5
Desenvolvimento Computacional

Os métodos propostos no capítulo anterior são extremamente simples de implementar. Nas tabelas abaixo temos os métodos para curvas no plano e curvas no espaço. Cada uma delas conterá também a forma com que foi calculada a estimativa das derivadas, o nome do método e as variações apresentados.

Curvas no Plano		
Tipo de estimativa	Nome do Método	Variação
Minimização Separada	Coordenadas Independentes	Rotação
	Coordenadas Dependentes	
Minimização Conjunta	Minimização Simultânea das Coordenads	Mudança na estimativa do parâmetro

Tabela 5.1: Tabela dos métodos para curvas no plano

Curvas no Espaço		
Tipo de estimativa	Nome do Método	Variação
Minimização Separada	Coordenadas Independentes	———
Minimização Conjunta	Minimização Simultânea das Coordenads	Mudança na estimativa do parâmetro

Tabela 5.2: Tabela dos métodos para curvas no espaço

Fazer uma mudança de parâmetro significa mudar a estimativa de s_i segundo a descrição do capítulo anterior.

As variações foram feitas na tentativa de obtermos uma melhor aproximação das curvas por parábolas ou cúbicas.

5.1
Curvas no Plano

Os algoritmos propostos para curvas planares seguem diretamente da análise da Seção 4.1: calculamos os coeficientes a, b, c, e, f, g e h e resolvemos o método MQP. Fornecemos no Algorítmo 1 um pseudocódigo do método MQP.

Algoritmo 1

$\Delta l[] = \mathrm{a} = \mathrm{b} = \mathrm{c} = \mathrm{e} = \mathrm{f} = \mathrm{g} = \mathrm{h} = 0$;
for $\mathrm{i} = 1 \ldots 2\mathrm{q}$ **do**
  $\Delta l[\mathrm{i}] \leftarrow \Delta l[\mathrm{i\text{-}1}] + ||p_{j-q+i-1}p_{j-q+i}||$;
end for
$m = \Delta l[\mathrm{j}]$;
for $\mathrm{i} = 0 \ldots 2\mathrm{q}$ **do**
  $\Delta l[\mathrm{i}] \leftarrow \Delta l[\mathrm{i}]$ - m; {Centrando Δl em j}
  $w = \mathrm{weight}(\ \Delta l[\mathrm{i}]\ )^2$;
  $a \leftarrow a + w\ \ (\Delta l[\mathrm{i}])^2$;
  $b \leftarrow b + w\ \ (\Delta l[\mathrm{i}])^3$;
  $c \leftarrow c + w\ \ (\Delta l[\mathrm{i}])^4$;
  $e \leftarrow e + w\ \ (\Delta l[\mathrm{i}])\ \ (x_{j+i}\text{-}x_j)$;
  $f \leftarrow f + w\ \ (\Delta l[\mathrm{i}])\ \ (y_{j+i}\text{-}y_j)$;
  $g \leftarrow g + w\ \ (\Delta l[\mathrm{i}])^2\ \ (x_{j+i}\text{-}x_j)$;
  $h \leftarrow h + w\ \ (\Delta l[\mathrm{i}])^2\ \ (y_{j+i}\text{-}y_j)$;
end for
$d = ac - b^2$; {determinante}

5.1.1
Método das Coordenadas Independentes

Este método estima x'_j, y'_j, x''_j e y''_j. Para tal estimativa, encontramos as soluções resolvendo dois MQP para x e y independentemente. O vetor tangente encontrado, $\mathbf{T} = (x'_j, y'_j)$ não necessáriamente tem norma unitária e consequentemente o vetor (x''_j, y''_j) não necessariamente é perpendicular ao vetor $\mathbf{T}$. O vetor normal ($\mathbf{N}$) é dado por $\dfrac{sinal(\kappa)(x''_j, y''_j)}{\sqrt{(x'')^2 + (y'')^2}}$.

Apresentamos um pseudocódigo desse método no Algoritmo 2.

Algoritmo 2

call Set Weighted Least Squares Variables (j);
$\mathbf{T}_x \leftarrow (ce - bf)/d$;
$\mathbf{T}_y \leftarrow (cg - bh)/d$;
$\mathbf{N}_x \leftarrow (af - be)/d$;
$\mathbf{N}_y \leftarrow (ah - bg)/d$;
$\kappa \leftarrow (eh - fg)/d$;
$\mathbf{N} \leftarrow \mathrm{sinal}(\kappa)\ \ (\mathbf{N}/||\mathbf{N}||)$;

5.1.2
Método das Coordenadas Dependentes

Da mesma forma que o método anterior, este método também estima x'_j, y'_j, x''_j e y''_j. Nesse caso fazemos a suposição que a curva está parametrizada pelo comprimento de arco, introduzindo as condições

$$\begin{cases} x'^2 + y'^2 = 1 \\ x'x'' + y'y'' = 0 \end{cases},$$

que são facilmente observadas no capítulo 2.

Deste modo, com estas equações podemos usar o valor estimado de x'_j e x''_j para estimar o valor de y'_j e y''_j, ou vice-versa. Para tal escolha verificamos o valor absoluto das funções coordenadas do vetor tangente, ou seja, $|\mathbf{T}_x| < |\mathbf{T}_y|$.

Um pseudocódigo desse algorítimo é apresentado no Algoritmo 3. Note que solucionamos o MQP para uma coordenada e deduzimos a estimativa da outra coordenada a partir dessa.

Algoritmo 3

1: **call** Set Weighted Least Squares Variables (j);
2: $\mathbf{T}_x \leftarrow (ce - bf)/d$;
3: $\mathbf{T}_y \leftarrow (cg - bh)/d$;
4: **if** $|\mathbf{T}_x| < |\mathbf{T}_y|$ **then** {Considerando x(y)}
5: $\quad \mathbf{T}_y \leftarrow \text{sinal}(\mathbf{T}_y)\sqrt{(1 - \mathbf{T}_x^2)}$;
6: $\quad \mathbf{N}_x \leftarrow (af - be)/d$;
7: $\quad \mathbf{N}_y \leftarrow$ -$(\mathbf{T}_x\mathbf{N}_x)/\mathbf{T}_y$;
8: **else** {Considerando y(x)}
9: $\quad \mathbf{T}_x \leftarrow \text{sinal}(\mathbf{T}_x)\sqrt{(1 - \mathbf{T}_y^2)}$;
10: $\quad \mathbf{N}_y \leftarrow (ah - bg)/d$;
11: $\quad \mathbf{N}_x \leftarrow$ -$(\mathbf{T}_y\mathbf{N}_y)/\mathbf{T}_x$;
12: **end if**
13: $\kappa \leftarrow \mathbf{T}_x\mathbf{N}_y$ - $\mathbf{T}_y\mathbf{N}_x$;
14: $\mathbf{N}_x \leftarrow$ -$\mathbf{T}_y$; $\mathbf{N}_y \leftarrow \mathbf{T}_x$;

Neste algoritmo temos garantidas as propriedades geométricas do vetor tangente e do vetor normal: $\mathbf{T}$ é unitário e $\mathbf{N}$ é ortogonal a $\mathbf{T}$. O uso deste algoritmo satisfaz bem quando a curva é quase o gráfico de uma função no eixo $y = f(x)$. O melhor eixo é escolhido na linha 4 do Algorítmo 10. De qualquer maneira um rotação ajuda a aproximá-la no caso.

5.1.3 Método com Rotação

A aproximação por mínimos quadrados funciona bem onde pontos são bem distribuídos. De qualquer forma, mesmo nos casos simples como da circunferência os pontos podem ser quase alinhados verticalmente. Para evitar esta situação, como em [8], escolhemos um dos eixos x ou y como referência para parametrização (veja seção 5.1.2), mesmo que, como em [8], a parábola degenere para um reta quando a tangente está em 45^o.

No método das coordenadas dependentes, a precisão numérica está diminuindo com o ângulo. Para compensar este erro numérico, podemos estimar primeiro a tangente com um dos métodos apresentados, e então usá-la para melhor distribuir a amostra. Esta operação é feita de forma simples por uma rotação nos pontos antes da soma (antes da linha 12 no Algorítmo 1), como podemos ver abaixo

$$\begin{bmatrix} x_i - x_j \\ y_i - y_j \end{bmatrix} \mapsto \begin{bmatrix} \mathbf{T}_x & \mathbf{T}_y \\ -\mathbf{T}_y & \mathbf{T}_x \end{bmatrix} \begin{bmatrix} x_i - x_j \\ y_i - y_j \end{bmatrix} \quad .$$

5.1.4
Método da Minimização Simultânea das Coordenadas (MMSC)

Da mesma forma que os métodos anteriores este método também estima x'_j, y'_j, x''_j e y''_j e utiliza as mesmas condições do método anterior. Estamos supondo que os pontos amostrados vem de uma curva que está parametrizada pelo comprimento de arco, ou seja,

$$\begin{cases} {x'}^2 + {y'}^2 = 1 \\ x'x'' + y'y'' = 0 \end{cases} \quad .$$

Diferentemente dos métodos anteriores este método não faz uma estimatida das derivadas de forma independente; pelo contrário, este método estima as derivadas das funções coordenadas ao mesmo tempo tentando satisfazer as condições propostas.

Para cada ponto da amostra calculamos o valor de θ correspondente, já que este será utilizado no cálculo da curvatura, do vetor tangente e do vetor normal no respectivo ponto. Para estimar o θ, inicialmente necessitamos do sinal de M e N para sabermos em qual quadrante ele se encontra. Após sabermos o quadrante utilizaremos o método da bissecção para acharmos uma valor aproximado de θ. Um pseudocódigo do método da bissecção é apresentado no Algoritmo 4. A função Φ mencionada no método é a mesma função dada em (4-16). Esta função deve ser construída fora do algoritmo e chamada quando necessário.

Algoritmo 4 Bissecção

$inicial$, $final$
if $\Phi(inicial) < 10^{-20}$ **then**
  $\theta \leftarrow inicial;$
else
  $\theta \leftarrow (final + inicial)/2;$
  while $|\Phi(\theta)| > 10^{-20}$
    if $\Phi(\theta)\Phi(inicial) < 0$ **then**
      $inicial \leftarrow inicial;$
      $final \leftarrow \theta;$
    else
      $inicial \leftarrow \theta;$
      $final \leftarrow final;$
    end if
    $\theta \leftarrow (final + inicial)/2;$
    if $|inicial - final| < 10^{-12}$ **then**
      $\theta \leftarrow inicial;$
    end if
  end while
end if

Um pseudocódigo do MMSC é apresentado no Algoritmo 5.

Algoritmo 5

call Set Weighted Least Squares Variables (j);
$M \leftarrow fh + eg$ e $N \leftarrow fg - eh$;
if $h \geq 0$ e $g \geq 0$ **then**
  $\gamma \leftarrow \arctan(g/h)$;
end if
if $h < 0$ e $g > 0$ **then**
  $\gamma \leftarrow \arctan(g/h) - \pi$;
end if
if $h \leq 0$ e $g > 0$ **then**
  $\gamma \leftarrow \arctan(g/h) + \pi$;
end if
if $h > 0$ e $g < 0$ **then**
  $\gamma \leftarrow \arctan(g/h)$;
end if
if $h = 0$ e $g \neq 0$ **then**
  $\gamma \leftarrow \pi/2$;
end if
if $M \geq 0$ e $N \geq 0$ **then**
  $inicial \leftarrow \pi/2 - \gamma$ e $final \leftarrow \pi - \gamma$;
  call Bissecção;
  $\kappa \leftarrow (1/c)(h\cos(\theta) - g\,\mathrm{sen}(\theta))$;
  $\mathbf{T}_x \leftarrow \cos(\theta)$, $\mathbf{T}_y \leftarrow \mathrm{sen}(\theta)$, $\mathbf{N}_x \leftarrow -\sin(\theta)$, $\mathrm{N}_y \leftarrow \cos(\theta)$;
end if
if $M < 0$ e $N > 0$ **then**
  $inicial \leftarrow -\pi - \gamma$ e $final \leftarrow -\pi/2 - \gamma$
  call Bissecção;
  $\kappa \leftarrow (1/c)(h\cos(\theta) - g\,\mathrm{sen}(\theta))$;
  $\mathbf{T}_x \leftarrow \cos(\theta)$, $\mathbf{T}_y \leftarrow \mathrm{sen}(\theta)$, $\mathbf{N}_x \leftarrow -\sin(\theta)$, $\mathrm{N}_y \leftarrow \cos(\theta)$;
end if
if $M > 0$ e $N < 0$ **then**
  $inicial \leftarrow -\gamma$; e $final \leftarrow \pi/2 - \gamma$
  call Bissecção;
  $\kappa \leftarrow (1/c)(h\cos(\theta) - g\,\mathrm{sen}(\theta))$;
  $\mathbf{T}_x \leftarrow \cos(\theta)$, $\mathbf{T}_y \leftarrow \mathrm{sen}(\theta)$, $\mathbf{N}_x \leftarrow -\sin(\theta)$, $\mathrm{N}_y \leftarrow \cos(\theta)$;
end if
if $M \leq 0$ e $N \leq 0$ **then**
  $inicial \leftarrow -\pi/2 - \gamma$ e $final \leftarrow -\gamma$;
  call Bissecção;
  $\kappa \leftarrow (1/c)(h\cos(\theta) - g\,\mathrm{sen}(\theta))$;
  $\mathbf{T}_x \leftarrow \cos(\theta)$, $\mathbf{T}_y \leftarrow \mathrm{sen}(\theta)$, $\mathbf{N}_x \leftarrow -\sin(\theta)$, $\mathrm{N}_y \leftarrow \cos(\theta)$;
end if

5.1.5
MMSC com Mudança no Parâmetro

Da mesma forma que o outro método estimamos inicialmente a tangente com o MMSC e usamos essa estimativa para melhor distribuir a amostra.

Após a estimativa do vetor tangente, fazemos uma mudança na estimativa do parâmetro. Enquanto no outro usávamos a distância entre $\mathbf{p}_k$ e $\mathbf{p}_{k+1}$, ou seja, Δl_k, nesse usaremos apenas $|x_k - x_{k+1}|$, com $x_k = x_k \cos(\theta) + y_k \mathrm{sen}(\theta)$. Essa mudança faz sentido pois ao rodarmos a tangente teremos $\mathbf{T} = (1, 0)$ e $\mathbf{N} = (0, 1)$, assim temos que $x(s) = s$ e $y(s) = \kappa \frac{s^2}{2}$.

Esse procedimente deve ser feito no Algorítmo 1 antes da estimativa das valores de a, b, c, e,f,g e h.

5.2
Curvas no Espaço

Os algoritmos propostos para curvas no espaço seguem diretamente da análise da Seção 4.2, ou seja, calculamos os coeficientes a_1, a_2, a_3, a_4, a_5, a_6, e_1, e_2, e_3, f_1, f_2, f_3, g_1, g_2, g_3 e resolvemos o método MQP. Um pseudo código é fornecido no Algoritmo 6 para o método do MQP.

Algoritmo 6

$\Delta l[] = a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = e_1 = e_2 = e_3 = f_1 = f_2 = f_3 = g_1 = g_2 = g_3 = 0;$
for i = 1 ... 2q **do**
  Δl[i] $\leftarrow \Delta l$[i-1] + $||p_{j-q+i-1}p_{j-q+i}||$;
end for
$m = \Delta l$[j] ;
for i = 0 ... 2q **do**
  Δl[i] $\leftarrow \Delta l$[i] - m;{Centrando Δl em j}
  w = weight(Δl[i])^2;
  $a_1 \leftarrow a_1 + w\ \ (\Delta l[i])^2;$
  $a_2 \leftarrow a_2 + w\ \ (\Delta l[i])^3;$
  $a_3 \leftarrow a_3 + w\ \ (\Delta l[i])^4;$
  $a_4 \leftarrow a_4 + w\ \ (\Delta l[i])^4;$
  $a_5 \leftarrow a_5 + w\ \ (\Delta l[i])^5;$
  $a_6 \leftarrow a_6 + w\ \ (\Delta l[i])^6;$
  $e_1 \leftarrow e_1 + w\ \ (\Delta l[i])\ \ (x_{j+i}\text{-}x_j);$
  $f_1 \leftarrow f_1 + w\ \ (\Delta l[i])\ \ (y_{j+i}\text{-}y_j);$
  $g_1 \leftarrow g_1 + w\ \ (\Delta l[i])\ \ (z_{j+i}\text{-}z_j);$
  $e_2 \leftarrow e_2 + w\ \ (\Delta l[i])^2\ \ (x_{j+i}\text{-}x_j);$
  $f_2 \leftarrow f_2 + w\ \ (\Delta l[i])^2\ \ (y_{j+i}\text{-}y_j);$
  $g_2 \leftarrow g_2 + w\ \ (\Delta l[i])^2\ \ (z_{j+i}\text{-}z_j);$
  $e_3 \leftarrow e_3 + w\ \ (\Delta l[i])^3\ \ (x_{j+i}\text{-}x_j);$
  $f_3 \leftarrow f_3 + w\ \ (\Delta l[i])^3\ \ (y_{j+i}\text{-}y_j);$
  $g_3 \leftarrow g_3 + w\ \ (\Delta l[i])^3\ \ (z_{j+i}\text{-}z_j);$
end for
$d = a_1a_4a_6 - a_1a_5a_5 - a_2a_2a_6 + 2a_2a_3a_5 - a_3a_3a_4;$

5.2.1 Coordenadas Independentes

Este método estima x'_j, x''_j, x'''_j, y'_j, y''_j, y'''_j, z'_j, z''_j e z'''_j. Para tal estimativa encontramos a solução resolvendo três MQP para cada uma das funções coordenadas x, y, z independentemente. Dessa forma o vetor tangente encontrado $\mathbf{T} = (x'_j, y'_j, z'_j)$ não necessáriamente tem norma unitária e consequantemente o vetor (x''_j, y''_j, z''_j) não necessáriamente é perpendicular ao vetor $\mathbf{T}$. A normalização do vetor normal estimado é feita de acordo com $sinal(\kappa)$.

O vetor binormal é obtido a partir do produto vetorial de $\mathbf{T}$ com $\mathbf{N}$ e a normalização de $\mathbf{B}$ é feita de acordo com o $sinal(\tau)$.

Apresentamos um pseudocódigo desse método no Algoritmo 7.

Algoritmo 7

call Set Weighted Least Squares Variables (j);
$\mathbf{T}_x \leftarrow (e_1a_4a_6 - e_1a_5a_5 - e_2a_2a_6 + e_2a_3a_5 + e_3a_2a_5 - e_3a_3a_4)/d$;
$\mathbf{T}_y \leftarrow (f_1a_4a_6 - f_1a_5a_5 - f_2a_2a_6 + f_2a_3a_5 + f_3a_2a_5 - f_3a_3a_4)/d$;
$\mathbf{T}_z \leftarrow (g_1a_4a_6 - g_1a_5a_5 - g_2a_2a_6 + g_2a_3a_5 + g_3a_2a_5 - g_3a_3a_4)/d$;
$\mathbf{N}_x \leftarrow -(e_1a_2a_6 - e_1a_3a_5 - e_2a_1a_6 + e_2a_3a_3 + e_3a_1a_5 - e_3a_2a_3)/d$;
$\mathbf{N}_y \leftarrow -(e_1a_2a_6 - e_1a_3a_5 - e_2a_1a_6 + e_2a_3a_3 + e_3a_1a_5 - e_3a_2a_3)/d$;
$\mathbf{N}_z \leftarrow -(g_1a_2a_6 - g_1a_3a_5 - g_2a_1a_6 + g_2a_3a_3 + g_3a_1a_5 - g_3a_2a_3)/d$;
$\mathbf{B}_x \leftarrow -(e_1a_2a_5 - e_1a_3a_4 - e_2a_1a_5 + e_2a_2a_3 + e_3a_1a_4 - e_3a_2a_2)/d$;
$\mathbf{B}_y \leftarrow -(f_1a_2a_5 - f_1a_3a_4 - f_2a_1a_5 + f_2a_2a_3 + f_3a_1a_4 - f_3a_2a_2)/d$;
$\mathbf{B}_z \leftarrow -(g_1a_2a_5 - g_1a_3a_4 - g_2a_1a_5 + g_2a_2a_3 + g_3a_1a_4 - g_3a_2a_2)/d$;
$a \leftarrow (\mathbf{T}_y\mathbf{N}_z - \mathbf{T}_z\mathbf{N}_y)^2 + (\mathbf{T}_z\mathbf{N}_x - \mathbf{T}_x\mathbf{N}_z)^2 + (\mathbf{T}_x\mathbf{N}_y - \mathbf{T}_y\mathbf{N}_x)^2$;
$nrm_{\mathbf{T}} \leftarrow \sqrt{\mathbf{T}_x^2 + \mathbf{T}_y^2 + \mathbf{T}_z^2}$;
$\kappa \leftarrow \sqrt{a}/nrm_{\mathbf{T}}^3$;
$\tau \leftarrow -((\mathbf{T}_y\mathbf{N}_z - \mathbf{T}_z\mathbf{N}_y)\mathbf{B}_x + (\mathbf{T}_z\mathbf{N}_x - \mathbf{T}_x\mathbf{N}_z)\mathbf{B}_y + (\mathbf{T}_x\mathbf{N}_y - \mathbf{T}_y\mathbf{N}_x)\mathbf{B}_z)/a$;
$\mathbf{T} \leftarrow \quad (\mathbf{T}/nrm_{\mathbf{T}})$;
$\mathbf{N} \leftarrow \mathrm{sinal}(\kappa) \quad (\mathbf{N}/||\mathbf{N}||)$;
$\mathbf{B} \leftarrow \mathrm{sinal}(\tau) \quad (\mathbf{T}_y\mathbf{N}_z - \mathbf{T}_z\mathbf{N}_y, \mathbf{T}_z\mathbf{N}_x - \mathbf{T}_x\mathbf{N}_z, \mathbf{T}_x\mathbf{N}_y - \mathbf{T}_y\mathbf{N}_x)/||\mathbf{B}||$;

5.2.2
Método da Minimização Simultânea das Coordenadas (MMSC)

Da mesma forma que o método anterior este também estima x'_j, x''_j, y'_j, y''_j, z'_j e z''_j. Contudo neste método não estimaremos a terceira derivada das funções coordenadas.

Diferentemente do método anterior este método faz uma estimativa das derivadas das funções coordenadas ao mesmo tempo, ou seja, para cada ponto obtemos os vetores v_1, v_2, v_3 e calculamos o valor de θ (para curvas no espaço θ é o ângulo entre v_1 e $\mathbf{T}$ na base $[v_1, v_2, v_3]$), já que este será usado na estimativa da curvatura, do vetor tangente e do vetor normal.

Na base $[v_1, v_2, v_3]$ estimamos os valores de M e N para sabermos em qual quadrante θ se encontra. Após sabermos o quadrante, utilizaremos novamente o método da bissecção (vide Algorítmo 4) para acharmos um valor aproximado para θ. A função Φ mencionada no método é a mesma função dada em (4-35). Esta função deve ser construida fora do algoritimo e chamada quando necessário.

Um pseudocódigo do MMSC é apresentado no Algoritmo 8.

Algoritmo 8

call Set Weighted Least Squares Variables (j);
$M \leftarrow \upsilon_1 \cdot E$ e $N = \upsilon_2 \cdot E$;
$\gamma \leftarrow \pi/2$;
if $M \geq 0$ e $N \geq 0$ **then**
    $inicial \leftarrow \pi/2 - \gamma$ e $final \leftarrow \pi - \gamma$;
    call Bissecção;
    $\kappa \leftarrow ||G||\mathrm{sen}(\theta)/a_4$;
    $\mathbf{T} \leftarrow \cos(\theta)\upsilon_1 + \mathrm{sen}(\theta)\upsilon_2$;
    $\mathbf{N} \leftarrow \mathrm{sen}(\theta)\upsilon_1 - \cos(\theta)\upsilon_2$;
    $\mathbf{B} \leftarrow \mathbf{T} \times \mathbf{N}$;
end if
if $M < 0$ e $N > 0$ **then**
    $inicial \leftarrow -\pi - \gamma$ e $final \leftarrow -\pi/2 - \gamma$
    call Bissecção;
    $\kappa \leftarrow ||G||\mathrm{sen}(\theta)/a_4$;
    $\mathbf{T} \leftarrow \cos(\theta)\upsilon_1 + \mathrm{sen}(\theta)\upsilon_2$;
    $\mathbf{N} \leftarrow \mathrm{sen}(\theta)\upsilon_1 - \cos(\theta)\upsilon_2$;
    $\mathbf{B} \leftarrow \mathbf{T} \times \mathbf{N}$;
end if
if $M > 0$ e $N < 0$ **then**
    $inicial \leftarrow -\gamma$; e $final \leftarrow \pi/2 - \gamma$
    call Bissecção;
    $\kappa \leftarrow ||G||\mathrm{sen}(\theta)/a_4$;
    $\mathbf{T} \leftarrow \cos(\theta)\upsilon_1 + \mathrm{sen}(\theta)\upsilon_2$;
    $\mathbf{N} \leftarrow \mathrm{sen}(\theta)\upsilon_1 - \cos(\theta)\upsilon_2$;
    $\mathbf{B} \leftarrow \mathbf{T} \times \mathbf{N}$;
end if
if $M \leq 0$ e $N \leq 0$ **then**
    $inicial \leftarrow -\pi/2 - \gamma$ e $final \leftarrow -\gamma$;
    call Bissecção;
    $\kappa \leftarrow ||G||\mathrm{sen}(\theta)/a_4$;
    $\mathbf{T} \leftarrow \cos(\theta)\upsilon_1 + \mathrm{sen}(\theta)\upsilon_2$;
    $\mathbf{N} \leftarrow \mathrm{sen}(\theta)\upsilon_1 - \cos(\theta)\upsilon_2$;
    $\mathbf{B} \leftarrow \mathbf{T} \times \mathbf{N}$;
end if

5.2.3
MMSC com Mudança de Parâmetro

Da mesma forma que o método para curvas no plano, estimamos inicialmente a tangente com o MMSC e usamos essa estimativa para melhor distribuir a amostra.

Este método também faz uma mudança na estimativa do parâmetro,uma vez obtido o vetor tangente pelo MMSC. Ou seja, substituímos o valor de $\Delta l_k = ||\mathbf{p}_k - \mathbf{p}_{k+1}||$ por $\Delta l_k = |x_k - x_{k+1}|$, com $x_k = x_k\mathbf{T}_x + y_k\mathbf{T}_y + z_k\mathbf{T}_z$. Essa mudança faz sentido uma vez que teremos $\mathbf{T}$ e $\mathbf{N}$ na base $[\upsilon_1, \upsilon_2, \upsilon_3]$ respectivamente igual a $(1,0,0)$ e $(0,1,0)$, assim vem de (4-25) que $x(s) = s$, $y(s) = \kappa\frac{s^2}{2}$ e $z(s) = 0$.

Esse procedimente deve ser feito no Algoritmo 6 antes da estimativa das valores de a_1, a_2, a_3, a_4, a_5, a_6, e_1, e_2, e_3, f_1, f_2, f_3, g_1, g_2, g_3.

5.3
Condições de Fronteira

Introduzimos nossos algoritmos para calcular a curvatura e torção em um ponto $\mathbf{p}_j$ com $2q+1$ amostras centrada em $\mathbf{p}_j$. Desta forma para pontos próximos da fronteira da curva, esta condição não é satisfeita. Neste caso podemos reduzir o tamanho de q na janela móvel, ou simplesmente calcular a curvatura e a torção usando uma janela não centrada. Nesse último caso, para curvas no plano, não temos as garantias teóricas da seção 4.1. No entanto, nossos experimentos permanecem coerentes, porém menos precisos.