its center in (x_0, y_0) and radius R .

The FRep geometric kernel allows designers to create complex bodies from primitives as shown in (17), using modeling operations. A constructed complex body complies the classification rule (3) and includes all parameters of used primitives, such as center coordinates (x_0, y_0) and radius R of a disk primitive (17). The FRep geometric kernel supports R-functions [7] determining set-theoretic operations. Two examples are presented in Figure 3. Moreover, the FRep supports blending, bending, offsetting, sweeping [8], bounded blending, trimming [9], metamorphosis, etc. [10].

2.3. Visualization

The developed system visualizes objects via level contours (see Figure 4) obtained during the slicing procedure. Therefore, contours serve two goals—visualization and direct slicing for G-code generation (see Section 2.4). Although this is not the fastest way to draw a body, this technique allows us to represent printed objects layer by layer. This approach is useful for interactions with 3D printers, as it helps detect possible manufacturing issues, e.g., overhangs.

The system uses the contouring algorithm described in [18]. A user defines an interval of z coordinates and a step size to set up the slice where the system calculates contours. For each value of z , the algorithm computes contours as a sequence of line segments. The length of these segments depends on the required accuracy. We can speed up the contouring process using the interval arithmetic in the marching squares (MS) algorithm [19,21]. To render generated contours, the freglut [22] library is used.

Union

$$\vee = \max(f_1, f_2)$$

Set difference

$$\setminus = f_1 - f_2 - \sqrt{f_1^2 - f_2^2}$$

Figure 3. Union and set difference operations on functionally defined disks.

Figure 4. Visualization of an FRep model layer by layer.

2.4. Direct Slicing

The concept of direct manufacturing assumes the preparation of a 3D model for fabrication without an intermediate step of surface generation in a special format such as STL. This means that 2D contours for toolpath generation are extracted directly during the slicing process of an FRep object. One of the first mentions of direct manufacturing for AM can be found in [23], and its application for FRep models is described in [19]. Direct manufacturing of the 3D model for 3D printing consists of the following stages:

1. Slicing of the model.
2. Contour extraction (contouring).
3. Generation of the supports and infilling.
4. Generation of the management protocol for the additive manufacturing equipment.

The system supports direct manufacturing for fused filament fabrication (FFF), direct metal deposition (DMD) and digital light processing (DLP) 3D printers. The MS algorithm with the adaptive optimization technique is applied for contour extraction. The open source CuraEngine [24] is used to generate supports and infilling. It allows creation of different infill patterns (see Figure 5) and generation of management protocol (G-code) for FFF and DMD printers. The DLP equipment requires the CWS format. The FRep CAM generates the G-code for print bed control and a set of raster layers to control layer illumination, included in CWS files.

Figure 5. The infill patterns: (a) grid, (b) lines.

2.5. 3D Printing

Figure 6 (bottom) shows the metal 3D part manufactured according to the geometry shown in Figure 7a, left. It is still attached to the cylindrical substrate with marks of previously printed parts here. The part was printed at a 3D printer TruPrint 1000 (Trumpf) of stainless steel powder (Höganäs AB 316L, Germany). The analysis of the powder is

presented in the study [25]. The printing process parameters were chosen according to the manufacturer recommendations. Material parameters, such as Young's modulus and Poisson's ratio, were obtained from results of tensile tests performed according to the ASTM E8/E8M procedure [26], using an Instron 5969 testing machine (Instron). The plate samples with gage section of $2.0 \times 6.0 \text{ mm}^2$ and gage length of 25.0 mm were tested at the loading speed of 10^{-3} s^{-1} .

Figure 6. Realized components of the FRep system.

Figure 7. Cont.

Figure 7. Optimal geometries obtained with modified algorithm (**left**) and original algorithm (**right**) (see Section 3.1): (a) the three-point bending beam optimization case; (b) the optimization case where the beam is supported horizontally in two bottom points, and equal loads applied vertically in upper corners; (c) the cantilever beam optimization case; (d) the optimization case where the beam is fixed along the bottom side, and the horizontal load linearly distributed along the left side of the beam; (e) the optimization case where the beam is fixed horizontally in two bottom points, and equal loads applied horizontally in upper corners.

2.6. Experimental Validation

The 3D printed part was tested in the three-point bending mode following the procedure specified in ASTM D790 [27]. An Instron 5969 testing machine equipped with 6.0 mm diameter round supports and a 12.0 mm diameter loading bar for the three-point bending test was used. The tests were conducted with a loading bar displacement rate of 0.2 mm/min, which moved up to a load of 50 kN, according to Figure 8. To measure the strain, we used the Digital Image Correlation System (Correlated Solutions, Irmo, SC, USA) recording the displacement of points during the test. The images were taken at every 100 N load increment. To increase the stability of the part during the test, several flutes were manufactured as shown in Figure 9 (red dotted area).

Figure 8. The experimental test design.

Figure 9. Cont.

Figure 9. (a) The manufactured part attached to the cylindrical substrate. (b) The image from Digital Image Correlation System. (c) The location of tracked points on the part. (d) The test photos of the part under loads 0 and 15 kN. (e) The test and calculation results of vertical displacement for points P1 and P2. (f) The test and calculation results of vertical displacement for point P3.

3. Results and Discussion

3.1. Modifications of the Topology Optimization Algorithm

The free-form function (7) uses bilinear basis functions instead of MultiQuadric (MQ) basis proposed in the original algorithm [20]. The use of bilinear basis functions has several advantages. First, the α_i from (7) has a simple interpretation. The only spline basis function placed at some point (x_i, y_i) with positive coefficient α_i means that its non-zero region $[x_i - 1, x_i + 1], [y_i - 1, y_i + 1]$ is filled with material. Otherwise, if α_i is negative, no material is present in $[x_i - 1, x_i + 1], [y_i - 1, y_i + 1]$. Two spline basis functions $\phi_i(x, y)$ and $\phi_j(x, y)$ with shared non-zero regions and corresponding α_i and α_j of opposite signs define a portion of the body in the shared region with the boundary defined by the equation:

$$\alpha_i(1 - |x - x_i|)(1 - |y - y_i|) = -\alpha_j(1 - |x - x_j|)(1 - |y - y_j|). \quad (18)$$

The Equation (18) defines a conic section in 2D. Figure 2b shows the case of two bilinear spline basis functions with the shared region and corresponding α_i and α_j of opposite signs. The black region on the right is the domain where the considered function is positive.

Second, the algorithm proposed in [20] works with a rectangular domain D and the bilinear spline (7) can accurately describe it. Consider the domain as shown in Figure 2c covered by 91×26 spline basis functions. Three non-zero regions of the bilinear spline basis functions are shown in green, seven shared non-zero regions of the bilinear spline basis functions are shown in red. Grid knots are the centers of their non-

zero regions. Every non-zero region of a spline basis function covers four white regions. However, it is not the case for the basis functions placed at the boundary of D . One can define the whole rectangular domain D by a finite number of spline functions in the form proposed in Equation (7).

Two conditions hold for all these functions. The α_i values corresponding to all spline basis functions placed at the boundary, equal zero, and others are positive. The optimization algorithm requires the free-form function (7) to have properties of a signed distance function near the boundary. Therefore, for example, for the basis functions placed at the boundary, α_i can be initialized with zeros, and in other cases—with unities.

One more reason to use bilinear splines in Equation (3) is the FEM step. At this step, we approximate the Young's modulus of the rectangular element proportionally to the element's density. For the elements with no material, the Young's modulus is assumed to be zero. Young's modulus of elements filled with material is assumed to be equal to that of proposed material. The algorithm estimates the density of elements partially located at the boundary. It uses exact values of the bilinear spline (7).

Another modification [20] relates to Steps 2 and 6 of the initial optimization algorithm. In the examples from [20], the defining function initially has positive values at the boundary of the domain D . From the FRep perspective, it means that the body has a surface somewhere outside D . However, it is not correct. Mathematically, α_i at the boundary of the domain D has to be zero or negative.

Suppose at some iteration i of the algorithm we obtained the defining function $f_i(x, y)$. The idea above can be written as:

$$f_i(x_j, y_j) = \min(f_{i-1}(x_j, y_j), 0) \quad (19)$$

where (x_j, y_j) is a boundary point of D .

The rule introduced above has an exception. Consider the boundary conditions for the case shown in Figure 10a, left. The shown body has three special points. There are two points where the body touches rollers and the point where force F is applied. Let us suppose that another body produces this force. Therefore, near these three points, we have a region where we cannot easily separate optimized body and external bodies. These points always contain a portion of material. Thus, Equation (19) does not hold for them.

One implementation issue should be noted (see the MATLAB code in the Appendix A). Consider the initialization of the bilinear spline defining the body shown in Figure 10a, right. It is a meaningful option of the body design, but it has holes at the boundary of D . Small negative values are used in Equation (19) instead of zero to allow the algorithm to go over similar shapes.

This modification of Equation (19) is also useful in one more situation. During the optimization process, at some iteration i , we can obtain the defining function (7) with all positive α_j . It is true for all points except ones mentioned in (19). Replacement of zero with some small negative value prevents the algorithm error at the normalization step (10). The above-mentioned modifications make the algorithm more robust.

We compared the proposed modified topology optimization algorithm with the original one [20]. The laptop with the following characteristics was used for tests: Windows 10 Enterprise 64-bit operating system, x64-based processor Intel(R) Core(TM) i5-8250U CPU @ 1.60 GHz 1.80 GHz, and 8.00 GB (7.86 GB usable) of installed memory (RAM). Both algorithms, the original one and the modified one, were run 10 times with following initial datasets. The domain grid is as shown in Figure 2c. The initial geometry is as shown in Figure 10a, right. Boundary conditions are taken from Figure 10a, left (see details in Appendix A). The desired volume, V_0 , was chosen as 45% of the volume of the entire D . The Young's modulus of stainless steel, E_0 , is assumed to be 205 GPa, and the Poisson's ratio, ν , is assumed to be 0.27. Stainless steel is a material obtained of the powder used for 3D printing. The properties of material were studied in [25]. In the original algorithm all parameters of the updating scheme for $f_i(x, y)$ were preserved, except the time step. The time step was changed to 0.2. These parameters were also used for the modified algorithm.

Figure 10. Optimization cases and initial assumptions: (a) the three-point bending beam case; (b) the beam is supported horizontally in two bottom points, and equal loads applied vertically in upper corners; (c) the cantilever beam case; (d) the beam is fixed along the bottom side, and the horizontal load linearly distributed along the left side of the beam; (e) the beam is fixed horizontally in two bottom points, and equal loads applied horizontally in upper corners

Both algorithms exploit an isotropic material model at the FEM step. However, the research [25] showed that 3D printed parts have anisotropy. Nevertheless, the validation results (see Section 3.3) showed that the isotropic model of material is suitable for the considered cases.

Results of the computational test are shown in Table 1. We can see that the modified algorithm is approximately twice as fast as the original one. It can be easily explained using different spline basis functions. When the bilinear spline is used instead of the MQ-spline,

the time-consuming operation of matrix inversion is not needed. Two obtained optimized bodies are shown in Figure 7a.

Table 1. Results of algorithms comparison.

Test Number	Modified Algorithm, s	Wei's Algorithm, s
1	36.8	80.8
2	35.2	77.3
3	36.1	81.7
4	35.9	75.8
5	35.2	76.6
6	35.9	77.3
7	35.4	76.2
8	35.5	76.4
9	35.9	76.1
10	36.3	76.8
Average time, s	35.8	77.5
Number of iterations	213	179
Objective function	1.71×10^5	1.71×10^5
Volume	0.45	0.45

Four more optimization cases shown in Figure 10 were considered to check the observed tendency. Figure 7 shows the results of optimization algorithms testing. Efficiency of the algorithms for these cases is compared in Table 2. The results show that the modified algorithm is faster.

The difference in efficiency is much higher for finer grids of basis functions in the domain. The original algorithm became 20 times slower than the modified algorithm at the case shown in Figure 7c, which can be explained by the matrix inversion used in the Wei's optimization.

At the same time, both algorithms produce similar values of the objective function, differing in the third significant digit, depending on the optimization case.

Small modifications of the algorithm listed in Appendix A were used to produce models shown in Figure 7. These modifications are listed in Appendix B.

Table 2. Efficiency of algorithms for four cases.

Case	Time		Objective Function	
	Modified Algorithm, s	Wei's Algorithm, s	Modified Algorithm	Wei's Algorithm
Figure 10b	21.9	51.9	1.06×10^6	1.06×10^6
Figure 10c	22.2	448.6	6.00×10^5	6.02×10^5
Figure 10d	165.2	14902.2	1.47×10^{10}	1.59×10^{10}
Figure 10e	24.4	437.1	2.48×10^7	2.48×10^7

As in the case of direct slicing, more accurate optimization takes more time and requires more RAM installed. Both tasks, slicing and optimization, lead to the solution of the trade-off between precision and time. Although calculations on the mentioned laptop provide valid data.

3.2. The Developed FRep CAD/CAM System

The proposed FRep CAD/CAM system [28] is written in C++ using the Qt Creator environment. 3D solids in the system are defined by implementations of C++ template function with two groups of parameters. The first group consists of current coordinates X, Y, Z. The second is an array used for parametrization of the FRep model. The template engine uses not only built-in data types (float, double), but also allows the use of interval [29] and affine arithmetic [30] to optimize computations [18]. Moreover, the developed system is equipped with the HyperFun interpreter [31] to provide modeling capabilities. Some primitives are implemented as built-in functions, for example, a free-form object used for topology optimization. The performance of the system when working with built-in functions is much higher.

The developed system uses the contouring algorithm from [18] both for visualization and direct slicing. This algorithm uses MS approach on regular and adaptive grids. Users can choose interval or affine arithmetic for grid adaptation, and use finer grids at the boundary. The effect of adaptive grids is significant when a task requires high accuracy. According to [18] the system performs contouring up to six times faster on adaptive grids than on regular ones. This can be observed on grids with cells 500 times smaller than model sizes.

The system translates generated contours into G-codes, or exports them into STL files. The support of STL format is necessary as some 3D printing equipment has proprietary G-code protocols. Therefore, two manufacturing scenarios can be implemented—direct slicing and STL-based slicing. The second option is slower. According to [19] on some models the direct slicing runs up to 300 times faster than STL-based one.

The developed system has other advantages of FRep. Models created in the system are fully parameterized and have no cracks, overhangs and inverted faces. Moreover, the size of files with models is significantly smaller. For example, the model shown in Figure 4 occupies near 30 KB as an FRep defining function. The same model in STL format occupies approximately 100 MB, which is 3000 times more. This difference increases dramatically with increasing complexity of topology or in the case of porous models.

This system has a broad set of applications. FRep is more attractive in applications where models have microstructures. These models require high precision when they have a small feature size. Complex models with such qualities appear in natural gas separation. Therefore, our system can be applied in this field [32]. Moreover, similar models exist in the implantation [33]. The use of our software for the design of topologically optimal implants seems attractive.

3.3. Results of the Experimental Validation

To validate the developed FRep CAD/CAM system, we printed the part shown in Figure 7a left, and conducted its mechanical testing. Dimensions of the printed part (Figure 9a) are $86 \times 25 \times 20$ mm. The part was cut from the substrate and tested in the three-point bending mode (Figure 9b) at the Instron 5969 testing machine.

Results of the experimental validation are presented as load-displacement curves for several control points. The vertical component of displacement is used in these curves. The obtained load-displacement curve for point P3 (Figure 9c) is presented as red dots in Figure 9f. Computed results for the same load are presented as the solid red line. One can clearly see the disagreement between computational and experimental data.

This discrepancy between the analytical and experimental results of the red lines can be attributed to flutes (Figure 9c) manufactured to increase the stability of the part during the test. A sagging was observed in their areas (Figure 9d) during the testing process. It influenced the load-displacement curve with the displacement value of the whole printed part. The vertical displacement of point P3 was calculated relative to point P0 (Figure 9c) for subtracting the influence of sagging from the load-displacement test curve. Figure 9f shows the load-displacement curves relative to point P0 as green patterns. The correlation between the experimental and predicted results holds up to a load of 15 kN.

During the compression test, it was possible to achieve local plastic deformations. Depending on the distance from P0, the contribution of plastic deformation to the vertical displacement differs. Therefore, additional points, P1 and P2 (Figure 9c), were considered. The computational results for those points are presented in Figure 9e. The correlation between the experimental and computational results for the point P2 is preserved up to the load of 10 kN. For the point P1 the correlation with computational results can be observed up to the load of 5 kN.

We can conclude that the contribution of plastic deformation is the highest in the vicinity of the point P0. The deflection from the elasticity is observed for point P1. It influences the vertical displacements of points P2 and P3. We can conclude that mechanical testing results repeat the theoretical results within the elastic field region, where the computational model was applied. At the loads exceeding 15 kN, the contribution of plasticity starts to grow, and it becomes impossible to compare the computational and experimental results obtained during optimization. At the same time, this experimental validation showed that the proposed approach can be freely used for 3D printed parts that are made up of ceramics [34] and other materials without plastic deformation.

Moreover, we can compare the experimental value of the work done by the applied force and the theoretical values of elastic deformation energy. The upper limit of elastic deformations was observed at the point P3 at the load of 15 kN. This point lies in the vicinity of the loaded point of the body. Therefore, we can calculate an approximate value of the work done by the applied force as follows:

$$A = \frac{Fd}{2} \quad (20)$$

where F is the load and d is the deformation at a given point. The load equals 15 kN, d equals 0.053 mm. Thus, the work A equals 0.4 J. The FEM routine of Algorithm 1 provides the same value of elastic energy at these load and dimensions of the printed part.

4. Conclusions

This paper proposes the FRep-based CAD/CAM system for additive manufacturing. The FRep geometric kernel provides several opportunities for design, analysis and topology optimization. The efficient and robust topology optimization algorithm was developed for the system. The algorithm is based on the level-set method in combination with the defining function of FRep geometry. The proposed algorithm runs up to 100 times faster than the original algorithm, depending on the boundary conditions and the grid size. The interface to 3D printers, implemented in the system, allows manufacturing of functional parts with desired mechanical properties directly from the CAD/CAM system. The validation performed shows the fitness of the simulation model and its ability to accurately predict the elastic deformations of the manufactured part at control points. Moreover, the validation has also shown the directions for further development of the system.

In future research, authors intend to add the automatic generation of supports to the direct slicing algorithm. The optimization algorithm will be updated to operate with several materials and anisotropic materials. In the next version of the system, the analysis procedure will include the model of plasticity. These revisions will allow us to study how printing conditions influence objective functions of optimized parts. Different regimes of 3D printing affect the anisotropic properties of the material. We plan to study their impact on topology optimization.

Author Contributions: Methodology, D.P., S.E. and A.P.; software, D.P. and E.M.; validation, Y.K.; formal analysis, D.P. and Y.K.; writing—original draft preparation, D.P. and Y.K.; writing—review and editing, E.M., A.P. A.S. and S.E.; visualization, D.P., Y.K. and A.S.; supervision, A.P.; project administration, I.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in FigShare at <https://doi.org/10.6084/m9.figshare.15142209.v1>.

Acknowledgments: The authors kindly thank employees of the Center for Design, Materials and Manufacturing of Skoltech Stepan Konev for conducting the mechanical test and consulting with the experimental part of the research and Boris Fedulov for consulting with the simulation and result analysis. This research is funded by the Skoltech Institute of Science and Technology (Skoltech). The authors also would like to thank the anonymous reviewers for their thoughtful and detailed comments on our paper.

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

Abbreviations

The following abbreviations are used in this manuscript:

AM	Additive manufacturing
BRep	Boundary representation
CAD	Computer-aided design
CAM	Computer-aided manufacturing
CNC	computer numerical control
DLP	Digital light processing
DMD	Direct metal deposition
FFF	Fused filament fabrication
FRep	Function representation
MS	Marching squares
PDE	Partial differential equations
SIMP	Solid isotropic material with penalization

Appendix A. MATLAB Implementation of the Optimization Algorithm

```

1  nelx=90;nely=25;volfrac=0.45;
2  %% LEVEL-SET FUNCTION INITIALIZATION
3  r = nely*0.1;%RADIUS OF INITIAL HOLES
4  hX = nelx*[repmat([1/6,5/6],1,3),repmat([0,1/3,2/3,1],1,2),1/2];
5  hY = nely*[kron([0,1/2,1],ones(1,2)),kron([1/4,3/4],ones(1,4)),1/2];
6  [X,Y] = meshgrid(0:1:nelx,0:1:nely);
7  dX = bsxfun(@minus,repmat(X,[1,1,numel(hX)]),reshape(hX,1,1,numel(hX)));
8  dY = bsxfun(@minus,repmat(Y,[1,1,numel(hY)]),reshape(hY,1,1,numel(hY)));
9  Phi = max(-3,min(3,min(sqrt(dX.^2+dY.^2)-r,[],3)));
10 %% RADIAL BASIS FUNCTION INITIALIZATION
11 nNode = (nely+1)*(nelx+1);
12 pGpX = ([Phi(:,2:end) zeros(nely+1,1)] - [zeros(nely+1,1) Phi(:,1:end-1)]);
13 pGpY = ([Phi(2:end,:);zeros(1,nelx+1)] - [zeros(1,nelx+1); Phi(1:end-1,:)]);
14 Boundary = setdiff(unique([1:1:nely+1, nNode-nely:1:nNode, ...
    nely+1:nely+1:nNode, 1:nely+1:nNode-nely]),
15 [(nely+1)*(ceil(nelx/2)+1),0.1*nelx*(nely+1)+1,0.9*nelx*(nely+1)+1]);
16 Phi(Boundary)=-1e-9;
17 Alpha = Phi(:);
18 %% FINITE ELEMENT ANALYSIS PREPARATION
19 E0 = 1; Emin = 1e-9; nu = 0.3; %MATERIAL PROPERTIES
20 A11 = [12 3 -6 -3; 3 12 3 0; -6 3 12 -3; -3 0 -3 12];
21 A12 = [-6 -3 0 3; -3 -6 -3 -6; 0 -3 -6 3; 3 -6 3 -6];
22 B11 = [-4 3 -2 9; 3 -4 -9 4; -2 -9 -4 -3; 9 4 -3 -4];
23 B12 = [ 2 -3 4 -9; -3 2 9 -2; 4 9 2 3; -9 -2 3 2];
24 KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
25 eleN1 = repmat((1:nely)',1,nelx)+kron(0:nelx-1,(nely+1)*ones(nely,1));
26 eleNode = repmat(eleN1(:,1,4))+repmat([0,nely+[1,2],1],nelx*nely,1);
27 edofMat = kron(eleNode,[2,2])+repmat([-1,0],nelx*nely,4);
28 iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
29 jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
30 %% BOUNDARY CONDITION DEFINITION

```

```

31 F = sparse(2*((nely+1)*(ceil(nelx/2)+1)),1,-100,2*nNode,1);%NODAL LOADS
32 fixeddofs = [2*0.1*nelx*(nely+1)+2,2*0.9*nelx*(nely+1)+2];
33 freedofs = setdiff(1:2*nNode,fixeddofs);
34 U = zeros(2*nNode,1);
35 %% ITERATION OPTIMIZATION
36 nLoop = 250; nRelax = 30;
37 dt = 0.2; Delta = 10; mu = 20; gamma = 0.05;
38 comp = zeros(nLoop,1); vol = zeros(nLoop,1);
39 for iT = 1:nLoop
40 %% FINITE ELEMENT ANALYSIS
41 [s,t] = meshgrid(-1:0.1:1,-1:0.1:1);
42 tmpPhi = (1-s(:)).*(1-t(:))/4*Phi(eleNode(:,1))'+(1+s(:)).*(1-t(:))/4*...
43 Phi(eleNode(:,2))'+(1+s(:)).*(1+t(:))/4*Phi(eleNode(:,3))'+...
44 (1-s(:)).*(1+t(:))/4*Phi(eleNode(:,4))';
45 eleVol = sum(tmpPhi>0,1)'/numel(s);
46 vol(iT) = sum(eleVol)/(nelx*nely);
47 sK = reshape(KE(:)*(Emin+eleVol'*(E0-Emin)),64*nelx*nely,1);
48 K = sparse(iK,jK,sK); K = (K+K')/2;
49 U(freedofs,1) = K(freedofs,freedofs)\F(freedofs,1);
50 eleComp = sum((U(edofMat)*KE).*(U(edofMat),2).*(Emin+eleVol*(E0-Emin)));
51 comp(iT) = sum(eleComp);
52 %% DISPLAY RESULTS
53 fprintf('No.%i, Obj:%f, Vol:%f\n',[iT,comp(iT),vol(iT)]);
54 figure(1); contourf(Phi,[0,0]);
55 colormap([0,0,0]); set(gcf,'color','w'); axis equal; axis off;
56 figure(2); surf(Phi); caxis([-12,12]);
57 axis equal; axis([0,nelx,0,nely,-12,12]); view(3);
58 figure(3); subplot(2,1,1); plot(comp(1:iT),'-'); title('Compliance');
59 subplot(2,1,2); plot(vol(1:iT),'-'); title('Volume fraction');
60 %% CONVERGENCE CHECK
61 if iT>nRelax && abs(vol(iT)-volfrac)/volfrac<1e-3 && ...
62 all(abs(comp(iT)-comp(iT-9:iT-1))/comp(iT)<1e-3)
63 break;
64 end
65 %% LAGRANGE MULTIPLIER
66 if iT<=nRelax
67 lag = mu*(vol(iT)-vol(1)+(vol(1)-volfrac)*iT/nRelax);
68 else
69 lag = lag+gamma*(vol(iT)-volfrac);
70 gamma = min(gamma+0.05,5);
71 end
72 %% LEVEL-SET FUNCTION EVOLUTION
73 pGpX = ([Phi(:,2:end) zeros(nely+1,1)] - [zeros(nely+1,1) Phi(:,1:end-1)]);
74 pGpY = ([Phi(2:end,:);zeros(1,nelx+1)] - [zeros(1,nelx+1); Phi(1:end-1,:)]);
75 gradPhi = sqrt((pGpX).^2+(pGpY).^2);
76 indexDelta = (abs(Phi(:))<Delta);
77 DeltaPhi = zeros(size(Phi));
78 DeltaPhi(indexDelta) = 0.75/Delta*(1-Phi(indexDelta).^2/Delta^2);
79 eleComp = reshape(eleComp,nely,nelx);
80 eleCompLR = [eleComp(:,1),eleComp]+[eleComp,eleComp(:,end)];
81 nodeComp = ([eleCompLR;eleCompLR(end,:)]+[eleCompLR(1,:);eleCompLR])/4;
82 B = (nodeComp(:)/median(nodeComp(:))-lag).*DeltaPhi(:)*Delta/0.75;
83 Alpha = Alpha+dt*B;
84 Alpha = Alpha/mean(gradPhi(unique(eleNode((eleVol<1 & eleVol>0),:))));
85 Phi = reshape(Alpha,nely+1,nelx+1);
86 Phi(Boundary)=min(Phi(Boundary),-1e-9);
87 end

```

Appendix B. Modifications of the Optimization Algorithm

There are changes to achieve model Figure 7b.

Lines 14–15:

```
1 Boundary = setdiff(unique([1:1:nely+1, nNode-nely:1:nNode, ...
    nely+1:nely+1:nNode, 1:nely+1:nNode-nely]),
2 [(nely+1), nNode, 0.4*nely*(nely+1)+1, 0.6*nely*(nely+1)+1]);
```

Lines 31–32:

```
1 F = sparse([2*(nely+1), 2*nNode], [1, 1], [-100, -100], 2*nNode, 1);
2 fixeddofs = [2*0.4*nely*(nely+1)+2, 2*0.6*nely*(nely+1)+2];
```

There are changes to achieve model Figure 7c.

Line 1:

```
1 nelx=120;nely=60;volfrac=0.5;
```

Lines 14–15:

```
1 Boundary = setdiff(unique([1:1:nely+1, nNode-nely:1:nNode, ...
    nely+1:nely+1:nNode, 1:nely+1:nNode-nely]),
2 ((nely+1)*nelx+ceil(nely/2)+1));
```

Lines 31–32:

```
1 F = sparse(2*((nely+1)*nelx+ceil(nely/2)+1), 1, -100, 2*nNode, 1);
2 fixeddofs = 1:1:2*(nely+1);
```

There are changes to achieve model Figure 7d.

Line 1:

```
1 nelx=90;nely=180;volfrac=0.5;
```

Lines 14–15:

```
1 Boundary = setdiff(unique([1:1:nely+1, nNode-nely:1:nNode, ...
    nely+1:nely+1:nNode, 1:nely+1:nNode-nely]),
2 [[1:1:(nely+1)], [1:((nely+1)*2):(nNode-(nely+1)*2+1)]]);
```

Lines 31–32:

```
1 F = sparse([2*[1:1:(nely+1)]-1], ones(1, numel([1:1:(nely+1)])), ...
2 [1:1: numel([1:1:(nely+1)])], 2*nNode, 1);
3 fixeddofs = [1:((nely+1)*2):(nNode-(nely+1)*2+1)], ...
4 [1:((nely+1)*2):(nNode-(nely+1)*2+1)+1];
```

There are changes to achieve model Figure 7e.

Line 1:

```
1 nelx=60;nely=120;volfrac=0.5;
```

Lines 14–15:

```
1 Boundary = setdiff(unique([1:1:nely+1, nNode-nely:1:nNode, ...
    nely+1:nely+1:nNode, 1:nely+1:nNode-nely]),
2 [(nely+1), nNode, 0.4*nely*(nely+1)+1, 0.6*nely*(nely+1)+1]);
```

Lines 31–32:


```

1 F = sparse([2*(nely+1)-1,2*nNode-1],[1,1],[-100,-100],2*nNode,1);
2 fixeddofs = [2*0.4*nely*(nely+1)+1,2*0.4*nely*(nely+1)+2, ...
3 2*0.6*nely*(nely+1)+1,2*0.6*nely*(nely+1)+2];

```

References

- Zhang, B.; Goel, A.; Ghalsasi, O.; Anand, S. CAD-based design and pre-processing tools for additive manufacturing. *J. Manuf. Syst.* **2019**, *52*, 227–241. [CrossRef]
- Gisario, A.; Kazarian, M.; Martina, F.; Mehrpouya, M. Metal additive manufacturing in the commercial aviation industry: A review. *J. Manuf. Syst.* **2019**, *53*, 124–149. [CrossRef]
- Kumar, V.; Dutta, D. An assessment of data formats for layered manufacturing. *Adv. Eng. Softw.* **1997**, *28*, 151–164. [CrossRef]
- Schumacher, C.; Bickel, B.; Rys, J.; Marschner, S.; Daraio, C.; Gross, M. Microstructures to Control Elasticity in 3D Printing. *ACM Trans. Graph.* **2015**, *34*. [CrossRef]
- Vilbrandt, C.; Pasko, G.; Pasko, A.; Fayolle, P.A.; Vilbrandt, T.; Goodwin, J.R.; Goodwin, J.M.; Kunii, T.L. Cultural Heritage Preservation Using Constructive Shape Modeling. *Comput. Graph. Forum* **2004**, *23*, 25–41. [CrossRef]
- Pasko, A.; Adzhiev, V.; Sourin, A.; Savchenko, V. Function representation in geometric modeling: Concepts, implementation and applications. *Vis. Comput.* **1995**, *11*, 429–446. [CrossRef]
- Rvachev, V. Method of R-functions in boundary-value problems. *Sov. Appl. Mech.* **1975**, *11*, 345–354. [CrossRef]
- Sourin, A.I.; Pasko, A.A. Function representation for sweeping by a moving solid. *IEEE Trans. Vis. Comput. Graph.* **1996**, *2*, 11–18. [CrossRef]
- Pasko, G.; Pasko, A. Trimming implicit surfaces. *Vis. Comput.* **2004**, *20*, 437–447. [CrossRef]
- Pasko, G.; Pasko, A.; Kunii, T. Space-time blending. *Comput. Animat. Virtual Worlds* **2004**, *15*, 109–121. [CrossRef]
- Rvachev, V.L.; Sheiko, T.I.; Shapiro, V. Application of the method of R-functions to integration of differential equations with partial derivatives. *Cybern. Syst. Anal.* **1999**, *35*, 1–18. [CrossRef]
- Chen, J.; Shapiro, V.; Suresh, K.; Tsukanov, I. Shape optimization with topological changes and parametric control. *Int. J. Numer. Methods Eng.* **2007**, *71*, 313–346. [CrossRef]
- Gibou, F.; Fedkiw, R.; Osher, S. A review of level-set methods and some recent applications. *J. Comput. Phys.* **2018**, *353*, 82–109. [CrossRef]
- Van Dijk, N.P.; Maute, K.; Langelaar, M.; Van Keulen, F. Level-set methods for structural topology optimization: A review. *Struct. Multidiscip. Optim.* **2013**, *48*, 437–472. [CrossRef]
- Wang, Y.; Luo, Z.; Kang, Z.; Zhang, N. A multi-material level set-based topology and shape optimization method. *Comput. Methods Appl. Mech. Eng.* **2015**, *283*, 1570–1586. [CrossRef]
- Safonov, A.A. 3D topology optimization of continuous fiber-reinforced structures via natural evolution method. *Compos. Struct.* **2019**, *215*, 289–297. [CrossRef]
- Goh, G.; Toh, W.; Yap, Y.; Ng, T.; Yeong, W. Additively manufactured continuous carbon fiber-reinforced thermoplastic for topology optimized unmanned aerial vehicle structures. *Compos. Part B Eng.* **2021**, *216*, 108840. [CrossRef]
- Popov, D.; Maltsev, E.; Fryazinov, O.; Pasko, A.; Akhatov, I. Efficient contouring of functionally represented objects for additive manufacturing. *Comput. Aided Des.* **2020**, *129*, 102917. [CrossRef]
- Song, Y.; Yang, Z.; Liu, Y.; Deng, J. Function representation based slicer for 3D printing. *Comput. Aided Geom. Des.* **2018**, *62*, 276–293. [CrossRef]
- Wei, P.; Li, Z.; Li, X.; Wang, M.Y. An 88-line MATLAB code for the parameterized level set method based topology optimization using radial basis functions. *Struct. Multidiscip. Optim.* **2018**, *58*, 831–849. [CrossRef]
- Lopes, H.; Oliveira, J.B.; de Figueiredo, L.H. Robust adaptive polygonal approximation of implicit curves. *Comput. Graph.* **2002**, *26*, 841–852. [CrossRef]
- Olszta, P.W.; Umbach, A.; Baker, S.; Fay, J.F.; Tsiombikas, J.; Niehorster, D.C. The Free OpenGL Utility Toolkit. 2019. Available online: <http://freeglut.sourceforge.net> (accessed on 10 August 2021).
- Jamieson, R.; Hacker, H. Direct slicing of CAD models for rapid prototyping. *Rapid Prototyp. J.* **1995**, *1*, 4–12. [CrossRef]
- Ultimaker. CuraEngine. 2013. Available online: <https://github.com/Ultimaker/CuraEngine> (accessed on 10 August 2021).
- Kuzminova, Y.; Firsov, D.; Konev, S.; Dudin, A.; Dagesyan, S.; Akhatov, I.; Evlashin, S. Structure control of 316L stainless steel through an additive manufacturing. *Lett. Mater.* **2019**, *9*, 551–555. [CrossRef]
- Standard Test Methods for Tension Testing of Metallic Materials; ASTM International Standard: West Conshohocken, PA, USA, 2021.
- Standard Test Methods for Flexural Properties of Unreinforced and Reinforced Plastics and Electrical Insulating Materials; ASTM International Standard: West Conshohocken, PA, USA, 2017.
- Skoltech. FRepADCAM. 2020. Available online: <https://github.com/Torrero/FRepCAM> (accessed on 10 August 2021). [CrossRef]
- Moore, R. *Interval Analysis*; Prentice-Hall: Englewood Cliff, NJ, USA, 1966.
- Stolfi, J.; Figueiredo, L. An Introduction to Affine Arithmetic. *Trends Appl. Comput. Math.* **2003**, *4*, 297–312. [CrossRef]
- Digital Materialization Group. HyperFun Project Description. 2014. Available online: <http://hyperfun.org/> (accessed on 10 August 2021). [CrossRef]

32. Maltsev, E.; Popov, D.; Chugunov, S.; Pasko, A.; Akhatov, I. An Accelerated Slicing Algorithm for Frep Models. *Appl. Sci.* **2021**, *11*, 6767. [[CrossRef](#)]
33. Safonov, A.; Maltsev, E.; Chugunov, S.; Tikhonov, A.; Konev, S.; Evlashin, S.; Popov, D.; Pasko, A.; Akhatov, I. Design and Fabrication of Complex-Shaped Ceramic Bone Implants via 3D Printing Based on Laser Stereolithography. *Appl. Sci.* **2020**, *10*, 7138.
34. Safonov, A.; Chugunov, S.; Tikhonov, A.; Gusev, M.; Akhatov, I. Numerical simulation of sintering for 3D-printed ceramics via SOVS model. *Ceram. Int.* **2019**, *45*, 19027–19035.