



COMPROMETIDA COM A PROMOÇÃO DO DESENVOLVIMENTO DA ENGENHARIA E DAS CIÊNCIAS MECÂNICAS

7º CONGRESSO BRASILEIRO DE ENGENHARIA DE FABRICAÇÃO
7th BRAZILIAN CONGRESS ON MANUFACTURING ENGINEERING
20 a 24 de maio de 2013 – Penedo, Itatiaia – RJ -Brasil
May 20th to 24th, 2013–Penedo, Itatiaia – RJ – Brazil

PROGRAMAÇÃO DO CORTE DE CHAPAS MODELADAS EM B-REP

Dalberto Dias da Costa, dalberto@ufpr.br¹
Gilberto Passos Lima, gibapassos@gmail.com²

¹Universidade Federal do Paraná, Departamento de Engenharia Mecânica, Curitiba - PR

² Universidade Federal do Paraná, Pós-graduação em Engenharia de Produção, Curitiba – PR

Resumo: A aplicação do Comando Numérico Computadorizado (CNC) na programação do corte de chapas é, atualmente, uma prática corriqueira na indústria metal-mecânica. As máquinas CNC são utilizadas como um recurso complementar para vários processos de corte, tais como: laser, jato de água e plasma. O processo se inicia com um modelo CAD (Computer-Aided Design), o qual é exportado para um sistema CAM (Computer-Aided Manufacturing). O sistema CAM é justificado como essencial, principalmente, para a geração da trajetória de corte em 3D. Geralmente, o formato DXF (Drawing eXchange Format) é selecionado como um meio de transferência de dados entre os sistemas CAD-CAM para o corte de peças em 2D. Apesar de sua simplicidade, o formato DXF não contém informações topológicas sobre a peça modelada e sua estrutura de dados não foi concebida para cobrir todo o ciclo de vida do produto. Por outro lado, a contínua aceitação da modelagem 3D, em especial a modelagem sólida em B-Rep (Boundary Representation) vem demandando novos meios para o interfaceamento dos sistemas CAD-CAM. Nesse novo cenário, o uso de um formato mais completo e durável, tal como o STEP (Standard for the Exchange of Product Model Data) definido pela norma ISO 10303, torna-se mandatário. Entretanto, isto levanta duas questões: como programar máquinas e sistemas que ainda fazem uso da modelagem em 2D? E também, como a superfície de corte pode ser reconhecida e extraída automaticamente? O objetivo deste trabalho é responder a essas questões delineando uma metodologia para separar as faces planas de um modelo B-Rep formatado em STEP e extrair seus contornos internos e externos. Após esse procedimento, a trajetória de corte é calculada, levando-se em consideração o ajuste necessário para compensar a largura da sangria do corte. Um protótipo computacional foi implementado em MS Visual Basic e avaliado com peças modeladas em dois softwares CAD comerciais.

Palavras-chave: Corte de chapas, B-rep, STEP, Programação CNC

1. INTRODUÇÃO

No setor metal-mecânico o corte de chapas com auxílio da tecnologia de comando numérico computadorizado (CNC) se tornou um meio atrativo para redução dos tempos de preparação e, por conseguinte, da redução do custo de fabricação de peças em pequenos lotes. A razão disto está na flexibilidade propiciada pelas máquinas CNC e no emprego de basicamente uma única ferramenta, neste caso denominada bocal de corte, para o processamento de peças com as mais variadas geometrias (DeGarmo et al., 1997).

Atualmente, o corte pelos processos a laser e jato de água, por exemplo, pode ser feito sobre superfícies planas e curvas. No caso de superfícies planas, a trajetória de corte pode ser programada manualmente. Todavia, para as superfícies curvas existem softwares dedicados que possibilitam a geração automática do caminho de corte em três dimensões (3D). Diversos fabricantes, como por exemplo a Triumph GmbH (Triumph, 2012) e a Flow Coporation (Flow, 2012) possuem softwares dedicados à programação automática.

Aplicações do corte em 3D podem ser vistas na finalização de peças estampadas. Esse tipo de situação surge como necessidade de reduzir o tempo de construção e o custo das matrizes para estampagem. Desse modo, a rebarbação, ou pequenos detalhes, podem ser executados com laser sobre as superfícies curvas após a etapa de conformação. Não obstante a esse estágio tecnológico, grande parte da aplicação dos processos de corte a laser, jato de água e plasma ainda se dá na produção de peças planas.

No corte de chapas planas, a utilização de modelos em duas dimensões ainda é uma prática predominante. No caso de modelos digitais, o formato DXF (Drawing eXchange Format) é considerado o formato *de facto* para esse tipo de aplicação. Jackson e Mittal (1993) e Marques et al (2011) são exemplos de trabalhos desenvolvidos para a geração automática de corte a partir de arquivos DXF. Por se tratar de uma modelagem bidimensional, o formato DXF elimina a necessidade de se identificar previamente a face a ser cortada. Isto, obviamente, facilita a programação automática.

Entretanto, a modelagem de peças em 3D vem, rapidamente, substituindo aquela em duas dimensões. Segundo Foggiatto et al. (2007), uma das vantagens da modelagem em 3D está na interatividade com outras áreas. Ainda segundo eles: “O modelo pode ser utilizado para calcular o seu volume, propriedades de massa e momentos de inércia,

realizar montagens, verificar interferências, gerar os desenhos de fabricação, simular o seu comportamento mecânico, prototipagem rápida, usinagem por CNC, entre outros”.

Além disso, a técnica de modelagem sólida B-Rep (*Boundary Representation*) formatada em conformidade com a família de normas ISO 10303, também conhecida como STEP (Standard for the Exchange of Product Model Data), tem sido proposta como uma condição necessária ao gerenciamento do ciclo de vida de um produto (Gielingh, 2008).

A partir desse cenário, podem-se formular as seguintes questões: como programar automaticamente o corte de chapas, em máquinas e sistemas ainda baseados em duas dimensões (2D), a partir de modelos em 3D? E mais, como identificar automaticamente a face a ser cortada? O objetivo neste trabalho é responder a essas duas questões.

Este trabalho deve ser entendido como uma continuidade de um projeto iniciado por um dos autores e reportado em Marques et al (2011). No artigo citado, os autores propuseram uma metodologia para a programação automática do corte de chapas a partir de modelos armazenados em DXF. Para auxiliar na resposta à primeira questão formulada anteriormente, propõe-se que os métodos para o cálculo da correção do raio de corte (CRC), definição do ponto de início do corte e geração do código CN — já desenvolvidos no referido trabalho — sejam integrados à metodologia aqui proposta, a qual será descrita a seguir.

2. METODOLOGIA

Os métodos descritos a seguir têm como propósito a elaboração de um programa computacional para a extração de informações geométricas e topológicas de um componente definido em três dimensões (3D) em conformidade com a técnica de modelagem sólida B-Rep.

Ao seu escopo impõem-se as seguintes restrições: o sólido em questão deve ser uma chapa plana; as fronteiras que limitam as superfícies da chapa podem ser compostas por segmentos de reta, arcos de círculos, círculos ou pela combinação dos mesmos e, por último, a estrutura de dados deve ser arquivada na forma de texto, tal como definido pelo formato STEP em conformidade com a norma ISO 10303 parte 21 e Protocolo de Aplicação (AP – *Application Protocol*) 203 ou 214 (ISO, 2002; ISO, 2011; ISO, 2010).

O fato de o componente modelado ser uma chapa implica na existência de, no mínimo, duas superfícies planas e paralelas. Além disso, a soma da área bruta de um par de superfícies paralelas, isto é sem a subtração de seus elementos internos, deve ser superior à soma das áreas das demais superfícies externas que compõem o sólido. Dessa forma, define-se a espessura da chapa como sendo a distância entre essas duas superfícies de maior área. Em outros termos, pode-se dizer que os componentes, aqui denominados como chapas, são sólidos com superfícies planas e de pequena espessura.

O par de superfícies de maior área poderá ter fronteiras internas, sendo que às mesmas se aplica a segunda restrição mencionada acima. Entretanto, para as demais superfícies não se admite a existência de fronteiras internas. A fronteira externa e as internas, caso existam, devem ser idênticas nas duas superfícies de maior área.

Para a segunda restrição mencionada acima, acrescenta-se ainda a questão de unicidade dos vértices. Isto é, um vértice só pode ser utilizado uma única vez na definição de uma fronteira, seja interna ou externa. Isto implica na impossibilidade de intersecção entre os contornos de fronteiras diferentes.

A geração automática de programas CNC para o corte de chapas, a partir de modelos digitais oriundos de sistema CAD, pode ser sintetizada nas seguintes etapas (Marques et al., 2011): Extração dos dados geométricos; conversão dos mesmos para uma estrutura proprietária; cálculo da correção do raio de corte pelo ajuste (*offset*) dos contornos; encadeamento do corte e geração do código CN.

A metodologia a ser detalhada adiante será direcionada ao problema da extração dos dados geométricos em arquivos do tipo STEP e conversão dos mesmos para uma estrutura proprietária, já estabelecida por Marques et al. (2011). As demais etapas já foram tratadas no referido artigo e não serão mais objetos de análise.

2.1. Arquivos B-Rep em formato STEP

Os componentes sólidos, quando modelados em B-Rep podem ser formatados em STEP em conformidade com um dos Protocolos de Aplicação: AP 203 “Configuration controlled 3D design of mechanical parts and assemblies” ou o AP214 “Core data for automotive mechanical design processes”. Esses protocolos estabelecem um modelo para a transferência de dados de acordo com um dado contexto ou segmento industrial. Dependendo do tipo de software CAD utilizado, o usuário pode escolher entre o AP 203 ou o AP 214. Esse modelo estabelecido vai além das informações geométricas e topológicas. Por conseguinte, sobre o componente modelado podem ser incluídas, por exemplo, informações sobre a empresa, o responsável pela criação, o material e tolerâncias (ISO, 2011; ISO, 2010).

Entretanto, como o presente trabalho se limita à extração dos dados geométricos e topológicos, o enfoque foi dado às normas ISO 10303 parte 21 e parte 42, as quais tratam, respectivamente, do formato de dados quando armazenados em arquivos físicos e da representação geométrica e topológica (ISO, 2002; ISO, 2003).

O arquivo em STEP, de acordo com a ISO 10303 parte 21, deve ser gerado como texto com caracteres do *American Standard Code for Information Interchange* (ASCII) e organizado em seções. Uma dessas seções é denominada “DATA” e contém, entre outros, os dados geométricos e topológicos. O início e fim de cada seção deve ser indicado no arquivo, como por exemplo: “DATA;” e “ENDSEC;”. O fim de uma linha, ou bloco de dados, é marcado pelo caractere “;”. O início do arquivo é estabelecido pela linha contendo o bloco de dados: “ISO-10303-21;” e seu final pelo bloco

“END-ISO-10303-21;”. Além da seção “DATA”, um arquivo STEP ainda contém outra denominada “HEADER”, mas que não será contemplada neste trabalho.

As informações geométricas e topológicas são hierarquicamente organizadas por entidades, tal como apresentado esquematicamente na Fig. (1A). As entidades “CLOSED_SHELL”, “ADVANCED_FACE”, “FACE_BOUND”, “FACE_OUTER_BOUND”, “EDGE_LOOP”, “ORIENTED_EDGE”, “EDGE_CURVE” e “VERTEX_POINT” definem a topologia do sólido e as demais a sua geometria.

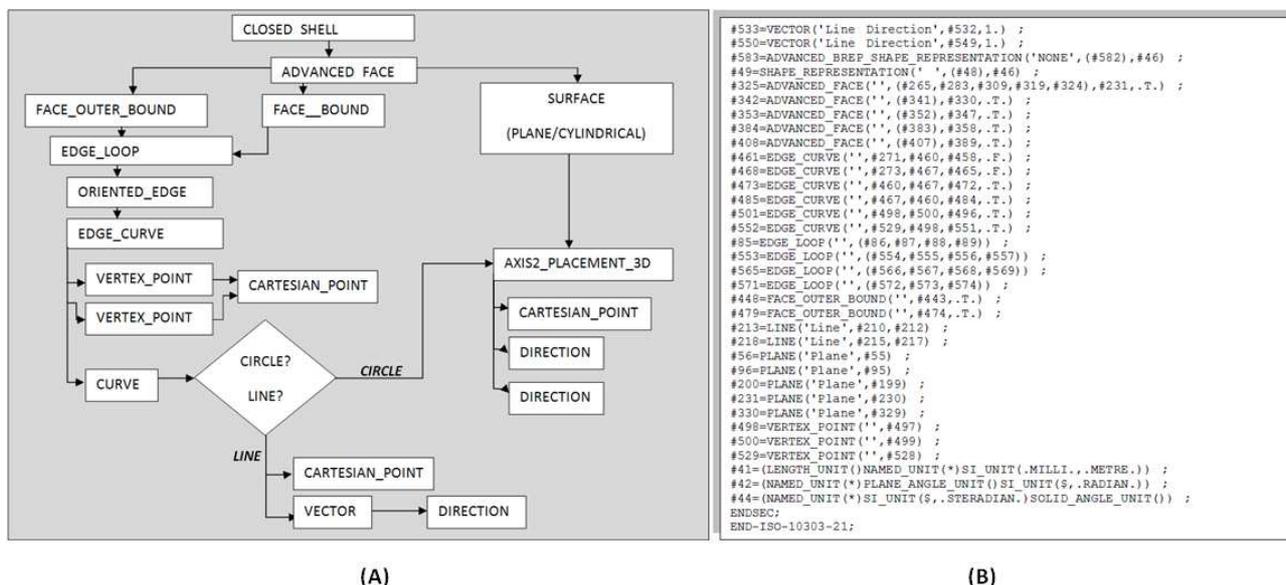


Figura 1. Entidades geométricas e topológicas empregadas na modelagem B-Rep. (A) esquema com relacionamentos. (B) Extrato modificado de um arquivo STEP.

Dentro do arquivo, as entidades são identificadas por um bloco de dados composto pelo caractere “#” seguido de um número inteiro que funciona como um contador de entidades. Entretanto, as entidades não são enumeradas pela sua hierarquia e, nem mesmo, na ordem em que aparecem no arquivo. Dessa forma, o identificador funciona como um apontador e, por conseguinte, não deve ser duplicado. A Figura 1B contém parte de um arquivo STEP gerado pelo software Catia® (Dassault Systèmes, França – Versão 5.13) referente a um componente modelado em B-Rep e formatado conforme o AP203 (ISO, 2011). Por meio desta figura, pode-se observar que as entidades são organizadas por classes. Não obstante, em outros software CAD, por exemplo o Solidworks® (Dassault Systèmes, França – Versão 2007) as entidades são enumeradas de acordo com sua ordem de gravação e, portanto, não são agrupadas.

Pode-se observar pela Fig. (1B) que cada linha do arquivo se inicia com o apontador, seguido pelo caractere “=”, pelo nome da entidade, o caractere “(”, depois pelos atributos da entidade, pelo parêntese de fechamento “)” e pelo “;” que indica o fim da linha. Em alguns casos, ver por exemplo os arquivos gerados pelo Solidworks®, esses blocos de dados são separados por espaços, mas que não exercem nenhuma influência na informação armazenada.

De um modo geral, uma linha de dados é separada em três partes: a primeira é composta pelo apontador, que está à esquerda do nome da entidade; a segunda contém o nome da entidade e a última, a sua direita, corresponde aos seus atributos. As linhas de dados organizadas desta maneira são empregadas para correspondência interna; o que é denominado na ISO 10303 parte 21 como “*internal mapping*”. Não obstante, existem linhas de dados que são empregadas para correspondência externa “*external mapping*” e, geralmente, para entidades complexas, ou seja, aquelas cujos atributos são outras entidades.

As entidades complexas não serão contempladas aqui, exceto aquela que estabelece a unidade dimensional para comprimentos, a qual deve aparecer no arquivo tal como mostrado pelo apontador “#41” da Fig. (1B).

2.2. Pré-processamento do arquivo STEP

Em um arquivo STEP (ISO, 2002) os dados geométricos e topológicos estão contidos em uma seção indicada por “DATA;” e finalizada por “ENDSEC;”. Cada linha, nessa seção, é iniciada com o caractere “#” e encerrada por “;”. Dessa forma, a extração dos dados de interesse pode ser feita linha a linha. Para cada linha lida, uma, ou mais entidades são separadas e o processamento de seus atributos é feito em conformidade com o seu contexto. A Figura (2A) contém, na forma de um fluxograma, o pseudocódigo referente ao processamento das linhas do arquivo. Cada linha recuperada deve ser posteriormente processada para a separação das entidades e de seus atributos, tal como esquematizado por meio da Fig. (2B).

A separação das entidades, e de seus atributos, se baseia na identificação de caracteres especiais que atuam como marcadores dos limites de cada bloco de dados. Esses caracteres são os seguintes: o “#”, referente ao identificador da entidade; o “=” que divide a linha lida nos blocos B1 e B2; o “(” que separa a entidade (B3) do restante da linha (B4) e

o “;” que indica o fim da linha. O procedimento esquematizado por meio da Fig. (2B) se aplica tanto na separação de entidades com correspondência interna (*internal mapping*), como para aquelas com correspondência externa (*external mapping*).

Do ponto de vista da Programação Orientada por Objetos (POO), as entidades, sejam de natureza topológica-geométrica ou não, podem ser entendidas como classes. Isto possibilita o processamento dos atributos, tendo como base a correspondência entre o nome da entidade encontrada no arquivo com um objeto existente. O estabelecimento dessa correspondência pode ser feito em tempo de desenvolvimento (*early binding*) ou em tempo de execução (*late binding*).

A principal vantagem em se estabelecer uma equivalência entre as entidades que serão separadas e os objetos com propriedades e métodos preestabelecidos está na facilidade de se implementar essa correspondência. Em contrapartida, qualquer modificação futura na definição das entidades implicará na necessidade de revisão de todo o código elaborado. Este inconveniente pode ser minorado se as correspondências forem estabelecidas em tempo de execução. Para isso, uma base contendo as classes de interesse deve ser mantida de forma independente do código empregado para o processamento do arquivo. A única exigência é que cada classe implementada tenha um método denominado, por exemplo, “*executa*” que será chamado toda vez que o nome da entidade for separado da linha lida.

Métodos diferentes devem ser moldados de acordo com as especificidades de cada entidade. Não obstante, eles devem se comportar de forma similar em três aspectos. O primeiro refere-se ao armazenamento do identificador, já separado em B1. Esse armazenamento é importante, pois será por meio dele que as relações topológicas e geométricas serão estabelecidas, tal como apresentado na Fig. (1A). O segundo aspecto está na forma de separação dos atributos, a qual deve se basear no processamento do bloco B4, caractere por caractere e tendo os parêntesis (“(”, “)”) e o apóstrofo “'” como marcadores. O outro aspecto está relacionado ao conteúdo retornado pelo método, isto é, o residual de B4. Esse conteúdo pode ser, no mínimo, o caractere “;” para o caso de entidades simples (*internal mapping*). Entretanto, ele pode conter outras entidades para o caso de “*external mapping*”. Nesse caso, B4 é atualizado com esse conteúdo, convertido em B2 e reprocessado de forma recursiva.

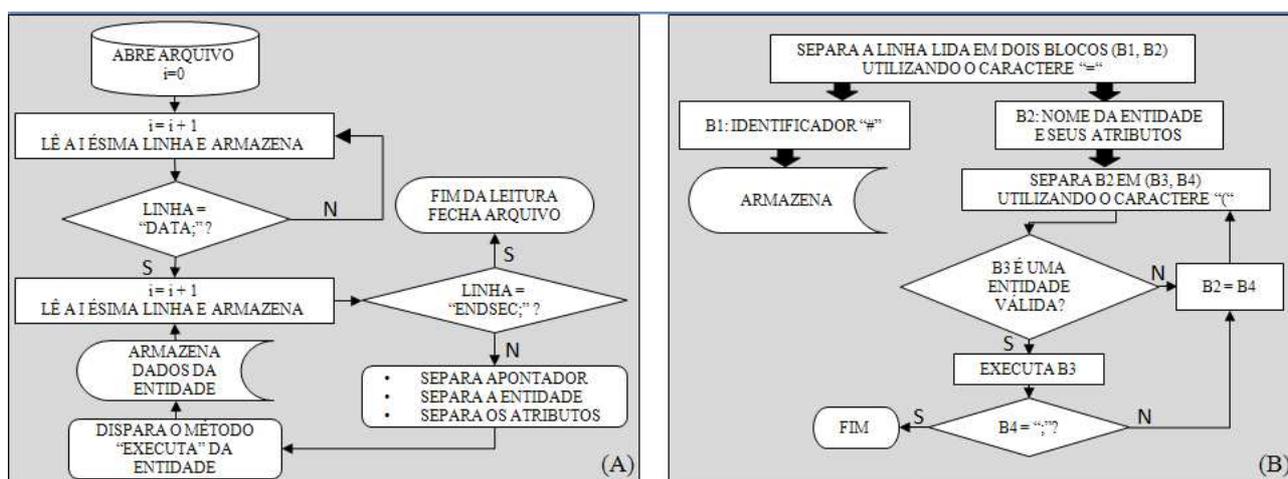


Figura 2. Pseudocódigos. (A) Processamento das linhas. (B) Extração das entidades e seus atributos.

2.3. Conversão dos dados oriundos do formato STEP para uma estrutura interna proprietária

A opção por uma estrutura de dados própria se justifica, embora seja ela redundante do ponto de vista geométrico, pelo seguinte motivo: o corte de chapas, tal como delimitado aqui, ocorre sobre um plano definido em duas dimensões, algo que não está explicitado na modelagem 3D. Diante disso, a estrutura aqui proposta funcionará como uma interface entre o modelo 3D em B-rep e o gerador de trajetórias já desenvolvido e reportado por Marques et al. (2011).

Após a separação das entidades geométricas e topológicas oriundas do modelo em STEP, as mesmas são processadas para que uma única face plana, aquela sobre a qual se dará o corte, seja encontrada e os seus contornos internos — referentes às entidades *FACE_BOUND* — e o externo definido pela entidade *FACE_OUTER-BOUND* sejam extraídos.

A face de corte e os seus contornos serão armazenados em cinco classes específicas, denominadas aqui por *FACE*, *CTN*, *RETA*, *ARCO* e *CIRCULO*, tal como apresentado na Fig. (3). A primeira classe — *FACE* — possui cinco propriedades: **Adv** para manter um registro da face original (*ADVANCED_FACE*); um apontador para seu contorno externo (**Cext**); uma lista de apontadores para os contornos internos (**Cint**); a normal (**Nr**), dada por um versor recuperado da face original e o ponto de referência (**Pr**), também oriundo da entidade original.

A classe *CTN* é empregada para manter o registro e organização dos contornos. Cada objeto instanciado a partir dessa classe terá as seguintes propriedades: uma lista (**Plygn**) com os apontadores para os elementos geométricos que compõem o contorno e definidos pelas classes *RETA*, *ARCO* e *CIRCULO*; o tipo (**Tp**) do contorno se composto por

um círculo, somente por retas, somente por arcos ou misto (arcos e retas); a área (**Area**); a posição (**Pos**) interna ou externa em relação ao contorno e o retângulo envoltório (**Caixa**).

As classes **RETA**, **ARCO** e **CIRCULO** possuem propriedades geométricas bidimensionais e que serão utilizadas para o cálculo da trajetória de corte. Uma propriedade comum às três é o identificador (**Id**). Esse identificador é o mesmo que aparece na lista **Plygn** da classe **CTN** e é armazenado externamente em uma lista denominada **AGER**, não apresentada na Fig (3). Essa lista, que pode ser entendida como uma classe superior, estabelece o relacionamento entre os contornos e seus elementos geométricos por meio de duas propriedades: **Id**, que mantém a correspondência e o tipo (**Tp**) de elemento (reta, arco ou círculo) incorporado.

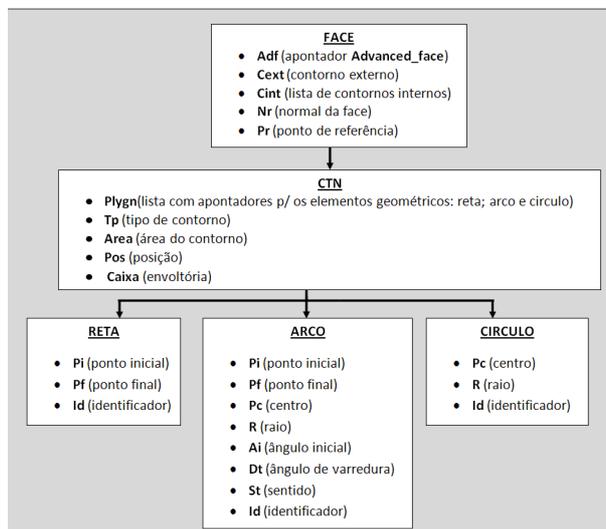


Figura 3. Objetos da estrutura própria e seus relacionamentos

O preenchimento da estrutura própria delineada anteriormente se inicia com o processamento da entidade **CLOSED_SHELL**, tal como esquematizado por meio da Fig. (4A). O objeto **CLOSED_SHELL** possui uma lista com as faces (**ADVANCED_FACE**) que compõem a chapa. Após a recuperação da primeira face é verificado se a sua superfície é plana. Superfícies não planas serão descartadas, tal como delimitado no escopo desta metodologia.

Para cada objeto **ADVANCED_FACE** válido deve-se criar uma instância da classe **FACE** e encontrar a lista de arestas (**EDGE_LOOP**) relativas às suas fronteiras internas (**FACE_BOUND**) e externas (**FACE_OUTER_BOUND**), tal como apresentado na Fig. (4B). Para cada **EDGE_LOOP** um novo objeto da classe **CTN** será instanciado. Cada objeto **EDGE_LOOP** contém uma lista de arestas (**ORIENTED_EDGE**) que devem ser processadas individualmente, tal como esquematizado por meio da Fig. (5).

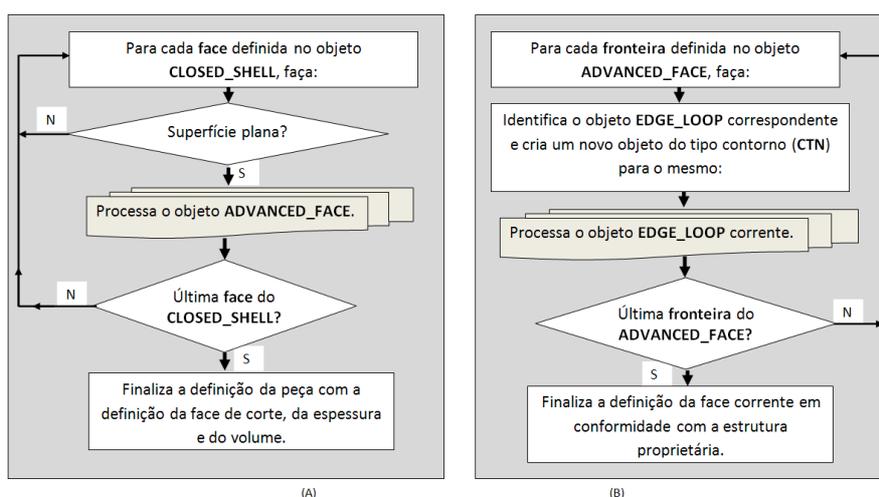


Figura 4. Pseudocódigos. (A) – processamento das faces. (B) – processamento das fronteiras.

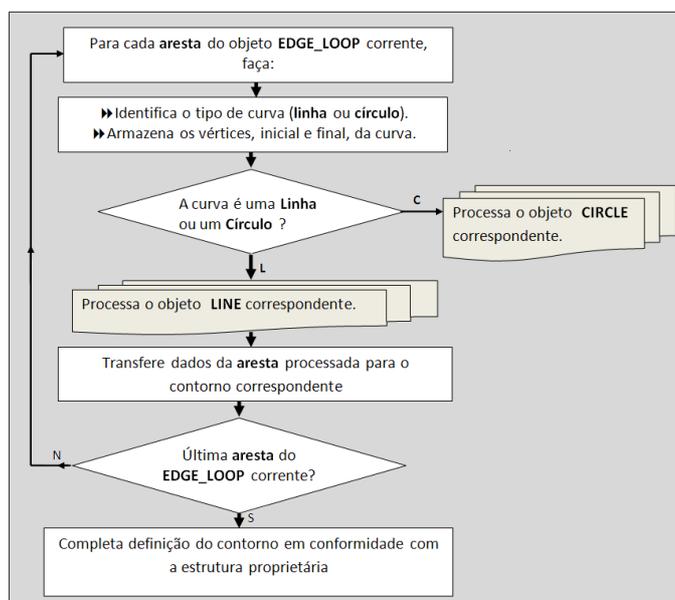


Figura 5. Pseudocódigo para o processamento das arestas.

Para cada objeto `ORIENTED_EDGE` encontrado são identificados os vértices (`VERTEX`) que definem suas extremidades, sua orientação topológica e o tipo de curva (linha ou círculo) sobre a qual a aresta foi definida. A posição, inicial ou final, dos vértices é definida de acordo com a orientação topológica da aresta. Se ela for verdadeira (`TRUE`), então o vértice inicial será o primeiro vértice listado e o final será o segundo. Caso contrário (`FALSE`) os vértices serão invertidos. Isto significa dizer que a aresta está sendo percorrida, na formação do contorno, no sentido oposto à sua definição original.

Após a identificação do tipo de curva, linha ou círculo, a mesma será processada pelo método correspondente, vide figuras (6A) e (6B). Deve-se observar que a estrutura de dados `STEP` não define, diretamente, a entidade “arco”, mas sim como parte de um círculo limitado por dois vértices, denominado “trimmed curve”. Além disso, quando a curva for um círculo completo ela será dividida em dois arcos, cuja soma será 360° . Por conseguinte, as entidades “`CIRCLE`” serão convertidas, para a estrutura interna, em entidades “`ARCO`” ou em “`CIRCULO`”, no caso de dois arcos concêntricos consecutivos, cuja soma seja 360° .

Os vértices de ambas as curvas mencionadas anteriormente são definidos por pontos cartesianos em 3D. Essas coordenadas têm como origem, e orientação, o sistema de coordenadas global do componente (`SCG`). Entretanto, as faces que são limitadas por essas curvas possuem um sistema de coordenadas local (`SCL`). Diante disso, torna-se necessário efetuar uma transformação rígida, do sistema global para o local, para que as curvas possam ser definidas em duas dimensões.

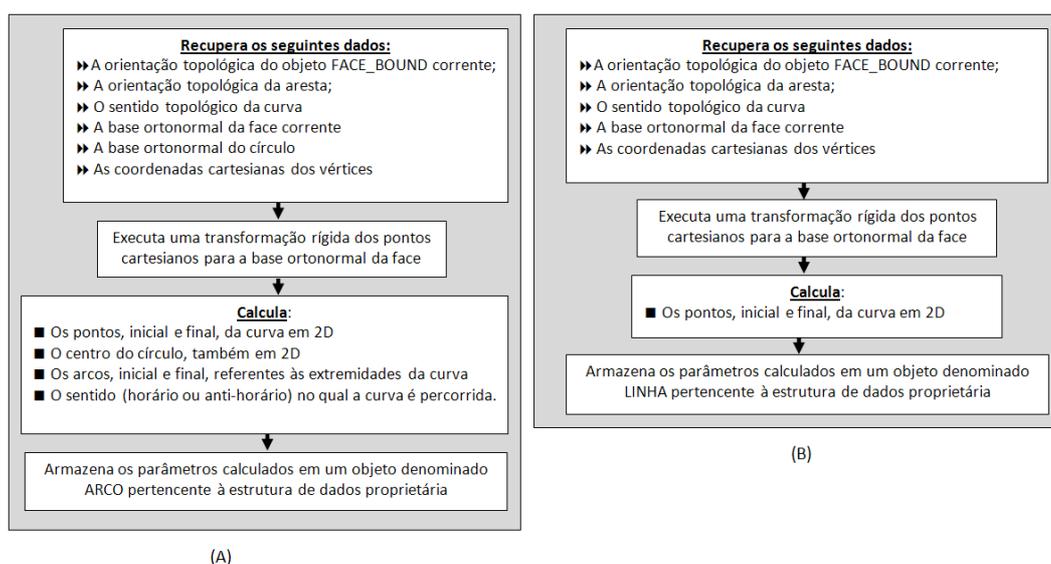


Figura 6. Pseudocódigos. (A) processamento dos arcos. (B) processamento das linhas.

Um ponto de interesse (`P`) pertencente à curva definida pelo `SCG` do componente pode ser definido, em relação ao `SCL` da face correspondente, por meio da Eq. (1). Nesta, os versores \hat{u} e \hat{v} definem o plano da face e as direções (x, y)

respectivamente. Eles são obtidos a partir da entidade “AXIS2_PLACEMENT_3D” correspondente. Essa entidade é uma base ortonormal definida, indiretamente, por um ponto cartesiano de referência (\mathbf{Q}), um versor normal ($\hat{\mathbf{n}}$) que dá a direção do eixo \mathbf{z} e outro versor ($\hat{\mathbf{a}}$) para a localização aproximada do eixo \mathbf{x} . O versor $\hat{\mathbf{u}}$, que é o unitário de \vec{U} , dá a direção correta do eixo \mathbf{x} e é obtido por meio da Eq. (2), tal como recomendado pela ISO 10303-42. O versor $\hat{\mathbf{v}}$ é calculado pelo produto vetorial $\hat{\mathbf{n}} \times \hat{\mathbf{u}}$ e define a direção do eixo \mathbf{y} .

As novas coordenadas (\mathbf{x} , \mathbf{y}) para o ponto de interesse são obtidas pela solução da Eq. (1), tal como proposto por Schneider e Eberly (2003), e expressas pelas equações (3) e (4).

$$P = Q + x\hat{\mathbf{u}} + y\hat{\mathbf{v}} \quad (1)$$

$$\vec{U} = \hat{\mathbf{a}} - (\hat{\mathbf{a}} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} \quad (2)$$

$$x = \hat{\mathbf{u}}(P - Q) \quad (3)$$

$$y = \hat{\mathbf{v}}(P - Q) \quad (4)$$

Além dos parâmetros geométricos — centro, raio e vértices — os arcos devem ser definidos quanto ao sentido: horário ou anti-horário. Para isso, deve-se realizar uma análise conjunta da orientação topológica da aresta; do sentido topológico da curva e do sentido geométrico resultante das bases ortonormais da face e do círculo.

Dentro das condições aqui impostas à modelagem das chapas, uma aresta deve ser utilizada duas vezes na definição de duas faces adjacentes. Se a geometria da aresta for um círculo, então existirá associada à mesma uma entidade “AXIS2_PLACEMENT_3D”, a partir da qual se pode obter uma base ortonormal. A normal ($\hat{\mathbf{n}}_c$) dessa base é paralela à geratriz do cilindro. Essa normal e aquela referente à face plana ($\hat{\mathbf{n}}_f$) do cilindro coincidem em direção, mas podem divergir quanto ao sentido.

A primeira vez que uma “aresta circular” for utilizada na definição de uma face, o sentido da normal ($\hat{\mathbf{n}}_c$) coincidirá com ($\hat{\mathbf{n}}_f$) se, quando visto sobre essa face, o sentido da aresta (EDGE_CURVE) for o mesmo da curva base (CIRCLE). Na norma ISO10303-42 um círculo (C) de raio (R) é definido por meio da Eq. (5) e o seu sentido positivo de varredura é dado pelo sentido do vetor tangente (T) a um ponto (x, y) do círculo, tal como dado pela Eq. (6).

$$C(x, y, z) = x^2 + y^2 - R^2 \quad (5)$$

$$T = (-C_y, C_x, 0) \quad (6)$$

Diante do exposto, pode-se estabelecer um atributo de orientação (*OrtNrm*) para as normais $\hat{\mathbf{n}}_c$ e $\hat{\mathbf{n}}_f$. Se elas forem coincidentes, *OrtNrm* será verdadeiro; caso contrário será falso. Uma maneira de se determinar o sentido do arco (StA) é pela aplicação da Eq. (7), a qual é baseada no operador “não ou exclusivo \odot “. O sentido será anti-horário se StA for verdadeiro e horário, caso contrário. Essa equação é similar àquela proposta na ISO-10303-42 para a determinação da orientação (ou sentido) de entidades topológicas. Nessa equação, *OrtOEd* e *SmsCrv* representam a orientação topológica da aresta (ORIENTED_EDGE) e o sentido topológico da curva (EDGE_CURVE), respectivamente.

$$StA = OrtOEd \odot SmsCrv \odot OrtNrm \quad (7)$$

3. IMPLEMENTAÇÃO, RESULTADOS E DISCUSSÃO

A metodologia proposta foi implementada em Visual Basic Express[®] (Microsoft, USA - Versão 2010) e incorporado a um protótipo já existente, tal como mencionado na seção “Introdução”. Vários modelos de chapas foram desenhados e convertidos para o formato STEP com os softwares Catia[®] e Solidworks[®].

A Figura (7) contém uma imagem obtida de uma das peças modeladas em 3D com o Solidworks[®]. Essa peça é similar àquela avaliada por Marques et al. (2011) em DXF. A referida peça obedece às restrições impostas nesta metodologia. A Figura (8) contém uma imagem da tela do protótipo criado contendo uma vista da face de corte da referida peça; propriedades da chapa e o programa CN gerado, considerando uma sangria de 2mm.

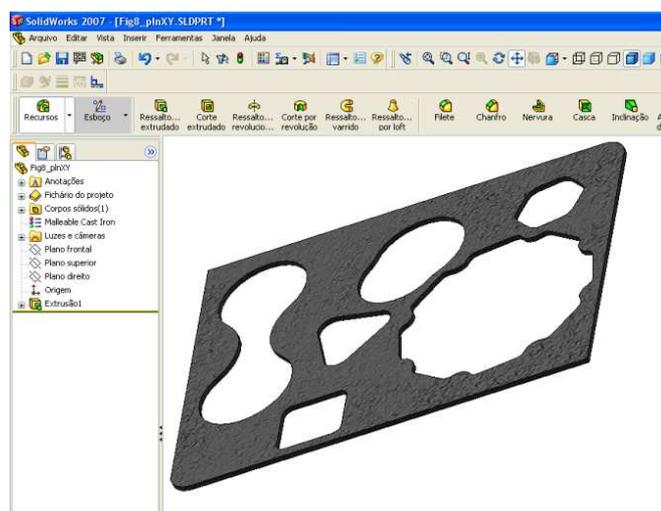


Figura 7. Modelagem de uma das peças avaliadas.

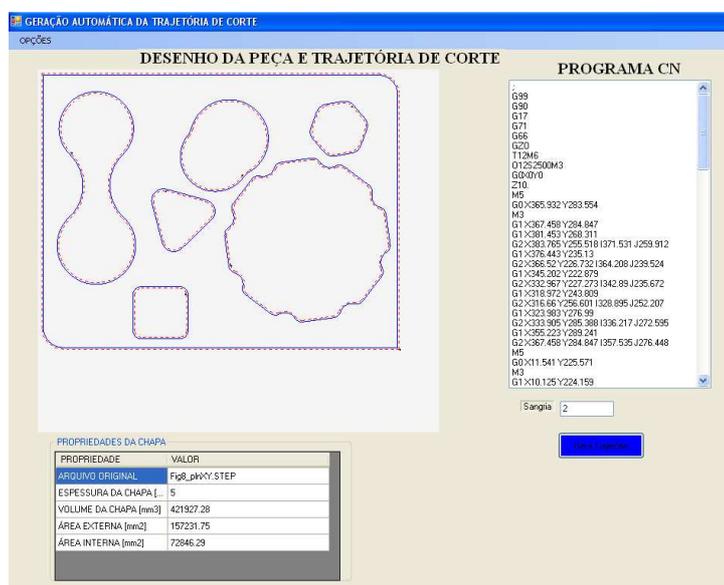


Figura 8. Imagem da tela do protótipo desenvolvido.

Um conjunto de peças contendo elementos similares aos contornos internos da peça representada na Fig. (7) e Fig. (8) foi modelado com os softwares Solidworks® e Catia® e convertidos em STEP por meio de seus respectivos pós-processadores. O software Catia® exporta diretamente para um arquivo em conformidade com a ISO 10303 - AP 203. No caso do Solidworks®, existem duas opções: o AP203 e o AP214. Por conseguinte, o conjunto modelado no Solidworks® foi exportado para os dois AP's disponíveis.

Nas três situações mencionadas anteriormente, o protótipo implementado produziu os mesmos resultados; isto é, foram encontradas as mesmas faces de corte, os mesmos contornos e foram gerados códigos CN idênticos. Não obstante, deve-se registrar que para uma mesma peça foram produzidos três arquivos STEP diferentes. Diferenças entre os arquivos gerados em conformidade com os AP's 203 e 214 são esperadas, pois os mesmos têm propósitos distintos. Todavia, do ponto de vista geométrico e topológico, eles são idênticos, quando gerados pelo Solidworks®.

Entretanto, quando se comparou os arquivos oriundos de protocolos similares, mas de softwares distintos, foram observadas três diferenças relativas às entidades geométricas e topológicas. A primeira refere-se à inexistência da entidade "AXIS2_PLACEMENT_3D", referenciada por "ADVANCED_BREP_SHAPE_REPRESENTATION", nos arquivos oriundos do Catia®. A segunda é a ausência, também no Catia®, da direção auxiliar "ref_direction" associada à entidade "AXIS2_PLACEMENT_3D" dos círculos. Por último, observaram-se dois métodos distintos para a indicação do sentido dos arcos. Nos arquivos exportados com o Catia® isto é feito, exclusivamente, por meio de um atributo

booleano (*same_sense*) definido na entidade topológica “EDGE_CURVE”, o qual é o equivalente ao atributo “*SmsCrv*” da Eq (7). Por outro lado, no Solidworks® o sentido do arco é dado, exclusivamente, pelas direções principais (\hat{n}_c e \hat{n}_f) da base ortornormal do círculo e da face, respectivamente.

Para contornar os problemas decorrentes da primeira e segunda diferenças mencionadas anteriormente, optou-se por implementar a etapa de pré-processamento, tal como descrito na seção 2.2, utilizando a técnica de ligação tardia (*late binding*). Isto possibilitou o estabelecimento de uma correspondência entre as entidades encontradas no arquivo, independente de sua procedência, com aquelas consideradas necessárias e suficientes para a representação topológica-geométrica das chapas. Dessa forma, entidades e/ou atributos que são considerados como opcionais, ou relacionados com informações complementares, são descartados durante a leitura dos arquivos por não possuírem um método “**executa**” implementado.

Além do motivo exposto anteriormente, o emprego da técnica de ligação tardia facilita a análise das diferenças entre modelos criados por softwares diferentes. Essa análise pode ser feita pela comparação da lista de entidades presentes ou ausentes em um determinado arquivo. A referida lista pode ser obtida facilmente, pois não existe a necessidade de programar suas entidades como classes em tempo de projeto. Por conseguinte, qualquer entidade que, futuramente, for considerada de interesse poderá ser processada pela simples inclusão de um método “**executa**” dedicado à mesma, sem a necessidade de modificar o programa já existente.

A implementação da etapa de pré-processamento com a técnica de ligação tardia possibilitou a criação de dois conjuntos de entidades. O primeiro contém as entidades relacionadas diretamente com a geometria e topologia do componente, tal como esquematizado por meio da Fig. 1(A). O segundo conjunto contém as entidades consideradas aqui como complementares e que, no estágio atual de implementação, foram descartadas.

O problema decorrente da terceira diferença relatada inviabiliza a utilização da técnica de ligação tardia para a programação da segunda parte da metodologia, tal como descrito na seção 2.3. Diante disso, optou-se pela utilização da ligação antecipada (*early binding*), ou seja, a conversão dos dados oriundos do formato STEP para uma estrutura interna proprietária foi programada em tempo de projeto.

A partir da opção pela ligação antecipada, a definição do sentido dos arcos foi resolvida pela aplicação da Eq. (7). Nos arquivos oriundos do software Catia®, o atributo *OrtNrm* foi sempre **verdadeiro**, pois as normais da face e do círculo apontavam no mesmo sentido. Em contrapartida, o atributo *SmsCrv* foi sempre **verdadeiro** nos modelos gerados pelo Solidworks®. Por conseguinte, independente do software utilizado, apenas dois atributos foram considerados na análise do sentido do arco. O terceiro, por ser sempre **verdadeiro**, não influenciou no resultado final.

Como conclusão, pode-se dizer que a metodologia proposta, sua implementação computacional e os testes realizados até o presente possibilitaram alcançar o objetivo traçado neste trabalho. Isto é, a face a ser cortada na chapa, assim como a trajetória para o corte de seus contornos, podem ser extraídos e programados automaticamente, a partir de modelos 3D sólidos formatados em STEP.

4. REFERÊNCIAS

- DeGarmo, P.E., Black, J.T. and Kohser, R.A., 1997, “Materials and Processes in Manufacturing”, Prentice Hall, New Jersey, USA, 1259p.
- Flow Corporation, disponível em <http://www.flowwaterjet.com>, acesso em 01/02/2012.
- Foggiatto, J. A., Volpato, N. e Bontorin, A. C. B., 2007, “Recomendações para Modelagem em Sistemas CAD-3D”, 4º Congresso Brasileiro de Engenharia de Fabricação, Estância de São Pedro, Brasil.
- Gielingh, W., 2008, “An assessment of the current state of product data technologies”, Computer-Aided Design, Vol. 40, Issue 7, pp. 750-759.
- ISO, ISO 10303-21, 2002. “Industrial automation systems and integration -- Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure”.
- ISO, ISO 10303-42, 2003. “Industrial automation systems and integration -- Product data representation and exchange - Part 42: Integrated generic resource: Geometric and topological representation”.
- ISO, ISO 10303-214, 2010. “Industrial automation systems and integration -- Product data representation and exchange - Part 214: Application protocol: Core data for automotive mechanical design processes”.
- ISO, ISO 10303-203, 2011. “Industrial automation systems and integration -- Product data representation and exchange - Part 203: Application protocol: Configuration controlled 3D design of mechanical parts and assemblies”.
- Jackson, S. D., Mittal, R. O., 1993, “Automatic generation of 2-axis laser-cutter NC machine program and path planning from CAD”, Computers in Industry, Vol. 21, Issue 2, pp. 223-231.
- Marques, A., Da Costa, D.D., Dos Santos, T.M.A., 2011, “Geração Automática da Trajetória de Corte a Partir de Arquivos DXF”, 6º Congresso Brasileiro de Engenharia de Fabricação, Caxias do Sul, Brasil.
- Schneider, P. J., Eberly, D. H., 2003, “Geometric tools for computer graphics”, Morgan Kaufmann Publishers, Amsterdam: Boston, 1009 p.
- Trumpf GmbH, disponível em <http://www.trumpf.com/>, acesso em 01/02/2012.

5. DIREITOS AUTORAIS

Os autores são os únicos responsáveis pelo conteúdo do material impresso incluído no seu trabalho.



COMPROMETIDA COM A PROMOÇÃO DO DESENVOLVIMENTO
DA ENGENHARIA E DAS CIÊNCIAS MECÂNICAS

7º CONGRESSO BRASILEIRO DE ENGENHARIA DE FABRICAÇÃO
7th BRAZILIAN CONGRESS ON MANUFACTURING ENGINEERING
20 a 24 de maio de 2013 – Penedo, Itatiaia – RJ -Brasil
May 20th to 24th, 2013–Penedo, Itatiaia – RJ – Brazil

CNC PROGRAMMING OF SHEET METAL MODELED AS B-REP SOLIDS

Dalberto Dias da Costa, dalberto@ufpr.br¹
Gilberto Passos Lima, gibapassos@gmail.com²

¹Universidade Federal do Paraná, Departamento de Engenharia Mecânica, Curitiba - PR

² Universidade Federal do Paraná, Pós-graduação em Engenharia de Produção, Curitiba – PR

Abstract. *The application of Computerized Numerical Control (CNC) to programming sheet metal cutting is, nowadays, a common practice in the metalworking industries. The CNC machines are applied as a complementary resource to several cutting techniques, such as: laser, waterjet and plasma. The process starts from a CAD (Computer-Aided Design) model which is exported to a CAM (Computer-Aided Manufacturing) system. The CAM system is justified as essential, mainly to generate the tool path in 3D cutting. Generally, DXF (Drawing eXchange Format) is selected as a medium to data transfer between the CAD-CAM systems to cutting 2D parts. Despite of its simplicity, the DXF format does not contain topological information related to the modeled part and its data structure was not conceived to cover the whole product life cycle. On the other hand, the continuous acceptance of 3D modeling, in special the geometric solid modeling with Boundary Representation (B-Rep), has demanded new media to interface the CAD-CAM systems. In this new scenario, the use of more complete and changelessness format, such as the STEP (Standard for the Exchange of Product Model Data) defined according to ISO 10303, has become an obligation. However, it has raised two important questions: how to programming machine and systems that are still based on 2D? And also, how the planar cutting surface can be, automatically, recognized and extracted? The purpose of this work is to answer to these questions. The present paper presents a methodology to separate the planar faces of a B-Rep model stored according the STEP format and extracted their internal and external contours. After that procedure, the cutting path is calculated, taking in account the necessary offset to compensate the kerf width. A computational prototype was implemented with MS –Visual Basic and evaluated with parts modeled using two commercial CAD systems.*

Keywords: *Sheet metal cutting, B-Rep, STEP, NC programming*