

Algoritmo para preenchimento poroso em impressão 3D através de deposição segmentada

Algorithm for porous filling in 3D printing through segmented deposition

RESUMO

Fabio Seiti Fukuda
fabiofukuda@alunos.utfpr.edu.br
Universidade Tecnológica
Federal do Paraná, Curitiba,
Paraná, Brasil

Neri Volpato
nvolpato@utfpr.edu.br
Universidade Tecnológica
Federal do Paraná, Curitiba,
Paraná, Brasil

Felipe Gonçalves Di Nisio
felipenisio@alunos.utfpr.edu.br
Universidade Tecnológica
Federal do Paraná, Curitiba,
Paraná, Brasil

Algumas alternativas de se obter meios porosos por Manufatura Aditiva criam padrões através da trajetória do bico extrusor. Este trabalho, no entanto, descreve um algoritmo para a obtenção de um padrão de preenchimento variando os parâmetros de extrusão e velocidade do *raster*, gerando poros de maneira controlada. A partir de um preenchimento do tipo *raster* (zigue-zague), algumas linhas são segmentadas, ou seja, divididas em trechos finos (baixa extrusão) e grossos (alta extrusão). O algoritmo foi implementado no *software* de planejamento de processo denominado de RP3 (*Rapid Prototyping Process Planning*) e utilizado para obter preenchimentos de uma camada com poros planejados. Os resultados mostram que os parâmetros definidos foram adequados para se fabricar um meio poroso.

PALAVRAS-CHAVE: Materiais porosos. Algoritmos. Processos de fabricação.

ABSTRACT

Some alternatives for obtaining porous media by Additive Manufacturing create patterns through the trajectory of the extruder nozzle. This paper, however, describes an algorithm for obtaining a filling pattern varying the extrusion parameters and *raster* speed, generating pores in a controlled manner. From a *raster* filling (zigzag), some lines are divided into thin (low extrusion) and thick (high extrusion) segments. The algorithm was implemented in the process planning software called RP3 (*Rapid Prototyping Process Planning*) and used to obtain one layer fillings with planned pores. The results show that the defined parameters were adequate to manufacture a porous medium.

KEYWORDS: Porous materials. Algorithms. Manufacturing processes.

Recebido: 19 ago. 2020.

Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

Materiais porosos possuem aplicações em diversos setores da indústria, como aeroespacial, automotiva, área médica, geração de energia, e muitos outros. Embora a sua fabricação possa ocorrer por diversas maneiras, a manufatura aditiva (AM – *Additive Manufacturing*) pelo princípio de extrusão de material possibilita que os meios porosos sejam obtidos de maneira planejada, controlando muitas vezes a quantidade, tamanho e geometria desses poros.

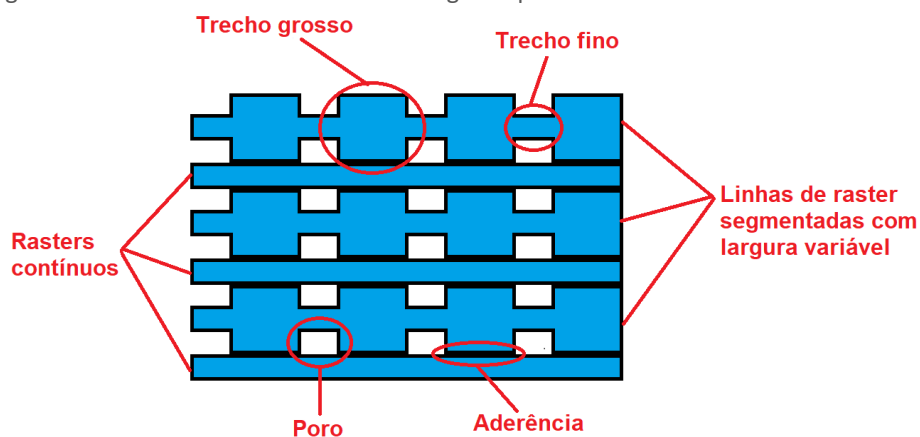
Segundo Nield e Bejan (2013), um meio poroso é um material de matriz rígida com vazios interconectados, pelos quais torna-se possível a passagem de fluídos. Em outras palavras, o poro é um pequeno volume do corpo em que não há material. Na impressão 3D, isso significa que o preenchimento deste corpo é diferente de 100%. Entretanto, preenchimentos menores simplesmente distanciam as linhas de preenchimento (linhas de *raster*) uma das outras, criando espaços entre elas (*gaps*) que impedem a aderência uma na outra. No estudo de Shigueoka (2019) foram utilizadas estratégias de geração de meios porosos através do controle de tamanho do *gap*, desalinhamento dos *rasters* entre as camadas e agrupamento de filamentos no preenchimento.

Este estudo tem como objetivo o desenvolvimento de um algoritmo computacional para planejamento de processo de impressão 3D que permita variar parâmetros de extrusão em um segmento de *raster*, gerando como resultado corpos porosos. Não foi encontrado na literatura nenhum trabalho que utilize essa mesma estratégia para obtenção de meios porosos.

MATERIAL E MÉTODOS

A proposta de preenchimento deste trabalho consiste em definir linhas possuindo segmentos de larguras diferentes (trechos finos e grossos), que podem ser intercaladas com linhas de *rasters* contínuas (sem variação de larguras). Os segmentos grossos devem aderir lateralmente com as linhas contínuas. Assim, nos locais em que não há aderência são formados os poros. A Figura 1 ilustra como deve ser a geometria gerada pelo preenchimento desenvolvido.

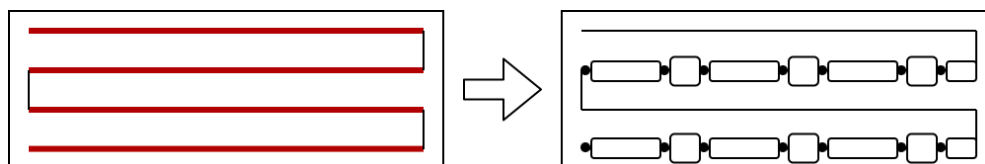
Figura 1 – Geometria resultante da estratégia de preenchimento com extrusão variável.



Fonte: Autor.

Esse padrão é criado em cima de um preenchimento do tipo *raster* previamente calculado. É necessário dizer que esse preenchimento é representado por uma lista ordenada de pontos que indicam o caminho do bico extrusor durante a impressão. Assim, o algoritmo coleta um par de pontos que representa uma linha de *raster* e a divide, calculando uma outra lista de pontos colineares que informam se o trecho até o seu sucessor é fino ou grosso.

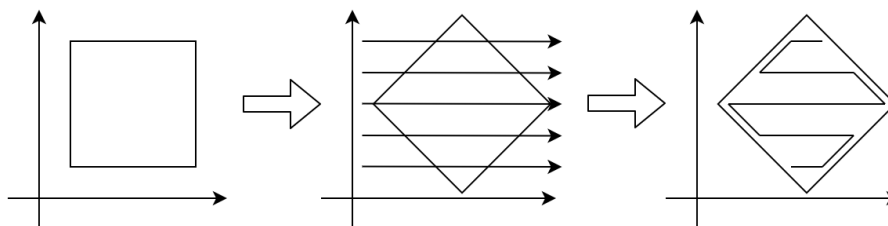
Figura 2 - Divisão das linhas de *raster* de um preenchimento.



Fonte: Autor.

A fim de simplificar cálculos (como será demonstrado mais adiante), foram utilizados apenas preenchimentos com ângulo de *raster* a 0° . Assim, no algoritmo de preenchimento, foi utilizada a estratégia apresentada por Faust (2019) de rotacionar os contornos por meio de transformações lineares, e então calcular o preenchimento com linhas de *raster* paralelas ao eixo das abscissas.

Figura 3 - Cálculo de preenchimento com rotação do contorno.

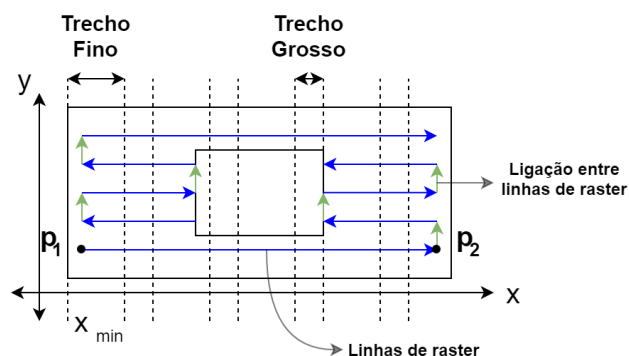


Fonte: Autor.

O preenchimento previamente calculado, sobre o qual foi aplicado o algoritmo, forma uma trajetória zigue-zague. Ou seja, em um preenchimento com ângulo de *raster* igual a 0° , uma linha de *raster* aponta para o sentido positivo de x , e a próxima aponta para o sentido negativo. Logo, para criar a alternância de linha de largura contínua e outra de largura variável, foram divididas apenas as linhas de *raster* que apontam para o sentido positivo do eixo x .

Com o cálculo do preenchimento finalizado, são feitas as chamadas da função que divide as linhas de *raster*. Essa quebra de linha segue um padrão para que, no final, os poros criados pelo preenchimento fiquem alinhados. Assim, a partir da menor coordenada em x da peça (x_{\min}), soma-se alternadamente as distâncias de trechos finos e grossos. A Figura 4 mostra esse padrão.

Figura 4 - Padrão utilizado na divisão das linhas.



Fonte: Autor.

Essa função de quebra de linha recebe como parâmetros os dois pontos p_1 e p_2 que representam a linha de raster e as distâncias de trechos finos e grossos, e retorna uma lista ordenada de pontos que representa a linha de largura variada.

Inicialmente, é computado o primeiro ponto após p_1 de acordo com o padrão mostrado na Figura 4. A lógica desse cálculo é demonstrada pelo seguinte pseudocódigo e pelas ilustrações da Figura 5.

Figura 5 : Lógica para se encontrar o primeiro ponto após p_1 .

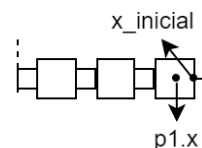
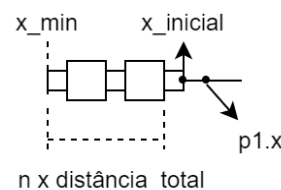
distância_total = distância_fino + distância_grossa

//Divisão inteira entre $(p1.x - x_{min})$ e distância_total
 $n = (\text{inteiro})(p1.x - x_{min}) / \text{distância_total}$

$x_{inicial} = n * \text{distância_total} + x_{min} + \text{distância_fino}$

.....

se $x_{inicial}$ for menor que $p1.x$, então:
 $x_{inicial} = x_{inicial} + \text{distância_grossa}$



Fonte: Autor.

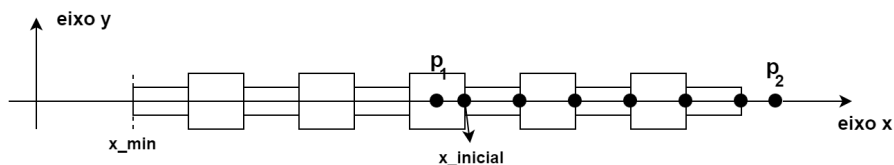
No entanto, caso o $x_{inicial}$ ultrapasse a coordenada x de p_2 , essa linha não é dividida, e a função retorna uma lista vazia. Caso contrário, tanto p_1 quanto esse último ponto computado são adicionados a uma lista ordenada de pontos.

É necessário enfatizar que a estrutura de dados dos pontos nesse algoritmo guarda as coordenadas no eixo cartesiano e também a instrução para a formação do próximo trecho. Ou seja, cada ponto informa se o percurso até o seu sucessor é um segmento grosso ou delgado. Portanto, a lista a ser retornada pela função deve iniciar com p_1 para indicar a formação do primeiro trecho, mas ela não precisa terminar com p_2 , já que este ponto indica o fim da linha, e, sendo assim, não deve conter nenhuma instrução para a produção do próximo trecho.

Para a obtenção dos próximos pontos, basta ir somando alternadamente as distâncias de trecho fino e grosso à coordenada x desse primeiro ponto, até que se atinja a coordenada x de p_2 , como mostra a Figura 6. Assim, em cada uma dessas somas é criado um novo ponto, o qual é adicionado à lista. Nessa etapa fica evidente a simplicidade de se trabalhar com linhas paralelas ao eixo das

abscissas, já que basta trabalhar com a coordenada x dos pontos. Por fim, essa lista de pontos é retornada pela função.

Figura 6 - Pontos que compõem a linha de *raster*.



Fonte: Autor.

Para se obter o controle da geometria dos poros obtidos, o algoritmo permite selecionar a velocidade de impressão, a distância e a quantidade de material extrudado dos trechos grossos e finos. Outro parâmetro de impressão que pode ser controlado é o *gap* - a distância entre as linhas de *raster*.

O algoritmo foi implementado no programa RP3 (*Rapid Prototyping Process Planning*), um *software* de planejamento de processo de impressão em desenvolvimento pelo Núcleo de Manufatura Aditiva de Ferramental (NUFER) da UTFPR Campus Curitiba.

Após a implementação, esta foi testada para o planejamento de preenchimento de uma peça em formato STL (*StereoLithography*). O modelo geométrico da peça é um paralelepípedo sólido de 100 x 74 x 20 mm, com um furo central de diâmetro 25 mm. Utilizou-se uma distância de 5 mm para trechos finos e 3 mm para os grossos, e também um *gap* entre as linhas de 2,4 mm. O resultado foi renderizado na interface do usuário no RP3 para observar e analisar o funcionamento do código.

Além disso, foi impressa uma peça de teste com apenas uma camada, na qual essa forma de preenchimento foi aplicada. O modelo geométrico da peça é um paralelepípedo sólido de 50 x 30 x 3 mm. A impressão foi realizada na impressora 3D Cloner DH+ no laboratório do NUFER, na Sede Ecoville da UTFPR Campus Curitiba. Neste teste, os trechos finos foram feitos com distância de 1 mm, enquanto que os trechos grossos foram gerados com distância de 3 mm. O *gap* entre as linhas foi de 0,18 mm.

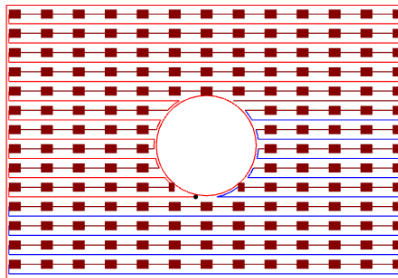
RESULTADOS E DISCUSSÃO

A criação das linhas de largura variável acaba gerando uma grande quantidade de pontos. A Figura 7 mostra uma camada da peça na qual foi aplicado o algoritmo de divisão de linhas. Para o cálculo das linhas de largura variada em todas as camadas da peça (79 camadas iguais), foram calculados e inseridos na trajetória de preenchimento um total de 27334 pontos.

Logo, para não precisar armazenar essa quantidade de pontos, deve ser aplicado o algoritmo sobre o preenchimento apenas quando essas linhas de largura variada forem efetivamente necessárias. Por exemplo, quando for renderizar o preenchimento na interface com o usuário, percorre-se a lista de pontos do preenchimento (este ainda sem linhas de largura variável), e quando é encontrado uma linha de *raster* que deve ser dividida, é feita a chamada da função que a fragmenta em uma lista de pontos, como já foi descrito neste artigo. Assim, essa lista é renderizada na interface com o usuário, e logo em

seguida é desalocada da memória. O mesmo serve para a geração do G-code (código a ser enviado a impressora) ao final do planejamento do processo.

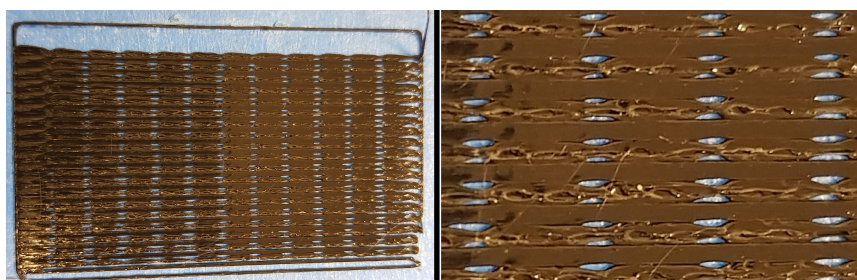
Figura 7 - Exemplo de camada na qual foi aplicado o algoritmo de divisão de linhas.



Fonte: Autor

A Figura 8 apresenta o resultado da impressão da camada utilizando o algoritmo, mostrando que o G-code gerado estava correto. Pode ser observado que houve aderência lateral das linhas impressas nos trechos grossos, e a formação de poros na região dos trechos finos.

Figura 8- Resultado da peça impressa (lado esquerdo) e aproximação na região dos poros (lado direito).



Fonte: Autor.

CONCLUSÃO

O algoritmo implementado para geração de um preenchimento poroso foi eficiente em segmentar e definir parâmetros de processo adequados aos mesmos. O G-code gerado pelo programa foi executado pela impressora de acordo com o programado.

Pesquisas ainda estão sendo realizadas em cima desse algoritmo porque ele calcula apenas as posições dos trechos finos e grosso, ficando a cargo do usuário o controle das configurações de extrusão e velocidade do bico da impressora na confecção dos trechos finos e grossos. Os melhores parâmetros para a confecção do preenchimento poroso ainda estão sendo estudados.

Além disso, o algoritmo permite o cálculo para todas as camadas sem a rotação do ângulo de *raster* entre elas, gerando camadas idênticas nas peças. É desejado, no futuro, que seja possível a confecção de peças com camadas que variem o ângulo entre si, e que os poros sejam alinhados a fim de obter materiais porosos tridimensionais com diferentes configurações.

AGRADECIMENTOS

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela bolsa concedida. Aos professores Rodrigo Minetto e Ricardo Dutra pelas orientações e melhorias indicadas durante a implementação do algoritmo. Ao nosso orientador Prof. Neri Volpato pela confiança em nós para a realização dessa pesquisa.

REFERÊNCIAS

FAUST, R.C. **Método paralelo de cálculo da trajetória de preenchimento em zigue-zague para manufatura aditiva**. 2019. Dissertação (Mestrado em Engenharia Mecânica e de Materiais) - Universidade Tecnológica Federal do Paraná, Curitiba, 2019. Disponível em:
<https://repositorio.utfpr.edu.br/jspui/handle/1/4772>. Acesso em: 30 ago. 2020.

NIELD, D. A.; BEJAN, A. **Convection in Porous Media**. 3. ed. New York: Springer, 2013.

SHIGUEOKA, M. O. **Desenvolvimento de estratégias de preenchimento para obter meios porosos tipo Lattice por manufatura aditiva por extrusão de material com análise de permeabilidade**. 2019. Dissertação (Mestrado em Engenharia Mecânica e dos Materiais) – Universidade Tecnológica Federal do Paraná, Curitiba, 2019. Disponível em:
<http://repositorio.utfpr.edu.br/jspui/handle/1/4636>. Acesso em: 25 ago. 2020.