

A Computational Procedure for Interpolation of Triangular Parametric Surfaces

Thales M. Vieira

Clarissa C. Marques

Adelailson Peixoto

Hilário Alencar

thales@pos.mat.ufal.br

clarissa@pos.mat.ufal.br

adelailson@pos.mat.ufal.br

hilario@pos.mat.ufal.br

Departamento de Matemática, Universidade Federal de Alagoas

BR 104 Norte, Km 97, 57072-970, Maceió - AL - Brazil

William W. Lira

william@ctec.ufal.br

Departamento de Engenharia Estrutural, Universidade Federal de Alagoas

BR 104 Norte, Km 97, 57072-970, Maceió - AL - Brazil

Ailton Cruz

ailton@tci.ufal.br

Departamento de Tecnologia da Informação, Universidade Federal de Alagoas

BR 104 Norte, Km 97, 57072-970, Maceió - AL - Brazil

Abstract. *This work presents a computational procedure to represent triangular parametric surfaces. These surfaces are defined by three boundary curves, and they are usually used in geometric modeling of engineering real problems. The representation of triangular surfaces using conventional methods, like NURBS (Non Uniform Rational B-Splines), is not a trivial task, because the description of the parametric space normally requires rectangular domains, which cannot be applied to treatment of triangular surfaces. The procedure proposed in this work is divided into three parts. The first is related to the surface's boundary curves discretization and the generation of a triangular elements mesh that is used for surface representation. The second part uses an algorithm to establish a discrete parametric space associated to mesh. The last part uses a method to recover Cartesian coordinates and derivatives values associated with any parametric coordinate given, assuring a smooth transition in the surface's parametric space. This procedure is implemented in C++, using the paradigm of object-oriented programming, which makes reuse and expansion of computational code easier. To validate the proposed method, a graphic environment for visualization of curves and surfaces is presented, allowing interactive object handling.*

Keywords: *Geometric Modeling, Triangular Parametric Surfaces, NURBS.*

1. INTRODUÇÃO

As curvas e superfícies NURBS (*Non Uniform Rational B-Splines*) são atualmente o padrão da indústria para a representação, design e troca de dados referentes a objetos geométricos processados por computador. Com essa técnica, é possível obter representações de curvas e superfícies contínuas com um conjunto finito de pontos, permitindo a recuperação sem perdas de qualquer informação do modelo matemático.

Dentre os diversos problemas estudados no campo da Modelagem Geométrica, constam as superfícies paramétricas. Essas superfícies são definidas por parametrizações $\varphi: D \rightarrow \mathbb{R}^3$ que mapeiam $(u, v) \rightarrow (x^1(u, v), x^2(u, v), x^3(u, v))$, onde $D \subset \mathbb{R}^2$ é um conjunto aberto denominado espaço paramétrico (Fig. 1).

Dentre as superfícies paramétricas mais utilizadas na modelagem geométrica de problemas reais de Engenharia estão as superfícies triangulares (Fig. 2). Estas apresentam certas particularidades que dificultam suas modelagens usando técnicas convencionais, como NURBS, e são caracterizadas por possuírem três curvas de bordo delimitando seu contorno.

Normalmente, as superfícies paramétricas são construídas a partir de um espaço paramétrico retangular, como ocorre com as NURBS. Porém, não é possível parametrizar uma superfície triangular a partir de um retângulo, pois estes conjuntos não são homeomorfos.

Diversas abordagens vêm tentando resolver o problema das superfícies paramétricas triangulares. Uma nova definição para NURBS, chamada NURBS triangular, foi proposta por Qin e Terzopoulos (1997) e permite a modelagem de superfícies usando um espaço paramétrico arbitrário não retangular.

Uma outra abordagem usada atualmente é apresentada por Floater (1997). Em seu trabalho, Floater mostra uma maneira de se criar um espaço paramétrico discreto para uma superfície representada por triangulações (superfície triangulada). Naturalmente, o espaço paramétrico discreto obtido requer o uso de outros métodos para obtenção de um ponto da superfície, dado um ponto qualquer do espaço paramétrico.

Este trabalho apresenta um procedimento computacional para representação de superfícies paramétricas triangulares. Essas superfícies são obtidas a partir de suas curvas de bordo. O procedimento utilizado é baseado em três etapas. A primeira etapa

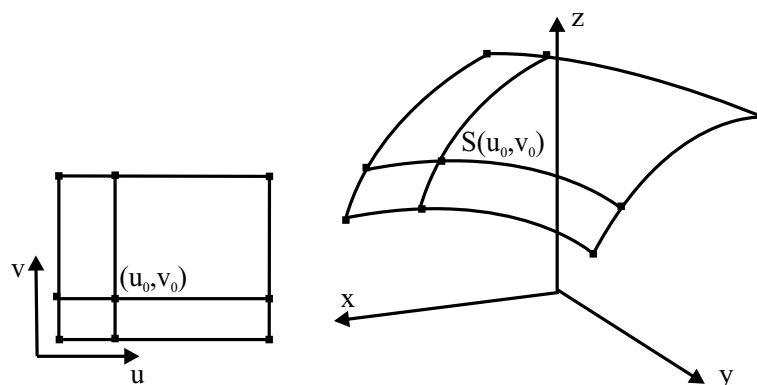


Figura 1: Espaço Paramétrico e Superfície Paramétrica.

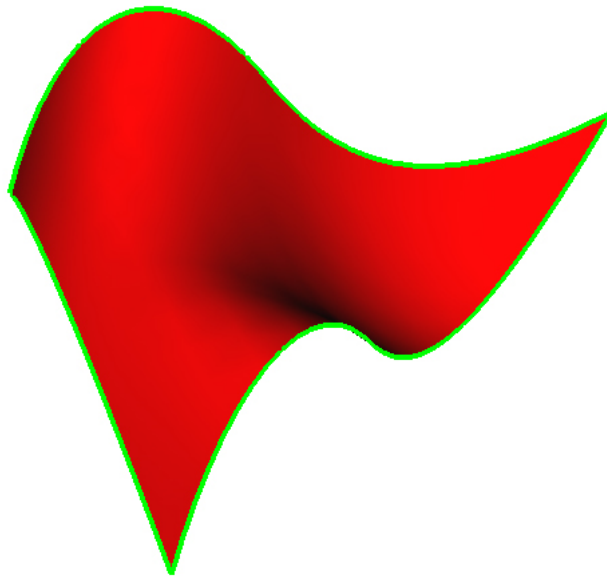


Figura 2: Superfície Paramétrica Triangular.

é responsável pela discretização das curvas de contorno e geração de uma malha de elementos triangulares. Com esse propósito, foi utilizada uma técnica de mapeamento transfinito proposta por Gordon e Hall (1973). A segunda etapa refere-se à criação de um espaço paramétrico discreto para a malha obtida no primeiro módulo. Essa etapa se baseia no trabalho de Floater. A terceira etapa está relacionado com a recuperação de pontos da superfície, e serão utilizadas duas idéias: recuperação usando coordenadas baricêntricas e recuperação usando SPR (*Superconvergent Patch Recovery*, Zienkiewicz e Zhu (1994)). Este último conceito é aplicado em análise numérica, pelo método dos Elementos Finitos (Bathe, 1982), e foi adaptado para o problema em questão. A Fig. 3 ilustra os três passos do procedimento.

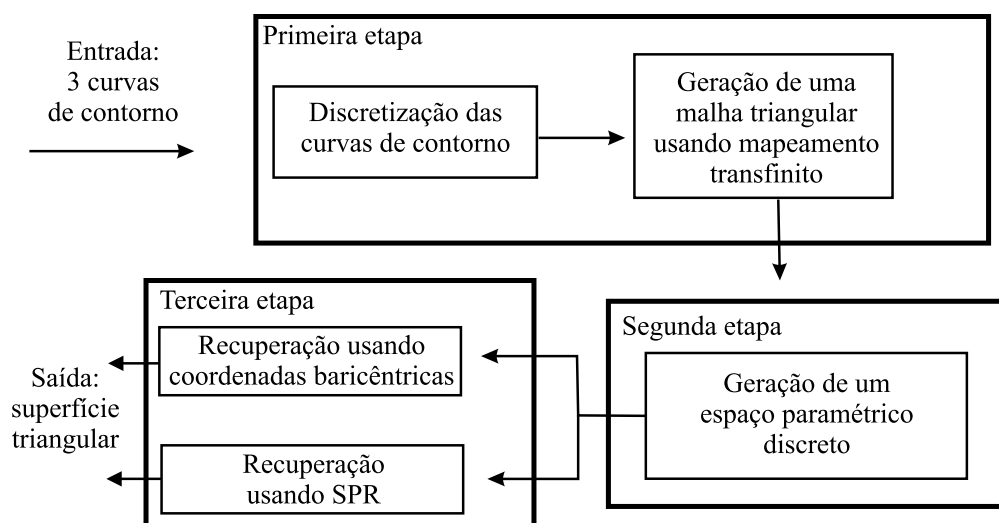


Figura 3: Procedimento para geração de uma superfície paramétrica triangular.

É interessante ressaltar que, com o propósito de gerar superfícies paramétricas triangulares, este trabalho reúne quatro procedimentos independentes. As implementações computacionais das duas primeiras etapas já existem, mas são utilizadas separadamente com outros objetivos. A originalidade deste trabalho está na integração desses procedimentos com o objetivo de gerar superfícies paramétricas triangulares. Além disso, também é original deste trabalho a adaptação da técnica de recuperação usando SPR.

Finalmente, foi desenvolvida uma biblioteca para modelagem de superfícies triangulares, usando a linguagem C++ (Stroustrup, 1991) e a API OpenGL (Shreiner et al., 2003). Essa biblioteca promove a integração dos três módulos, tanto com a recuperação usando coordenadas baricêntricas como com a recuperação usando SPR.

2. MAPEAMENTOS TRANSFINITOS

As técnicas de mapeamentos transfinitos são uma classe de métodos para estabelecer sistemas de coordenadas curvilíneas definidos pelo contorno de domínios arbitrários. O primeiro método associado a essas técnicas foi desenvolvido inicialmente por Gordon e Hall (1973). Neste trabalho, foram abordadas especificamente as regiões triangulares limitadas por três curvas de contorno.

Essas regiões triangulares são obtidas a partir de projetores lineares. Sejam u, v e w o sistema de coordenadas de área triangular normalizada, estabelecido sobre o interior da região, sujeito à seguinte restrição: $u + v + w = 1$. Uma região triangular fica bem definida pelo projetor trilinear

$$S(u, v, w) = \frac{1}{2} \left[\left(\frac{u}{1-v} \right) \beta(v) + \left(\frac{w}{1-v} \right) \psi(1-v) + \left(\frac{v}{1-w} \right) \psi(w) + \left(\frac{u}{1-w} \right) \alpha(1-w) + \left(\frac{w}{1-u} \right) \alpha(u) + \left(\frac{v}{1-u} \right) \beta(1-u) - w\alpha(0) - u\beta(0) - \psi(0) \right] \quad (1)$$

O conceito de projetor utilizado nesse mapeamento fornece uma maneira natural de se criar uma malha triangular. Os nós da malha gerada são obtidos pela intersecção de isocurvas dessa região. Naturalmente, essas isocurvas são obtidas fixando-se um dos parâmetros u, v ou w . Desse modo, as regiões triangulares delimitadas por essas curvas serão justamente os elementos triangulares da malha.

A implementação dessa etapa usa um algoritmo (Miranda e Martha, 2000) que recebe como entrada uma amostra do contorno e a quantidade n de pontos que cada lado do contorno possui. A estrutura de dados conta com um vetor que armazena os pontos da malha e um vetor que armazena sequencialmente índices dos pontos, de modo que, a cada três índices, um triângulo da malha fica bem definido. Este algoritmo usa o conceito de mapeamento transfinito descrito nessa seção e pode ser dividido em três partes:

1. os pontos do contorno são posicionados em determinadas posições da malha;
2. os pontos interiores são calculados e armazenados no vetor de pontos de acordo com a técnica de mapeamento transfinito;
3. os elementos triangulares são definidos e armazenados no vetor de triângulos.

É interessante observar que esse algoritmo se baseia justamente na idéia de encontrar intersecções de isocurvas do projetor trilinear definido pelas três curvas de contorno. A Fig. 4 ilustra essa etapa.

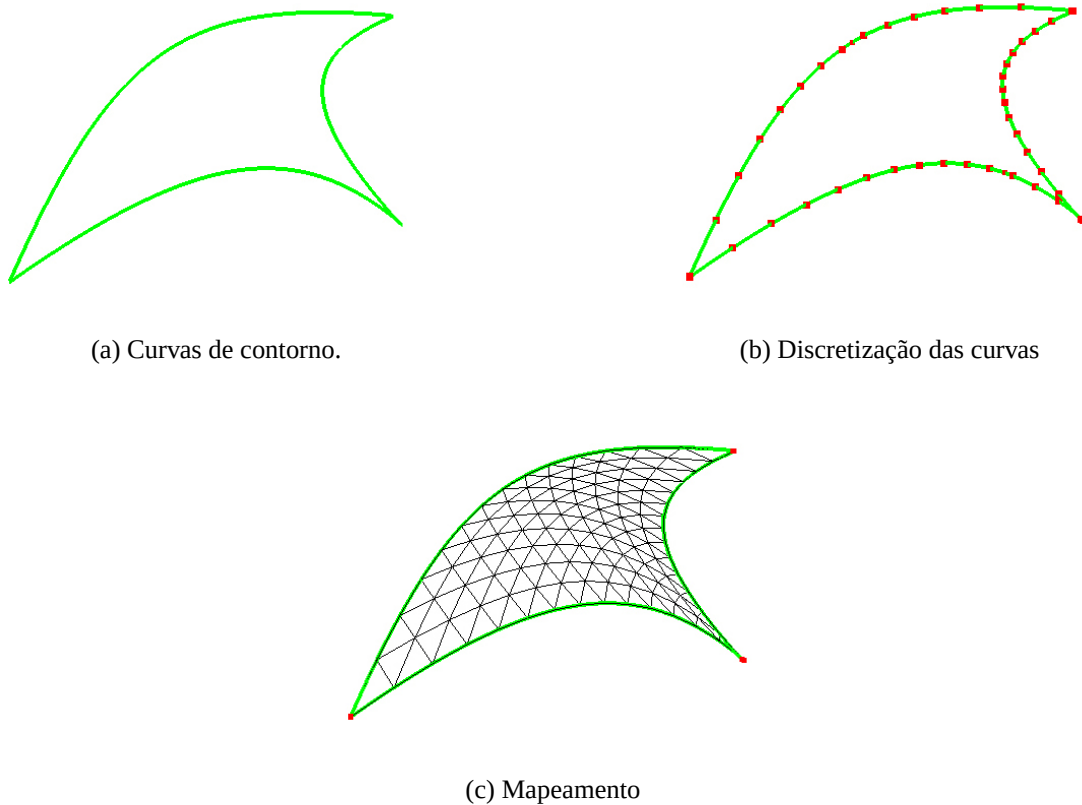


Figura 4: Discretização das curvas de contorno e geração da malha.

3. GERAÇÃO DO ESPAÇO PARAMÉTRICO

A técnica utilizada, desenvolvida por Floater (1997) e implementada por Miranda e Martha (2004), se baseia no trabalho apresentado por Tutte (1963) sobre teoria dos grafos. A solução adotada mapeia cada triângulo da triangulação da superfície em um triângulo no espaço paramétrico. Desse modo, cada ponto (x, y, z) da triangulação é associado a um ponto (u, v) do espaço paramétrico.

Neste trabalho, foi utilizada a biblioteca PFS (Miranda e Martha, 2004), que apresenta uma implementação de acordo com o trabalho de Floater. A entrada de dados da PFS é uma malha topologicamente consistente. O procedimento inicial consiste em encontrar um plano de mínimos quadrados com os *Border Stones* da malha da superfície. Definem-se como *Border Stones* os nós localizados em cantos agudos da malha. Esses nós são calculados automaticamente pela biblioteca PFS.

Em seguida, é criado um sistema de coordenadas locais nesse plano, de modo que o eixo u é orientado do primeiro para o segundo *Border Stone*. Os *Border Stones* são projetados no plano e mapeados para um sistema normalizado nesse mesmo plano.

As curvas de contorno serão mapeadas para retas definidas por *Border Stones*, de modo que os nós do contorno da malha são mapeados respeitando o comprimento do arco ao longo da curva.

A determinação dos valores paramétricos relacionados a nós do interior da malha é feita inicialmente pela projeção desses nós no plano. Porém, valores mais interessantes são obtidos usando alguma forma de parametrização.

Floater propõe várias formas de obter uma parametrização. A diferença entre as formas de parametrização está no modo de calcular fatores de ponderação relativos a um nó e seus nós adjacentes.

A parametrização usando preservação de forma é utilizada neste trabalho. Ela se baseia no uso de combinações convexas. O trabalho de Tutte apresenta uma técnica para desenho de grafos planos com linhas retas usando combinações convexas. Essas combinações devem seguir algumas regras pré-definidas. Mais informações podem ser encontradas em (Floater, 1997). A Fig. 5 exibe uma malha de elementos triangulares e seu espaço paramétrico correspondente.

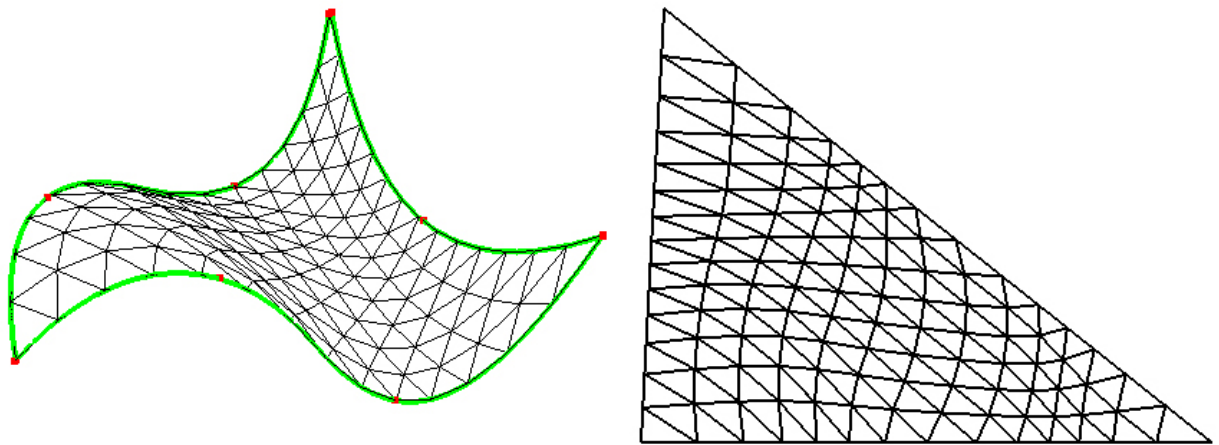


Figura 5: Malha de triângulos e espaço paramétrico gerado usando a PFS.

4. RECUPERAÇÃO USANDO COORDENADAS BARICÊNTRICAS

Esta seção apresenta uma técnica utilizada para recuperar, a partir das coordenadas paramétricas, os valores e derivadas correspondentes na superfície.

Seja T um triângulo formado pelos vértices v_0, v_1, v_2 . Dado um ponto x qualquer do plano, podemos representar x da seguinte maneira:

$$x = av_0 + bv_1 + cv_2, \quad a + b + c = 1, \quad (2)$$

onde (a, b, c) são as coordenadas baricêntricas de x em relação ao triângulo T .

Este conceito apresenta as seguintes propriedades:

- a soma das coordenadas baricêntricas sempre é igual a um;
- as coordenadas baricêntricas são sempre positivas no interior do triângulo;

- se o ponto está localizado em uma aresta do triângulo, uma coordenada será zero;
- se o ponto está localizado em um vértice do triângulo, duas coordenadas serão nulas;
- se o ponto está localizado fora do triângulo, pelo menos uma coordenada é negativa.

A Fig. 6 ilustra o uso de coordenadas baricêntricas para recuperar valores em um triângulo. Neste exemplo, sabe-se que os vértices A, B e C têm valores $f(A), f(B)$ e $f(C)$ associados. Dado um ponto x de coordenadas baricêntricas (a, b, c) no triângulo, é possível achar o valor de $f(x)$ que é dado por $af(A) + bf(B) + cf(C)$.

Usando coordenadas baricêntricas, um ponto qualquer da superfície pode ser obtido da seguinte maneira:

1. dado um ponto $x = (u, v)$ do espaço paramétrico, determina-se o triângulo T , formado pelos vértices v_0, v_1, v_2 , do espaço paramétrico, que contém x ;
2. obtêm-se as coordenadas baricêntricas (a, b, c) do ponto x em relação ao triângulo T .
3. recupera-se os pontos $f(v_0), f(v_1)$ e $f(v_2)$ da superfície correspondentes aos pontos v_0, v_1 e v_2 do espaço paramétrico;
4. o ponto $f(x)$ da superfície é calculado usando as coordenadas baricêntricas de x no triângulo formado pelos pontos $f(v_0), f(v_1)$ e $f(v_2)$. Então:

$$f(x) = af(v_0) + bf(v_1) + cf(v_2). \quad (3)$$

O mesmo procedimento é utilizado para recuperação das derivadas.

É interessante ressaltar que a determinação do triângulo que contém um ponto x é feita testando-se cada triângulo da triangulação, até que se encontre o triângulo no qual as coordenadas baricêntricas sejam todas positivas. Uma maneira de otimizar esse processo consiste em armazenar para cada triângulo informações relativas à sua caixa delimitadora, ou seja, as coordenadas do retângulo que envolve o triângulo (*bounding*

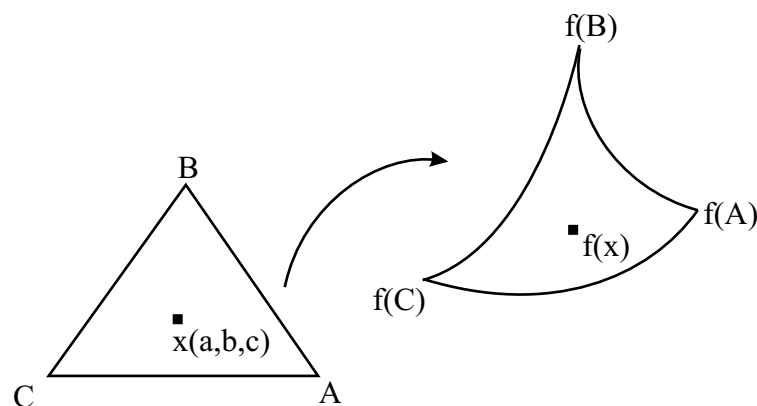


Figura 6: Recuperação de valores usando coordenadas baricêntricas.

box). Dessa maneira, comparações simples são feitas em cada triângulo para se determinar a possibilidade de o ponto estar contido na caixa que envolve o triângulo.

Outra forma para otimizar o processo é a utilização de estruturas de dados especiais (árvores) para armazenar as informações de cada triângulo. Esse procedimento ainda não está implementado neste trabalho.

Por se tratar de uma interpolação linear local em cada triângulo, sem haver qualquer processo de suavização, essa técnica não gera superfícies suaves, mas tem baixo custo computacional. A Fig. 7 exibe uma superfície triangular recuperada usando essa técnica.

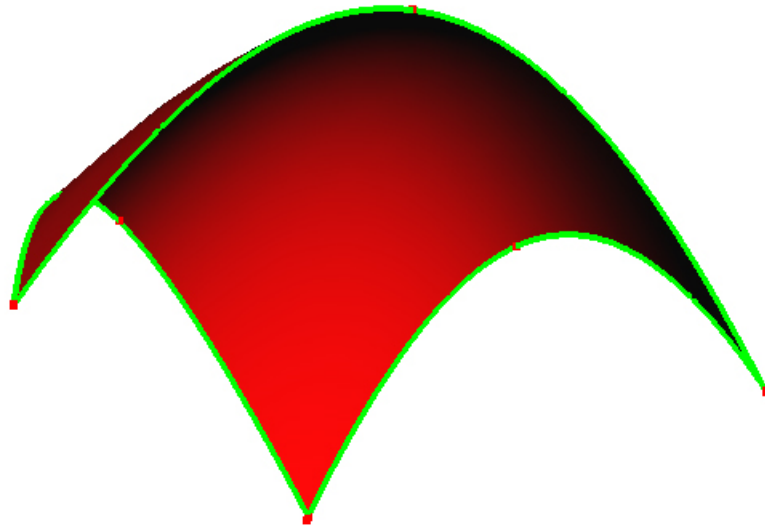


Figura 7: Superfície triangular recuperada por coordenadas baricêntricas.

5. RECUPERAÇÃO USANDO SPR

A recuperação usando SPR é uma técnica que busca obter superfícies mais suaves que as obtidas por recuperação usando coordenadas baricêntricas. Esta seção contém um breve resumo da técnica original e descreve sua adaptação feita neste trabalho.

5.1 SPR original

A técnica SPR (*Superconvergent Patch Recovery*) é aplicada em análise numérica pelo Método dos Elementos Finitos. Essa técnica foi desenvolvida inicialmente por Zienkiewicz e Zhu (1994) e é bastante utilizada em Engenharia, por exemplo, para se obter um campo de tensões aproximado a partir de pontos de amostragem.

A aproximação de um campo genérico é dada por

$$\psi = Pa, \quad (4)$$

onde P contém termos polinomiais e a é um conjunto de parâmetros desconhecidos.

Os termos polinomiais devem ser previamente definidos de acordo com a aplicação. Um exemplo simples é o plano representado por $P = [1, u, v]$. Nesse caso, é necessário calcular os parâmetros $a = [a_0, a_1, a_2]$.

O cálculo do conjunto de parâmetros a se baseia em sistemas de mínimos quadrados ponderados. A amostra considerada para esse cálculo é obtida a partir de grupos de elementos denominados *patches*.

Desse modo, dado um nó qualquer do campo, deve-se inicialmente criar um *patch*. No SPR original, a formação de *patches* segue um conjunto de regras e restrições. A Fig. 8 exibe um *patch* com quatro elementos quadriláteros formado por um nó central. Cada elemento possui quatro pontos de amostragem denotados por um símbolo x. Esses pontos possuem, opcionalmente, um peso associado que é determinado pelo inverso da distância do ponto de amostragem ao nó formador de *patch*. São esses pontos de amostragem que serão levados em consideração para gerar a expansão polinomial.

Então, sendo (u_i, v_i) as coordenadas Cartesianas de um ponto de amostragem i , $\sigma(u_i, v_i)$ o valor associado ao ponto de amostragem e w_i seus pesos, é necessário minimizar a função

$$F = \sum_{i=1}^m w_i^2 [\sigma(u_i, v_i) - \psi(u_i, v_i)]^2. \quad (5)$$

Substituindo a Eq. (4) na Eq. (5), chega-se à forma

$$F = \sum_{i=1}^m w_i^2 [\sigma(u_i, v_i) - P(u_i, v_i)a]^2. \quad (6)$$

Para minimizar essa função, é necessário fazer $\frac{\delta F}{\delta a} = 0$, chegando-se ao sistema de equações lineares

$$Aa = b, \quad (7)$$

onde

$$A = \sum_{i=1}^m w_i^2 P^t(u_i, v_i) P(u_i, v_i) \quad (8)$$

$$b = \sum_{i=1}^m w_i^2 P^t(u_i, v_i) \sigma(u_i, v_i). \quad (9)$$

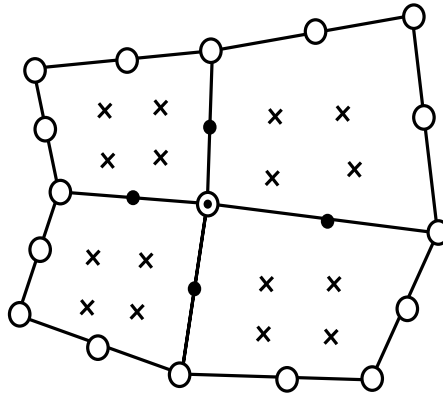


Figura 8: *Patch* gerado por um nó do campo.

Desse modo, é possível calcular o vetor a correspondente ao conjunto de parâmetros do polinômio, calculando-se a matriz inversa de A .

Finalmente, as coordenadas do nó podem ser aplicadas à expansão polinomial para se recuperar o valor aproximado.

5.2 SPR adaptado

A SPR original foi adaptada neste trabalho para a recuperação de pontos da superfície a partir de um conjunto de pontos do espaço paramétrico. Existem algumas diferenças fundamentais entre os dois procedimentos.

No caso da SPR original, os pontos formadores de *patch* são nós do campo. Os elementos contidos no *patch* possuem pontos de amostragem que determinam a expansão polinomial dada pela Eq. (4). Além disso, apenas uma quantidade finita de pontos podem ser determinados, para garantir a convergência dos *patches*.

Porém, no caso da recuperação de pontos da superfície, é necessário se calcular, a partir de um ponto qualquer do espaço paramétrico, suas coordenadas correspondentes na superfície.

Neste trabalho, os *patches* são criados a partir de alguns nós do espaço paramétrico no início do processo, seguindo uma regra que garante que os *patches* sejam todos disjuntos e cubram todos os triângulos do espaço paramétrico. A Fig. 9 exibe um espaço paramétrico triangulado dividido em *patches*.

Cada *patch* do espaço paramétrico tem seu polinômio gerado a partir de uma amostragem formada pelos vértices dos triângulos nele contidos. Neste trabalho, foram considerados polinômios quadráticos com os termos polinomiais $P = [1, u, v, u^2, uv, v^2]$, para geração de superfícies suaves.

Além disso, para a obtenção de superfícies suaves, é necessário garantir a transição suave entre os *patches*. Com essa finalidade, algumas adaptações foram desenvolvidas.

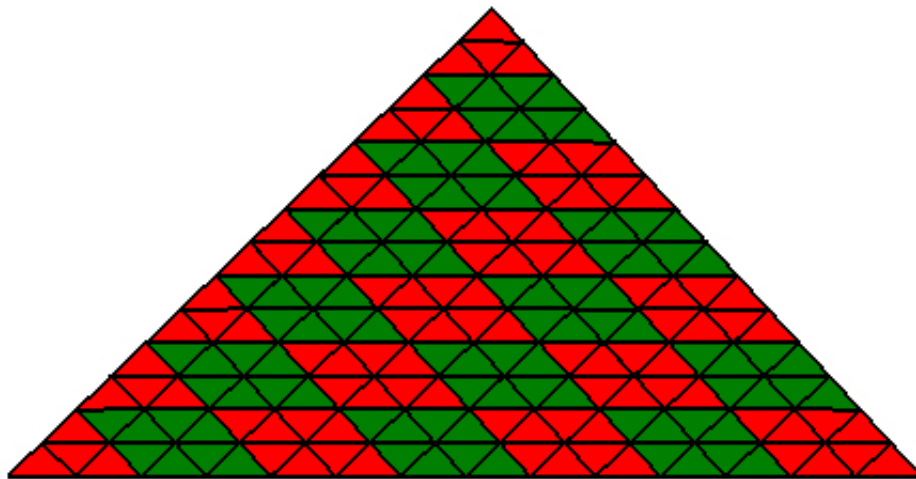


Figura 9: Espaço paramétrico triangulado dividido em *patches*.

Dado um espaço paramétrico dividido em patches, o cálculo de um ponto da superfície a partir de um ponto (u, v) qualquer do espaço paramétrico ocorre da seguinte maneira:

1. determina-se o triângulo T do espaço paramétrico que contém o ponto (u, v) ;
2. determina-se o *patch* P no qual o triângulo está contido;
3. determinam-se todos os triângulos $T_{i,j}$ que compartilham o vértice v_i do triângulo T (inclusive o próprio T);
4. determinam-se os *patches* $P_{i,j}$ que contêm cada triângulo $T_{i,j}$;
5. calcula-se, a partir do polinômio $p_{i,j}$ gerado por cada *patch* $P_{i,j}$, o valor $p_{i,j}(u, v)$;
6. para cada i , é feita uma média M_i dos valores $p_{i,j}(u, v)$ usando todos os patches $P_{i,j}$. Esse cálculo é dado pela equação

$$M_i(u, v) = \frac{1}{n_i} \left(\sum_{j=1}^{n_i} p_{i,j}(u, v) \right), \quad (10)$$

onde n_i é a quantidade de triângulos que compartilham o vértice v_i ;

7. calculam-se as coordenadas baricêntricas c_1, c_2 e c_3 relativas aos vértices v_1, v_2 e v_3 do triângulo T ;
8. o valor final é obtido pela seguinte fórmula:

$$V(u, v) = \sum_{i=1}^3 c_i M_i(u, v). \quad (11)$$

O procedimento acima é utilizado também para recuperação das derivadas. O procedimento para determinação do triângulo que contém o ponto (u, v) está descrito na seção de recuperação usando coordenadas baricêntricas. A determinação do *patch* que contém cada triângulo é facilmente executada usando-se estruturas de dados simples.

A formulação acima garante a transição suave entre os patches, pelo uso de coordenadas baricêntricas. A Fig. 10 exibe uma superfície triangular recuperada usando SPR.

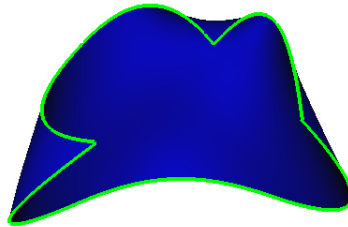


Figura 10: Superfície triangular recuperada por SPR.

6. EXEMPLOS

Esta seção ilustra o procedimento para interpolação de superfícies paramétricas triangulares, desde a discretização das curvas de contorno até as duas técnicas de recuperação, usando coordenadas baricêntricas e SPR. É importante ressaltar que todas as imagens foram obtidas do ambiente gráfico desenvolvido neste trabalho para modelagem e visualização de superfícies.

A Fig. 11 exibe três curvas de contorno e pontos representando a discretização a ser utilizada para a geração da malha de elementos triangulares.

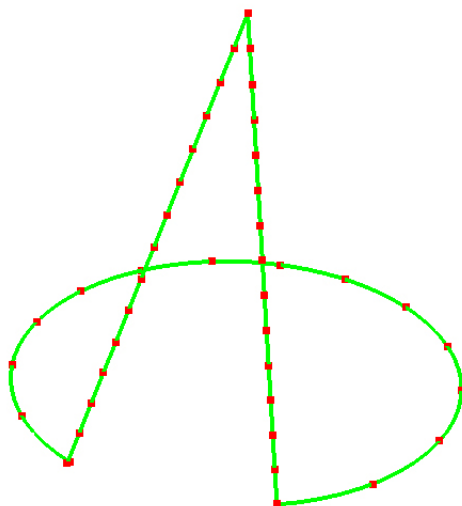


Figura 11: Discretização das curvas de contorno.

A malha gerada a partir da discretização acima pode ser vista na Fig. 12. Essa figura também exibe o espaço paramétrico gerado pela biblioteca PFS.

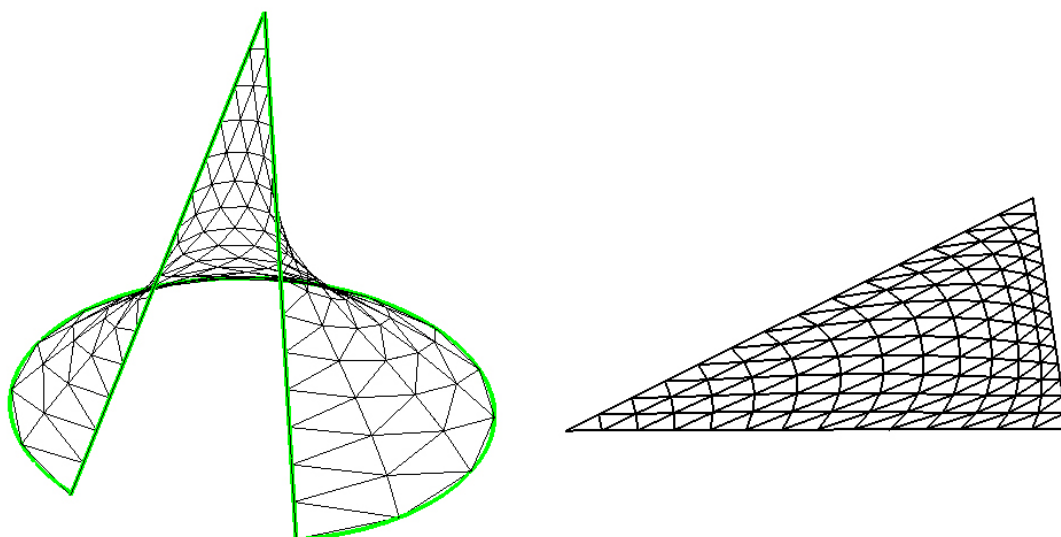


Figura 12: Geração da malha de elementos triangulares e espaço paramétrico obtido.

Finalmente, a Fig. 13 e a Fig. 14 exibem a superfície recuperada usando a recuperação por coordenadas baricêntricas e SPR respectivamente.

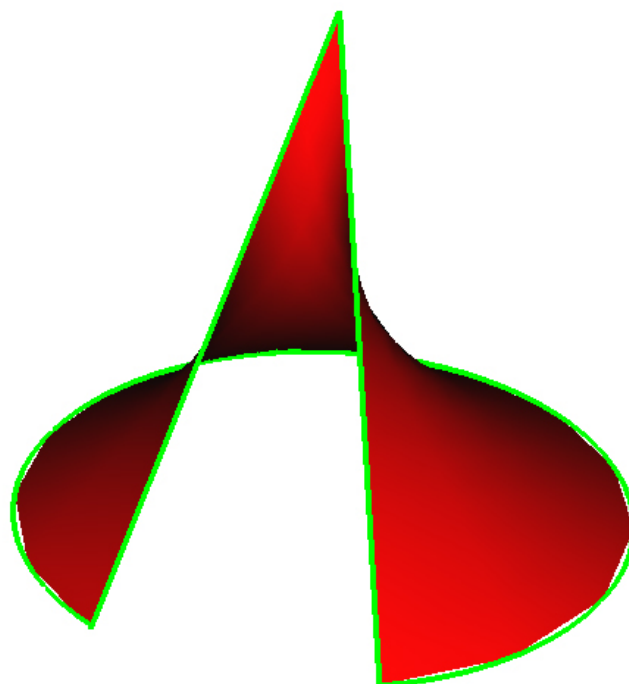


Figura 13: Superfície recuperada usando coordenadas baricêntricas.

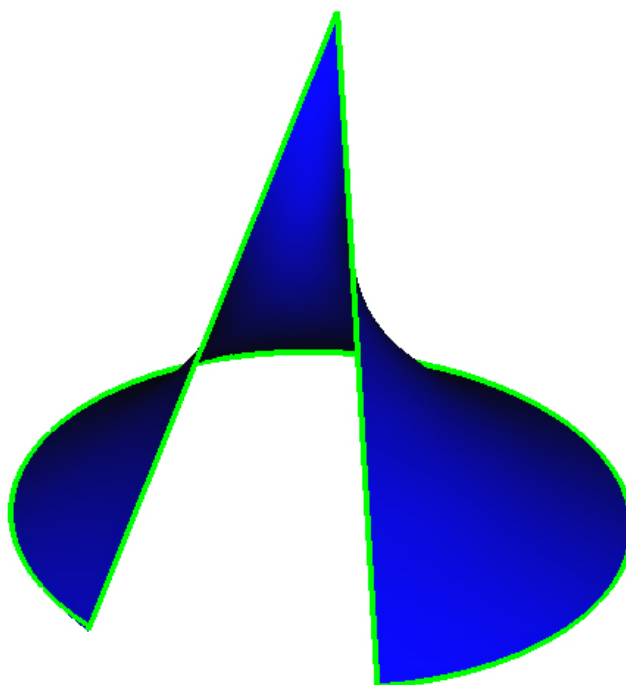


Figura 14: Superfície recuperada usando SPR.

7. CONCLUSÃO E TRABALHOS FUTUROS

Tendo em vista a dificuldade para a modelagem de superfícies triangulares, este trabalho apresentou uma alternativa para a modelagem de superfícies paramétricas triangulares a partir de três curvas de contorno. De modo geral, o procedimento apresentou bons resultados, com algumas observações a serem feitas.

A técnica de recuperação usando SPR apresentou bons resultados, mas o método de mínimos quadrados usado para geração do polinômio não garante que a superfície gerada irá respeitar todos os pontos da malha de elementos triangulares. Desse modo, o contorno da superfície pode apresentar pequenas diferenças em relação à discretização do contorno.

São sugestões para trabalhos futuros o desenvolvimento de um tratamento específico para garantir que as superfícies respeitem as curvas de contorno, um estudo comparativo entre as superfícies geradas a partir de coordenadas baricêntricas e SPR, e o desenvolvimento de métodos para geração de *patches* em triangulações arbitrárias.

Agradecimentos

O primeiro autor agradece ao PIBIC/CNPq e à Fundação de Amparo à Pesquisa do Estado de Alagoas (FAPEAL) pelo apoio durante a realização deste trabalho.

O quinto autor é bolsista recém-doutor e agradece ao apoio e suporte financeiro provido pela agência CNPq (processo número 305341/02-2).

REFERÊNCIAS

- Bathe, K. J., 1982. *Finite Element Procedures in Engineering Analysis*. Prentice Hall, New Jersey, 1 edition.
- Floater, M. S., 1997. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, vol. 14, n. 4, pp. 231–250.
- Gordon, W. & Hall, C. A., 1973. Construction of curvilinear co-ordinate systems and application to mesh generation. *International Journal for Numerical Methods in Engineering*, vol. 7, pp. 461–477.
- Miranda, A. C. O. & Martha, L. F., 2000. Uma biblioteca computacional para geração de malhas bidimensionais e tridimensionais de elementos finitos. In *Anais do XXI CILAMCE: 21st Iberian Latin-American Congress on Computational Methods in Engineering*.
- Miranda, A. C. O. & Martha, L. F., 2004. Geração de malhas estruturadas a partir de um espaço paramétrico de triangulações não estruturadas. In *Anais do XXV CILAMCE: 25nd Iberian Latin-American Congress on Computational Methods in Engineering*.
- Qin, H. & Terzopoulos, D., 1997. Triangular nurbs and their dynamic generalizations. *Computer Aided Geometric Design*, vol. 14, n. 4, pp. 325–347.
- Shreiner, D., Woo, M., Neider, J., & Davis, T., 2003. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley, Massachusetts, 4 edition.

- Stroustrup, B., 1991. *The C++ Programming Language*. Addison-Wesley, Massachusetts, 2 edition.
- Tutte, W. T., 1963. How to draw a graph. *Proc. London Math. Soc.*, vol. 3, n. 13, pp. 743–768.
- Zienkiewicz, O. & Zhu, J., 1994. The SPR recovery and boundaries. *International Journal for Numerical Methods in Engineering*, vol. 37, pp. 3195–3196.