

UM SISTEMA DE MODELAGEM GEOMÉTRICA USANDO SPLINES

Ludgero Kilunji Mateus da Silva
Prof. Orientador Robson Cavalcanti Lins

Departamento de Estatística e Informática
Universidade Católica de Pernambuco – CCT
Rua do Príncipe, 526, Boa Vista, Cep: 50050-900 Recife-PE
E-mail: getme@dei.unicap.br
rcl@dei.unicap.br

Resumo

As aproximações construídas usando *splines* podem preservar muitas propriedades matemáticas e geométricas das curvas e superfícies. Em computação gráfica, a incorporação dessas propriedades implica na fidelidade do objeto modelado. Na busca dessa fidelidade, muitas *splines* foram propostas. Este artigo apresenta uma comparação do efeito visual entre várias *splines* cúbicas baseada em uma formulação unificada.

Abstract

Approximations built using *splines* can preserve many mathematical and geometric properties of curves and surfaces. In Computer Graphics, the combination of the above properties warrants the accuracy of the model with respect to the object modeled. In the quest of this accuracy, many *splines* have been proposed. This paper presents a comparison of the visual effect among the various cubic *splines* based on a unified formulation.

Palavras-chave: *spline*, aproximação de funções, formulação unificada.

1 Introdução

Modelos geométricos são utilizados em uma grande variedade de aplicações de computação gráfica como sistemas de *design* em engenharia e arquitetura, sistemas de diagnósticos médicos e na representação de dados obtidos através de fenômenos químicos, físicos e geológicos. Muitas destas aplicações necessitam de uma representação bem definida destes modelos de forma a obter-se uma melhor aproximação dos dados ou objetos reais no computador. Na modelagem de objetos reais, como a fuselagem de um avião, pode-se usar basicamente as coordenadas dos pontos que compõem grande parte do objeto. No entanto, esta aproximação pode ser inviável em termos de capacidade de armazenamento em um computador de memória reduzida. Na prática, objetos são aproximados com pedaços de planos, curvas, superfícies ou outras formas geométricas que podem ser matematicamente descritas. *Splines*, que podem ser matematicamente descritas, foram propostas como objeto de estudo e análise devido à preservação de muitas propriedades matemáticas e geométricas que

determinam a fidelidade do objeto modelado. Deste modo, dada uma curva ou superfície cuja função f que a define é desconhecida, é possível construir, a partir de um número modesto de pontos da curva ou superfície, uma boa aproximação de f usando *splines*. Sendo assim, muitas *splines* foram propostas como: *B-splines*, *Beta-splines*, *Nu-splines*, *Gama-splines*, *WF-splines*, etc. Em [Lin96] foi desenvolvida uma formulação geral que permite construir todas ou a maioria das *splines* conhecidas. Os algoritmos e resultados da implementação da formulação unificada foram abordados em [dS99] para o caso da modelagem de curvas no plano.

2 Descrição das *Splines*

Matematicamente, uma curva ou superfície construída usando uma seqüência de pontos depende de métodos de aproximação ou interpolação. Para o caso das *splines* cúbicas, a curva ou superfície resultante será uma função polinomial por partes de grau ≤ 3 com a propriedade de possuírem continuidade em relação à posição, à tangente e a curvatura nos pontos de junção de seus segmentos.

Deste modo, pode-se dizer que uma função *spline* de grau $\leq n$ com uma seqüência de juntas U é qualquer combinação linear de funções base $B_{i,n}(u)$ de grau $\leq n$ para essa seqüência de juntas [Lin96].

A equação para a representação de uma curva *spline* é expressa por:

$$Q(u) = \sum_{i=0}^m d_i \cdot B_{i,n}(u) \quad , \quad d_i \in R \quad (2.1)$$

e

$$Q(u, w) = \sum_{i=0}^n \sum_{j=0}^m d_{i,j} \cdot B_{i,k}(u) \cdot B'_{j,l}(w) \quad , \quad d_{i,j} \in R \quad (2.2)$$

para superfícies *spline* [FV92], [RA90].

Importa salientar que a continuidade da curva é o fator fundamental para muitas aplicações de modelagem geométrica, pois garante a preservação da forma do objeto modelado. Portanto, uma função *spline* de grau $\leq n$, possui continuidade $n-1$ nas suas junções. Neste artigo será dada atenção especial as funções *spline* de grau ≤ 3 que possuem continuidades C^0 , C^1 e C^2 e são utilizadas em um grande número de aplicações. Em muitas aplicações de Desenho Auxiliado por Computador (CAD) em que o ajuste de curvas é utilizado, precisa-se modificar partes da curva sem influenciar o restante. Esta propriedade é chamada de controle local, pois as modificações se restringem a uma parte específica da curva. Devido à falta de controle local nas técnicas de aproximação dos mínimos quadrados e de interpolação de Lagrange, *splines* são muito utilizadas na modelagem geométrica, já que apresentam maior eficiência em termos de armazenamento e manipulação. As *splines* descritas em [FV90], [Lin96] e [dS99] foram analisadas e a implementação de seus algoritmos é apresentada neste artigo.

3 Abordagem Unificada

As diversas curvas e superfícies *spline* são adequadas para diferentes tipos de aplicações. No entanto, uma aplicação pode requerer o uso de mais de uma técnica de modelagem geométrica como, por exemplo, no design das chamadas *freeform shapes* em que o objeto a ser modelado não pode ser representado analiticamente em termos de curvas ou superfícies

cônicas. Deste modo recorre-se à abordagem unificada das curvas *spline* desenvolvida em [Lin96], pois permite que, a partir de um único procedimento, todas as curvas *spline* mencionadas anteriormente sejam representadas, desde que valores adequados sejam fornecidos aos parâmetros de cada curva *spline*. Em [Lin96], as bases *bspline* são consideradas bases gerais, pois se reduzem às outras bases *spline* através da escolha adequada dos parâmetros \mathbf{b}_1 e \mathbf{b}_2 .

Para gerar uma curva *spline* de grau ≤ 3 sobre uma seqüência de juntas uniforme são necessárias as seguintes substituições de parâmetros:

- *B-spline*: $\mathbf{b}_1 = 1$ e $\mathbf{b}_2 = 0$.
- *n-spline*: $\mathbf{b}_1 = 1$ e $\mathbf{b}_2 = \mathbf{n}$.
- *gspline*: $\mathbf{b}_1 = 1$ e $\mathbf{b}_2 = f(\mathbf{g})$.
- *WF-spline*: $\mathbf{b}_1 = f(A)$ e $\mathbf{b}_2 = g(A,B)$.

Para gerar uma curva *bspline* são somente necessários os parâmetros $\mathbf{b}_1 > 0$ e $\mathbf{b}_2 \neq 0$. A figura 1 mostra a relação entre várias *splines* conhecidas.

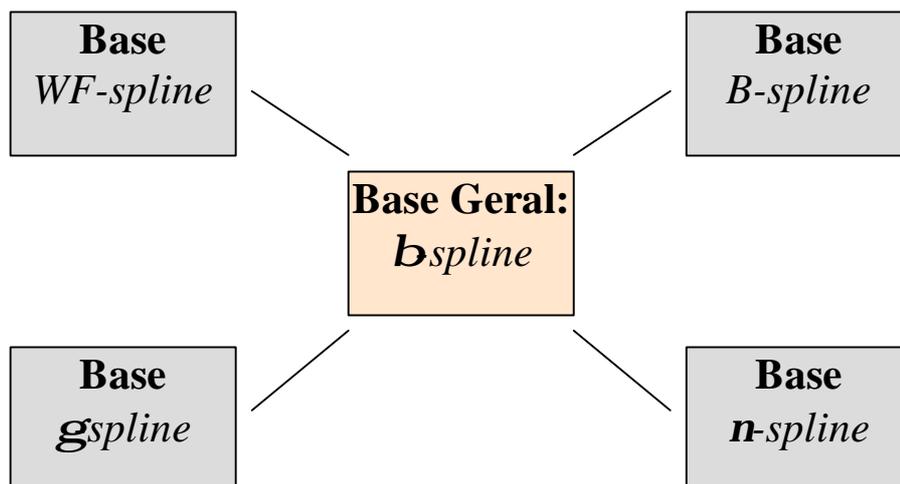


Figura 1: Relação entre as diversas bases *spline*.

4 Algoritmo da Curva *Spline*

A implementação dos algoritmos para as curvas e superfícies *spline* seguiu a seguinte metodologia:

- Aquisição dos dados para a implementação do algoritmo.
- Implementação da curva *B-spline*.
- Implementação do algoritmo da formulação unificada.
- Teste dos algoritmos implementados e análise dos resultados obtidos.

Os algoritmos para as curvas *spline* foram implementados em linguagem C. Na fase de aquisição dos dados, foram analisadas várias possibilidades de entrada dos dados, seja de forma interativa ou a partir de um arquivo. A última alternativa foi adotada devido à facilidade na alteração dos dados. Em seguida foi implementada a curva *B-spline* com base na formulação matricial [RA90]. Através da análise dos resultados desta implementação foi

possível a migração para a abordagem unificada das curvas *spline* cuja descrição do algoritmo é apresentada a seguir:

```

<Leia o conjunto de pontos de controle de um arquivo dado.>
<Escolha a curva spline.>
<Leia os parâmetros da curva escolhida.>
<Converta os parâmetros lidos para os parâmetros  $\mathbf{b}_1$  e  $\mathbf{b}_2$ .>
<Calcule a matriz dos coeficientes da base spline.>
Para(s = 1; s ≤ ( npts – grau_u); s = s + 1) faça    /* contagem dos segmentos */
{
    <Seleciona os pontos de controle no intervalo que define o segmento da curva spline.>

    Para(u_cont = 0.0; u_cont ≤ 1.0; u_cont = u_cont + (1.0/prec)) faça
    {
        <Calcule os valores do vetor de potências  $U_i$ .>
        <Calcule os valores das bases  $B_{i,n}(u)$ .>
        <Calcule as coordenadas do ponto da curva spline dentro do intervalo.>
        <Armazena as coordenadas do ponto em um arquivo.>
    }
}

```

em que npts: número de pontos de controle,
 grau_u: grau da curva *spline*,
 prec: número de pontos no intervalo parametrizado[0,1].

A conversão dos parâmetros da curva escolhida para os parâmetros \mathbf{b}_1 e \mathbf{b}_2 é feita através da formulação unificada mencionada anteriormente e o cálculo da matriz dos coeficientes da base *spline* foi descrito em [Lin96]. Vale adiantar que a matriz de coeficientes é inerente a cada tipo de *spline*, ou seja, teremos diferentes matrizes para diferentes curvas ou superfícies *spline* de acordo com valores adequados dos parâmetros \mathbf{b}_1 e \mathbf{b}_2 . O cálculo do vetor de potências depende do grau da curva. Para curvas de grau ≤ 3 deve-se calcular o vetor $[u^3 \ u^2 \ u \ 1]$ para cada $u \in [0,1]$. Por último, é necessário calcular as bases *spline* gerais descritas na abordagem unificada [Lin96] no intervalo parametrizado $[0,1]$ para cada u .

A seleção dos pontos de controle é feita de forma que os $m+1$ pontos lidos d_0, d_1, \dots, d_m originem $m-(grau-1)$ segmentos, ou seja, se $grau = 3$, serão criados $m-2$ segmentos de curva, como mostra a figura 2.

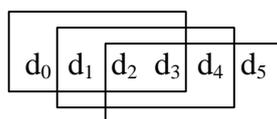


Figura 2: Seleção dos pontos de controle de uma curva *spline* de grau = 3.

5 Algoritmo da Superfície *Spline*

A implementação dos algoritmos para as superfícies *spline* serve como uma extensão aos algoritmos para curvas. Os algoritmos foram implementados em linguagem C e também em Java e seguem a mesma metodologia adotada para as curvas *spline*. A entrada de dados é feita através da leitura de um arquivo com os vértices de controle. Nesta fase foi implementado o algoritmo para as superfícies *B-spline* e analisados os seus resultados. Na fase seguinte foi implementado o algoritmo da abordagem unificada para superfícies *spline* que é descrito a seguir:

```

<Leia as malhas de pontos de controle a partir de um arquivo.>
<Escolha uma superfície spline.>
<Leia os parâmetros da superfície spline escolhida.>
<Converta os parâmetros lidos para os parâmetros  $\mathbf{b}_1$  e  $\mathbf{b}_2$  da base geral.>
<Calcule a matriz dos coeficientes e sua transposta>
Para(s = 1; s ≤ (npts – grau_u); s = s+1)
{
  Para(u_cont = 0.0; u_cont ≤ 1.0; u_cont = u_cont + (1.0/prec_1))
  {
    <Calcule os valores do vetor de potências  $U_i$  na direção  $u$ .>
    <Calcule os valores das bases  $B_{i,k}(u)$  na direção  $u$ .>
    Para(t = 1; t ≤ (mpts – grau_w); t = t+1)
    {
      <Selecione a malha de pontos de controle que define o segmento de curva.>
      Para(w_cont = 0.0; w_cont ≤ 1.0; w_cont = (1.0/prec_2))
      {
        <Calcule os valores do vetor de potências  $W_i$  na direção  $w$ .>
        <Calcule os valores das bases  $B'_{j,l}(w)$  na direção  $w$ .>
        <Calcule as coordenadas do ponto da superfície spline dentro do
          intervalo.>
        <Armazena as coordenadas do ponto em um arquivo.>
      }
    }
  }
}

```

A seleção da malha de pontos de controle é efetuada de acordo com a direção u ou w que é fixada. A figura 3 mostra como as malhas dos vértices são selecionadas para uma direção u .

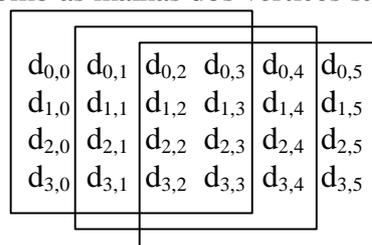


Figura 3: Seleção das malhas de pontos de controle.

6 Resultados

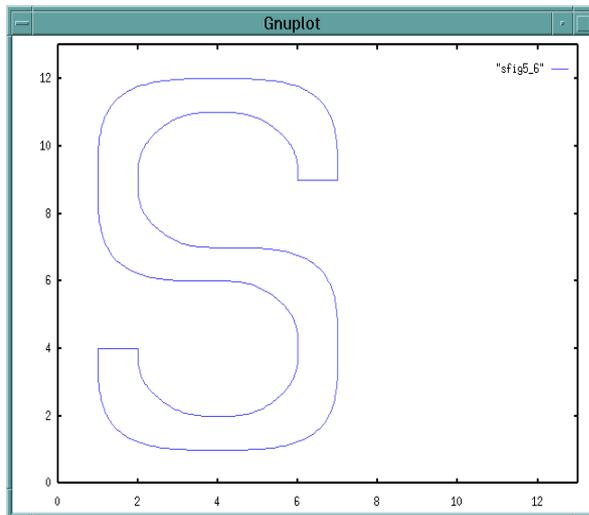


Figura 4: Curva n -spline com $n = 0$.

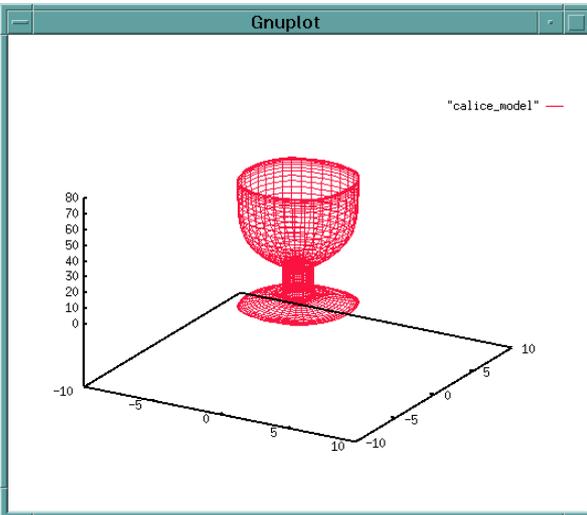


Figura 5: Superfície B -spline.

Os algoritmos da curva e da superfície *spline* com a abordagem unificada propostos em [Lin96] foram implementados e alguns dos resultados são mostrados nas figuras 4 e 5. Na figura 4, observa-se um tipo de letra que foi obtido através de um número reduzido de pontos de controle que definem a forma do objeto. Para este caso, foi utilizada uma n -spline (*Nu-spline*) com parâmetro $n = 0$. A figura 5 mostra o modelo de um cálice resultante da implementação do algoritmo para uma superfície B -spline. Para isso, foi criada uma malha a partir de um número reduzido de pontos de controle que foi usada como dado de entrada na implementação. Observa-se, nas duas figuras, a suavidade das curvas que definem o objeto, garantida pela continuidade nas suas junções. Em função desta característica, os objetos modelados podem ser usados em projetos e softwares que integram mais recursos de representação e transformações geométricas.

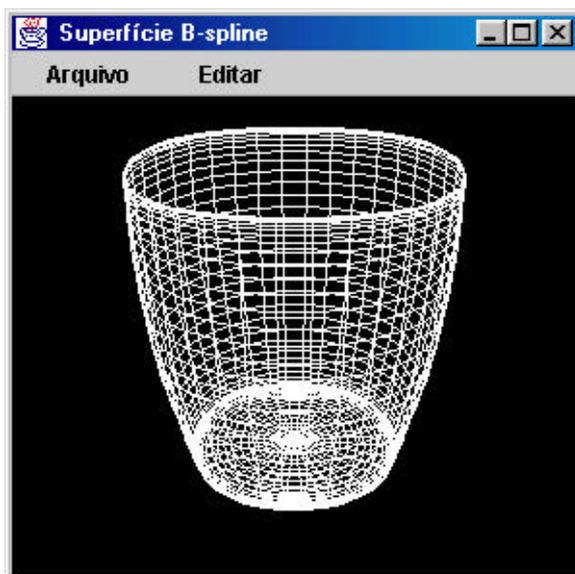


Figura 6: Superfície B -spline.

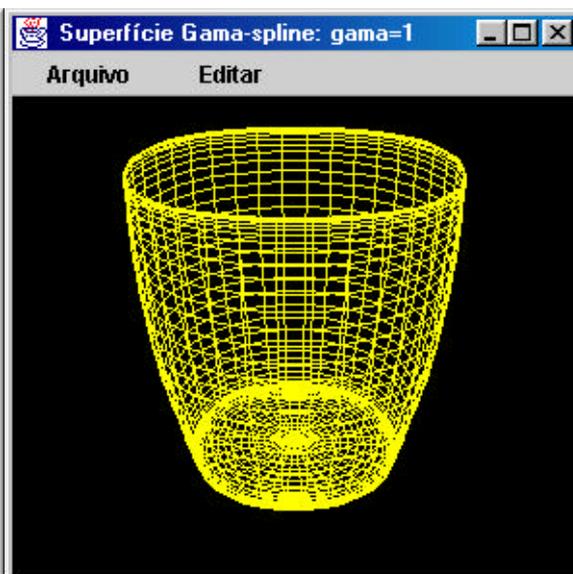


Figura 7: Superfície g spline com $g = 1$.

As figuras 6 e 7 mostram objetos modelados com a implementação da formulação unificada em Java, utilizando classes do pacote Java 3D. A figura 6 ilustra o modelo de um copo resultante da implementação do algoritmo para uma superfície *B-spline*. Na figura 7, podemos observar um modelo equivalente, utilizando uma *gspline* (*Gama-spline*) com o parâmetro $g=1$. A equivalência dos dois objetos deve-se ao fato de que os parâmetros das duas superfícies reduzem-se aos mesmos parâmetros para a base geral, ou seja, cada uma gera os parâmetros $b_1=1$ e $b_2=0$. Esta particularidade pode ser utilizada caso se pretenda representar uma curva ou superfície a partir de outra, desde que sejam passados para ela os parâmetros adequados.

7 Trabalhos Futuros

Para dar continuidade a este trabalho, pretende-se implantar um ambiente visual com a linguagem de programação Java de forma a acrescentar mais recursos como manipulação de objetos, interação com o usuário e aplicação de textura e iluminação para tornar os objetos mais realísticos.

8 Conclusões

Neste artigo, foi apresentado o algoritmo da formulação unificada para a maioria das *splines* e os resultados de sua implementação foram analisados. No entanto, as *WF-spline* não foram analisadas em virtude da não disponibilidade de material bibliográfico mais específico. De uma forma geral, os resultados obtidos são bastante satisfatórios e estão de acordo com a base teórica. Vale salientar que, visualmente, as *splines* podem ser: *diferentes* para parâmetros arbitrários e *equivalentes* para parâmetros adequados.

Referências

- [Ana93] Anand, V. B. **Computer Graphics and Geometric Modeling for Engineers**. Clemson University, Canada, 1993.
- [Bar88] Barsky, B. A. **Computer Graphics and Geometric Modeling Using Beta-splines**. Springer-Verlag Berlin Heidelberg, 1988.
- [dS99] Da Silva, L. K. **Um Sistema de Modelagem Geométrica Usando Splines**. Relatório Final de Iniciação Científica, PIBIC/FACEPE PE, 1999.
- [FV92] Foley, J. D.; Van Dam, A.; et al. **Computer Graphics: Principles And Practice**. Addison-Wesley, Massachusetts, 2nd Edition, 1992.
- [Lin96] Lins, R. C. **Splines e Modelagem Geométrica**. Dissertação de Mestrado, Campina Grande, PB, 1996.
- [RA90] Rogers, D. F. and Adams, V. A. **Mathematical Elements for Computer Graphics**. McGraw-Hill, New York, 2nd edition, 1990.