UNIVERSIDADE FEDERAL DE RORAIMA

CENTRO DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

COMPUTAÇÃO GRÁFICA

ALGORITMOS DE BEZIER E CASTELJAU PARA RASTERIZAÇÃO DE CURVAS E SUTHERLAND PARA RECORTE

FABIO VITOR DE OLIVEIRA NORONHA

BOA VISTA, RR

DEZEMBRO DE 2017

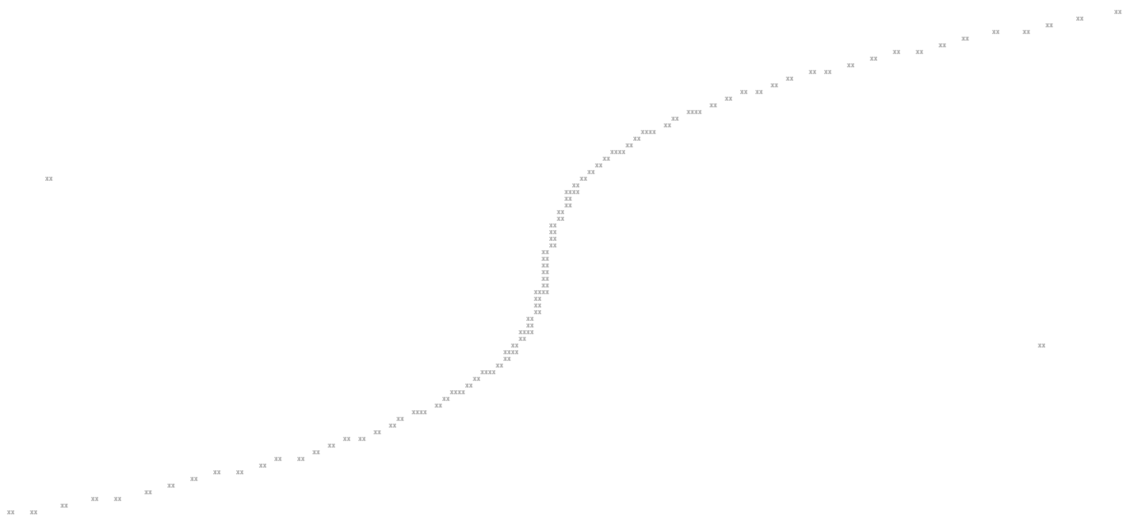
RASTERIZAÇÃO DE CURVAS

ALGORITMO DE BEZIER

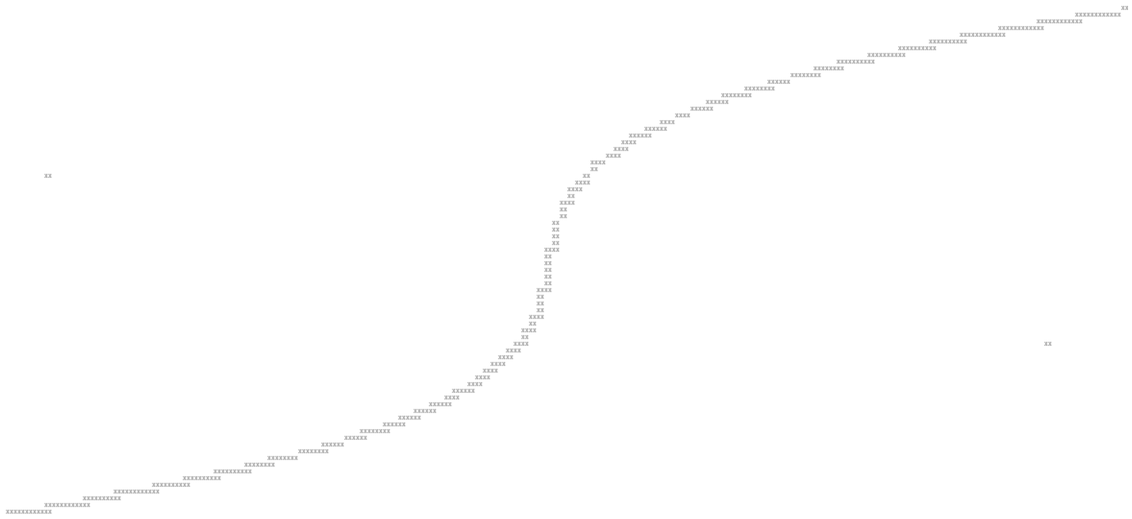
O algoritmo de Bezier foi implementado e testado, por ter base na equação paramétrica ele depende de uma constante que chamaremos PASSO para desenhar as curvas. O tamanho no PASSO influencia em como a curva será desenhada, em outras palavras, quanto menor o passo menor a chance de existirem falhas na curva.

Exemplo 1: Foi desenhada a curva composta pelos pontos [(5, 5), (140, 30), (10, 55), (150, 80)] e usados variáveis PASSO iguais a 0.1, 0.01 e 0.001 como mostram as imagens respectivamente, abaixo:

Curva de Bezier com PASSO = 0.1



Curva de Bezier com PASSO = 0.01



Curva de Bezier com PASSO = 0.001

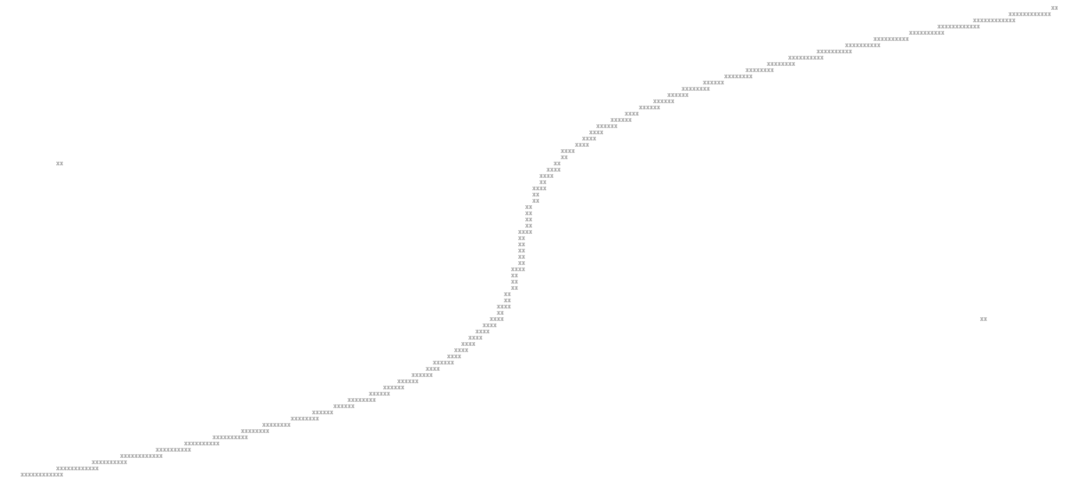
ALGORITMO DE CASTELJAU

O algoritmo de Casteljau obteve resultados similares visualmente em relação ao de Bezier, porém o mesmo possui uma singularidade: funciona somente para curvas definidas com 4 pontos.

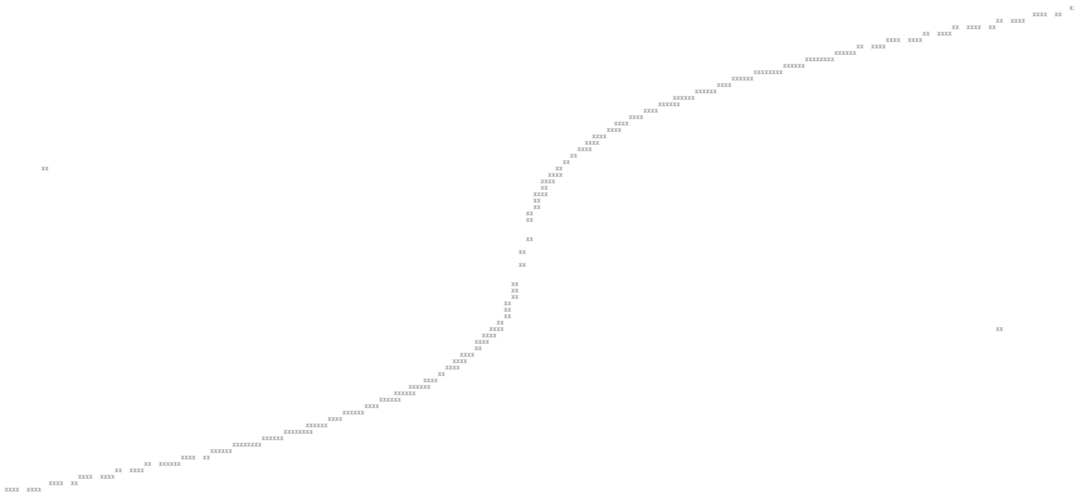
O algoritmo de Casteljau possui também uma variável que podemos chamar de PASSO que influencia na precisão da curva desenhada, similarmente ao de Bezier quanto menor o PASSO, mais bem desenhada a curva será.

Outra diferença entre esses algoritmos é o custo computacional, uma vez que Casteljau faz menos operações aritméticas e as que o mesmo faz incluem apenas soma e divisão por 2, operações estas que são relativamente leves para o computador.

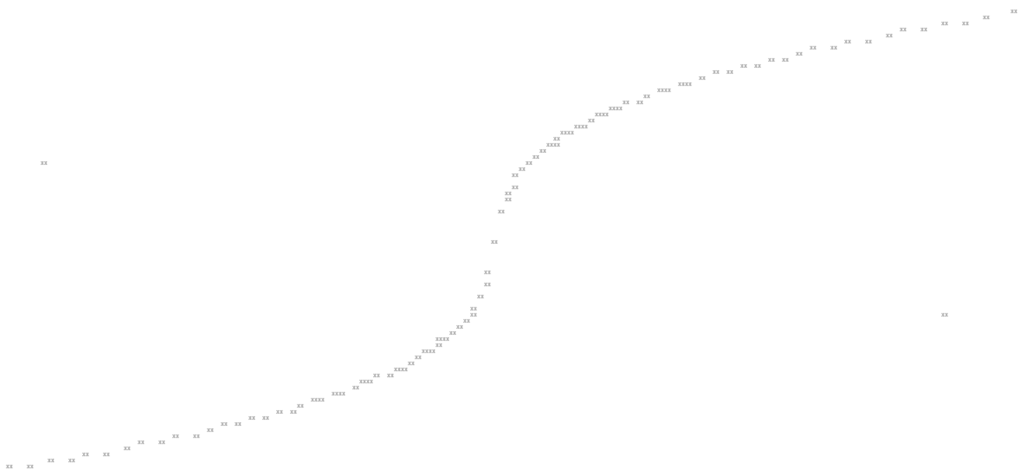
Exemplo 2: A mesma curva testada anteriormente com o algoritmo de Bezier foi testada com o de Casteljau - [(5, 5), (140, 30), (10, 55), (150, 80)] – com as variáveis PASSO iguais a 0.1, 0.5 e 1. As imagens abaixo mostram os resultados:



Curva de Bezier com PASSO = 0.1



Curva de Bezier com PASSO = 0.5



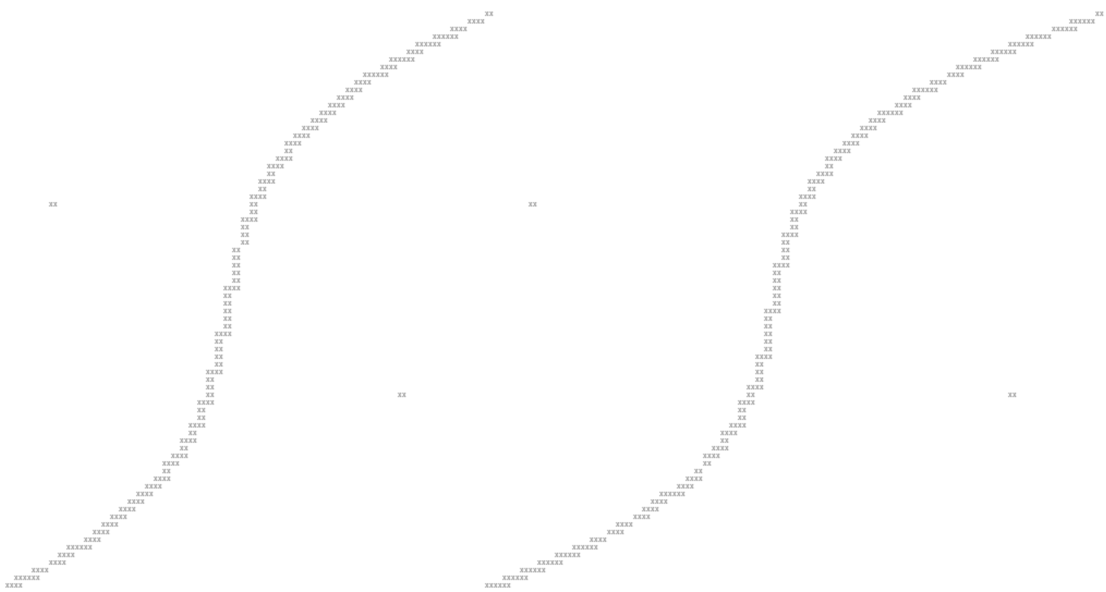
Curva de Bezier com PASSO = 1.0

BEZIER VS CASTELJAU

Comparando os dois algoritmos a principal diferença que notamos é que, enquanto Bezier funciona para curvas definidas com N pontos, Casteljau funciona para curvas definidas em 4 pontos, apenas. Sua justificativa é que essas curvas eram as mais usadas na época em que os algoritmos foram desenvolvidos.

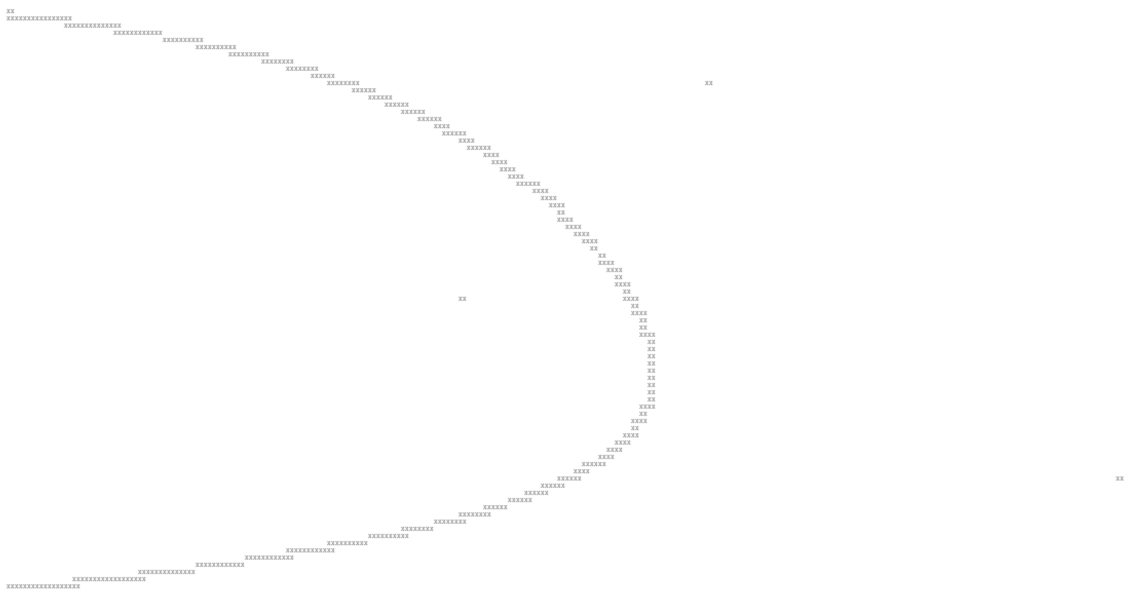
Outra diferença está no modo de cálculo que os dois algoritmos adotam, Bezier traz uma abordagem muito mais geral e complexa utilizando um somatório dos pesos de cada ponto em relação ao ponto que será definido na reta com auxílio do Polinómio de Bernstein que, por sua vez, utiliza o binômio de Newton. Já Casteljau utiliza apenas sucessivos cálculos de médias entre seguimentos de retas afim de encontrar os mesmos pontos. Isso torna Casteljau mais leve de ser processado pelo computador.

Exemplo 3: Foi definida a curva [(5, 5), (50, 30), (10, 55), (60, 80)] para os dois algoritmos e o resultado segue na imagem:



As curvas desenhadas utilizaram, da esquerda para direita, os algoritmos de Bezier e Casteljau respectivamente. Repare na similaridade visual.

Exemplo 4: Outra curva desenhada utilizando o algoritmo de Bezier, dessa vez definida com 5 pontos.

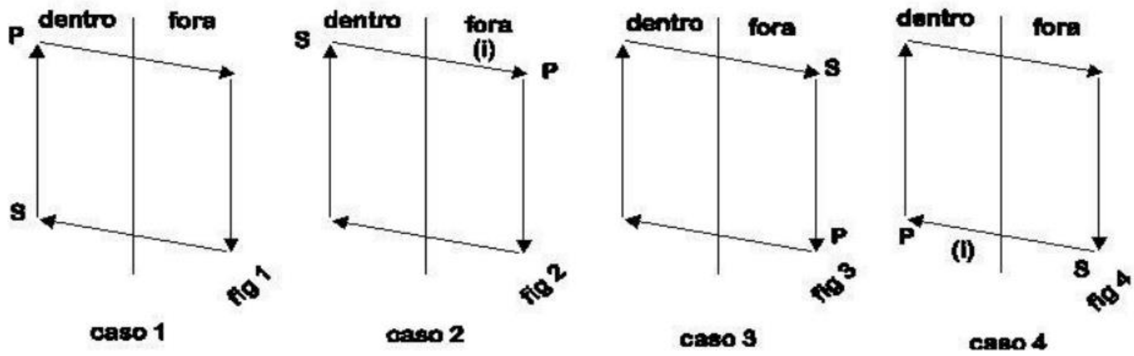


ALGORITMO DE SUTHERLAND PARA RECORTE

O algoritmo de Sutherland possui solução é bem simples e elegante para o problema de recorte. O mesmo foi formulado para cortar qualquer polígono com um retângulo definido, sua aplicação funciona assim:

Primeiramente aplica-se o algoritmo em 4 partes, uma vez para o lado direito, abaixo, esquerdo e acima. Para cada passo escolhe-se uma borda do retângulo com o qual estamos cortando o polígono respectivamente de acordo com o lado no qual estamos aplicando o algoritmo.

Em seguida analisa-se cada segmento de reta em relação à borda do retângulo escolhida e a classifica em um dos 4 possíveis casos, como mostra a figura abaixo:



A ação que deve ser tomada em cada caso é:

Caso 1: O vértice P é adicionado à lista de saídas.

Caso 2: Adiciona-se o vértice I que é o ponto de interseção entre o segmento de reta SP e a borda do retângulo que estamos analisando à lista de saídas.

Caso 3: Descarta-se os dois vértices S e P.

Caso 4: Adiciona-se o vértice I que é o ponto de interseção entre o segmento de reta SP e a borda do retângulo que estamos analisando e o vértice P à lista de saídas.

Por fim adiciona-se o primeiro vértice da lista de saídas ao fim da mesma afim de fechar o polígono. Terminado o processamento de um lado, o algoritmo é aplicado novamente no próximo lado com os novos vértices sendo a lista de saídas. A lista de saída da aplicação do último lado é o novo polígono recortado.

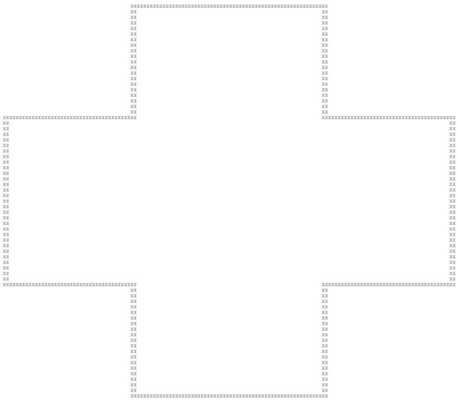
Foram testados 4 polígonos e o algoritmo de recorte proposto por Sutherland funcionou corretamente em todos os exemplos testados, são eles:

TRIÂNGULO RETÂNGULO

O triângulo retângulo foi cortado com sucesso como mostram as figuras abaixo, respectivamente, antes e depois do corte:

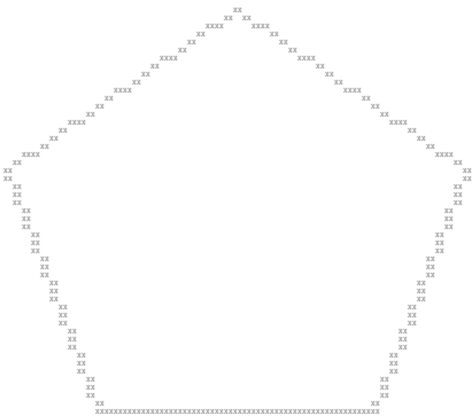
POLÍGONO NÃO CONVEXO I (EM FORMA DE CRUZ)

O mesmo foi cortado como esperado, como mostram as figuras abaixo contendo o antes e o depois do recorte respectivamente:



PENTÁGONO

O pentágono resultando do corte foi satisfatório como mostram as imagens abaixo:



POLÍGONO NÃO CONVEXO II

Este que seria o polígono mais trabalhoso de ser recortado, teoricamente, foi recortado perfeitamente. O resultado pode ser visto nas figuras abaixo, antes e depois do recorte: