

Relatório sobre o algoritmo de União de Curvas Bézier Grau 3 e NURBS Grau 4

Fernanda Maria de Souza¹

¹Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina (UDESC)

1. Introdução

Este relatório apresenta o conjunto de técnicas e bibliotecas utilizadas para desenvolvimento da união de curvas Bézier Grau 3 e NURBS Grau 5 com continuidade C0, G0, G1 e G2 (não confirmado). Além disso, também é apresentado um manual de uso para novos usuários, desde instalação na sua própria máquina até compilação do algoritmo.

2. Instalação

Para instalação e configuração do ambiente, basta baixar o arquivo .zip e extrair o conteúdo. Após, será necessário instalar as bibliotecas utilizadas:

```
pip3 install numpy
```

```
pip3 install matplotlib
```

Com as bibliotecas instaladas, apenas execute o arquivo omog.py como:

```
python3 omog.py
```

3. Técnicas utilizadas

Para o desenvolvimento do algoritmo foi utilizada a linguagem Python3, com algumas bibliotecas extras como numpy e matplotlib. A primeira foi utilizada para manipulação de arrays e arranjo dos pontos, e a segunda para plotagem e manipulação das curvas. Além dessas, também foi utilizada a biblioteca math para operações matemáticas principalmente para o cálculo das continuidades.

3.1. C0

Para cálculo da continuidade C0, primeiramente é definido o último ponto da NURBS e o primeiro ponto da Bézier. Após é calculado o delta X e delta Y, onde a fórmula $x_2 - x_1$ e $y_2 - y_1$ é aplicada. Com essa operação, é incrementado o delta para cada ponto da array de pontos. A Figura 1 ilustra o algoritmo implementado para cálculo de C0.

3.2. G1

Com relação ao G1, foram instanciadas duas variáveis: o penúltimo ponto da NURBS e o primeiro ponto da Bézier. Após, foi calculado o delta X e delta Y com relação a esses dois pontos onde novamente a fórmula $x_2 - x_1$ e $y_2 - y_1$ foi aplicada. Com os deltas é verificado que o novo segundo ponto da bézier é o primeiro ponto dela em $X - \text{delta X}$, e igualmente o primeiro ponto dela em $Y - \text{delta Y}$. Abaixo, na Figura 2 é mostrado essa execução em Python3.

```

113     # C0
114     ultimoPontoNURBS = arrayDePontosNURBS[4]
115     primeiroPontoBezier = arrayDePontos[0]
116     deltaX = ultimoPontoNURBS[0] - primeiroPontoBezier[0]
117     deltaY = ultimoPontoNURBS[1] - primeiroPontoBezier[1]
118
119     for ponto in arrayDePontos:
120         ponto[0] = ponto[0] + deltaX
121         ponto[1] = ponto[1] + deltaY
122

```

Figura 1. Código para cálculo de C0.

```

123     # G1
124     penultimoNURBS = arrayDePontosNURBS[3]
125     primeiroBEZIER = arrayDePontos[0]
126
127     deltaXC1 = penultimoNURBS[0] - primeiroBEZIER[0]
128     deltaYC1 = penultimoNURBS[1] - primeiroBEZIER[1]
129
130     arrayDePontos[1][0] = primeiroBEZIER[0] - deltaXC1
131     arrayDePontos[1][1] = primeiroBEZIER[1] - deltaYC1
132

```

Figura 2. Código para cálculo de G1.

3.3. G2

Para o cálculo do G2, foram feitas as mesmas operações de G1, porém agora com relação ao antepenúltimo ponto da NURBS e o primeiro ponto da Bézier. Verificou-se o mesmo delta X e delta Y entre os pontos, para após aplicação e substituição no terceiro ponto da Bézier. Com essa solução que foi criada após a apresentação ao professor, acredito ter encontrado G2. A implementação está disponível na Figura 3 abaixo.

```

133     # G2
134     antiPenultimoNURBS = arrayDePontosNURBS[2]
135     primeiroBEZIER = arrayDePontos[0]
136
137     deltaXC1 = antiPenultimoNURBS[0] - primeiroBEZIER[0]
138     deltaYC1 = antiPenultimoNURBS[1] - primeiroBEZIER[1]
139
140     arrayDePontos[2][0] = primeiroBEZIER[0] - deltaXC1
141     arrayDePontos[2][1] = primeiroBEZIER[1] - deltaYC1
142

```

Figura 3. Código para cálculo de G2.

4. Manual de Uso

O sistema tem seu funcionamento de forma muito simples, bastando apenas a inserção de pontos para geração das curvas com continuidade $G2/G1$. Após executado o arquivo `omog.py` será gerado um gráfico em branco, siga os seguintes passos:

1. Selecione cinco pontos para formação da primeira curva (NURBS). Somente após selecionar os 5 pontos, a curva será gerada.
2. Depois, selecione quatro pontos para formação da segunda curva (Bézier). Somente após selecionar os 4 pontos, a curva será gerada.
3. Por fim, a curva é gerada com continuidade $C0$ e $G1/G2$.

5. Conclusões

Portanto, neste trabalho pode-se concluir como funciona a implementação das próprias curvas Bézier e NURBS, bem como entender como funciona a junção de duas curvas com diferentes continuidades. Apesar de não conseguir implementar $C1$ e $C2$, os aprendizados referentes a $C0$, $G1$ e $G2$ foram suficientes para entender e projetar a ideia de como seria uma implementação de $C1$ e $C2$ (que não foi aplicada nesse projeto).