

ANO  
2019

RENATO DA MOTTA BUSTAMANTE | FATIADOR AMF COM SUBDIVISÃO DE  
TRIÂNGULOS CURVOS E AJUSTE DE CURVAS



UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC  
CENTRO DE CIÊNCIAS E TECNOLOGIA – CCT  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA – PPGCA

Este trabalho analisa novos padrões de arquivos utilizados em *Additive Manufacture* (AM), com ênfase no padrão *Additive Manufacturing File* (AMF) que utiliza malha de triângulos curvos na representação de modelos 3D. Foi desenvolvido um fatiador (AMF-AC) que utiliza o ajuste de curvas e a intersecção de curvas para melhorar o aproveitamento dessa malha em relação a fidelidade da peça impressa. Os resultados obtidos pelo fatiador proposto são avaliados com relação ao modelo original do arquivo AMF e quantificado o percentual de erro de fidelidade a partir das áreas obtidas durante a intersecção do fatiador.

Orientador: Dr. Roberto Silvio Ubertino Rosso Junior

JOINVILLE, 2019

DISSERTAÇÃO DE MESTRADO

**FATIADOR AMF COM SUBDIVISÃO  
DE TRIÂNGULOS CURVOS E AJUSTE  
DE CURVAS**

RENATO DA MOTTA BUSTAMANTE

JOINVILLE, 2019



**RENATO DA MOTTA BUSTAMANTE**

**FATIADOR AMF COM SUBDIVISÃO DE TRIÂNGULOS CURVOS  
E AJUSTE DE CURVAS**

Dissertação apresentada ao Programa de Pós-graduação em Computação Aplicada, da Universidade do Estado de Santa Catarina, como requisito parcial para obtenção do grau de Mestre em Computação Aplicada.

Orientador: Prof. Dr. Roberto Silvio Ubertino Rosso Junior

**JOINVILLE  
2019**

**Ficha catalográfica elaborada pelo programa de geração automática da  
Biblioteca Setorial do CCT/UDESC,  
com os dados fornecidos pelo(a) autor(a)**

Bustamante, Renato da Motta  
Fatiador AMF com Subdivisão De Triângulos Curvos E  
Ajuste De Curvas / Renato da Motta Bustamante. -- 2019.  
104 p.

Orientador: Roberto Silvio Ubertino Rosso Junior  
Dissertação (mestrado) -- Universidade do Estado de  
Santa Catarina, Centro de Ciências Tecnológicas, Programa  
de Pós-Graduação em Computação Aplicada, Joinville, 2019.

1. Manufatura aditiva. 2. Fatiador. 3. Ajuste de curvas. I.  
Junior, Roberto Silvio Ubertino Rosso . II. Universidade do  
Estado de Santa Catarina, Centro de Ciências Tecnológicas,  
Programa de Pós-Graduação em Computação Aplicada. III.  
Título.

**Fatiador AMF com Subdivisão de Triângulos e Ajuste de Curvas**

por

**Renato da Motta Bustamante**

Esta dissertação foi julgada adequada para obtenção do título de

**Mestre em Computação Aplicada**

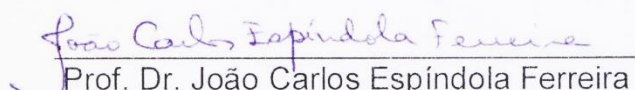
Área de concentração em “Ciência da Computação”,  
e aprovada em sua forma final pelo

CURSO DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA  
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA  
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.

Banca Examinadora:

  
\_\_\_\_\_  
Prof. Dr. Roberto Silvio Ubertino Rosso  
Junior  
CCT/UDESC (Orientador/Presidente)

  
\_\_\_\_\_  
Prof. Dr. Marcelo da Silva Hounsell  
CCT/UDESC

  
\_\_\_\_\_  
Prof. Dr. João Carlos Espíndola Ferreira  
CT/UFSC

Joinville, SC, 26 de fevereiro de 2019.



## RESUMO

Nos últimos anos as impressoras 3D utilizadas na fabricação por Manufatura Aditiva, em inglês *Additive Manufacture* (AM), têm evoluído no acabamento das peças, em novos materiais, em segmentos de aplicação, na redução do tempo de produção e no custo. Contudo, o fatiador que é o *software* responsável por definir as camadas que compõem as peças fabricadas em AM, ainda deixa a desejar nos quesitos de tempo de processamento e fidelidade nos modelos com superfícies curvas complexas. Com base no estudo realizado neste trabalho foram identificados novos padrões de arquivos, que conseguem representar com maior precisão as superfícies curvas dos modelos 3D, como por exemplo o padrão *Additive Manufacturing File* (AMF) especificado pela norma ISO 52915:2016. No entanto, os fatiadores encontrados utilizando o AMF ainda não aproveitam todo o potencial de sua representação para definir as camadas que compõem a peça. Considerando esse cenário, este trabalho desenvolve um fatiador eficiente para o padrão AMF utilizando o ajuste de curvas com o método dos mínimos quadrados, que aproxima curvas paramétricas a partir de sequências de pontos. Isso possibilita uma melhoria na suavidade das superfícies e na fidelidade da peça fabricada. O resultado obtido do ajuste de curvas no fatiador proposto foi avaliado em relação a suas dimensões e aspectos visuais, em comparação ao modelo original do arquivo AMF. Faz parte também deste trabalho a geração código em linguagem de programação de controle numérico (CN) com curvas paramétricas, utilizadas na produção das peças de teste que demonstraram melhorias na suavidade da superfície das peças e reduções no erro de fidelidade durante a análise dos resultados.

**Palavras-chave:** Manufatura aditiva; Fatiador; Ajuste de curvas.



## ABSTRACT

Over the last years, 3D printers utilized on Additive Manufacturing (AM) have improved: the finishing of the pieces, the use of new materials, other segments of application, lead time reduction and lower costs. Nevertheless, the slicer that is the *software* that defines the layers in AM, can be improved, especially regarding the time needed in processing and the fidelity on models with complex curved surfaces. Based on study, new files standards were found, that are able to represent the curve surfaces of the 3D models with much more accuracy. The ISO52915:2016 defines the *Additive Manufacturing File* (AMF) standard, which is an example of representation. However, the slicers that use this standard still do not explore its full potential for representation to define the layers that compound the piece. In this scenario, this work has the objective to implement an efficient slicer to the AMF standard, fitting curves by the Least Squares method, closing in parametric curves from sequences of points, which will enable to improve the accuracy of slicer. The result of this new curve fitting on the proposed slicer will be evaluated regarding the original model of the AMF file. It is also part of this study to generate the numerical control (NC) code with parametric curves, used to produce test subject pieces that will be evaluated regarding its dimensions and visual aspects as part of the result analysis.

**Keywords:** Additive manufacturing; Slicer; Fitting Curve.



## **AGRADECIMENTOS**

Agradecimento especial ao meu orientador, Roberto S. U. Rosso Jr., que com seu incentivo e sua orientação foi possível chegar até aqui. Meu muito obrigado pela paciência, encorajamento, aprendizado para que eu alcançasse o sucesso do trabalho.

Agradeço aos colaboradores e parceiros deste projeto, professor Fernando Lafratta (UDESC) e Carlos Sacchelli (UFSC), pela prontidão em contribuir, pela disponibilidade e apoio nos diversos momentos.

Aos colegas de mestrado, professores e estudantes do Laboratório de Pesquisa em Aplicações Visuais (LARVA), pela contribuição e companhia durante este tempo.

E a todos os professores, família, namorada e amigos que durante toda a minha vida estiveram presentes e contribuíram, para que hoje este projeto se concretizasse.



## LISTA DE ILUSTRAÇÕES

Figura 1 – Componentes fundamentais do método FDM de AM. ....	31
Figura 2 – Métodos de solidificação dos materiais em processos de AM. ....	33
Figura 3 – Modelo 3D representado no padrão STL em diferentes LODs de malha poligonal. ....	34
Figura 4 – Localização dos erros Real e de Corda entre a representação poligonal e o modelo CAD. ....	34
Figura 5 – Representação da altura da cúspide em camadas de borda retangulares. ....	35
Figura 6 – (a) triângulo plano definido por 3 vértices, (b) triângulo curvo definido por 3 normais, (c) triângulo curvo definido por 2 tangentes, (d) subdivisão dos triângulos curvos. ....	38
Figura 7 – Escopo da STEP AP 242 2ª edição. ....	40
Figura 8 – Fluxograma do planejamento de processo. ....	41
Figura 9 – Diferentes abordagens de fatiamento: uniforme, adaptativo e adaptativo local. ....	44
Figura 10 – Etapas do fatiamento em malha de triângulos planos. ....	45
Figura 11 – Divisão de vértices utilizando Quadtree. ....	46
Figura 12 – Agrupamento de vértices pelo algoritmo KNN. ....	46
Figura 13 – Aplicação das regras de ligação dos pontos na proposta de Zeng et al. (2011). ....	47
Figura 14 – (a) triângulo sem divisão, (b) triângulo dividido com grau 1, (c) triângulo dividido com grau 2. ....	48
Figura 15 – Ponto $V_{01}$ na curva cúbica de Hermite. ....	48
Figura 16 – Etapas do processo de interpolação das curvas de Hermite no AMF. ....	49
Figura 17 – Ajuste de curva cúbica de Bézier em uma amostra de sequência de pontos. ....	50
Figura 18 – Pontos de controle ( $CP$ ) utilizados na representação com superfície biquadrada de Bézier em forma triangular (no caso, $CP_{11}$ , $CP_{12}$ e $CP_{13}$ são iguais). ....	56
Figura 19 – Triângulo de Steiner, onde $P_{ij}$ são os pontos de controle e $w_{ij}$ são os pesos associados. ....	57
Figura 20 – Cenários de possíveis singularidade para o fatiador. ....	58

Figura 21 – Triângulo com arestas e vértices de acordo com posição em relação ao eixo z e índice STL.....	59
Figura 22 – Aproximação pelo método <i>Max-Fit Biarc</i> dos pontos de intersecção interrompida em $p_4$ .....	60
Figura 23 – (a) borda de contorno linear, (b) bordas de contorno com aproximação de curvas.....	60
Figura 24 – (a) normais $n$ , tangentes $t$ e pontos $p$ que definem a curva de Bézier, (b) pontos de intersecção entre a curva aproximada e o plano. ....	61
Figura 25 – (a) secções no poliedro, (b) fatiamento sem ajuste de curva, (c) fatiador com ajuste de curva e intersecção de arestas curvas. ....	61
Figura 26 – Captura de tela do <i>software</i> MatterControl com simulação do percurso para impressão 3D .....	67
Figura 27 – Fluxograma do fatiamento realizado pelo MatterControl. ....	68
Figura 28 – Propagação do idRoot na subdivisão do poliedro com 20 triângulos curvos. ....	71
Figura 29 – Normais, tangentes e vértices utilizados no cálculo na subdivisão do triângulo curvo.....	71
Figura 30 – Sequências de operações realizadas para calcular a normal nos três métodos propostos. ....	72
Figura 31 – Subdivisões realizadas utilizados os métodos (a), (b) e (c) para calcular a normal. ....	72
Figura 32 – (a) um ponto de intersecção, (b) dois pontos de intersecção, (c) sem intersecção.....	74
Figura 33 – Casos resultantes da intersecção com triângulos curvos: (a) três arestas com duas intersecções cada, (b) duas arestas com duas intersecções cada, (c) uma aresta com duas intersecções, (d) uma aresta com duas intersecções e duas arestas com uma intersecção, (e) caso padrão.....	75
Figura 34 – (a) AMF original, (b) primeira subdivisão, (c) segunda subdivisão, (d) pontos de intersecção com a camada, (e) ajuste de curvas na sequência de pontos. ....	76
Figura 35 – Comparação de suavidade na curva entre os fatiamentos gerados pelo fatiador AMF-AC sem ajuste curva (a) e com ajuste de curva (b). ....	78
Figura 36 – Comparação entre a suavidade das superfícies das simulações 3D obtidas pelo fatiador. ....	79

Figura 37 - Exemplos de sequencias de pontos ajustadas em curvas de Bézier pelo método de Newton-Raphson no Fatiador AMF-AC. ....	79
Figura 38 – Simulação do G-Code com os comandos G5 e G1. ....	80
Figura 39 – Simulação colorida das curvas cúbicas de Bézier ajustadas pelo fatiador AMF-AC. ....	82
Figura 40 – Modelos 3D, em padrão AMF, utilizados nos testes com o fatiador AMF-AC. ....	85
Figura 41 – Destaque no serrilhado causado pela falha na regulagem da impressora RepRap. ....	86
Figura 42 – Peças fabricadas utilizando a impressora RepRap com o comando G5 no G-Code. ....	88
Figura 43 – Destaque das superfícies com ajuste de curva (laranja) e sem ajuste de curva (branca). ....	89
Figura 44 – Fotografia ampliada das curvas sem ajuste e com ajuste, fabricadas com filamento ABS. ....	89
Figura 45 – procedimentos realizados para quantificar a área dos níveis analisados. ....	90
Figura 46 – Sobreposição das simulações mostrando inflexões na superfície do solido 6SUB. ....	91
Figura 47 – Peças fabricadas durante os testes de impressão com seus problemas identificados. ....	92
Figura 48 – Detalhamento dos métodos e matérias utilizados na impressão por AM. ....	104
Gráfico 1– Gráfico de crescimento de vendas de equipamentos de AM .....	29
Gráfico 2 – (a) gráfico de nº de publicações por ano, (b) quantidade de artigos por palavra chave. ....	55
Quadro 1 – Codificação Explícita dos arquivos STL em ASCII. ....	37
Quadro 2 – Estrutura organizacional das <i>tags</i> utilizadas no padrão AMF. ....	39
Quadro 3 – Comandos de movimentos do <i>firmware</i> Marlin. ....	53
Quadro 4 – Atributos da classe SimpleFace. ....	70
Quadro 5 – Exemplo de arquivo G-Code exportado pelo fatiador AMF-AC. ....	83



## LISTA DE TABELAS

Tabela 1 – Classificação dos métodos de AM por deposição e trajeto de solidificação .....	30
Tabela 2 – Tabela de fatiadores.....	52
Tabela 3 – Módulos e suas respectivas classes principais do <i>software</i> MatterControl. .....	67
Tabela 4 – Valores das porcentagens da redução de erro de fidelidade nos 18 níveis analisados.....	91
Tabela 5 – Características e informações sobre os fatiadores analisados. ....	106
Tabela 6 – Áreas obtidas nas intersecções realizadas entre simulações de peças e 18 níveis diferentes.....	107



## LISTA DE ABREVIATURAS E SIGLAS

<b>AM</b>	<i>Additive Manufacturing</i>
<b>AMF</b>	<i>Additive Manufacturing File</i>
<b>CAD</b>	<i>Computer Aided Design</i>
<b>CN</b>	Controle Numérico
<b>FDM</b>	<i>Modelagem por Fusão e Deposição</i>
<b>GD&amp;T</b>	<i>Geometric dimensioning and tolerancing</i>
<b>PP</b>	<i>Processing Planning</i>
<b>PLA</b>	<i>Polylactic acid</i>
<b>SPI</b>	<i>Surface-Plane Intersection</i>
<b>STEP</b>	<i>Standard for the Exchange of Product Model Data</i>
<b>SLS</b>	<i>Sinterização Seletiva a Laser</i>
<b>STL</b>	<i>Stereolithography File Format</i>
<b>SLA</b>	Estereolitografia
<b>NURBS</b>	Superfícies de B-Splines Racionais Não-Uniformes
<b>ISO</b>	<i>International Organization for Standardization</i>
<b>XML</b>	<i>Extensible Markup Language</i>



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>25</b>
1.1	OBJETIVOS .....	27
1.2	ESCOPO.....	27
1.3	ESTRUTURA .....	28
1.4	METODOLOGIA.....	28
<b>2</b>	<b>FUNDAMENTAÇÃO</b> .....	<b>29</b>
2.1	MANUFATURA ADITIVA.....	29
<b>2.1.1</b>	<b><i>Fused Deposition Modeling (FDM)</i></b> .....	<b>31</b>
<b>2.1.2</b>	<b>Outros métodos de AM</b> .....	<b>32</b>
2.2	MÉTRICAS DE AVALIAÇÃO DOS MÉTODOS DE AM .....	33
<b>2.2.1</b>	<b>Custo</b> .....	<b>33</b>
<b>2.2.2</b>	<b>Fidelidade</b> .....	<b>34</b>
2.2.2.1	Erros de simplificação do modelo CAD .....	34
2.2.2.2	Erros do fatiador de sólidos .....	35
2.2.2.3	Erros do método de impressão.....	35
<b>2.2.3</b>	<b>Funcionalidade</b> .....	<b>36</b>
2.3	PADRÕES DE REPRESENTAÇÃO .....	36
<b>2.3.1</b>	<b>STL</b> .....	<b>37</b>
<b>2.3.2</b>	<b>AMF</b> .....	<b>38</b>
2.3.2.1	Estrutura do Arquivo.....	38
<b>2.3.3</b>	<b>STEP</b> .....	<b>40</b>
2.4	PLANEJAMENTO DE PROCESSO (PP) .....	41
<b>2.4.1</b>	<b>Requisitos do modelo 3D</b> .....	<b>42</b>
<b>2.4.2</b>	<b>Orientação de impressão</b> .....	<b>42</b>
<b>2.4.3</b>	<b>Estruturas de suporte</b> .....	<b>43</b>
<b>2.4.4</b>	<b>Fatiamento</b> .....	<b>44</b>
2.4.4.1	Tamanho de camada .....	44
2.4.4.2	Algoritmos de fatiamento .....	45
2.4.4.2.1	Fatiamento em malhas triangulares .....	45
2.4.4.2.2	Fatiamento em arquivos CAD.....	47
2.4.4.2.3	Fatiamento em arquivos AMF.....	48

2.4.4.3	Ajuste dos pontos de intersecção .....	50
<b>2.4.5</b>	<b>Simulador .....</b>	<b>52</b>
<b>2.4.6</b>	<b>Gerador de código em linguagem de programação de CN .....</b>	<b>53</b>
2.5	CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	54
<b>3</b>	<b>TRABALHOS RELACIONADOS .....</b>	<b>55</b>
3.1	PESQUISA BIBLIOGRÁFICA .....	55
3.2	NOVAS REPRESENTAÇÕES .....	56
<b>3.2.1</b>	<b>Malha de Bézier.....</b>	<b>56</b>
<b>3.2.2</b>	<b>Malha de Steiner .....</b>	<b>57</b>
3.3	FATIADORES COMPUTACIONALMENTE EFICIENTES.....	58
<b>3.3.1</b>	<b>Fatiador baseado no deslocamento dos vértices .....</b>	<b>58</b>
<b>3.3.2</b>	<b>Fatiador baseado na ordem de leitura do arquivo .....</b>	<b>59</b>
3.4	APROXIMAÇÕES DE CURVAS EM FATIADORES .....	60
<b>3.4.1</b>	<b>Aproximação com <i>Biacr: Max-Fit Biarc Method</i>.....</b>	<b>60</b>
<b>3.4.2</b>	<b>Aproximação com <i>B-Splines</i> .....</b>	<b>61</b>
3.5	DISCUSSÃO DOS TRABALHOS .....	62
<b>3.5.1</b>	<b>Sobre as novas representações .....</b>	<b>62</b>
<b>3.5.2</b>	<b>Sobre a suavização dos contornos.....</b>	<b>62</b>
<b>3.5.3</b>	<b>Sobre a eficiência do fatiador .....</b>	<b>63</b>
3.6	CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	63
<b>4</b>	<b>DESENVOLVIMENTO DO FATIADOR AMF-AC .....</b>	<b>65</b>
4.1	ARQUITETURA DO SOFTWARE MATTERCONTROL.....	66
4.2	ESTRUTURA DO FATIADOR AMF-AC .....	69
4.3	PRÉ-PROCESSAMENTO DA MALHA TRIANGULAR CURVA .....	69
<b>4.3.1</b>	<b>Leitura do arquivo AMF.....</b>	<b>70</b>
<b>4.3.2</b>	<b>Subdivisão da malha triangular curva .....</b>	<b>70</b>
<b>4.3.3</b>	<b>Deslocamentos de vértices.....</b>	<b>73</b>
4.4	INTERSECÇÃO COM AS CAMADAS .....	73
4.5	UNIÃO DOS PONTOS DE INTERSECÇÃO .....	74
4.6	AJUSTE DE CURVA .....	76
4.7	GERAÇÃO CÓDIGO EM LINGUAGEM DE G-CODE.....	80
4.8	CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	83

<b>5</b>	<b>TESTES E ANÁLISE DE RESULTADOS .....</b>	<b>85</b>
5.1	MODELOS 3D EM AMF .....	85
5.2	IMPRESSORAS FDM .....	86
5.3	DEFINIÇÃO DE PARÂMETROS .....	87
5.4	ANÁLISE DAS PEÇAS RESULTANTES .....	87
5.5	PROBLEMAS ENCONTRADOS .....	92
5.6	CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	93
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>95</b>
	<b>REFERÊNCIAS .....</b>	<b>99</b>
	<b>ANEXO A – RELAÇÃO DE MATÉRIAS E MÉTODOS DE IMPRESSÃO.....</b>	<b>104</b>
	<b>APÊNDICE A - PESQUISA DE FATIADORES EXISTENTES.....</b>	<b>106</b>
	<b>APÊNDICE B – SIMULAÇÕES DE INTERSECÇÕES.....</b>	<b>107</b>



## 1 INTRODUÇÃO

*Additive Manufacture (AM)*, *solid freeform fabrication* ou também conhecida como *3D printing*, é definida pela *ASTM International Committee* como o processo de junção de materiais para criação de peças a partir de modelos tridimensionais, pela adição de camadas repetidas vezes até que se tenha uma peça completa (WEBER et al., 2013).

Originada a partir de estudos de topografia e foto-escultura, a AM foi aplicada comercialmente somente a partir de 1987, pela empresa 3D Systems com o método de Stereolitografia (SLA). Esse método inicialmente produzia peças para moldes e prototipagem com o intuito de substituir os antigos métodos manuais de construção (BOURELL et al., 2009).

Desde então, a AM tem evoluído no acabamento das peças, em novos materiais, na redução do tempo de produção e do custo. Com destaque nas impressoras 3D *desktop*, que ampliaram os segmentos de aplicação da AM, devido a sua simplicidade funcional e o baixo custo. Atualmente, essa tecnologia já pode ser encontrada nos mais variados segmentos: mobiliário, iluminação, vestuário, alimentício, médico, bioimpressão, artístico, aeronáutico, restauração e construção civil (MONTEIRO, 2015).

Com o aumento do número de usuários da AM, é importante que o planejamento de processo (PP), onde ocorrem as configurações iniciais para o processo de fabricação, se torne automatizado, rápido e confiável. Dessa forma, os usuários das diversas áreas de conhecimento poderão utilizar as impressoras 3D sem a necessidade de um conhecimento profundo da tecnologia envolvida.

O PP inicia-se normalmente com a escolha do padrão de representação do modelo 3D. Esses são utilizados pelos *software* de impressoras 3D e repositórios *online* de modelos 3D como: *Thingiverse*<sup>1</sup>, *Instructables*<sup>2</sup> e *3D HUBS*<sup>3</sup>, para armazenar de uma forma simples a informação necessária para fabricação da peça. Atualmente, o padrão mais utilizado em AM é o *Stereolithography File Format (STL)* (ISO/ASTM 52915:2016).

Com mais de 30 anos e uma estrutura simples, o STL já está consolidado na maioria dos *software* de impressora encontrados. No entanto, ele não consegue representar algumas características do modelo, como por exemplo superfícies curvas

---

<sup>1</sup> <https://www.thingiverse.com>, acessado em 01/06/2017.

<sup>2</sup> <https://www.instructables.com>, acessado em 01/06/2017.

<sup>3</sup> <https://www.3dhubs.com>, acessado em 01/06/2017.

e diferentes tipos de materiais. O STL também se torna ineficiente na representação de modelos 3D complexos, pois seu padrão estrutural faz com que os seus arquivos se tornem muito grandes e sua operação se torne lenta.

Em resposta às limitações do STL, a ISO 52915:2016 apresenta o padrão de arquivo *Additive Manufacturing File* (AMF); nele o modelo 3D é representado por triângulos curvos de Hermite, que mantém as curvaturas suaves e reduz o erro causado pela simplificação das superfícies em relação ao STL. O padrão AMF também é dotado de estruturas que possibilitam armazenar diferentes tipos de materiais utilizados no modelo, e também informações de rotações, escalonamento e multiplicidade de objetos para a fabricação da peça em AM (ISO/ASTM 52915:2016).

Outra etapa importante do PP é o fatiamento, também conhecidos como algoritmos de *Surface-Plane Intersection* (SPI). Esses são fundamentais para a AM, já que sem o fatiamento do modelo, não há como identificar os locais sólidos de cada camada que compõe a peça final. Nesse sentido, muitos pesquisadores desenvolveram técnicas de aperfeiçoamento para o fatiamento de arquivos STL, em busca de melhorias no tempo de processamento, nas camadas adaptativas, no paralelismo e na suavização de superfícies através de ajuste de curvas (ZHAO et al., 2009; ZHANG e JOSHI, 2015; KIM et al., 2010; KOC et al., 2000; TATA et al., 1998).

Apesar do avanço que o padrão AMF proporciona na representação do modelo 3D, esse ainda tem seu processo de fatiamento em AM muito semelhante ao utilizado pelo padrão STL, pois nele é feito o uso da simplificação dos triângulos curvos em triângulos planos. Essa simplificação é que causa a perda de exatidão das superfícies curvas e as desconsidera no restante do processo de fatiamento, reduzindo a fidelidade da peça final.

Embora os padrões de arquivos representados por triângulos curvos tenham trazido melhorias na representação dos modelos, os processos de fatiamento utilizados por eles ainda implicam em aproximações que causam erro de Tolerância Geométrica e Dimensional, do inglês *Geometric Dimensioning and Tolerancing* (GD&T), o que reduz a fidelidade entre a peça física e o modelo digital em 3D (VAIDYA e ANAND, 2016).

Além das aproximações ocasionadas pela escolha do padrão de representação outras aproximações também podem ser necessárias durante a geração código em linguagem de programação de controle número (CN), linguagem utilizada no controle da impressora 3D. Essas aproximações ocorrem ao gerar uma sequência de comandos para representar uma curva paramétrica utilizando apenas segmentos de retas e arcos de circunferências.

Contudo, não foi encontrado na literatura, nenhum fatiador de AMF capaz de manter a suavidade e a fidelidade das superfícies representadas pelas malhas de triângulos curvos, sem subdivisões exaustivas da malha para reduzir o erro de GD&T. Tampouco fatiadores de AMF que utilizassem curvas paramétricas na geração do código em linguagem de CN.

## 1.1 OBJETIVOS

O presente trabalho visa desenvolver um algoritmo de fatiamento de modelos 3D para o padrão AMF, que mantenha a suavidade das superfícies e reduza o erro de fidelidade, de forma computacionalmente mais eficiente em relação aos fatiadores existentes para AMF.

Como objetivos específicos, destacam-se:

- Desenvolver o fatiador com intersecção de arestas curvas;
- Ajustar as curvas dos trajetos das camadas;
- Gerar comandos de curvas paramétricas na Linguagem de CN;
- Reduzir o erro de fidelidade em relação ao fatiador padrão do AMF;
- Apresentar resultados com peças fabricadas em AM.

## 1.2 ESCOPO

Este trabalho está restrito ao aspecto de fidelidade relacionado a curvaturas no plano horizontal da superfície externa do modelo, não são realizados aprimoramentos em relação a outras curvaturas.

Para o funcionamento do fatiador são considerados apenas modelos 3D sólidos (*2-manifolds*) representados no padrão AMF que devem ser fatiados individualmente no centro da base da impressora 3D, assim como a impressão das peças também é individual.

Os modelos devem estar descritos no arquivo, em sua posição e orientação correta, não faz parte do escopo aplicar transformações no modelo AMF. Também não fazem parte do escopo a geração de suportes, e os comandos de retração do filamento.

Para efetuar a impressão das peças são utilizadas as impressoras 3D do tipo FDM, sendo elas: a Replicator2X da marca Makerbot<sup>1</sup> disponibilizada pelo autor, e a impressora RepRap disponibilizada pela Universidade Federal de Santa Catarina (UFSC) – Campus Joinville.

---

<sup>1</sup> Empresa americana fabricante de impressoras 3D, adquirida pela Stratasys em 2013.

São utilizados filamentos do tipo Acrilonitrila Butadieno Estireno (ABS) e Polyactic Acid (PLA) para a impressão das peças de teste.

### 1.3 ESTRUTURA

Este trabalho está organizado em 6 capítulos. No capítulo 2 são apresentados os fundamentos básicos no processo de AM, padrões de arquivos e fatiadores existentes. No capítulo 3, são relacionadas as técnicas aplicadas nos fatiadores atuais e as abordagens de padrões de arquivos. Em sequência, no capítulo 4 é apresentado o desenvolvimento do fatiador AMF-AC. Resultados e análise dos testes são apresentados no capítulo 5, e por fim, a conclusão no capítulo 6.

### 1.4 METODOLOGIA

Segundo os paradigmas de Eden (2007), esta pesquisa pode ser enquadrada como tecnocrática, pois procura conhecimento a *posteriori* por meio de testes de confiabilidade. Já do ponto de vista de objetivo, ela pode ser classificada como descritiva, indo além da identificação da existência de relações entre as variáveis, identificando uma nova aplicação de intersecções no processo de fatiamento para AM.

O raciocínio seguido pela pesquisa é dado de forma dedutiva, partindo de premissas tidas como verdadeiras, como as operações de intersecção sobre sólidos, para se obter uma conclusão também verdadeira, ou seja, o conjunto das intersecções com as camadas de impressão (MARCONI e LAKATOS, 2005).

Quanto a natureza das variáveis que envolvem a pesquisa, pode-se afirmar que a análise dos dados é quantitativa e qualitativa, já que a pesquisa envolve testes de confiabilidade dos resultados e análise visual das peças, segundo Gil (2002). Algumas variáveis da pesquisa são independentes, podendo ser executadas sem correlação com outras variáveis, tal como a intersecção da superfície com o plano de corte para cada camada. Outras variáveis têm caráter dependente, como a definição de caminhos percorridos para o processo de fabricação por AM uma vez que esta definição depende das intersecções geradas anteriormente na etapa de fatiamento.

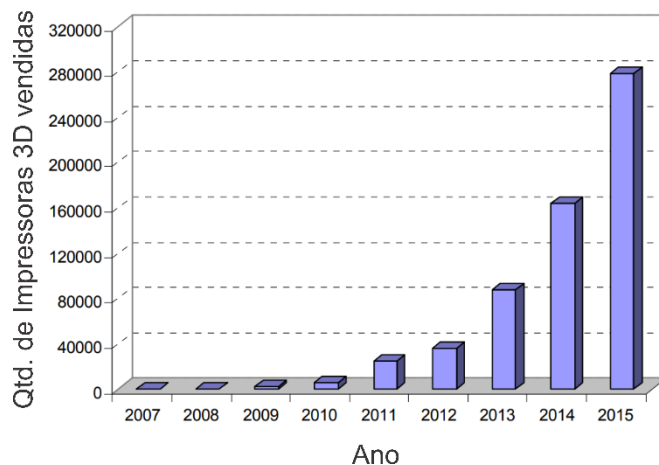
## 2 FUNDAMENTAÇÃO

Neste capítulo são apresentados conceitos básicos dos métodos de fabricação de peças por manufatura aditiva (AM), com suas características, vantagens e tecnologias atuais. Também são detalhadas as etapas que compõem o planejamento do processo (PP) de AM em modelagem por deposição fundida (FDM), incluindo métricas para avaliação de desempenho e arquivos de importação.

### 2.1 MANUFATURA ADITIVA

Ao longo das últimas quatro décadas, o processo de fabricação por AM expandiu-se entre diversas tecnologias e métodos, que variam da adição por camada de papel até complexos sistemas de deposição de metal (BOURELL et al., 2009). Tendo um crescimento ainda maior nos últimos anos devido à redução de preços, que ampliaram o mercado consumidor e consequentemente o número de vendas, como mostra o Gráfico 1, com o número de vendas entre os anos de 2007 a 2015 (CAFFREY et al., 2016).

Gráfico 1– Gráfico de crescimento de vendas de equipamentos de AM



Fonte: adaptado de (CAFFREY et al., 2016)

Este crescimento faz com que a AM seja uma área de grande potencial para pesquisas, por ser uma tecnologia recente, com ainda muitas possibilidades de inovações para serem aplicadas nos processos de impressão, as quais podem trazer grandes impactos econômicos, devido ao grande interesse comercial na área.

O crescimento da área também trouxe um aumento no número de métodos de AM, em geral com o mesmo princípio básico da construção pela adição de camadas bidimensionais. Desta forma, para identificar as semelhanças entre os diversos métodos, é importante classificá-los de acordo com duas características fundamentais: a deposição do material e o trajeto de solidificação do material (LIVESU et al., 2017). A deposição de material se ramifica em local ou por camada, e o trajeto de solidificação em vetorial e discreto, como mostra a Tabela 1, com a classificação dos métodos de AM.

Tabela 1 – Classificação dos métodos de AM por deposição e trajeto de solidificação

		Trajeto de Solidificação do Material	
		Vetorial	Discreto
Deposição de Material	Local	FDM	PolyJet
		LENS	
		EBAM	
	Camada	SLA	DLP
		SLS	BJ
		EBM	MJG
		SLM	

Fonte: Produção do autor.

A deposição de material local é mais econômica em relação a deposição por camada, pois no modo local o material é depositado somente sobre o volume que representa o modelo 3D, evitando que áreas externas do modelo recebam o material durante a produção da peça.

No entanto, o modo de deposição por camada que efetua o preenchimento da camada inteira com material, possui vantagens estruturais que podem evitar a necessidade de suporte, que será detalhado na seção 2.4.3. Outra característica deste modo de impressão, é que a quantidade de material e o tempo gasto no processo estão relacionados com o tamanho da base de impressão, uma vez que a deposição de material não depende do modelo 3D, e sim da área de deposição.

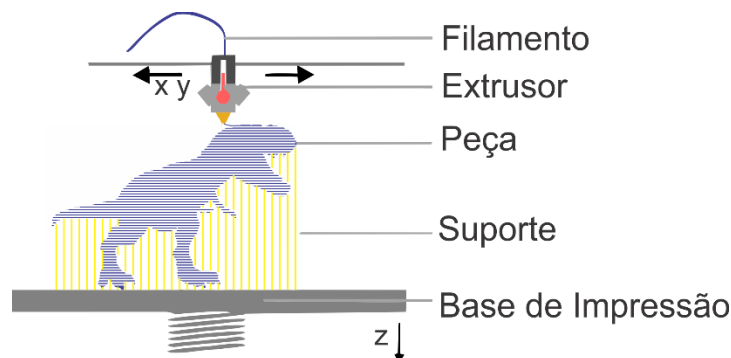
O trajeto de solidificação é a sequência em que as partículas do material são solidificadas, podendo ser de modo vetorial ou *raster* (discreto). No vetorial, são solidificadas as linhas ou curvas definidas nas camadas da peça, enquanto no modo *raster* a solidificação ocorre percorrendo os pontos da camada de forma sequencial e solidificando-os onde houver partes do modelo.

### 2.1.1 Fused Deposition Modeling (FDM)

O método de FDM é uma tecnologia de AM que utiliza material fundido (plástico, metal e outros substratos) com deposição de material local, reconhecido pela sua eficácia em relação ao tempo e baixo custo de impressão (MONTEIRO, 2015). Estas características são importantes para trabalhos acadêmicos que não possuem muitos recursos financeiros e são realizados em prazos curtos. Impressoras desta tecnologia possuem valores de mercado inferiores a 10% em relação as impressoras encontradas do tipo de Sinterização Seletiva a Laser (SLS).

O princípio do funcionamento do método de FDM é semelhante ao utilizado em pistolas de cola quente, porém em FDM o filamento (material) é introduzido no extrusor que se movimenta paralelamente em relação a base de impressão, como ilustra a Figura 1. Os componentes básicos de uma impressora de FDM são: um sistema de motores de precisão para movimentação do extrusor nos eixos x, y, assim como para a base de impressão no eixo z, é necessário também um motor para empurrar o filamento para dentro do extrusor. É esta combinação de movimentos pré-programados que fazem a impressão da peça através de camadas de filamento fundido depositado no trajeto sobre a base.

Figura 1 – Componentes fundamentais do método FDM de AM.



Fonte: Produção do autor.

Os materiais utilizados no método de FDM são normalmente termoplásticos biodegradáveis, como o Ácido Polilático (PLA) derivado de amido de milho, ou não biodegradáveis como o Acrilonitrila Butadieno Estireno (ABS) derivado do petróleo. Existem outros materiais que também podem ser encontrados nas impressoras do tipo FDM como: filamentos de Politereftalato de Etileno Glicol (PETG), que tem como vantagem ser um material reciclável e filamentos de Poliuretano Termoplástico (TPU), que produzem peças flexíveis e macias e filamentos solúveis ideais para realização de suportes durante a impressão.

### 2.1.2 Outros métodos de AM

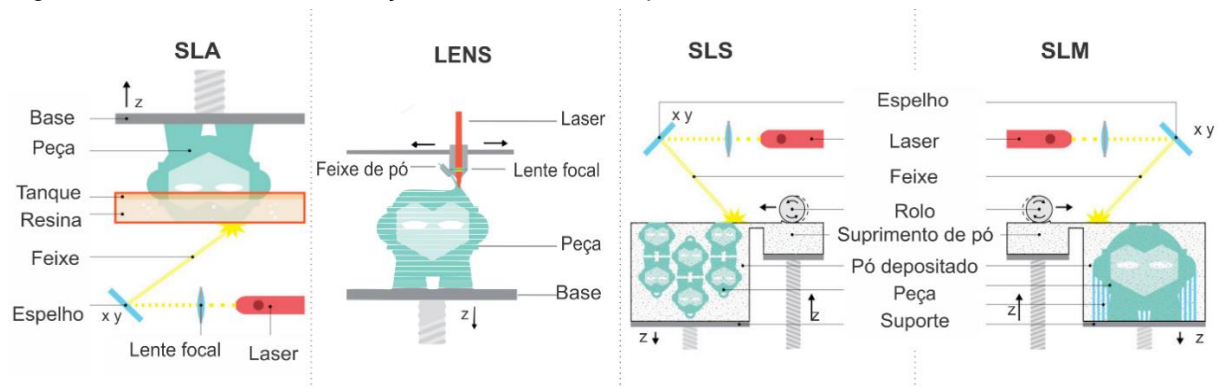
Existem ainda outros métodos de AM que também podem ser encontrados em utilização, alguns usados para produzir peças mais detalhadas, outros para peças mais resistentes, alguns para peças coloridas ou até para produção de peças orgânicas. Esta seção destina-se a apresentar o funcionamento, os elementos essenciais (indicados na Figura 2), as vantagens e desvantagens de alguns métodos de AM. O Anexo A complementa com mais métodos, organizados em uma estrutura de árvore, com os materiais utilizados de acordo com os métodos de AM, e agrupados pela técnica de solidificação.

O método de Estereolitografia (SLA) é considerado o primeiro método de AM e o mais amplamente utilizado. O processo de impressão pode ser feito com a peça na posição normal ou invertida, onde o material (líquido fotossensível) é colocado em um recipiente que cobre toda a base, caracterizando-o como deposição por camada. A solidificação ocorre através da radiação ultravioleta, e conforme a peça vai sendo construída, a base vai emergindo ou submergindo no líquido. Este método consegue melhores acabamentos em superfícies do que outros métodos como o FDM, por exemplo. Entretanto o preço deste tipo de impressora 3D ainda é superior aos equipamentos FDM disponíveis, e seu material tem um preço superior aos filamentos de ABS e PLA por quilo (LIVESU et al., 2017; MONTEIRO, 2015).

O método de Modelagem por Laser de Engenharia (LENS – *Laser Engineering Net-Shape*) funciona com a deposição local de jatos de pó que fundem no ponto focal do laser unindo-se com as camadas inferiores da peça. Neste método são utilizados materiais como titânio, aço inoxidável e alumínio. Sua peça final apresenta uma densidade uniforme e boa microestrutura, com propriedades similares ou até mesmo melhores que de alguns metais convencionais. No entanto, é um método de produção muito caro que exige pós-processamento de usinagem do metal e lapidação, o que demanda ainda mais tempo e mais custo (POPOOLA et al., 2016).

Os métodos de SLS e fusão seletiva a laser (SLM - *Selective Laser Melting*), realizam a deposição de material por camada, através de um rolo que aplica uma fina camada de pó sobre a superfície de impressão e em seguida sinteriza o pó nos pontos específicos da camada. O pó utilizado na impressão pode variar entre diversos tipos de substratos, como polímeros, metais, cerâmicas e plásticos. A desvantagem deste método é que ele exige máquinas de grande porte, pois necessita de um ambiente selado e preenchido com nitrogênio ou fechado a vácuo, evitando a entrada de oxigênio no processo de fusão do material. O processo também deve ocorrer em temperatura constante e próxima da temperatura de fusão do material, implicando no aumento do tempo de produção, devido a espera necessária para resfriamento da peça (MONTEIRO, 2015).

Figura 2 – Métodos de solidificação dos materiais em processos de AM.



Fonte: Adaptado de Obsessively Geek 3D [<http://www.og3dprinting.com/3d-printing/>]

## 2.2 MÉTRICAS DE AVALIAÇÃO DOS MÉTODOS DE AM

As métricas de avaliação são divididas em três categorias: custo, fidelidade e funcionalidade, juntas elas servem para medir a eficiência do método utilizado na fabricação das peças em AM. No entanto, também podem ser utilizadas para avaliar o desempenho do fatiador desenvolvido (AMF-AC), realizando comparações entre outros fatiadores no mesmo método de AM.

### 2.2.1 Custo

O custo de produção pode ser calculado e dividido em três fases: pré-produção, produção e pós-produção. A soma do custo das três fases é essencial para o planejamento das empresas que utilizam AM. Ainda que em alguns casos pode haver a necessidade de repetir determinadas fases do processo, devendo ser adicionadas no custo total (ALEXANDER et al., 1998).

- **Pré-produção:** O custo é calculado pelo somatório do tempo de trabalho gasto, nas etapas de configuração da máquina de AM, fatiamento do modelo e geração dos comandos de máquina;
- **Produção:** O cálculo da produção é o somatório de tempo gasto percorrendo os trechos sem efetivamente agregar material a peça, com o tempo gasto nos trechos em que há construção da peça. Inclui também trechos de construção de suporte/âncora e base de apoio quando houver. Por fim, são somados os custos de funcionamento da máquina, e o custo dos materiais utilizados;
- **Pós-produção:** Ao finalizar a impressão da peça é necessário medir o tempo para remover os suportes e materiais excedentes. E registrar também o custo dos materiais utilizados se houver a necessidade de polimento e/ou pintura de acabamento.

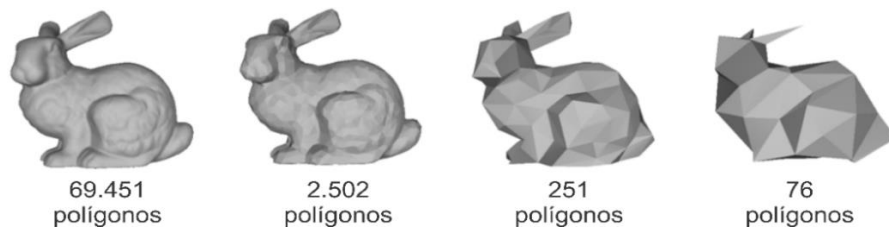
## 2.2.2 Fidelidade

A Fidelidade é uma característica muito importante nos processos de fabricação, pois representa o quanto a peça final está semelhante ao modelo projetado no *software* CAD. No cálculo da fidelidade são utilizadas medidas de erros associados com a simplificação do modelo CAD, com o fatiador de sólidos ou com a mecânica do método de impressão. Esses erros podem ser calculados utilizando metodologias de GD&T, que avaliam as dimensões da peça física, ou através de simulações computacionais.

### 2.2.2.1 Erros de simplificação do modelo CAD

Ao representar o modelo CAD através de uma malha poligonal, são necessárias aproximações das superfícies curvas do CAD; essas aproximações podem variar e implicam no nível de detalhamento, em inglês *Level of Detail* (LOD), o que influencia na quantidade de polígonos da malha, exemplificado na Figura 3.

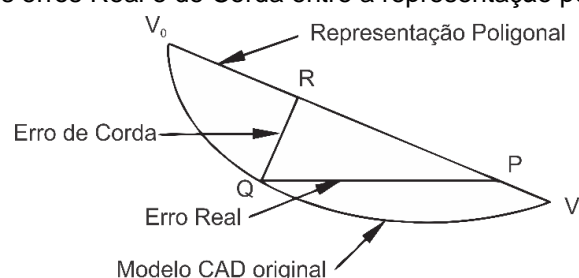
Figura 3 – Modelo 3D representado no padrão STL em diferentes LODs de malha poligonal.



Fonte: adaptado de [<http://slideplayer.com/slide/6993336/>]

Para avaliar o erro das representações poligonais em relação ao modelo CAD, são usados os somatórios das distâncias  $\overline{QR}$  e  $\overline{QP}$ , ilustradas na Figura 4, essas representam respectivamente o Erro de Corda e o Erro Real. O Erro de Corda é calculado pelo comprimento da maior distância entre a superfície CAD e as arestas utilizadas na representação poligonal. Para calcular o Erro Real apresentado por Zhao et al. (2009), é preciso calcular a distância horizontal entre o ponto Q até a aresta  $\overline{V_0V_1}$ , o autor indica esta abordagem como mais apropriada para o cálculo do erro em processos de AM devido ao fatiamento que também ocorre horizontalmente.

Figura 4 – Localização dos erros Real e de Corda entre a representação poligonal e o modelo CAD.



Fonte: Traduzido de Zhao et al. (2009).

### 2.2.2.2 Erros do fatiador de sólidos

Em geral o fatiamento de superfícies planas não acarreta erros, com exceções de casos especiais, como o fatiador de Kim et al. (2010), que desloca vértices para evitar sua intersecção sobre a camada, provocando uma pequena alteração na peça.

Já os fatiadores de superfícies com curvas paramétricas (NURBS, Bézier e Hermite *patches*) em AM necessitam realizar aproximações das curvas ou simplificações destas para encontrar sua intersecção com o plano.

O erro gerado por intersecções através da simplificação da curva pode ser calculado da mesma forma que o Erro de Corda. No entanto, se o fatiador realiza a aproximação da curva pelos pontos de intersecção, é possível calcular o valor do erro através do somatório das distâncias entre os pontos de intersecção com a curva gerada. Esse erro pode ser calculado pelo método dos mínimos quadrados, expresso na Equação (2.1), onde  $Q_k$  representa os pontos reais de intersecção e  $C(\bar{u}_k)$  a curva paramétrica calculada.

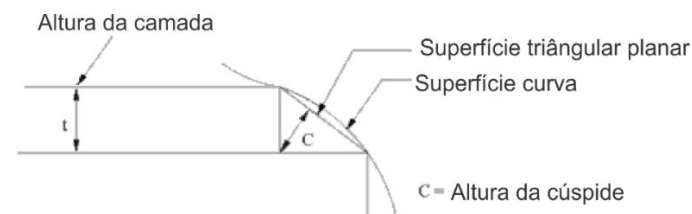
$$e = \sum_{k=0}^j |Q_k - C(\bar{u}_k)|^2 \quad (2.1)$$

A partir das camadas obtidas pelo fatiador também é possível realizar comparações entre suas áreas, desta forma pode-se avaliar os resultados dos métodos de simplificações aplicados aos modelos 3D. No entanto, deve-se garantir que valor da área não se subtraia nas curvaturas com concavidades opostas nos contornos que compõem essas camadas.

### 2.2.2.3 Erros do método de impressão

Uma das causas de perda de fidelidade acontece devido ao empilhamento de camadas dos métodos de AM, conhecido também como “efeito escada”. Uma das técnicas para avaliar essa característica é o cálculo da altura da cúspide, representado pelo caractere  $C$  na Figura 5, considerando as bordas das camadas com formas retangulares (DOLENC e MÄKELÄ, 1994) ou angulares (HOPE et al., 1997).

Figura 5 – Representação da altura da cúspide em camadas de borda retangulares.



Fonte: Traduzido de PANDEY et al., 2003.

Outro parâmetro para avaliar a fidelidade do protótipo é a diferença entre o volume do modelo CAD e o volume da peça. Este parâmetro é mais indicado para medir o efeito escada do que a altura da cúspide (C) (TAUFIK e JAIN, 2014). O método da diferença volumétrica foi proposto por Masood et al. (2000), primeiramente testado nos sólidos primitivos (cilindro, cubo, pirâmide e esfera) e posteriormente aplicado em estruturas mais complexas.

### 2.2.3 Funcionalidade

É comum que, ao fabricar um protótipo de um determinado projeto, se deseje testar sua funcionalidade. Como por exemplo, verificar a aerodinâmica de um protótipo de modelo de carro ou avião. Para isto, serão submetidas forças sobre o protótipo que, portanto, deve ser avaliado sobre o aspecto da resistência mecânica para que não sofra rupturas durante testes. Zhou et al. (2013) desenvolveram um algoritmo para analisar a resistência mecânica da estrutura do modelo, identificando pontos de fragilidade.

A fabricação de peças por AM trouxe novos parâmetros na área de *design*. Modelos que até então eram apenas para estudos teóricos, passaram a se tornar possíveis de fabricação, assim como novas estruturas com melhor aproveitamento de calor, luz e som também puderam ser testadas. Neste contexto, novos trabalhos apresentam algumas formas de avaliação de funcionalidades como: distribuição de peso, dissipação de calor e propagação de ondas sonoras (DEDE et al., 2015; BRACKETT et al., 2011; PRÉVOST et al., 2016).

## 2.3 PADRÕES DE REPRESENTAÇÃO

Os padrões de representação apresentados nesta seção são referentes aos arquivos utilizados para exportar os modelos 3D, de um programa CAD para o *software* da impressora. Esses arquivos são usados na configuração inicial no processo de fabricação em AM para geração das camadas de composição da peça.

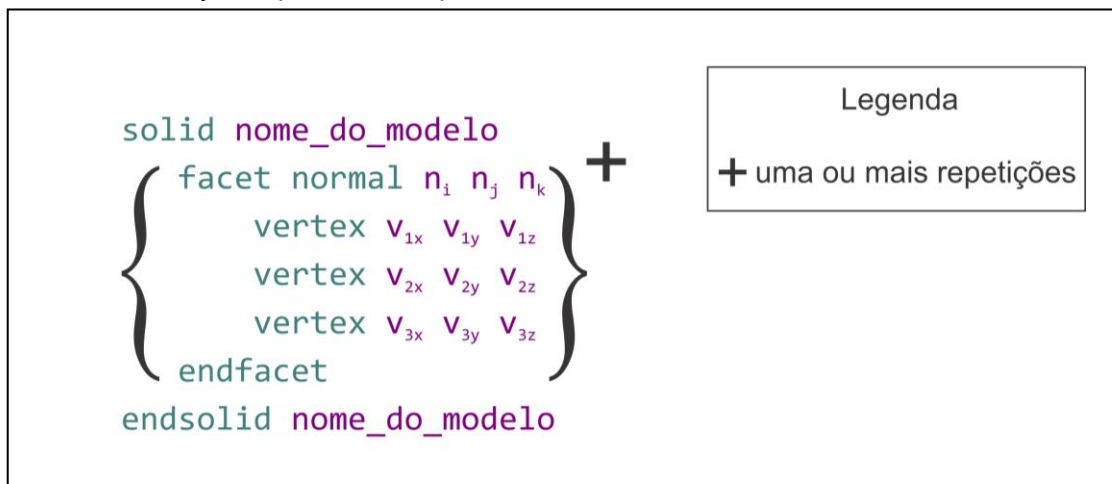
Os arquivos STL, AMF e STEP (*Standard for the Exchange of Product Model data*), são alguns exemplos de arquivos utilizados atualmente na AM. Eles podem representar o modelo com ou sem a simplificação do modelo CAD original, para isso são utilizadas estruturas de malhas de triângulos, superfícies biquadráticas, *sweeping* e *B-Spline*, entre outras (ZHOU, 2005). A seguir são apresentadas as principais características estruturais desses padrões de arquivos.

### 2.3.1 STL

O STL, também conhecido como Linguagem Triangular Padrão ou Linguagem de Tecelagem Padrão, foi criado inicialmente pela 3D Systems em 1987, como linguagem nativa do *software* Stereolithography CAD, é atualmente conhecido como o arquivo padrão para importação em *software* para impressoras 3D, devido a sua simplicidade estrutural e pela facilidade proporcionada no processo de fatiamento (GRIMM, 2004; ISO/ASTM 52915:2016)

Contendo uma representação do sólido a partir da malha de triângulos planos, o STL pode ser descrito em ASCII, com a estrutura básica ilustrada no Quadro 1, semelhante ao padrão XML (*Extensible Markup Language*). Sua linguagem estrutural inicia-se abrindo a seção do sólido com o comando *solid*, seguido do nome do modelo, e dentro da seção *solid* estão as seções *facet*. Para cada triângulo que compõem a malha do modelo, é criada uma seção *facet*, que é composta pela normal da face e os vértices do triângulo.

Quadro 1 – Codificação Explícita dos arquivos STL em ASCII.



Fonte: Produção do autor.

Esta estrutura apesar de simples, serve para representação aproximada de modelos complexos (com superfícies curvas e irregulares) e tem sido largamente utilizada nos processos de AM, porém novas tecnologias de AM possibilitam maior detalhamento da superfície do modelo. Para se representar este detalhamento em arquivos STL, é necessário um número exponencialmente maior de triângulos, comprometendo sua eficiência na etapa de fatiamento e também tornando os arquivos STL muito grandes.

### 2.3.2 AMF

Outro padrão desenvolvido para a utilização em AM foi o *Additive Manufacturing File Format* (AMF), que descreve os modelos 3D utilizando malhas triangulares curvas (ISO/ASTM 52915:2016). Este padrão foi apresentado inicialmente em 2013 pela norma ISO/ASTM 52915:2013, mas foi revisado e publicado novamente em 2016 o qual está em processo de revisão outra vez.

As vantagens do padrão AMF em comparação com o STL estão relacionadas a utilização desses triângulos curvos, que melhoram a precisão e reduzem a quantidade de dados necessários para representar um modelo se comparado à mesma fidelidade de um modelo em STL. O AMF também traz outras vantagens para a AM como: a capacidade de armazenar informações de diferentes tipos de materiais e texturas (multimateriais), informações de orientação do modelo, múltiplas definições de modelos e agrupamentos para impressão em sequência.

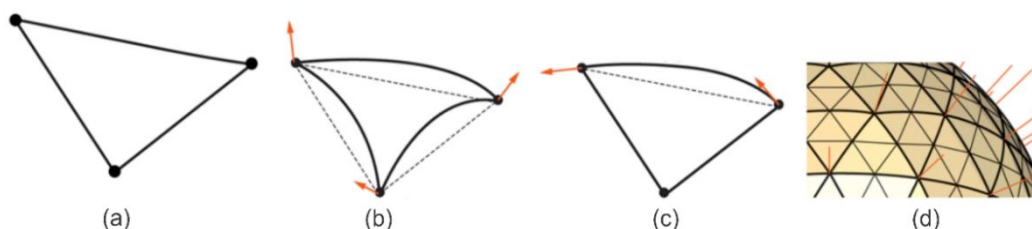
#### 2.3.2.1 Estrutura do Arquivo

Os triângulos curvos em AMF são definidos com curvas cúbicas de Hermite em suas arestas, e essa estrutura permite a divisão de cada triângulo curvo em novos triângulos curvos menores, conforme representado na Figura 6 (d). Outra vantagem do AMF é a compatibilidade com o padrão STL, pois permite também a representação dos triângulos planos através de 3 vértices sem curvaturas em suas arestas (Figura 6 (a)).

As curvas que compõem os triângulos devem estar expressas na estrutura do arquivo através das normais de seus vértices, como na Figura 6 (b), ou através das tangentes dos vértices, como ilustra a Figura 6 (c). Curvas definidas de formas diferentes podem estar presentes nas arestas de um mesmo triângulo, o que possibilita outras combinações no mesmo triângulo, além das apresentadas na Figura 6.

Outra consideração sobre a estrutura é que, para um vértice com uma normal definida, implica que todas as arestas adjacentes a ele tenham uma curvatura contínua no vértice. Para casos em que a normal não esteja definida, mas que uma tangente definida no vértice exista, a curvatura não precisa ser necessariamente contínua no vértice.

Figura 6 – (a) triângulo plano definido por 3 vértices, (b) triângulo curvo definido por 3 normais, (c) triângulo curvo definido por 2 tangentes, (d) subdivisão dos triângulos curvos.

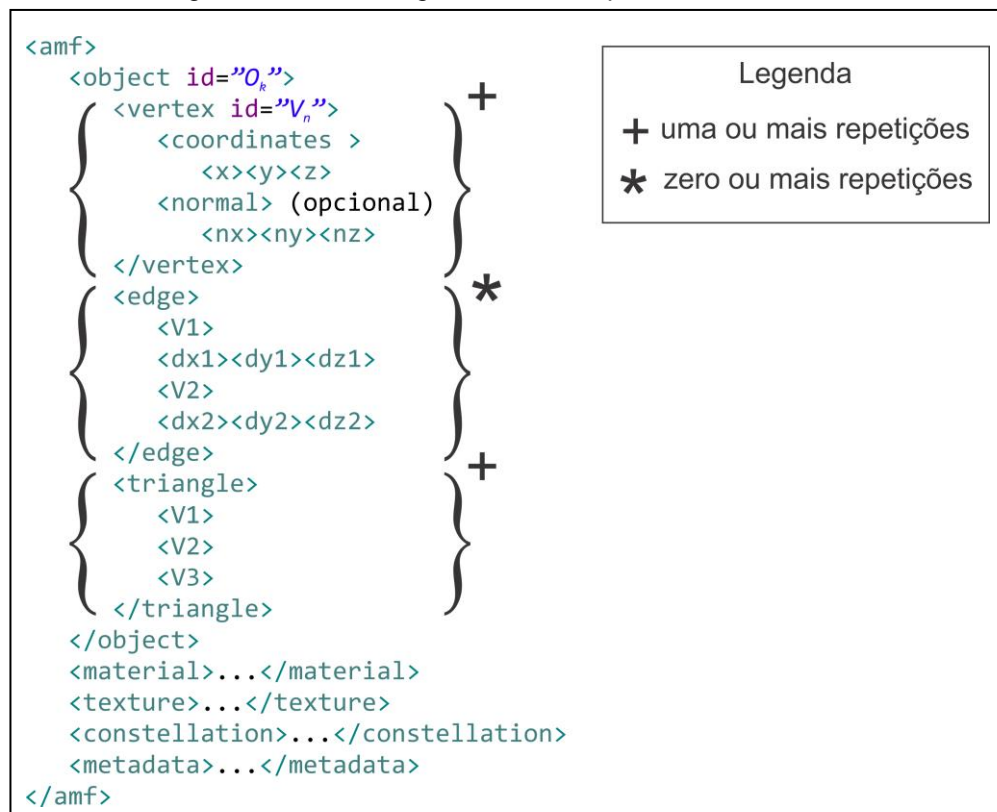


Fonte: ISO/ASTM 52915:2016.

A linguagem utilizada para descrever os modelos 3D no padrão AMF é baseada em *Extensible Markup Language* (XML), e pode ser facilmente exportada por programas CAD (ISO/ASTM 52915:2016).

O Quadro 2 mostra as principais *tags* (seções) que compõem o arquivo AMF, nele os objetos (*objects*) são compostos por uma lista de vértices (*vertex*), uma lista de triângulos (*triangles*) e opcionalmente uma lista de arestas (*edges*). O vértice (*vertex*) é composto por suas coordenadas cartesianas e opcionalmente por uma normal, e ele também deve receber um identificador (*id*) de acordo com a ordem de apresentação no arquivo. A aresta (*edge*) é composta pelos identificadores de dois vértices e uma tangente para cada vértice identificado. E o triângulo (*triangle*) é definido por 3 identificadores de vértices.

Quadro 2 – Estrutura organizacional das *tags* utilizadas no padrão AMF



Fonte: Produção do autor.

As *tags material*, *texture*, e *metadata*, servem para descrever informações de textura e características físicas dos materiais que devem ser utilizados na produção da peça em AM.

A *tag constellation* descreve uma lista de objetos com posicionamentos e rotações para impressões em grande quantidade.

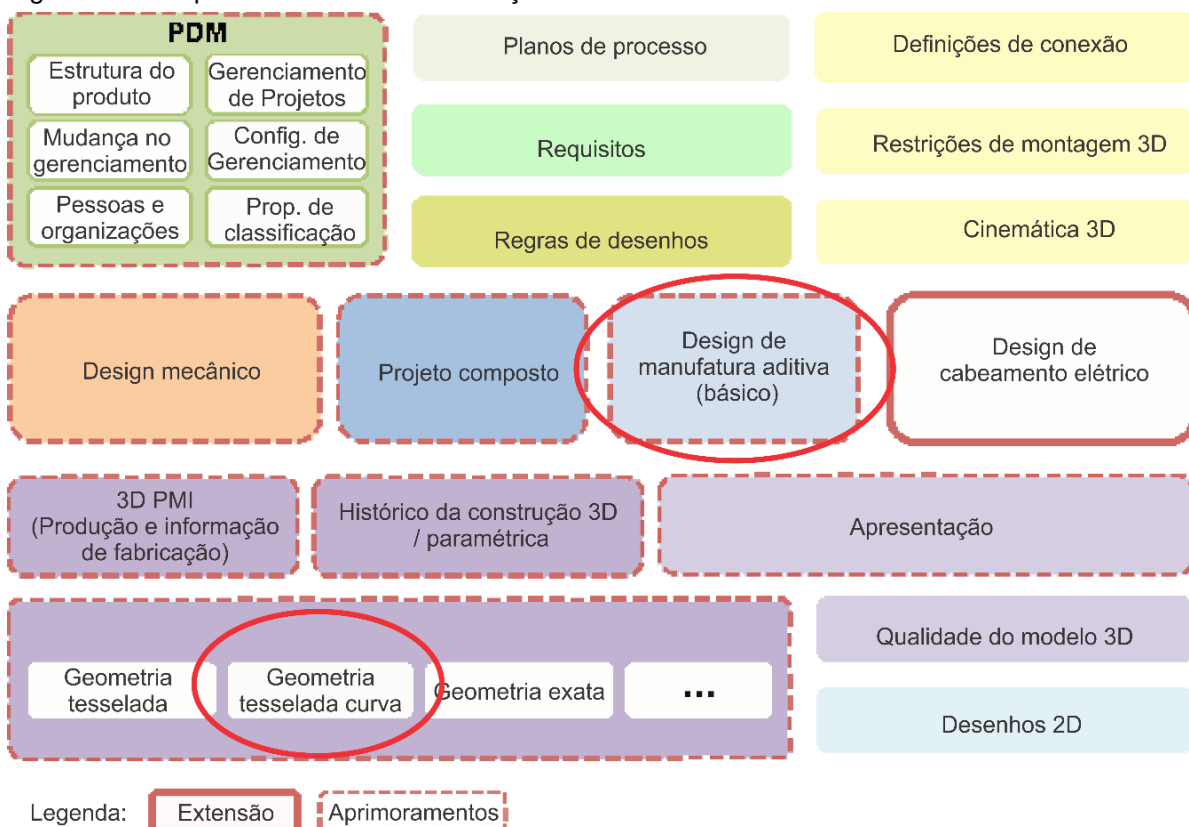
### 2.3.3 STEP

O padrão *Standard for the Exchange of Product Model Data* (STEP) é o nome informal atribuído a ISO 10303-1:1994, criada com o objetivo de propiciar interoperabilidade entre os programas CAD, CAM, CAE (ZHOU, 2005).

Em 2014, as indústrias automotiva e aeroespacial, em uma iniciativa conjunta criaram a primeira edição da AP 242 que complementou o padrão STEP, trazendo avanços na padronização das representações e processos de produção. Porém, atualmente as maiores companhias de desenvolvimento de *software* CAD, têm agregado funcionalidades dos processos de AM em seus *software*. E para que o padrão STEP continue a promover a interoperabilidade entre esses *software* é essencial que sejam complementadas essas funcionalidades. Com base nisso, está em desenvolvimento a segunda edição da ISO 10303-242:2014, com seu escopo de aplicação ilustrado na Figura 7, que propõe aprimoramentos direcionados para a fabricação de peças em AM com novas geometrias 3D e parâmetros de impressão.

Entre as geometrias 3D agregadas na segunda edição da AP 242, está a malha de triângulos curvos utilizada no padrão AMF (em destaque na Figura 7), o que evidencia a importância deste tipo de representação nos processos de AM.

Figura 7 – Escopo da STEP AP 242 2ª edição.



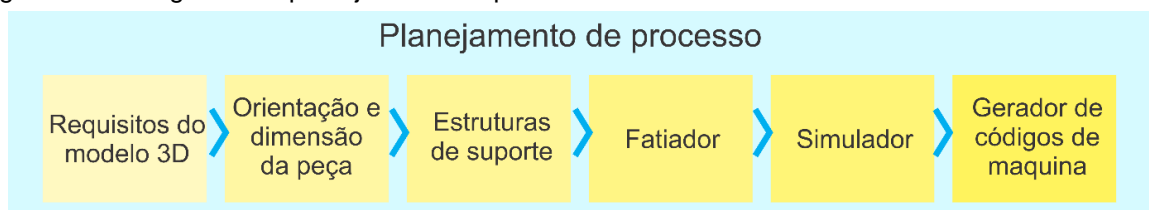
Fonte: Traduzido e adaptado de STEP AP242 Project [<http://www.ap242.org/edition-2/>]

## 2.4 PLANEJAMENTO DE PROCESSO (PP)

Planejamento de processo é a fase que acontece entre a criação do modelo 3D e a fabricação da peça. Inicialmente definida em quatro etapas: definição de orientação, geração de suporte, fatiamento e geração da linguagem de controle numérico (CN) (MARSAN et al., 1997). Entretanto outras pesquisas incluem novas etapas, alguns autores consideram a etapa de correção e ajustes do modelo como primeira do PP, outros incluem a escolha dos parâmetros de impressão no processo (LIVESU et al., 2017).

Esta pesquisa aborda o PP da forma sequencial e organizada em 6 etapas (ilustradas na Figura 8):

Figura 8 – Fluxograma do planejamento de processo.



Fonte: Produção do autor.

- Requisitos do modelo 3D são as técnicas associadas a inspeção do modelo 3D por falhas de modelagem, ou seja, encontrar problemas na superfície que possam atrapalhar a fabricação da peça;
- Orientação e dimensão da peça é a etapa responsável por definir a melhor orientação para a fabricar a peça dado um objetivo, aqui são definidos também parâmetros de dimensionamento e agrupamentos de peças;
- Estruturas de suporte geradas nesta etapa são responsáveis por sustentar regiões da peça onde não há base para deposição do material em AM;
- Fatiador é a etapa fundamental do planejamento, pois aqui são calculadas intersecções entre o modelo 3D com o plano das camadas que compõem os contornos da peça;
- Simulador é o *software* utilizado para visualizar as camadas definidas pelo fatiador, esta etapa é importante para verificar o processo de fatiamento antes de iniciar a fabricação da peça;
- Gerador de códigos de máquina, nesta etapa os dados dos contornos obtidos pelo fatiador são convertidos em comandos de linguagem de CN que definem os trajetos a serem percorridos de acordo com o *firmware*<sup>1</sup> da impressora 3D.

<sup>1</sup> *Software* de baixo nível utilizado para controlar a impressora 3D.

### 2.4.1 Requisitos do modelo 3D

Quando modeladores projetam peças, nem sempre podem se preocupar com as definições do modelo e simultaneamente com as dificuldades da fabricação por AM.

Em alguns casos a utilização de interações booleanas em superfícies detalhadas, acarretam pequenas falhas imperceptíveis ao usuário do *software* CAD. Attene (2010) comenta em sua pesquisa sobre a dificuldade do usuário em identificar essas falhas na malha representativa, que ao ser submetida ao fatiador, estará suscetível a erros durante o processamento das camadas, caso o algoritmo de fatiamento não seja capaz de analisar estas falhas.

Outra verificação feita nesta etapa é com relação ao formato do modelo, aqui verifica-se se o modelo terá problemas durante fabricação por AM, o trabalho desenvolvido por Telea e Jalba (2011) analisa a representação tridimensional identificando partes do modelo com espessura menor que a especificada pelo método de AM. Semelhante a este, Rolland-Neviere et al. (2013) também analisa o modelo representado por malha de triângulos quanto a espessura.

Outros trabalhos propuseram soluções mais automatizadas, aplicando correções diretas na forma do modelo. Chen e Wang (2013) desenvolveram um algoritmo capaz de aumentar a espessura de uma superfície, muito eficaz na fabricação de superfícies obtidas por malhas de pontos em *scanners* 3D. A pesquisa de Luo et al. (2012) também traz uma abordagem de correção automatizada que efetua o seccionamento do modelo em partes menores cabíveis na impressora 3D.

### 2.4.2 Orientação de impressão

Esta outra etapa do PP está associada a orientação angular escolhida para fabricação da peça. Ao rotacionar o modelo 3D, ocorrem alterações que podem contribuir positivamente ou negativamente na produção da peça. Estruturas de suporte podem ser minimizadas ou até suprimidas dependendo da escolha de orientação do modelo. Assim como faces ficam melhor definidas quando estão entre ângulos pequenos com o eixo de impressão, outras alterações podem ocorrer afetando a resistência, o tempo de produção, entre outros aspectos de fidelidade e funcionalidade da peça.

No passado, os trabalhos priorizavam a escolha de apenas uma métrica (custo, fidelidade ou funcionalidade) ao rotacionar o modelo (HUR e LEE, 1998; SABOURIN et al., 1997). No entanto, atualmente já há propostas com técnicas de aproximação e classificadores, que conseguem otimizar várias funcionalidades simultaneamente (WANG et al., 2016).

### 2.4.3 Estruturas de suporte

Teoricamente, AM é capaz de fabricar qualquer tipo de forma ou estrutura complexa de sólidos modelados em 3D preenchendo camada após camada até que a peça esteja completamente pronta. No entanto, algumas considerações devem ser feitas levando em conta a força da gravidade, pois muitos dos métodos de fabricação em AM dependem de uma base para deposição do material. Para suprir essa necessidade, pesquisadores desenvolveram técnicas de geração de suporte para saliências do sólido que não possuem base de apoio durante a impressão 3D, atualmente essa é uma importante etapa do PP na fabricação de estruturas 3D complexas (HU et al., 2015).

Categorizado como externo e interno, o suporte pode estar presente externamente visível e removível ou internamente fazendo parte da estrutura da peça. Existem diversas técnicas atuais de geração de suporte, em sua maior parte as técnicas tentam otimizar o uso de material e o tempo de impressão com a redução do volume de suporte (LIVESU et al., 2017):

- O suporte externo é necessário para saliências da superfície externa que ultrapassem determinado ângulo de inclinação e tamanho. Pode ser encontrado também em situações em que ocorram ilhas de solidificação, onde o material é preenchido isoladamente do restante da peça. Existem alguns métodos conhecidos para criação de suporte externo: métodos que copiam topologias de árvores, outros com estruturas geométricas unidas e algumas estruturas de suporte com parede inclinada (HUANG et al., 2009; CACACE et al., 2014; JIN et al., 2015). Diferente das abordagens anteriores Hu et al. (2015) propuseram uma técnica que altera a forma do modelo 3D otimizando o volume de suporte necessário;
- O suporte interno (preenchimento interno) está relacionado a região da peça com maior influência na economia de material e tempo de impressão, pois o volume interno da peça cresce cubicamente em relação as outras dimensões de tamanho (LIVESU et al., 2017). Esse suporte pode ser definido de forma densa ou esparsa, e sua utilização está ligada ao método de fabricação de AM. Os métodos de impressão com deposição de material por solidificação de camada normalmente não utilizam suporte interno esparsa, pois o material interno contido nos espaços vazios não pode ser reaproveitado. Por outro lado o uso de suporte interno esparsa em métodos de solidificação local consegue aproveitar padrões geométricos existentes que reduzem o material no preenchimento esparsa (ZHAO et al., 2016; MAKHANOV e ANOTAIPAIBOON, 2007; GRIFFITHS, 1994; COX et al., 1994 apud LIVESU et al., 2017).

## 2.4.4 Fatiamento

O fatiamento é a etapa de maior importância do processo de planejamento em AM, pois é nessa etapa que o modelo 3D é dividido em camadas de contorno utilizadas para fabricação da peça em AM (LIVESU et al., 2017), sem o qual não se produz a peça.

Para obter os contornos, o fatiador deve interseccionar a superfície do modelo 3D com camadas em diferentes alturas. Com isto, as abordagens de aprimoramentos estão relacionadas principalmente em três características:

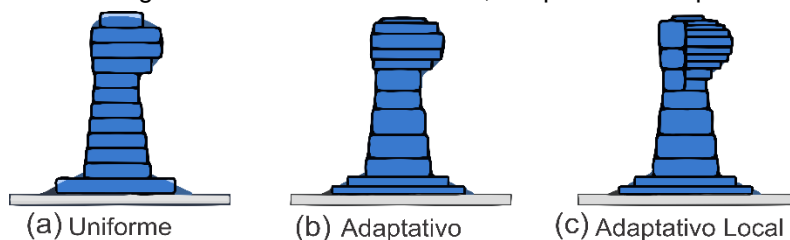
- Variação da altura entre as camadas;
- Fatiamento em diferentes representações;
- Ajuste de pontos de intersecção.

### 2.4.4.1 Tamanho de camada

O cálculo de tamanho de camada foi um dos primeiros aprimoramentos estudados nos fatiadores, e tem como objetivo reduzir o tempo de impressão sem comprometer a qualidade da superfície (KULKARNI et al., 2000).

Os trabalhos relacionados ao tamanho da camada foram propostos inicialmente utilizando tamanho uniforme para todas as camadas da peça (Figura 9 (a)), nessa altura das camadas é dada com base no erro de fidelidade da peça, (KIRSCHMAN e JARA-ALMONTE, 1992).

Figura 9 – Diferentes abordagens de fatiamento: uniforme, adaptativo e adaptativo local.



Fonte: Traduzido de LIVESU et al. (2017).

Fatiadores com variações de tamanhos de camadas ao longo do modelo 3D, são chamados fatiadores adaptativos (Figura 9 (b)), e conseguem ganhos significativos na redução de tempo de impressão, no entanto provocam perda de fidelidade em alguns locais da peça (DOLENC; MÄKELÄ, 1994; SUH; WOZNY, 1994).

Por fim a abordagem de Sabourin et al. (1997), chamada de fatiamento adaptativo local, subdivide o modelo em partes verticais e aplica o fatiamento com camadas de tamanhos diferentes (Figura 9 (c)), essa abordagem é limitada ao espaço físico ocupado pelo deslocamento da extrusora em impressoras FDM.

#### 2.4.4.2 Algoritmos de fatiamento

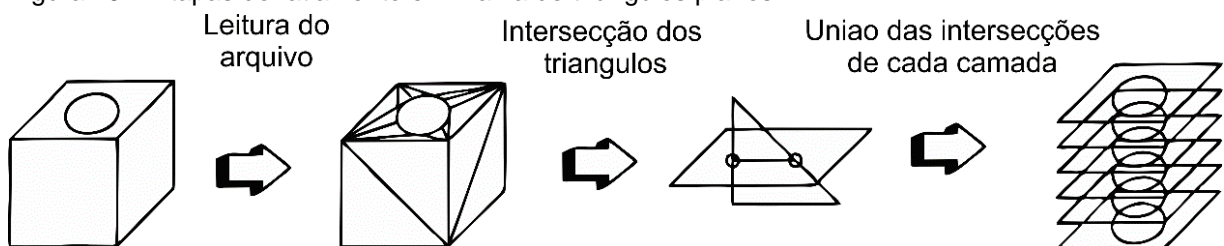
Os algoritmos de fatiamento são responsáveis por encontrar os contornos resultantes da intersecção do modelo com as camadas, esse processamento está fortemente relacionado com a representação do modelo 3D (LIVESU et al., 2017).

Em uma pesquisa sistemática realizada neste trabalho, foi constatado que as representações de malha triangular (STL), representações originais do CAD e a malha de triângulos curvos, destacam-se por serem as mais utilizadas nos fatiadores existentes.

##### 2.4.4.2.1. Fatiamento em malhas triangulares

O propósito básico dos algoritmos de fatiamento em malhas triangulares é encontrar todos os segmentos de retas que interseccionam os triângulos da malha com as camadas, e as unir formando o contorno da peça (KIRSCHMAN e JARA-ALMONTE, 1992). A Figura 10 ilustra as etapas realizadas no fatiador paralelo de Kirschman e Jara-Almonte (1992), esse realiza as intersecções das camadas de forma paralela e independente umas das outras, proporcionando um melhor aproveitamento dos processadores multicores que permitem trabalhar em um ambiente multitarefa.

Figura 10 – Etapas do fatiamento em malha de triângulos planos.



Fonte: Adaptado de Kirschman e Jara-Almonte (1992).

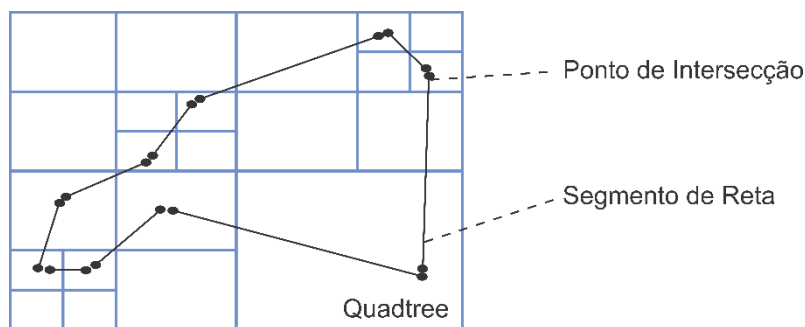
Mcmains e Séquin (1999) desenvolveram um fatiador de STL que funciona a partir dos pontos de intersecção da primeira camada, e segue com o cálculo do deslocamento de cada ponto nas camadas seguintes. Essa abordagem reduz o tempo de processamento utilizado na união dos segmentos de reta obtidos na intersecção, pois aplica o deslocamento dos pontos de intersecção somente quando o contorno da camada muda em relação ao anterior, isso reduz a quantidade de contornos processados.

Se a malha de triângulos estiver de acordo com os requisitos de definição do sólido fechado, as intersecções deverão formar contornos isolados uns dos outros (sem intersecções entre eles). Porém, como a malha de triângulos representada no padrão STL não contém informações sobre triângulos adjacentes, é preciso definir a sequência e direção das arestas para poder as unir adequadamente e formar o contorno da camada (KIRSCHMAN e JARA-ALMONTE, 1992).

Esse processo de união pode ser feito de forma exaustiva, procurando quais pontos de intersecção estão sobre o mesmo local analisando todos os pontos obtidos, ou pode ser feito com a ajuda de classificadores para separar os pontos com menor distância entre eles.

Para realizar a união dos segmentos de retas, o fatiador *Slic3r*<sup>1</sup> de código-aberto utiliza a estrutura *Quadtree* para separar o conjunto de pontos em pequenas regiões. No *Slic3r*, o algoritmo que faz essas divisões na área do contorno é simples de ser implementado, são necessárias apenas divisões sucessivas no plano da camada até que cada subdivisão contenha apenas 2 pontos de intersecção de segmentos de retas diferentes, como exemplifica a Figura 11.

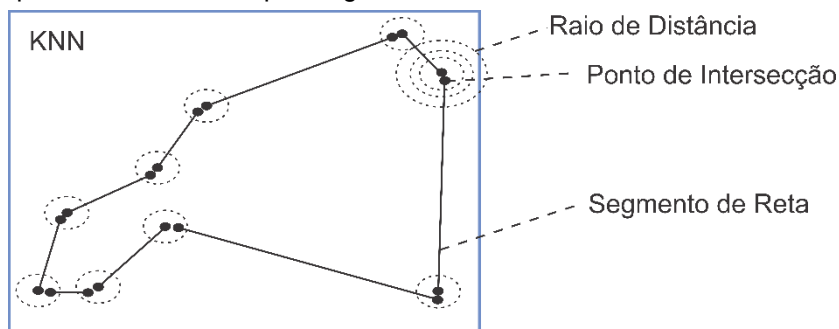
Figura 11 – Divisão de vértices utilizando Quadtree.



Fonte: Produção do autor.

Já o *Skeinforge*<sup>2</sup> que é outro fatiador de código-aberto, faz a união dos segmentos de retas utilizando o algoritmo *K Nearest Neighbors* (KNN), que agrupa os pontos mais próximos (Figura 12). O KNN também é encontrado em outras abordagens de AM, como na construção de contornos de camadas através de nuvem de pontos em modelos obtidos por *scanner* 3D (PERCOCO e GALANTUCCI, 2008).

Figura 12 – Agrupamento de vértices pelo algoritmo KNN



Fonte: Produção do autor.

<sup>1</sup> <http://reprap.org/wiki/Slic3r>, disponível em 01/02/2017.

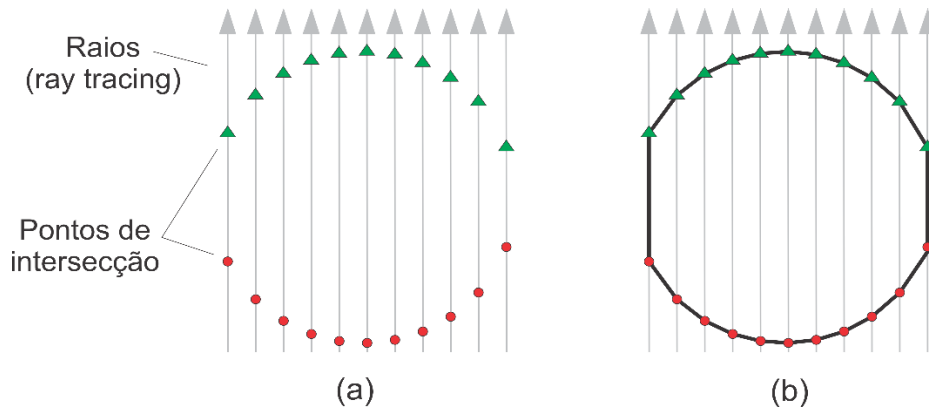
<sup>2</sup> <http://reprap.org/wiki/Skeinforge>, disponível em 01/02/2017.

#### 2.4.4.2.2. Fatiamento em arquivos CAD

O padrão de arquivo STL é o padrão mais utilizado nos *software* de impressoras 3D, mas suas limitações representativas reduzem a fidelidade do modelo e consequentemente da peça (PANDEY et al., 2003). Considerando isso, alguns métodos de fatiamento têm optado por trabalhar com as representações originais dos modeladores CAD (como por exemplo o padrão STEP), esses são chamados de fatiadores diretos (UM et al., 2017). Os fatiadores diretos também contribuem com a redução do tempo gasto em transformações de arquivo e eliminam o espaço de memória gasto em armazenamento com o STL.

Os trabalhos de Starly et al. (2005) e Zeng et al. (2011) propuseram fatiadores de Superfícies *B-Splines* Racionais Não-Uniformes (NURBS) em arquivos do tipo CAD, os autores utilizam a intersecção entre a superfície com retas do algoritmo de *ray-tracing*, uma técnica utilizada para simulação dos raios de luz. Com esta técnica conseguem encontrar os pontos de intersecção (Figura 13 (a)) com as NURBS e interliga-los com segmentos de retas (Figura 13 (b)) que compõem os contornos das camadas.

Figura 13 – Aplicação das regras de ligação dos pontos na proposta de Zeng et al. (2011).



Fonte: Adaptado de Zeng et al. (2011).

Zhou (2005) utiliza o padrão STEP *Part 42* para representar o modelo 3D e seu algoritmo de fatiamento transforma cada superfície NURBS em pequenas malhas de triângulos planos que são interseccionadas separadamente com cada camada, para então unir os pontos encontrados formando o contorno das camadas.

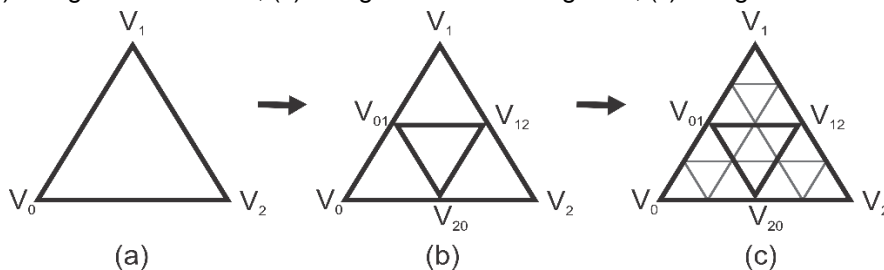
Também em fatiadores diretos os algoritmos de fatiamento ainda necessitam de aproximações para encontrar o contorno de intersecção, essas aproximações causam simplificações no modelo 3D, e impactam na fidelidade da peça fabricada em AM (VAIDYA e ANAND, 2016).

### 2.4.4.2.3. Fatiamento em arquivos AMF

Na especificação do padrão AMF, determina-se que o fatiamento seja realizado após duas subdivisões dos triângulos curvos, e que sejam utilizados os mesmos algoritmos de fatiamento em malhas triangulares planas para encontrar o contorno das camadas (ISO/ASTM 52915:2016).

O processo de subdivisão do triângulo funciona através de divisões feitas em suas arestas definidas por curvas de Hermite, na Figura 14 (a) é mostrado um exemplo de como são feitas essas divisões para o triângulo ( $V_0$ ,  $V_1$  e  $V_2$ ). O número de divisões realizadas em cada triângulo curvo é chamado de grau de profundidade, e ao realizar uma divisão de grau 1 nesse triângulo são obtidos quatro novos triângulos definidos em: ( $V_0$ ,  $V_{01}$ ,  $V_{20}$ ), ( $V_{01}$ ,  $V_1$ ,  $V_{12}$ ), ( $V_{20}$ ,  $V_{12}$ ,  $V_2$ ), ( $V_{01}$ ,  $V_{12}$ ,  $V_{20}$ ), conforme ilustrado na Figura 14 (b). Esse processo de divisão repete-se novamente para os 4 triângulos encontrados até que sejam definidos 16 novos triângulos, como mostra a Figura 14 (c), para o triângulo dividido com grau de profundidade 5 (ISO/ASTM 52915:2016).

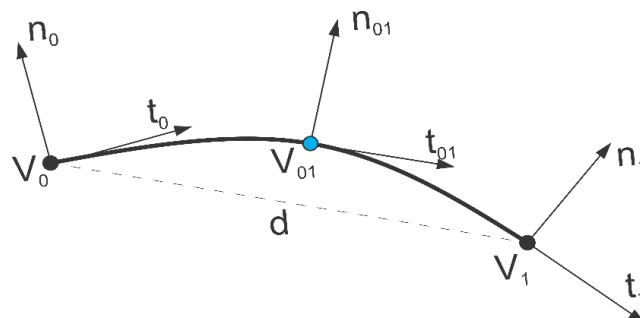
Figura 14 – (a) triângulo sem divisão, (b) triângulo dividido com grau 1, (c) triângulo dividido com grau 2.



Fonte: Adaptado de ISO/ASTM 52915:2016.

Para encontrar o ponto de divisão  $V_{01}$  da curva de Hermite, ilustrado na Figura 15, é necessário definir as tangentes nos vértices  $V_0$  e  $V_1$ . As tangentes podem: estar definidas explicitamente no arquivo, ser calculadas através da normal  $n$  quando definida no arquivo, ou calculadas a partir do produto vetorial das duas arestas do triângulo que conectam o vértice; respectivamente nessa ordem de prioridade.

Figura 15 – Ponto  $V_{01}$  na curva cúbica de Hermite.



Fonte: Adaptado de ISO/ASTM 52915:2016.

As equações utilizadas para calcular as tangentes dos vértices representados na Figura 15 são as equações (2.2) e (2.3), definidas pela norma ISO 52915:2016, e para calcular o ponto médio da curva utiliza-se o interpolador de Hermite apresentado na equação (2.4) com o parâmetro interpolador  $s$  igual a 0.5 (Mortenson et al., 1999).

$$t_0 = |d| \frac{-(n_0 \times d) \times n_0}{\|(n_0 \times d) \times n_0\|} \quad (2.2)$$

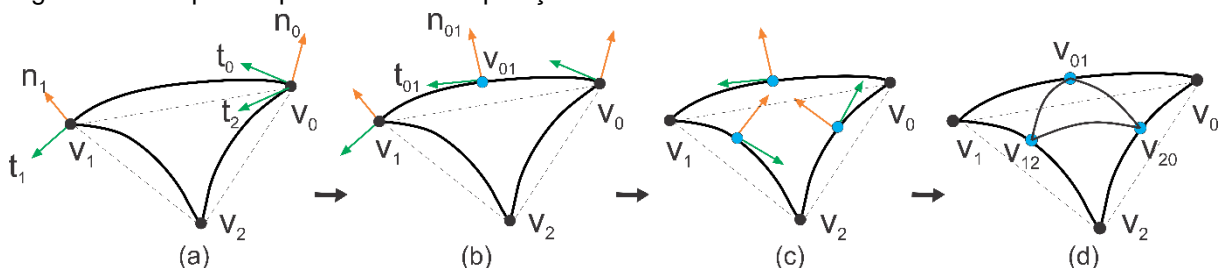
$$t(s) = (6s^2 - 6s)V_0 + (3s^3 - 4s + 1)t_0 + (-6s^2 + 6s)V_1 + (3s^2 - 2s)t_1 \quad (2.3)$$

$$h(s) = (2s^3 - 3s^2 + 1)V_0 + (s^3 - 2s^2 + s)t_0 + (-2s^3 + 3s^2)V_1 + (s^3 - s^2)t_1 \quad (2.4)$$

Para dividir o triângulo curvo, deve-se encontrar o ponto de divisão de cada uma das 3 arestas do triângulo, ilustrado por  $V_{01}$  na Figura 15. As 9 etapas apresentadas a seguir explicam detalhadamente como cada cálculo deve ser executado para encontrar todos esses pontos (ISO/ASTM 52915:2016):

1. Verificar se a normal  $n_0$  do vértice  $V_0$  está expressa no arquivo, caso contrário calcular  $n_0$  pelo produto vetorial das duas tangentes ( $t_0$ ,  $t_2$ ) que se encontram em  $v_0$ , se não houver tangente, utiliza-se a aresta linear entre os pontos do triângulo, etapa ilustrada na Figura 16 (a);
2. Repetir a etapa 1 para calcular  $n_1$  no vértice  $V_1$ ;
3. Se a tangente  $t_0$  do vértice  $V_0$  não for explicitamente especificada no arquivo, deve-se calcular  $t_0$  através da equação (2.3);
4. Repetir a etapa 3 para calcular  $t_1$  no vértice  $V_1$ ;
5. Calcular o ponto central da curva,  $v_{01}$  utilizando interpolação cúbica de Hermite, dada pela equação (2.4) com  $s = 0,5$ , etapa ilustrada na Figura 16 (b);
6. Computar a tangente  $t_{01}$  no ponto central da curva, utilizando a derivada da interpolação de Hermite, expressa na equação (2.3) com  $s = 0,5$ ;
7. O padrão AMF não especifica, mas a normal  $n_{01}$  também deve ser calculada.
8. Repetir as etapas 1 a 7 para os vértices  $V_1$  e  $V_2$ .
9. Por fim, os 3 vértices calculados (Figura 16 (c)) são conectados, formando 4 novos triângulos curvos, ilustrado na Figura 16 (d).

Figura 16 – Etapas do processo de interpolação das curvas de Hermite no AMF



Fonte: Produção do autor.

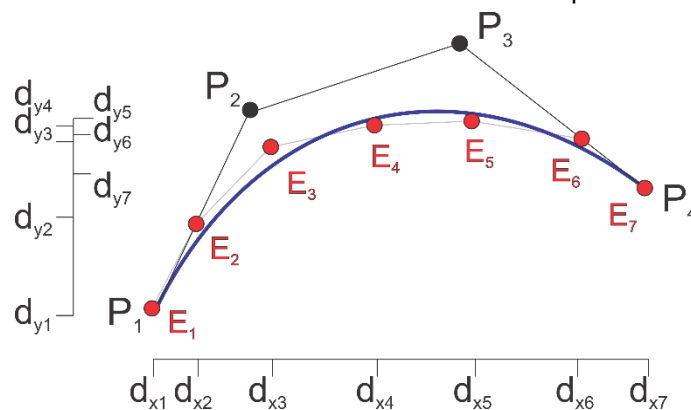
### 2.4.4.3 Ajuste dos pontos de intersecção

Nas definições de PP propostas por Marsan et al. (1997) e Livesu et al. (2017) não há etapas destinadas ao ajuste dos pontos de intersecção, que são obtidos durante o fatiamento do modelo. Entretanto algumas abordagens propõem o ajuste desses pontos de intersecção logo após a etapa de fatiamento (KOC et al., 2000; MA et al., 2004; ZHAO et al., 2009).

O ajuste dos pontos de intersecção visa melhorar a fidelidade do trajeto de cada camada com o modelo 3D. Para isso é necessário escolher um método que modifique o trajeto original por outro que represente melhor a intersecção do modelo. Os métodos encontrados na literatura utilizam cálculos matemáticos como o ajuste de curvas com mínimos quadrados (ZHAO et al., 2009), diferença no ângulo da tangente (KOC et al., 2000) e aproximações com redes neurais (PANDEY et al., 2003), para substituir sequências de pontos de intersecção por curvas.

Nesta seção será apresentado o método de ajuste de curva cúbica de Bézier minimizado pelo método dos mínimos quadrados, aplicado neste trabalho para encontrar os pontos de controle em curvas cúbicas de Bézier. A Figura 17 ilustra a curva com seus pontos de controle  $P$ , aproximada pelo método dos mínimos quadrados a partir da sequência de pontos de intersecção  $E$ .

Figura 17 – Ajuste de curva cúbica de Bézier em uma amostra de sequência de pontos



Fonte: Produção do autor.

A curva cúbica de Bézier, escrita na equação (2.5), foi selecionada para o ajuste de curva por ser a única curva paramétrica encontrada sendo representada em linguagem de CN, detalhada na seção (2.4.6). A curva cúbica de Bézier é definida pelo interpolador  $B(t_i)$ , onde  $t$  é o parâmetro interpolador da curva e  $P$  são os pontos de controle (Mortenson et al., 1999).

$$B(t) = (1 - t)^3 P_0 + 3t(1 - t)^2 P_2 + 3t^2(1 - t) P_3 + P_4 t^3 \quad (2.5)$$

Após definida a Curva de Bézier, é necessário também definir os parâmetros de entrada para o método dos mínimos quadrados. Esses serão utilizados para definir o sistema de equações a partir do cálculo das derivadas parciais.

O objetivo do ajuste de curva é reduzir a diferença entre a curva e os pontos de intersecção. O valor dessa diferença representa a função de erro da aproximação. Essa função é dada pelo somatório das distâncias entre os pontos e a curva ajustada, dado pela equação (2.6) onde  $y$  é a coordenada  $X$  ou  $Y$  do ponto de intersecção e  $B$  a função da curva paramétrica (PEREIRA, 2014):

$$S(P_n) = \sum_{i=1}^n [y - B(t_i)]^2 \quad (2.6)$$

Para associar os pontos de intersecção com os valores da curva paramétrica, ou seja, encontrar o valor de  $t_i$  da equação (2.6), utiliza-se a distância  $d$ , da Figura 17, entre os pontos de intersecção dividida pelo somatório de distâncias, esse cálculo deve ser realizado para as coordenada  $X$  e  $Y$  utilizando a equação (2.7) a seguir:

$$t_i = \frac{|P_{x(i)} - d_{x(i-1)}|}{\sum_{i=2}^n |P_{x(i)} - P_{x(i-1)}|} \quad (2.7)$$

O ponto crítico mínimo da função de erro da aproximação (Equação (2.6)) é quando a derivada dessa função é igual a zero, isso é garantido pelo somatório dos quadrados ser positivo na derivada de segunda ordem (ALMEIDA et al., 2015). Portanto para encontrar o menor erro deve-se encontrar a derivada de cada ponto de controle da equação (2.6), e com a união destas gerar um sistema de equações. Desta forma, o ajuste pelo método dos mínimos quadrados é dado pela resolução desse sistema expresso pelas equações (2.8) e (2.9), onde  $n$  é a quantidade de pontos de intersecção:

$$(1 - t) = C \quad (2.8)$$

$$\left( \begin{array}{l} \sum_{i=1}^n C^6 P_0 + \sum_{i=1}^n 3tC^5 P_1 + \sum_{i=1}^n 3t^2 C^4 P_2 + \sum_{i=1}^n C^3 t^3 P_3 = \sum_{i=1}^n C^3 y \\ \sum_{i=1}^n 3C^5 t P_0 + \sum_{i=1}^n 9t^2 C^4 P_1 + \sum_{i=1}^n 9t^3 C^3 P_2 + \sum_{i=1}^n 3C^2 t^4 P_3 = \sum_{i=1}^n 3C^2 t y \\ \sum_{i=1}^n 3C^4 t^2 P_0 + \sum_{i=1}^n 9t^3 C^3 P_1 + \sum_{i=1}^n 9t^4 C^2 P_2 + \sum_{i=1}^n 3Ct^5 P_3 = \sum_{i=1}^n 3Ct^2 y \\ \sum_{i=1}^n t^3 C^3 P_0 + \sum_{i=1}^n 3t^4 C^2 P_1 + \sum_{i=1}^n 3t^5 C P_2 + \sum_{i=1}^n t^6 P_3 = \sum_{i=1}^n t^3 y \end{array} \right) \quad (2.9)$$

### 2.4.5 Simulador

A simulação da AM é essencial para confirmação do usuário sobre as etapas anteriores do PP, é onde os trajetos obtidos no fatiamento das camadas são ilustrados graficamente e apresentados ao usuário. Os *software* que realizam a simulação normalmente estão interligados ao fatiador e a escolha dos parâmetros de impressão.

Neste trabalho foram analisados os *software* de impressoras 3D existentes no mercado, e classificados por: método de intersecção, representações aceitas, se realiza transformações no modelo (escalonamento, rotação e translação), linguagem de programação, código-aberto entre outras características, complementadas no Apêndice A.

Da análise anterior foram selecionados os *software* com interface gráfica (GUI) e com código-aberto disponível para *download* na internet. Esses foram então analisados novamente com relação ao arquivo AMF e apresentados na Tabela 2.

Tabela 2 – Tabela de fatiadores

Fabricante	Fatiador	Linguagem de programação	Importa AMF	Visual 3D
Bread	Nick-Parker	Java	×	×
Alexrj	Slic3r	Perl, C++	✓	✓
Ultimaker	CuraEngine	C++/Python	×	✓
Bmander	Skeinforge	Python	×	×
<b>MatterHackers</b>	<b>MatterControl</b>	<b>c#</b>	✓	✓
FormLabs	Hackathon-Slicer	JavaScript	×	✓

Fonte: Produção do autor.

A Tabela 2 destaca o fabricante MatterHackers, que apresenta o *software* MatterControl com melhor compatibilidade para alterações, devido a organização estrutural do seu código; possui também uma interface de usuário gráfica (GUI) que possibilita a visualização do modelo 3D, a simulação do fatiamento e aceita o padrão AMF para importação de modelos 3D.

Sua estrutura é dividida em módulos, todos programados em linguagem *c#*, o que é uma vantagem em relação a outros *software*, que normalmente separam o módulo do fatiador em uma linguagem diferente da GUI.

O *software* é considerado “*All-in-one*” dos *software* para impressoras 3D, pois serve para: ajustar modelos 3D, calibrar a impressora, fatiar o modelo, gerar o código em linguagem de CN e também possibilita controlar a impressora em tempo de execução.

### 2.4.6 Gerador de código em linguagem de programação de CN

A fase de geração de código em linguagem de CN é considerada a última fase do PP (MARSAN et al., 1997). É nesta fase que os trajetos de contorno obtidos pelo fatiamento do modelo serão traduzidos em comandos de linguagem de CN, como por exemplo a linguagem definida na norma ISO 6983-1 (G-Code). As informações desses trajetos podem ser importadas através de arquivos ou obtidas internamente pelo fatiador.

O mesmo *software* responsável pela simulação pode também gerar o arquivo com a linguagem de CN, como é realizado pelo fatiador MatterControl. Desta forma, é importante saber qual a linguagem de controle será utilizada para a tradução, para isso primeiramente deve-se identificar qual o *firmware* utilizado. Existem diversos *firmware* disponíveis no mercado, no entanto apenas um *software* de código aberto foi encontrado capaz de interpretar curvas paramétricas nos comandos de linguagem de CN.

Em pesquisa tradicional realizada buscando por *firmware* de código-aberto disponíveis para *download*, foram encontrados 14 *firmware* ainda em utilização (Sprinter, Marlin, Sailfish, Repetier-Firmware, etc) e 9 outros que estão fora de utilização (Makerbot etc). Dos *firmware* que ainda estavam em utilização, apenas o Marlin possui comando para curvas paramétricas do tipo Bézier cúbica (RepRap, 2017).

Marlin é um *firmware* de impressora 3D que trabalha com microcontroladores em placas Arduino, com desenvolvimento inspirado em dois projetos de *firmware* existentes, o Sprinter<sup>1</sup> e o Grbl<sup>2</sup>. O *firmware* é flexível com aproximadamente 140 parâmetros que devem ser configurados para trabalhar com diferentes tipos de *hardware* de impressoras 3D (STANCIULESCU et al., 2015).

O Quadro 3 apresenta comandos utilizados na extrusão com Marlin. Entre eles *G0* e *G1* com parâmetros *E* relativo ao material, *F* para informar a velocidade do extrusor e *XYZ* para indicar as coordenadas do ponto final do segmento de reta. No comando *G2* o parâmetro *E* é encontrado novamente, o diferencial desse está na curva que pode ser informada pelo raio ou pelo offset. E por fim, o comando *G5* que utiliza os pontos de controle e o ponto final da curva para definir a curva paramétrica de Bézier.

Quadro 3 – Comandos de movimentos do *firmware* Marlin.

linear	G0 G1	[E<pos>]	F<rate>	X<pos>	Y<pos>	Z<pos>
circular	G2	[E<pos>]	I<offset>	J<offset>	P<radius>	X<pos> Y<pos> Z<pos>
Bézier cubic	G5	[E<pos>]	I<pos>	J<pos>	P<pos>	Q<pos> X<pos> Y<pos>

Fonte: Gcode Marlin Documentation [<http://marlinfw.org/docs/gcode/G005.html>].

<sup>1</sup> <https://github.com/kliment/Sprinter>, disponível em 01/06/2018.

<sup>2</sup> <https://github.com/grbl/grbl/wik>, disponível em 01/06/2018.

## 2.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo foram apresentados os conceitos fundamentais para o desenvolvimento da proposta. Inicialmente foram vistos métodos de fabricação na AM, em destaque o FDM, e logo após métricas de avaliação foram detalhadas para possibilitar avaliar melhorias obtidas com a aplicação da proposta. Os padrões de arquivos também estão detalhados neste capítulo, com foco na representação de malhas de triângulos curvos e o padrão AMF.

Uma visão geral sobre o processo de fatiamento em AM foi apresentada neste capítulo, e apresentado também o método de ajuste de curva por mínimos quadrados utilizado por alguns fatiadores. Por fim, foram analisados os *firmware* de código aberto, disponíveis para utilização neste trabalho.

### 3 TRABALHOS RELACIONADOS

Neste capítulo são identificados, classificados e discutidos os principais trabalhos que visam melhorar a fidelidade e reduzir o custo das peças produzidas em AM, particularmente relacionados as novas representações de superfícies curvas e aos fatiadores de sólidos. Estão relacionados também, trabalhos que aplicam ajuste de curvas ou outras aproximações que possam ser aplicadas no fatiamento de arquivos AMF, com o objetivo de reduzir o erro de fidelidade das peças produzidas em AM.

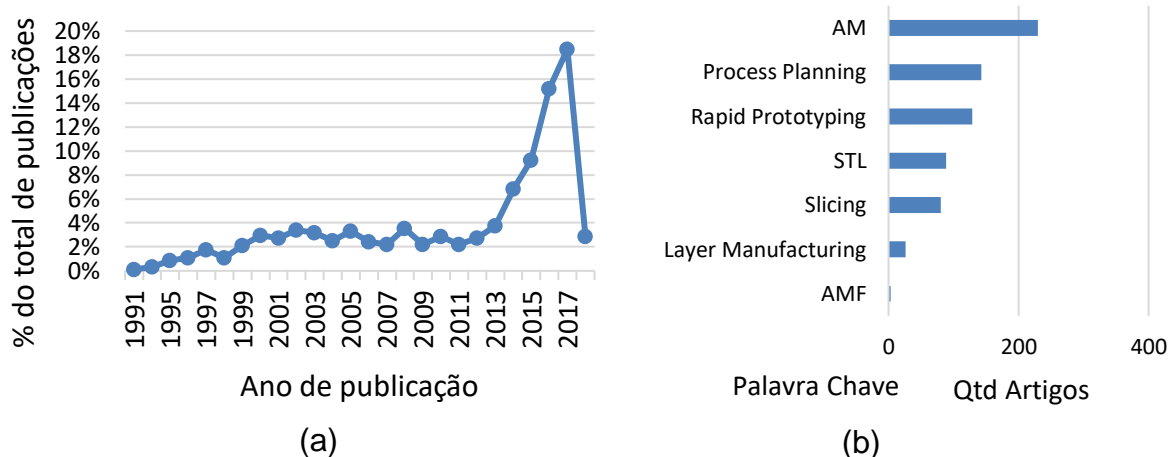
#### 3.1 PESQUISA BIBLIOGRÁFICA

Primeiramente, em 2018, foi realizada uma busca para definir as palavras chaves: Additive, Layered, Rapid, Manufacturing, Prototyping, Process Planning, Slicer e Fitting Curve, que serviram para a criação da frase de busca utilizada na Pesquisa Bibliográfica Sistemática (PBS). A seguinte frase de busca foi escrita seguindo o padrão de formação de Wazlawick et al. (2008) e aplicada na busca por tópicos das bases *Scopus* e *Web of Science*, por possuírem os mais completos mecanismos de busca (BUCHINGER et al., 2014):

```
((slic* OR ("process plan*") OR ("fit* curve*")) AND
("additiv* manufactur*") OR ("layer* manufactur*") OR ("rapid* propotyp*"))
```

Resultando em 529 (*Scopus*) e mais 366 artigos (*Web of Science*), estes foram filtrados para remover duplicatas, artigos *short* e artigos não publicados. Do resultado também foram criados os gráficos (Gráfico 2) que mostram o aumento do número de publicações (a) e a quantidade de artigos encontrados por cada palavra chave (b).

Gráfico 2 – (a) gráfico de nº de publicações por ano, (b) quantidade de artigos por palavra chave.



Fonte: Produção do autor.

Durante a fase de seleção subjetiva da PBS foram selecionados os artigos com foco no contexto da pesquisa pelos critérios de inclusão. Incluídos conforme: (1) apresentar algoritmos de fatiamento em AM, (2) abordar novas representações de modelos 3D, (3) fazer avaliação de erros de representação (4) apresentar aproximações de curvas no fatiamento. E excluídos conforme: (1) fazer fatiamento em multi direção, (2) gerar comandos de máquina para manufatura por usinagem, (3) propor novas topologias de suportes, (4) tratar apenas erro de funcionalidade em AM.

Os artigos a seguir foram escolhidos dentre os 157 artigos obtidos pela PBS, por terem como objetivo a melhoria da fidelidade da peça em AM através de aprimoramentos no processo de fatiamento ou representação do modelo 3D.

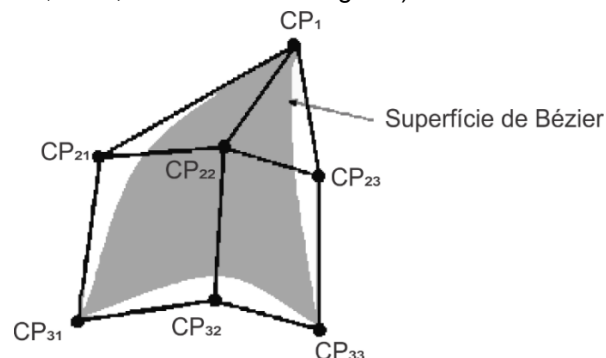
## 3.2 NOVAS REPRESENTAÇÕES

Esta seção apresenta trabalhos que propuseram novas estruturas de representações para os modelos 3D em AM, com foco nas malhas de triângulos curvos.

### 3.2.1 Malha de Bézier

Allavarapu et al. (2013) propuseram um padrão de arquivo utilizando como superfícies de representação, a malha triangular de Bézier derivada da malha biquadrada de Bézier pela união dos pontos de controle superiores ( $CP_1$  apresentado na figura 18), de modo a obter um formato triangular curvo. O fatiamento da malha triangular de Bézier é realizado por aproximações de pontos discretos gerados na malha, o que implica em erro de GD&T nos trajetos de contorno. Apesar de citar o padrão de arquivo AMF no trabalho, os pesquisadores realizam os testes comparativos apenas entre a malha triangular de Bézier com o padrão STL, obtendo assim resultados satisfatórios em geral.

Figura 18 – Pontos de controle ( $CP$ ) utilizados na representação com superfície biquadrada de Bézier em forma triangular (no caso,  $CP_{11}$ ,  $CP_{12}$  e  $CP_{13}$  são iguais).

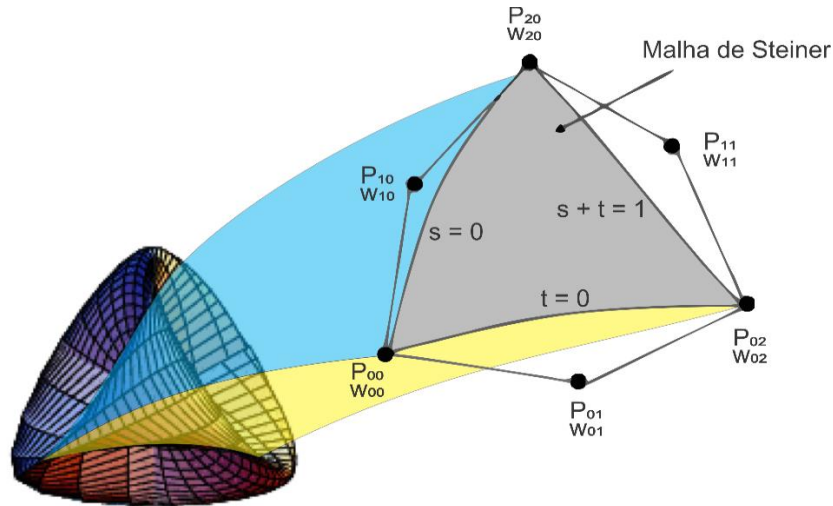


Fonte: Adaptado de Allavarapu (2013).

### 3.2.2 Malha de Steiner

Paul e Anand (2015) propuseram a criação de um novo padrão de arquivo para AM utilizando a malha de triângulos de Steiner (SEDERBERG e ANDERSON, 1985, apud PAUL e ANAND 2015), definido pelo triângulo delimitado na superfície biquadrada de Steiner, como mostrado na Figura 19.

Figura 19 – Triângulo de Steiner, onde  $P_{ij}$  são os pontos de controle e  $w_{ij}$  são os pesos associados.



Fonte: Adaptado de Paul e Anand (2015).

Diferente dos fatiamentos em malhas de Bézier e Hermite, que fazem uso de aproximações. Na malha de Steiner, a curva de intersecção da superfície com a camada é calculada de forma analítica e computacionalmente tratável, a partir da Equação (3.1), onde  $P_{ij}$  são os pontos de controle e  $w_{ij}$  são os pesos associados aos pontos, visualizada na Equação (3.2) de forma simplificada pelas constantes  $C$ . Devido a isto, o fatiamento da malha de Steiner consegue obter resultados com erros de GD&T menores.

No entanto os testes realizados na pesquisa foram apenas em superfícies isoladas, não sendo aplicado em representações de sólidos fechados, o que dificulta a comprovação dos resultados para os métodos de AM.

$$S(s, t) = \frac{\sum_{i+j \leq 2} \left( \frac{2!}{i!j!(2-i-j)!} \right) s^i t^j (1-s-t)^{2-i-j} w_{ij} P_{ij}}{\sum_{i+j \leq 2} \left( \frac{2!}{i!j!(2-i-j)!} \right) s^i t^j (1-s-t)^{2-i-j} w_{ij}} = h \quad (3.1)$$

$$= C_1 s^2 + C_2 t^2 + C_3 st + C_4 s + C_5 t + C_6 = 0 \quad (3.2)$$

### 3.3 FATIADORES COMPUTACIONALMENTE EFICIENTES

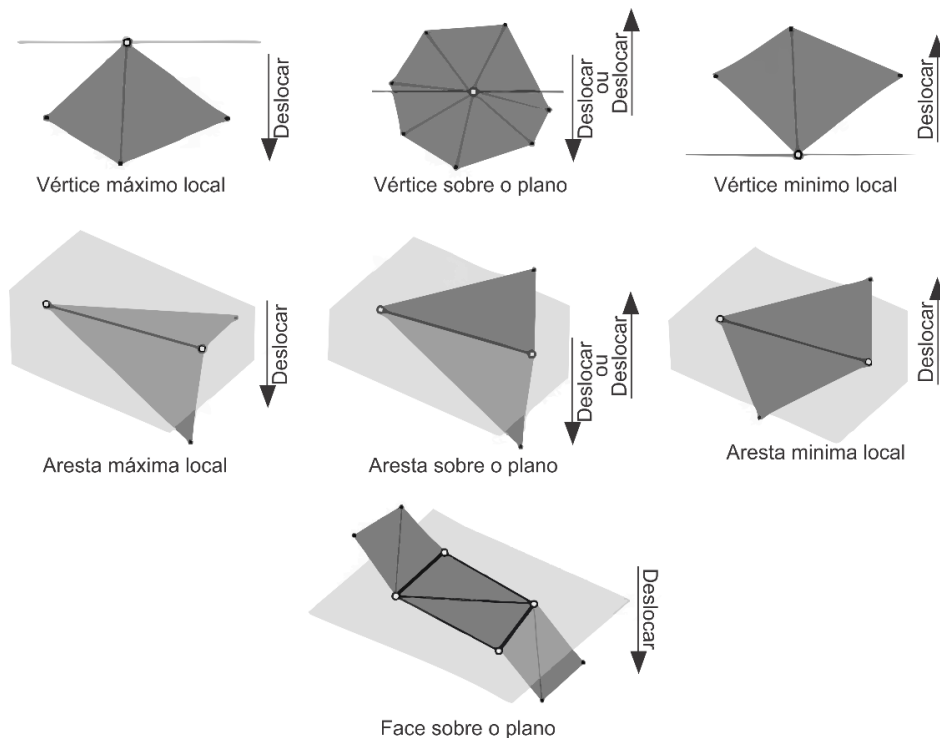
São apresentados nesta seção trabalhos que propuseram aprimoramentos em algoritmos de fatiamento de poliedros, aplicáveis ao padrão AMF.

#### 3.3.1 Fatiador baseado no deslocamento dos vértices

Kim et al. (2010) apresentaram um algoritmo de fatiamento de sólidos poliédricos (STL) com pré-processamento para evitar intersecções que causem irregularidades na etapa de união dos pontos de contorno das camadas. Segundo os autores, as irregularidades que ocorrem devido ao posicionamento dos vértices sobre o plano, são mais complexas de se resolver ao se deslocar o vértice de posição. Portanto o algoritmo realiza o deslocamento de todos os vértices que estiverem sobre os planos das camadas durante a etapa de pré-processamento, o que exige a leitura de todos os triângulos do arquivo antes do início do processo de intersecções.com objetivo melhorar a eficiência do fatiado, reduzindo possivelmente o tempo de processamento.

Na etapa de pré-processamento do fatiador proposto, são identificados 7 singularidades diferentes ilustrados na Figura 20, que ocorrem quando os vértices, arestas ou face do triângulo se encontram no mesmo nível do plano de intersecção. Para cada singularidade é indicado deslocar o vértice em uma fração mínima (sendo menor que o espaço entre camadas) na direção do eixo z e sentido indicado na Figura 20.

Figura 20 – Cenários de possíveis singularidade para o fatiador.



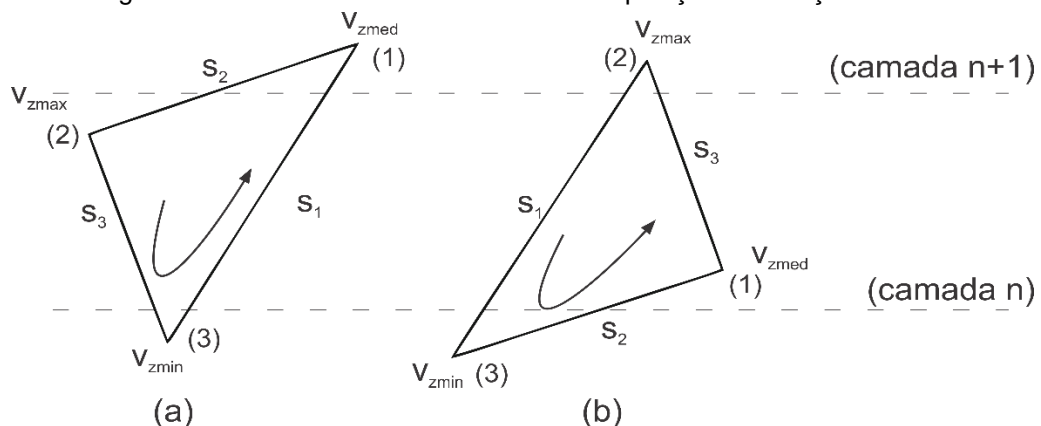
Fonte: Adaptado de Kim et al. (2010).

### 3.3.2 Fatiador baseado na ordem de leitura do arquivo

Zhang e Joshi (2015) propuseram um algoritmo eficiente de fatiamento para arquivos STL, que reduziu o tempo de processamento computacional em 82% em relação ao fatiador de Kim et al. (2010). O algoritmo executa a intersecção dos triângulos da malha pela ordem descrita no arquivo, sem necessidade de realizar uma ordenação prévia dos triângulos. Isto implica em interseccionar individualmente cada triângulo com os planos que o transpassam, o que pode causar pontos de intersecção duplicados.

Para evitar essas intersecções duplicadas em uma mesma aresta, o algoritmo escolhe e identifica a aresta que deve ser interseccionada como *Forward Edge* (FE). Para identificar as *FEs*, inicialmente analisam-se os vértices do triângulo com relação ao eixo Z ( $V_{zmin}$ ,  $V_{zmed}$  e  $V_{zmax}$ ) e os índices atribuídos a eles que definem a orientação do triângulo, em seguida as arestas são classificadas em  $s_1$ ,  $s_2$  ou  $s_3$ , onde  $s_1 = \{V_{zmin}, V_{zmax}\}$ ,  $s_2 = \{V_{zmin}, V_{zmed}\}$  e  $s_3 = \{V_{zmed}, V_{zmax}\}$ . Após definir a orientação do triângulo e os vértices  $s_1$ ,  $s_2$  e  $s_3$ , é então realizado o julgamento das arestas *FEs*. A aresta  $s_1$  será *FE* quando o vertice  $z_{max}$  preceder  $z_{min}$ , conforme ilustra a Figura 21 (b), caso contrário as arestas  $s_2$  ou  $s_3$  é que devem ser classificadas como *FE* dependendo da posição da camada interseccionada, conforme ilustra a Figura 21 (a).

Figura 21 – Triângulo com arestas e vértices de acordo com posição em relação ao eixo z e índice STL.



Fonte: Produção do autor.

Quando obtidas as intersecções devem ser armazenadas em uma lista encadeada que ajudará no processo de geração dos caminhos realizado pelo algoritmo Eficiente de Construção de Contorno (ECC) (ZHANG e JOSHI, 2015). O algoritmo ECC analisa e interliga as intersecções obtidas das arestas *FE* para formar os contornos de cada camada.

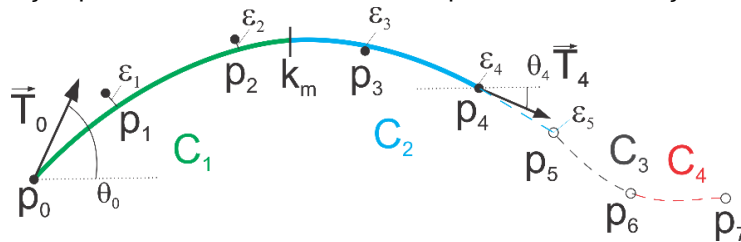
### 3.4 APROXIMAÇÕES DE CURVAS EM FATIADORES

Nesta seção são apresentadas aproximações de curvas utilizadas em fatiadores para aumentar a fidelidade das peças em AM, que possam servir de referência para criação do fatiador AMF-AC.

#### 3.4.1 Aproximação com *Biacr*: *Max-Fit Biacr Method*

Koc et al. (2000) propuseram o método *Max-Fit Biacr* para suavizar o contorno das camadas geradas pelos fatiadores de arquivos STL. O método de aproximação *Max-Fit Biacr* utiliza a concatenação de curvas Biacs (Figura 22), que são formadas pela união de dois arcos ( $C_1, C_2$ ), com continuidade geométrica de fator um<sup>1</sup> (G1) no ponto de união  $K_m$ . Estas curvas podem assumir diferentes formatos (*C-shaped*, *S-shaped* ou *Fillet-shaped*) quando alteradas as direções e proporções das curvaturas de seus arcos.

Figura 22 – Aproximação pelo método *Max-Fit Biacr* dos pontos de intersecção interrompida em  $p_4$

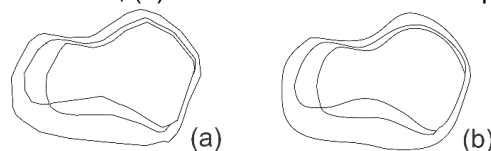


Fonte: Produção do autor.

O processo de construção da curva Biacr representado na Figura 22 inicia com a sequência de 3 pontos ( $p_0, p_1$  e  $p_2$ ) obtidos do fatiador, posteriormente são acrescentados novos pontos ( $p_3$  até  $p_5$ ) e recalculados o ponto médio da curva ( $K_m$ ) e a tangente ( $\vec{T}$ ) do ponto adicionado. Com isso testa-se o valor de  $\theta$ , que representa o ângulo entre a tangente  $T$  e o próximo ponto a ser acrescentado, se  $\theta$  for superior ao valor máximo, o algoritmo então inicia uma nova curva Biacr com os próximos 3 pontos de intersecção da sequência.

Os autores comparam o fatiador linear de arquivos STL (Figura 23 (a)) e o fatiador Max-Fit Biacr, (Figura 23 (b)) e apresentam os resultados, com reduções no número de segmentos do contorno em 78% e no Erro de Corda em 95% (Seção 2.2.2.1), para modelos com superfícies curvas.

Figura 23 – (a) borda de contorno linear, (b) bordas de contorno com aproximação de curvas.



Fonte: Adaptado de Koc et al. (2000).

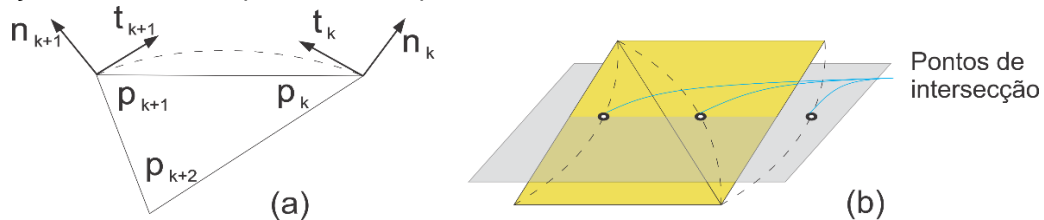
<sup>1</sup> Direção das tangentes das curvas são iguais no ponto de junção, conhecido como G1 (FOLEY et al., 1990).

### 3.4.2 Aproximação com *B-Splines*

Outro trabalho que tem como objetivo melhorar a fidelidade da peça através de aproximações foi desenvolvido por (ZHAO et al., 2009). Neste os autores também propuseram a métrica de Erro Real (apresentada na seção 2.2.2) que, segundo eles, serve como o melhor comparador para a fidelidade da peça quando utilizado em AM.

O processo de fatiamento desenvolvido inicia com a criação das normais nos vértices, uma vez que os arquivos STL só possuem normais das faces. Os autores utilizam o método de cálculo da normal baseado no trabalho de Grit e Charles (1998 apud ZHAO et al., 2009, p. 32) para relacionar as normais das faces adjacentes ao vértice criando também uma normal para o vértice. Em seguida são encontradas as tangentes  $t$  que são utilizadas para calcular os pontos de controle que definem as curvas de Bézier nas arestas, ilustradas pelas linhas pontilhadas nas Figura 24 (a) e (b). Estas, por sua vez são utilizadas para calcular as intersecções, ao invés das linhas dos triângulos planos do arquivo STL.

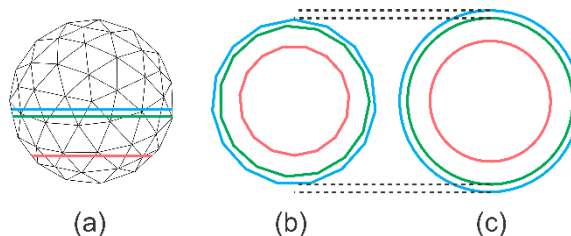
Figura 24 – (a) normais  $n$ , tangentes  $t$  e pontos  $p$  que definem a curva de Bézier, (b) pontos de intersecção entre a curva aproximada e o plano.



Fonte: Produção do autor.

Por fim, os pontos de intersecção são utilizados para criar a curva *B-Spline*. Os autores minimizam o erro da aproximação pelo método dos mínimos quadrados, que reduz a soma das distâncias entre os pontos de intersecção com a curva interpoladora. As Figura 25 (a) e (b) ilustram exemplos de cortes obtidos em fatiadores padrões de STL, sem aproximações de curvas, e compara com o fatiamento (Figura 25 (c)) obtido pelo fatiador de ZHAO et al., 2009. Esse obteve um resultado de 50% a menos no Erro Real devido a mudança de representação do contorno por curvas *B-Spline*.

Figura 25 – (a) secções no poliedro, (b) fatiamento sem ajuste de curva, (c) fatiador com ajuste de curva e intersecção de arestas curvas.



Fonte: Adaptado de Zhao et al. (2009).

### 3.5 DISCUSSÃO DOS TRABALHOS

Através da revisão dos trabalhos relacionados, pôde-se perceber algumas similaridades nas técnicas utilizadas com objetivo de melhorar a fidelidade da peça, entre elas a utilização de malhas de triângulos curvos para a representação do modelo 3D, e o ajuste de curvas nos pontos de intersecção do contorno das camadas.

#### 3.5.1 Sobre as novas representações

As estruturas de representações propostas por Paul e Anand (2015) e Allavarapu et al. (2013) se assemelham ao padrão AMF pela utilização de triângulos curvos. Porém, possuem algoritmos de fatiamento bem distintos, sendo que o padrão AMF e a malha triangular de Bézier necessitam de aproximações para definir o contorno das camadas. Já o trabalho de Paul e Anand (2015) com a malha de triângulos de Steiner tem o fatiamento mais eficiente no quesito fidelidade, pelo fato de calcular a curva de intersecção de forma analítica. O problema da representação de Paul e Anand (2015) é que ainda não foi testada em superfícies complexas e nem em modelos sólidos, dificultando sua aplicação em AM.

#### 3.5.2 Sobre a suavização dos contornos

O trabalho de Zhao et al. (2009) tem características similares aos trabalhos de representações triangulares curvas, devido a geração de curvas nas arestas dos triângulos do arquivo STL para melhorar sua fidelidade com o modelo 3D. Os resultados obtidos pelos autores evidenciam as vantagens de se utilizar representações em malhas triangulares curvas.

Também é possível observar que os algoritmos de fatiamento apresentados não distinguem com precisão em qual ponto do contorno a curva ajustada deve alternar sua concavidade, e por isso utilizam de técnicas para identificar esses locais. Como por exemplo a técnica apresentada por Koc et al. (2000) que calcula o ângulo da tangente sempre que aproxima a curva Biarc aos pontos de intersecção. Já Zhao et al. (2009) no entanto utilizam curvas *B-Spline* com os pontos de controle ajustados pela minimização da soma do erro total do contorno da camada, o que não garante com precisão a localização da inversão de concavidade da curva.

Sendo assim, se considerar que o ajuste de curva em malhas de triângulos curvos, representados no padrão AMF, ocorra apenas entre os pontos obtidos de um único triângulo curvo. Para esse não há necessidade de utilizar técnicas para localizar a alteração de direção da concavidade, pois a direção da concavidade do triângulo curvo, em AMF, se mantém em toda superfície do triangulo curvo.

### 3.5.3 Sobre a eficiência do fatiador

A busca realizada para obtenção dos trabalhos relacionados também foi direcionada aos artigos com abordagens de aprimoramentos no fatiador, que pudessem ser aplicados no desenvolvimento do fatiador AMF-AC. Os trabalhos de Kim et al. (2010) e Zhang e Joshi (2015) contribuem com ideias que podem reduzir o tempo computacional necessário para o fatiamento.

Zhang e Joshi (2015) demonstram em seus resultados que a escolha de análise pela ordem descrita no arquivo STL traz reduções de tempo significativas no processamento do fatiador. Desta forma torna-se desnecessária a ordenação prévia dos triângulos planares, o que torna a aplicação dessa modificação interessante ao padrão AMF.

Os aprimoramentos propostos no trabalho de Kim et al. (2010) também aparentam ser aplicáveis ao padrão AMF, pois sua estrutura também possibilita que os vértices sejam definidos sobre as camadas do fatiamento. Porém, nos casos de arestas ou faces coincidentes com a camada, o algoritmo de fatiamento deve tratar de forma diferenciada as arestas considerando que no padrão AMF elas podem ser representadas por curvas.

## 3.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

O capítulo descreveu os trabalhos relacionados ao processo de fabricação por AM, com foco nas estruturas de representação e no aprimoramento de fatiadores. Foram apresentados 2 trabalhos sobre malhas triangulares curvas, 2 trabalhos de aprimoramento no algoritmo de fatiamento e por fim mais 2 trabalhos de algoritmos de fatiamento com aproximações de curvas.

Com base nos trabalhos apresentados pode-se perceber que a utilização de representação com triângulos curvos, é uma abordagem que traz benefícios de fidelidade ao processo de AM. Sendo que o padrão AMF já é um padrão aprovado e em utilização no mercado, diferente dos apresentados que ainda estão em desenvolvimento.

Verifica-se também os aprimoramentos realizados nos trabalhos relacionados que contribuem com ideias para o fatiador AMF-AC. Reduzindo o processamento através do fatiamento sem ordenação prévia dos triângulos, e aprimorando a fidelidade da peça com métodos de aproximações de curvas.



## 4 DESENVOLVIMENTO DO FATIADOR AMF-AC

Este capítulo apresenta em detalhes cada etapa desenvolvida do fatiador AMF-AC. O desenvolvimento desse fatiador tem como intuito de reduzir o erro de fidelidade e suavizar as curvaturas das peças de forma eficiente. Para tanto, são introduzidos os requisitos do fatiador, bem como sua arquitetura e algoritmo de fatiamento.

Para o funcionamento correto do fatiador AMF-AC, são obrigatórios os seguintes pré-requisitos:

- Serão aceitos apenas modelos sólidos fechados no padrão AMF;
- Triângulos curvos não podem ter inflexão na curvatura, como exige a norma ISO 52915:2016;
- O parâmetro 'altura de camada' será informado pelo usuário e uniforme;
- Será reaproveitado o código do *software* MatterControl para realizar a implementação do AMF-AC.

Os requisitos que definem as funcionalidades do fatiador AMF-AC, do ponto de vista do usuário, são os requisitos funcionais (RF) do *software*, entre eles estão:

- RF1. Fatiar modelos 3D descritos em AMF;
- RF2. Tornar opcional o ajuste de curva e a intersecção de arestas curvas;
- RF3. Possibilitar a visualização das camadas geradas;
- RF4. Fornecer um arquivo de linguagem de CN para FDM;
- RF5. Calcular a redução do erro de fidelidade.

Como requisitos não funcionais (RNF) foram definidos:

- RNF1. Leitura dos triângulos curvos nos arquivos AMF;
- RNF2. Subdividir os triângulos curvos;
- RNF3. Fatiar o modelo em camadas uniformes;
- RNF4. Ajustar o contorno em curvas paramétricas;
- RNF5. Avaliar parâmetros para grau de profundidade na subdivisão;
- RNF6. Gerar o código em linguagem de CN com curvas paramétricas.

O padrão AMF será utilizado neste trabalho, devido a sua compatibilidade com o STL, suas definições de texturas e ao suporte de superfícies curvas, que fazem com que o padrão AMF seja o principal candidato para substituir o padrão STL na AM (WANG, Q. et al., 2016).

Para aproveitar de forma eficiente o potencial do AMF, este trabalho aplica o ajuste de curvas nos pontos de intersecção obtidos a partir do fatiamento. Para isso, antes desse fatiamento ocorrer são realizadas três subdivisões (grau de profundidade 21) nos triângulos curvos, essa quantidade de subdivisões foi definida a partir dos testes preliminares. Conforme relatado nos trabalhos de Zhao et al. (2009) e Koc et al. (2000), a utilização do ajuste de curvas no fatiamento pode trazer melhorias para a fidelidade da peça. Dessa forma, deseja-se obter a fidelidade representada apenas as subdivisões mais profundas, onde é necessário um maior processamento computacional.

O ajuste de curva foi implementado de forma opcional a fim de proporcionar também o fatiamento sem ajuste de curvas, para que se possa ter um parâmetro de comparação.

Como o fatiador AMF-AC foi desenvolvido utilizando alguns módulos existentes do *software* MatterControl, é necessário compreender a arquitetura e o funcionamento dos seus principais módulos. Para isso, as localizações de onde foram implementados os novos métodos do fatiador AMF-AC são apresentadas no fluxograma da Figura 27.

#### 4.1 ARQUITETURA DO SOFTWARE MATTERCONTROL

Entre os *software* de fatiamento para impressoras 3D encontrados na literatura e no mercado, o *software* MatterControl foi selecionado para a realização deste trabalho por apresentar um estágio maduro de desenvolvimento, possuir compatibilidades de importação para o padrão AMF, por ser gratuito e de código aberto e estar implementado utilizando somente uma linguagem de programação.

O MatterControl é totalmente escrito na linguagem orientada a objetos Csharp (c#), e sua arquitetura é dividida em cinco módulos: gráfico, fatiamento, testes, plugins e ferramentas. Dentre esses módulos, os mais relevantes para este trabalho são o módulo gráfico e o módulo de fatiamento, pois são responsáveis por exibir o modelo 3D e realizar as intersecções do modelo com as camadas, respectivamente.

Na Tabela 3 são apresentados os módulos com suas principais classes; em destaque as classes envolvidas no processo de fatiamento do modelo onde são encontrados a maioria dos métodos alterados durante a criação do fatiador AMF-AC. Duas novas classes foram adicionadas ao modulo de fatiamento, são elas: a classe MMQ (Mínimos Quadrados) e Newton-Raphson, onde se encontram os métodos responsáveis pelos cálculos matemáticos utilizados durante o fatiamento implementado no AMF-AC.

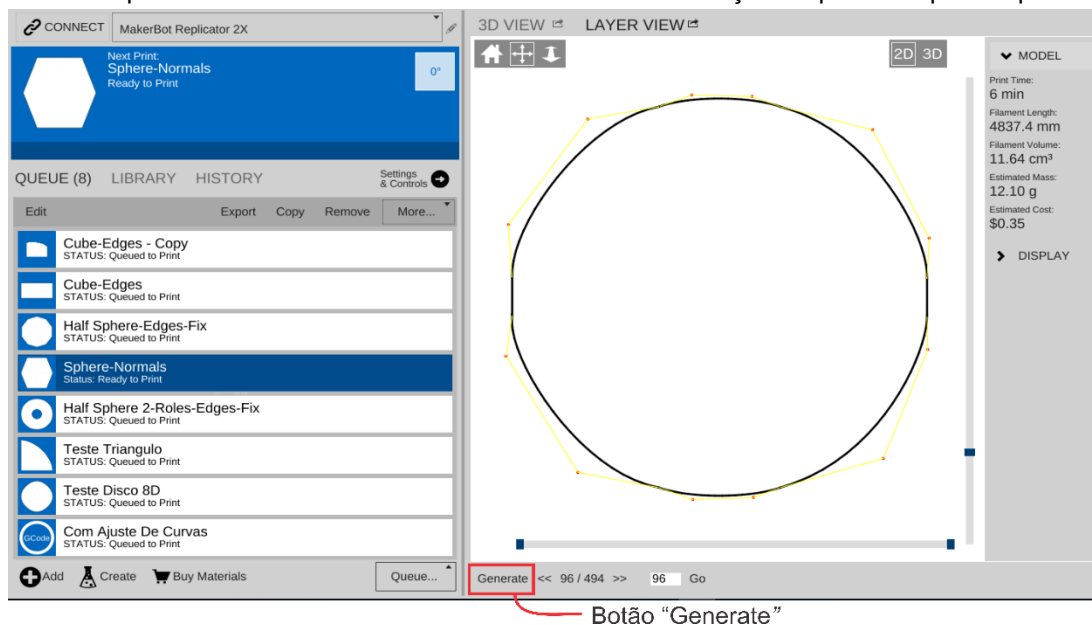
Tabela 3 – Módulos e suas respectivas classes principais do *software* MatterControl.

Módulos				
Gráfico	Fatiamento	Plugins	Ferramentas	Testes
Classes				
GuiWidget	<b>fffProcessor</b>	BrailleBuilder	PrinterEmulator	Agg.tests
GCodeRender	<b>SimpleMesh</b>	TextCreator	CsharpSqlite	MatterSlice.tests
GcodeFile	<b>ExtruderLayer</b>			
MeshViewer	<b>Slicer</b>			
...	<b>GcodeExport</b>			
	<b>MMQ</b>			
	<b>Newton-Raphson</b>			
	...			

Fonte: Produção do autor.

O módulo gráfico, além de exibir o modelo 3D, é também responsável pela importação dos arquivos de modelo 3D para o *software* podendo estar no padrão STL ou AMF. Apesar de possibilitar a importação do AMF, o *software* processa esse da mesma forma que o padrão STL, desconsiderando os triângulos curvos. Portanto no desenvolvimento do AMF-AC foi necessário implementar, na classe SimpleMesh, o método de importação que considera as informações dos triângulos curvos.

Nesse módulo também ocorre a simulação do código G-Code, que pode ser gerado ou importado pelo *software*. Isso possibilita sua apresentação gráfica de forma que o usuário possa visualizar os caminhos a serem percorridos durante a impressão da peça. Para o fatiador AMF-AC foi desenvolvido um novo método para apresentar as curvas paramétricas utilizadas no G-Code, a interface do *software* com essa visualização é apresentado na Figura 26.

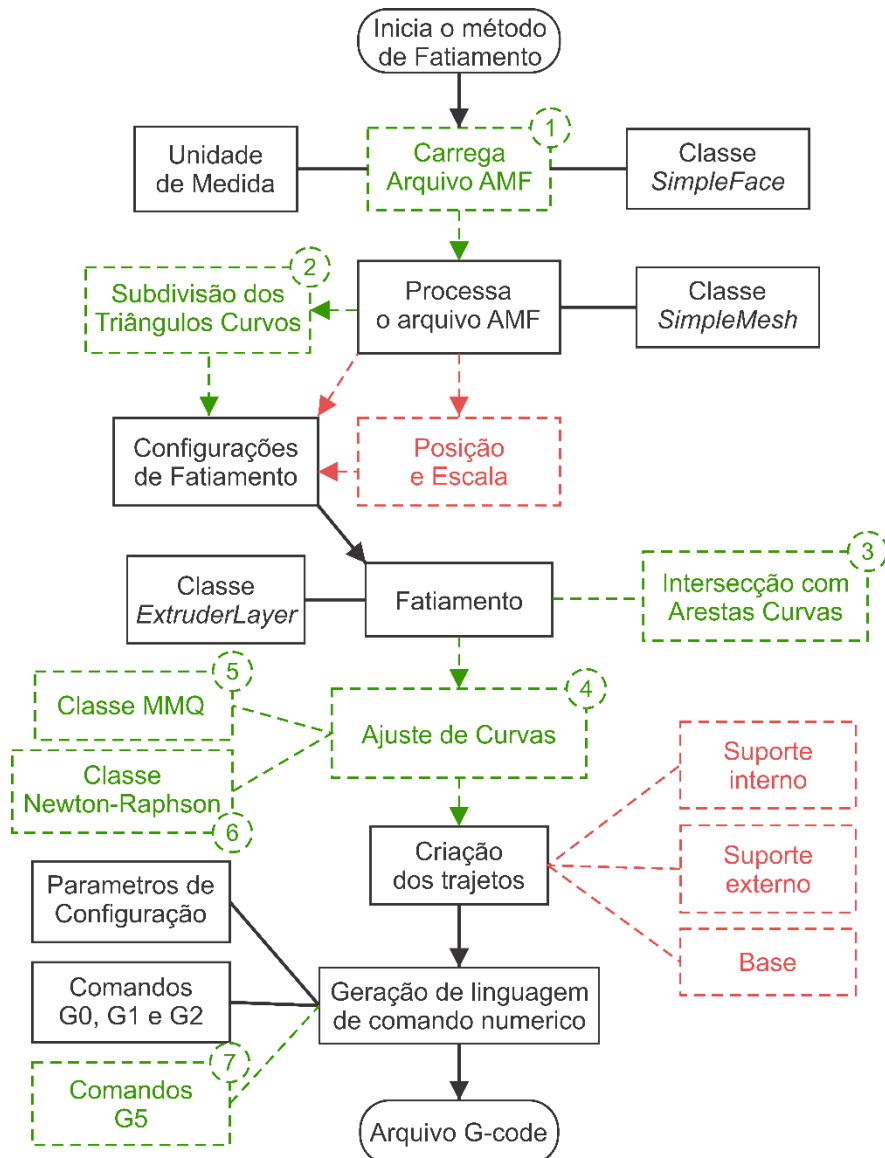
Figura 26 – Captura de tela do *software* MatterControl com simulação do percurso para impressão 3D

Fonte: Produção do autor.

No módulo de fatiamento são implementadas as classes responsáveis pela lógica de fatiamento, portanto é nesse módulo que estão presentes as alterações mais significativas do AMF-AC. Nele são realizadas as etapas de intersecção com as camadas, união dos vértices do contorno, o ajuste de curvas e a geração de código em linguagem de CN.

O AMF-AC foi desenvolvido com a adição de novos métodos e classes implementadas dentro do *software* MatterControl, apresentado, no fluxograma na Figura 27. Em destaque, na cor verde, as etapas do processo de fatiamento nas quais essas novas classes e métodos foram introduzidas. Também na Figura 27, estão identificadas, na cor vermelha, algumas funcionalidades que foram removidas do fatiador AMF-AC, com objetivo de manter o foco na proposta deste trabalho.

Figura 27 – Fluxograma do fatiamento realizado pelo MatterControl.



Fonte: Produção do autor.

## 4.2 ESTRUTURA DO FATIADOR AMF-AC

As modificações e adições realizadas no *software* MatterControl para o desenvolvimento do fatiador AMF-AC foram organizadas de acordo com o PP em AM. Portanto, esse trabalho inicia com a implementação dos métodos essenciais de pré-processamento no arquivo AMF para o fatiamento, descritos na seção 4.3, entre eles a leitura do AMF e as subdivisões dos triângulos curvos.

De acordo com o PP, operações de orientação e estruturas de suporte são as etapas seguintes a serem consideradas após o pré-processamento, entretanto para manter o foco no objetivo deste trabalho, essas etapas não foram contempladas neste desenvolvimento.

Desta forma, após o pré-processamento são desenvolvidas as etapas relativas ao fatiamento com o ajuste de curva e a geração de código em linguagem de CN para impressão 3D. Portanto, este trabalho tem seu desenvolvimento estruturado em cinco etapas, são elas:

- a) Pré-processamento da malha triangular curva;
- b) Intersecção com as camadas;
- c) União dos pontos de intersecção;
- d) Ajuste de curva;
- e) Geração de código em linguagem de CN.

As subseções seguintes apresentam o detalhamento de cada uma das etapas de desenvolvimento e seus algoritmos associado ao fatiador AMF-AC.

## 4.3 PRÉ-PROCESSAMENTO DA MALHA TRIANGULAR CURVA

O AMF-AC realiza o processo de fatiamento de um determinado modelo em AMF através da busca pelos pontos de intersecção entre a malha triangular curva e as camadas; para isso é necessário ler e aplicar antes algumas operações de pré-processamento na malha triangular curva:

- a) Inicialmente deve-se realizar a leitura das informações (vértices, normais e tangentes) que descrevem a malha de triângulos curvos.
- b) Em seguida é feita a subdivisão da malha triangular curva do arquivo AMF para aumentar a precisão da superfície, o que causa um aumento do número de triângulos necessários para representar o modelo;
- c) Após realizar a subdivisão, os vértices que estejam sobre as camadas são deslocados para evitar casos singulares durante a intersecção.

### 4.3.1 Leitura do arquivo AMF

Neste trabalho foi desenvolvido o método de leitura para arquivos AMF capaz de extrair as informações que definem as arestas curvas com tangentes e normais descritas no arquivo, diferentemente do método de importação existente no *software* MatterControl, que trata todos os triângulos como triângulos planos, utilizando somente as informações dos vértices.

Para cada triângulo curvo extraído do arquivo AMF é acrescentada uma instância da classe *SimpleFace* a uma lista interna do fatiador AMF-AC. As variáveis que compõem esta classe são apresentadas no Quadro 4; essa também armazena um identificador (*idRoot*) que é atribuído a cada um dos triângulos iniciais. Esse identificador é utilizado novamente pelo fatiador na identificação das sequências de pontos de intersecção originadas de um mesmo triângulo curvo inicial, durante a etapa de ajuste de curva do fatiador AMF-AC.

Quadro 4 – Atributos da classe SimpleFace.

```
public class SimpleFace
{
    public IntPoint[] Vertices { get; set; }
    public Vector3[] Normals { get; set; }
    public Vector3[,] Edges { get; set; }
    public int idRoot { get; set; }
    ...
}
```

Fonte: Produção do autor.

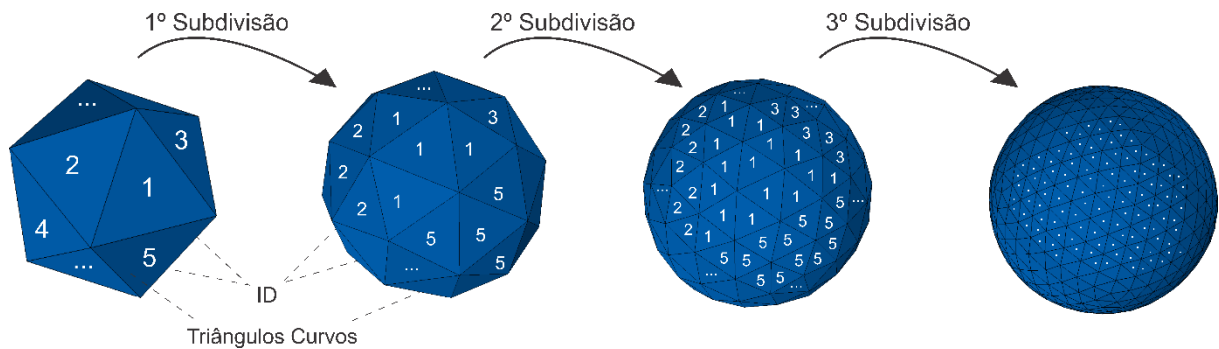
### 4.3.2 Subdivisão da malha triangular curva

Após extrair as informações de definições dos triângulos curvos do arquivo AMF foi então possível a implementação do método de subdivisão dos triângulos curvos. De acordo com a especificação do padrão AMF, o fatiamento deve ocorrer após a subdivisão dos triângulos curvos. Este trabalho também aborda a subdivisão antes do fatiamento, uma vez que a subdivisão está relacionada a quantidade de intersecções necessárias durante a etapa de fatiamento.

O padrão AMF também especifica que o processo de subdivisão dos triângulos curvos deve ser realizado através de sucessivas subdivisões, feitas nas arestas curvas dos triângulos, definidas pelas curvas cúbicas de Hermite (seção 2.4.4.2.3). De acordo com a norma devem ser realizadas duas subdivisões antes do fatiamento, no entanto este número pode variar de acordo com a necessidade de detalhamento. Conforme previsto na ISO 52915:2016, para calcular essas divisões são necessárias: a equação (2.3) de cálculo das tangentes dos vértices, e o interpolador de Hermite apresentado na equação (2.4) para calcular o ponto médio de cada curva.

Para manter a informação de qual triângulo curvo foi originado, o identificador *idRoot* é passado como parâmetro para os triângulos gerados na etapa de subdivisão. Estes, por sua vez, passam o mesmo identificador para os triângulos gerados em suas subdivisões seguintes. A Figura 28 ilustra uma esfera descrita em AMF com 20 triângulos curvos e como os identificadores se propagam a cada subdivisão realizada nessa esfera.

Figura 28 – Propagação do *idRoot* na subdivisão do poliedro com 20 triângulos curvos.

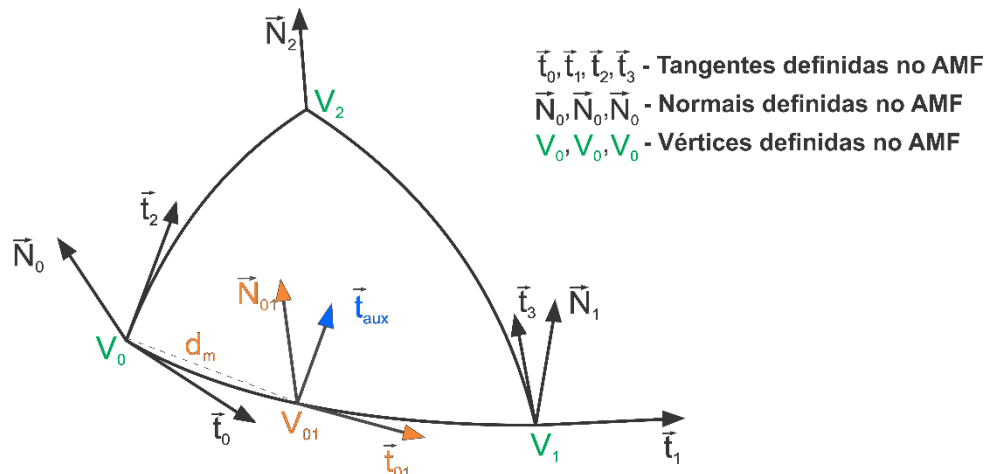


Fonte: Produção do autor.

Durante o processo de subdivisão também é necessária a geração de vetores normais nos pontos médios das curvas de Hermite. Como não há instruções na ISO 52915:2016 sobre como deve ser calculada a normal no ponto médio da aresta, ilustrada na Figura 29, foram então propostos diferentes métodos para este cálculo.

Em geral os métodos propostos objetivam encontrar a tangente  $\vec{t}_{aux}$  que representa a influência dos vetores  $\vec{t}_3$  e  $\vec{t}_2$  sobre o ponto médio  $V_{01}$  da curva  $\overline{V_0V_1}$ . Obtido o valor de  $\vec{t}_{aux}$ , é calculado o produto vetorial entre  $\vec{t}_{aux}$  e  $\vec{t}_{01}$  para encontrar a normal  $N_{01}$ .

Figura 29 – Normais, tangentes e vértices utilizados no cálculo na subdivisão do triângulo curvo.

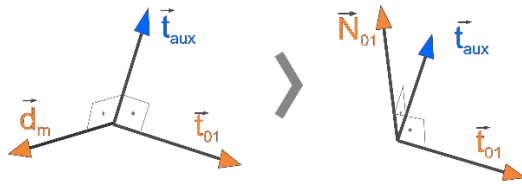


Fonte: Produção do autor.

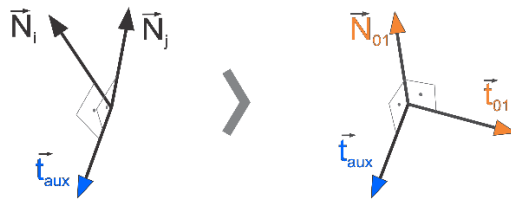
A partir de testes preliminares foram selecionados três métodos para gerar a normal no ponto médio da curva de Hermite. O primeiro método utilizando o produto vetorial (Figura 30 (a)) entre a distância  $\vec{d}_m$  e a tangente  $\vec{t}_{01}$  no ponto  $V_{01}$ , o segundo com o produto vetorial entre as normais dos vértices adjacentes (Figura 30 (b)) e o terceiro com a soma entre as normais dos vértices adjacentes (Figura 30 (c)).

Figura 30 – Sequências de operações realizadas para calcular a normal nos três métodos propostos.

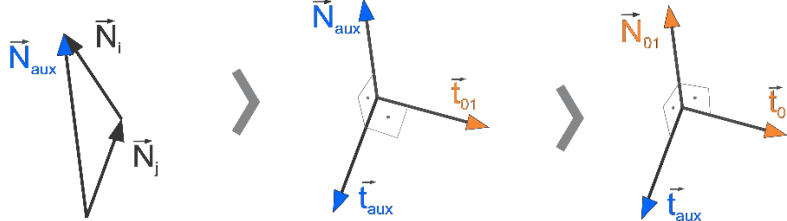
a) Produto vetorial: distância x tangente



b) Produto vetorial: normal x normal



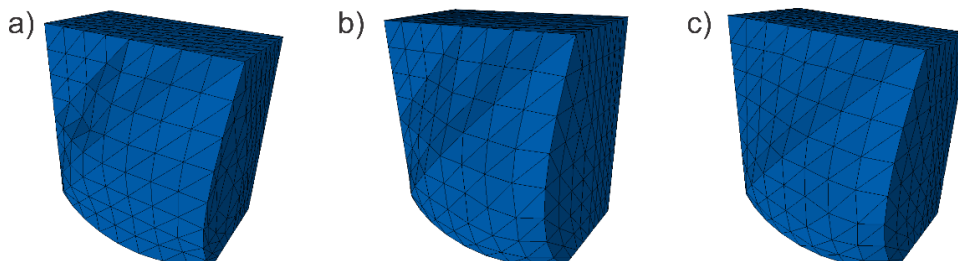
c) Soma: normal x normal



Fonte: Produção do autor.

Sendo assim, no desenvolvimento do fatiador AMF-AC, será utilizado para calcular a normal a soma entre as normais dos vértices adjacentes. Método esse, escolhido após análise visual das simulações, por apresentar melhores resultados na continuidade da superfície curva, ilustrado na Figura 31(c), em comparação com os métodos que utilizam produto vetorial entre as normais  $N_0$  e  $N_1$  (Figura 31(b)), e entre a distância  $\vec{d}_m$  e a tangente  $\vec{t}_{01}$  (Figura 31(a)).

Figura 31 – Subdivisões realizadas utilizando os métodos (a), (b) e (c) para calcular a normal.



Fonte: Produção do autor.

Com isso, pretende-se também reduzir o número de cálculos necessários, se comparado a outras propostas de cálculo de normal em triângulos curvos conforme os trabalhos de Hagen e Pottmann (1989), Loop et al. (2009) e Yang; Zheng (2013).

### 4.3.3 Deslocamentos de vértices

Ainda no pré-processamento, é feita uma revisão dos vértices para identificar vértices que estejam sobre os planos de corte das camadas, com altura definida pelo usuário. Os vértices que estiverem sobre o plano são deslocados de acordo com as direções definidas no trabalho de Kim et al. (2010), seção 3.3.1, em  $10^{-4}$  milímetros de modo que esse deslocamento não represente alterações perceptíveis a peça.

Ao finalizar a etapa de pré-processamento, o fatiador AMF-AC iniciará a etapa de intersecção das camadas com as arestas dos triângulos.

## 4.4 INTERSECÇÃO COM AS CAMADAS

Diferentemente do trabalho de Zhao et al. (2009), que propuseram uma aproximação da normal nos vértices a fim de criar curvas nas aresta dos triângulos, em arquivos AMF não é necessário fazer esta aproximação, visto que as arestas já estão definidas por curvas de Hermite. Portanto, nesta etapa de desenvolvimento do fatiador AMF-AC ocorrem as intersecções entre as curvas de Hermite a as camadas equidistantes (fatiamento uniforme), com a altura definida pelo usuário. A intersecção da aresta curva é aplicada de forma opcional para possibilitar a sua desativação quando for necessário realizar o fatiamento de acordo com padrão do AMF.

Para encontrar os pontos de intersecção da curva com uma camada, é necessário igualar a coordenada  $z$  da curva de Hermite, equação (2.4), com a altura da camada representada na variável  $h$  da equação (4.1).

$$f(s) = (2s^3 - 3s^2 + 1)V_{0z} + (s^3 - 2s^2 + s)t_{0z} + (3s^2 - 2s^3)V_{1z} + (s^3 - s^2)t_{1z} - h \quad (4.1)$$

A partir da equação (4.1) são calculadas as possíveis raízes, ou seja, os possíveis valores de  $s$ . Essas são limitadas em no máximo duas, por conta do pré-requisito que limita a inflexão nas arestas do triângulo. Cada raiz  $s$  encontrada é um ponto da curva de Hermite intersectada pelo plano.

O algoritmo utilizado nesta proposta para encontrar as raízes da equação (4.1) é o Newton-Raphson, expresso na equação (4.2), onde a cada iteração  $n$  o resultado da função converge para a raiz da equação  $f(s)$ . Considerando que a curva não possui inflexões, como exige a norma ISO 52915:2016, a implementação deste algoritmo se

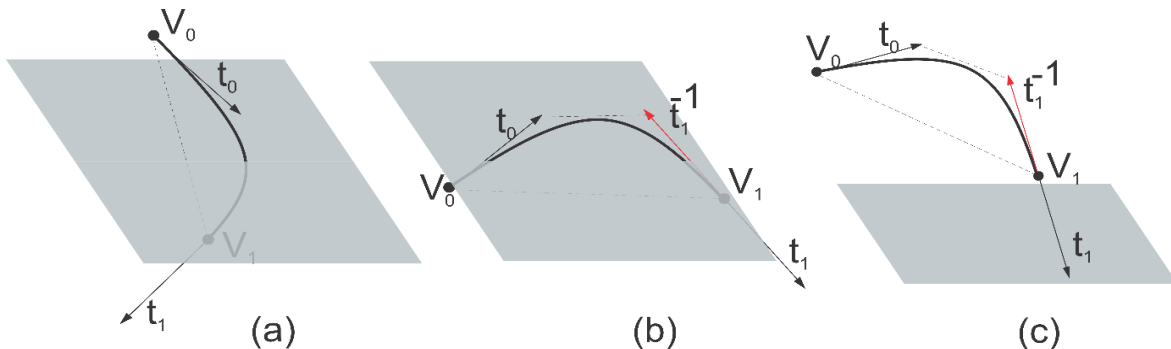
torna mais eficiente ao utilizar as extremidades da curva como pontos de partida para convergência do algoritmo. Devido a essa convergência é necessário definir um erro máximo para que o algoritmo termine a execução, erro que foi definido em  $10^{-4}$  durante os testes preliminares. Essa aproximação de raízes implica que os pontos de intersecção sejam aproximados também.

$$f(s)_{(n+1)} = f(s)_{(n)} - \left( \frac{f(s)}{\partial f(s)} \right) \quad (4.2)$$

Para reduzir o tempo de processamento das intersecções no fatiador AMF-AC, três hipóteses que definem o número de intersecções (raízes) são consideradas:

- Um plano entre os vértices da curva, que gera um único ponto de intersecção (Figura 32 (a));
- Um plano entre as tangentes  $\vec{t}_0$  e  $\vec{t}_1$  inversa, geram dois pontos de intersecção (Figura 32 (b));
- Nenhuma das alternativas anteriores, não intersecta a curva (Figura 32 (c)).

Figura 32 – (a) um ponto de intersecção, (b) dois pontos de intersecção, (c) sem intersecção.



Fonte: Produção do autor.

#### 4.5 UNIÃO DOS PONTOS DE INTERSECÇÃO

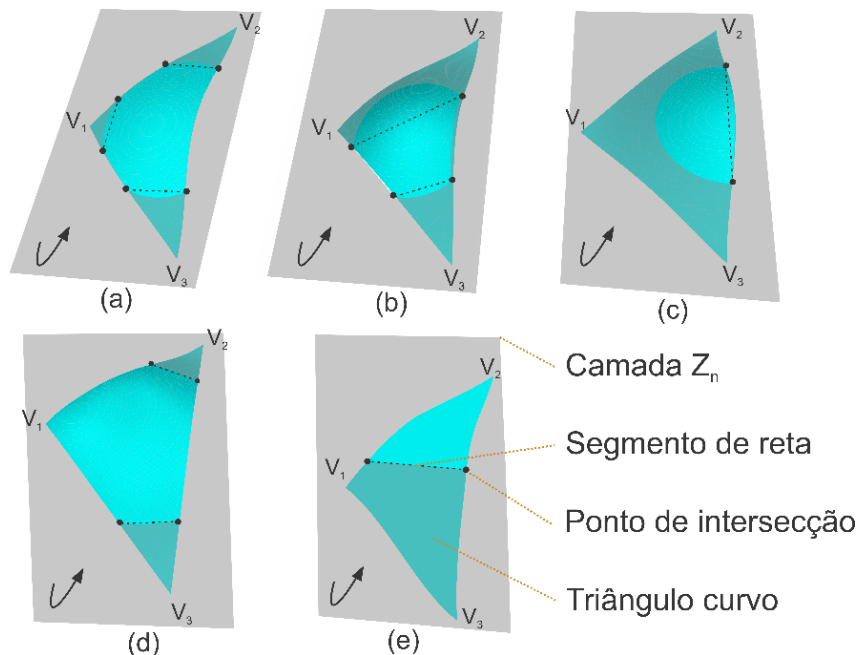
Após encontrar os pontos de intersecção aproximados, é necessário unir os pontos encontrados em segmentos de retas para cada um dos triângulos. Em triângulos planos a intersecção com a camada gera apenas um segmento de reta, no entanto, ao utilizar os triângulos curvos do AMF a quantidade de segmentos gerados pode variar em até cinco casos. Esses estão relacionados a quantidade de intersecções e as arestas dos triângulos curvos, em uma camada por vez:

- Triângulo com duas intersecções nas três arestas (Figura 33 (a));
- Triângulo com duas intersecções em duas arestas (Figura 33 (b));

- (c) Triângulo com duas intersecções em uma aresta (Figura 33 (c));
- (d) Triângulo com uma intersecção em duas arestas e duas intersecções na terceira aresta (Figura 33 (d)).
- (e) Triângulo com uma intersecção em duas arestas (Figura 33 (e));

Para os casos (a), (b), (d) e (e), sempre que houver intersecções em mais de uma aresta do triângulo, essas intersecções devem ser conectadas por segmentos de reta na ordem anti-horária e entre arestas diferentes. Para o caso (c), quando apenas uma aresta do triângulo está interseccionada em dois pontos, deve-se conectá-los com um segmento de reta.

Figura 33 – Casos resultantes da intersecção com triângulos curvos: (a) três arestas com duas intersecções cada, (b) duas arestas com duas intersecções cada, (c) uma aresta com duas intersecções, (d) uma aresta com duas intersecções e duas arestas com uma intersecção, (e) caso padrão.



Fonte: Produção do autor.

Tendo os segmentos de reta obtidos a partir das intersecções entre os triângulos curvos com as camadas, o próximo passo será unir os segmentos de reta em polígonos que representam os trajetos das camadas. Esse procedimento foi realizado para cada camada interseccionada, utilizando o algoritmo KNN (seção 2.4.4.2.1), com o objetivo de encontrar os pontos de intersecção que estejam em posições diferentes, mas que representem o mesmo vértice do polígono.

Essa duplicação acontece devido ao erro de aproximação do algoritmo de Newton-Raphson utilizado para encontrar as intersecções. Desta forma, quando o algoritmo encontrar dois pontos muito próximos ele deverá substituir o ponto inicial do segmento de reta pelo ponto final do segmento adjacente.

#### 4.6 AJUSTE DE CURVA

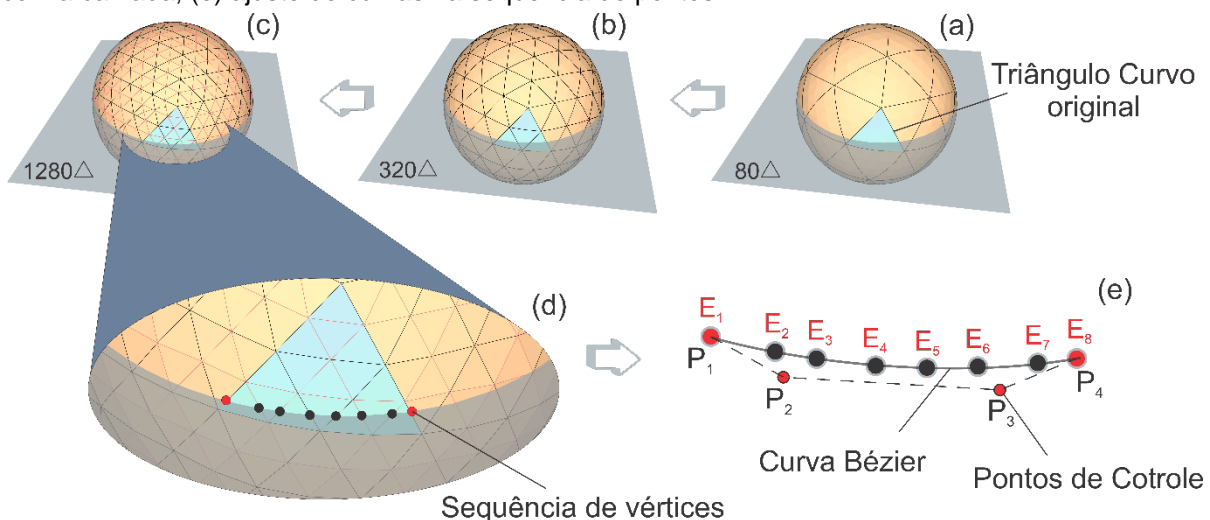
A etapa de ajuste de curva do fatiador AMF-AC é onde os pontos de intersecção são aproximados em curvas paramétricas. O ajuste de curva tem como principal objetivo suavizar as curvaturas horizontais, reduzindo o erro de fidelidade entre a peça e o modelo; outra vantagem do ajuste de curva é que este possibilita a utilização de curvas paramétricas, neste trabalho são usadas as curvas cúbicas de Bézier, para a definição dos contornos de cada camada.

Como durante a etapa de geração de código em linguagem de CN, será utilizado como base para os comandos de máquina G-Code o *firmware* Marlin. Foi escolhido trabalhar com curvas de Bézier pelo fato deste *firmware* interpretar apenas esse tipo de curva paramétrica.

As sequências de pontos de intersecção que serão aproximados com o ajuste de curva, são definidas pelas intersecções geradas a partir dos triângulos resultantes das subdivisões com as camadas. Cada sequência de pontos deve ter no mínimo três pontos para que possa ser ajustada em uma curva, caso contrário os dois pontos são utilizados para definir um segmento de reta.

A Figura 34 ilustra as etapas do processo de definição das sequências; nele uma sequência nova é criada para cada agrupamento de pontos de intersecção obtida dos triângulos com o mesmo idRoot (da classe apresentada no Quadro 4), desta forma cada sequência terá apenas pontos de intersecções que originaram das subdivisões de um mesmo triângulo curvo inicialmente descrito no arquivo AMF. Em seguida as sequências de pontos de intersecção (Figura 34 (d)) são ajustadas em curvas cúbicas de Bézier (Figura 34 (e)) e substituídas no contorno da camada.

Figura 34 – (a) AMF original, (b) primeira subdivisão, (c) segunda subdivisão, (d) pontos de intersecção com a camada, (e) ajuste de curvas na sequência de pontos.



Com base nos trabalhos encontrados, foi escolhido para ajustar as curvas o método dos mínimos quadrados. O método constitui principalmente em minimizar a distância entre os pontos de intersecção até um determinado polinômio, alterando a posição dos pontos de controle desse polinômio; no caso deste trabalho a amostra é a sequência de pontos de intersecção, e o polinômio é a curva cúbica de Bézier. Essa minimização é obtida a partir da derivada do somatório das distâncias igualada a zero, como explica a seção 2.4.4.3.

No entanto, nesta proposta são usados como variáveis apenas os pontos de controle  $P_2$  e  $P_3$ , ilustrados na Figura 34, para ajustar a curva, pois os pontos de controle nas extremidades são definidos pelos pontos de entrada e saída da sequência de pontos de intersecção, e são colocados como constantes na equação, e com isso se reduz a complexidade do sistema linear.

O sistema de equações gerado a partir das derivações pode ser solucionado através da Eliminação de Gauss, ou de forma iterativa pelo método de Newton-Raphson (ITOH e OHNO, 1993; SHAO e ZHOU, 1996). Para a solução do sistema, este trabalho utiliza apenas o método de Newton-Raphson, a fim de reaproveitar a lógica implementada na etapa de intersecção das curvas com as camadas (seção 4.4).

Para aplicar o método de Newton-Raphson, primeiramente se reescreve a equação (2.9) na equação (4.3) para melhor compreensão, onde  $P_n$  são os pontos de controle e  $C$  é dado por  $(1 - t)$ , cada somatório da equação é calculado pela soma das distâncias entre os pontos de intersecção expressa na variável  $t$ .

$$\begin{aligned}
 f_1(P) &= \sum_{i=1}^n C^6 P_0 + \sum_{i=1}^n 3tC^5 P_1 + \sum_{i=1}^n 3t^2 C^4 P_2 + \sum_{i=1}^n C^3 t^3 P_3 - \sum_{i=1}^n C^3 = 0 \\
 f_2(P) &= \sum_{i=1}^n 3C^5 t P_0 + \sum_{i=1}^n 9t^2 C^4 P_1 + \sum_{i=1}^n 9t^3 C^3 P_2 + \sum_{i=1}^n 3C^2 t^4 P_3 - \sum_{i=1}^n 3C^2 t = 0 \\
 f_3(P) &= \sum_{i=1}^n 3C^4 t^2 P_0 + \sum_{i=1}^n 9t^3 C^3 P_1 + \sum_{i=1}^n 9t^4 C^2 P_2 + \sum_{i=1}^n 3Ct^5 P_3 - \sum_{i=1}^n 3Ct^2 = 0 \\
 f_4(P) &= \sum_{i=1}^n t^3 C^3 P_0 + \sum_{i=1}^n 3t^4 C^2 P_1 + \sum_{i=1}^n 3t^5 C P_2 + \sum_{i=1}^n t^6 P_3 - \sum_{i=1}^n t^3 = 0
 \end{aligned} \tag{4.3}$$

O procedimento iterativo aplicado no método de Newton-Raphson, é definido pela equação (4.4), essa equação tem o objetivo de retornar a cada iteração uma solução melhor para o sistema. Sendo assim, o método de Newton-Raphson consiste em reduzir a diferença em  $P_{(n+1)}$  e  $P_{(n)}$  em um valor mínimo, sendo esse valor considerado como o erro do método iterativo.

$$P_{(n+1)} = P_{(n)} - J_F^{-1}(P)F(P) \tag{4.4}$$

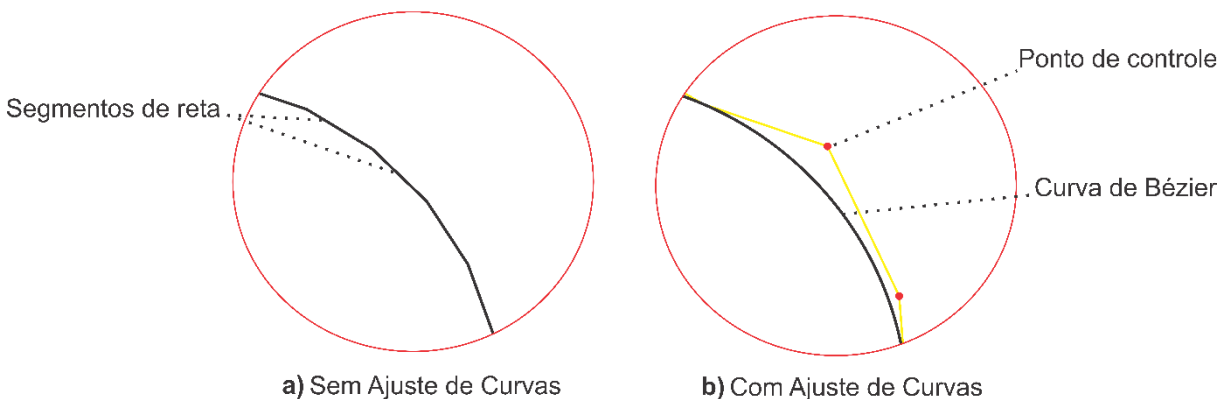
Para calcular  $J_F^{-1}$  é necessário encontrar a matriz jacobiana, definida pela equação (4.5), onde  $\frac{\partial f_1}{\partial P_1}(x)$  é a derivada da função  $f_1$ , dada na equação (4.3), em relação a variável  $P_1$ . Para encontrar os outros elementos da matriz são utilizadas definições de maneira análoga a apresentada para  $\frac{\partial f_1}{\partial P_1}(x)$ .

$$J(P) = \begin{bmatrix} \frac{\partial f_1}{\partial P_1}(P) & \frac{\partial f_1}{\partial P_2}(P) & \dots & \frac{\partial f_1}{\partial P_n}(P) \\ \frac{\partial f_2}{\partial P_1}(P) & \frac{\partial f_2}{\partial P_2}(P) & \dots & \frac{\partial f_2}{\partial P_n}(P) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial P_1}(P) & \frac{\partial f_n}{\partial P_2}(P) & \dots & \frac{\partial f_n}{\partial P_n}(P) \end{bmatrix} \quad (4.5)$$

Sendo assim, o resultado de cada iteração deve retornar uma matriz de soluções  $P$  que deve ser comparada em diferença de valor com o resultado da iteração anterior, a fim de gerar um valor que representa o erro obtido em cada iteração. Quando o erro da diferença entre as iterações for menor que  $10^{-6}$ , valor obtido a partir dos testes preliminares, a solução encontrada é aceita pelo algoritmo.

Quando aplicado o ajuste de curvas com o método dos mínimos quadrados no desenvolvimento do fatiador AMF-AC, foi possível observar uma melhora na suavidade das curvaturas horizontais nos trajetos. O fatiamento tradicional, sem ajuste de curva, ilustrado na Figura 35 (a) teve como resultado a composição aparente dos segmentos de retas em forma de curva, diferentemente do fatiamento com ajuste de curva ilustrado na Figura 35 (b), que demonstra uma curvatura suave e contínua.

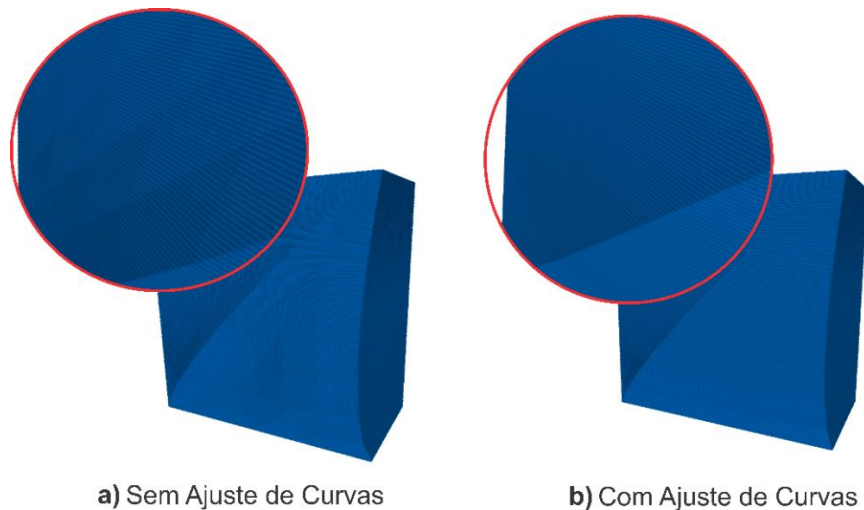
Figura 35 – Comparação de suavidade na curva entre os fatiamentos gerados pelo fatiador AMF-AC sem ajuste curva (a) e com ajuste de curva (b).



Fonte: Produção do autor.

A simulação 3D, que é basicamente uma união dos trajetos de todas as camadas, mostrou que o fatiamento com o ajuste de curva possui superfícies mais suaves em comparação a simulação do fatiamento sem o ajuste de curva. Essa comparação é ilustrada na Figura 36, na qual é possível identificar que a simulação sem o ajuste (a) apresenta triângulos resultantes da subdivisão realizada na etapa anterior do desenvolvimento, que sem a contribuição do ajuste de curva teriam que ser subdivididos novamente para melhorar a suavidade da superfície.

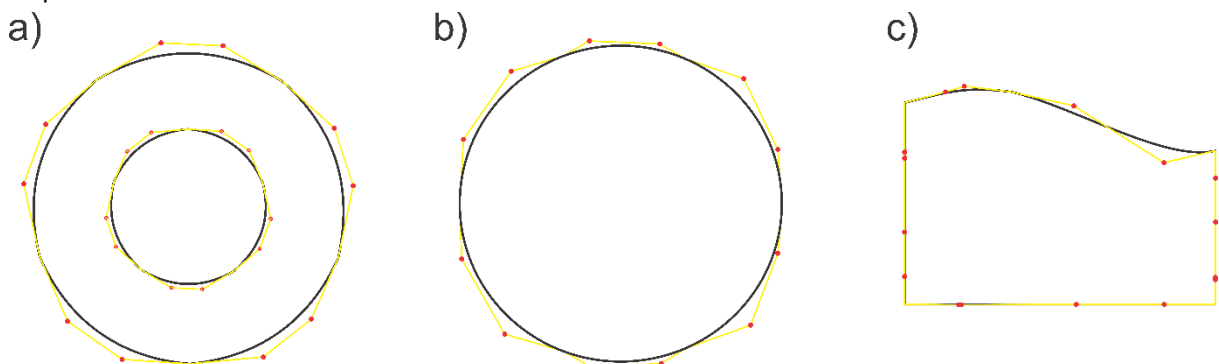
Figura 36 – Comparação entre a suavidade das superfícies das simulações 3D obtidas pelo fatiador.



Fonte: Produção do autor.

Finalmente, com a substituição das sequências de pontos de intersecção por curvas cúbicas de Bézier, foi então possível gerar os trajetos completos das camadas dos modelos. Alguns exemplos de trajetos gerados por esta abordagem são apresentados na Figura 37, onde os pontos vermelhos são pontos de controle e os segmentos de retas amarelos ajudam a identificar as curvas de Bézier. Esses trajetos, por sua vez são transformados em códigos de máquina na etapa seguinte do fatiador.

Figura 37 - Exemplos de sequencias de pontos ajustadas em curvas de Bézier pelo método de Newton-Raphson no Fatiador AMF-AC.



Fonte: Produção do autor.

#### 4.7 GERAÇÃO CÓDIGO EM LINGUAGEM G-CODE

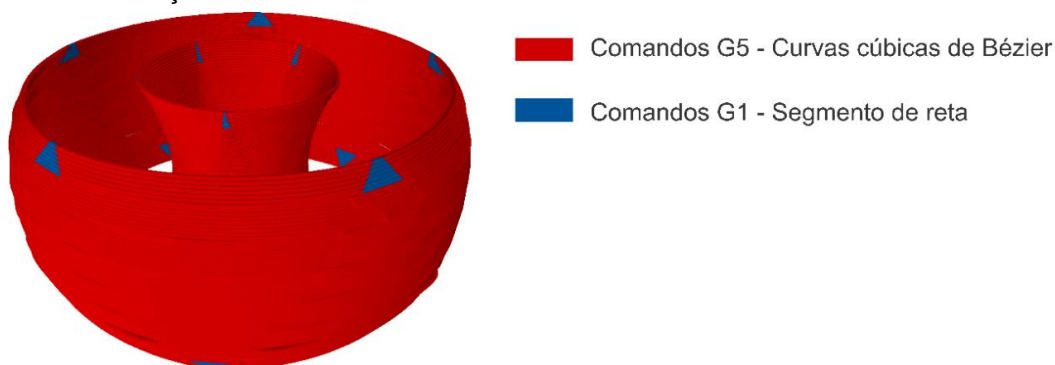
Para comprovar o funcionamento do fatiador AMF-AC é importante que o fatiador seja capaz de gerar o código em linguagem de CN, o que viabiliza a impressão das peças de teste na maioria das impressoras 3D. Portanto, para o desenvolvimento desta etapa foi escolhido trabalhar com a linguagem G-Code, que é a mais utilizada em impressoras 3D (ZELTMANN et al., 2016). Com isso torna-se possível a realização de testes nas impressoras FDM disponíveis para este trabalho.

Também é necessário definir para qual *firmware* o G-Code será desenvolvido, e essa escolha se faz necessária por causa das diferentes variações existentes entre os *firmwares* na interpretação da linguagem G-Code. Para isso foi escolhido trabalhar com o *firmware* Marlin, por ser o único *firmware* encontrado de código livre para impressoras FDM que é capaz de interpretar as curvas paramétricas, nesse caso, as curvas cúbicas de Bézier.

Foram implementados no fatiador AMF-AC métodos para transformar as curvas de Bézier, obtidas do ajuste de curvas, em comandos de curvas para o arquivo G-Code. Como as sequências com apenas dois pontos de intersecção não foram ajustadas, essas mantiveram seus comandos utilizando segmentos de retas, o que possibilitou o reaproveitamento de métodos já existentes no *software* MatterControl.

Como apresentado na seção 2.4.6, seguindo a documentação do *firmware* Marlin, cada curva de Bézier foi transformada em um comando G5, e cada segmento de reta foi transformado em um comando G1. Esses comandos são ilustrados na Figura 38, onde é possível perceber que ao menos uma das extremidades de cada triângulo curvo, inicialmente descrito no arquivo AMF, não foi ajustada para G5. Esse fenômeno ocorre por conta do posicionamento e tamanho dos triângulos em relação ao fatiamento horizontal. Sempre que a subdivisão resultar em algum triângulo com altura maior que a da camada, a sequência de pontos de intersecção desse triângulo resultará em apenas dois pontos de intersecção formando um segmento de reta (G1).

Figura 38 – Simulação do G-Code com os comandos G5 e G1.



Neste trabalho o cálculo das estruturas de suporte foi removido do fatiador, ou seja, não há geração de código para elas no G-Code. Assim, quando o algoritmo de geração de código em linguagem de CN estiver escrevendo as informações da camada, este deve ignorar qualquer cálculo relacionado ao suporte interno ou externo.

O Algoritmo 1 apresenta um resumo da lógica para geração do arquivo G-Code implementada no fatiador AMF-AC. O algoritmo inicia sempre que a interface do *software* emite um evento de clique do botão “generate”, localizado na parte inferior da interface gráfica da Figura 26. Ao receber esse evento, o fatiador reutiliza a lógica já existente no *software* MatterControl representada nas linhas 2 e 3; para isso o algoritmo cria um arquivo de saída do tipo G-Code, e então uma série de parâmetros definidos pelo usuário são escritos nesse arquivo.

Em seguida, na linha 4, o algoritmo executa para cada camada a repetição com o método de ajuste de curva (linha 5), e a tradução dos trajetos em comandos G-Code (entre as linhas 6 e 12).

Os comandos G-Code são inseridos no arquivo de acordo com o resultado do ajuste de curva. O ajuste de curva retorna várias sequências de pontos que podem representar curvas (três pontos) ou segmentos de reta (dois pontos). Para cada sequência maior que dois (linha 7), escreve-se o comando G5 utilizando os dois pontos de controle e o ponto final da curva (linha 8), caso contrário escreve-se um comando G1 utilizando o ponto final do segmento de reta (linha 11).

---

#### **Algoritmo 1** Geração de arquivo G-Code.

---

Linha

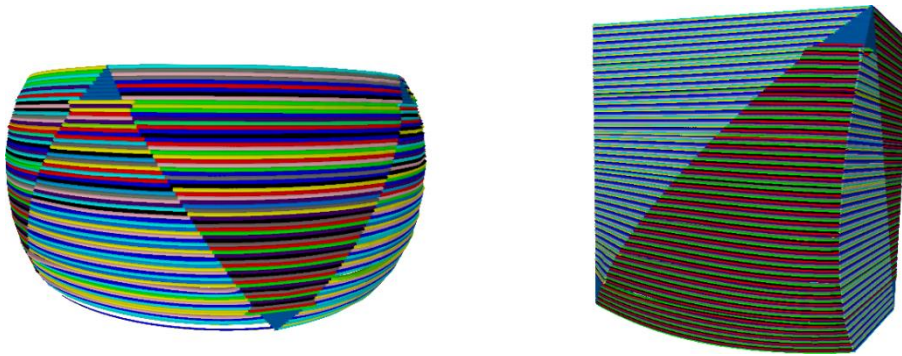
```

1:   input generate event
2:   gcode ← file.Load()
3:   gcode.write(preSetup())
4:   for (slicingData.Count()) {
5:     ExtruderLayer.OptimizePolygonsAMF()
6:     for (Polygons.Count()) {
7:       if (Sequence.idRoot.Count() > 2) {
8:         gcode.write(G5)
9:       }
10:      else {
11:        gcode.write(G1)
12:      }
13:    }
14:  }
```

---

Também foram desenvolvidos neste trabalho métodos capazes de apresentar graficamente as curvas de Bézier no G-Code gerado, tanto em 2D quanto em 3D. Na representação em 3D, cada comando de curva, escrito por G5, é ilustrado por uma curva, colorida com uma única cor escolhida aleatoriamente. Ilustradas na Figura 39, as representações em 3D priorizam mostrar estas curvas da forma que fiquem mais fáceis de visualizar sua localização com relação ao modelo. Já as representações em 2D, ilustradas nas Figura 35 e Figura 37, priorizam mostrar os pontos de controle de cada curva.

Figura 39 – Simulação colorida das curvas cúbicas de Bézier ajustadas pelo fatiador AMF-AC.



Fonte: Produção do autor.

O gerador de código em linguagem de CN é responsável por exportar o G-Code em um arquivo que contém todos os comandos de máquina necessários para a impressão completa da peça.

O Quadro 5 mostra partes de um arquivo exportado pelo fatiador AMF-AC, onde é possível visualizar exemplos dos comandos utilizados. Este arquivo G-Code inicia com os parâmetros de inicialização que são um conjunto de comandos essenciais para preparar o equipamento para a impressão. Nesse conjunto estão comandos como:

**G90 e M82.** Indicam que as coordenadas utilizadas terão posição absoluta;

**G92 E0.** Reinicia a contagem do filamento;

**G28.** Move o extrusor para a coordenada  $(0,0,0)$ ;

**M104 e M109.** Aquece e aguarda o aquecimento do extrusor até determinada temperatura;

**M106.** Define a velocidade da ventoinha do extrusor.

Nas linhas seguintes do arquivo são encontrados os comandos de extrusão e movimentação em curvas de Bézier e seguimentos de retas, G5 e G1, que em conjunto fazem o extrusor percorrer os trajetos definidos pelo fatiador. Ao percorrer esses trajetos, o extrusor da impressora deposita o material e imprime a peça.

Quadro 5 – Exemplo de arquivo G-Code exportado pelo fatiador AMF-AC.

Linha	Comandos G-Code
1:	<b>; Parâmetros de inicialização</b>
2:	<b>G90</b>
3:	<b>M82</b>
4:	<b>M106 S0</b>
5:	<b>M104 S220 T0</b>
6:	<b>M109 S220 T0</b>
7:	<b>G28</b>
8:	<b>G92 E0</b>
...	...
1010:	<b>G1 X138.468 Y69.877 E152.8949</b>
...	...
1120:	<b>; Camada:10</b>
1121:	<b>G5 E176.90081 F1368 I-2.917 J-1.098 P2.99 Q-1.131 X139.029 Y69.67</b>
1122:	<b>G5 E176.93181 I0.141 J-0.125 P-0.162 Q0.098 X138.502 Y70.053</b>
1123:	<b>G5 E177.2222 I-1.977 J2.464 P-0.149 Q-3.14 X136.357 Y76.654</b>
1124:	<b>G5 E177.2532 I-0.074 J-0.174 P0.044 Q0.184 X136.558 Y77.276</b>
1125:	<b>G5 E177.54355 I1.758 J2.674 P-3.005 Q-0.827 X142.174 Y81.354</b>
1126:	<b>G5 E177.57481 I-0.189 J0.015 P0.189 Q0.017 X142.826 Y81.355</b>
1127:	<b>G5 E177.86507 I3.068 J-0.842 P-1.733 Q2.622 X148.442 Y77.276</b>
1128:	<b>G5 E177.89594 I-0.045 J0.184 P0.073 Q-0.174 X148.643 Y76.654</b>
1129:	<b>G5 E178.18594 I0.142 J-3.129 P1.993 Q2.493 X146.498 Y70.053</b>
...	...

Fonte: Produção do autor.

Por fim, com a etapa de exportação do arquivo G-Code concluída no desenvolvimento do fatiador AMF-AC, este torna-se habilitado a exportar os comandos de curvas ajustadas para realização dos testes de impressão em impressoras 3D.

#### 4.8 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Neste capítulo é mostrado o método de fatiamento utilizando ajuste de curvas com o objetivo de tornar possível a geração de trajetória com curvas de Bézier, melhorando sua precisão em relação ao modelo CAD quando comparado a trajetórias apenas com segmentos de reta.

O método é desenvolvido em cinco etapas principais, que descrevem todo o funcionamento necessário para realizar o fatiamento de arquivos AMF, e ajustar os pontos de intersecção com a camada de forma a gerar curvas paramétricas para compor o contorno das camadas. E por fim a geração do código CN em linguagem G-Code desenvolvida para o *firmware* Marlin.



## 5 TESTES E ANÁLISE DE RESULTADOS

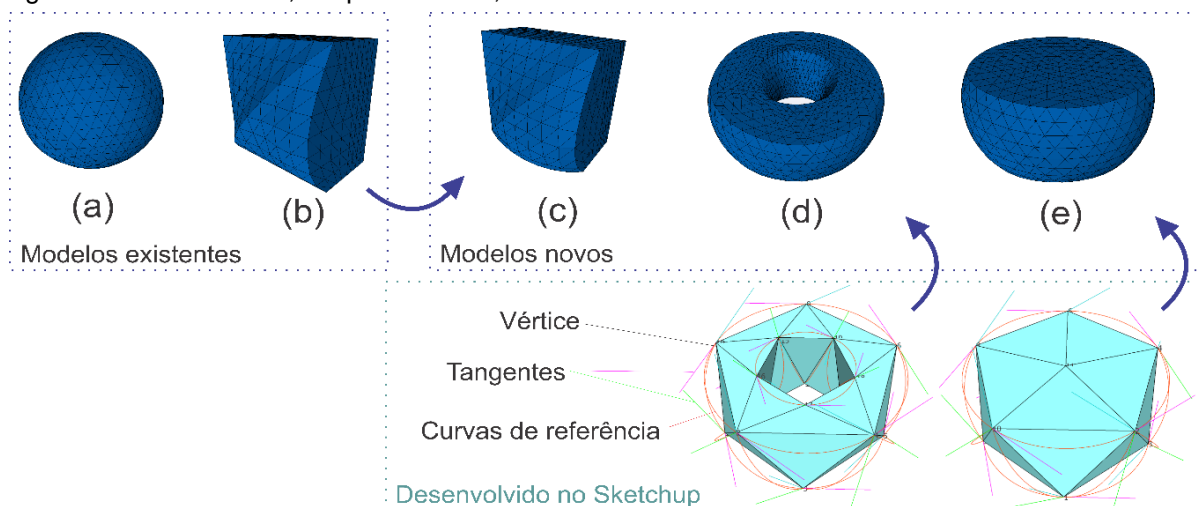
Para avaliar a aplicabilidade do fatiador AMF-AC no processo de produção com AM, foram utilizados modelos 3D, no padrão AMF, para a realização dos testes. Cada modelo 3D foi submetido ao processo de fatiamento desenvolvido neste trabalho. Este capítulo apresenta os modelos em AMF utilizados, e quais foram os resultados obtidos nas impressões de cada modelo. As peças resultantes dessas impressões foram analisadas com relação ao aspecto de fidelidade (apresentado na seção 2.2).

### 5.1 MODELOS 3D EM AMF

Entre os modelos 3D utilizados nos testes de impressão, dois foram obtidos pelo *website*<sup>1</sup> disponibilizado na ISO 52915:2016; esses são ilustrados na Figura 40 (a) e (b). Outros três modelos foram criados neste trabalho com auxílio do *software* de modelagem 3D Sketchup<sup>2</sup>. No entanto, assim como todos os outros *software* testados neste trabalho, o Sketchup também não possibilita gerar os triângulos curvos de acordo com a especificação do AMF.

Portanto, a criação dos novos modelos 3D em AMF foi realizada de forma manual, sendo necessário encontrar os vértices e as tangentes das superfícies curvas e escrever suas coordenadas em um arquivo texto. Esse método foi aplicado nas criações dos modelos (d) e (e) ilustrados na Figura 40, e para o modelo (c) foi necessário escrever apenas as tangentes da nova aresta curva na superfície inferior do modelo (b).

Figura 40 – Modelos 3D, em padrão AMF, utilizados nos testes com o fatiador AMF-AC.



Fonte: Produção do autor.

<sup>1</sup> <http://amf.wikispaces.com>, disponível em 1/06/2017.

<sup>2</sup> <https://www.sketchup.com>, disponível em 14/02/2019.

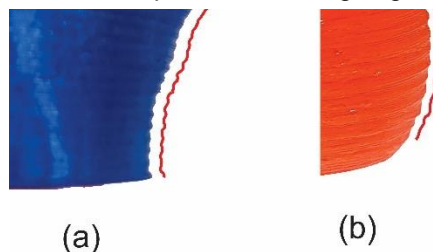
## 5.2 IMPRESSORAS FDM

Foi realizada uma parceria com a UFSC - Joinville que disponibilizou uma impressora RepRap com a placa Arduino Mega 2560 e com o *firmware* Marlin versão 1.1.0 instalado. Também para realizar a fabricação das peças, um rolo de filamento do tipo PLA na cor laranja, o qual possui como característica não necessitar o aquecimento da mesa de impressão, o que reduz o tempo de fabricação das peças, se comparado a outros tipos de filamento, como por exemplo o ABS.

Porém, para conseguir comparar os resultados com maior precisão foi utilizada também a impressora MakerBot Replicator 2X, com filamento do tipo ABS na cor branca, que tem uma maior precisão se comparada a impressora RepRap da UFSC - Joinville. No entanto, como a impressora MakerBot não interpreta comandos de curvas paramétricas, nessa foram utilizados apenas comandos de reta na linguagem de CN dessa impressora.

Outra característica observada relacionada ao método de impressão foi o efeito escada. As peças fabricadas pela impressora RepRap apresentaram um serrilhado acentuado, quando comparado com outras impressoras semelhantes, causado pelo efeito escada, destacado em vermelho na Figura 41 (b). A Figura 41 também apresenta a peça (a), em cor azul, com data de fabricação anterior a este trabalho, comprovando que a causa desta característica não está associada ao fatiador desenvolvido.

Figura 41 – Destaque no serrilhado causado pela falha na regulagem da impressora RepRap.



Fonte: Produção do autor.

A causa do efeito escada acentuado está relacionada com a altura das camadas, que durante os testes sofreram um aumento por causa da regulagem aplicada ao eixo Z da impressora. No entanto o efeito escada não sofreu alteração pelo método de fatiamento proposto; desta forma ele pode ser desconsiderado na análise dos resultados obtidos.

### 5.3 DEFINIÇÃO DE PARÂMETROS

Para cada uma das impressoras foram utilizados parâmetros diferentes de velocidade e temperatura de extrusão do filamento. As peças fabricadas na impressora RepRap utilizaram velocidade de 600mm/min e uma temperatura de 220°C. Para a impressora da MakerBot, os testes foram realizados com velocidade de 1200mm/min com uma temperatura de 210°C, e devido a essa impressora utilizar o filamento do tipo ABS foi necessário o pré-aquecimento da base de impressão a 120°C.

Outro parâmetro necessário para a impressão dos testes, que foi definido antes mesmo do fatiamento de cada modelo, foi a quantidade de subdivisões realizadas nos triângulos curvos. Essa quantidade de subdivisões está relacionada com a quantidade de pontos de intersecção obtidos no fatiamento, portanto implica diretamente na qualidade do ajuste das curvas de Bézier nesses pontos de intersecção. Para obter curvas mais precisas, foram aplicadas três subdivisões nos triângulos curvos do arquivo AMF, desta forma garante uma faixa de dois até 16 pontos de intersecção por triângulo curvo.

### 5.4 ANÁLISE DAS PEÇAS RESULTANTES

Em geral, as peças fabricadas utilizando o fatiador AMF-AC, tiveram resultados melhores que o fatiador padrão do AMF. Através das peças fabricadas foi possível perceber que o ajuste de curva contribuiu com a suavização das curvaturas horizontais.

As peças fabricadas na impressora RepRap tiveram um tempo médio de fabricação de 33 minutos. Esse tempo foi considerado elevado, em comparação com as peças impressas na impressora MakerBot, tendo em vista que todas as peças foram fabricadas sem suporte interno (com o interior da peça vazio), sendo necessário imprimir apenas as laterais externas. A causa deste tempo de impressão elevado foi atribuída ao parâmetro de velocidade de extrusão, que foi relativamente baixo se comparado com ao utilizado na impressora MakerBot.

A primeira peça fabricada nos testes foi a peça de formato cilíndrico com as laterais verticalmente arredondadas, e devido a isso recebeu o nome de peça “Disco”. O tempo de fabricação dessa peça foi de 20 minutos e produziu uma peça com dimensões finais de 35 milímetros de diâmetro e 15 milímetros de altura.

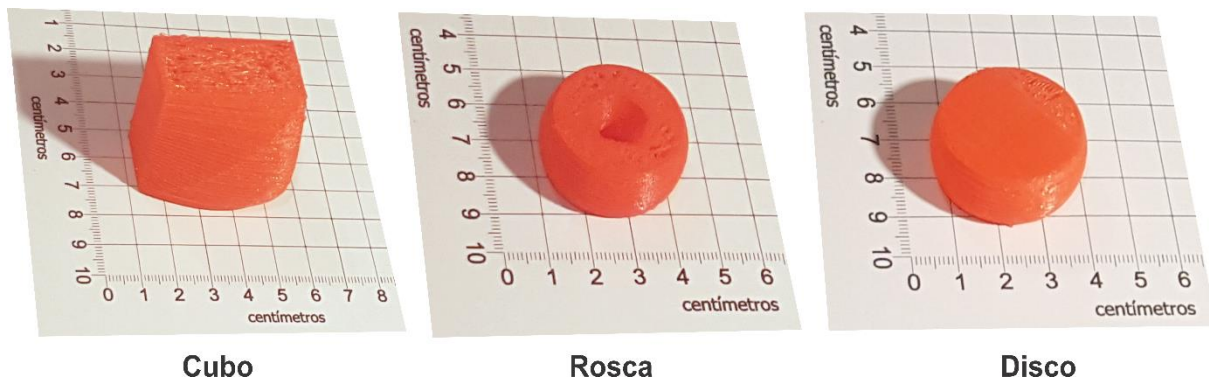
A peça Cubo obteve um tempo de fabricação de 48 minutos. Essa peça possui dimensões de 40 milímetros de altura, 41 milímetros de comprimento e 31 milímetros de largura. Para a fabricação da peça Rosca foram necessários 32 minutos, gerando uma peça com dimensões de 35 milímetros de diâmetro e 15 milímetros de altura.

Cada peça fabricada tem características únicas, planejadas para possibilitar a análise de cada uma das características de forma independente. Como, por exemplo a peça Cubo que é composta por triângulos curvos e triângulos planos com ângulos retos entre si; outro exemplo é o modelo Rosca, que possui trecho interno e externo para cada camada que o compõe; por fim, o modelo Disco que apresenta continuidade entre os triângulos curvos que compõem a superfície externa.

A peça originada a partir do modelo Cubo é mostrada na Figura 42, possuindo curvaturas suaves geradas para cada triângulo curvo, como previsto na simulação ilustrada anteriormente na Figura 39. Essa peça possui como principal característica os seus cantos com ângulos retos, que mostram a abordagem utilizada neste trabalho para o ajuste de curva, que mantém os ângulos definidos originalmente no modelo. Esta é uma abordagem diferente de outras, como a utilizada no trabalho de ZHAO et al. (2009), onde todos os pontos de intersecção de cada camada são aproximados em apenas uma curva B-Spline contínua.

Nas peças de teste Disco e Rosca, apresentadas na Figura 42, é possível visualizar que as superfícies contínuas definidas no arquivo AMF se mantiveram contínuas na região de união entre os triângulos curvos.

Figura 42 – Peças fabricadas utilizando a impressora RepRap com o comando G5 no G-Code.



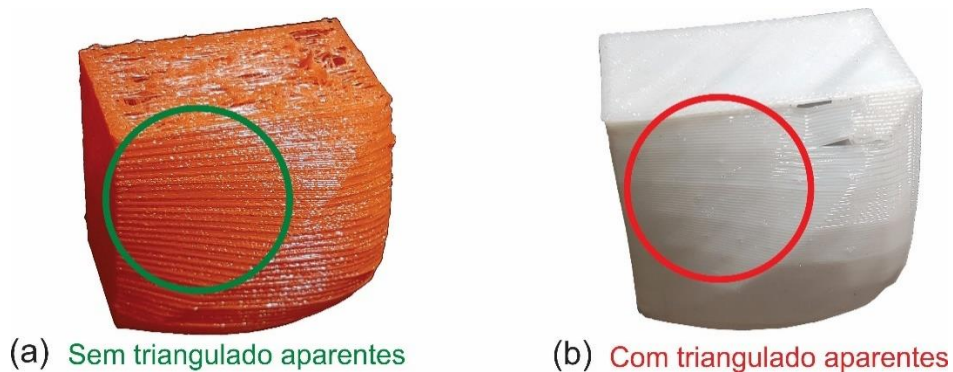
Fonte: Produção do autor.

A peça Rosca apresenta outra característica fundamental para comprovar o funcionamento do método proposto, que é a definição de múltiplos contornos por camada. Essa peça apresenta uma abertura central que implica em dois trajetos na mesma camada, onde cada um deles deve ser adequado pelo ajuste de curvas de forma independente, sem comprometer a identificação das áreas internas da peça. O fechamento da parte superior da peça fabricada mostra que o método aplicado manteve a definição da área interna da peça mesmo com mais de um contorno por camada.

Além das peças de teste fabricadas com a impressora RepRap e o filamento de cor laranja, para cada modelo 3D foi fabricada mais uma peça, sem aplicar o ajuste de curva, na impressora MakerBot, com o filamento de cor branca.

A Figura 43 (b) destaca o triangulado que ocorre devido as subdivisões dos triângulos curvos no arquivo AMF. Como não há o ajuste de curva entre os pontos de intersecção nessa peça, cada ponto é ligado por um segmento de reta durante o fatiamento, o que cria pequenos triângulos planos nas superfícies da peça fabricada.

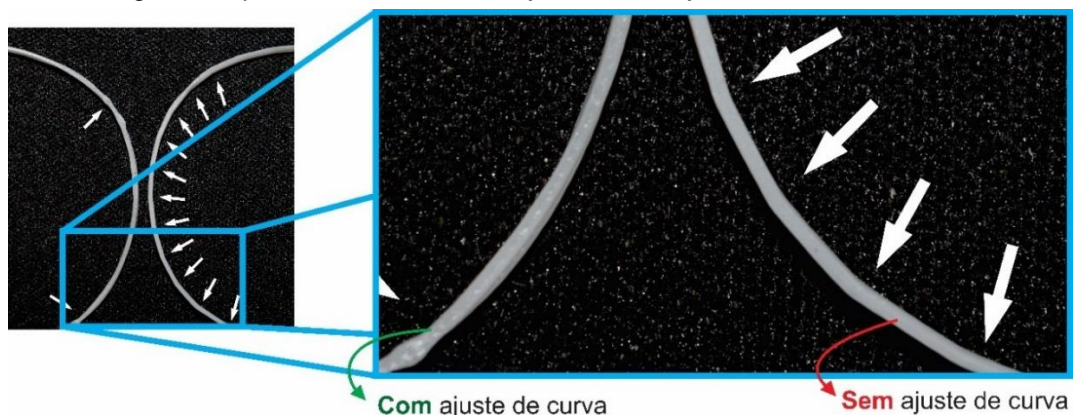
Figura 43 – Destaque das superfícies com ajuste de curva (laranja) e sem ajuste de curva (branca).



Fonte: Produção do autor.

Outro teste foi realizado com o objetivo de visualizar o impacto do ajuste de curvas no trajeto. Para esse teste foram geradas 10 camadas do contorno externo de um sólido circular com a impressora MakerBot. Como essa impressora não interpreta o comando G5, foi realizada a interpolação da curva de Bézier utilizando comandos G1 para simular a curva paramétrica. A Figura 44 mostra o ajuste de curva aplicado ao fatiamento do AMF (esquerda), o que reduziu o número de pontos de descontinuidade no trajeto da camada, mesmo sem utilizar o comando G5, quando comparado ao fatiamento sem o ajuste de curva (direita).

Figura 44 – Fotografia ampliada das curvas sem ajuste e com ajuste, fabricadas com filamento ABS.



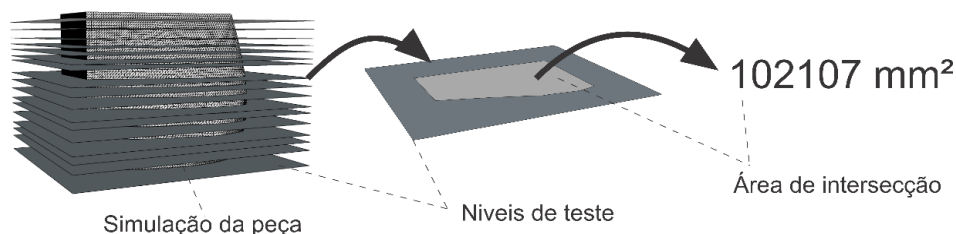
Fonte: Produção do autor.

Não foi possível analisar a diferença de volume com apenas as peças físicas produzidas nos testes de impressão. Esta análise havia sido planejada para ser executada com a imersão das peças em um recipiente contendo líquido, no entanto as superfícies das peças possuem muitas aberturas e irregularidades que impossibilitaram esse tipo de análise. Essas irregularidades foram atribuídas a qualidade de impressão obtida pela impressora RepRap e também pela ausência de suporte interno.

Portanto, para obter valores quantitativos sobre a redução do erro na aplicação do ajuste de curva, foram realizadas simulações de diferentes métodos de fatiamento da peça Cubo. Em cada método utilizado foram realizadas intersecções em níveis diferentes, com o objetivo de obter as áreas internas da peça. A fim de realizar uma análise semelhante à utilizada no Erro Real proposto por Zhao et al. (2009).

Os procedimentos de intersecções entre os níveis com as simulações das peças, são ilustrados na Figura 45, e foram realizados utilizando os *software* MakePrintable<sup>1</sup> e Sketchup. No MakePrintable, os trajetos gerados pelo fatiador AMF-AC foram convertidos em modelos sólidos, que em seguida foram importados no *software* Sketchup, possibilitando o cálculo da área de intersecção em cada nível.

Figura 45 – procedimentos realizados para quantificar a área dos níveis analisados.



Fonte: Produção do autor.

As áreas obtidas nas intersecções entre as simulações das peças com os níveis são comparadas com as áreas obtidas da intersecção utilizando os modelos AMF com três (3SUB) e seis (6SUB) subdivisões dos triângulos curvos, e são apresentadas no Apêndice B. Para cada simulação, foi gerada uma porcentagem obtida no cálculo de redução do erro de fidelidade, que é composto pelas diferenças entre as áreas. Esses resultados são apresentados na Tabela 4, com alguns valores discrepantes, destacados em negrito.

As reduções do erro na simulação utilizando apenas o ajuste de curva, exibido na segunda coluna (amarela) da Tabela 4, demonstram que apenas a aplicação do ajuste de curva não foi suficiente para melhorar a fidelidade entre a peça e o modelo, devido a variação dos valores entre negativo e positivo obtidos nos diferentes níveis.

<sup>1</sup> Software com transformações entre arquivos G-Code e STL.

Tabela 4 – Valores das porcentagens da redução de erro de fidelidade nos 18 níveis analisados

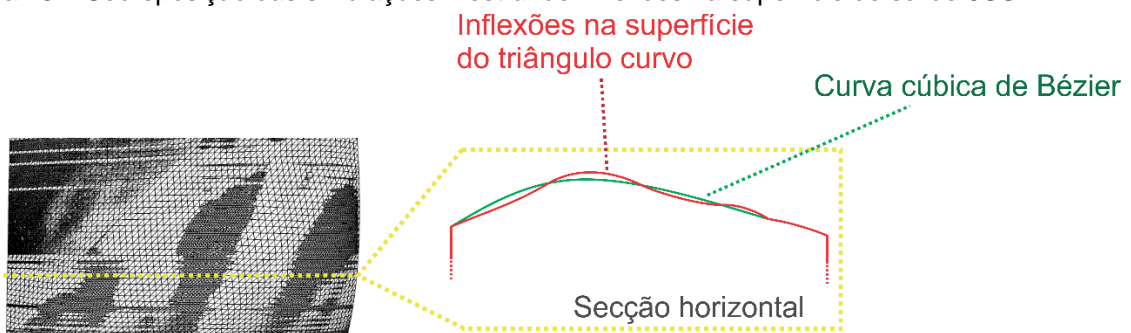
Níveis	Redução do Erro de Fidelidade		
	Com Ajuste de Curva	Com Ajuste de Curva e Intersecção em Curva	Com Intersecção em Curva
1	-20.30%	-87.1%	-10.3%
2	-23.20%	-20.0%	-49.2%
3	-6.40%	-47.6%	-96.4%
4	7.30%	-42.9%	-88.7%
5	-98.00%	-35.7%	96.9%
6	-11.20%	-21.2%	-96.1%
7	-23.50%	-55.9%	-100.0%
8	33.90%	-24.6%	-98.5%
9	47.70%	-13.6%	-97.9%
10	-94.40%	-75.5%	-89.6%
11	21.00%	-38.2%	-90.6%
12	-69.60%	-37.9%	-99.3%
13	1.20%	-61.0%	-88.9%
14	10.30%	-75.6%	-68.4%
<b>Médias</b>	<b>-16.09%</b>	<b>-45.49%</b>	<b>-69.79%</b>

Fonte: Produção do autor.

Além do ajuste de curva, neste trabalho foi realizado o fatiamento considerando também as curvas cúbicas de Hermite descritas nas arestas de cada triângulo curvo. Desta forma, para analisar o resultado final obtido pelo fatiador AMF-AC, outra simulação foi realizada, apresentada na terceira coluna (verde) da Tabela 4, considerando as intersecções das arestas curvas e o ajuste de curva. Em geral, nessa simulação houveram reduções no valor do erro, como esperado no comportamento do fatiador.

No entanto, ao analisar os resultados da simulação utilizando para o fatiamento apenas as arestas curvas, apresentado na quarta coluna (cinza) da Tabela 4, foi possível perceber uma redução maior do erro. Esse resultado foi atribuído a quantidade de inflexões na superfície do triângulo curvo ser maior que as quantidade de inflexões possíveis nas curvas cúbicas de Bézier, essas inflexões se destacam na sobreposição do sólido 6SUB com a simulação da peça utilizando G5 apresentada na Figura 46.

Figura 46 – Sobreposição das simulações mostrando inflexões na superfície do sólido 6SUB.



Fonte: Produção do autor.

## 5.5 PROBLEMAS ENCONTRADOS

Em alguns casos o fatiador AMF-AC não obteve sucesso na impressão das peças. A causa da maioria desses resultados indesejados está relacionada ao fato de algumas funcionalidades ainda não terem sido implementadas no fatiador como previsto no escopo do trabalho.

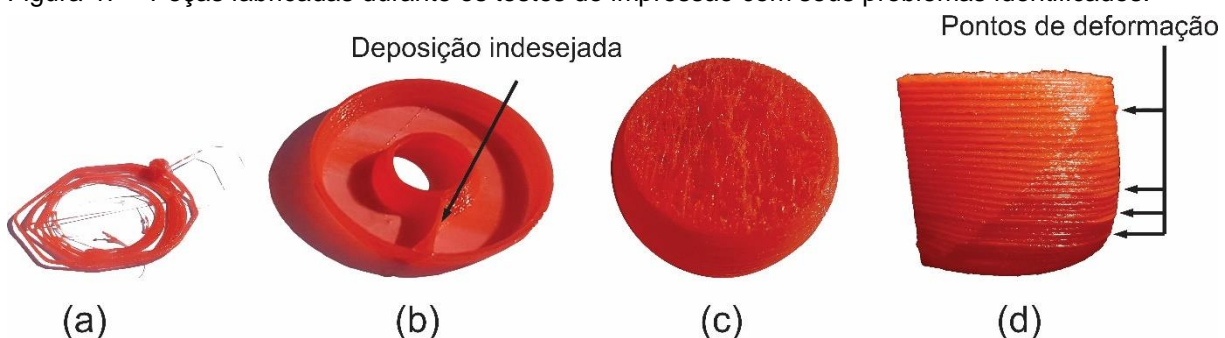
A Figura 47 apresenta peças que tiveram problemas durante a impressão. Dentre elas a Figura 47 (a), com o filamento enrolado, é um exemplo do resultado obtido pela impressão do modelo esférico (Figura 40(a)). Esse resultado demonstra a necessidade de implementação do cálculo do suporte externo no fatiador AMF-AC, pois a falta do mesmo fez com que o filamento das camadas superiores fosse depositado sem apoio.

Outro exemplo de funcionalidade ainda não implementada é a retração, que é o movimento contrário a saída de filamento no extrusor. Essa retração deve ser realizada ao final de cada trecho impresso, evitando assim que o filamento continue saindo do extrusor em locais indesejados. Esta deposição indesejada é ilustrada na Figura 47 (b) na região de ligação entre os trechos circulares, essa ligação não é parte do modelo e não deveria existir.

Com a impressão das peças de teste outra funcionalidade se mostrou necessária para melhorar qualidade na impressão. Ilustrada na Figura 47 (c) a peça Disco apresentou irregularidades na superfície de fechamento superior, causadas pela falta de suporte interno que foi prevista como fora do escopo do trabalho.

Por fim, a peça Cubo é ilustrada na Figura 47 (d) com destaque nos pontos de deformação da superfície curva. Essas deformações ocorrem devido a mudança na quantidade e na localização dos pontos de intersecção, utilizados no ajuste de curva para um determinado triângulo curvo. Ou seja, nas regiões onde a quantidade de pontos se altera com relação a camada de nível adjacente, ocorre também uma mudança drástica na posição dos pontos de intersecção, o que causa uma alteração significativa nas curvas ajustadas entre as camadas.

Figura 47 – Peças fabricadas durante os testes de impressão com seus problemas identificados.



Fonte: Produção do autor.

## 5.6 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Foram apresentadas neste capítulo as condições e definições dos testes realizados, assim como as análises sobre os resultados obtidos nas peças fabricadas. Esse capítulo ilustra os modelos que foram submetidos ao processo de fatiamento pelo AMF-AC. Em seguida, são descritas as impressoras e quais parâmetros foram utilizados durante as impressões das peças. Finalmente, as peças obtidas dos testes e suas simulações são analisadas quanto a melhoria na fidelidade bem como os valores quantitativos também são apresentados.

Com relação ao resultado obtido do fatiador AMF-AC, na avaliação das curvas geradas pelo ajuste de curvas aplicado aos pontos de intersecção, foi possível perceber uma melhora na suavidade, tanto nos trajetos das camadas quanto nas superfícies das peças fabricadas.

As simulações das impressões serviram para quantificar as diferenças entre as áreas obtidas a partir das intersecções entre os 18 níveis com as peças e os modelos. Os resultados dessas intersecções mostraram que a diferença entre o modelo AMF e a peça, foi menor na maioria dos casos quando utilizado o fatiador AMF-AC, do que quando utilizado o fatiador padrão do AMF.

Ao realizar as impressões com a linguagem G-Code utilizando o comando G5 para representar as curvas paramétricas, comprovou que esse comando pode ser utilizado com sucesso na fabricação de peças em FDM com o *firmware* Marlin. Sendo assim, o fatiador AMF-AC apresenta outra vantagem, pois seu fatiamento resultou em curvas paramétricas que, como comentado, podem ser utilizadas em impressoras 3D.

Portanto, o fatiador AMF-AC mostrou-se eficaz ao fatiar os modelos descritos no padrão AMF, e conseguiu gerar o G-Code com as curvas de Bézier. Mostrou também que a aplicação do ajuste de curva é possível mesmo que seja necessário interpolar a curva resultante em comandos G1.



## 6 CONCLUSÕES E TRABALHOS FUTUROS

A fabricação por AM tem evoluído em muitas características e nos mais diversos segmentos de aplicação, devido a sua simplicidade funcional e baixo custo de produção. Com isso, houve um aumento do número de usuários das impressoras de AM, tornando importante que a etapa de configuração do processo de fabricação se torne automatizada, rápida e confiável. Este trabalho contribuiu com o processo de configuração na escolha do padrão de representação do modelo 3D, pois foram apresentadas as vantagens relacionadas a utilização dos padrões com malhas de triângulos curvos, e também foram desenvolvidas melhorias automatizadas para o processo de fatiamento.

A análise dos trabalhos relacionados ao processo de fatiamento em AM mostrou um potencial da área na utilização de representações com triângulos curvos como substituto para a malha de triângulos planos utilizada no padrão STL. Embora os padrões de arquivos com malhas de triângulos curvos tenham trazido melhorias na representação dos modelos, o processo de fatiamento utilizado nesses padrões ainda implica em aproximações que causam erro de GD&T. Mesmo com o padrão AMF já aprovado e em utilização no mercado, esse ainda tem o seu processo de fatiamento em AM muito semelhante ao utilizado pelo padrão STL.

Nesse sentido, o presente trabalho agregou benefícios ao fatiamento de malhas triangulares curvas, mais especificamente com o AMF. Verificou-se também os aprimoramentos realizados nos trabalhos relacionados que contribuíram no fatiamento dos triângulos curvos; esses métodos ajudaram na resolução dos casos especiais na intersecção dos triângulos com as camadas. Desta forma, o desenvolvimento deste trabalho resultou na criação do fatiador AMF-AC, construído a partir do código do *software* MatterControl, que realiza o fatiamento de sólidos representados de fato em AMF. Diferente dos fatiadores existentes, o AMF-AC considera as informações de curvas descritas nos arquivos AMF pelas normais dos vértices e tangentes das arestas.

A criação do fatiador AMF-AC teve como foco a suavização das superfícies curvas horizontais através do uso do ajuste de curvas, e na redução do erro de fidelidade com a aplicação de intersecções nas arestas curvas dos triângulos. O desenvolvimento do fatiador AMF-AC seguiu as etapas definidas no PP, apresentadas na fundamentação teórica, onde em cada etapa foram aplicados os aprimoramentos necessários para alcançar os objetivos elencados neste trabalho.

Outra contribuição do trabalho foi o cálculo utilizado para obter a normal no ponto médio de cada aresta durante a subdivisão dos triângulos curvos descritos no padrão AMF. Foram propostos três métodos, sendo que para cada um dos métodos propostos foram realizadas simulações onde foi possível perceber as diferenças nas superfícies. Dentre os métodos simulados, foi escolhido utilizar para o desenvolvimento do fatiador o método de cálculo da normal através da soma entre as normais dos vértices adjacentes, pois essa abordagem apresentou uma melhor suavidade na curvatura da superfície gerada.

Na etapa das intersecções, realizada pelo fatiador AMF-AC logo após a subdivisão dos triângulos curvos, a fidelidade do fatiamento com a peça foi aprimorada pela utilização da intersecção da camada com a curva de Hermite definida nas arestas do triângulo curvo. Para isso foi utilizado o algoritmo de Newton-Raphson no cálculo das raízes da equação de intersecção entre a curva com o plano horizontal. Na análise dos resultados obtidos nas simulações, pôde-se perceber que a diferença nas áreas das intersecções obtidas pelo fatiamento, considerando as arestas curvas, foi em média 69,79% menor quando comparada com as intersecções considerando as arestas retas.

No entanto, a etapa de ajuste de curva não apresentou os mesmos resultados que os obtidos pelas intersecções nas análises quantitativas das simulações; nesses resultados as diferenças das áreas calculadas mostraram que o ajuste de curva proposto precisa ainda ser aprimorado. Em análise mais profunda, pode-se observar que a superfície gerada a partir das subdivisões apresentou múltiplas inflexões, exigindo uma curva de maior grau em seu ajuste. Entretanto, o resultado obtido pelo fatiador AMF-AC ainda foi positivo em relação a suavização das superfícies, tanto para as simulações quanto para as peças fabricadas nas impressoras 3D.

O fatiador desenvolvido também foi capaz de gerar o código em linguagem de CN com comandos para curvas de Bézier. Isso demonstrou outra vantagem na utilização do ajuste de curvas, pois implica que a qualidade da impressão das curvas dependa apenas da impressora. Assim, remove a tarefa do usuário de executar uma simplificação ao modelo durante o processo de impressão da peça.

Foi possível verificar que o fatiador obteve resultados satisfatórios na maioria dos casos. Através das peças fabricadas foi possível perceber que o AMF-AC contribuiu com a suavização das curvaturas horizontais, e que a intersecção das arestas curvas melhorou a fidelidade entre o modelo e a peça. Desta forma, o fatiador AMF-AC mostrou-se eficaz e promissor quando aplicado em representações que utilizem triângulos curvos de Hermite.

Em trabalhos futuros pretende-se aplicar os métodos do fatiador AMF-AC em outros padrões que utilizem malhas de triângulos curvos, como por exemplo a nova versão do STEP AP242 edição 2 (em atualização pela ISO) que propõe utilizar malhas cúbicas de Bézier.

O fatiador também poderia receber contribuições provindas de outras abordagens, como por exemplo a abordagem de Yang e Zheng (2013), que combina as informações dos três vértices do triângulo no cálculo da normal durante a subdivisão da malha triangular curva. Para avaliação desses e outros métodos de cálculo da normal pode-se realizar a sobreposição de modelos, ilustrada durante simulação na Figura 46.

Como a geração de suporte e o cálculo da retração do filamento não foram inclusos no escopo do fatiador, estas são funcionalidades que poderiam ser implementadas em uma segunda edição do fatiador AMF-AC. O que implicaria em interseccionar a curva de Bézier com os trajetos do suporte, após o ajuste de curvas.

O trabalho de Koc et al. (2000) também poderia ser combinado com a proposta deste trabalho, gerando uma solução para o problema do ajuste de curva nas múltiplas inflexões das superfícies. A abordagem de Koc et al. (2000) de analisar o ângulo da tangente da curva com os pontos de intersecção poderia ser aplicada na geração das curvas de Bézier, desta forma não seria necessário aumentar o grau da curva para conseguir melhorar os resultados obtidos no ajuste de curva.

Outro trabalho futuro poderia ser aplicado na etapa das subdivisões dos triângulos curvos, identificando locais onde o número de arestas que interseccionam a camada é insuficiente para manter a qualidade no ajuste de curvas. Nestes locais seria possível realizar subdivisões adaptativas para aumentar o número de interseções, até uma quantidade mínima, e melhorar o ajuste de curva.

Por fim, para avaliar com maior precisão a melhora da fidelidade na peça obtida pela aplicação do fatiador AMF-AC, o ideal seria utilizar uma mesma impressora 3D com melhor qualidade de resolução, e que possibilite a utilização de comandos para curvas paramétricas em sua linguagem de CN. Isso possivelmente abriria outras possibilidades de comparações físicas nas peças, como por exemplo os métodos de comparação de volume.



## REFERÊNCIAS

- ALEXANDER, P.; ALLEN, S.; DUTTA, D., Part orientation and build cost determination in layered manufacturing. **Computer-Aided Design**, v. 30, n. 97, p. 343 - 356, 1998.
- ALLAVARAPU, S.; PAUL, R.; ANAND, S., A new additive manufacturing file format using bezier patches. **Transactions of the North American Manufacturing Research Institution of SME**, v. 41, p. 580 - 590, 2013.
- ALMEIDA, Renato N., **O Método Dos Mínimos Quadrados: Estudo E Aplicações Para O Ensino Médio**. 2015. 69 p. Dissertação (mestrado) - Universidade Estadual Do Norte Fluminense, Mestrado em Matemática, Rio de Janeiro, UENF, 2015.
- ATTENE, M., A lightweight approach to repairing digitized polygon meshes. **Visual Computer**, v. 26, n. 11, p. 1393 - 1406, 2010.
- BOURELL, D. L. D.; BEAMAN, J. J.; LEU, M. C.; ROSEN, D. W., A brief history of additive manufacturing and the 2009 roadmap for additive manufacturing: looking back and looking ahead. **US-Turkey Workshop On Rapid Technologies**, 2009.
- BRACKETT, D.; ASHCROFT, I.; HAGUE, R., Topology optimization for additive manufacturing. **22nd Annual international solid freeform fabrication symposium**, p. 348 - 362, 2011.
- CACACE, S.; CRISTIANI, E.; ROCCHI, L., A level set based method for fixing overhangs in 3D printing. **Applied Mathematical Modelling**, v. 44, p. 446 - 455, 2017.
- CAFFREY, T.; WOHLERS, T.; CAMPBELL, R.I., Executive summary of the Wohlers Report 2016, **Wohlers Associates**, 2016.
- CHEN, Y.; WANG, C. C. L., Regulating complex geometries using layered depth-normal images for rapid prototyping and manufacturing. **Rapid Prototyping Journal**, v. 19, n. 4, p. 253 - 268, 2013.
- COX, J. J.; TAKEZAKI, Y.; FERGUSON, H. R. P.; KOHKONEN, K. E.; MULKAY, E. L., Space-filling curves in tool-path applications. **Computer-Aided Design**, v. 26, n. 3, p. 215 - 224, 1994.
- DEDE, E. M.; JOSHI, S. N.; ZHOU, F., Topology Optimization, Additive Layer Manufacturing and Experimental Testing of an Air-Cooled Heat Sink. **Journal of Mechanical Design**, v. 137, n. 11, 2015.
- DOLENC, A.; MÄKELÄ, I., Slicing procedures for layered manufacturing techniques. **Computer-Aided Design**, v. 26, n. 2, p. 119 - 126, 1994.
- EDEN, A. H., Three paradigms of computer science. **Minds and Machines**, 2007.
- FOLEY, J. D.; DAM, A. V.; FEINER, S. K.; HUGHES, J. F., **Computer Graphics: principles and practice**, 2ª edição, 1990.

- GIL, A. G., **Como Elaborar Projetos de Pesquisa**. 2002.
- GRIFFITHS, J. G., Toolpath based on Hilbert's curve. **Computer-Aided Design**, v. 26, n. 11, p. 839 - 844, 1994.
- GRIMM, T., User's guide to rapid prototyping. 2004.
- HAGEN, H.; POTTMANN, H., Curvature Continuous Triangular Interpolants. **Mathematical Methods in Computer Aided Geometric Design**, p. 373 - 384, 1989.
- HOPE, R. L.; ROTH, R. N.; JACOBS, P. A., Adaptive slicing with sloping layer surfaces. **Rapid Prototyping Journal**, v. 3, n. 3, p. 13, 1997.
- HU, K.; JIN, S.; WANG, C. C. L., Support slimming for single material based additive manufacturing. **Computer Aided Design**, v. 65, p. 1 - 10, 2015.
- HUANG, X.; YE, C.; WU, S.; GUO, K.; MO, J., Sloping wall structure support generation for fused deposition modeling. **International Journal of Advanced Manufacturing Technology**, v. 42, n. 11-12, p. 1074 - 1081, 2009.
- HUR, J.; LEE, K., The development of a CAD environment to determine the preferred build-up direction for layered manufacturing. **International Journal of Advanced Manufacturing Technology**, v. 14, n. 4, p. 247 - 254, 1998.
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, **ISO 52915: Additive manufacturing: IT applications in industry: Specification for additive manufacturing file format (AMF)**. p. 26. 2016
- ITOH, K.; OHNO, Y., A curve fitting algorithm for character fonts. **Electronic Publishing Origination, Dissemination, and Design**, v. 6, n. 3, p. 195, 1993.
- JIN, Y.; HE, Y.; FU, J. ZHONG, Support generation for additive manufacturing based on sliced data. **International Journal of Advanced Manufacturing Technology**, v. 80, n. 9 - 12, p. 2041 -2052, 2015.
- KIM, H. J.; WIE, K. H.; AHN, S. H.; CHOO, H. S.; JUN, C. S., Slicing algorithm for polyhedral models based on vertex shifting. **International Journal of Precision Engineering and Manufacturing**, v. 11, n. 5, p. 803 - 807, 2010.
- KIRSCHMAN, C.; JARA-ALMONTE, C., A parallel slicing algorithm for solid freeform fabrication processes. **Solid Freeform Fabrication Symposium**, p. 26 - 33, 1992.
- KOC, B.; MA, Y.; LEE, Y. Y., Smoothing STL files by Max-Fit biarc curves for rapid prototyping. **Rapid Prototyping Journal**, v. 6, n. 3, p. 186 - 205, 2000.
- KULKARNI, P.; MARSAN, A.; DUTTA, D., A review of process planning techniques in layered manufacturing. **Rapid Prototyping Journal**, v. 6, n. 1, p. 18 - 35, 2000.
- PEREIRA, L. R., **Ajuste De Curva B-Spline Fechada Com Peso**. 2014. 151 f. Dissertação (mestrado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Mecânica, Uberlândia, 2014.
- LIVESU, M.; ELLERO, S.; MARTÍNEZ, J.; LEFEBVRE, S.; ATTENE, M., From 3D models to 3D prints: an overview of the processing pipeline. **Computer Graphics Forum**, v. 36, n. 2, p. 537 - 564, 2017.

- LOOP, C.; SCHAEFER, S.; NI, T.; CASTAÑO, I., Approximating subdivision surfaces with Gregory patches for hardware tessellation. **ACM Transactions on Graphics**, v. 28, n. 5, p. 1, 2009.
- LUO, L.; BARAN, I.; RUSINKIEWICZ, S.; MATUSIK, W., Chopper: Partitioning Models into 3D-Printable Parts Linjie. **ACM Transactions on Graphics**, v. 31, n. 6, p. 1, 2012.
- MA, W.; BUT, W. C.; HE, P., NURBS-based adaptive slicing for efficient rapid prototyping. **Computer Aided Design**, v. 36, n. 13, p. 1309 - 1325, 2004.
- MAKHANOV, S. S.; ANOTAIPAIBOON, W., Advanced Numerical Methods to Optimize Cutting Operations of Five-Axis Milling Machines. 2007.
- MARCONI, M.; LAKATOS, E., Fundamentos de metodologia científica. 2005.
- MARSAN, A. L.; ALLEN, S.; KULKARNI, P.; DUTTA, D., An Integrated Software System for Process Planning for Layered Manufacturing. **Solid Freeform Fabrication Symposium**, p. 661 - 668, 1997.
- MASOOD, S. H.; RATTANAWONG, W.; IOVENITTI, P., Part build orientations based on volumetric error in fused deposition modelling. **International Journal of Advanced Manufacturing Technology**, v. 16, n. 3, p. 162 - 168, 2000.
- MCMAINS, S.; SÉQUIN, C., A coherent sweep plane slicer for layered manufacturing. **Proceedings of the fifth ACM symposium on Solid modeling and applications**, p. 285 - 295, 1999.
- MOHAN, P. P.; VENKATA, R. N.; DHANDE, S. G., Slicing procedures in layered manufacturing: a review. **Rapid Prototyping Journal**, v. 9, n. 5, p. 274 -288, 2003.
- MONTEIRO, M. T. F., **A Impressão 3D No Meio Produtivo e o Design: um estudo na fabricação de jóias**. 2015. 129 f. Dissertação (mestrado) - Universidade do Estado de Minas Gerais, Programa de Pós-Graduação em Design, Belo Horizonte, UFMG, 2015.
- MORTENSON, M. E., **Mathematics for Computer Graphics Applications**. Industrial Press Inc, p. 264, 1999.
- PANDEY, P. M.; REDDY, N. V.; DHANDE, S. G., Real time adaptive slicing for fused deposition modelling. **International Journal of Machine Tools and Manufacture**, v. 43, n. 1, p. 61 - 71, 2003.
- PAUL, R.; ANAND, S., A new Steiner patch based file format for Additive Manufacturing processes. **Computer Aided Design**, v. 63, p. 86 - 100, 2015.
- PERCOCO, G.; GALANTUCCI, L. M., Local-genetic slicing of point clouds for rapid prototyping. **Rapid Prototyping Journal**, v. 14, n. 3, p. 161 - 166, 2008.
- POPOOLA, P.; FAROTADE, G.; FATOBA, O.; POPOOLA, O., Laser Engineering Net Shaping Method in the Area of Development of Functionally Graded Materials ( FGMs ) for Aero Engine Applications - A Review. **Intech**, p. 383 - 400, 2016.
- PRÉVOST, R.; JAROSZ, W.; BÄCHER, M.; JAROSZ, W.; SORKINE-HORNUNG, O., Balancing 3D Models with Movable Masses. **Proceedings of the Vision, Modeling**

and Visualization Workshop, p. 8, 2016.

ROLLAND-NEVIÈRE, X.; DOËRR, G.; ALLIEZ, P., Robust diameter-based thickness estimation of 3D objects. **Graphical Models**, v. 75, n. 6, p. 279 - 296, 2013.

SABOURIN, E.; HOUSER, S. A.; BOHN, J. H., Accurate exterior ,fast interior layered manufacturing. **Rapid Prototyping Journal**, v. 3, n. 2, p. 44 - 52, 1997.

SEDERBERG, T. W.; ANDERSON, D. C., Steiner Surface Patches. **IEEE Computer Graphics and Applications**, v. 5, n. 5, p. 23–36, 1985.

SHAO, L.; ZHOU, H., Curve Fitting with Bézier Cubics. **Graphical Models and Image Processing**, v. 58, n. 3, p. 223 - 232, 1996.

STANCIULESCU, S.; SCHULZE, S.; WASOWSKI, A., Forked and integrated variants in an open-source firmware project. **2015 IEEE 31st International Conference on Software Maintenance and Evolution, ICSME 2015**, p. 151 - 160, 2015.

STARLY, B.; LAU, A.; SUN, W.; LAU, W.; BRADBURY, T., Direct slicing of STEP based NURBS models for layered manufacturing. **CAD Computer Aided Design**, v. 37, n. 4, p. 387 - 397, 2005.

SUH, Y. S.; WOZNY, M. J., Adaptive Slicing of Solid Freeform Fabrication Processes. **Solid Freeform Proceedings, University of Texas**, p. 404 - 411, 1994.

TATA, K.; FADEL, G.; BAGCHI, A. and AZIZ, N. Efficient slicing for layered manufacturing. **Rapid Prototyping Journal**, v. 4, n. 4, p. 151 - 167, 1998.

TAUFIK, M.; JAIN, P. K., Volumetric Error Control in Layered Manufacturing. **Volume 4: 19th Design for Manufacturing and the Life Cycle Conference; 8th International Conference on Micro- and Nanosystems**, n. August, p. V004T06A017, 2014.

TELEA, A.; JALBA, A., Voxel-based assessment of printability of 3D shapes. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 6671 LNCS, p. 393 - 404, 2011.

UM, J.; RAUCH, M.; HASCOËT, J. Y.; STROUD, I., STEP-NC compliant process planning of additive manufacturing: remanufacturing. **International Journal of Advanced Manufacturing Technology**, v. 88, n. 5-8, p. 1215 - 1230, 2017.

VAIDYA, R.; ANAND, S., Image Processing Assisted Tools for Pre- and Post-processing Operations in Additive Manufacturing. **Procedia Manufacturing**, v. 5, p. 958–973, 2016.

WANG, Q.; YANG, P.; HAN, W.; WEI, Q.; WANG, N., A New Slicing Method for AMF Model with Topology Structure. **2016 13th International Conference on Embedded Software and Systems**, p. 125 - 130, 2016.

WANG, W. M.; ZANNI, C.; KOBELT, L., Improved surface quality in 3D printing by optimizing the printing direction. **Computer Graphics Forum**, v. 35, n. 2, p. 59 - 70, 2016.

WEBER, C. L.; PENA, V.; MICALI, M. K. et al., The Role of the National Science Foundation in the Origin and Evolution of Additive Manufacturing in the United States. 2013.

YANG, X.; ZHENG, J., Shape aware normal interpolation for curved surface shading from polyhedral approximation. **User Modeling and User-Adapted Interaction**, v. 29, n. 3, p. 189 - 201, 2013.

ZELTMANN, S. E.; GUPTA, N.; TSOUTSOS, N. G. et al., Manufacturing and Security Challenges in 3D Printing. **Jom**, v. 68, n. 7, p. 1872–1881, 2016.

ZENG, L.; LAI, L. M. L.; QI, D.; LAI, Y. H.; YUEN, M. M. F., Efficient slicing procedure based on adaptive layer depth normal image. **Computer Aided Design**, v. 43, n. 12, p. 1577 - 1586, 2011.

ZHANG, Z.; JOSHI, S., An improved slicing algorithm with efficient contour construction using STL files. **International Journal of Advanced Manufacturing Technology**, v. 80, n. 5-8, p. 1347 - 1362, 2015.

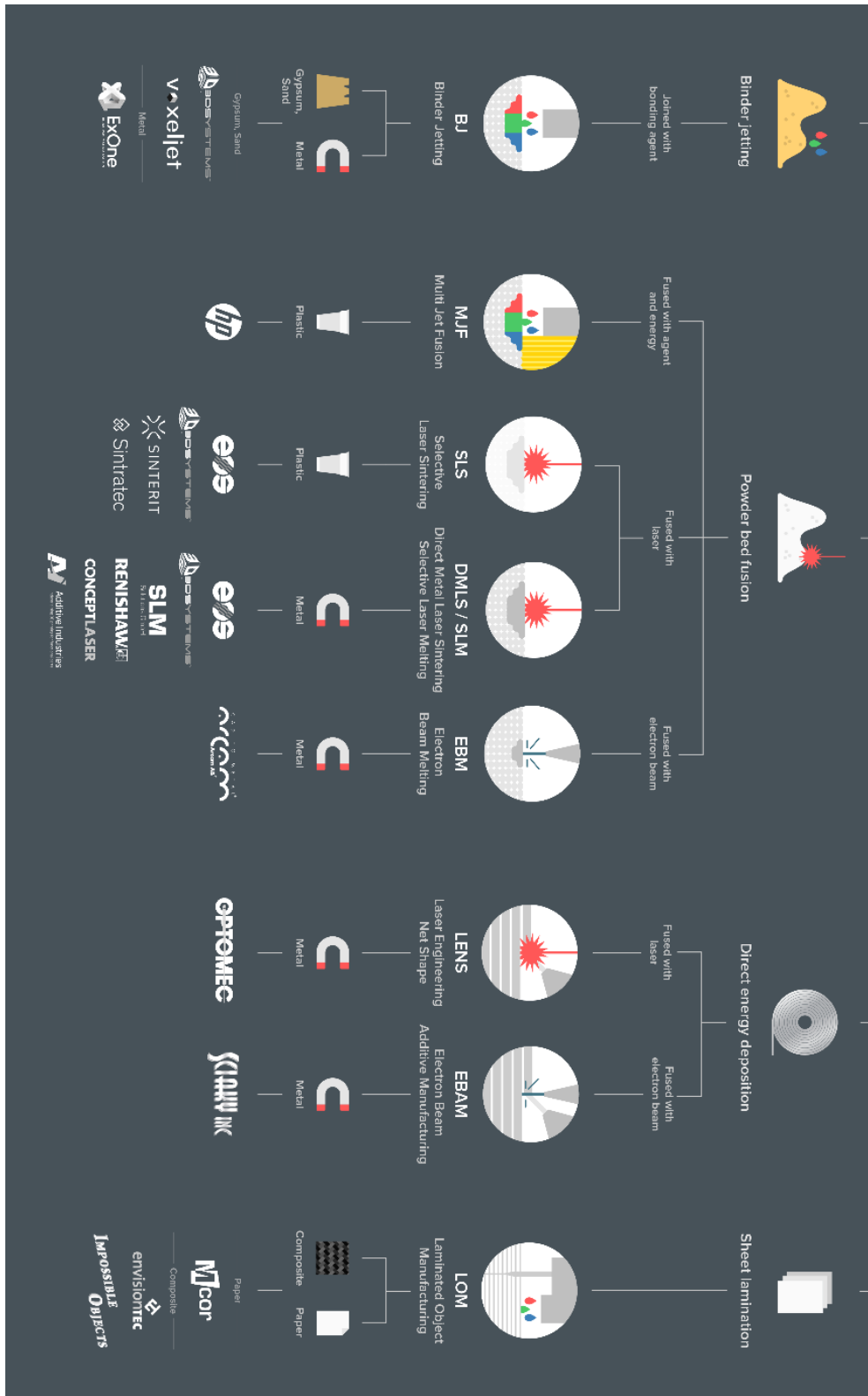
ZHAO, J.; XIA, R.; LIU, W.; WANG, H., A computing method for accurate slice contours based on an STL model. **Virtual and Physical Prototyping**, v. 4, n. 1, p. 29 - 37, 2009.

ZHOU, M. Y., STEP-based approach for direct slicing of CAD models for layered manufacturing. **International Journal of Production Research**, v. 43, n. 15, p. 3273 - 3285, 2005.

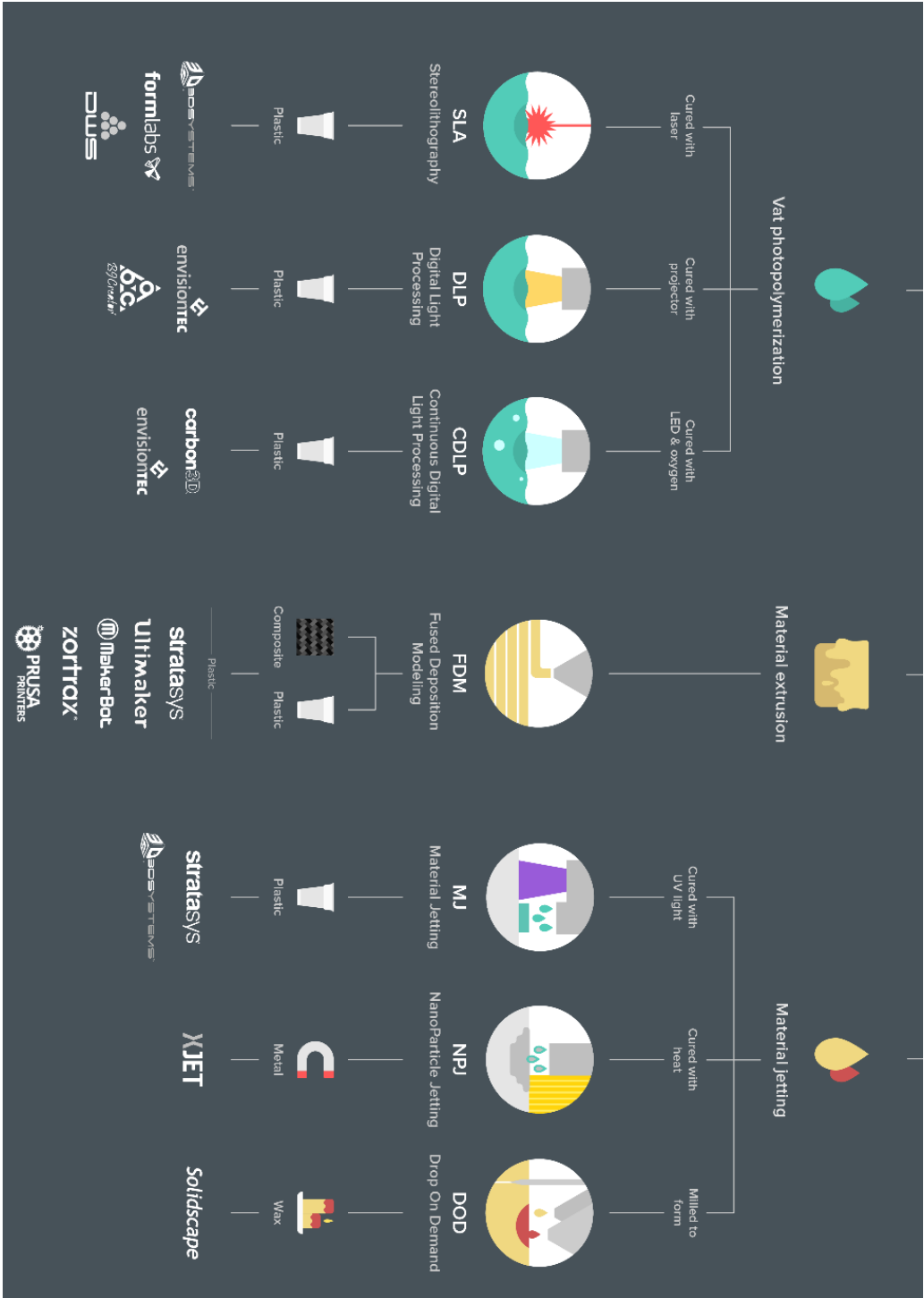
ZHOU, Q.; PANETTA, J.; ZORIN, D. Worst-case structural analysis. **ACM Transactions on Graphics**, v. 32, n. 4, p. 1, 2013.

## ANEXO A – RELAÇÃO DE MATÉRIAS E MÉTODOS DE IMPRESSÃO

Figura 48 – Detalhamento dos métodos e matérias utilizados na impressão por AM.



continua



Fonte: 3D Hubs [<https://www.3dprintingmedia.network/3d-hubs-publishes-complete-3d-printing-technologies-infographic/>]

## APÊNDICE A - PESQUISA DE FATIADORES EXISTENTES

Tabela 5 – Características e informações sobre os fatiadores analisados.

Empresa/ Criador	software nome	Codigo Aberto	Linguagem	Fatiamento			Algoritmo de Intersecção Implementado	adaptativo	Entrada		
				FDM	SLA	SLS			STL	AMF	STEP
Nick-Parker	Bread	✓	JAVA	✓	x	x		✓			
Revarbat	Mandoline	✓	C++	✓				✓		x	
Alexji	Slic3r	✓	Perl, C++	✓				✓			
Ultimaker	CuraEngine	✓	C++	✓			Versão C++ do Skeinforge	✓			
Bmander	Skeinforge	✓	Python	✓	✓	✓	Intersecção com a camada z	✓		x	
Simplify3D	Simplify3D	x		✓	x	x		✓			
MakerBot	Print	x		✓	x	x		x	✓	x	
MakerBot	Desktop	x		✓	x	x		x	✓	x	
MakerBot	Hackathon-slicer	✓	JavaScript	x	✓	x	Voxel/stencil buffer	x	✓	x	
Formlabs	Preform	x		x		x					
Formlabs	Ssschliizer	✓	C	✓			Intersecção com a camada z	✓			
Dronus	CraftWare	✓		✓				✓			
CraftBot 3D printer	CraftWare	✓		✓				✓			
KIS	KISSlicer	x		✓				✓			
MatterControl	MatterSlice	✓	C#	✓			Versão C# do CuraEngine	✓		✓	
Stewart Allen	kiri		JavaScript				Intersecção com a camada z				
	E3D										
ReprRap Creative Machines Lab	ReprRapPro_Slicer						Intersecção com a camada z				
	AMF Editor						não considera triângulos curvos			✓	

Fonte: Produção do autor.

## APÊNDICE B – SIMULAÇÕES DE INTERSECÇÕES

Tabela 6 – Áreas obtidas nas intersecções realizadas entre simulações de peças e 18 níveis diferentes.

Id dos Níveis	Área da intersecção (6SUB)	Área da intersecção (3SUB)	Diferença entre 6SUB e 3SUB	Com Ajuste de Curva		
				Área da intersecção	Diferença entre 6SUB	Redução do erro
1	83359	83048	311	83111	248	-20.30%
2	86486	86301	185	86344	142	-23.20%
3	89870	89620	250	89636	234	-6.40%
4	92252	91951	301	91929	323	7.30%
5	94212	94114	98	94210	2	-98.00%
6	96497	96318	179	96338	159	-11.20%
7	98030	97783	247	97841	189	-23.50%
8	102107	101774	333	101661	446	33.90%
9	104056	103578	478	103350	706	47.70%
10	105752	105632	120	105549	203	69.20%
11	107276	107007	269	107291	15	-94.40%
12	108027	107809	218	107759	268	22.90%
13	108837	108313	524	108203	634	21.00%
14	108800	108494	306	108893	93	-69.60%
15	107992	107576	416	107500	492	18.30%
16	106573	106058	515	106052	521	1.20%
17	104420	103983	437	104347	73	-83.30%
18	101103	100222	881	100131	972	10.30%

continua

Id dos Níveis	Com Ajuste de Curva e Intersecção em Curva				Com Intersecção em Curva			
	Área da intersecção	Diferença entre 6SUB	Redução do erro	Área da intersecção	Diferença entre 6SUB	Redução do erro		
1	83319	40	-87.1%	83638	279	-10.3%		
2	86338	148	-20.0%	86580	94	-49.2%		
3	89739	131	-47.6%	89879	9	-96.4%		
4	92080	172	-42.9%	92286	34	-88.7%		
5	94149	63	-35.7%	94405	193	96.9%		
6	96356	141	-21.2%	96490	7	-96.1%		
7	97921	109	-55.9%	98030	0	-100.0%		
8	101856	251	-24.6%	102112	5	-98.5%		
9	103643	413	-13.6%	104066	10	-97.9%		
10	105466	286	138.3%	105672	80	-33.3%		
11	107342	66	-75.5%	107304	28	-89.6%		
12	107690	337	54.6%	107997	30	-86.2%		
13	108513	324	-38.2%	108788	49	-90.6%		
14	108990	190	-37.9%	108802	2	-99.3%		
15	107720	272	-34.6%	109289	1297	211.8%		
16	106372	201	-61.0%	106630	57	-88.9%		
17	104470	50	-88.6%	105747	1327	203.7%		
18	100888	215	-75.6%	101381	278	-68.4%		