



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

IGOR DE MELO XAVIER

OTIMIZAÇÃO DA QUALIDADE DE IMPRESSÃO 3D ATRAVÉS DO USO DE
SPLINES DE BIARCOS EM G-CODE: UM ESTUDO EXPERIMENTAL EM
GEOMETRIAS COMPLEXAS

Recife
2024

IGOR DE MELO XAVIER

OTIMIZAÇÃO DA QUALIDADE DE IMPRESSÃO 3D ATRAVÉS DO USO DE
SPLINES DE BIARCOS EM G-CODE: UM ESTUDO EXPERIMENTAL EM
GEOMETRIAS COMPLEXAS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de mestre em Ciência da Computação. Área de Concentração: Correspondente ao indicado na ata de defesa.

Orientador: Silvio de Barros Melo

Coorientador: José Edeson de Melo Siqueira

Recife

2024

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Xavier, Igor de Melo.

Otimização da qualidade de impressão 3D através do uso de splines de biarcos em G-code: um estudo experimental em geometrias complexas / Igor de Melo Xavier. - Recife, 2024.
61f.: il.

Dissertação (Mestrado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-Graduação em Ciência da Computação, 2024.

Orientação: Silvio de Barros Melo.

Coorientação: José Edeson de Melo Siqueira.

Inclui referências e apêndices.

1. Impressão 3D; 2. Modelagem; 3. Biarcos. I. Melo, Silvio de Barros. II. Siqueira, José Edeson de Melo. III. Título.

UFPE-Biblioteca Central

Igor de Melo Xavier

“Otimização da qualidade de impressão 3D através do uso de splines de biarcos em G-code: um estudo experimental em geometrias complexas”

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de mestre em Ciência da Computação. Área de Concentração: Computação Gráfica.

Aprovado em: 21 novembro de 2024.

BANCA EXAMINADORA

Prof. Dr. Filipe Carlos de Albuquerque Calegário

Centro de Informática/UFPE

Prof. Dr. Walter Franklin Marques Correia

Centro de Artes e Comunicação/UFPE

Prof. Dr. Jose Edeson de Melo Siqueira

Centro de Artes e Comunicação/UFPE

RESUMO

A manufatura aditiva, amplamente conhecida como impressão 3D, tem transformado diversas indústrias ao permitir a prototipação rápida de objetos com geometrias complexas, ou seja, objetos que apresentam grandes variações de curvatura. No entanto, a qualidade das superfícies impressas ainda enfrenta limitações, especialmente nestas formas curvas, onde a suavidade é crucial para o acabamento estético e funcional. Tradicionalmente, os caminhos percorridos pelo cabeçote de impressão são descritos por segmentos de reta entre pontos adjacentes amostrados no processo de fatiamento, o que resulta em superfícies facetadas e pouco precisas. A fim de superar esse problema, foi proposto na literatura o uso de splines G^1 aproximativa de biarcos no lugar de segmentos de reta no G-code, uma vez que essa técnica permite conectar diversos biarcos de circunferência de maneira suave, preservando a continuidade tangencial e eliminando cúspides. O objetivo principal deste trabalho é realizar um estudo experimental que avalie os ganhos qualitativos proporcionados pelos splines G^1 de biarcos na impressão 3D de objetos com geometrias complexas. Para isso, serão gerados G-codes utilizando tanto o método tradicional de interpolação por segmentos retilíneos, aquele utilizado pelos softwares fatiadores, quanto o método aproximativo proposto com biarcos. Os G-codes serão comparados e será avaliado a eficiência do método proposto. Foi comprovado através da análise de código G-code exportado que o algoritmo desenvolvido calculou o mesmo trajeto utilizando comandos G2 e G3 de trajeto, entregando uma precisão curva ao invés daquela limitada por segmentos de reta. Esse estudo busca contribuir para o aprimoramento das técnicas de fatiamento e, conseqüentemente, para a melhoria da qualidade da impressão 3D em aplicações industriais e de design.

Palavras-chaves: Impressão 3D, modelagem, biarcos, geometrias complexas

ABSTRACT

Additive manufacturing, widely known as 3D printing, has revolutionized various industries by enabling rapid prototyping of objects with complex geometries, that is, objects exhibiting significant curvature variations. However, the quality of printed surfaces still faces limitations, especially in these curved forms where smoothness is crucial for both aesthetic and functional finishes. Traditionally, the paths followed by the printhead are described by straight-line segments connecting adjacent points sampled during the slicing process, which results in faceted and imprecise surfaces. To address this issue, the use of G^1 approximate splines based on biarc curves has been proposed in the literature as a substitute for straight-line segments in the G-code. This technique allows multiple circular biarcs to be connected smoothly, preserving tangential continuity and eliminating cusps. The primary goal of this study is to conduct an experimental evaluation of the qualitative improvements provided by G^1 biarc splines in the 3D printing of objects with complex geometries. For this purpose, G-codes will be generated using both the traditional method of straight-line interpolation, commonly employed by slicer software, and the proposed approximation method with biarcs. These G-codes will be compared to assess the efficiency of the proposed method. It is expected that the use of G^1 biarc splines will result in smoother and higher-quality trajectories without significantly increasing processing time. This study aims to contribute to the advancement of slicing techniques and, consequently, to the improvement of 3D printing quality in industrial and design applications.

Keywords: 3D printing, modeling, biarcs, complex geometries

LISTA DE ILUSTRAÇÕES

Figura 1: Esquema de impressora 3D por FDM e suas estruturas básicas

Figura 2: Esquema dos processamentos gráficos que um modelo tridimensional sofre durante o pipeline da impressão 3D

Figura 3: Sólido 3D no CAD e a curva de perfilamento geradora da forma

Figura 4: Visualização do G-code no PrusaSlicer

Figura 5: Snippet de G-code - Exemplo de uso completo dos comandos G3

Figura 6: Impressora 3D em estado finalizado de uma impressão

Figura 7: Trajetos de G-code em segmentos de reta

Figura 8: Dados associados às tangentes t_B e t_A ; exemplo de biarco que une o ponto A ao B e que concorda com as tangentes t_B e t_A

Figura 9: Construção geométrica da curva

Figura 10: Sólido projetado a partir da curva

Figura 11: Detalhamento do efeito de tesselação do modelo em .STL no Paint 3D

Tabela 1: Códigos G e M para G-code

Snippet 1: G-code com apenas comandos G1 (segmentos de reta)

Snippet 2: Extração das coordenadas no arquivo de saída da fatiadora

Snippet 3: Geração de splines

Snippet 4: Geração de arcos

Snippet 5: Múltiplas iterações para diferentes camadas

Snippet 6: Salvamento de um arquivo G-code novo

Snippet 7: Visualização da figura da spline interpolada

Snippet 8: G-code de impressão a partir do código gerado pelo PrusaSlicer

Snippet 9: G-code em biarcos, um exemplo de uso completo dos comandos G3

Snippet 10: G-code em biarcos, um exemplo de uso completo dos comandos G2

LISTA DE ABREVIATURAS E SIGLAS

AM: Manufatura Aditiva

CAD: Desenho Assistido por Computador

CNN: Redes Neurais Convolucionais

FDM: Modelagem por Deposição Fundida

FFF: Fabricação por Filamento Fundido

G-code: Código de Controle Numérico

G1: Comando G para Movimento Linear

G2: Comando G para Movimento Circular Sentido Horário

G3: Comando G para Movimento Circular Sentido Anti-Horário

LLMs: Modelos de Linguagem de Grande Escala

ML: Aprendizado de Máquina

Numpy: Biblioteca de Python para Manipulação Numérica

NURBS: Non Uniform Rational Basis Spline

SLA: Estereolitografia

STL: Formato de Arquivo Estereolitográfico

XYZ: Coordenadas Tridimensionais

Z-Score: Pontuação padronizada em estatísticas

SUMÁRIO

1. INTRODUÇÃO	9
1.1 Discussão	9
1.2 Problema de pesquisa	23
1.3 Hipótese	24
1.4 Objetivos	25
1.4.1 Objetivo Geral	25
1.4.2 Objetivos Específicos	25
1.5 Metodologia	25
1.6 Justificativa	27
2. REVISÃO DA LITERATURA	30
2.1 Impressão 3D	30
2.2 G-code	33
2.3 Biarcos	35
3. MODELAGEM 3D	37
4. ALGORITMO DE GERAÇÃO DE BIARCOS	40
5. APLICAÇÃO DO ALGORITMO	45
6. AVALIAÇÃO E DISCUSSÃO	49
7. CONCLUSÃO	53
8. REFERÊNCIAS	55
APÊNDICE A – Algoritmo de conversão de G-code	60

1. INTRODUÇÃO

1.1 Discussão

A prototipagem rápida representa um grande avanço na indústria criativa e na testagem de protótipos de diversas naturezas. Dentro dela, a impressão 3D configura um avanço sem precedentes na rapidez de materialização de objetos e artefatos tridimensionais físicos diversos. As áreas criativas, de design, engenharia e saúde experimentam nas últimas décadas inúmeros avanços no estado da arte na associação de suas respectivas áreas e a impressão 3D. Não diferentemente, a área do ensino e educação utilizam a impressão 3D como ferramenta de aproximação dos estudantes nos diversos níveis de ensino com a Cultura Maker e em competências criativas e tecnológicas. Todas essas aplicações remetem a um constante avanço tecnológico no pipeline da impressão 3D que ocorre e é fortemente buscado pela comunidade acadêmica e também por entusiastas (NGO et al., 2018).

A prototipagem rápida pode ser dividida em duas modalidades: a) manufatura subtrativa e b) manufatura aditiva. A manufatura subtrativa implica na remoção de material de um bloco até formar a peça desejada, muito comum em máquinas de Controle Numérico Computadorizado (CNC). A manufatura aditiva configura diversas técnicas de materialização de objetos e é popularmente chamada de impressão 3D (HORVARTH, 2014).

Desde a criação de produtos simples até itens complexos e de alta precisão, a manufatura aditiva se destaca pelo uso de diversas técnicas. Dentre as técnicas mais relevantes, temos a Estereolitografia (SLA) que foi o primeiro processo de impressão 3D. Lançado em 1988, utiliza um feixe de laser direcionado por espelhos e lentes para curar uma resina fotossensível à base de acrilato ou epóxi. A polimerização pode ocorrer numa precisão de até 0,075 mm, permitindo a construção camada a camada, em espessuras entre 0,5 mm e 0,025 mm, sobre uma plataforma submersa que desce após cada camada ser finalizada (BARROS, 2012). Avanços tecnológicos mostram utilização de outras técnicas de polimerização, como a de fóton duplo (SONG; XING, 2015). Após a construção, a peça é limpa com solventes para remover a resina não curada, as estruturas de suporte são retiradas,

e a peça passa por pós-cura em forno, seguido de acabamento manual. Apesar de ser amplamente difundido e proporcionar alta precisão e qualidade de superfície, esse método exige pós-processamento e estrutura de suporte para áreas desconectadas, além de usar resina tóxica.

A Manufatura de Objetos em Lâminas (LOM) utiliza lâminas alimentadas por bobinas. Cada lâmina possui um adesivo ativado por um rolo aquecido que une as camadas, e um laser recorta a seção desejada enquanto marca o material excedente, que serve como suporte para a próxima camada e pode ser removido manualmente. É compatível com materiais diversos, como metais, cerâmicas, plásticos, e é especialmente usado com papel, exigindo um selador após a finalização. Processos de acabamento, como lixamento e pintura, também podem ser aplicados. As vantagens do LOM incluem alta velocidade de construção e ausência de necessidade de pós-cura ou estruturas de suporte. As desvantagens incluem a dificuldade para testar encaixes, a remoção trabalhosa do suporte e uma certa limitação em peças com detalhes finos ou vazios internos (GROSS et al., 2016).

Na técnica de Sinterização Seletiva a Laser (SLS), a câmara de construção é preenchida com material em forma de pó, que é então aquecido até próximo de seu ponto de fusão. Um laser percorre o material em forma de pó, sinterizando os pontos de contato entre partículas ao invés de fundi-las completamente, formando uma camada sólida representando a seção transversal da peça. O excesso de pó é retirado, e a plataforma de apoio é rebaixada para criar a camada seguinte. Esta técnica aceita qualquer material que possa ser transformado em pó e possua características físico-químicas compatíveis com a sinterização. Uma vantagem dessa técnica é a variedade de materiais que podem ser usados, incluindo areia, metal, polímeros e cerâmica. Também é possível empilhar várias peças para serem fabricadas simultaneamente. Outras vantagens incluem a possibilidade de utilizar diversos tipos de materiais no mesmo equipamento, facilidade na remoção do material de suporte, e a produção tanto de peças de visualização quanto funcionais. No entanto, esse processo tem limitações como o acabamento superficial moderado e o alto custo de equipamento e consumo de energia (FINA et al., 2017).

Similarmente ao SLS, temos a impressora 3D por *inkjet*. *Nela, é utilizado* uma base de pó nivelada por um rolo. No entanto, ao invés de utilizar um laser para fundir

o material, um cabeçote deposita a jato um ligante químico sobre o pó, formando camadas conforme o design em arquivo. A cada camada o rolo adiciona mais pó e o processo se repete (GROSS et al., 2016).

A Impressão Tridimensional (3DP) é semelhante à técnica de Inkjet, pois uma cabeça de impressão deposita um aglutinante sobre um leito de pó, formando a peça camada a camada. O pó não aglutinado atua como suporte, dispensando materiais adicionais para suportes. Esse processo permite flexibilidade de materiais, incluindo cerâmica, metal, polímeros e gesso. No entanto, a peça inicial é frágil, exigindo pós-processamento com resinas ou metais para aumento de resistência, semelhante ao Polyjet, que também usa pós-processamento (BARROS, 2012).

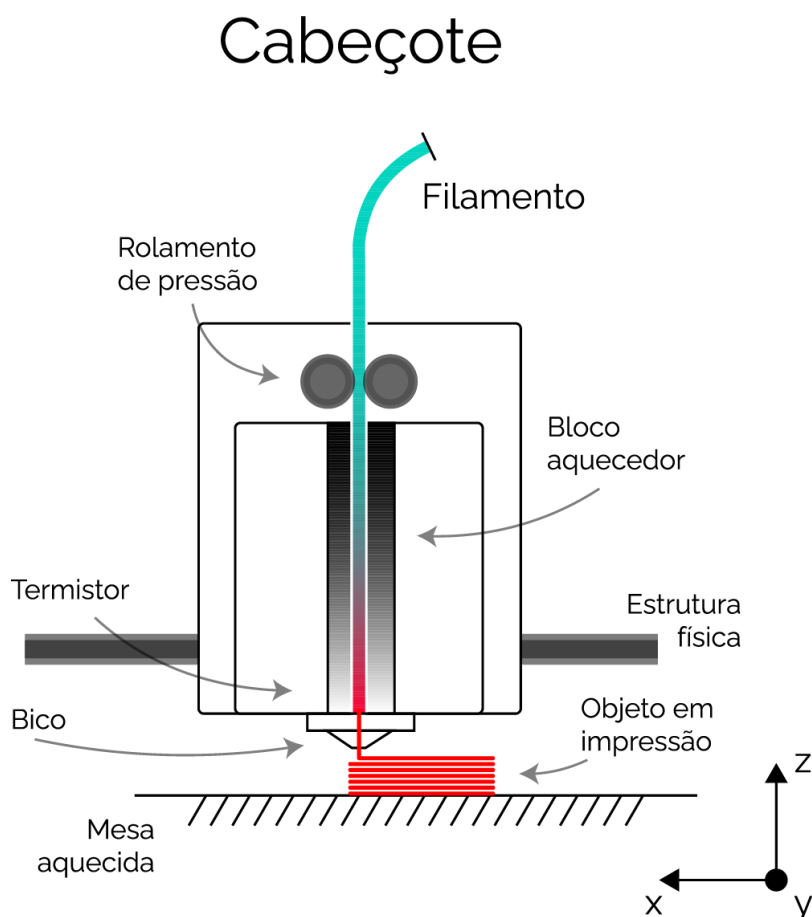
Na técnica de Cura Sólida na Base (SGC), luz ultravioleta é utilizada para curar resina fotossensível, permitindo a cura de uma camada inteira ao mesmo tempo. Isso é feito ao projetar uma máscara que define as áreas de cura, localizada entre a lâmpada UV e a plataforma de construção. A depender do modelo do equipamento, o material não curado pode ser aspirado e substituído por cera, que age como suporte. e é posteriormente removida com solvente. A superfície é fresada antes de cada nova camada. Essa técnica permite construções rápidas e a fabricação de protótipos de grande porte, sendo eficiente para aplicações que exigem agilidade e escala (BARROS, 2012).

A Conformação Próxima ao Formato Final Via Laser (LENS) configura um sistema que utiliza um laser de alta potência para fundir pó metálico diretamente na superfície de construção, com um bico que deposita o material e se movimenta em três eixos, permitindo a criação de peças complexas sem necessidade de suportes. A mesa de suporte também é rotativa, ampliando a versatilidade do processo. Embora tenha baixa resolução e requeira acabamento posterior, a técnica permite a fabricação e reparo de peças metálicas de alta qualidade mecânica. É capaz de produzir protótipos densos (BARROS, 2012).

A impressão eletrohidrodinâmica (EHD) utiliza forças eletrostáticas para manipular partículas ou fibras, criando estruturas micro e nanoescalares de alta precisão, com resolução de até 100 nm. Esse sistema é composto por uma bomba de seringa, bico metálico, controlador de alta voltagem, e tinta imprimível. O campo elétrico entre o bico e o substrato forma um “cone de Taylor” na tinta, ejetando gotas

ao superar as forças de tensão superficial e viscosidade. A EHD permite controle preciso de modos de jateamento, incluindo gotejamento e injeção pulsante, úteis em dispositivos como transistores e sensores flexíveis.

Figura 1 - Esquema de impressora 3D por FDM e suas estruturas básicas



Fonte: Adaptado de HUTMACHER, D. (2001)

O presente trabalho tem como escopo a técnica manufatura aditiva por modelagem de deposição fundida (Fused Deposition Modeling - FDM) que é aquela caracterizada pela deposição organizada de filamento de um material fundido numa superfície. A impressão 3D por FDM constrói o objeto camada por camada, a partir de um projeto digital, sem desperdício significativo. A popularidade dessa tecnologia se deve ao desenvolvimento e à disseminação das próprias impressoras 3D, que transformaram o cenário industrial e abriram novas possibilidades para a prototipagem rápida e a criação de uma comunidade. Também é importante citar sua capacidade de prototipar de maneira rápida objetos complexos com um alto grau de personalização e precisão, tornando o processo altamente disseminado (BARROS, 2012).

Diversos trabalhos mostram a importância dessa técnica de impressão 3D. Segundo Gutemberg, a FDM é considerada a mais vantajosa devido aos baixos custos de implementação e manutenção, resultando em um custo por protótipo reduzido. Embora tenha um tempo de construção mais longo, seu baixo ruído, mínimo desperdício de material e capacidade limitada de tamanho a tornam ideal para ambientes não-industriais (BARROS, 2024). Sua acessibilidade, tanto em termos de custo quanto de usabilidade, tornaram a FDM a modalidade mais utilizada pela comunidade científica. Aplicações na área do ensino refletem a importância da modalidade, difundindo-se em todos os níveis educacionais, perpassando do ensino básico ao ensino superior (MARTINS et al., 2021). Na área de medicina, se destaca pela capacidade de criar objetos de alta fidelidade a em escalas minúsculas (HUTMACHER, 2001).

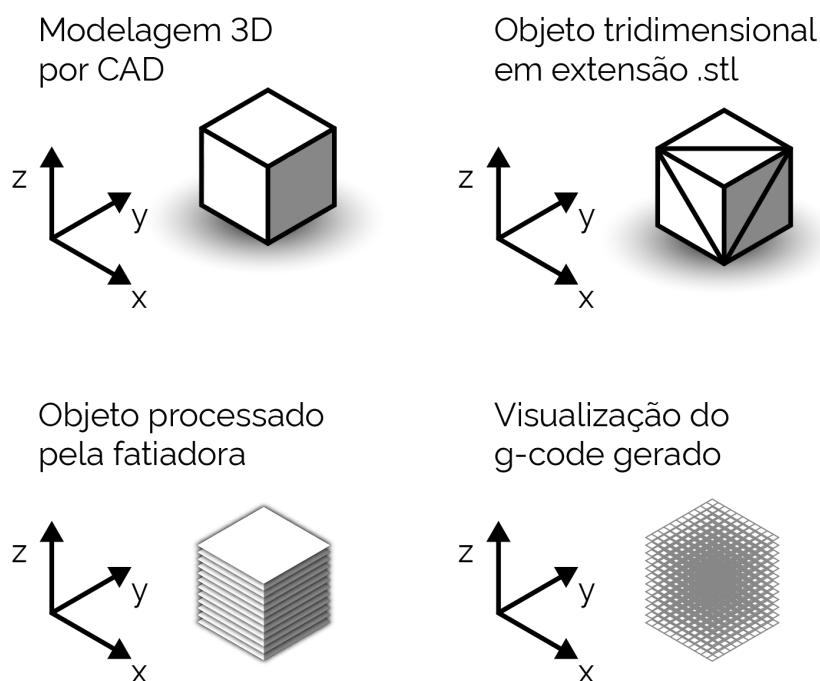
Ela tem sido amplamente utilizada em pesquisas de engenharia para a produção de elementos de concreto com formas complexas. O uso de FDM permite testar a viabilidade de novos materiais sustentáveis, como geopolímeros ativados por álcali (AAMs), que oferecem uma alternativa ao cimento Portland comum (OPC). Nas pesquisas, a FDM facilita a criação de geopolímeros com ativadores sólidos, simplificando a mistura e reduzindo o impacto ambiental. Assim, a engenharia utiliza a FDM para desenvolver soluções de construção mais ecológicas e eficientes em larga escala (AL-NOAIMAT, 2023). Já na área do ensino, modelos impressos por FDM são utilizados como material didático ao mesmo tempo que a modelagem 3D funciona como meio de aprendizado. A impressão 3D tem sido aplicada ao ensino de embriologia, superando limitações das aulas expositivas e imagens bidimensionais. Um estudo mostra que utilizar modelos 3D de estruturas embrionárias numa aula de biologia melhora o aprendizado e superação de dificuldades epistemológicas, ou seja, foi um facilitador no que diz respeito a forma que o indivíduo lida com o conteúdo (RONCAGLIO, 2020). Um outro trabalho destaca como a Geometria Gráfica, aliada à cultura maker e à impressão 3D, pode estimular habilidades visuais e cognitivas em alunos do Ensino Fundamental e Superior (SCANONI, 2023).

Além dos materiais disponíveis para a impressão 3D por FDM já difundidos, como os termoplásticos acrilonitrila butadieno estireno (ABS) e ácido polilático (PLA) (Reprap.org) existem outros diferentes insumos, como o politereftalato de

etileno (PET) e o náilon (*Nylon*). O poliuretano Termoplástico (TPU) oferece a característica de ser flexível, possibilitando outras aplicações. Acetato de Polivinilo (PVA) é um material solúvel e que pode ser usado na impressão 3D, para construção de suportes pro objeto principal. PP (polipropileno), PC (policarbonato), Nylon e CPE (copoliéster) (WISHBOX, 2020). Naturalmente, o campo da Engenharia de Materiais inova na área ao associar outros materiais aos termoplásticos, como a fibra de carbono e madeira (DAS, 2021), conferindo novas propriedades na manufatura. Ligas metálicas, cerâmica e concretos também constituem o grupo de materiais utilizados na técnica de impressão 3D por FDM (NGO, 2018).

A evolução da impressão 3D por FMD se beneficia na convergência entre hardware e software acessíveis e colaborativos, com destaque para o Arduino, código aberto das fatiadoras e softwares de modelagem gratuitos como Blender. O ambiente econômico e técnico foi favorecido pela expiração de patentes e a ascensão de plataformas de financiamento coletivo, permitindo que pequenos inventores levassem impressoras 3D ao mercado com pouco capital. O Arduino, lançado em 2005 como um microcontrolador de baixo custo e fácil de programar, forneceu a base de hardware para muitos sistemas de controle de impressoras 3D. Combinado com uma vasta gama de sensores e motores, ele permite a coordenação precisa dos movimentos e a deposição de material. Paralelamente, a introdução de repositórios de código aberto como o GitHub em 2008 possibilitou uma colaboração massiva e contínua entre desenvolvedores. O firmware que controla o hardware, assim como os *softwares* fatiadores tornaram-se amplamente acessíveis e aprimorados pela comunidade (HORVATH, 2014).

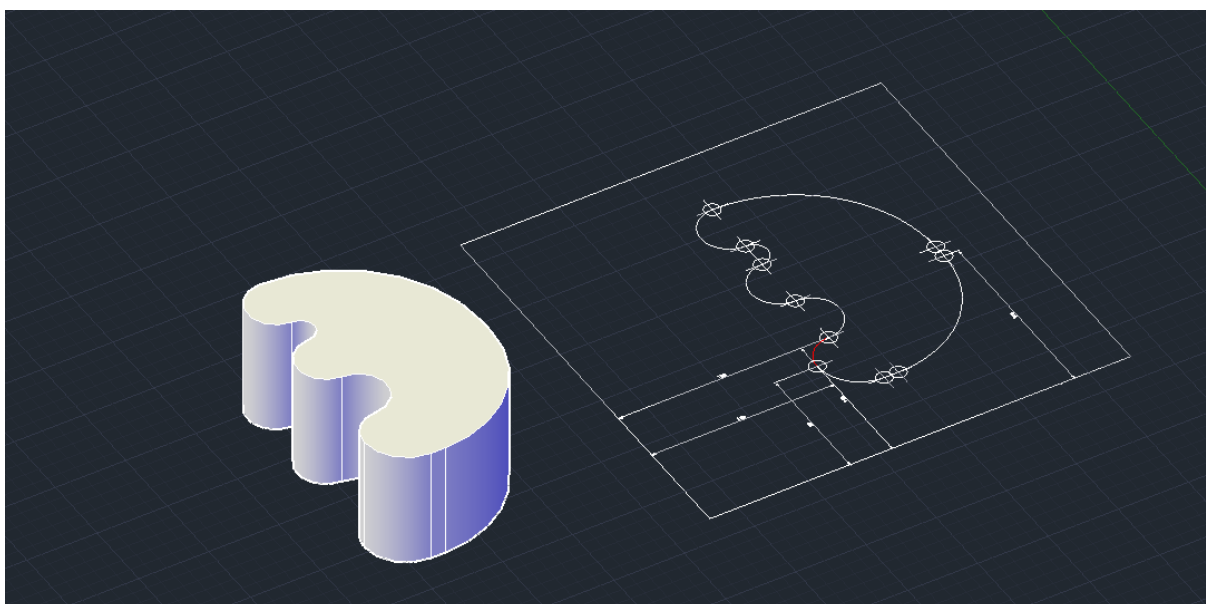
Figura 2 - Esquema dos processamentos gráficos que um modelo tridimensional sofre durante o pipeline da impressão 3D



Fonte: Adaptado de HUTMACHER, D. (2001)

A impressão 3D por FMD é organizada num pipeline muito específico, que inclusive pode ser observado separadamente em cada uma de suas etapas. Tais etapas são divididas em: a) modelagem 3d; b) geração de código de impressão; c) impressão 3D. A etapa de modelagem tridimensional envolve processos criativos, de pesquisa, usabilidade, engenharia, ergonomia, e de pré-projeto muito específicos para cada caso. Esboços, modelagem manual, projetos de desenho assistido por computador (*computer-aided design*, i.e., CAD) definem métodos de desenvolvimento do design do objeto (LIVESU, 2017).

Figura 3 - Sólido 3D no CAD e a curva de perfilamento geradora da forma

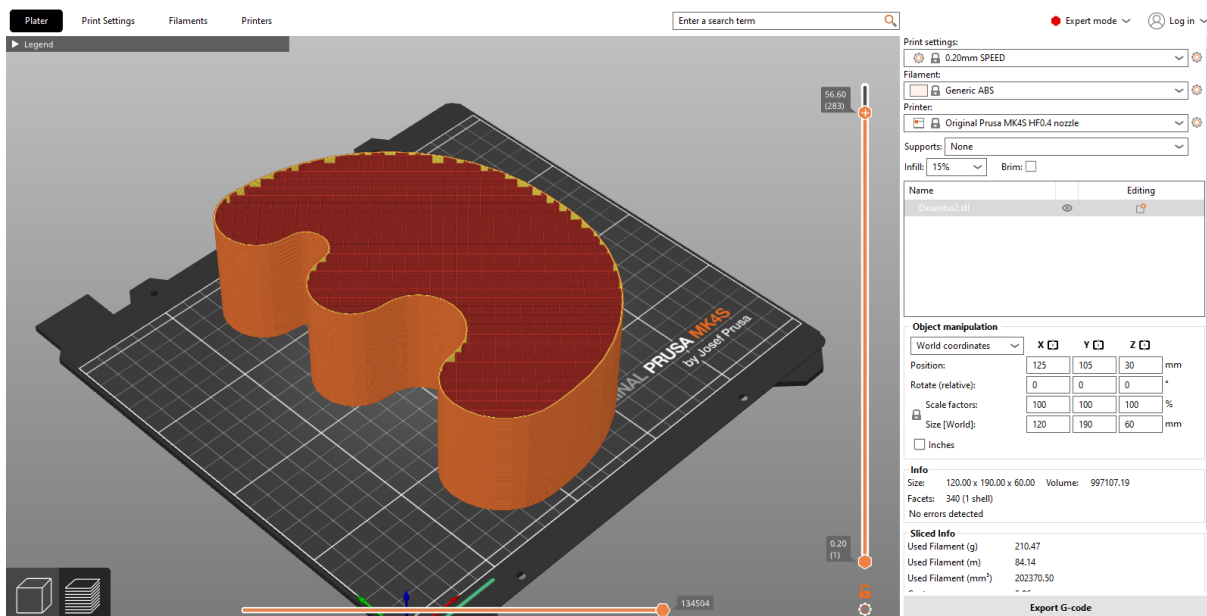


Fonte: Elaborado pelo autor (2024)

Como exemplo, a impressão de uma prótese envolveria todos esses e outros processos muito mais específicos da área e da natureza do modelo tridimensional. Obrigatoriamente, no pipeline difundido na atualidade, o modelo tridimensional desenvolvido em CAD deverá ser convertido num arquivo de extensão .STL para a geração do G-code por software de fatiamento.

A geração do G-code, ou seja, geração de código de impressão, consiste na utilização de software apropriado que consiga calcular o trajeto preciso da impressão do objeto que foi modelado no computador (PANDEY, 2003). A técnica utilizada por tais programas é o cálculo de múltiplas interseções pré-determinadas entre o objeto 3D e o plano xy , a fim de resultar em pontos amostrados em cada uma das interseções. Se o usuário desejar, por exemplo, que cada camada do objeto final tenha 0,3mm, o software irá operar interseções digitais em intervalos de 0,3mm.

Figura 4 - Visualização do G-code no PrusaSlicer



Nota: Em laranja claro, o G-code referente à casca; em laranja escuro, o preenchimento. Em amarelo, a camada interna da casca.

Fonte: Elaborado pelo autor (2024)

Após as interseções, o software cria caminhos ordenados a partir dessa série de conjunto de pontos na linguagem de programação G-code. Esta etapa do pipeline também envolve decisões importantes: além da espessura das camadas do objeto, é possível escolher diferentes características da impressão: tipo e densidade de preenchimento interno, espessura da casca, tampas e base do objeto e a possibilidade de usar suportes de impressão. A depender da aplicabilidade e objetivo da impressão 3D, tais parâmetros podem auxiliar o usuário a ter um produto que atenda às suas necessidades imediatas (PANDEY, 2003).

G-code é uma linguagem de programação utilizada para controlar máquinas CNC (*Computer Numerical Control*) e impressoras 3D. Originalmente desenvolvida para fresadoras na indústria, ela permite a automação precisa de processos de manufatura. O G-code é composto por comandos que instruem a máquina sobre como se mover, em que velocidade operar e quais trajetórias seguir, seja para cortar materiais, como em tornos e fresadoras, ou para depositar material, camada por camada, como nas impressoras 3D. Além disso, ele controla parâmetros essenciais, como temperaturas da base e do cabeçote de impressão, velocidades de movimentação dos motores e posicionamento nas coordenadas xyz.

Snippet de código 1 - G-code com apenas comandos G1 (segmentos de reta)

```
G1 X95.383 Y62.176 E.1769  
G1 X95.245 Y62.097 E.00538  
G1 X90.416 Y60.096 E.17693  
G1 X90.262 Y60.055 E.00539  
G1 X79.989 Y58.703 E.35073  
G1 X75.319 Y56.768 E.17111  
G1 X71.309 Y53.691 E.17109  
G1 X68.232 Y49.681 E.17109  
G1 X66.297 Y45.011 E.17111  
G1 X65.637 Y39.994 E.17128  
G1 X66.372 Y33.466 E.22236  
G1 X68.544 Y27.259 E.22259
```

G-code tem uma estrutura simples e consistente. A depender do *flavour* do G-code, ou seja, do *firmware*, os elementos que compõem a linha de código podem variar. O exemplo acima mostra o modelo "G# X# Y# E#" de comando, coordenadas e velocidade. O comando inicial, representado por "G#", define a ação. O comando G00 instrui a máquina a se mover rapidamente, na velocidade máxima, de sua posição atual até um ponto especificado pelas coordenadas do comando, resultando em um deslocamento em linha reta. É um movimento não-cortante, usado para posicionar rapidamente a máquina na posição inicial de um trabalho, otimizando tempo e eficiência. O comando G01 instrui a máquina a se mover em linha reta com uma velocidade específica, definida pelo valor F, até as coordenadas finais indicadas por X, Y e Z. Diferente do G00, usado apenas para posicionamento, o G01 realiza o trabalho principal da máquina, que, no nosso caso, é extrudar material em impressoras 3D.

O comando G02 instrui a máquina a realizar um movimento circular no sentido horário, conectando dois pontos através de um arco. Este comando exige informações adicionais, como o centro da rotação, para calcular a trajetória do arco. O ponto inicial do movimento circular é definido pela posição final do comando anterior ou pela posição atual da máquina, necessitando somente a informação do ponto final do arco. A trajetória é subordinada aos pontos e ao raio do arco. Se o centro estiver mais distante do ponto inicial, o arco terá uma pequena abertura; por outro lado, se o centro estiver mais próximo, o arco será mais fechado.

O comando G03, semelhante ao G02, instrui a máquina a realizar movimentos circulares, mas no sentido anti-horário. Assim como no G02, o movimento conecta dois pontos por meio de um arco, sendo necessário definir o centro de rotação para calcular a trajetória. Novamente, o ponto inicial é determinado pela posição atual da máquina ou pela posição final do comando anterior.

Combinando os comandos G01, G02 e G03, é possível gerar trajetórias precisas para criar formas complexas. Ferramentas como software CAM facilitam esse processo, otimizando a programação de movimentos, mesmo para configurações complexas, com alta segurança e eficiência. A seguir, mostraremos uma tabela com outros comandos presentes:

Tabela 1 - Códigos G e M para G-code

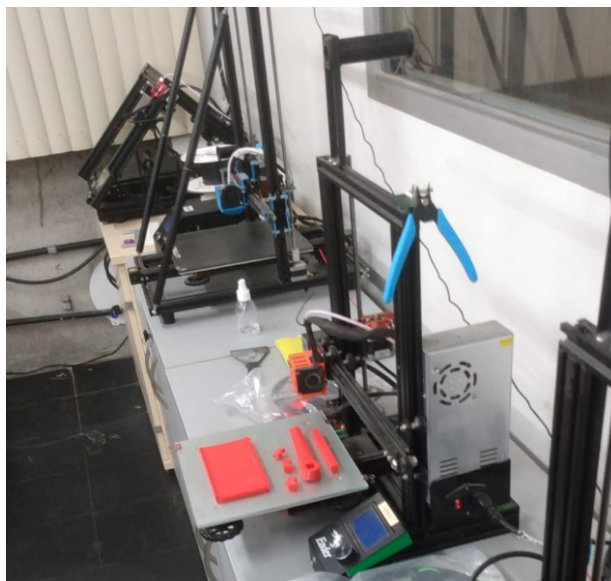
Código	Descrição
G20/G21	Seleciona as unidades: G20 para polegadas, G21 para milímetros.
G17/G18/G19	Seleciona o plano de trabalho: G17 para XY, G18 para XZ, G19 para YZ.
G28	Movimenta a ferramenta para a posição de referência (home).
G90/G91	Define o posicionamento: G90 para modo absoluto, G91 para modo relativo.
G00	Posicionamento a velocidade máxima
G01	Interpolação linear
G02	Interpolação circular (sentido horário)
G03	Interpolação circular (sentido anti-horário)
M00	Parada do programa.
M02	Fim do programa.
M03	Liga o spindle no sentido horário.
M04	Liga o spindle no sentido anti-horário.
M05	Desliga o spindle.
M06	Troca de ferramenta.
M08	Liga o resfriamento por fluido.
M09	Desliga o resfriamento por fluido.
M30	Fim do programa e volta ao início.
M104	Inicia o aquecimento do extrusor (impressoras 3D).
M109	Espera até o extrusor atingir a temperatura desejada.
M140	Inicia o aquecimento da base (impressoras 3D).
M190	Espera até a base atingir a temperatura desejada.
M106	Define a velocidade do ventilador (impressoras 3D).

Fonte: Baseado em SMID, P. (2008)

É importante notar que M03, M04 e M05 não se aplicam às impressoras 3D, por serem utilizados para acionar fresadoras e outras ferramentas CNC. Esses comandos são sucedidos pelas coordenadas de destino, indicadas pelos valores de X, Y e Z. F## define a velocidade do movimento, e é conhecido como taxa de alimentação. Por exemplo, a linha G01 X247.951560 Y11.817060 Z-1.000000 F400 instrui a máquina a mover-se em linha reta até as coordenadas X= 247.951560, Y= 11.817060 e Z= -1.000000, com uma velocidade de 400 mm/min. Este movimento é

regido por configurações prévias no G-code, como o comando G21, que define as unidades em milímetros. Se o comando G20 fosse usado, as unidades seriam alteradas para polegadas. Assim, o G-code traduz conceitos geométricos em movimentos precisos, garantindo que a impressora 3D execute suas funções de forma eficiente e exata (WATER, 2024).

Figura 5 - Impressora 3D em estado finalizado de uma impressão.



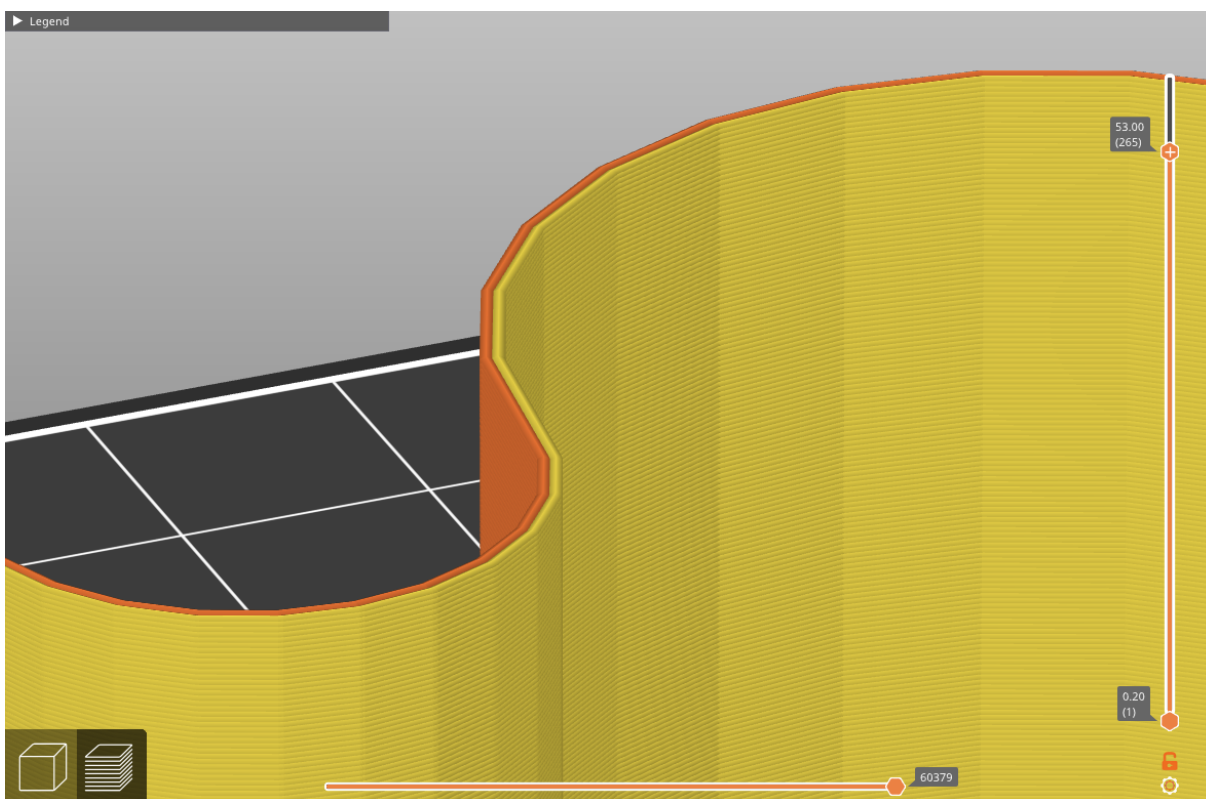
Fonte: Elaborado pelo autor (2024)

A próxima grande etapa do pipeline é a impressão 3D, caracterizada pela interpretação do G-code pela impressora 3D. Algumas impressoras oferecem opções extra, como pré-aquecimento do extrusor ou da mesa. Calibragem nos eixos, opções de inserção, remoção ou troca de filamento, organização de filamentos em impressões multimateriais (GOH, 2021) são outras opções que o usuário pode usufruir (BELL, 2014). Ao final das opções extra de impressão a máquina irá performar o G-code, obtendo-se o objeto materializado. Assim, o usuário poderá utilizar o objeto como produto final ou realizar os devidos testes e experimentos.

O processo de impressão 3D é baseado na deposição de material fundido camada por camada, seguindo trajetórias definidas por um código de controle numérico conhecido como G-code. Na maioria das impressoras 3D, o G-code utilizado descreve o movimento do cabeçote de impressão por meio de segmentos de reta entre pontos amostrados ao longo da curva gerada na fase de fatiamento do modelo. Esse método de interpolação retilínea entre os pontos amostrados pode

resultar em um objeto impresso com superfícies angulosas, pois a representação de uma curva suave por segmentos retos causa discontinuidades visuais, especialmente quando a densidade de pontos não é suficientemente alta. Isso é particularmente problemático em modelos com superfícies curvas complexas, de variada curvatura, onde o aumento na densidade de amostragem para melhorar a qualidade resulta em maiores tempos de processamento e de impressão. Isso representa um desafio em modelos de geometria curva, onde a suavidade e definição da superfície é essencial para o desempenho mecânico e a estética dos objetos impressos. Além da tesselação gerada ao transformar um arquivo CAD em .STL, causando perda de resolução do projeto, sabemos que os conjuntos de pontos gerados serão performados como segmentos de retas. Isso cria um acúmulo considerável de distorções do que idealmente deveria ser o objeto.

Figura 6 - Trajetos de G-code em segmentos de reta



Fonte: Elaborado pelo autor (2024)

A questão da tesselação, fundamental no formato STL utilizado na impressão 3D, também apresenta desafios significativos, especialmente em superfícies curvas, onde a aproximação por triângulos planos pode comprometer a precisão do modelo final (OROPALLO, 2016). Mitigar essas limitações é crucial, dado o amplo uso da

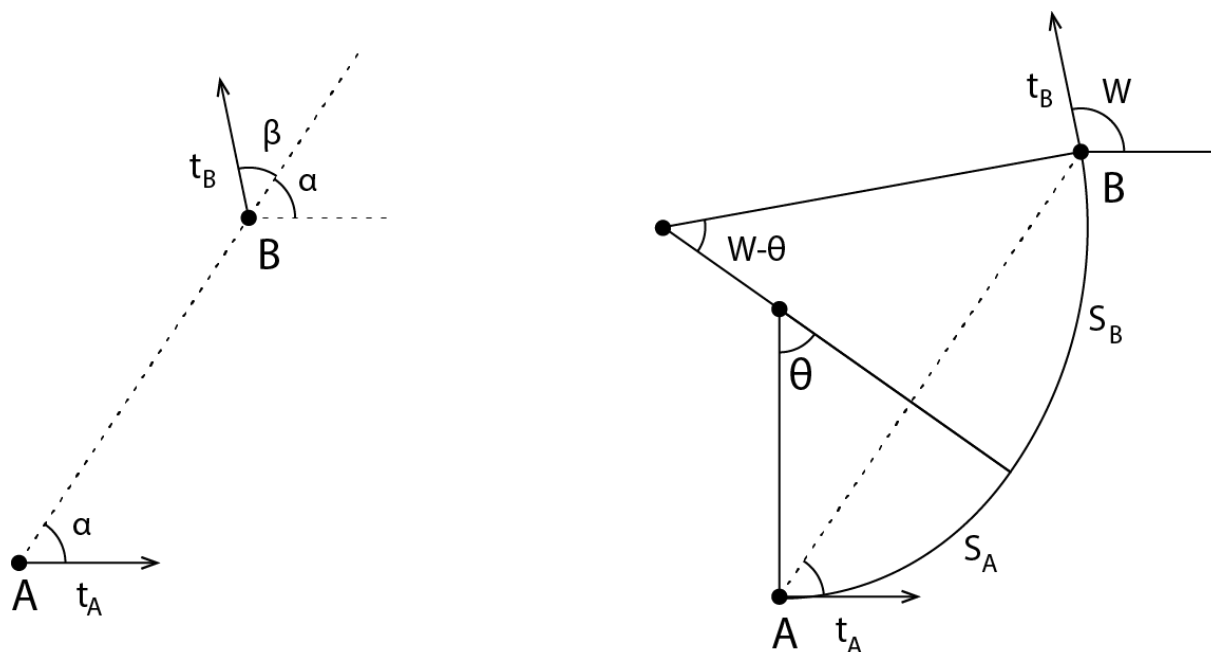
impressão 3D em áreas como medicina, onde próteses e órgãos impressos exigem alta precisão; ensino, para modelos didáticos detalhados e não-ambíguos; e engenharia, para componentes complexos e funcionais.

No entanto, é sabido que no G-code existem comandos específicos para criação de arcos de circunferências utilizando os seguintes dados de entrada: ponto inicial, ponto final e centro da circunferência (WATER, 2024).

A exigência de uma constante evolução no processo de impressão 3D sugere que pesquisadores desenvolvam novas técnicas que busquem mitigar os problemas que surgem no pipeline da impressão 3D (WANG, 2016). Numa proposta experimental, um estudo apresentou um conjunto de algoritmos que gera partes separadas de um modelo 3D de entrada. Cada parte pode ser orientada para impressão de forma que as normais da superfície externa estejam separadas da direção de impressão. Esse método evita a formação de artefatos em degraus, eliminando a necessidade de estruturas de suporte na superfície externa. No entanto, o problema da aparência de tesselação em impressões 3D por FDM necessita de mais pesquisas (WANG, 2016).

Com isso, uma solução proposta para esse problema envolve a substituição de segmentos de reta por arcos de circunferência no G-code, uma vez que a maioria das impressoras 3D possui comandos nativos para interpolação de arcos (SÁNCHEZ-REYES, 2022). No entanto, o uso de arcos simples não resolve completamente o problema, uma vez que a junção inadequada entre dois arcos pode resultar em descontinuidades tangenciais (cúspides), que também afetam a qualidade da superfície impressa. Para resolver essa questão, foi desenvolvido um método que interpola os pontos amostrados utilizando splines G^1 de biarcos – uma técnica que apresenta graus de liberdade suficientes para conectar de forma suave cada par de arcos consecutivos, respeitando as tangentes nos pontos de junção (GREINER; HORMANN, 1997).

Figura 7 - Dados associados às tangentes t_B e t_A ; exemplo de biarco que une o ponto A ao B e que concorda com as tangentes t_B e t_A .



Fonte: Adaptado de MEEK, D. S. e WALTON, D. J. (1991)

Dado que a entrada do método é formada apenas por pontos amostrados, as tangentes precisam ser inferidas, neste caso, utilizando-se uma técnica aproximativa. Esse método promete melhorar significativamente a qualidade das superfícies impressas, sem a necessidade de aumentar a densidade de amostragem, o que torna o processo de fatiamento mais eficiente.

1.2 Problema de pesquisa

Na impressão 3D, a aparência final do objeto frequentemente apresenta um aspecto de mosaico ou estriado, devido à natureza da fabricação camada a camada e à geometria do modelo digital que serve como base para a impressão. Esse efeito se dá porque o processo construtivo utiliza finas camadas sobrepostas de material, formando o objeto tridimensional gradualmente. A resolução de uma impressora 3D, ou seja, a espessura mínima de camada que ela consegue depositar, influencia diretamente o acabamento superficial e o detalhamento da peça. Objetos com curvas ou superfícies inclinadas, por exemplo, tendem a mostrar mais claramente essa aparência segmentada, pois as camadas sucessivas têm que “acompanhar” a

geometria curva do modelo, gerando um efeito de degraus. A estratégia de fatiamento do modelo, definida pelo software, e a orientação do objeto durante a impressão também impactam esse resultado visual. Tecnologias de pós-processamento, como polimento ou tratamento térmico, são muitas vezes aplicadas para melhorar o acabamento e reduzir a rugosidade da superfície. Essas práticas são essenciais para aprimorar a estética de peças impressas, especialmente em aplicações onde a aparência visual e o toque suave são requisitos importantes, como em prototipagem de produtos finais e peças decorativas (WU, 2018).

O principal problema desta pesquisa é a perda de qualidade nas trajetórias definidas pelo G-code, no que diz respeito à tesselação e o uso de segmentos de reta nos arquivos de impressão 3D. Os objetos impressos em 3D, por utilizarem de segmentos de reta entre pontos amostrados no G-code, ficam limitados quanto à aparência final, requerendo um tamanho de arquivo com muitos polígonos para obter-se uma qualidade adequada. Mesmo com a adoção de arcos de circunferência, a junção inadequada entre os arcos pode causar descontinuidades tangenciais, resultando em superfícies de baixa qualidade.

A questão central é: Como uma proposta algorítmica pode contribuir para a melhoria no pipeline da impressão 3D por FDM no que diz respeito ao melhoramento dos códigos G-code?

1.3 Hipótese

A hipótese desta pesquisa é que a substituição dos segmentos retilíneos convencionais no G-code por splines G^1 de biarcos melhorará a resolução da trajetória, contribuindo no desenvolvimento do estado da arte da impressão 3D, sem aumentar significativamente o tempo de processamento do fatiamento. A técnica de biarcos preserva a continuidade tangencial entre as curvas, eliminando o efeito de tesselação (como cúspides) e proporcionando trajetórias contínuas e concordantes.

1.4 Objetivos

1.4.1 Objetivo Geral

Investigar experimentalmente o impacto do uso de splines G^1 de biarcos na qualidade das superfícies de objetos de geometria complexa impressos em 3D, comparando-os com o método tradicional de interpolação por segmentos de reta, com ênfase na suavidade das superfícies e no tempo de processamento do G-code.

1.4.2 Objetivos Específicos

1. Implementar um algoritmo para extrair de um arquivo-texto com G-code de um objeto, os pontos amostrados por fatia.
2. Implementar um dos algoritmos de extração de tangentes sugeridos por MEEK, D. S. e WALTON, D. J. (1991).
3. Implementar o algoritmo Meek-Walton [4], a partir dos pontos amostrados e suas tangentes, produzir uma sequência de biarcos que aproxima a curva de fatiamento da qual os pontos foram amostrados, e gerando como saída os comandos G-code correspondentes a arcos circulares.
5. Avaliar e comparar os caminhos de G-code do objeto tridimensional original.
6. Analisar o impacto do uso de splines G^1 de biarcos no tempo de processamento do fatiamento.

1.5 Metodologia

O processo de avaliação da proposta do presente trabalho consiste na comparação entre resultados provenientes de dois arquivos G-code. Com isso, a metodologia se divide em diferentes etapas. A primeira trata da criação do modelo 3D que atenda os requisitos do experimento, seguido do fatiamento no software PrusaSlicer para geração do G-code base. Posteriormente, haverá o processamento de tal G-code num algoritmo que transforma as trajetórias de segmentos de reta em

trajetórias em biarcos e, por fim, a avaliação deste novo G-code e da performance geral do algoritmo

O escopo para criação do objeto tridimensional a ser modelado é aqui descrito como um sólido pouco convencional, gerado por uma curva fechada, não-convexo e que dispensa a necessidade de suportes físicos no processo da impressão 3D. Naturalmente, a partir do que foi estabelecido, o objeto deverá possuir pelo menos uma superfície curva. O objeto deverá ser uma extrusão reta (em 90°) de tal curva fechada. Sendo assim, estão dispensados do experimento os poliedros regulares e irregulares, esfera e quaisquer elipsóides, superfícies derivadas de curvas cônicas e planos e retas. Sobre o resultado do processo de fatiamento de tal sólido, as interseções deverão resultar em curvas fechadas não-convexas em pelo menos uma interseção. Para modelar o sólido, será utilizado o AutoCAD. A construção será feita utilizando comandos de construção de circunferências, retas e extrusão do perfil num objeto tridimensional. Para a impressão 3D, o modelo tridimensional é exportado, obrigatoriamente, no formato STL.

O software utilizado para geração do G-code será o PrusaSlicer (versão 2.8.1), que é próprio para o método de impressão 3D por FDM, utilizando os parâmetros padrão de fatiamento. O arquivo a ser utilizado é de extensão GCODE (.gcode), diferente do output mais utilizado pelo software que é o BINARY GCODE (.bgcode), por ter implementação mais simples, com informações facilmente extraídas do arquivo base e, conseqüentemente, ser mais simplificada sua inserção no algoritmo proposto.

O software PrusaSlicer utilizará seu próprio algoritmo para para geração dos caminhos de impressão 3D. Tal algoritmo consiste em cálculos de interseção entre o plano xy e o objeto 3D, resultando numa série de pontos amostrados. Estes pontos registram literalmente a direção e a trajetória que o bico extrusor perfumaria na mesa da impressora 3D, obrigatoriamente em trajetórias descritas por segmentos de retas ligadas ponto a ponto.

O algoritmo de fatiamento do software, neste caso, é subordinado aos parâmetros de impressão, que são diversos. Os parâmetros que correspondem à estrutura física externa do objeto são os de altura da primeira camada de impressão,

altura das demais camadas e espessura das cascas do objeto (casca vertical corresponde à estrutura lateral do objeto e a casca horizontal corresponde às estruturas do(s) cume(s) do objeto). Outra classe de parâmetros são os de velocidades do bico extrusor, temperatura de fusão do insumo. Os parâmetros de preenchimento, ou *infill*, estabelecerão a melhor estrutura e densidade dentro das opções oferecidas pelo software. Parâmetros como Raft, Skirt e Brim Support Material são outras opções que dão um suporte ao objeto. No entanto, nossa proposta não precisará utilizar os parâmetros de suporte basal ou estrutura, já que o objeto 3D será modelado com tal requerimento.

Com o arquivo G-code gerado pelo PrusaSlicer, esse dado será transformado por um algoritmo que será descrito em detalhes posteriormente. O objetivo do script é transformar os segmentos de retas que compõem as trajetórias da casca do objeto em trajetórias de biarcos conectadas perfeitamente. Por utilizar os mesmos pontos amostrados do G-code inicial, esperamos que o aspecto do trajeto do G-code seja semelhante ao original. A avaliação dos resultados se dará por comparação dos trajetos G-code do produto original e aquele já processado pelo algoritmo, observando a melhoria da suavidade e confirmando a utilização de comandos G2 e G3.

1.6 Justificativa

O avanço das técnicas de manufatura aditiva exige soluções que melhorem a qualidade dos objetos impressos sem comprometer a eficiência do processo. Existe uma constante busca no aprimoramento da técnica de impressão 3D por FDM. Usuários por vezes se sentem sobrecarregados com tantos parâmetros para ajustar numa impressão 3D (KRISTIAWAN et al., 2021). A indústria cada vez mais exige um padrão de qualidade em todas as etapas do pipeline de impressão 3D, e não seria diferente para a geração do G-code. O uso de splines G^1 de biarcos oferece uma abordagem promissora para suavizar superfícies curvas e complexas em impressão 3D, otimizando o caminho do cabeçote da impressora. Esta pesquisa se justifica pela necessidade de melhorar a precisão geométrica e a suavidade das superfícies

impressas, que são fundamentais em aplicações industriais e de design de produto, além de reduzir o tempo de processamento do fatiamento e os custos operacionais.

Um trabalho que também aborda a utilização de comandos G2 e G3 apresentou importantes contribuições, destacando a eficácia desses comandos na geração de trajetórias curvas precisas e a materialização de modelos físicos com alta fidelidade geométrica. Essa abordagem revelou a viabilidade de utilizar ferramentas avançadas de CAD e técnicas paramétricas para criar objetos tridimensionais curvos, otimizando o tempo de processamento e reduzindo falhas no acabamento superficial (PENANTE, 2019).

A melhoria da superfície externa em peças impressas em 3D é um tópico amplamente discutido na literatura científica. Diversos estudos propõem métodos para aprimorar o acabamento superficial, como técnicas de pós-processamento, ajustes nos parâmetros de impressão e desenvolvimento de novos materiais. Conforme discutido por Song et al. (2016), a utilização de técnicas como anti-aliasing já mostrou avanços significativos na redução de defeitos de escada em superfícies inclinadas. Tal discussão reforça a importância de métodos que otimizem as trajetórias de impressão, explorando potencialidades de comandos nativos do G-code, como G2 e G3.

Além disso, os benefícios futuros da impressão 3D são amplamente reconhecidos, tanto na customização e prototipagem rápida quanto na sustentabilidade. Segundo Ivkić et al. (2024), a integração da impressão 3D em ecossistemas de manufatura como serviço pode criar cadeias de produção mais ágeis e ambientalmente conscientes. A presente pesquisa se insere nesse contexto ao propor uma técnica que pode ser implementada em pipelines industriais, reduzindo tempos de processamento e aumentando a qualidade final dos produtos.

A constante evolução da manufatura aditiva, especialmente na impressão 3D, exige soluções inovadoras para superar desafios técnicos relacionados à qualidade das superfícies e à eficiência dos processos. No contexto deste trabalho, que explora o uso de splines G^1 de biarcos no G-code como alternativa aos segmentos de reta, busca-se melhorar a suavidade e a precisão geométrica de superfícies curvas complexas, minimizando artefatos e descontinuidades. Essa abordagem

atende a demandas crescentes por excelência na produção industrial, design de produtos e aplicações educacionais e médicas.

Nossa pesquisa se justifica na ampliação dessa perspectiva ao explorar como um objeto modelado com arcos em sentidos opostos, garantindo o uso de comandos G2 e G3, podem expandir ainda mais esse campo. O sólido, projetado para evitar a necessidade de suportes adicionais durante a impressão, maximiza a eficiência do processo. Além disso, o trabalho supracitado reforça a relevância do estudo de tesselação e seus impactos no formato STL, o que conecta diretamente com nossa proposta de analisar como essas técnicas podem beneficiar aplicações da impressão 3D. Assim, este trabalho é fundamentado pela trajetória científica já consolidada, avançando no detalhamento e aplicabilidade de comandos e métodos que ampliam as possibilidades da impressão 3D.

Quanto aos benefícios futuros que as contribuições podem trazer, além dos ganhos estéticos, podemos falar sobre uma maior personalização de produtos, redução de custos de produção e diminuição de resíduos. Além disso, a impressão 3D pode contribuir para cadeias de suprimento mais ágeis e sustentáveis.

Espera-se que a adoção de splines G^1 de biarcos traga contribuições significativas à comunidade científica e tecnológica, não apenas no aprimoramento da qualidade superficial dos objetos impressos, mas também na exploração de novos horizontes para aplicação da impressão 3D, consolidando sua relevância como tecnologia essencial para a inovação em áreas diversas, como engenharia, medicina e educação.

2. REVISÃO DA LITERATURA

2.1 Impressão 3D

A impressão 3D, ou manufatura aditiva, surgiu como uma revolução tecnológica desde sua criação nos anos 1980, permitindo a construção de objetos camada por camada. Inicialmente concebida como estereolitografia (SLA) por Charles W. Hull, inicialmente, como a tecnologia para criação de objetos sólidos camada por camada por meio da solidificação seletiva de uma resina líquida, essa tecnologia evoluiu rapidamente, com métodos como a modelagem por deposição fundida (FDM) consolidando-se no mercado de prototipagem e produção em pequena escala. Gibson et al. (2015) apontam que a popularização da impressão 3D foi impulsionada pela expiração de patentes, tornando hardware e software mais acessíveis. Assim, observa-se que a impressão 3D, embasada por avanços em ciência da computação, revolucionou a fabricação, permitindo uma personalização e precisão antes inviáveis.

Além disso, a integração com a ciência da computação desempenhou um papel crucial, permitindo modelagem complexa e controle preciso dos processos. Gibson (2015) destacou que o uso de algoritmos e ferramentas como CAD (*Computer-Aided Design*) possibilita a conversão de conceitos criativos em modelos tridimensionais funcionais. Com isso, fica claro que a evolução da impressão 3D está ligada ao desenvolvimento computacional.

Em 1989, a tecnologia foi comercializada pela 3D Systems, consolidando o SLA como um método pioneiro de impressão 3D, ainda amplamente usado em grandes máquinas comerciais. Na mesma época, no Instituto de Tecnologia de Massachusetts (MIT) e na Universidade do Texas, outros avanços focavam na sinterização a laser seletiva (SLS), onde lasers eram usados para fundir pó camada a camada. Em 1989, S. Scott e Lisa Crump registraram a patente do *Fused Deposition Modeling* (FDM), também chamado de *Fused Filament Fabrication* (FFF), tecnologia baseada na extrusão controlada de filamento plástico fundido, que deu origem à Stratasys, Ltd. Com a expiração de patentes-chave em 2005, o FDM impulsionou o movimento RepRap (abreviação de “*Replicating Rapid Prototyper*”,

um projeto de impressoras autorreplicadoras) (GUERRERO et al., 2014) e a popularização de impressoras 3D de baixo custo.

Impulsionada por iniciativas colaborativas e pela popularização de plataformas de crowdfunding, como o Kickstarter, lançado em 2009, a impressão 3D experimenta um momento em que ela deixa de ser uma área somente dominada pela indústria e passa a ser apropriada também por uma comunidade de desenvolvedores. A partir desse ano, o desenvolvimento de impressoras 3D dividiu-se em dois grandes grupos: empresas focadas em impressoras industriais proprietárias e uma ampla rede de entusiastas trabalhando em impressoras de código aberto, principalmente no contexto do movimento RepRap. Com o surgimento do Kickstarter, pequenos empreendedores e desenvolvedores independentes encontraram uma forma de viabilizar projetos de impressão 3D, angariando fundos de doadores e oferecendo protótipos ou impressoras finalizadas como recompensa aos apoiadores. Além de possibilitar a democratização da impressão 3D, o *crowdfunding* incentivou a inovação, permitindo que diversos projetos de impressoras baseadas em *Fused Filament Fabrication* (FFF) ou em tecnologias emergentes atingissem o mercado.

Essa prática ajudou a estabelecer um ambiente competitivo e dinâmico para a impressão 3D de baixo custo, levando à rápida evolução da tecnologia e ao surgimento de novas possibilidades na fabricação digital e prototipagem rápida (HORVATH, 2014) através do campo da Ciência da Computação.

Goh (2020) revisou a aplicação de aprendizado de máquina (ML) em diversos aspectos da AM, como design, otimização de processos, monitoramento in situ, ajuste de materiais e cibersegurança. Técnicas como redes neurais artificiais (ANN) e convolucionais (CNN) demonstram eficácia em tarefas como simulações numéricas, detecção de anomalias e recomendações de design. Desafios incluem custos computacionais elevados, dificuldades na padronização de dados e a necessidade de conjuntos de dados grandes e bem rotulados. Técnicas de aprendizado não supervisionado e por reforço são destacadas como promissoras para monitoramento em tempo real sem necessidade de dados rotulados.

A aplicação de inteligência artificial (IA) na manufatura aditiva (AM) tem demonstrado grande potencial para otimizar processos e aprimorar a qualidade de produtos fabricados. Ciccone (2023) revisa sistematicamente 48 artigos para explorar como algoritmos de IA estão sendo usados para superar desafios da AM, como, por exemplo, na otimização de parâmetros, previsão de propriedades de materiais, monitoramento em tempo real e redução de custos. As tecnologias de AM, como fusão em leito de pó (PBF) e FDM, são frequentemente abordadas, devido aos altos custos operacionais e necessidades de precisão. Modelos de ML e aprendizado profundo (DL), incluindo redes neurais artificiais (ANNs) e redes convolucionais (CNNs), têm sido aplicados com sucesso para prever falhas, otimizar trajetórias de impressão e simular processos. O estudo também destaca as lacunas tecnológicas e sugere futuras direções de pesquisa, como a integração de dados para criar soluções em tempo real. Essa revisão fornece uma visão abrangente das possibilidades da IA em AM, enfatizando como essas ferramentas podem tornar os processos mais eficientes, reduzir desperdícios e permitir a criação de produtos com geometrias complexas e alto desempenho, impactando significativamente setores como aeroespacial, automotivo e biomédico Wang et al. (2020) propuseram uma abordagem baseada em IA para identificar e quantificar deformações térmicas em peças impressas, permitindo compensações durante a fabricação e melhorando a qualidade final dos produtos. Essa metodologia utiliza dados escaneados e modelos de referência para prever distorções, demonstrando a capacidade da IA em resolver desafios complexos na manufatura aditiva.

Em suma, a evolução da impressão 3D está intrinsecamente ligada aos avanços da ciência da computação e, mais recentemente, à incorporação de técnicas de inteligência artificial. Essas tecnologias têm permitido melhorias significativas na precisão, eficiência e qualidade dos produtos fabricados, além de viabilizar soluções inovadoras para desafios complexos na manufatura aditiva.

Outros avanços recentes incluem estudos de materiais. A exemplo da polimerização de dois fótons para impressão em nível molecular e o *Contour Crafting*, técnica de impressão em larga escala em concreto (LYU et al., 2021). A evolução constante da impressão 3D reflete a rápida e diversificada inovação do setor, abrangendo desde aplicações industriais até a biomedicina (HORVATH, 2014).

IA auxiliou uma pesquisa na elaboração de métodos de desenvolvimento de biocompósitos (VERMA, 2021). Zhu (2021) mostra suas contribuições combinando impressão 3D e IA, com foco na personalização de dispositivos biomédicos e manufatura inteligente.

2.2 G-code

O G-code, originalmente criado para máquinas CNC, também desempenha um papel essencial na impressão 3D, controlando o movimento do cabeçote de impressão. Smid (2008), em seu extenso e completo manual, explica que o G-code é estruturado em comandos como G01 (movimento linear) e G02/G03 (movimentos circulares), o que possibilita a execução de trajetórias complexas. Isso é particularmente relevante em impressoras 3D, que transformam comandos digitais em movimentos precisos.

Penantes (2017) explora a manipulação de comandos como G02 e G03 para criar arcos paramétricos, demonstrando que ajustes no G-code podem melhorar a qualidade das superfícies impressas. Ele afirma: “A manipulação dos comandos G2 e G3 no G-code permite a geração precisa de arcos paramétricos em impressões 3D, ampliando as capacidades de fabricação de geometrias complexas” (PENANTES, 2017, p. 30). Essa flexibilidade destaca o G-code como um componente crítico na otimização do pipeline da impressão 3D.

O trabalho de Jignasu et al. (2023) explora o potencial de Grande Modelos de Linguagem (LLMs) na compreensão, depuração e manipulação de G-code. Ele apresenta a primeira avaliação sistemática de modelos avançados, como GPT-4 e outros, para tarefas que incluem correção de erros, transformações geométricas e entendimento completo de arquivos G-code. Além disso, os autores introduzem técnicas de “*prompt engineering*” para lidar com os desafios de interpretar e processar essa linguagem de baixo nível, destacando a capacidade dos LLMs de raciocinar sobre geometrias e realizar modificações precisas em G-code. A pesquisa enfatiza como esses modelos podem otimizar processos de fabricação aditiva, permitindo uma maior automação no fluxo de trabalho, desde o design até o monitoramento em tempo real. Entre as contribuições, estão estratégias práticas para aplicar LLMs em ambientes industriais, demonstrando sua aplicabilidade na

redução de erros e no aumento da eficiência em operações de impressão 3D. Este trabalho ressalta os avanços no uso de inteligência artificial para a manufatura aditiva e sugere futuras direções para ampliar o uso de LLMs na manipulação e otimização de G-code.

Beckwith et al. (2021) aborda, em outra pesquisa, a detecção de edições maliciosas em arquivos de G-code utilizados em MA, focando na segurança cibernética de processos digitais de fabricação. Por meio de um experimento de equipes "*red-team/blue-team*", os autores investigam métodos para identificar alterações que comprometem a integridade estrutural das peças impressas. A equipe *red* introduziu vulnerabilidades específicas no G-code, como alterações em comandos G1 e G0, reduzindo extrusões ou criando lacunas no material. A equipe *blue* utilizou análises estatísticas e algoritmos de aprendizado de máquina para detectar anomalias. Os resultados mostram que abordagens combinadas de análise estatística e aprendizado de máquina são eficazes para identificar compromissos em G-code, embora desafios permaneçam, especialmente com defeitos globais que não causam mudanças abruptas nos dados. O trabalho destaca a importância de estratégias proativas de segurança em manufatura digital e sugere a necessidade de maior integração entre algoritmos avançados e grandes conjuntos de dados para melhorar a detecção e a prevenção de ataques em G-code.

Em outro trabalho, Jignasu et al. (2024) apresenta o *Slice-100K*, um dataset multimodal pioneiro com mais de 100.000 arquivos de G-code, modelos CAD em STL, renderizações e metadados, voltado para aplicações em manufatura aditiva baseada em extrusão. A pesquisa destaca a importância do G-code como linguagem intermediária entre design digital e fabricação física e busca otimizar sua utilização por meio de modelos de linguagem avançados (LLMs). O *Slice-100K* foi desenvolvido a partir de fontes como Objaverse-XL e Thingi10K, abrangendo ampla diversidade de arquivos próprios para impressão 3D. As contribuições incluem métodos para tradução de "*flavours*", ou seja, dialetos, de G-code (como Sailfish para Marlin) e transformações geométricas, demonstrando como os LLMs podem facilitar a depuração e manipulação de G-code. O estudo propõe novos caminhos para o desenvolvimento de modelos fundacionais específicos para manufatura, apontando o *Slice-100K* como recurso essencial para fomentar inovações em impressão 3D e melhorar a eficiência em processos de fabricação.

Um framework para simulação baseada em física do processo de impressão por FDM, utilizou o conceito de gêmeos digitais para processos de impressão 3D. A abordagem utiliza métodos de malhas adaptativas em *octree* e representação *voxel* para converter arquivos G-code em modelos geométricos adequados para análise. O framework realiza simulações térmicas em tempo real durante o processo de impressão, abordando desafios como distribuição de calor, padrões de preenchimento e otimização do uso de materiais. Além disso, o método é validado utilizando modelos 3D de alta complexidade, como o modelo *Stanford Bunny*, modelo 3D de um coelho, e a estátua Moai, demonstrando alta escalabilidade e eficiência computacional. As contribuições incluem a capacidade de prever a qualidade da impressão, identificar possíveis defeitos e explorar sequências de impressão de forma rápida e precisa, utilizando dados do G-code. Este trabalho estabelece bases sólidas para futuras integrações com controle de feedback e algoritmos preditivos, oferecendo avanços significativos na qualidade e eficiência da manufatura aditiva (GAMDHA et al., 2020).

2.3 Biarcos

O trabalho de Meek e Walton (1991) apresenta métodos para aproximar dados discretos utilizando splines de arco, especificamente curvas G^1 compostas por arcos circulares e segmentos de reta. Essas curvas são amplamente utilizadas em máquinas de corte controladas automaticamente, como as empregadas em manufatura aditiva e processos de CNC. A pesquisa propõe modificações simples em dois algoritmos existentes para gerar splines de arco aproximados sem a necessidade de resolver sistemas complexos de equações não lineares. O método utiliza biarcos, formados por dois arcos conectados de maneira contínua em tangente, para aproximar pontos discretos com alta precisão e dentro de um limite especificado de erro. As contribuições incluem dois algoritmos: o método do arco mais longo, que minimiza o número de biarcos, e o método de bissetção, que encontra uma solução mais rapidamente, mas utiliza mais biarcos. Os resultados mostram que ambos os métodos são eficazes para representar dados com splines

de arco, com exemplos que ilustram aplicações práticas em modelagem de curvas fechadas e aproximação geométrica.

Um outro trabalho também apresentou método para construção de splines de biarcos. O trabalho mostrou que interpolação de sequências de pontos ordenados no plano, otimiza funções-alvo, como de comprimento mínimo, de integral do módulo da curvatura ou de energia mínima (curvatura ao quadrado). O algoritmo proposto combina um problema de programação não-linear (NLP) com gradientes e matrizes hessianas para melhorar a performance computacional. Testes numéricos validam a eficiência do método em aplicações práticas, como controle de trajetória para veículos e manufatura assistida por computador (CAD/CAM). A abordagem permite ajustar ângulos e otimizar propriedades geométricas, demonstrando precisão na interpolação e flexibilidade para diferentes contextos industriais. (BERTOLAZZI; FREGO; BIRAL 2019)

Uma análise abrangente do uso de Redes Neurais Artificiais (ANN) para a otimização de processos de manufatura aditiva foi tema de um estudo. Nele, foi explorado como as ANN podem ser empregadas para superar os desafios relacionados à complexidade do controle de parâmetros de impressão, como espessura da camada, velocidade de escaneamento e energia aplicada: parâmetros diretamente ligados ao G-code, referido no estudo como CNC para impressão 3D. Além do trabalho discutir aplicações em monitoramento de processos, ele analisa correlações entre parâmetros de impressão e propriedades dos produtos finais e recomendações de design. Apesar das limitações em termos de datasets e requisitos computacionais, o estudo identifica avanços promissores no uso de ANN, como melhorias na precisão de predição e redução de tempo de simulações, sugerindo seu potencial para integração em sistemas de manufatura avançada e digitalização industrial (MAHMOOD et al. 2020).

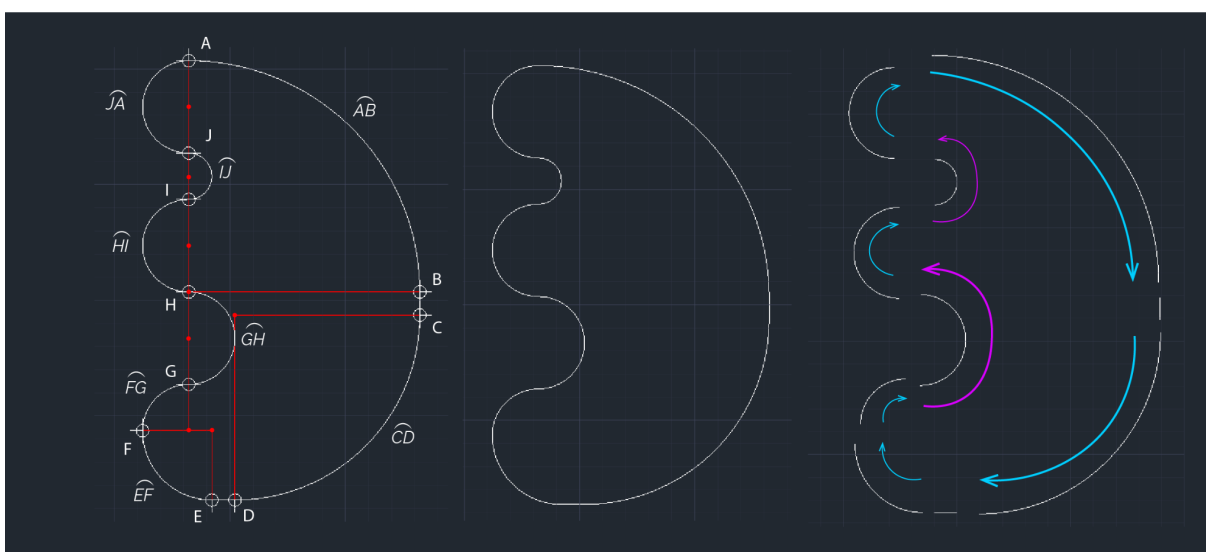
3. MODELAGEM 3D

O processo de modelagem de uma curva fechada complexa no AutoCAD, descrito neste projeto, envolve a criação de um trajeto formado por arcos que alternam sentidos horário e anti-horário, conferindo um caráter único à geometria. Essa abordagem difere de formas simples, como circunferências e elipses, pois a

alternância dos arcos cria trajetórias que não apenas seguem um padrão uniforme, mas combinam transições suaves e mudanças de sentido. A curva foi desenhada a partir de trechos de arcos de circunferência de 180° e 90° de abertura, alternando o sentido e ângulo. Dois pequenos segmentos de retas foram utilizados na construção.

As curvas foram construídas a partir do uso de NURBS, relevantes na área da computação gráfica, sendo como um modelo matemático de representação visual de curvas. Piegl e Tiller (1997) apontam que as NURBS (*Non-Uniform Rational B-Splines*) são ferramentas fundamentais para representar curvas e superfícies de alta precisão. Essa abordagem permite conectar suavemente arcos consecutivos, respeitando a continuidade tangencial, essencial para impressão de geometrias complexas. Ao utilizar segmentos de retas minúsculos para representar tais curvas, as NURBS oferece uma visualização satisfatória no *software* de modelagem 3D.

Figura 8 - Construção geométrica da curva

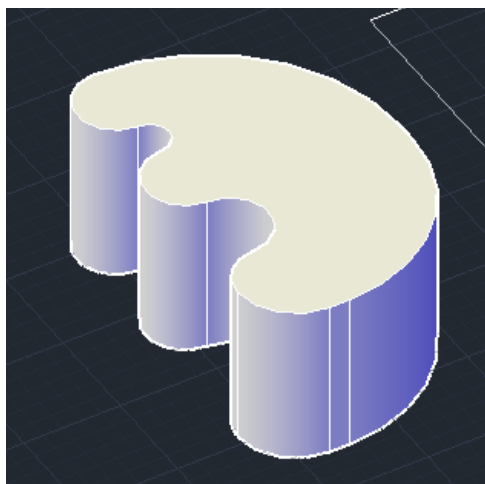


Nota: À esquerda, a curva está fechada. Os arcos AB, CD, EF, FG, GH, HI, IJ, JA e os segmentos de reta BC e DE têm tangência concordantes. Ao centro, a curva é apresentada. À direita, explicita-se os arcos em sentido horário na cor azul, e os arcos em sentido anti-horário na cor roxa.

Fonte: Elaborado pelo autor (2024)

Após a modelagem dessa curva fechada, utiliza-se o comando de extrusão no AutoCAD para gerar um sólido tridimensional, formando uma extrusão reta perpendicular ao plano original da curva, em 90° . Essa escolha gera um sólido com características geométricas complexas e não-convexas, dispensando a necessidade de suportes adicionais durante a impressão 3D.

Figura 9 - Sólido projetado a partir da curva



Fonte: Elaborado pelo autor (2024)

A justificativa para essa construção geométrica está diretamente relacionada ao processo de geração do G-code, onde os comandos G2 e G3 serão utilizados para traçar trajetórias em arcos de circunferência. Esses comandos são essenciais para reproduzir com precisão as seções curvas do modelo, que variam entre sentidos horário e anti-horário. A alternância dessas trajetórias contribui para a complexidade do objeto impresso e reflete diretamente a fidelidade do modelo digital ao sólido físico.

Figura 10 - Detalhamento do efeito de tesselação do modelo em .STL no Paint 3D



Fonte: Elaborado pelo autor (2024)

Naturalmente, é possível ver na Figura 10 o efeito da tesselação. Isto é, o processo de facetamento da geometria curva previamente modelada, ocasionando uma simplificação, ressaltando os planos que compõe o sólido já transformado na nova extensão de arquivo (.STL). Isto é inevitável, até certo ponto, pois extensão é necessária para o uso do software de fatiamento das impressoras 3D utilizadas. A única forma de contornar a situação é utilizando modelos com uma contagem de vértices altíssima, criando arquivos pesados e de difícil processamento.

A decisão de modelar um objeto não-convexo com superfícies curvas busca explorar e colocar em prova o algoritmo. O formato STL exportado do AutoCAD, contendo a descrição por tesselação da superfície, possibilita o fatiamento preciso do modelo. O efeito de mosaico pode ser observado tanto no software de fatiamento ou em qualquer software de modelagem 3D. Como o algoritmo funciona selecionando os pontos do perfil de casca do objeto presentes no G-code, o modelo extrusado num ângulo de 90° é ideal para a replicação e comparação.

Ao evitar formas simples, como esferas, poliedros ou elipsóides, e focar na criação de um objeto tridimensional a partir de curvas fechadas complexas, o experimento demonstra que essa técnica de modelagem pode gerar objetos que testam o algoritmo a certo nível de profundidade. O uso de comandos como G2 e G3 durante a impressão é o real escopo do teste, e pode evidenciar a importância de modelagens atentas a tais detalhes. Esses comandos assegurarão que o objeto impresso mantenha alta precisão e reflita todas as nuances da curva original. Essa abordagem valoriza tanto a estética do produto final quanto a funcionalidade dos comandos.

4. ALGORITMO DE GERAÇÃO DE BIARCOS

Neste capítulo, explicamos o desenvolvimento de um algoritmo para converter comandos lineares (G1) do G-code em comandos de arco (G2 e G3). A transformação, fundamentada no uso de biarcos, visa reduzir as imperfeições visuais em superfícies curvas produzidas por impressoras 3D. Este método busca suavizar o caminho do cabeçote de impressão ao longo de curvas complexas, assegurando continuidade tangencial e, assim, elevando a qualidade da superfície impressa.

A seguir, descreveremos o processo de transformação de uma sequência de pontos em uma curva suave com biarcos, incluindo as bibliotecas utilizadas e cada função implementada.

O código utiliza as seguintes bibliotecas:

- I. *numpy*: Para manipulação numérica e operações vetoriais, essencial no cálculo dos pontos e tangentes.
- II. *matplotlib.pyplot*: Para visualização gráfica, permitindo traçar a curva e os pontos amostrados.
- III. *scipy.interpolate (splprep e splev)*: Para interpolação de splines, produzindo uma curva contínua que serve como base para os biarcos.
- IV. *re*: Para a manipulação e extração de coordenadas do G-code em formato de texto.

A função *extract_coordinates* analisa o conteúdo do arquivo G-code gerado por um software de fatiamento (como o *PrusaSlicer*), identificando coordenadas usando expressões regulares. Estas coordenadas são convertidas em uma lista de pontos no formato de par ordenado (x, y).

Snippet de código 2 - Extração das coordenadas
no arquivo de saída da fatiadora

```
def extract_coordinates(file_content):
    pattern = r'; objects_info = \{"objects":\{"name":"[^"]+", "polygon":(\[[\s.*?\\]\])'
    match = re.search(pattern, file_content)

    if match:
        coordinates_str = match.group(1)
        points = eval(coordinates_str)
        return points
    else:
        print("Nenhum ponto encontrado no arquivo.")
        return []
```

Após extrair as coordenadas, utilizamos *splprep* para gerar uma spline suave que passa pelos pontos, com ajuste para manter continuidade e suavidade.

- I. *splprep*: ajusta uma spline com base nos pontos de entrada. O parâmetro $s=3$ permite certa suavização, enquanto $per=True$ faz a spline ser periódica (fechada).
- II. *splev*: avalia a spline em um conjunto de valores uniformemente distribuídos para gerar uma curva contínua.

Snippet de código 3 - Geração de splines

```
tck, u = splprep([x, y], s=3, per=True)
num_samples = 500
u_new = np.linspace(0, 1, num_samples)
x_new, y_new = splev(u_new, tck)
dx, dy = splev(u_new, tck, der=1)
```

A função *arc_from_tangents* calcula o centro e o raio do arco que conecta dois pontos consecutivos (p_0 , p_1) com suas tangentes (t_0 , t_1).

- I. Tangentes Perpendiculares: Gera vetores perpendiculares para t_0 e t_1 , que ajudam a determinar a localização do centro do arco.
- II. Solução Linear: Resolva o sistema linear usando `np.linalg.solve` para calcular a distância entre os pontos e o centro do arco.

- III. Limite de Raio: Limita o raio do arco para evitar arcos excessivamente grandes, retornando `None` se o arco ultrapassa o raio máximo permitido.

Snippet de código 4 - Geração de arcos

```
def arc_from_tangents(p0, t0, p1, t1, max_radius=1000):
    t0_perp = np.array([-t0[1], t0[0]])
    t1_perp = np.array([-t1[1], t1[0]])
    A = np.array([t0_perp, -t1_perp]).T
    b = p1 - p0
    try:
        s = np.linalg.solve(A, b)
        center = p0 + s[0] * t0_perp
        radius = np.linalg.norm(center - p0)

        if radius > max_radius:
            return None
        return center, radius
    except np.linalg.LinAlgError:
        return None
```

Neste ponto, o algoritmo começa a gerar o G-code para cada camada da impressão 3D. Cada camada incrementa a altura `Z`, criando uma sequência de biarcos conectados por arcos G2 e G3.

- I. Estrutura de Camadas: Cada camada é especificada por um incremento *Z_increment*.
- II. Conversão de Segmentos em Arcos: Para cada par de pontos consecutivos (p0, p1), a função *arc_from_tangents* calcula se um arco pode ser utilizado. Caso contrário, o código opta por uma linha reta (`G1`).
- III. G2 e G3: O tipo de arco (G2 ou G3) depende da orientação da curva, determinada pelo produto vetorial entre a tangente inicial e o vetor entre os pontos.

Snippet de código 5 - Múltiplas iterações para diferentes camadas

```

for layer in range(num_layers):
    Z = Z_start + layer * Z_increment
    x0, y0 = x_new[0], y_new[0]
    gcode_lines.append(f"G0 X{x0:.3f} Y{y0:.3f} Z{Z:.3f}")

    for i in range(len(x_new) - 1):
        p0 = np.array([x_new[i], y_new[i]])
        p1 = np.array([x_new[i + 1], y_new[i + 1]])
        t0 = np.array([dx[i], dy[i]])
        t1 = np.array([dx[i + 1], dy[i + 1]])
        t0 /= np.linalg.norm(t0)
        t1 /= np.linalg.norm(t1)

        result = arc_from_tangents(p0, t0, p1, t1)
        if result is None:
            gcode_lines.append(f"G1 X{p1[0]:.3f} Y{p1[1]:.3f}")
        else:
            center, radius = result
            I, J = center[0] - p0[0], center[1] - p0[1]
            cross = np.cross(t0, p1 - p0)
            gcode_cmd = "G2" if cross < 0 else "G3"
            gcode_lines.append(f"{gcode_cmd} X{p1[0]:.3f} Y{p1[1]:.3f} I{I:.3f} J{J:.3f}")

```

Por fim, o código escreve o G-code resultante em um arquivo *output_biarc_only.gcode*, permitindo que a trajetória de biarcos seja executada por uma impressora 3D.

Snippet de código 6 - Salvamento de um arquivo G-code novo

```

with open('output_biarc_only.gcode', 'w') as f:
    for line in gcode_lines:
        f.write(line + '\n')
print("G-code gerado! 'output_biarc_only.gcode'.")

```

Para validar visualmente o algoritmo, o código gera uma figura da spline interpolada em comparação com os pontos amostrados.

Snippet de código 7 - Visualização da figura da spline interpolada

```
plt.figure(figsize=(8, 8))  
plt.plot(x_new, y_new, 'b-', label='Spline Interpolada')  
plt.plot(x_new, y_new, 'ro', label='Pontos Amostrados')  
plt.axis('equal')  
plt.legend()  
plt.show()
```

O algoritmo descrito proporciona uma maneira eficaz de transformar segmentos lineares em trajetórias suaves utilizando biarcos. Com a interpolação por splines e a subsequente geração de arcos, o G-code final resulta em uma impressão 3D mais precisa, especialmente em superfícies curvas complexas, sem a necessidade de aumentar a densidade de amostragem.

5. APLICAÇÃO DO ALGORITMO

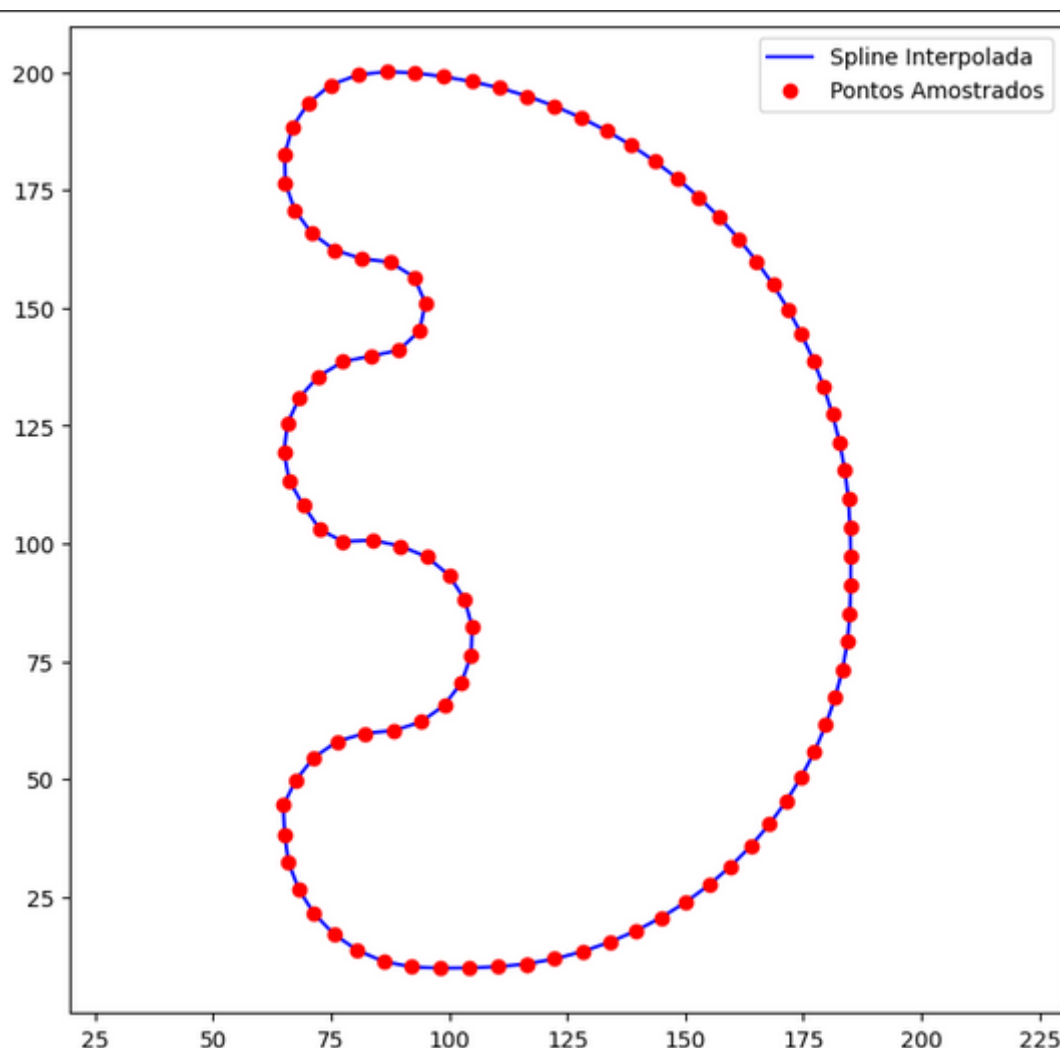
Com o arquivo .STL do objeto modelado, obteremos o G-code após fatiamento no software PrusaSlicer. Ele será composto de comandos G1 de trajetos retilíneos, como visto no snippet abaixo:

Snippet de código 8 - G-code de impressão a partir do código gerado pelo PrusaSlicer

```
G1 X80.055 Y138.456 E.42082
G1 X75.447 Y136.547 E.271
G1 X71.489 Y133.511 E.27103
G1 X68.453 Y129.553 E.27103
G1 X66.544 Y124.945 E.271
G1 X65.893 Y120 E.27099
G1 X66.544 Y115.055 E.27099
G1 X68.453 Y110.447 E.271
G1 X71.489 Y106.489 E.27103
G1 X75.447 Y103.453 E.27103
G1 X80.055 Y101.544 E.271
G1 X89.46 Y100.306 E.51541
G1 X90.484 Y100.15 E.05628
G1 X94.67 Y98.416 E.24618
G1 X95.515 Y98.042 E.05021
G1 X98.936 Y95.417 E.23429
G1 X99.826 Y94.706 E.06189
G1 X103.042 Y90.515 E.28703
G1 X103.127 Y90.367 E.00927
G1 X105.148 Y85.487 E.28698
G1 X105.193 Y85.322 E.00929
G1 X105.882 Y80.085 E.28699
G1 X105.882 Y79.915 E.00924
G1 X105.193 Y74.678 E.28699
G1 X105.148 Y74.513 E.00929
G1 X103.127 Y69.633 E.28698
G1 X103.042 Y69.485 E.00927
G1 X100.041 Y65.575 E.2678
G1 X99.706 Y65.174 E.02839
G1 X95.515 Y61.958 E.28703
G1 X95.367 Y61.873 E.00927
G1 X90.487 Y59.852 E.28698
G1 X90.322 Y59.807 E.00929
G1 X80.055 Y58.456 E.56264
G1 X75.447 Y56.547 E.271
G1 X71.489 Y53.511 E.27103
G1 X68.453 Y49.553 E.27103
G1 X66.544 Y44.945 E.271
G1 X65.892 Y39.992 E.27143
G1 X66.621 Y33.523 E.3537
G1 X68.774 Y27.37 E.35418
```

Com o G-code em mãos, o algoritmo entra em ação e solicita o arquivo.

Figura 11 - Exemplo de visualização dos pontos amostrados com a interpolação dos pontos



Nota: O parâmetro “num_samples” é igual a 100.

Fonte: Elaborado pelo autor (2024)

O algoritmo carrega o arquivo e processa imediatamente, devolvendo um arquivo G-code com o código novo, com comandos G2 e G3. O algoritmo retorna também a amostra da fatia do objeto descrito no G-code já interpolada a partir dos pontos amostrados originais, como mostra a Figura 11. O usuário pode ajustar o parâmetro “num_samples” e reprocessar a geração para obter um resultado satisfatório de interpolação.

Podemos perceber nos *snippets* abaixo que o código utiliza comandos G2 e G3 de fato.

Snippet de código 9 - G-code em biarcos, um exemplo de uso completo dos comandos G3

```
G3 X85.080 Y60.046 I168.412 J-856.790
G3 X83.869 Y59.803 I36.132 J-182.462
G3 X82.662 Y59.549 I20.806 J-101.524
G3 X81.462 Y59.277 I15.106 J-69.495
G3 X80.275 Y58.984 I12.260 J-52.049
G3 X79.104 Y58.662 I10.634 J-40.927
G3 X77.954 Y58.307 I9.638 J-33.151
G3 X76.830 Y57.912 I9.008 J-27.386
G3 X75.737 Y57.472 I8.614 J-22.947
G3 X74.678 Y56.981 I8.387 J-19.445
G3 X73.659 Y56.434 I8.293 J-16.642
G3 X72.683 Y55.824 I8.323 J-14.383
G3 X71.756 Y55.147 I8.478 J-12.559
G3 X70.882 Y54.396 I8.776 J-11.086
G3 X70.066 Y53.567 I9.727 J-10.418
G3 X69.307 Y52.666 I11.501 J-10.460
G3 X68.608 Y51.700 I13.348 J-10.408
G3 X67.969 Y50.678 I15.212 J-10.225
```

Snippet de código 10 - G-code em biarcos, um exemplo de uso completo dos comandos G2

```
G2 X85.071 Y100.255 I-40.771 J-264.099
G2 X86.274 Y100.056 I-13.692 J-86.088
G2 X87.471 Y99.835 I-8.856 J-51.067
G2 X88.659 Y99.581 I-7.107 J-35.942
G2 X89.833 Y99.282 I-6.381 J-27.453
G2 X90.988 Y98.929 I-6.409 J-23.038
G2 X92.122 Y98.519 I-7.415 J-22.235
G2 X93.230 Y98.051 I-8.416 J-21.487
G2 X94.309 Y97.527 I-9.344 J-20.621
G2 X95.356 Y96.949 I-10.193 J-19.653
G2 X96.366 Y96.315 I-10.962 J-18.604
G2 X97.336 Y95.628 I-11.654 J-17.493
G2 X98.264 Y94.889 I-12.274 J-16.339
G2 X99.145 Y94.097 I-12.830 J-15.160
G2 X99.976 Y93.255 I-13.332 J-13.972
G2 X100.753 Y92.362 I-13.795 J-12.788
G2 X101.473 Y91.420 I-14.232 J-11.618
G2 X102.132 Y90.429 I-14.659 J-10.466
G2 X102.727 Y89.391 I-15.282 J-9.453
G2 X103.256 Y88.309 I-16.476 J-8.733
G2 X103.718 Y87.190 I-17.592 J-7.915
```

Utilizando o simulador *G-code Q'n'dirty toolpath simulator*, podemos visualizar os caminhos descritos pelo algoritmo em comandos G2 G3.

Figura 12 - Visualização no *G-code Q'n'dirty toolpath simulator* das novas trajetórias

G-Code Q'n'dirty toolpath simulator

[Fork me on GitHub](#)

Paste your g-code in the left-hand window or drop a file on the page and see the preview of your tool path on the right.
The right-hand pane are interactive, drag them to change the point of view.

```

49977 G3 X88.804 Y11.096 I11.437 J45.605
49978 G3 X89.983 Y10.848 I11.100 J49.602
49979 G3 X91.170 Y10.625 I10.754 J54.031
49980 G3 X92.362 Y10.427 I10.399 J58.970
49981 G3 X93.560 Y10.252 I10.039 J64.521
49982 G3 X94.760 Y10.099 I9.679 J70.824
49983 G3 X95.962 Y9.965 I9.341 J75.685
49984 G3 X97.164 Y9.850 I8.960 J76.015
49985 G3 X98.367 Y9.755 I8.664 J76.203
49986 G3 X99.570 Y9.678 I8.467 J76.387
49987 G3 X100.773 Y9.621 I8.270 J76.567
49988 G3 X101.977 Y9.582 I8.071 J76.742
49989 G3 X103.180 Y9.562 I7.871 J76.912
49990 G3 X104.382 Y9.561 I7.669 J77.077
49991 G3 X105.585 Y9.579 I7.464 J77.236
49992 G3 X106.786 Y9.615 I7.240 J77.388
49993 G3 X107.986 Y9.670 I7.009 J77.534
49994 G3 X109.186 Y9.743 I6.768 J77.672
49995 G3 X110.384 Y9.835 I6.517 J77.803
49996 G3 X111.581 Y9.946 I6.259 J77.926
49997 G3 X112.776 Y10.075 I5.999 J78.040
49998 G3 X113.970 Y10.222 I5.731 J78.146
49999 G3 X115.161 Y10.387 I5.455 J78.242
50000 G3 X116.351 Y10.571 I5.171 J78.328
50001

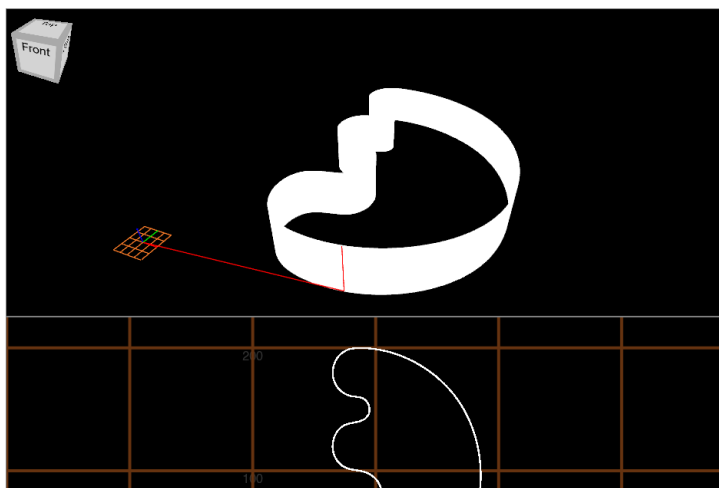
```

[Simulate](#) [Load a bigger sample](#)

Total Duration:
52m8s

Bounds (@tool center):

	min	max
X	0	116.351
Y	0	199.3032
Z	0	30



Fonte: Elaborado pelo autor (2024)

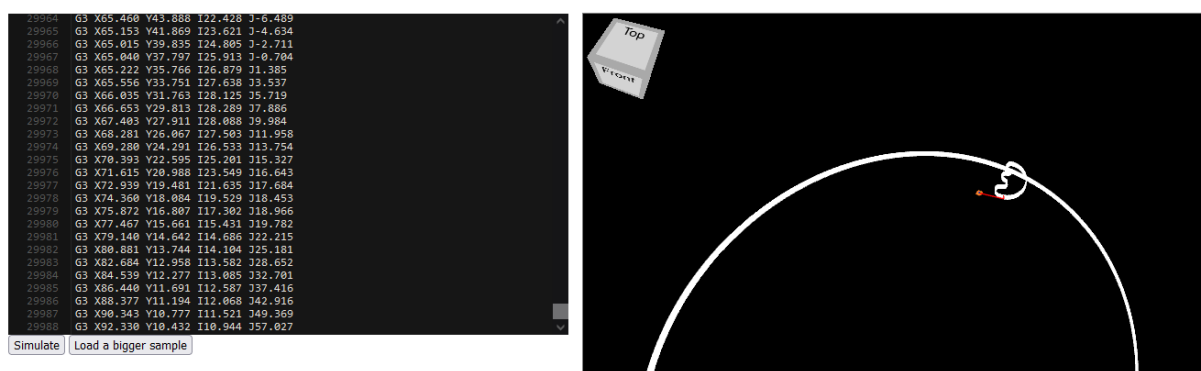
6. AVALIAÇÃO E DISCUSSÃO

Os resultados apontam que o algoritmo obteve sucesso em processar um arquivo G-code original e devolver um novo, recalculado a partir dos pontos interpolados. O número de amostras “num_samples” pode ser ajustado facilmente para adaptação à geometria do G-code inicial. Foi experimentado que o número de amostras está diretamente relacionado à qualidade do G-code por biarcos. Por exemplo, fazendo num_samples = 300, o G-code apresenta falhas de cálculo da geometria.

Figura 13 - Visualização no *G-code Q'n'dirty toolpath simulator* das novas trajetórias fazendo num_samples = 300

G-Code Q'n'dirty toolpath simulator

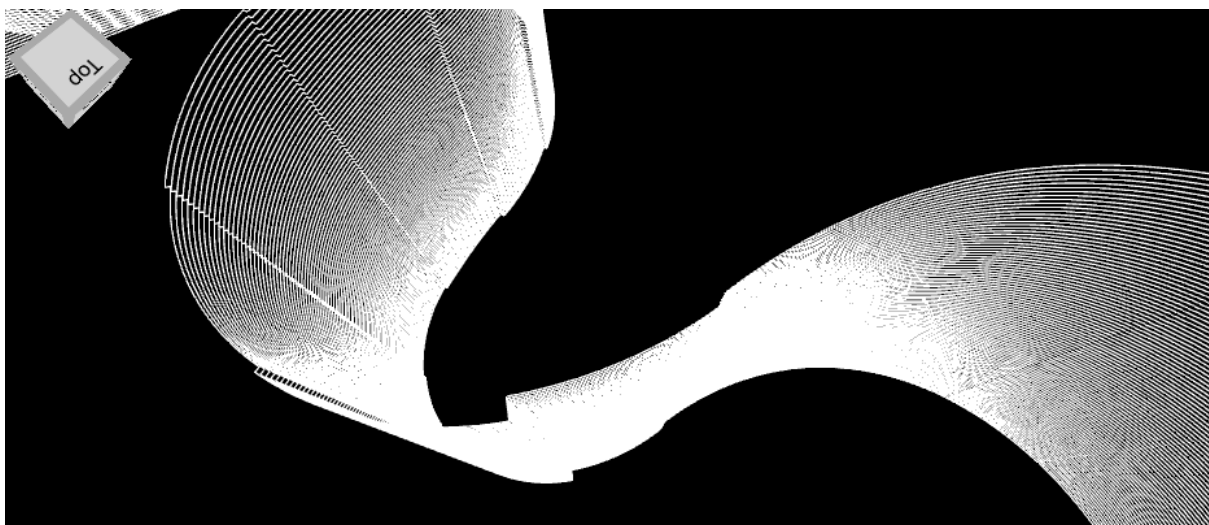
Paste your g-code in the left-hand window or drop a file on the page and see the preview of your tool path on the right.
The right-hand pane are interactive, drag them to change the point of view.



Fonte: Elaborado pelo autor (2024)

Utilizando um fator baixíssimo de interpolações, os arcos começam a deixar de apresentar tangência entre eles. Observamos na Figura 15 a característica abrupta entre os segmentos em biarcos.

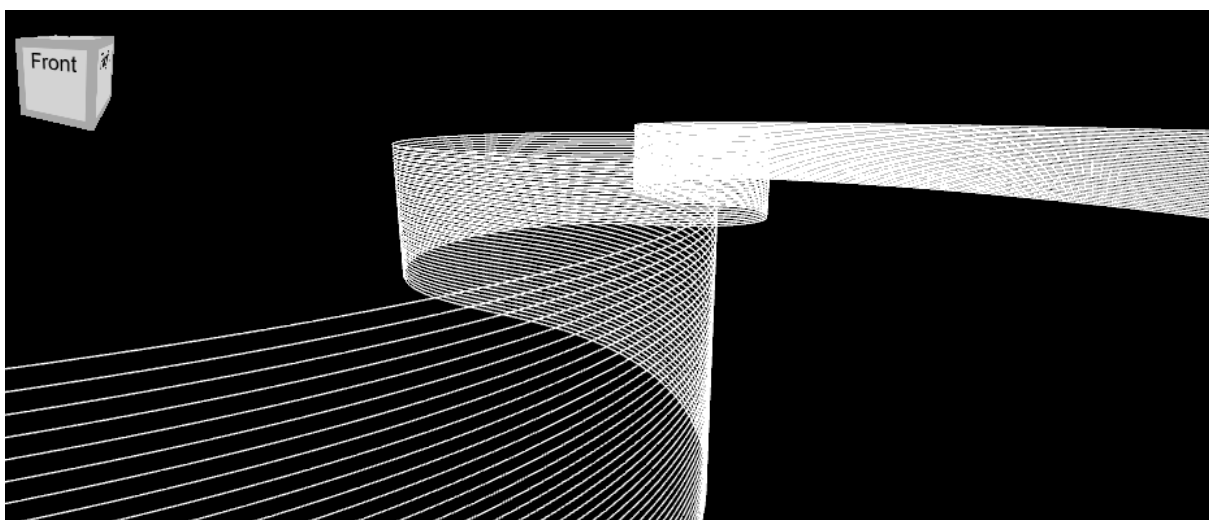
Figura 14 - Visualização no *G-code Q'n'dirty toolpath simulator* das novas trajetórias fazendo $\text{num_samples} = 80$



Fonte: Elaborado pelo autor (2024)

Durante os experimentos, foi percebido que utilizar um fator de $\text{num_samples} = 500$ faz o trajeto não apresentar as falhas citadas acima. Tanto as inconsistências tangenciais quanto às falhas de cálculo da geometria são ajustadas com o valor aqui citado.

Figura 15 - Visualização no *G-code Q'n'dirty toolpath simulator* das novas trajetórias fazendo $\text{num_samples} = 500$



Fonte: Elaborado pelo autor (2024)

A avaliação experimental comparou o desempenho do G-code convencional baseado em segmentos de reta e o método proposto com splines de biarcos. Foi constatado através da análise do G-code gerado pelo algoritmo que as trajetórias geradas pelos biarcos apresentaram comandos circulares (utilizando estritamente comandos G2 e G3). A visualização no simulador corrobora que a trajetória é descrita por curvas, pois ela não mais apresenta o efeito decorrente da tesselação previamente descrito. Essa melhoria é especialmente visível em geometrias curvas complexas, confirmando a hipótese de que o método suaviza as trajetórias sem comprometer a precisão. Assim, o uso do algoritmo de geração de biarcos em G-code é comprovadamente eficiente na entrega de um código de impressão baseado em curvas concordantes.

Além da análise qualitativa, foram realizados testes quantitativos para mensurar a suavidade dos trajetos. Ao comparar as mudanças angulares entre segmentos consecutivos, verificou-se que os biarcos mantiveram a continuidade tangencial de forma consistente, eliminando cúspides e descontinuidades características do método tradicional. Isso confirma que o algoritmo não apenas preserva a suavidade, mas também gera trajetórias mais adequadas para impressões de alta complexidade. Em síntese, o método de biarcos oferece ganhos claros na qualidade do trajeto.

Em termos absolutos, não houve uma significativa diminuição no tempo de processamento e impressão. Entretanto, o resultado representa efetivamente uma melhor escolha no trade-off entre modelos, uma vez que fornece uma qualidade de impressão sem tesselação em um tempo semelhante ao modelo com comandos G1.

A análise crítica dos resultados aponta que, enquanto o método de biarcos supera o G-code convencional em qualidade superficial e suavidade, algumas limitações ainda devem ser consideradas. Por exemplo, a adaptação do algoritmo para suportar variações extremas de curvatura pode ser melhorada, e a integração direta com softwares de fatiamento requer otimizações adicionais. Dessa forma, o método ainda possui espaço para evoluir, especialmente em termos de aplicabilidade universal.

Finalmente, as implicações deste estudo indicam um avanço significativo no pipeline de impressão 3D, especialmente para manufatura aditiva de geometrias

curvas e complexas. O uso de biarcos não só aprimora a qualidade dos objetos impressos, mas também estabelece um novo paradigma para o desenvolvimento de algoritmos de fatiamento. Assim, os resultados reforçam o potencial dos biarcos como uma alternativa promissora no campo da impressão 3D.

7. CONCLUSÃO

A impressão 3D é uma tecnologia que impacta um alcance de áreas, desde a saúde até a engenharia, passando pelo design, educação e manufatura. Sua versatilidade e capacidade de produzir objetos com geometrias complexas tornam-na indispensável para a inovação em inúmeros campos. Entretanto, desafios técnicos ainda persistem, especialmente no que se refere à qualidade de superfícies curvas e à eficiência dos processos de fatiamento e impressão.

Com isso, nosso trabalho, ao investigar a aplicação de splines G^1 de biarcos em substituição aos segmentos de reta no G-code, busca oferecer uma perspectiva viável para suavizar superfícies impressas sem sacrificar a eficiência operacional. Essa abordagem tem potencial para mitigar o aspecto facetado das superfícies utilizando tangenciamento contínuo entre biarcos, contribuindo na área da impressão 3D, permitindo maior qualidade estética e funcional.

Os resultados desta pesquisa incluem avanços na precisão geométrica das peças a serem impressas, no que diz respeito à criação de G-codes mais precisos e visualmente mais confiáveis. O trabalho também entrega uma alternativa satisfatória na abordagem do problema dos artefatos visuais indesejados causados pela tesselação, uma vez que entrega uma menor dependência de remodelamento de objetos 3D, realizando um *bypass* no pipeline. Além disso, o estudo estabelece uma base para futuras implementações práticas, como o desenvolvimento de softwares de fatiamento que incorporem o uso de biarcos de maneira otimizada. Tais ferramentas poderão beneficiar diretamente usuários finais, sejam eles indústrias, profissionais ou entusiastas da impressão 3D.

O estado da arte atual da impressão 3D por FDM é baseada na técnica de trajetórias por comandos G1 no G-code. Simplesmente é arbitrado que aumente-se a densidade de geometria do produto para obter-se um resultado. Porém, isso representa uma limitação tanto no aspecto da otimização, quanto da dimensão da manufatura. Numa fabricação em escala de dimensões ampliadas, podemos inferir que os modelos teriam uma densidade de vértices muito altas, ocasionando num tamanho de arquivo muito alto para processamento, para obter resolução satisfatória. Este trabalho pode funcionar como base para desenvolvimento de novas

tecnologias na área, como implementação completa da técnica de biarcos numa fatiadora especializada, desenvolvimento de impressoras especializadas em trajetória em arcos de circunferências ou até um pipeline que envolva ignorar a etapa de transformação em arquivo .STL do modelo – enfatizando o uso de modelagem por NURBS.

Em trabalhos futuros, é possível vislumbrar aplicações mais generalistas deste projeto. O algoritmo atual tem como escopo por em teste a exportação de novo arquivo em extensão G-code, transformando os trajetos originais de *input* em caminhos circulares, atestando sua eficácia computacional através da observação do código. Porém, é necessário expandir o escopo para geração de arquivos de fatiamento do objeto original em sua totalidade, incluindo as informações adicionais dos parâmetros iniciais de fatiamento. A não-trivialidade do *pipeline* de impressão 3D amplia a complexidade e o número de tópicos a serem estudados. A questão dos suportes, preenchimentos do objeto e espessura de casca do objeto apresentam desafios bastante distintos apesar de estritamente relacionados. Outro tópico interessante é a utilização desse algoritmo no “*upscale*” de modelos 3D mais simples, utilizando a interpolação de pontos presente no código, promovendo acabamentos mais suaves em possíveis objetos impressos. A forma com que um algoritmo desse tipo pode auxiliar a explorar e experimentar soluções a diferentes problemas é de interessante relevância para a área.

Por fim, enfatizamos que esta pesquisa não apenas contribui para o avanço técnico da manufatura aditiva, mas também reforça a importância da constante busca por melhorias em processos inovadores. Acreditamos que nossos achados possam inspirar novas investigações. Este trabalho, portanto, se posiciona como uma contribuição relevante e prática para o aperfeiçoamento do pipeline de impressão 3D, abrindo caminhos para aplicações mais eficientes e acessíveis em diversas áreas.

8. REFERÊNCIAS

AGAR, D. A.; DAS, A. K.; RUDOLFSSON, M.; LARSSON, S. H. **A review on wood powders in 3D printing: processes, properties and potential applications.** Journal of the Brazilian Society of Mechanical Sciences and Engineering, v.15, p.241-255, nov./dez. 2021. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2238785421007730>. Acesso em: 16 nov. 2024.

AL-NOAIMAT, Y. A.; GHAFAR, S. H.; CHOUGAN, M.; AL-KHEETAN, M. J. **A review of 3D printing low-carbon concrete with one-part geopolymer: Engineering, environmental and economic feasibility.** Case Studies in Construction Materials, v.18, e01818, jul. 2023. Disponível em: <https://doi.org/10.1016/j.cscm.2022.e01818>. Acesso em: 16 nov. 2024.

BARROS, G. X. da S. **Modelagem digital tridimensional para o desenvolvimento de prototipagem rápida: um enfoque sobre a modelagem orgânica.** 2012. Dissertação (Mestrado em Design) – Universidade Federal de Pernambuco, Recife, 2012.

BELL, C. **Maintaining and Troubleshooting Your 3D Printer.** Nova York: Apress, 2014. Disponível em: <https://link.springer.com/book/10.1007/978-1-4302-6808-6>. Acesso em: 16 nov. 2024.

BECKWITH, C. et al. **Needle in a Haystack: Detecting Subtle Malicious Edits to Additive Manufacturing G-code Files.** arXiv preprint, 2021. Disponível em: <https://arxiv.org/abs/2111.12746>. Acesso em: 16 nov. 2024.

BERTOLAZZI, Enrico; FREGO, Marco; BIRAL, Francesco. **Interpolating splines of biarcs from a sequence of planar points.** Computer-Aided Design and Applications, v.18, n.1, p.66–85, 2021. Disponível em: [https://cad-journal.net/files/vol_18/CAD_18\(1\)_2021_66-85.pdf](https://cad-journal.net/files/vol_18/CAD_18(1)_2021_66-85.pdf). Acesso em: 16 nov. 2024.

CICCONE, F.; BACCIAGLIA, A.; CERUTI, A. **Optimization with artificial intelligence in additive manufacturing: a systematic review.** Journal of the Brazilian Society of Mechanical Sciences and Engineering, v.43, n.8, p.1-32, maio 2023. Disponível em: <https://link.springer.com/article/10.1007/s40430-023-04200-2>. Acesso em: 16 nov. 2024.

CHACÓN, J. M. et al. **G-code generation in a NURBS workflow for precise additive manufacturing.** Rapid Prototyping Journal, v.28, n.10, p.1707-1722, dez. 2022. Disponível em: <https://doi.org/10.1108/RPJ-04-2022-0094>. Acesso em: 16 nov. 2024.

DE PAULA, Bruna Braga; MARTINS, Camila Bertini; OLIVEIRA, Tiago de. **Análise da crescente influência da cultura maker na educação: revisão sistemática da literatura no Brasil.** Educitec, v.7, jan./dez. 2021. DOI: <https://doi.org/10.31417/educitec.v7.1349>. Disponível em:

<https://sistemascmc.ifam.edu.br/educitec/index.php/educitec/article/view/1349>. Acesso em: 16 nov. 2024.

EQUIPE WISHBOX. **Material para impressão 3D: conheça os tipos de filamentos e saiba qual escolher**. 18 nov. 2020. Disponível em: <https://www.wishbox.net.br/blog/material-para-impressao-3d-conheca-os-tipos-de-filamentos-e-saiba-qual-escolher/>. Acesso em: 16 nov. 2024.

FINA, F. et al. **Selective laser sintering (SLS) 3D printing of medicines**. International Journal of Pharmaceutics, v.529, n.1-2, p.285-293, ago. 2017. Disponível em: <https://doi.org/10.1016/j.ijpharm.2017.06.082>. Acesso em: 16 nov. 2024.

GAMDHA, Dhruv; SAURABH, Kumar; GANAPATHYSUBRAMANIAN, Baskar; KRISHNAMURTHY, Adarsh. **Geometric modeling and physics simulation framework for building a digital twin of extrusion-based additive manufacturing**. maio 2023. Disponível em: <https://arxiv.org/pdf/2305.07120>. Acesso em: 16 nov. 2024.

GIBSON, I.; ROSEN, D. W.; STUCKER, B. **Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing**. 2. ed. Nova York: Springer, 2015.

GOH, G. D.; ZHANG, H.; CHONG, T. H.; YEONG, W. Y. **3D Printing of Multilayered and Multimaterial Electronics: A Review**. Advanced Electronic Materials, v.7, n.1, jan. 2021. Disponível em: <https://doi.org/10.1002/aelm.202100445>. Acesso em: 16 nov. 2024.

GOH, G. D.; SING, S. L.; YEONG, W. Y. **A review on machine learning in 3D printing: applications, potential, and challenges**. Artificial Intelligence Review, v.54, p.63–94, 2021. Publicado em: 16 jul. 2020. Disponível em: <https://doi.org/10.1007/s10462-020-09876-9>. Acesso em: 16 nov. 2024.

GREINER, G.; HORMANN, K. **Interpolating and approximating biarcs for curve design**. Computer Graphics Forum, v.16, n.3, p.13–22, 1997.

GROSS, B.; LOCKWOOD, S. Y.; SPENCE, D. M. **Recent Advances in Analytical Chemistry by 3D Printing**. Analytical Chemistry, v.89, n.1, p.57-70, jan. 2017. Disponível em: <https://doi.org/10.1021/acs.analchem.6b04344>. Acesso em: 16 nov. 2024.

HORVATH, J. **A Brief History of 3D Printing**. In: **Mastering 3D Printing**. Berkeley, CA: Apress, 2014. p. 3–10. Disponível em: https://link.springer.com/chapter/10.1007/978-1-4842-0025-4_1. Acesso em: 16 nov. 2024.

HUTMACHER, D. W. et al. **Mechanical properties and cell cultural response of polycaprolactone scaffolds designed and fabricated via fused deposition modeling.** Journal of Biomedical Materials Research, v.55, n.2, p.203–216, maio 2001. Disponível em: [https://doi.org/10.1002/1097-4636\(200105\)55:2<203::AID-JBM1007>3.0.CO;2-7](https://doi.org/10.1002/1097-4636(200105)55:2<203::AID-JBM1007>3.0.CO;2-7). Acesso em: 15 nov. 2024.

IVKIĆ, I. et al. **Towards a Cost-Benefit Analysis of Additive Manufacturing as a Service.** arXiv preprint, 2024. Disponível em: <https://arxiv.org/abs/2403.18882>. Acesso em: 16 nov. 2024.

JIGNASU, A. et al. **Towards Foundational AI Models for Additive Manufacturing: Language Models for G-Code Debugging, Manipulation, and Comprehension.** arXiv preprint, 2023. Disponível em: <https://arxiv.org/abs/2309.02465>. Acesso em: 16 nov. 2024.

JIGNASU, Anushrut et al. **Slice-100K: a multimodal dataset for extrusion-based 3D printing.** Submetido em 4 jul. 2024. Disponível em: <https://arxiv.org/abs/2407.04180>. Acesso em: 16 nov. 2024.

KRISTIAWAN, R. B. et al. **A review on the fused deposition modeling (FDM) 3D printing: Filament processing, materials, and printing parameters.** Open Engineering, v.11, n.1, p.639–649, jun. 2021. Disponível em: <https://doi.org/10.1515/eng-2021-0063>. Acesso em: 16 nov. 2024.

LIVESU, M. et al. **From 3D models to 3D prints: an overview of the processing pipeline.** Computer Graphics Forum, v.36, n.2, p.537–564, maio 2017. Disponível em: <https://doi.org/10.1111/cgf.13147>. Acesso em: 16 nov. 2024.

LYU, F. et al. **Overview of the Development of 3D-Printing Concrete: A Review.** Applied Sciences, v.11, n.21, p.9822, out. 2021. Disponível em: <https://doi.org/10.3390/app11219822>. Acesso em: 16 nov. 2024.

MAHMOOD, M. A.; VISAN, A. I.; RISTOSCU, C.; MIHAILESCU, I. N. **Artificial Neural Network Algorithms for 3D Printing.** Materials, v.14, n.1, p.163, dez. 2020. Disponível em: <https://doi.org/10.3390/ma14010163>. Acesso em: 16 nov. 2024.

MULDOON, Kirsty et al. **High precision 3D printing for micro to nano scale biomedical and electronic devices.** Micromachines, publicado em 18 abr. 2022. Disponível em: <https://www.mdpi.com/2072-666X/13/4/642>. Acesso em: 16 nov. 2024.

NGO, T. D. et al. **Additive manufacturing (3D printing): A review of materials, methods, applications and challenges.** Composites Part B: Engineering, v.143, p.172–196, jun. 2018. Disponível em: <https://doi.org/10.1016/j.compositesb.2018.02.012>. Acesso em: 16 nov. 2024.

PANDEY, P. M.; REDDY, N. V.; DHANDE, S. G. **Slicing procedures in layered manufacturing: a review.** Rapid Prototyping Journal, v.9, n.5, p.274–288, 2003.

Disponível em: <https://doi.org/10.1108/13552540310502185>. Acesso em: 16 nov. 2024.

PENANTES, V. C. K. **Experimentos em Impressão 3D: Manipulação dos comandos G2 e G3 na geração de arcos paramétricos**. 2017. Dissertação (Mestrado em Design) – Universidade Federal de Pernambuco, Recife, 2019.

PIEGL, L. A; OROPALLO, W. **Ten challenges in 3D printing**. Computer-Aided Design and Applications, v.32, p.135–148, jan. 2016. Disponível em: <https://doi.org/10.1007/s00366-015-0407-0>. Acesso em: 16 nov. 2024.

PIEGL, L.; TILLER, W. **The NURBS Book**. 2. ed. Nova York: Springer, 1997.

REPRAP.ORG. **Thermoplastics**. Disponível em: <https://reprap.org/wiki/Materials#Thermoplastics>. Acesso em: 16 nov. 2024.

ROMERO, L. et al. **Additive Manufacturing with RepRap Methodology: Current Situation and Future Prospects**. In: 2014 International Solid Freeform Fabrication Symposium. Austin, TX: [s.n.], 2014.

RONCAGLIO, Vanina. **A construção de modelos didáticos na impressora 3D: uma abordagem significativa para o ensino de embriologia**. 2020. Disponível em: <http://tede.unicentro.br:8080/jspui/handle/jspui/1464>. Acesso em: 16 nov. 2024.

SCANONI, L. C. da S. **A cultura maker na fabricação de material didático para o ensino da geometria gráfica**. 2023. Dissertação (Mestrado em Educação) – Universidade Federal de Pernambuco, Recife, 2023. Disponível em: <https://repositorio.ufpe.br/handle/123456789/51737>. Acesso em: 16 nov. 2024.

SMID, P. **CNC Programming Handbook**. 3. ed. Nova York: Industrial Press, 2008.

SONG, H.-C. et al. **Anti-aliasing for fused filament deposition**. arXiv preprint, 2016. Disponível em: <https://arxiv.org/abs/1609.03032>. Acesso em: 16 nov. 2024.

SONG, Xiaoyan; XING, Jinfeng. **3D printing technology based on two-photon polymerization**. CIESC Journal, v.66, n.9, p.3324–3332, 2015. DOI: 10.11949/j.issn.0438-1157.20150754. Disponível em: <https://hgxb.cip.com.cn/EN/10.11949/j.issn.0438-1157.20150754>. Acesso em: 16 nov. 2024.

VERMA, D. et al. **Advanced processing of 3D printed biocomposite materials using artificial intelligence**. Materials and Manufacturing Processes, v.37, n.5, p.518–538, ago. 2021. Disponível em: <https://doi.org/10.1080/10426914.2021.1945090>. Acesso em: 16 nov. 2024.

WANG, C. et al. **An Artificial-intelligence/Statistics Solution to Quantify Material Distortion for Thermal Compensation in Additive Manufacturing**. arXiv preprint, 2020. Disponível em: <https://arxiv.org/abs/2005.09084>. Acesso em: 16 nov. 2024.

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, 2016.

Disponível em: <https://onlinelibrary.wiley.com/doi/full/10.1111/cgf.12811>. Acesso em: 16 nov. 2024.

WATER, D. **G-code Explained | List of Most Important G-code Commands**. 2024. Disponível em: <https://howtomechatronics.com/tutorials/g-code-explained-list-of-most-important-g-code-commands/>. Acesso em: 16 nov. 2024.

WU, Hsin-Chieh; CHEN, Tin-Chih Toly. **Quality control issues in 3D-printing manufacturing: a review**. Rapid Prototyping Journal, v.24, n.3, p.607–614, abr. 2018. Disponível em: <https://www.emerald.com/insight/content/doi/10.1108/rpj-02-2017-0031/full/html>. Acesso em: 16 nov. 2024.

ZHANG, H.; GOH, G. D.; CHONG, T. H.; YEONG, W. Y. **3D Printing of Multilayered and Multimaterial Electronics: A Review**. Advanced Electronic Materials, v.7, n.1, jan. 2021. Disponível em: <https://doi.org/10.1002/aelm.202100445>. Acesso em: 16 nov. 2024.

ZHU, Z. et al. **3D-printed multifunctional materials enabled by artificial-intelligence-assisted fabrication technologies**. Nature Reviews Materials, v.6, p.27–47, jan. 2021. Disponível em: <https://doi.org/10.1038/s41578-020-00235-2>. Acesso em: 16 nov. 2024.

APÊNDICE A – Algoritmo de conversão de G-code

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import splprep, splev
import re

# Função para extrair coordenadas do GCODE
def extract_coordinates(file_content):
    pattern = r'; objects_info = \{"objects":\{"name":"[^"]+", "polygon":(\[[\s.*?\\]\])'
    match = re.search(pattern, file_content)

    if match:
        coordinates_str = match.group(1)
        points = eval(coordinates_str)
        return points
    else:
        print("Nenhum ponto encontrado no arquivo.")
        return []

# Função para calcular arco entre dois pontos e tangentes
def arc_from_tangents(p0, t0, p1, t1, max_radius=1000):
    t0_perp = np.array([-t0[1], t0[0]])
    t1_perp = np.array([-t1[1], t1[0]])
    A = np.array([t0_perp, -t1_perp]).T
    b = p1 - p0
    try:
        s = np.linalg.solve(A, b)
        center = p0 + s[0] * t0_perp
        radius = np.linalg.norm(center - p0)

        # Limitar o raio para evitar arcos excessivamente grandes
        if radius > max_radius:
            return None
        return center, radius
    except np.linalg.LinAlgError:
        return None

# Solicita upload de arquivo no ambiente atual
from google.colab import files
uploaded = files.upload()

# Lê o conteúdo do arquivo
file_content = next(iter(uploaded.values())).decode('utf-8')

# Extrai os pontos do conteúdo do arquivo GCODE
points = extract_coordinates(file_content)

# Prossegue se pontos foram encontrados
if points:
    points.append(points[0]) # Fecha a forma repetindo o primeiro ponto
    x = [p[0] for p in points]
    y = [p[1] for p in points]

    tck, u = splprep([x, y], s=3, per=True)

```

```

num_samples = 500
u_new = np.linspace(0, 1, num_samples)
x_new, y_new = splev(u_new, tck)
dx, dy = splev(u_new, tck, der=1)

num_layers = 40
Z_start = 0.3
Z_increment = 0.3
gcode_lines = []

for layer in range(num_layers):
    Z = Z_start + layer * Z_increment
    x0, y0 = x_new[0], y_new[0]
    gcode_lines.append(f"G0 X{x0:.3f} Y{y0:.3f} Z{Z:.3f}")

    for i in range(len(x_new) - 1):
        p0 = np.array([x_new[i], y_new[i]])
        p1 = np.array([x_new[i + 1], y_new[i + 1]])
        t0 = np.array([dx[i], dy[i]])
        t1 = np.array([dx[i + 1], dy[i + 1]])
        t0 /= np.linalg.norm(t0)
        t1 /= np.linalg.norm(t1)

        result = arc_from_tangents(p0, t0, p1, t1)
        if result is None:
            # Caso o arco seja muito grande, usar uma linha reta
            gcode_lines.append(f"G1 X{p1[0]:.3f} Y{p1[1]:.3f}")
        else:
            center, radius = result
            I, J = center[0] - p0[0], center[1] - p0[1]
            cross = np.cross(t0, p1 - p0)
            gcode_cmd = "G2" if cross < 0 else "G3"
            gcode_lines.append(f"{gcode_cmd} X{p1[0]:.3f} Y{p1[1]:.3f} I{I:.3f} J{J:.3f}")

    with open('output_biarc_only.gcode', 'w') as f:
        for line in gcode_lines:
            f.write(line + '\n')

    print("G-Code gerado! 'output_biarc_only.gcode'.")

plt.figure(figsize=(8, 8))
plt.plot(x_new, y_new, 'b-', label='Spline Interpolada')
plt.plot(x_new, y_new, 'ro', label='Pontos Amostrados')
plt.axis('equal')
plt.legend()
plt.show()

```