

O problema que enfrentaremos hoje mostra ainda o uso geral das variáveis.
Eu vou mostrar como o CNC pode nos ajudar no cálculo das quotas.
Em particular, vamos simular a "programação direta"!

Este é o problema que eu tenho sido solicitado por um usuário (que chamarei de Fausto) há algum tempo!

-Fausto-

"Muitas vezes você tem que fazer um trabalho e no desenho você pode encontrar apenas uma parte X ou Y e os graus (por exemplo 30).

Pessoalmente, se eu tiver perfis complexos uso o ManualGuide, pelo contrario eu programo normalmente com a ajuda de uma calculadora científica.

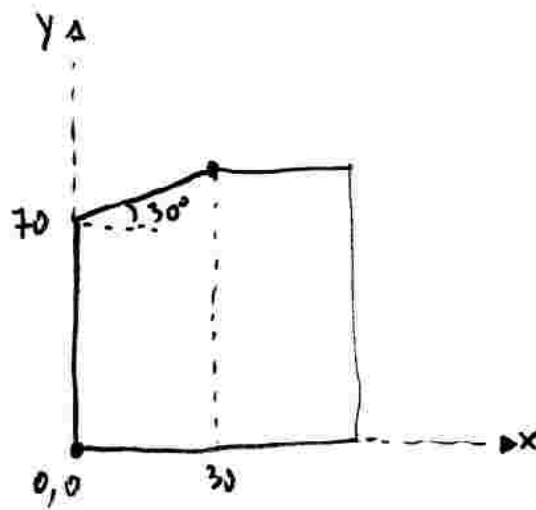


FIGURA 1

Mas eu me pergunto (figura 1): tenho que chegar até X30, mas não tenho a dimensão Y, mas sei que os Graus são 30.
Não querendo usar a calculadora ou um programa CAD como faço para calcular?

```
D1 G1 G41 X0 F500;  
G1 Y70;
```

Eu tentei com:

```
G1 Y70, A30;  
G1 X50;
```

mas não funciona, por que para minha CNC 'A' é o quarto eixo. Não ha algo simples e rápido para escrever? "

Este é o problema que enfrentaremos! (Vamos também continuar a reflexão sobre o contator).

Vários comandos CNC permitem a programação direta de quotas; tentar simular este conceito é, certamente, um bom trabalho, mas não deve ser um estímulo à preguiça. Eu, pessoalmente, acho que não é um problema calcular pontos. Experimentar novos métodos, alternativas diferentes, é contudo um bom exercício para aprender o paramétrico!

Neste contexto, vamos usar as variáveis comuns (#100 - #999), e algumas variáveis de sistema para leitura de posição #5001/5003 (posição atual, coordenadas da peça, no final do bloco).

E' possível resolver o problema com a programação clássica?

Temos que usar o paramétrico?

Esta é a primeira pergunta que deve ser feita no momento de escrever um programa paramétrico. Muitas vezes a solução está mais perto do que parece e nem sempre o paramétrico é a melhor solução. No problema proposto não sabemos um valor específico (X no exemplo), então temos que usar as variáveis.

As variáveis permitem que você escreva um programa com um sentido, sem conhecer os valores de certas entidades.

O problema é puramente matemático; assim precisa ser abordado!

Estudamos o problema e vamos encontrar uma solução (se tiver, naturalmente).

Em seguida iremos aplicá-la sob forma de programa CNC.

As relações entre os lados de um triângulo são numerosas, mas as mais conhecidas e usadas são baseadas na trigonometria. Aqui estão as principais fórmulas (fig. 2)

$$\text{SIN}(a)=B/C \quad \text{COS}(a)=A/C \quad \text{TAN}(a)=B/A$$

com (a)=ângulo

(A)=cateto adjacente o ângulo

(B) cateto oposto o ângulo

(C) hipotenusa

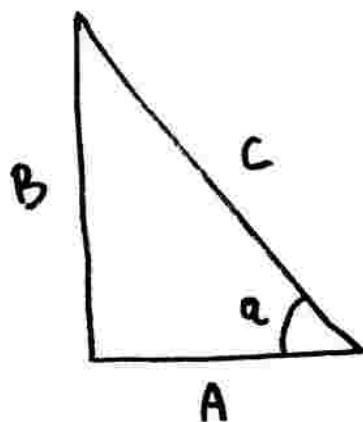


FIGURA 2

Ao escrever G1 Y70, A30 se pretende atingir o valor de X=30 através de uma linha inclinada de 30 graus em relação ao eixo X.

As relações entre os lados de um triângulo retângulo permitem que você escreva X em função de Y e o ângulo (a).

$$\text{TAN}(a)=Y/X, \quad X=Y/\text{TAN}(a), \quad \text{em seguida,} \quad \text{G1 X}[70/\text{TAN}(30)] \text{ Y70}$$

O valor de X se encontra dividindo o valor de Y pela tangente do ângulo (figura 3).

Aplicamos o princípio a figura (4).

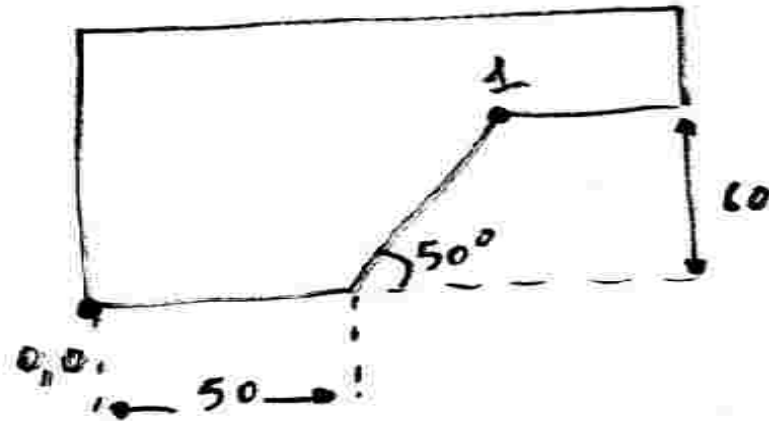


FIGURA 4

Calcula-se o valor do ponto 1. $x = 60 / \tan[50]$, se estamos no ponto $x=50$ $Y=0$ podemos escrever

```
G01 X[60/TAN[50]] Y60
```

Examinamos a figura (5).

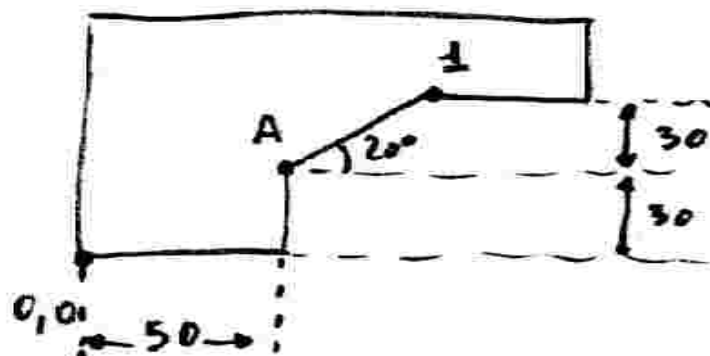


FIGURA 5

O procedimento de cálculo do ponto 1 não muda!

Devemos salientar, contudo, que os valores são referidos ao ponto A e não mais a origem (O): os valores de X e Y previamente encontrados são de tipo incrementais (G91).

```
G91 G01 X[30/TAN[20]] Y30
```

Os valores absolutos de X (p1) e Y (p1) são

```
X=[50 + [30/TAN[20]]] Y=30+30
```

As variáveis de sistema do CNC permitem estabelecer a localização exata, atual, da ferramenta, e por isso não é difícil implementar um programa paramétrico que permite a escrita incremental e a solução no estado absoluto (G90), o operador não deve mais lembrar a posição atual!

```
G90 G01 X[#5001+[30/TAN[20]]] Y[#5022+30]
```

Este exemplo mostra como nos programas ISO você pode usar as variáveis para realizar cálculos e determinar os pontos.

Bom gente, programar em G90 ou G91 é escolha de vocês!

O conceito também pode ser expandido para outras formas, permitindo, realizar perfis complexos.

Vamos ver uma aplicação, vejam a figura (6)

Bom, não vou dizer como resolver o exemplo, fica para você aplicar os conceitos aprendidos!

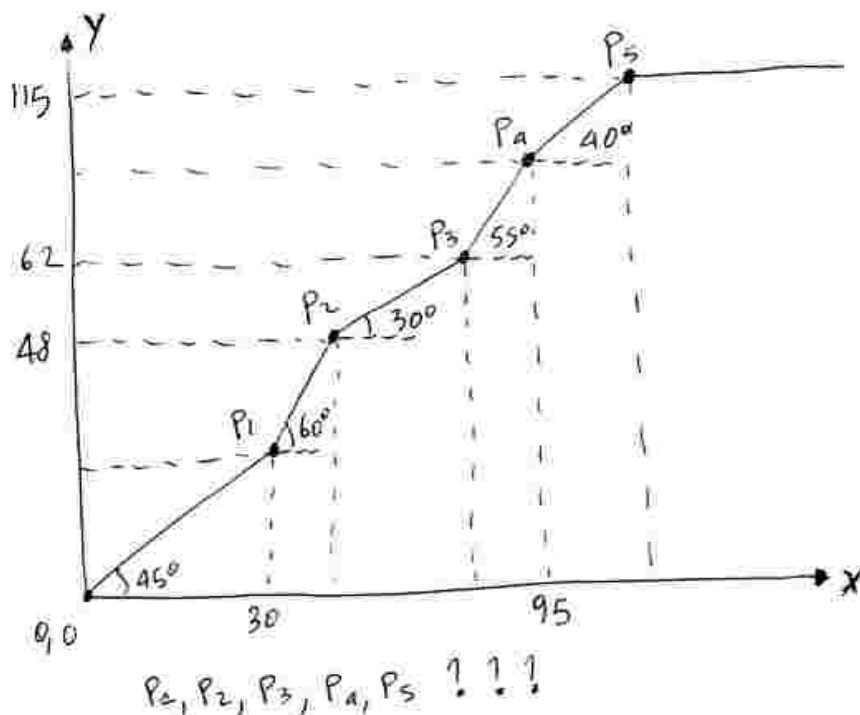


FIGURA 6

Nos programas de usinagem que escrevemos você também pode usar as frases macro, ou seja, funções matemáticas, lógicas e de comparação, ciclos ... Não devemos necessariamente escrever um programa macro para utilizar estas funções, assim como não estamos obrigados a utilizar apenas as variáveis comuns, mas cada conhecida variável.

Em "programação paramétrica: Exemplos (1)", falamos sobre contadores.

Vimos como o uso de uma variável comum permite que você execute certas ações.

Mas a utilização do contador pode ser aplicado a muitos aspectos da programação.

Aplicação: a repetição!

Problema: execução de um perfil repetido 100 vezes.

As soluções para este problema são diferentes! Com a programação clássica se pode escrever uma sub-rotina, por exemplo. Isso permite que você escreva o perfil uma só vez, que irá ser repetido quantas vezes foram necessárias.

Usando um contador aplica-se o mesmo princípio.

```
#100=100 (contador)
N10 "perfil programado" (perfil a ser usinado)
#100=#100-1 (diminuir o contador)
IF[#100NE0]GOTO10 (condição de repetição ou paragem)
M30
```

O exemplo apresenta um contador impostado para executar um perfil repetido.

O perfil é executado, em seguida, se reduz o contador.

A instrução IF verifica se chegamos ao valor 0, isto é, se nós fizemos as repetições necessárias. Bem, se for assim o programa termina (M30), caso contrário vai ser repetido o perfil (GOTO10).

A decisão de utilizar uma sub-rotina ou um contador depende de suas necessidades pessoais. A utilização de uma variável apresenta alguns aspectos a considerar. Este jeito permite que você tenha um único programa, se você precisa mudar algo, você o pode fazer sem chamar a sub-rotina. É uma escrita flexível, precisa realmente pouco para mudá-la.

Dissemos que em nossos programas podemos usar qualquer frase macro, o que acreditamos. O exemplo feito com o uso de um GOTO pode ser realizado de uma forma mais lucrativa usando a frase WHILE. Lembre-se que usando o GOTO, e este for direcionado para trás, o CNC deve ler todo o programa o que pode significar uma parada na usinagem. Usando o WHILE isso não acontece porque o programa para no ciclo.

```
#100=100
WHILE[#100NE0] DO1
"Perfil"
#100=#100-1
END1
M30
```

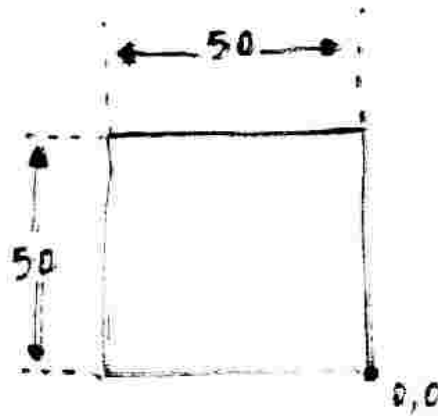
O funcionamento é idêntico ao anterior, o loop WHILE (isto é, a repetição) será executado até que a condição seja satisfeita (#100 for diferente de 0).

FIZEMOS VARIAS SUPOSIÇÕES; está na hora de experimentá-las!! Como?

Considere uma simples figura e tente aplicar os conceitos expostos.

Veja a figura abaixo, e tentamos aplicar as noções aprendidas!

Quando você experimenta seus programas sempre use figuras simples. Se você usar formas complexas podem ser gerados erros inesperados que escondem o verdadeiro problema.



Aqui é o programa que executa o processamento na forma clássica.

```
%
<K0100>
G90G0
T26M6 (FRESA D=10)
G90G0G54X10Y-10A0C0S1000M3
G43H26Z50M8
G10L12P2R5
D2
Z2
G01F1000Z0
#100=5
WHILE[#100NE0] DO1
G91G01Z-5
G90G01G41X0Y0
G01X-50
G01Y50
G01X0
G01Y0
G90G01G40X10Y-10
#100=#100-1
END1
G90G0Z200M9
M5
G08P0
M1
G90G0G53Z0
M30
%
```

O programa realiza 5 repetições delineando o perfil descrito.

(S e F são valores puramente abstratos)

Agora vamos escrever o programa usando a notação direta com o uso de variáveis. Então, em vez de escrever os valores numéricos de X e Y vamos por as relações trigonométricas apropriadas.

```

%
<K0100>
G90G0
T26M6 (FRESA D=10)
G90G0G54X10Y-10A0C0S1000M3
G43H26Z100M8
G10L12P2R5
D2
G01F1000Z90
#100=5
WHILE[#100NE0] DO1
G91G01Z-1
G90G01G41X0Y0
G91G01X[50*COS[180]]
G01Y [50*SIN[90]]
G01X [50*COS[0]]
G01Y [50*SIN[-90]]
G90G01G40X10Y-10
#100=#100-1
END1
G90G0Z200M9
M5
G08P0
M1
G90G0G53Z0
M30
%
```

O programa é o mesmo que acima. Lembrando que $\cos(180)=-1$, $\sin(90)=1$, $\cos(0)=1$, $\sin(-90)=-1$, a escrita fica a seguinte

```

G91G01X[50*COS[180]]  -->>  G91G01X50
G01Y [50*SIN[90]]    -->>  G01Y50
G01X [50*COS[0]]     -->>  G01X50
G01Y [50*SIN[-90]]   -->>  G01Y-50
```

Já mencionamos que o cálculo trigonométrico é incremental (G91), ou seja, a mudança é sempre em referência ao ponto atual.

Mas se queremos expressar o programa de uma forma absoluta? Como implementar esse conceito?

Se queremos escrever um programa em dimensões absolutas, precisamos saber a sua localização atual e adicionar o valor encontrado pelas fórmulas.

As variáveis de sistema 5001-5002-5003 fornecem a posição em coordenadas da peça (origem usinagem), sem corretores, no final da linha executada.

Usando estas variáveis junto à trigonometria obtemos o resultado desejado

```

%
<K0100>
G90G0
T26M6 (FRESA D=10)
G90G0G54X10Y-10A0C0S1000M3
G43H26Z100M8
G10L12P2R5
D2
G01F1000Z90
#100=5
WHILE[#100NE0] DO1
G91G01Z-1
G90G01G41X0Y0
G01X [#5001 + [50*COS[180]]] >> ( #5001=0, 0+(50*(-1))=-50, G01X-50 )
G01Y [#5002 + [50*SIN[90]]] >> ( #5002=0, 0+(50*(1))=50, G01Y50 )
G01X [#5001 + [50*COS[0]]] >> ( #5001=-50, -50+(50*(1))=0, G01X0 )
G01Y [#5002 + [50*SIN[-90]]] >> ( #5002=50, 50+(50*(-1))=0, G01Y0 )
G90G01G40X10Y-10
```

```
#100=#100-1
END1
G90G0Z200M9
M5
G08P0
M1
G90G0G53Z0
M30
%
```

Se você executa o programa pode ter a certeza que realmente funciona (além de erros de escrita). Tendo verificado o funcionamento correto, você pode usá-lo sempre que houver necessidade.

Este último exemplo aparentemente trivial, destaca um aspecto muito importante: o operador não precisa lembrar da posição atual é suficiente digitar a posição atual (X#5001, Y#5002) através das variáveis e adicionar a fórmula trigonométrica (o conceito é o mesmo que escrever em G91).

Pode parecer um pouco complicado no início, mas eu lhes asseguro que, depois de um pouco de prática você vai reconhecer a beleza e a força do presente escrito.

Fausto tem algo a expressar!

"Eu queria fazer alguns comentários!

Sempre há uma relação direta entre #100 e a remoção a ser feita, eu vou explicar. Se eu tiver que remover 100mm e eu quero fazer 50 repetições, eu preciso remover 2 milímetros a cada passagem. Se eu quiser executar 20 passagens tirarei 5mm. Mas se eu quisesse realizar uma remoção arbitrária, ou uma repetição arbitrária, encontrarei o problema que nem sempre há uma relação entre inteiros capazes de relacionar as variáveis em jogo. Por exemplo, eu gostaria de remover 2,347 milímetros, quantas repetições eu tenho que fazer? 42 ou 43? Como posso resolver isso? "

Fausto apresenta uma questão que não é estúpida: dar ao operador a possibilidade de dissociar o valor para ser tirado, do número de repetições.

Como posso verificar se o valor final de remoção não é excessivo ou menor do que o necessário?

Trata-se de um pequeno problema que pode ser resolvido de várias maneiras.

- * Pode-se calcular um valor médio perto do valor desejado
- * Usino a partir de um valor superior ou inferior para compensar a remoção de excesso ou defeito.

São todas maneiras corretas de trabalhar, mas aqui eu vou mostrar uma solução alternativa que faz uso de variáveis de sistema.

Não é melhor ou pior do que as outras formas de programa, mas mostra o uso de variáveis de sistema, que você precisa saber para explorar totalmente o poder da programação paramétrica.

(Como um exercício você pode implementar uma solução para este problema, como você acha que seja melhor: deixe-me saber suas ideias, nós desenvolveremos!)

Eu tenho que fazer um loop que verifica automaticamente o valor de Z removido! (Considere sempre o nosso perfil de 100mm)

Como sempre, pensar com cuidado antes de escrever seu programa, achar uma solução que é gravável com facilidade.

O conceito é a seguinte:

-Posicionamento no ciclo de início

-Cálculo (sem movimento) do valor do afastamento próxima Z

-Análise do cálculo:

Z> -100? podemos proceder pois não chegamos ao fim

Z = -100? podemos proceder pois é o valor final

Z <-100? PARE, você precisa fazer ajustes!

(Definir o valor absoluto para -100, para não exceder o valor final do Z)
-Execução do movimento
-Repetição do ciclo, novamente a análise, ou saída do ciclo

Como você pode ver eu tentei criar uma estrutura do comportamento de nosso programa.

Ter uma ideia clara do que escrever podemos escrever o nosso programa.

Para executar o ciclo utilizaremos a frase WHILE, para analisar a posição atual em Z, usamos a #5003 (posição no fim do bloco que no final do deslocamento).

```
#120=-100
#121=2.347
WHILE[#5003GE#120]DO1
#122=#5003-#121
#123=ABS[#120-#5003]
IF[#122LT#120]THEN#121=#123
G91G01Z-#121
-----
perfil
-----
END1
G90G0Z200M9
M30
```

As seguintes linhas definem o valor de Z e a remoção final para a única passagem
120 = -100 (valor final de Z)
121 = 2347 (remoção)

Vai o loop WHILE que permite a repetição do ciclo: a condição de repetição é a posição atual (#5003) maior ou igual ao valor de profundidade escrito em #120.

WHILE[#5003GE#120]DO1 (ciclo repetitivo e condições)

Antes de realizar o movimento é preciso avaliar o valor de Z para o ciclo seguinte (#122). Nós fazemos a diferença com o valor final para conhecer a porção faltando. Se a porção que falta é maior do que o valor de remoção, isto significa que tem de executar, pelo menos, mais um ciclo.

Se a porção que falta é menor do que o valor definido para a remoção significa que um novo ciclo excede o valor de Z para a profundidade máxima. Temos então que remover uma quantidade de material diferente da #121 mas igual ao valor da parte que falta. Em outras palavras, a remoção será representada por #121 ou #123 de acordo com a condição em que nos encontramos.

```
#122=#5003-#121 (#122 = valor atual + remoção, #122 = novo valor de Z)
#123=ABS[#120-#5003] (#23 = valor absoluto da diferença entre #120 e a a
posição atual)
IF[#122LT#120]THEN#121=#123 (COMPARAÇÃO)
```

Agora, podemos realizar o perfil e repetir n vezes a operação.

A condição #5003EQ#120 (que é sempre verificada) termina o ciclo e o final do programa.

Bem, por hoje chega!

O que podemos dizer sobre esse discurso?

As variáveis são usadas quando não sabemos a priori um valor numérico, ou para executar cálculos. Nos nossos programas, podemos usar qualquer tipo de variável ou frase macro. As variáveis permitem de escrever programas flexíveis, adaptáveis a diferentes situações!

Os exemplos são deliberadamente educacionais. O objetivo é descobrir como fazer uso pleno de nosso CNC através do uso de variáveis.

Hoje todo mundo tem um CAM que facilita muito o cálculo de determinados pontos.

Nossos comandos possuem uma miríade de funções que facilitam o nosso trabalho, permitindo que até mesmo quem não possui certos conhecimentos pode ser capaz de executar tarefas complexas. Isto não deve impedir-nos a compreender e testar os conceitos escondidos por trás desse poder, não deve ser uma desculpa para dizer, "já foi feito ... porque aplicar o paramétrico?"

Um dia você poderá escrever sua função ou alterar uma existente: isso requer conhecimento, visão!

Não importa quantos recursos temos, aprenda o paramétrico e você vai ver que há ainda muito espaço para a criatividade.

Como qualquer novo material, eu comecei com exemplos simples.

Tente compreender o conceito básico e não pense "mas que idiota, quem usaria ..." você vai ver que a realização de programas complexos encontram sua base em alguns conceitos simples! São as ideias que nos temos que nos tornam bons programadores: toda ideia, mais que parece louca, doida, é "o ponto de partida" para uma reflexão e nos dá a oportunidade de aprender mais e mais. Assim, mesmo quando já existem soluções para nossos problemas é sempre útil abordá-los de jeito diferente!