

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA MECÂNICA**

Vitor Lora Tormen

**IMPLEMENTAÇÃO EM PROGRAMAÇÃO PARAMETRIZADA CNC
PARA USINAGEM DE SUPERFÍCIES DE BEZIER**

Santa Maria, RS
2020

Vitor Lora Tormen

**IMPLEMENTAÇÃO EM PROGRAMAÇÃO PARAMETRIZADA CNC PARA
USINAGEM DE SUPERFÍCIES DE BEZIER**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Mecânica, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Engenheiro Mecânico.**

Orientador: Leandro Costa De Oliveira

Santa Maria, RS
2020

Vitor Lora Tormen

**IMPLEMENTAÇÃO EM PROGRAMAÇÃO PARAMETRIZADA CNC PARA
USINAGEM DE SUPERFÍCIES DE BEZIER**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Mecânica, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Engenheiro Mecânico.**

Aprovado em 18 de novembro de 2020:

Leandro Costa De Oliveira, Dr. Eng. (DEM/UFSM)
(Presidente/Orientador)

Alexandre Aparecido Buenos, Dr. Eng. (DEM/UFSM)

Marco Aurélio da Fontoura Gonçalves, Dr. Eng. (CTISM/UFSM)

Santa Maria, RS
2020

RESUMO

IMPLEMENTAÇÃO EM PROGRAMAÇÃO PARAMETRIZADA CNC PARA USINAGEM DE SUPERFÍCIES DE BEZIER

AUTOR: Vitor Lora Tormen

ORIENTADOR: Leandro Costa De Oliveira

Dentre os meios convencionais de usinagem e seus modos de programação, pode-se considerar para máquinas de comando numérico computadorizado que a linguagem manual de programação, mais conhecida como linguagem G, e a programação de Desenho assistido por computador/Manufatura assistida por computador são as mais usuais para sua lógica de operação. Sabendo que estas máquinas operam segundo determinadas variáveis e operações algébricas, é possível combinar modos de programação para criar um programa que reduza a quantidade de linhas de código, com determinadas lógicas, tal se fosse programado em linguagem manual ou programação CAD/CAM. Visto que estas máquinas operam segundo três ou mais eixos, este modo de linguagem combinada, também conhecida como linguagem parametrizada, é o escopo deste trabalho. Esta será baseada para programar o deslocamento da ferramenta de usinagem a partir da forma bruta inicial, através de superfícies de Bézier, o que acabará gerando uma movimentação determinante segundo as lógicas desejadas do código concebido, para cumprir fielmente a operação segundo esta forma geométrica idealizada.

Palavras-chave: Usinagem Parametrizada. Máquina CNC. Superfície Bézier.

ABSTRACT

IMPLEMENTATION IN PARAMETERIZED CNC PROGRAMMING FOR BEZIER SURFACE MACHINING

AUTHOR: Vitor Lora Tormen

ADVISOR: Leandro Costa De Oliveira

Among the conventional means of machining and their programming modes, it can be considered for computer numerically controlled machines that the manual programming language, better known as G language, and Computer-aided design / Computer-aided manufacturing are the most usual for its operating logic. Knowing that these machines operate according to certain variables and algebraic operations, it is possible to combine programming modes to create a program that reduces the number of lines of code, with certain logics, as if it were programmed in manual language or CAD / CAM. Since these machines operate along three or more axes, this combined language mode, also known as parameterized language, is the scope of this work. This will be based on programming the displacement of the machining tool from the initial raw form, through Bézier surfaces, which will end up generating a decisive movement according to the desired logic of the designed code, to faithfully carry out the operation according to this idealized geometric shape.

Keywords: Parametrized machining. CNC machine. Bezier Surface

LISTA DE FIGURAS

Figura 1 - Máquina fresadora NC controlada por fita perfurada	13
Figura 2 – Máquina CNC de grande porte	14
Figura 3 – Exemplo de programa CLFILE para uma determinada operação de usinagem.....	16
Figura 4 – Exemplo de modelo de usinagem em CAD e arquivo CLDATA da trajetória da ferramenta para a usinagem do modelo em software CAM.....	25
Figura 5 – Exemplo de programação em software para usinagem em CAD/CAM....	26
Figura 6 – Representação de superfície de bezier com seus pontos de controle	28
Figura 7 – Trajetória da ferramenta de topo esférico	30
Figura 8 – Análise do posicionamento dos vetores de referência da ferramenta de usinagem.....	31
Figura 9 – Trajetória da ferramenta de desbaste segundo os pontos CC e CL.....	32
Figura 10 – Tolerância para cálculo dos programas CNC.....	33
Figura 11 – Interpolação Linear(G1) e B-Spline nativa	33
Figura 12 – Representação da escala da superfície idealizada pelos pontos de localização e Implementação do código de usinagem no computador CNC.....	37

LISTA DE QUADROS

Quadro 1 – Exemplo de programação APT para uma forma geométrica definida	17
Quadro 2 – Códigos Padronizados ISO para programação	20
Quadro 3 - Endereçamento ISO 1056 para as Funções G	21
Quadro 4 – Endereçamento das funções M.....	22
Quadro 5 – Representação e significado dos Operadores de Macros principais	27
Quadro 6 – Código de programação CNC para usinagem do caso de estudo.....	36
Quadro 7 – Código parametrizado para superfície de Bézier.....	39

SUMÁRIO

1 INTRODUÇÃO	9
1.1 OBJETIVOS DO TRABALHO.....	11
1.2 OBJETIVOS ESPECÍFICOS DO TRABALHO.....	11
2 REVISÃO BIBLIOGRÁFICA	13
2.1 HISTÓRIA DAS MÁQUINAS DE COMANDO NUMÉRICO	13
2.2 LINGUAGEM DAS MÁQUINAS DE COMANDO NUMÉRICO	15
2.2.1 Linguagem APT	15
2.2.2 Linguagem EIA/ISO	18
2.2.3 Linguagem Interativa	19
2.2.4 Variáveis de endereçamento da máquina	19
2.3 MODOS DE PROGRAMAÇÃO	23
2.3.1 Programação manual	23
2.3.2 Programação CAD/CAM.....	24
2.3.3 Programação Parametrizada	27
2.4 SUPERFÍCIES PARAMÉTRICAS	28
2.5 TRAJETÓRIA DA FERRAMENTA	29
3 METODOLOGIA	35
4 RESULTADOS	39
5 CONCLUSÃO	45
REFERÊNCIAS BIBLIOGRÁFICAS	46

1 INTRODUÇÃO

Ao longo da história o ser humano tem utilizado dos materiais que lhe eram disponíveis no seu meio de existência para transformá-los ou conformá-los para gerar as mais diversas ferramentas para uso de sobrevivência e alteração de qualidade de vida para si e a sociedade no qual estava inserido.

Com o passar do tempo, o ser humano foi criando os mais diversos modos de ferramentaria e máquinas propriamente ditas, principalmente devido à revolução industrial, o que acabou gerando um melhor aproveitamento do esforço físico requerido para gerar os produtos finais desejados. A partir do século XIX é quando surgiu o conceito de Usinagem, como pode-se considerar a utilização de um dispositivo máquina-ferramenta para gerar operações de corte e desbaste em uma matéria-prima inicial para gerar uma forma final desejada.

O processo de usinagem de materiais é desenvolvido de diversas maneiras, seja pela ferramenta de corte, tirando material de forma paralela à peça de trabalho, em operações em que se gerará cavaco a partir de uma ferramenta para torneamento ou fresamento agindo tangencialmente em suas faces, seja com o eixo da ferramenta perpendicular à superfície da peça, gerando furos na mesma.

A usinagem por revolução é um dos principais processos, onde uma peça gira em torno de um eixo principal de rotação da máquina e a ferramenta se desloca simultaneamente segundo uma trajetória no mesmo sentido com o referido eixo da peça (SOUZA, 2007).

Para facilitar a operação de usinagem, existe o comando numérico computadorizado (CNC), o qual auxilia automatizando as máquinas-ferramentas, fazendo com que não seja necessária a interferência do operador durante a usinagem, e que perfis com formatos difíceis de serem usinados de forma convencional possam ser obtidos (OLIVEIRA, 2013).

Outra grande vantagem do CNC é a possibilidade de produção de peças de geometria complexa, com bastante precisão, de forma repetitiva. Anteriormente, se conseguia através de processos convencionais a fabricação de determinadas peças graças à habilidade e paciência de alguns operadores de máquinas-ferramentas. Porém esta habilidade que antes era restrita a peças artesanais não conseguiria ser reproduzida de modo a conferir exatidão para peças complexas em várias situações consecutivas (OLIVEIRA, 1997).

Utilizando as ferramentas de usinagem em máquinas de CNC obtém-se um aproveitamento muito maior nas operações necessárias da indústria, visto que com a tecnologia disponível, os benefícios a serem conseguidos com a aplicação desse conjunto máquina-ferramenta são de grande relevância, garantindo alta precisão na fabricação de peças por remoção de material, assim como uma maior flexibilidade, qualidade e redução dos prazos de entrega dos produtos acabados.

A programação em máquinas CNC pode ser realizada de diversas maneiras, seja manualmente, com os dados de entrada e as informações de usinagem sendo colocadas a próprio conhecimento do programador para a ferramenta de usinagem se deslocar segundo os comandos básicos da máquina; ou via CAD/CAM, que possui programação baseada em desenhos de trajetórias, feita com base em um modelo externo já elaborado em um software de desenho assistido por computador (CAD) e segundo um modelo de manufatura via computador (CAM), onde se definirá as trajetórias de usinagem e demais funções de manufatura.

O desenvolvimento de um programa CNC é uma tarefa bastante minuciosa, pois envolve uma série de dados relacionados com a geometria da peça a ser produzida, com o tipo de máquinas a serem utilizadas, com as ferramentas disponíveis e ainda todos os princípios de usinagem para se conseguir um produto final que cumpra os seus requisitos. Sendo assim, é necessário que seja feita por pessoal com experiência e capacitação nesta área (OLIVEIRA, 1997).

Todo programa CNC começa a ser elaborado a partir de um desenho da peça a ser produzida. O programador, ou sistema auxiliar então deve definir a forma de fixação da peça, a sequência de operações que serão executadas, as ferramentas e as condições de corte que serão utilizadas.

A etapa posterior trata-se da codificação do programa que pode ser feita nos comandos da máquina CNC, geralmente de forma manual pelo programador e operador da máquina, ou então em uma linguagem do tipo APT (*Automatically Programmed Tools*), que definirá as geometrias, as ferramentas e os movimentos, as funções da máquina-ferramenta assim como os comandos do sistema computacional.

No caso de ser pela linguagem APT, a codificação do programa deve ser processada para a geração das posições da ferramenta, para criar então um arquivo chamado CLDATA, *Cutter Location Data*. É de interesse utilizar um recurso de simulação para verificar erros de sintaxe, assim como movimentações perigosas não percebidas durante a operação de usinagem. Sendo feita esta simulação, pode-se

prosseguir para os testes na máquina, fazendo, de preferência, inicialmente uma usinagem com uma peça de prova para posteriormente considerar o programa como pronto e aprovado (OLIVEIRA, 1997).

Nos capítulos subsequentes deste trabalho será abordado as principais formas de linguagem e programação para máquinas de comando numérico computadorizado e suas funcionalidades, assim como formas geométricas para basear a ferramenta de trabalho de usinagem e como uma usinagem com programação de forma parametrizada pode gerar um resultado de usinagem esperado.

1.1 OBJETIVOS DO TRABALHO

O escopo do trabalho é programar e idealizar ciclos de usinagem com função parametrizada baseando-se em superfícies matemáticas de Bézier para a máquina de comando numérico computadorizado Romi 308, localizado no Núcleo de Automação e Processos de Fabricação da Universidade Federal de Santa Maria.

Tendo como base as variáveis disponíveis no teclado da máquina de Comando Numérico Computadorizado para a operação da programação do código parametrizado, e então sabendo que para a geração da localização dos pontos de controle que a ferramenta de usinagem se baseará são necessárias 3 variáveis para os eixos de movimentação da ferramenta, sabe-se que o teclado disponibilizará 27 variáveis algébricas diferentes, correspondendo às letras do alfabeto do teclado, o que acabará causando uma restrição de, no máximo, 9 pontos de controle que a malha de Superfície de Bézier possuirá para a ferramenta de desbaste se basear.

Portanto, será idealizado o código parametrizado em função desta limitação, porém não alterará o resultado final que se esperará da usinagem parametrizada.

1.2 OBJETIVOS ESPECÍFICOS DO TRABALHO

- Contextualizar a história das máquinas de comando numérico e as linguagens que elas utilizam, caracterizando as utilidades e particularidades de cada uma.
- Relatar os modos de programação correntes e mais utilizados para geração dos códigos das máquinas de comando numérico computadorizado e como a escolha de cada um reflete em diferentes resultados e modos de trajetória para

a usinagem dos materiais de trabalho escolhidas.

- Introduzir o desenvolvimento das Superfícies de modelamento matemático e suas utilidades ao longo do tempo.
- Definir os códigos dos ciclos de usinagem necessários à realização da programação paramétrica baseada em superfície de Bézier.

2 REVISÃO BIBLIOGRÁFICA

Sabendo que as máquinas CNC são as mais usuais para as operações industriais exigidas na usinagem de materiais diversos, este capítulo abordará a sua origem e de suas linguagens, programações e normas que as mesmas utilizam e que são de primordial importância para a compreensão de seu funcionamento.

2.1 HISTÓRIA DAS MÁQUINAS DE COMANDO NUMÉRICO

O comando numérico computadorizado como conhecemos hoje teve seu princípio após o final da Segunda Guerra Mundial. O motivo de seu desenvolvimento foi devido à Força Aérea dos Estados Unidos, que estava com necessidade de conseguir padrões mais exatos na fabricação de suas peças. Para atender essa solicitação, uma empresa denominada *Parsons Works* se propôs a desenvolver um sistema servo-controlado por dados fornecidos via computador. A empresa veio a se associar ao Instituto de Tecnologia de Massachussetts para então apresentar seu primeiro sistema de comando numérico em 1952, uma fresadora de 3 eixos (ASM, 1999 apud MIRANDA 2009).

O surgimento do controle numérico (NC) de usinagem se deu na década de 1940, quando as calculadoras de cartão perfurado eram utilizadas para localizar os cortes que eram feitos manualmente (MARCICANO, 2017).

Após este período, o processo de perfuração de fita também foi utilizado como um dispositivo de entrada de dados, e os orifícios presentes do cartão deviam ser interpretados como valores numéricos para a máquina realizar os cortes automáticos, assim como mostrado na Figura 1.

Figura 1 - Máquina fresadora NC controlada por fita perfurada



O controle numérico computadorizado é uma fase posterior ao desenvolvimento do controle numérico, ele trocou o controle de *hardware* por controle por *software*. Então foram desenvolvidos dois tipos de sistemas: o controle numérico direto e o controle numérico computadorizado. O que diferia um de outro, é que no primeiro um único computador central controlava diversas máquinas e no segundo, cada máquina era equipada com seu próprio processador (MARCICANO, 2017).

Segundo Cassaniga (2000), o comando numérico computadorizado é um dispositivo eletrônico que recebe as informações sobre como a máquina-ferramenta ou um dispositivo irá realizar a operação através de linguagem própria denominada programa CNC, processando essas informações e devolvendo-as ao sistema via impulsos elétricos. Estes sinais são responsáveis pelo acionamento dos motores, que irão fazer com que a máquina movimente seu conjunto mesa-máquina-ferramenta, permitindo a realização de operações sem a necessidade de intervenção do operador no processo.

A programação CNC começa quando as características da peça estão sendo analisadas para determinar os processos de fabricação exigidos. Após fixa-se a peça na máquina, assim como as ferramentas, sequências e condições de usinagem. A Figura 2 apresenta uma máquina de CNC de grande porte.

Figura 2 – Máquina CNC de grande porte



Fonte: cadcamengineering.net.

Um código para uma máquina de comando numérico computadorizado é constituído por uma sequência de sentenças lógicas que conterão informações geométricas e tecnológicas para a máquina durante a operação de usinagem que será

realizada.

A maioria dos controles numéricos computadorizados seguem os códigos normalizados da ISO 1056 e da Associação Alemã de Normas Técnicas DIN 66025. Esses códigos, colocados em uma sequência lógica, permitem que a máquina-ferramenta execute os movimentos entre a ferramenta e a peça. Tal movimentação torna possível a usinagem de uma peça (SILVA, 2008).

2.2 LINGUAGEM DAS MÁQUINAS DE COMANDO NUMÉRICO

As linguagens das máquinas de comando numérico são as instruções que a mesma compreende para realizar as operações de movimentação do seu conjunto mesa-máquina-ferramenta, que então garantirá com que a usinagem da peça de trabalho seja feita corretamente (MARCICANO, 2017).

Sendo um conjunto de instruções que a máquina interpretará, dependendo da forma de sua estrutura lógica, a máquina realizará suas operações de forma única, fazendo com que diferentes combinações de variáveis algébricas para diferentes funções de posicionamento ou ativamento de funções causem uma movimentação mais específica ou determinante para uma característica de acabamento necessária na forma final, e então na superfície da peça que está sendo trabalhada.

Diferentes linguagens para as máquinas de comando numérico computadorizado foram criadas. Cada qual com suas vantagens e características, a serem detalhadas a seguir.

2.2.1 Linguagem APT

Em 1958, por intermédio da EIA (*Electronic Industries Association*), realizaram-se estudos no sentido de padronizar os tipos de linguagem utilizadas pelos fabricantes de máquinas. A linguagem padronizada desenvolvida destinada à programação de máquinas era a APT (*Automatically Programmed Tool*) desenvolvida pelo Instituto de Tecnologia de *Massachusetts*, em 1956 (GONÇALVES, 2007).

O programador, utilizando a linguagem APT, descreve o programa fonte, onde será definido a geometria da peça e/ou a trajetória da ferramenta em função das formas padrão de linguagem dos dados geométricos e funções auxiliares.

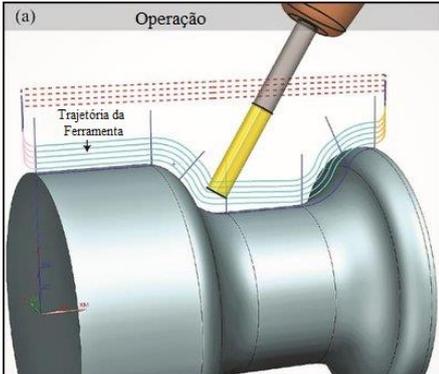
Na linguagem APT a programação é feita de modo que a peça esteja parada e

utiliza-se um sistema de coordenadas fixo da peça. A linguagem então faz com que a ferramenta de desbaste se mova a partir de segmentos de reta. Logo, se a trajetória que se deseja é curvilínea, ela será aproximada por um conjunto de segmentos de reta, com a precisão a ser determinada pelo programador (GONÇALVES, 2013).

A estrutura da linguagem APT consiste em declarações do tipo de geometria, movimento, pós-processamento e declarações auxiliares. Durante as definições de geometria, os elementos que serão utilizados para os desenhos são elementos como pontos, linhas, círculos entre outros, descrevendo dados de localização e construção para a definição correta. Na definição de movimento são feitos os comandos da ferramenta que ditam o tipo de ação que deve ser realizado e os dados que definem o movimento da ferramenta (GROOVER, 2010).

A função do processador da máquina de comando numérico é, através do programa fonte, realizar os cálculos geométricos, determinar o contorno da ferramenta e gerar um arquivo neutro CLDATA(*Cutter Location Data*) ou CLFILE(*Center line Data File*), conforme demonstrado na Figura 3. Depois então este arquivo é pós-processado, gerando um arquivo para a máquina específica (GONÇALVES, 2013).

Figura 3 – Exemplo de programa CLFILE para uma determinada operação de usinagem.

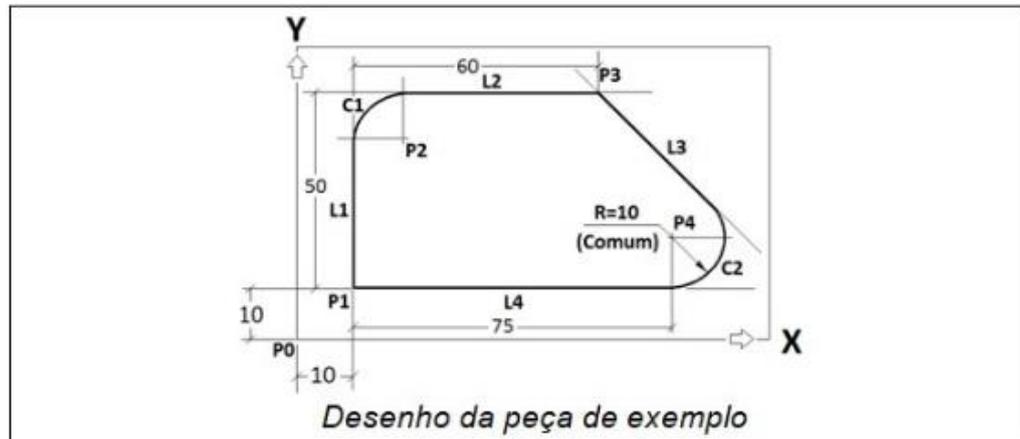
(a) Operação	(b) CL File					
	GOTO/70.2304	0.0000	39.9217	0.4804258	0.0000000	0.8770354
	GOTO/69.0912	0.0000	39.9831	0.4958416	0.0000000	0.8684130
	GOTO/67.9520	0.0000	40.0444	0.5111028	0.0000000	0.8595196
	GOTO/67.3607	0.0000	40.0746	0.5186738	0.0000000	0.8549722
	GOTO/66.7693	0.0000	40.1048	0.5262045	0.0000000	0.8503581
	GOTO/66.1782	0.0000	40.1577	0.5336941	0.0000000	0.8456776
	GOTO/65.5871	0.0000	40.2105	0.5411420	0.0000000	0.8409312
	GOTO/65.0757	0.0000	40.3074	0.5478189	0.0000000	0.8365970
	GOTO/64.5644	0.0000	40.4042	0.5544610	0.0000000	0.8322097
	GOTO/64.0742	0.0000	40.5441	0.5610680	0.0000000	0.8277697
	GOTO/63.5841	0.0000	40.6840	0.5676395	0.0000000	0.8232772
	GOTO/63.5134	0.0000	40.7686	0.5690736	0.0000000	0.8222866
	GOTO/63.4427	0.0000	40.8532	0.5705059	0.0000000	0.8212935
	GOTO/63.1134	0.0000	41.3164	0.5777371	0.0000000	0.8162229
	GOTO/62.7841	0.0000	41.7795	0.5849232	0.0000000	0.8110887
GOTO/62.4965	0.0000	42.2862	0.5920637	0.0000000	0.8058912	
GOTO/62.2089	0.0000	42.7929	0.5991579	0.0000000	0.8006309	

Fonte: Comak, 2018.

A linguagem APT fornece além de suas variáveis algébricas de programação, os seguintes elementos geométricos: pontos, conjuntos de pontos ou percursos, linhas, planos, vetores, cilindros, cones, esferas, superfícies quadráticas, curvas spline e até uma função para superfície esculpida(Por exemplo, superfície de Bezier). O Quadro 1 demonstra como fica o código para a usinagem de uma determinada

geometria segundo a programação APT (MARCICANO, 2017).

Quadro 1 - Exemplo de programação APT para uma forma geométrica definida



(continua)

APT NORMAL	Descrição
PARTNO / REGULAR APT	Nomeação ou numeração do programa
CLPRNT UNITS/ MM NOPOST	Unidades em milímetros
\$\$ GEOMETRY DEFINITION	Comentário para indicação das definições geométricas
P0= POINT/0.0,50	Ponto X=0, Y=0, Z=50
P1= POINT/10.10,0	Ponto X=10, Y=10, Z=0
P2= POINT/20.50,0	Ponto X=20, Y=50, Z=0
P3= POINT/70.60,0	Ponto X=70, Y=60, Z=0
P4= POINT/85.20,0	Ponto X=85, Y=20, Z=0
C1= CIRCLE/CENTER.P2,RADIUS,10	Círculo centro em P2 e raio 10
C2= CIRCLE/CENTER.P4,RADIUS,10	Círculo centro em P4 e raio 10
L1=LINE/P1,XSMAL,TANTO,C1	Linha por P1 e é tangente ao círculo C1, pelo lado do X menor
L2=LINE/P3,YLARG,TANTO,C1	Linha por P3 e é tangente ao círculo C1, pelo lado do Y maior
L3=LINE/P3,XLARG,TANTO,C2	Linha por P3 e é tangente ao círculo C2, pelo lado do X maior
L4=LINE/P1,YSMAL,TANTO,C2	Linha por P1 e é tangente ao círculo

Quadro 1 – Exemplo de programação APT para uma forma geométrica definida

(conclusão)

APT NORMAL	Descrição
	C2, pelo lado do Y menor
PL1=PLANE/P1,P2,P3	Plano 1 passa nos pontos P2, P2 e P3
PL2=PLANE/PARLEL,PL1,ZLARG,20	Plano 2 paralelo ao PL1 com 20mm de altura
\$\$END GEOMETRY DEFINITION	---
CUTTER/ 20,3	Ferramenta diâmetro 20mm e raio de canto 3mm
SPINDLE/ 3000,CW	Liga rotação no sentido horário a 3000 RPM
RAPID	Próximo movimento é rápido
FROM/ P0	Ponto de origem do programa P0
GOTO/ PL1	Vai em linha reta e rápido para o Plano 1
FEEDRAT/ 100	Liga avanço a 100mm por minuto
GOTO/ L1, PL1, L4	Vai até L1, PL1 e L4
TLLFT, GOLFT/ L1, TANTO, C1	A ferramenta está à esquerda da L1, vai para a esquerda até a tangente de C1
GOFWD/ C1, TANTO, L2	Vai em frente pelo C1 até a tangente da L2
GOFWD/ L2, PAST, L3	Vai em frente pelo L2 e passa por L3
GORGT/L3, TANTO, C2	Virar à direita, vai pela L3 até a tangente do C2
GOFWD/ C2, TANTO, L4	Vai em frente pelo C2 até a tangência da L4
GOFWD/ L4, PAST, L1	Vai em frente por L4 até passar por L1
RAPID	Próximo movimento é rápido
GO/ ON.PL2	Vai rápido até o plano PL2
FINI	Fim do programa

Fonte: Azevedo, 2019.

2.2.2 Linguagem EIA/ISO

A linguagem EIA/ISO, também conhecida como código G, é na atualidade a mais utilizada universalmente, tanto na programação manual, como na programação gráfica, onde é utilizado o CAM.

Os códigos EIA/ISO foram criados antes mesmo do aparecimento das máquinas CNC, eles eram usados nos escritórios em máquinas de escrever automáticas que utilizavam cartões perfurados(SCHNEIDER, 2015).

2.2.3 Linguagem Interativa

É uma linguagem por blocos parametrizados, possuindo blocos prontos e não se usando códigos propriamente ditos.

Ex. linguagem MAZATROL aplicando às máquinas MAZAK.

2.2.4 Variáveis de endereçamento da máquina

As letras e números que o teclado da máquina de comando numérico disponibiliza fazem parte dos comandos que a máquina interpretará para realizar a usinagem no material que será destinado ao desbaste. Segundo Oliveira (2011), podemos subdividir os elementos constituintes em:

Bloco de dados ou sentença: Conjunto de caracteres que têm o objetivo de informar ao comando as operações que devem ser realizadas. O caractere é a menor parte de um programa CNC e pode estar em três formas: algarismo, letra ou símbolo.

Letras de endereçamento: são instruções alfabéticas passadas para o comando que, por sua vez, podem executar um movimento ou simplesmente assumir uma nova função. Exemplo: G, L, K, I, etc.

Palavra: Uma palavra no código é uma combinação alfanumérica de caracteres, criando instruções para o sistema de controle. O número que segue a letra da palavra pode ser exclusivamente inteiro ou aceitar ponto decimal, bem como, pode não aceitar valor negativo ou aceitar, como ocorre em vários casos. Geralmente, as palavras indicam os eixos de posição, avanço, velocidade, comandos preparatórios, instruções miscelâneas e outras definições.

Quando os padrões de código para máquinas de controle numérico começaram a ser desenvolvidas, os fabricantes começaram a segui-las em grande medida. Uma das primeiras coisas a serem padronizadas foram os endereços de trabalho, que então seguem as 26 letras do alfabeto inglês, conforme o Quadro 2.

Quadro 2 – Códigos Padronizados ISO para programação

(continua)

A	Dimensão angular em torno do eixo X
B	Dimensão angular em torno do eixo Y
C	Dimensão angular em torno do eixo Z
D	Diâmetro da ferramenta
E	Dimensão angular em torno do eixo especial ou da segunda função de alimentação
F	Velocidade de avanço da ferramenta
G	Função preparatória
H	Comprimento da ferramenta
I	Centro do arco no eixo x
J	Centro do arco no eixo y
K	Centro do arco no eixo z
L	Função Local
M	Função miscelânea
N	Número de sequência
O	Número do programa
P	Terceira dimensão de avanço rápido ou dimensão de movimento terciário paralela a X
Q	Segunda dimensão de avanço rápido ou dimensão de movimento terciário paralela a Y
R	Primeira dimensão de avanço rápido ou dimensão de movimento terciário paralela a Z
S	Função da velocidade do eixo
T	Função da ferramenta
U	Dimensão de movimento secundário paralela a X
V	Dimensão do movimento secundário

Quadro 2 – Códigos Padronizados ISO para programação

(conclusão)

	Paralelo a Y
W	Dimensão de movimento secundário paralela a Z
X	Dimensão do movimento X primário
Y	Dimensão do movimento Y primário
Z	Dimensão primária do movimento Z

Fonte: Azevedo, 2019.

Algumas funções de importância a se citar na programação para usinagem em máquinas de Comando Numérico Computadorizado são as funções globais 'G' e as funções diversas ou Miscelâneas 'M'. Segundo Oliveira (2011), funções preparatórias são conhecidas como o código G, que tem como função ordenar ou ajustar o sistema de comando para obter uma desejada condição, modo ou estado de operação. O Quadro 3 apresenta uma lista com algumas funções preparatórias utilizadas na programação.

Quadro 3 – Endereçamento ISO 1056 para as funções G

(continua)

CÓDIGO G	FUNÇÃO
G00	Interpolação linear com avanço rápido
G01	Interpolação linear com avanço programado
G02	Interpolação circular sentido horário com avanço programado
G03	Interpolação circular sentido anti-horário com avanço programado
G04	Tempo de permanência
G17	Seleção do plano de trabalho XY
G18	Seleção do plano de trabalho ZX
G19	Seleção do plano de trabalho YZ
G28	Retorna a posição zero máquina
G40	Cancela compensação de raio de ferramenta

Quadro 3 – Endereçamento ISO 1056 para as funções G

(conclusão)

CÓDIGO G	FUNÇÃO
G41	Compensação de raio de ferramenta à esquerda
G42	Compensação de raio de ferramenta à direita
G43	Compensação de comprimento de ferramenta (positivo)
G44	Compensação de comprimento de ferramenta (negativo)
G53	Coordenada de trabalho em relação ao zero máquina
G54 a G59	Coordenadas de trabalho
G80	Cancelamento dos ciclos fixos
G81 a G89	Ciclos fixos
G90	Posicionamento absoluto
G91	Posicionamento incremental
G92	Estabelece sistema de coordenadas
G94	Avanço dado em milímetros por minutos
G95	Avanço em milímetros por rotação
G96	Rotação do fuso dado em velocidade de corte constante
G97	Rotação do fuso dado em RPM

Fonte: Groover, p. 155, 2011.

Quando fala-se de função diversa ou miscelânea para máquinas CNC se está dizendo sobre funções não cobertas pelos códigos G. Tal fato ocorre porque as funções M muitas vezes são exclusivas de alguma máquina particular. O Quadro 4 apresenta uma lista com funções miscelâneas mais comuns na programação (OLIVEIRA, 2011).

Quadro 4 – Endereçamento das funções M

(continua)

CÓDIGO M	FUNÇÃO
M00	Parada do programa
M01	Parada opcional do programa

Quadro 4 – Endereçamento das funções M

(conclusão)

CÓDIGO M	FUNÇÃO
M02	Fim de programa – mesma função do M30
M03	Liga eixo árvore sentido horário
M04	Liga eixo árvore sentido anti-horário
M05	Desliga eixo árvore
M06	Troca de ferramenta
M07	Liga fluído de corte alta pressão/ centro do eixo árvore
M08	Liga fluído de corte
M09	Desliga fluído de corte
M19	Orientação do eixo árvore
M30	Fim de programa

Fonte: Groover, p. 156, 2011.

2.3. MODOS DE PROGRAMAÇÃO

Como dito na introdução deste trabalho, há três tipos principais de formas a se programar em uma máquina de comando numérico computadorizado, a saber, programação manual, programação em formato CAD/CAM, e então a programação parametrizada.

2.3.1 Programação manual

Esta é a programação na qual os dados de entrada do programa e o código são colocados manualmente no computador da máquina pelo programador, sem nenhum recurso computacional externo a ser usado para se basear no código a se inserir na máquina CNC.

Segundo Gonçalves (2007), este tipo de programação é ideal para aplicações ponto a ponto em trabalhos de torneamento e fresamento que envolvam o processamento de peças geometricamente simples e que requeiram poucas operações.

2.3.2 Programação CAD/CAM

Sistemas CAD/CAM contribuem com a rapidez no processo de fabricação e automatização da programação de máquinas CNC para a manufatura do produto. Segundo Groover (2010), os sistemas têm vantagens como a possibilidade de simulação off-line, análise de tempo e custo de operação e otimização do caminho da ferramenta para reduzir tempos desnecessários na usinagem.

Quando for realizada usinagem através do sistema CAD/CAM/CNC, algumas etapas devem ser seguidas para se obter a forma final do modelo de forma desejada. Primeiro, deve ser feito um desenho do modelo geométrico em três dimensões que será gerado pelo sistema CAD. Após, deve-se enviar este modelo ao sistema CAM, onde serão feitas as trajetórias de usinagem que o operador do programa deverá informar, assim como demais funções de manufatura que o mesmo desejar.

Esta etapa de manufatura, escolha de ferramentas, parâmetros e trajetórias de usinagem é de suma importância, pois ela pode influenciar bastante no tempo de operação de usinagem e na qualidade superficial do modelo a ser feito. Depois de concluídos estas etapas, um programa NC será feito a partir destas características.

Este programa NC é gerado pelo sistema CAM que, a partir de suas características, pode gerar vários tipos de operações, tais como torneamento, eletroerosão, fresamento, entre outros. No processo de geração do programa NC se faz necessário adotar algumas tolerâncias que estarão relacionadas com a exatidão que a trajetória da ferramenta de usinagem seguirá o modelo criado pelo CAD e sendo necessário definir o método de interpolação selecionado para a escolha da trajetória de usinagem.

Na programação CNC, arcos correspondem a interpolações circulares, que podem se constituir no sentido horário (G2) ou anti-horário (G3). Como no caso da interpolação linear, o sentido depende da posição atual da ferramenta. No CAD, entretanto, os ângulos indicados para definir o arco são sempre no sentido anti-horário e apresentados em radianos (MALDANER, 2018).

Durante a programação do modelo em formato CAD/CAM se faz necessário estabelecer o ponto de origem das coordenadas requeridas para o deslocamento dos eixos. Nesta fase básica, cria-se o modelo sólido e o material a partir de blocos que representam a matéria-prima. Após, é necessário definir os parâmetros para a máquina produzir a peça, assim como definir o pós-processador a ser usado.

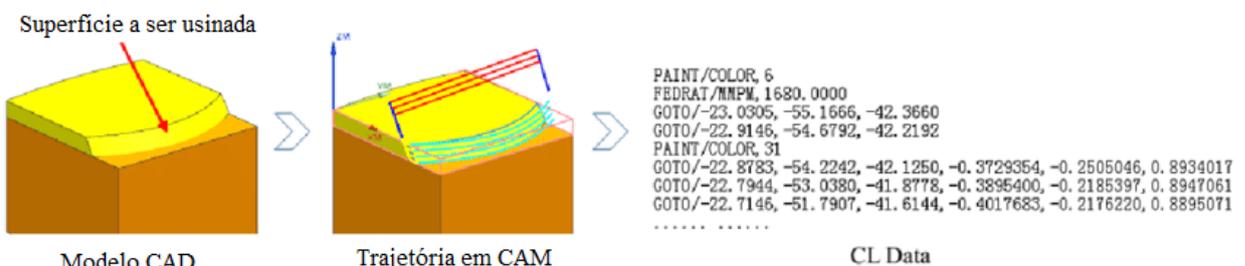
Um pós-processador é considerado uma ferramenta de apoio entre a fabricação auxiliada por computador e os sistemas de controle numérico das máquinas CNC, um tradutor, que então lê as instruções de fabricação emitidas por um sistema CAM e transcreve a linguagem gráfica gerada pelo sistema CAD/CAM em linguagem CNC relativa ao comando da máquina especificado.

A programação CNC em CAD/CAM foi gerada primeiramente para solucionar problemas que eram comuns durante a programação manual. Atualmente ela é reconhecida como um meio de programar tanto peças simples como de geometrias complexas, de forma mais rápida e precisa. As aplicações do CAD/CAM são de importância em indústria de moldes em geral, e aeroespacial, que trabalham com usinagem de peças complexas e que exigem bastante precisão (LYNCH, 1994).

Cada processo de usinagem é tratado de forma diferente um de outro, principalmente devido aos diferentes requisitos que são necessários. Por exemplo, para operação de desbaste de peça são necessários dados de avanço e velocidade relativa para a ferramenta que será destinada à usinagem.

Grande parte dos sistemas CAM gera arquivos de linguagem neutra que contêm instruções para uma máquina CNC. Estes estão em um formato binário chamado CLDATA ou ASCII (*American Standard Code for Information Interchange*), o qual é legível e geralmente escrito em linguagem APT(GONÇALVES, 2013). O formato ASCII aceita definições, geometrias simbólicas e instruções de usinagem, para depois gerar o código CLDATA que descreve uma operação de usinagem em condições absolutas, a exemplo da Figura 4.

Figura 4 – Exemplo de modelo de usinagem em CAD e arquivo CLDATA da trajetória da ferramenta para a usinagem do modelo em *software* CAM



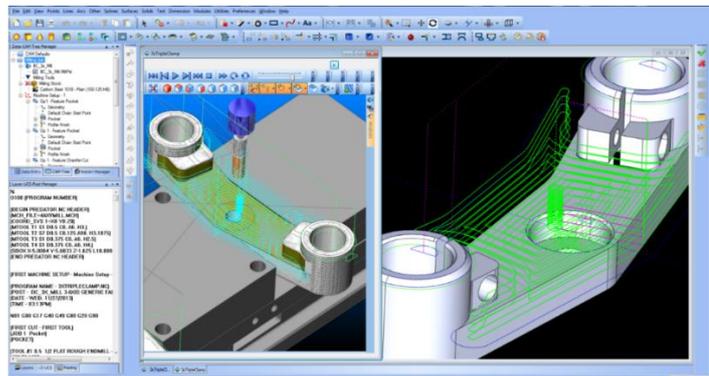
Fonte: Li et al., 2018.

Durante a formação do modelo em sistemas CAD/CAM, um meio de aprimorar

o sistema de manufatura e o processo de produção é transferir o programa para o CAPP (*Computer Aided Process Planning*), que faz o planejamento de processo auxiliado por computador. Assim, são inseridas informações auxiliares como definições de ferramentas, avanços, velocidades e estratégias de usinagem.

Definindo-se a tolerância com que a trajetória da ferramenta de usinagem irá seguir o modelo criado pela plataforma CAD, assim como o método de interpolação que será usado, o próximo passo é salvar os dados no sistema CAM para o pós-processador transformar estes dados para o CNC e o programa pronto será transferido para o comando, gerando um código legível para a máquina de comando numérico computadorizado, assim como definido na Figura 5.

Figura 5 – Exemplo de programação em *software* para usinagem em CAD/CAM



Fonte: BobCAD-CAM, 2015.

Segundo Monaro (2008), pode-se definir os principais tipos de interpolação para o programa em:

- Interpolação Linear: neste método o sistema CAM determina a trajetória da ferramenta com base em segmentos de retas para melhor representar a tolerância aplicada;
- Interpolação Circular: neste método, além da principal função de geração de segmentos de reta, são utilizados também arcos que possibilitam uma melhor adaptação à faixa de tolerância a ser utilizada. Os arcos permitem que a trajetória da ferramenta seja mais suave, ou seja, obtém-se uma melhor fidelidade de desbaste para superfícies complexas.
- Interpolação Polinomial: São utilizados segmentos de curvas, com base em modelos matemáticos, que geram uma trajetória da ferramenta mais suave e mais refinada considerando a trajetória do modelo idealizado.

2.3.3 Programação Parametrizada

A programação parametrizada pode ser comparada a qualquer linguagem de programação como as linguagens BASIC, C ou PASCAL, porém esta linguagem reside no controle do CNC e pode ser acessado ao nível do código G, podendo combinar técnicas de programação manuais com paramétricas.

Características como variáveis, aritmética, contadores, declarações de lógica e os ciclos de *loopings* que o código realizará estarão inseridos nesta linguagem, que então podem reduzir em muito o número de linhas e a chance de erro no código de programação, se o mesmo fosse realizado de forma manual pelo programador para o mesmo tipo de operação (Mundocnc.com.br).

Além de ter várias rotinas relacionadas ao computador, grande parte das versões de programação paramétrica têm rotinas relacionadas ao CNC com relativa profundidade. Pode-se citar como exemplo, macros que permitem ao usuário de CNC ter acesso a muitas propriedades do controle CNC (ferramenta de compensação, posicionamento do eixo, alarmes, geração e edição de código G). Estas coisas são impossíveis somente com utilização do código G comum, ou seja, somente funções manuais.

A função MACRO é utilizada quando se deseja trabalhar na programação de peças complexas, “famílias de peças” e outras funções especiais tais como: operações aritméticas, desvios condicionais, contador, comandos de comparação, etc. Os principais operadores de ciclos de macros são variáveis que representam uma inequação ou uma condição de uma ou mais operações que estão contidas na linha de código de um programa de usinagem e que então tem suas formas e significados principais como sendo exemplificados no Quadro 5.

Quadro 5 – Representação e significado dos Operadores de Macro principais

Operador	Significado
EQ	Igual a
NE	Não igual a (Diferente de)
GT	Maior que
GE	Maior ou igual que
LT	Menor que
LE	Menor ou igual que

Fonte: Elaborado pelo Autor

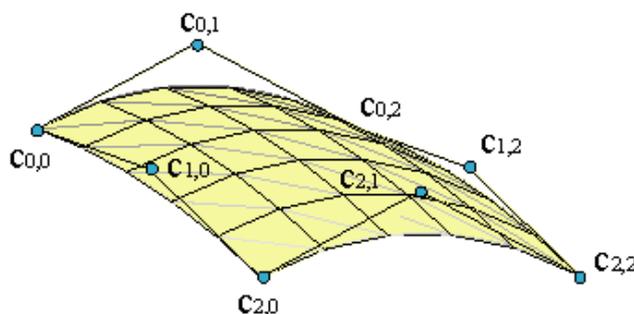
2.4 SUPERFÍCIES PARAMÉTRICAS

A representação paramétrica de curvas e superfícies representa uma ferramenta corriqueira em computação gráfica, principalmente em softwares CAD. Estas técnicas, que foram criadas para modelar fuselagens de aviões e perfis de automóveis, são usadas hoje em diversas áreas de softwares de produção computacional gráfica (TEIXEIRA, 2003).

Dentro do conjunto de superfícies paramétricas, uma de relevante aplicação é a superfície de Bézier. Pierre de Bézier foi um francês que trabalhava para a empresa Renault no século XX e começou a trabalhar com curvas paramétricas para sua utilização nos automóveis da empresa, o que hoje são chamadas curvas de Bézier.

Superfícies de Bézier, conforme apresentado na Figura 6, são produtos vetoriais de um polinômio de grau n . Como estas superfícies são definidas por pontos de controle, mudando as informações das coordenadas de quaisquer uns dos pontos pertencentes da superfície, esta se alterará, esticando-se a malha de controle da superfície.

Figura 6 – Representação de superfície de Bézier com seus pontos de controle



Fonte: JAMBRINA, 2020.

A representação paramétrica de superfícies é uma alternativa prática e conveniente para aplicação computacional. De forma idêntica à representação paramétrica de curvas, as superfícies são representadas por equações paramétricas independentes para cada dimensão. Nesse caso, o espaço paramétrico é bidimensional e a representação genérica de uma superfície tridimensional fica da forma:

$$\mathbf{S}(u, v) = \sum_{i=0}^m \sum_{j=0}^n P_{i,j} B_{i,m}(u) B_{j,n}(v) \quad (1)$$

em que $B(u, v)$ são as funções de peso nas direções u e v , $P_{i,j}$ os pontos de controle utilizados para representar a superfície e u e v são parâmetros que, normalmente, variam de 0 a 1, em uma superfície definida por $(m + 1) * (n + 1)$ pontos de controle. Considerando que se está a trabalhar com uma superfície com 9 pontos de controle, que é o caso de estudo, isso acabará gerando uma malha de 2x2 quadriláteros de superfície, significando que $m = 2$ e $n = 2$. As funções $B_{i,m}(u)$ e $B_{j,n}(v)$ ficam da forma,

$$B_{i,m}(u) = \frac{m!}{i!(m-i)!} * u^i * (1-u)^{m-i} \quad (2)$$

$$B_{j,n}(v) = \frac{n!}{j!(n-j)!} * v^j * (1-v)^{n-j} \quad (3)$$

Essa forma de representação relaciona dois espaços: o espaço paramétrico (bidimensional) e o espaço real (tridimensional). No espaço paramétrico, a superfície é um quadrado com uma unidade de lado. No espaço real, a superfície assume forma descrita nas equações paramétricas, porém, seu domínio permanece quadrilátero (TEIXEIRA, 2003).

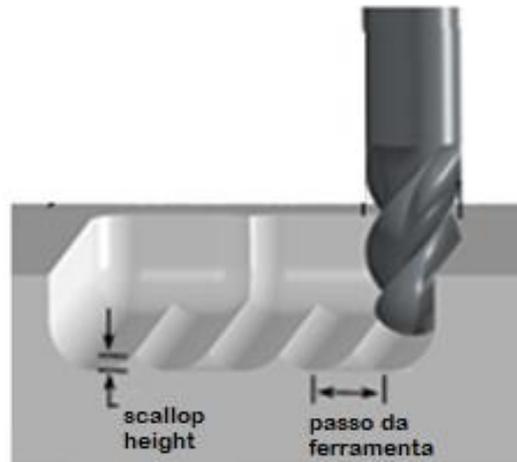
2.5 TRAJETÓRIA DA FERRAMENTA

Um dos parâmetros mais importantes a se considerar no processo de usinagem é o deslocamento da ferramenta de desbaste e como será sua trajetória, pois isso influenciará o modo como o material será retirado e como ficará a geometria final do produto acabado, considerando todas as suavidades, rugosidades e saliências presentes na superfície do material que está a ser trabalhado.

Quando se considera, por exemplo, ao caso de estudo deste trabalho, uma geometria da ferramenta de corte de topo esférico, durante a operação de fresamento, a ferramenta irá gerar na superfície da peça certas saliências com determinadas alturas, também chamadas de *scallop heights*, as quais podem ser de importância na hora de determinar a qualidade da superfície do produto acabado, conforme mostrado

na Figura 7.

Figura 7 – Trajetória da ferramenta de topo esférico



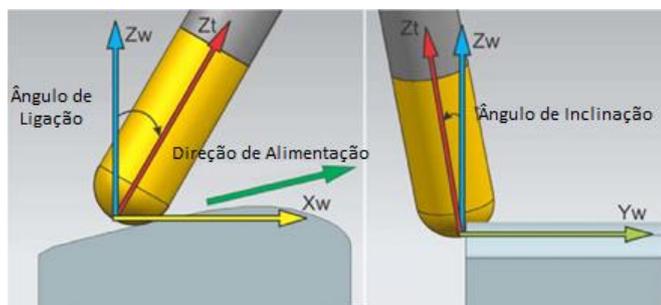
Fonte: CUSTOMPARTNET, 2008

O cálculo da trajetória da ferramenta deve ser considerado a principal função em softwares CAM, pois o resultado da operação de fabricação está diretamente relacionado com a qualidade da trajetória calculada, implicando alterações na superfície usinada, erros dimensionais assim como os tempos de cálculo de usinagem.

Dadas as informações de localização que a ferramenta de desbaste se baseará para percorrer sua trajetória, se faz necessário calcular a posição normalizada do seu vetor em relação à superfície da peça que será trabalhada, para a máquina realizar corretamente o deslocamento segundo as funções de posição do código parametrizado.

Considerando as informações das variáveis do código que a máquina de usinagem disponibiliza, assim como as informações da ferramenta de desbaste que a mesma possui, é possível determinar numericamente dentro das variáveis algébricas o posicionamento para gerar o vetor normalizado necessário da ferramenta, a exemplo da Figura 8, para cumprir fielmente a trajetória de desbaste segundo as superfícies de Bézier em código paramétrico, na peça de trabalho.

Figura 8 – Análise do posicionamento dos vetores de referência da ferramenta de usinagem



Fonte: Layegh, 2012.

Para a representação de geometrias em sistemas CAD e de controle de fabricação, são implementados equacionamentos matemáticos para desenvolver as suas funções. Isto se torna crucial na hora de gerar a melhor trajetória para a ferramenta de deslocamento de usinagem, principalmente devido ao fato que as geometrias em sistemas computacionais podem ser complexas, sendo necessários novos recursos de cálculos como os polinômios paramétricos (SOUZA E ULBRICH, 2009).

Pode-se dizer que os polinômios paramétricos são *splines*, ou um equacionamento capaz de representar uma curva em um sistema computacional. Estas funções podem ser classificadas como curvas de Bezier, *B-spline*, *Spline de Alkima*, NURBS (Non-Uniform Rational B-Spline), polinômios com diferentes graus de equação, dentre outros (GONÇALVES, 2013).

Estas curvas paramétricas tiveram sua origem pelo próprio Pierre Bezier, baseando-se primeiramente em Charles Hermite, sendo que então Bezier se baseia para criar seu software *Unisurf*, em 1972, com o objetivo de auxiliar nos painéis de automóveis produzidos pela Renault.

Essa foi a primeira utilização de sistemas computacionais para o modelar superfícies em projetos mecânicos. Hoje em dia, as teorias propostas por Bezier sofreram alterações, surgindo modelos *B-spline* e mais recentemente o NURBS, garantindo uma maior manipulação e controle da curva ou superfície gerada e conseqüentemente uma maior praticidade na representação de formas complexas (GONÇALVES, 2013).

Curvas *spline* são utilizadas com sucesso há décadas para representar curvas e superfícies em programas CAD. A vertente moderna é aplicar uma curva *Spline* para substituir uma interpolação de segmentos de reta quando se for requerida a forma de

descrever a trajetória da ferramenta (SOUZA E ULBRICH, 2009).

O modelamento das geometrias do modelo a ser concebido está ligado à descrição matemática compatível com o *software* a ser utilizado. A descrição matemática permite que a imagem do objeto seja exibida e manipulada em um terminal gráfico através de sinais de CPU do sistema CAD. Há diferentes técnicas para representar o objeto na modelagem geométrica, podemos classificar em três tipos principais:

2D: Representação bidimensional é usada para um objeto plano;

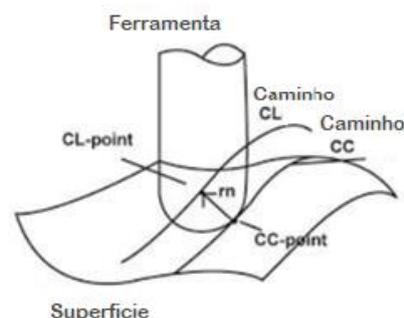
2½ D: Isso vai um pouco além da capacidade 2D, permitindo que um objeto tridimensional seja representado desde que não tenha detalhes na parede lateral;

3D: Permite modelagem tridimensional completa de uma geometria mais complexa.

Um parâmetro importante a se considerar quando se fala em trajetória de ferramenta, principalmente ao caso de estudo com ferramenta de desbaste com formato final semi-esférico, também conhecida como *ball-end-mill*, é que sua trajetória de desbaste no material bruto pode ser analisada matematicamente, considerando os pontos de contato com a superfície do material a ser trabalhado e o raio de topo da ferramenta.

Dito isso, pois se sabe que durante seu movimento, a ferramenta de desbaste irá interceptar a superfície do material original gerando o que se chama de *CC point*, ou *Cutter Contact point*, no qual é onde a ferramenta de forma semi-esférica tangencia o material bruto. Agora, para parâmetros de projeto e então a se considerar para programar a trajetória da ferramenta de desbaste, se faz necessário conhecer o valor da grandeza do raio da forma geométrica final da ferramenta para gerar o *CL point*, ou *Cutter Location point*, que será a trajetória que a ferramenta se deslocará segundo o centro de sua forma geométrica final, tal como demonstrado pela Figura 9.

Figura 9 – Trajetória da ferramenta de desbaste segundo os pontos CC e CL



Segundo Khan(2015), pode-se relacionar então a trajetória da ferramenta CL com a CC, a partir do conhecimento das variáveis do raio da ferramenta de desbaste e da direção normal, a partir da equação que segue

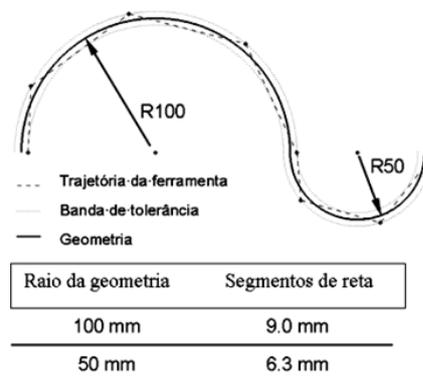
$$CL = CC + r * n \quad (4)$$

onde r é o raio de uma fresa de topo de esfera e n é o vetor normal no ponto CC.

Os algoritmos matemáticos e as metodologias para cálculo das trajetórias da ferramenta são criados em cada empresa desenvolvedora de *softwares* CAM, e então geralmente representam segredo industrial. O método tradicional de representar uma trajetória de ferramenta para usinagem de superfície complexa é ainda a interpolação linear de segmentos de retas, de acordo com a norma DIN 66025.

A movimentação da ferramenta de desbaste de usinagem segue trajetórias especificadas nos códigos que então se baseiam em segmentos de retas. Para a representação mais arredondada das curvas de desbaste na superfície da peça a ser trabalhada deverá ser diminuída a linearização do movimento para gerar o acabamento mais suave. Diferentes métodos de interpolação podem ser realizados para gerar as diferentes trajetórias que a ferramenta de usinagem se baseará. A Figura 10 representa uma trajetória linearizada baseada em segmentos de reta, dentro de uma banda de tolerância.

Figura 10 – Tolerância para cálculo dos programas CNC

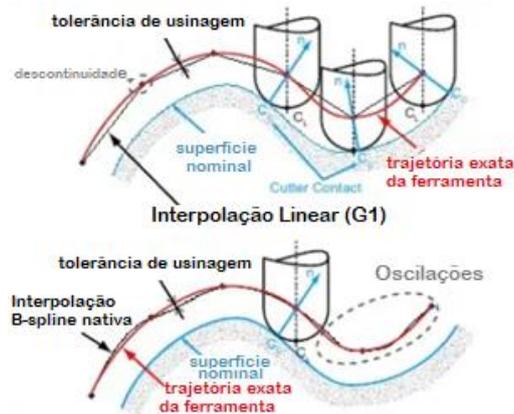


Fonte: Souza, 2004.

A possibilidade e os benefícios de se utilizar uma *spline* para representar trajetórias complexas de ferramenta acaba gerando um código de programa CNC que

não irá conter os comandos tradicionais G01, G02 e G03, mas sim uma nova sintaxe que representará uma trajetória complexa. A Figura 11 ilustra a trajetória de ferramenta e o polígono de controle que define a curva *spline*. Os pontos cartesianos são utilizados para definir o polígono(SOUZA, 2009).

Figura 11 – Interpolação Linear(G1) e *B-Spline* nativa



Fonte: Beudaert et al., 2014.

Há diversas sintaxes para as entradas de dados *Spline*, conforme o comando numérico a ser usado. Esses são gerados a partir de um programa CNC linear de segmentos de retas. Sendo então utilizada uma tolerância inicial para cálculo do programa linear adicionado à tolerância de cálculo do programa *Spline*(SOUZA, 2004; SOUZA E COELHO, 2007).

3 METODOLOGIA

A metodologia que este trabalho trata é sobre linguagens de programação aplicadas em códigos de usinagem para gerar as operações desejadas. Em específico neste trabalho será aplicado as teorias de usinagem parametrizada baseada em Superfície de Bezier, para utilizar-se das funções de endereçamento do código que a máquina interpretará, e então gerar um programa com determinadas lógicas algébricas e *loopings* para usinar as formas desejadas.

De forma a deixar mais transparente o significado e conexão das variáveis de construção de um código para máquina CNC se faz necessário explicar seus operadores básicos e funções lógicas essenciais para gerar um programa que cumpra fielmente um ciclo de usinagem proposto.

Quando uma operação começa seguindo um código padronizado para suas funções de movimento serem realizadas e então termina gerando um acabamento final na geometria da peça que estava para ser trabalhada no início da operação de usinagem dizemos que isto é um Ciclo de Usinagem, onde tivemos um desbaste do material inicial da peça, segundo uma função do código para a linguagem da máquina que então gerou a geometria final desejada.

A maioria dos operadores utilizados nos códigos de programação são “!” , “*”, “/” e por fim “+” e “-”. Os parênteses devem vir antes de qualquer um destes operadores, assim como os pontos representam a separação decimal dos números. Os operadores “+” e “-” só podem ser usados em uma expressão fechada por parênteses. E então todos os parênteses abertos devem ser fechados.

Os operadores são classificados em: Adição (+); Subtração (-); Multiplicação (*); Divisão (/); e Exponenciação (!).

Os operadores não podem aparecer sem os parênteses, de tal modo que a expressão “5!-3,” está escrita de forma incorreta, ao passo que a expressão “5!(-3),” está correta.

Se forem consideradas as linguagens de programação acima expostas, então aplicados ao caso de estudo deste trabalho ter-se-á a configuração de programação exposta na linguagem pelo Quadro 6, para o caso de uma superfície de Bezier idealizada. A Figura 12 mostra a implementação do código de usinagem no computador CNC.

Quadro 6 – Código de programação CNC para usinagem do caso de estudo

(continua)

G90

GXY

G39 A(-1) ;Coord. X de P00

B(0) ;Coord. Y de P00

C(-1) ;Coord. Z de P00

D(0) ;Coord. X de P01

F(0) ;Coord. Y de P01

I(-1) ;Coord. Z de P01

J(1) ;Coord. X de P02

K(0) ;Coord. Y de P02

L(-1) ;Coord. Z de P02

M(-1) ;Coord. X de P10

O(0) ;Coord. Y de P10

Q(0) ;Coord. Z de P10

R(0) ;Coord. X de P11

S(5) ;Coord. Y de P11

T(0) ;Coord. Z de P11

U(1) ;Coord. X de P12

V(0) ;Coord. Y de P12

W(0) ;Coord. Z de P12

G79 A(-1) ;Coord. X de P20

B(0) ;Coord. Y de P20

C(1) ;Coord. Z de P20

D(0) ;Coord. X de P21

F(0) ;Coord. Y de P21

I(1) ;Coord. Z de P21

J(1) ;Coord. X de P22

K(0) ;Coord. Y de P22

L(1) ;Coord. Z de P22

G79 Q(0) ;Contador entre 0 e 1

R(1) ;Contador entre 0 e 1

S(0) ;Passos

G79 U(1/GS) ;1 dividido pelo número de passos

N100 G39 X((1-GQ)!2) Y(2*(1-GQ)*GQ) Z(GQ!2) ;LX LY e LZ são armazenadas apenas para evitar posterior repetição

M(GE(GR-1)) H200 ;GR - 1 >= 0 então GR >= 1

M(LE(GR)) H400 ; GR <= 0

N900 G79 Q(GQ+GU)

M(LE(GQ-1)) H100 ; CQ - 1 <= 0 então CQ <= 1

H1000 ;Fim de programa

N200 G79 R(0)

Quadro 6 – Código de programação CNC para usinagem do caso de estudo

(conclusão)

```

N300 G79 X(((1-GR)!2*(LX*LA + LY*LD + LZ*LJ) + 2*(1-GR)*GR*(LX*LM + LY*LR +
LZ*LU) + GR!2*(LX*GA + LY*GD + LZ*GJ))
  G79 Y(((1-GR)!2*(LX*LB + LY*LF + LZ*LK) + 2*(1-GR)*GR*(LX*LO + LY*LS +
LZ*LV) + GR!2*(LX*GB + LY*GF + LZ*GK))
  G79 Z(((1-GR)!2*(LX*LC + LY*LI + LZ*LL) + 2*(1-GR)*GR*(LX*LQ + LY*LT +
LZ*LW) + GR!2*(LX*GC + LY*GI + LZ*GL))
G01 X(GX) Y(GY) Z(GZ)
G79 R(GR+GU)
  M(LE(GR-1)) H300 ;GR - 1 <= 0 então GR <= 1
G79 R(0)
G01 X(GX) Y(GY+0.2) Z(GZ)
G01 X(GX+(2*GU)) Y(GY+0.2) Z(GZ)
M(EQ(0)) H900

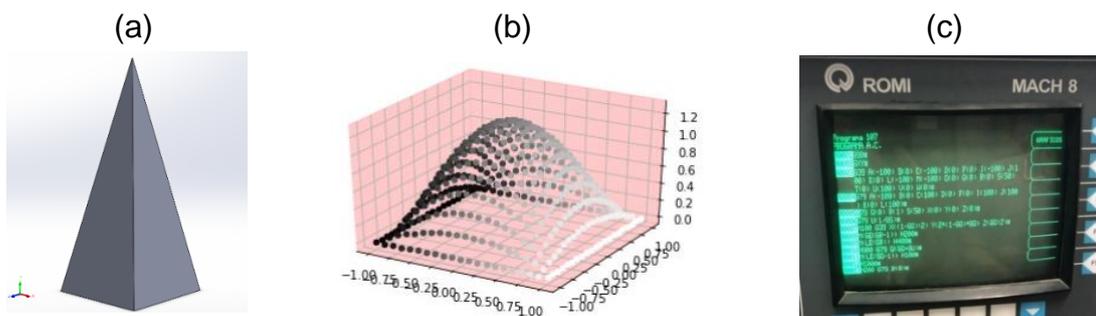
N400 G79 R(1)
N500 G79 X(((1-GR)!2*(LX*LA + LY*LD + LZ*LJ) + 2*(1-GR)*GR*(LX*LM + LY*LR +
LZ*LU) + GR!2*(LX*GA + LY*GD + LZ*GJ))
  G79 Y(((1-GR)!2*(LX*LB + LY*LF + LZ*LK) + 2*(1-GR)*GR*(LX*LO + LY*LS +
LZ*LV) + GR!2*(LX*GB + LY*GF + LZ*GK))
  G79 Z(((1-GR)!2*(LX*LC + LY*LI + LZ*LL) + 2*(1-GR)*GR*(LX*LQ + LY*LT +
LZ*LW) + GR!2*(LX*GC + LY*GI + LZ*GL))
G01 X(GX) Y(GY) Z(GZ)
G79 R(GR-GU)
  M(GE(GR)) H500 ; GR >= 0
G79 R(1)
G01 X(GX) Y(GY-0.2) Z(GZ)
G01 X(GX+(2*GU)) Y(GY-0.2) Z(GZ)
M(EQ(0)) H900

N1000 M2

```

Fonte: Elaborado pelo Autor.

Figura 12 – (a) Representação da escala da pirâmide idealizada pelos pontos de localização, (b) Superfície de Bezier gerada a partir da localização dos pontos de (a), (c) Implementação do código de usinagem no computador CNC.



Fonte: Elaborado pelo Autor.

Cada letra do alfabeto representada como variável de localização de um ponto no respectivo eixo de coordenadas está com um valor numérico agregado para representar a sua posição no espaço. Como são utilizados nove pontos de controle e necessitamos das informações dos eixos x, y, z para cada ponto, haverá a necessidade de utilizar no máximo as 27 variáveis, correspondendo a todas as letras do alfabeto disponíveis pela máquina CNC.

Quanto à informações importantes deste ciclo de programação, podemos dizer que a função G90 definirá as coordenadas absolutas, G39 A(-1)B(0) (define as variáveis LA=-1 e LB=0), G79 A(-1) B(0) (define as variáveis GA=-1 e GB=0). Ou seja, G39 define as variáveis “L” locais e G79 define as variáveis “G” globais, assim como G01 movimenta a ferramenta.

Durante a parte do código em que tem-se as funções N100 G39 X(((1-GQ)!2) Y(2*(1-GQ)*GQ) Z(GQ!2); assim como em N300 G79 X((1-GR)!2*(LX*LA + LY*LD + LZ*LJ) + 2*(1-GR)*GR*(LX*LM + LY*LR + LZ*LU) + GR!2*(LX*GA + LY*GD + LZ*GJ)) e N500 G79 X(((1-GR)!2*(LX*LA + LY*LD + LZ*LJ) + 2*(1-GR)*GR*(LX*LM + LY*LR + LZ*LU) + GR!2*(LX*GA + LY*GD + LZ*GJ)) foi implementada a equação de Bézier quadrática em x, y, z para as funções G e L do código. A equação a saber

$$B(t) = (1 - t^2)B_0 + 2t(1 - t)B_1 + t^2B_2, t \in [0,1]. \quad (5)$$

Onde o número de pontos de controle é n mais 1, t assume um valor tal que $t \in \mathbb{R}, 0 \leq t \leq 1$, B_i é o i-ésimo ponto de controle.

Quando se tem uma função M do tipo M(LE(GQ-1)) H200 será um teste para o programa, que se for válida a afirmação, irá para linha N200 e se for falso seguirá diretamente para a próxima linha. Neste caso o teste é LE que testa se o que está dentro é “menor ou igual a zero”, então se (GQ-1) for menor ou igual a zero o programa vai para a linha N200, seguindo as operações do código até M2, que significa o fim do programa.

4 RESULTADOS

Foi idealizada a usinagem em função paramétrica para o caso de estudo deste trabalho, onde as informações para a geração da superfície de Bezier foram ditadas de forma a criar um código com variáveis de localização dos pontos da superfície e movimentação da ferramenta seguindo um ciclo parametrizado.

De forma a exemplificar os conceitos de programação expostos acima, foi realizado um código para geração do programa segundo superfície de Bezier para uma condição parametrizada, assim como mostrado no Quadro 7.

Quadro 7 – Código parametrizado para superfície de Bézier

(continua)

G90	Define o sistema de coordenadas absolutas
GXY	Seleciona o plano XY
G39 A(-10) ;Coord. X de P00 B(0) ;Coord. Y de P00 C(-10) ;Coord. Z de P00 D(0) ;Coord. X de P01 F(0) ;Coord. Y de P01 I(-10) ;Coord. Z de P01 J(10) ;Coord. X de P02 K(0) ;Coord. Y de P02 L(-10) ;Coord. Z de P02 M(-10) ;Coord. X de P10 O(0) ;Coord. Y de P10 Q(0) ;Coord. Z de P10 R(0) ;Coord. X de P11 S(5) ;Coord. Y de P11 T(0) ;Coord. Z de P11 U(10) ;Coord. X de P12 V(0) ;Coord. Y de P12 W(0) ;Coord. Z de P12 G79 A(-10) ;Coord. X de P20 B(0) ;Coord. Y de P20 C(10) ;Coord. Z de P20 D(0) ;Coord. X de P21 F(0) ;Coord. Y de P21 I(10) ;Coord. Z de P21 J(10) ;Coord. X de P22 K(0) ;Coord. Y de P22 L(10) ;Coord. Z de P22	Nessa parte, são declaradas as variáveis para as coordenadas X, Y e Z dos 9 pontos de controle, totalizando 27 variáveis

Quadro 7 – Código parametrizado para superfície de Bézier

(continuação)

<p>Q(0) ;Contador entre 0 e 1 R(1) ;Contador entre 0 e 1</p>	<p>Declara os contadores. O contador “GQ”, por exemplo, inicia em 0, e a cada passe em X ele é incrementado em 1/GS, ou seja, 1/(número de passos em X). Quando esse contador chega em 1, significa que todos os passos em X já foram feitos. A mesma coisa acontece pra GR, só que para o eixo Y.</p>
<p>S(8) ;Passos em X W(8) ;Passos em Y</p>	<p>Número de passos em X e Y</p>
<p>T(1) ;Raio da ferramenta</p>	<p>Declara o raio da ferramenta</p>
<p>G79 X(LA) Y(LB) Z(LC-GT) N100 G39 X((1-GQ)!2) Y(2*(1-GQ)*GQ) Z(GQ!2)</p>	<p>Declara as variáveis GX, GY, GZ como sendo igual ao primeiro ponto, que tem as coordenadas LA, LB e LC, porém é feito um deslocamento do raio da ferramenta na direção Z pra não haver contato entre ferramenta e peça inicialmente. Essas variáveis LX, LY e LZ são calculadas agora pois seus valores mudam somente quando GQ muda, ou seja, somente quando a ferramenta vai para um novo passe em X. Aqui foi implementada as equações de Bezier quadráticas para os eixos x,y,z das funções L e G do código</p>
<p>M(GE(GR-1)) H200 ;GR - 1 >= 0 então GR >= 1 M(LE(GR)) H400 ;GR <= 0</p>	<p>Testa se GR é maior ou igual que 1. Caso positivo, vai para bloco N200 Testa se GR é menor ou igual a zero. Caso positivo, vai para o bloco N400</p>
<p>N900 G79 Q(GQ+(1/GS))</p>	<p>Incrementa o contador GQ. O</p>

Quadro 7 – Código parametrizado para superfície de Bézier

(continuação)

	programa é direcionado para esse bloco N900 depois que completa um passe em X
M(LE(GQ-1)) H100 ; se GQ - 1 <= 0 então GQ <= 1 H1000 ;Fim de programa	Testa se GQ é menor ou igual a 1. Se sim, volta pro bloco N100 pra ser executado outro passo em X. Caso contrário, vai pro bloco N1000
N200 G79 R(0)	Veio para esse bloco pois GR era maior ou igual que 1. É colocado o valor de GR igual a zero.
N300 G79 M(GY) O(GZ) X(((1-GR)!2*(LX*LA + LY*LD + LZ*LJ) + 2*(1-GR)*GR*(LX*LM + LY*LR + LZ*LU) + GR!2*(LX*GA + LY*GD + LZ*GJ)) Y(((1-GR)!2*(LX*LB + LY*LF + LZ*LK) + 2*(1-GR)*GR*(LX*LO + LY*LS + LZ*LV) + GR!2*(LX*GB + LY*GF + LZ*GK)) Z(((1-GR)!2*(LX*LC + LY*LI + LZ*LL) + 2*(1-GR)*GR*(LX*LQ + LY*LT + LZ*LW) + GR!2*(LX*GC + LY*GI + LZ*GL)) U(((1-GR)!2*(((1-(GQ-1/GS))!2)*LA + (2*(1-(GQ-1/GS))*(GQ-1/GS))*LD + ((GQ-1/GS)!2)*LJ) + 2*(1-GR)*GR*(((1-(GQ-1/GS))!2)*LM + (2*(1-(GQ-1/GS))*(GQ-1/GS))*LR + ((GQ-1/GS)!2)*LU) + GR!2*(((1-(GQ-1/GS))!2)*GA + (2*(1-(GQ-1/GS))*(GQ-1/GS))*GD + ((GQ-1/GS)!2)*GJ)) V(((1-GR)!2*(((1-(GQ-1/GS))!2)*LB + (2*(1-(GQ-1/GS))*(GQ-1/GS))*LF + ((GQ-1/GS)!2)*LK) + 2*(1-GR)*GR*(((1-(GQ-1/GS))!2)*LO + (2*(1-(GQ-1/GS))*(GQ-1/GS))*LS + ((GQ-1/GS)!2)*LV) + GR!2*(((1-(GQ-1/GS))!2)*GB + (2*(1-(GQ-1/GS))*(GQ-1/GS))*GF + ((GQ-1/GS)!2)*GK)) G01 X(((((-(GO-GZ))*(GV-	As variáveis GM e GO servem para armazenar os valores de GY e GZ que foram calculados anteriormente

Quadro 7 – Código parametrizado para superfície de Bézier

(continuação)

$GY)) + ((GM - GY) * (GZ - GZ)) + GX) - GX) * GT / (\sqrt{(((- ((GX - GX) * (GZ - GZ)) + ((GO - GZ) * (GU - GX))) + GY) - GY)!^2 + (((- ((GO - GZ) * (GV - GY)) + ((GM - GY) * (GZ - GZ)) + GX) - GX)!^2))}) + GX)$	<p>[GX, GO, GM] são as coordenadas do próximo ponto que a ferramenta deve ir se a mesma fosse um ponto. [GU, GZ, GV] são as coordenadas do ponto que representa o fim do vetor normal, com a origem em [GX, GO, GM].</p>
$Y((((- ((GM - GY) * (GU - GX)) + ((GX - GX) * (GV - GY))) + GZ) - GZ) * GT / (\sqrt{(((- ((GX - GX) * (GZ - GZ)) + ((GO - GZ) * (GU - GX))) + GY) - GY)!^2 + (((- ((GM - GY) * (GU - GX)) + ((GX - GX) * (GV - GY))) + GZ) - GZ)!^2})) + GZ)$ $Z((((- ((GX - GX) * (GZ - GZ)) + ((GO - GZ) * (GU - GX)) + GY) - GY) * GT / (\sqrt{(((- ((GX - GX) * (GZ - GZ)) + ((GO - GZ) * (GU - GX))) + GY) - GY)!^2 + (((- ((GM - GY) * (GU - GX)) + ((GX - GX) * (GV - GY))) + GZ) - GZ)!^2})) + GY)$	<p>É feito um produto vetorial para determinar as coordenadas do ponto (GX, GY, GZ) que a ferramenta deve se deslocar, considerando um deslocamento de GT (raio da ferramenta) na direção do vetor normal. A ferramenta é deslocada para esse ponto já que a função G01 foi usada.</p>
<p>G79 R(GR+(1/GW))</p>	<p>O contador GR é incrementado em 1/GW, ou seja, 1/(número de passos em Y)</p>
<p>M(LE(GR-1)) H300 ;GR - 1 <= 0 então GR <= 1</p>	<p>Se GR for menor ou igual que 1 significa que a ferramenta ainda precisa ser deslocada pra mais um ponto no mesmo passe em X, por isso, volta pro bloco N300. Caso contrário, passa pra próxima linha, pois o passe em X terminou.</p>
<p>G79 R(0)</p>	<p>GR = 0</p>
<p>G01 X(GX) Y(GZ) Z(GY)</p>	<p>A ferramenta é deslocada para o primeiro ponto que vai iniciar o próximo passe em X. E esse ponto tem as mesmas coordenadas Y e Z.</p>

Quadro 7 – Código parametrizado para superfície de Bézier

(continuação)

<p>G01 X(GX+(2*1/GS)) Y(GZ) Z(GY) G79 Y(((1-GR)!2*(LX*LB + LY*LF + LZ*LK) + 2*(1-GR)*GR*(LX*LO + LY*LS + LZ*LV) + GR!2*(LX*GB + LY*GF + LZ*GK))</p>	<p>São calculados os pontos GY e GZ novamente para serem armazenados nas variáveis GM e GO posteriormente.</p>
<p>G79 Z(((1-GR)!2*(LX*LC + LY*LI + LZ*LL) + 2*(1-GR)*GR*(LX*LQ + LY*LT + LZ*LW) + GR!2*(LX*GC + LY*GI + LZ*GL) + GT) M(EQ(0)) H900</p>	<p>Volta para o bloco N900</p>
<p>N400 G79 R(1) N500 G79 M(GY) G79 O(GZ) G79 X(((1-GR)!2*(LX*LA + LY*LD + LZ*LJ) + 2*(1-GR)*GR*(LX*LM + LY*LR + LZ*LU) + GR!2*(LX*GA + LY*GD + LZ*GJ)) G79 Y(((1-GR)!2*(LX*LB + LY*LF + LZ*LK) + 2*(1-GR)*GR*(LX*LO + LY*LS + LZ*LV) + GR!2*(LX*GB + LY*GF + LZ*GK)) G79 Z(((1-GR)!2*(LX*LC + LY*LI + LZ*LL) + 2*(1-GR)*GR*(LX*LQ + LY*LT + LZ*LW) + GR!2*(LX*GC + LY*GI + LZ*GL)) G79 U(((1-GR)!2*(((1-(GQ-1/GS))!2)*LA + (2*(1-(GQ-1/GS)))*(GQ-1/GS))*LD + ((GQ-1/GS)!2)*LJ) + 2*(1-GR)*GR*(((1-(GQ-1/GS))!2)*LM + (2*(1-(GQ-1/GS)))*(GQ-1/GS))*LR + ((GQ-1/GS)!2)*LU) + GR!2*(((1-(GQ-1/GS))!2)*GA + (2*(1-(GQ-1/GS)))*(GQ-1/GS))*GD + ((GQ-1/GS)!2)*GJ)) G79 V(((1-GR)!2*(((1-(GQ-1/GS))!2)*LB + (2*(1-(GQ-1/GS)))*(GQ-1/GS))*LF + ((GQ-1/GS)!2)*LK) + 2*(1-GR)*GR*(((1-(GQ-1/GS))!2)*LO + (2*(1-(GQ-1/GS)))*(GQ-1/GS))*LS + ((GQ-1/GS)!2)*LV) + GR!2*(((1-(GQ-1/GS))!2)*GB + (2*(1-(GQ-1/GS)))*(GQ-1/GS))*GF + ((GQ-1/GS)!2)*GK)) G01 X(-((((-(GO-GZ)*(GV-GY))+((GM-GY)*(GZ-GZ)))+GX)-GX)*GT/(sqrt((((-(GX-GX)*(GZ-GZ))+((GO-GZ)*(GU-GX)))+GY)-</p>	<p>A explicação é a mesma que anteriormente, dada a partir do bloco N200. Nesse caso, muda apenas o fato de que o passe em X é iniciado no final e se desloca para o início, de modo que a ferramenta faça um “Zig Zag” e não tenha que voltar para o ponto inicial toda vez que um novo passe é iniciado.</p>

Quadro 7 – Código parametrizado para superfície de Bézier

(conclusão)

<pre> GY!2+(((-(GO-GZ)*(GV- GY))+((GM-GY)*(GZ-GZ)))+GX)- GX!2))+GX) Y(((((-(GM-GY)*(GU-GX))+((GX- GX)*(GV-GY))+GZ)-GZ)*GT/(sqrt(((-(GX-GX)*(GZ-GZ))+((GO-GZ)*(GU- GX))+GY)-GY)!2+(((-(GM-GY)*(GU- GX))+((GX-GX)*(GV-GY))+GZ)- GZ)!2))+GZ) Z(((((-(GX-GX)*(GZ-GZ))+((GO- GZ)*(GU-GX))+GY)-GY)*GT/(sqrt(((-(GX-GX)*(GZ-GZ))+((GO-GZ)*(GU- GX))+GY)-GY)!2+(((-(GM-GY)*(GU- GX))+((GX-GX)*(GV-GY))+GZ)- GZ)!2))+GY) G79 R(GR-(1/GW)) M(GE(GR)) H500 ; GR >= 0 G79 R(1) G01 X(GX) Y(GZ) Z(GY) G01 X(GX+(2*1/GS)) Y(GZ) Z(GY) G79 Y(((1-GR)!2*(LX*LB + LY*LF + LZ*LK) + 2*(1-GR)*GR*(LX*LO + LY*LS + LZ*LV) + GR!2*(LX*GB + LY*GF + LZ*GK)) G79 Z(((1-GR)!2*(LX*LC + LY*LI + LZ*LL) + 2*(1-GR)*GR*(LX*LQ + LY*LT + LZ*LW) + GR!2*(LX*GC + LY*GI + LZ*GL) - GT) M(EQ(0)) H900 </pre>	
N1000 M2	Termina o programa com a função M2

Fonte: Elaborado pelo Autor

5 CONCLUSÃO

Considerando a metodologia de implementação do código na máquina ROMI de Comando numérico computadorizado, pôde-se observar que a ideia de programar ciclos de usinagem parametrizados baseados em Superfícies de Bezier satisfariam a condição de usinar corretamente formas geométricas de superfícies complexas e em plano tridimensional de forma eficiente e precisa, se fosse tido a oportunidade de realiza-las de forma prática nas máquinas CNC da Universidade Federal de Santa Maria.

Ciclos de usinagem em planos bidimensionais e em 2 e $\frac{1}{2}$ D são regularmente praticados em máquinas de comando numérico computadorizado, fazendo com que usinagem de materiais brutos seguindo trajetórias em planos tridimensionais se tornem um meio desafiador e determinante para se conseguir formas em que curvas suaves e definidas fiquem na forma do material usinado final.

Pôde-se observar que as implicações de seu uso tendem a ser mais positivas de que inconvenientes e impraticáveis, demonstrando, que, se, com recursos humanos e tecnológicos de qualidade, se consegue realizar a programação paramétrica de ciclos de usinagem específicos nas máquinas de comando numérico para, conseqüentemente, usinar de forma eficiente as formas dos materiais desejados.

REFERÊNCIAS BIBLIOGRÁFICAS

AZEVEDO, A.L de, **Métodos de programação CNC**. Disponível em: <<https://www.linkedin.com/pulse/métodos-de-programação-cnc-americo-luiz-de-azevedo/>>. Acesso em: 9 jun. 2020.

AZEVEDO, F.O DOMINGUES. **Linguagem de Programação de CNC Torno e Centro de Usinagem** – 1ªed. 30 abr. 2019.

BEUDAERT, X. LAVERNHE, S. TOURNIER, C. **Direct trajectory interpolation on the surface using an open CNC**. The International Journal of Advanced Manufacturing Technology, v. 75, n. 1-4, p. 535-546, 2014.

BOBCAD-CAM. **Cam Programming Technology Guaranteed To Increase Cnc Program Efficiency**, 2015. Disponível em: <<https://bobcad.com/cam-programming-technology-guaranteed-to-increase-cnc-program-efficiency/>>. Acesso em: 15 jun. 2020.

CADCAMENGINEERING: Disponível em: <<https://cadcamengineering.net/cnc-milling-machine-configuration/>>. Acesso em: 9 ago. 2020

CASSANIGA, F. A. **Fácil Programação do Controle Numérico**. Sorocaba: F. a C. Produções Editoriais, 2000.

COMAK, A. **Mechanics, Dynamics and Stability of Turn-Milling Operations**, DOI: 10.14288/1.0368954, 2018.

CUSTOMPARTNET **Scallop Heights**. Disponível em: <<https://dcain.etsin>>. Acesso em: 16 jun. 2020.

GONÇALVES, M. A. F. **Um estudo sobre implementação de ciclos de usinagem através de programação parametrizada em máquinas de comando numérico computadorizado**. Dissertação (Mestrado Engenharia de Produção) – Universidade Federal de Santa Maria, Santa Maria, RS, 2007.

GONÇALVES, M. A. F. **Geração de programas CNC através da implementação de funções direcionadas às características do processo produtivo**. Porto Alegre, 2013.

GROOVER, M. **Automação Industrial e Sistemas de Manufatura**. 3ed, São Paulo: Pearson Brasil, 2011.

GROOVER, M.P. **Fundamentals of Modern Manufacturing: materials, processes, and systems** – 4rd ed. John Wiley & Sons Inc., EUA, 2010.

JAMBRINA, F.L. **Superfícies de Bézier**. Disponível em: <<https://dcain.etsin.upm.es/~leonardo/tema5.htm>>. Acesso em: 25 jul. 2020.

KHAN, H. REHMAN. **Tool path generation, for complex surface machining, using point cloud data**, Doi.org/10.1016/j.procir.2014.07.076, 2015.

LAYEGH, E. **Effects of Lead and Tilt Angles on Cutting Forces, Torque and Cutting Power in Five-axis Sculptured Surface Milling**. 2012.

LI, D. ZHANG, W. SHANG, T. FLEISCHER, J. **A solid-analytical-based model for extracting cutter-workpiece engagement in 5-axis flank machining**, DOI: 10.1051/mateconf/201823701009, 2018.

LIGACONSULTORIAJR: Disponível em
<<https://www.ligaconsultoriajr.com.br/post/usinagem-cnc>>. Acesso em: 10 ago. 2020.

LYNCH, M. **The key concepts of CNC. Modern Machine Shop**, Cincinnati, vol 69, n11A, pp81-144, April 1997.

MALDANER, F.H. **Um sistema CAD/CAM dedicado às máquinas CNC Router**. Universidade Federal de Santa Maria – UFSM, Santa Maria, 2018.

MARCICANO, J.P.P. **Introdução ao Controle Numérico**. São Paulo. World Wide Web, 2017. Disponível em: <<http://www.poli.usp.br>>. Acesso em: 5 ago. 2020.

MIRANDA, R. J. C. **Desenvolvimento de um programa didático computacional destinado à geração de códigos de Comando Numérico a partir de modelos 3D obtidos em plataforma CAD considerando a técnica prototipagem rápida**. Dissertação Universidade Federal de Minas Gerais, MG, 2009.

MONARO, R.L.G; SCHUTZER, K. **Capacitação em Modelamento e Manufatura de Sólidos Utilizando Sistema CAD/CAM**, 2008. Disponível em: <http://www.unimep.br/phpg/mostraacademica/anais/4mostra/pdfs/337.pdf>. Acessado em: 15 jun. 2020

MUNDO CNC: Disponível em: <<http://mundocnc.com.br>>. Acesso em: 11 ago. 2020.

OLIVEIRA, C. A. L de, **Uma Metodologia de Integração CAD/CAM através da aplicação de Ciclos de Usinagem na programação CNC** Universidade Federal de Santa Maria – UFSM, Santa Maria, 2011.

OLIVEIRA, L.C. de, **Um sistema de geração de trajetórias de ferramentas em três eixos**, Universidade Federal do Rio Grande do Sul - UFRGS, Porto Alegre, 1997.

OLIVEIRA, L.C. de, **Otimização do caminho da ferramenta em processo de usinagem por revolução de peça com formato complexo**, 2013.

ROMI; **Manual de Programação Parametrizada**. São Paulo, 2015.

SCHNEIDER, E.L.K. **Adequação e controle de um pós-processador para centro de usinagem vertical**. Panambi, 2015.

SILVA, S. D. **CNC Programação de Comando Numérico Computadorizado**

Torneamento. 8ª ed. São Paulo: Érica, 2008, 312p.

SOUZA, A. F. **Conceitos Avançados Mundo CNC.** Disponível em: <<http://www.mundoconc.com.br>>. Acesso em: 20 set. 2020.

SOUZA, D.J.A. Usinagem de Superfícies definidas em CAD. **Estudos Tecnológicos Unisinos.** vol. 3, V. 1, 2007.

SOUZA, A. F. **Contribuições ao fresamento de geometrias complexas aplicando a tecnologia de usinagem com altas velocidades,** Tese (Doutorado em Engenharia Mecânica), Escola de Engenharia de São Carlos, São Paulo, SP, 2004.

SOUZA, A.F.; Coelho, R. T. Experimental investigation of feed rate limitation on high speed milling aimed at industrial applications. **The International Journal of Advanced Manufacturing Technology**, DOI 10.1007/s00170-006-0445-2, 2007.

SOUZA, A, F; Ulbrich, C, B, L, **Engenharia Integrada por computador e sistemas CAD/CAM/CAE.** São Paulo: Artlibre, 2009.

TEIXEIRA, F.G. **Modelamento paramétrico e geração de malha em superfícies para aplicações em engenharia.** Porto Alegre, 2003.