

Sample Book Title

Sample book subtitle

FIRST-NAME LAST-NAME¹

February 20, 2020

¹homepage: www.example.com

Contents

1	Apendice	1
1.1	Análise de partículas ou cúmulos	2
1.1.1	Detetor de cúmulos	2
1.1.2	Estatística da matriz G	6
1.2	least-squares fitting of cubic splines	8
1.2.1	Calculando o vetor de parâmetros p	8
1.2.2	Modelando a solução	9

List of Figures

1.1	Analisando partículas na matriz \mathbf{M} .	2
1.2	Diagrama de bloco do sistema.	2
1.3	Diagrama de fluxo del algoritmo de criação de cúmulos.	3
1.4	Descrição da função $funA$.	3
1.5	Descrição da função $funA$.	4
1.6	Descrição da função $funB$	5
1.7	Descrição da função $funB1$	5
1.8	Descrição da função $funB2$	6
1.9	Descrição da função $funB3$	7
1.10	Diagrama de bloco para obter estatística de \mathbf{G} .	7
1.11	Encaixe de $M = 6$ splines cúbicos num conjunto de $N = 500$ pontos $\{x_n, y_n\}$.	8

List of Tables

Chapter 1

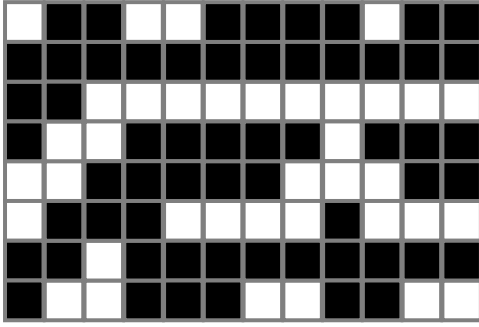
Apendice

Some methods utils in the work.

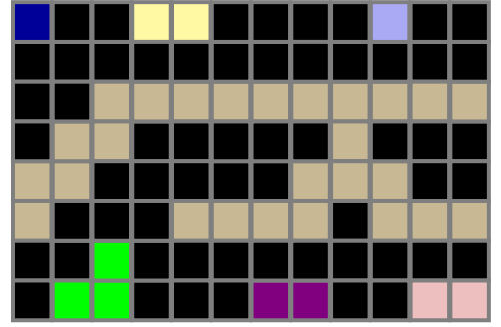
1.1 Análise de partículas ou cúmulos

Dada uma matriz binária $\mathbf{M} \in \mathbb{B}^{N_l \times N_c}$, que representa uma imagem em branco e preto, como é mostrado na Eq. (1.1) e na Figura 1.1a, onde os pixels de cor branca tem valor 1 e as de cor preta valor 0. Podemos representar \mathbf{M} como a agrupação de um conjunto de vetores coluna $\mathbf{M} = [\mathbf{m}_1 \ \mathbf{m}_2 \ \dots \ \mathbf{m}_c \ \dots \ \mathbf{m}_{N_c}]$, por outro lado podemos representar a um elemento da matriz \mathbf{M} como $\mathbf{M}(l, c)$, onde l e c representam a posição da linha e coluna respetivamente.

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (1.1)$$



(a) Matriz binária \mathbf{M} representada como uma imagem em branco e preto.



(b) Matriz \mathbf{G} com valores inteiros, representada como uma imagem com partículas (cúmulos) formados por grupos de pixels brancos.

Figure 1.1: Analisando partículas na matriz \mathbf{M} .

A Figura 1.2 mostra um diagrama de bloco de um detetor de cúmulos, o bloco recebe como entrada uma matriz binária \mathbf{M} e retorna uma matriz inteira $\mathbf{G} \in \mathbb{N}^{N_l \times N_c}$, criada pela agrupação dos pixels brancos em cúmulos, como na Figura 1.1b; onde todos os pixels num cúmulo são mapeados com o mesmo índice¹ id . O identificador $id = 0$ corresponde aos pixels pretos, é dizer os bits 0, qualquer outro id tomará um valor maior a zero.



Figure 1.2: Diagrama de bloco do sistema.

1.1.1 Detetor de cúmulos

Nosso problema principal é obter, a partir de uma matriz \mathbf{M} , uma matriz \mathbf{G} onde cada elemento contem o índice do cúmulo ao qual pertence, como mostra a Figura 1.1. Assim, pra cumprir este objetivo são usadas as funções $funA$ e $funB$, como mostra a Figura 1.3, onde são analisadas de maneira sequencial as colunas \mathbf{m}_c da matriz \mathbf{M} .

¹Na figura cada diferente id está representado com uma diferente cor.

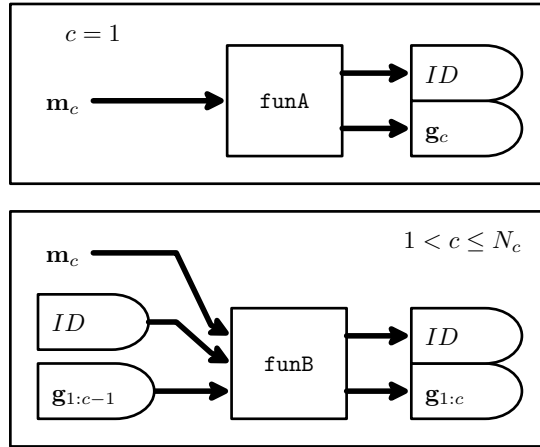
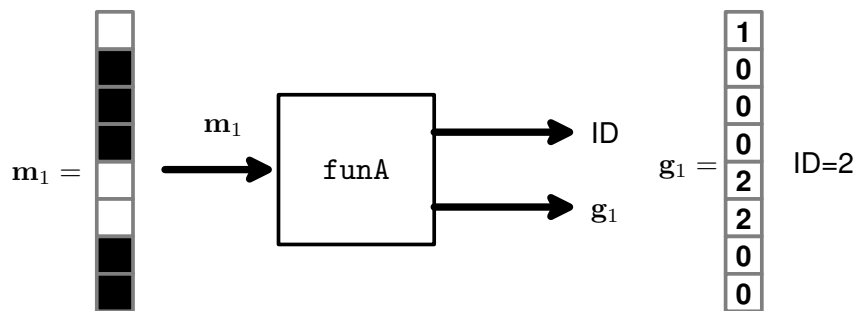


Figure 1.3: Diagrama de fluxo del algoritmo de criação de cúmulos.

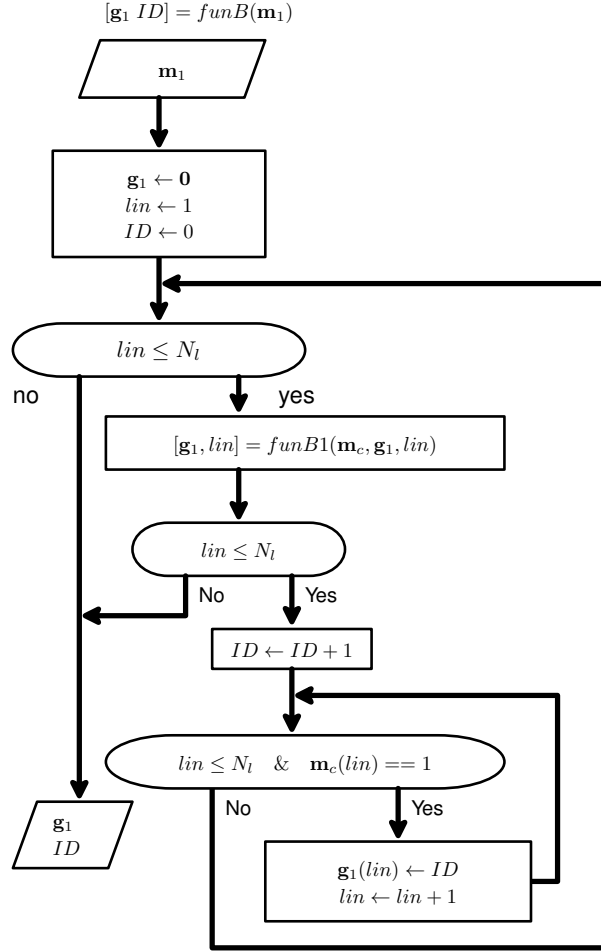
- A função *funA* recebe como parâmetro de entrada a coluna $\mathbf{m}_1 \in \mathbf{M}$. Retorna o valor dos índices na coluna $\mathbf{g}_1 \in \mathbf{G}$; adicionalmente a função retorna, *ID*, o valor do maior identificador *id* atribuído aos cúmulos nas operações.
- A função *funB* recebe como parâmetros de entrada a *c*-ésima coluna $\mathbf{m}_c \in \mathbf{M}$, o mapa de índices na submatriz $\mathbf{g}_{1:(c-1)} \in \mathbf{G}$ que abrange informações de índice desde a primeira coluna ate a $(c - 1)$ -ésima e *ID* o valor do maior identificador *id* atribuído aos cúmulos nas operações. A função retorna a matriz $\mathbf{g}_{1:c}$ com os dados dos índices desde a primeira coluna ate a *c*-ésima, adicionalmente a função retorna o novo valor de *ID*.

Algoritmo da função *funA*

A função *funA* recebe um vetor coluna \mathbf{m}_1 com uns e zeros, como é exemplificado na Figura 1.4, na saída a função retorna um vetor \mathbf{g}_1 com o mesmo tamanho porem com elementos com valores inteiros; este vetor representa o mapeamento dos índices *id* em cada cumulo de \mathbf{m}_1 .

Figure 1.4: Descrição da função *funA*.

Para calcular \mathbf{g}_1 se usa o vetor \mathbf{m}_1 ; elementos com zero em \mathbf{m}_1 provocam automaticamente elementos com um índice 0 em \mathbf{g}_1 ; por outro lado, se achamos uma região de 1's em \mathbf{m}_1 , estes recebem em \mathbf{g}_1 um mesmo índice, sendo que cada região de 1's tem um índice diferente. A função *funA* também retorna, *ID*, o máximo índice estabelecido em \mathbf{g}_1 . Um algoritmo que explica este procedimento, pode ser visto no diagrama de fluxo da Figura 1.5; o algoritmo usa a função *funB1* copia zeros do vetor \mathbf{m}_1 ao vetor \mathbf{g}_1 ; mais detalhes da função são explicados na Seção 1.1.1.

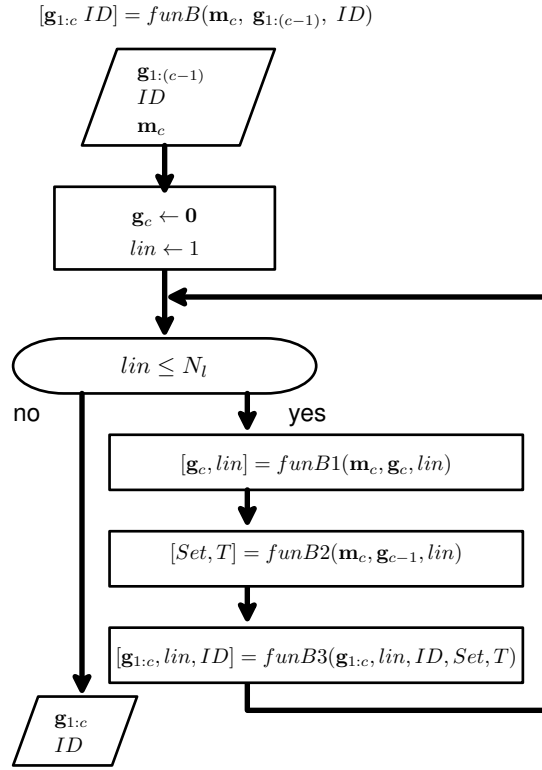
Figure 1.5: Descrição da função *funA*.

Algoritmo da função *funB*

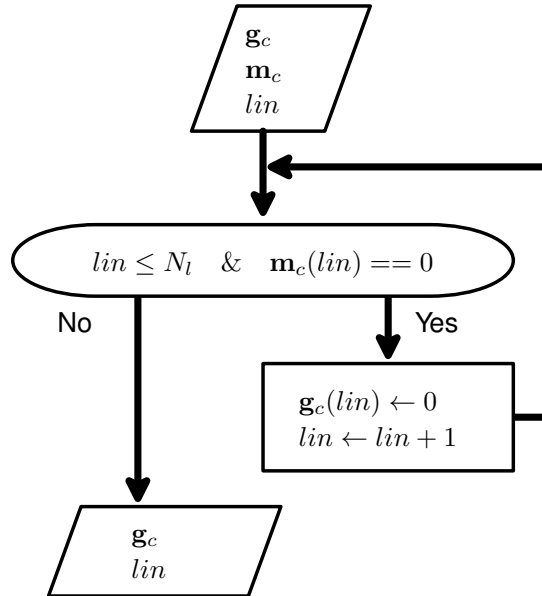
A função *funB* recebe como parâmetros de entrada um vetor \mathbf{m}_c , correspondente a c -ésima coluna da matriz \mathbf{M} , a submatriz $\mathbf{g}_{1:(c-1)} \in \mathbf{G}$ com o mapa de índices desde a primeira coluna até a $(c-1)$ -ésima coluna da matriz \mathbf{G} , e o máximo índice ID nos elementos da matriz $\mathbf{g}_{1:(c-1)}$. Na saída a função retorna uma matriz $\mathbf{g}_{1:c} \in \mathbf{G}$ com o mapa de grupos desde a primeira coluna até a c -ésima coluna da matriz \mathbf{G} , e o máximo índice ID achado na matriz $\mathbf{g}_{1:c}$. A Figura 1.6 mostra o diagrama fluxo do algoritmo da função *funB*. A função *funB* analisa os elementos do vetor \mathbf{m}_c , desde o primeiro elemento até o elemento N_l ; isto é realizado mediante as funções auxiliares *funB1*, *funB2* e *funB3*, de modo que estas funções modificam o conteúdo da matriz $\mathbf{g}_{1:c}$ e a variável ID .

Algoritmo da função *funB1*: A função recebe como parâmetros de entrada os vetores \mathbf{m}_c , \mathbf{g}_c e uma variável com a linha lin onde iniciara o análises. Se existem zeros no vetor \mathbf{m}_c , então se escrevem zeros nos elementos na mesma posição no vetor \mathbf{g}_c . A função finaliza quando se acha o primeiro 1 no vetor \mathbf{m}_c , carregando esta posição na variável lin , como mostra o diagrama de fluxo da Figura 1.7.

Algoritmo da função *funB2*: A função recebe como parâmetros de entrada os vetores \mathbf{m}_c , \mathbf{g}_{c-1} e uma variável com a linha lin que indica a posição do primeiro 1 achado com a função *funB1*. A função *funB2* analisa somente um cumulo de 1's no vetor \mathbf{m}_c , retornando um conjunto, *Set*, com os *id*'s no \mathbf{g}_{c-1} que estejam ao lado dos uns analisados no cúmulo em \mathbf{m}_c ; adicionalmente a função *funB2* retorna, T , o número de 1's do cúmulo. O algoritmo da função *funB2* pode ser visto no diagrama de fluxo da Figura 1.8; nele podemos perceber o uso

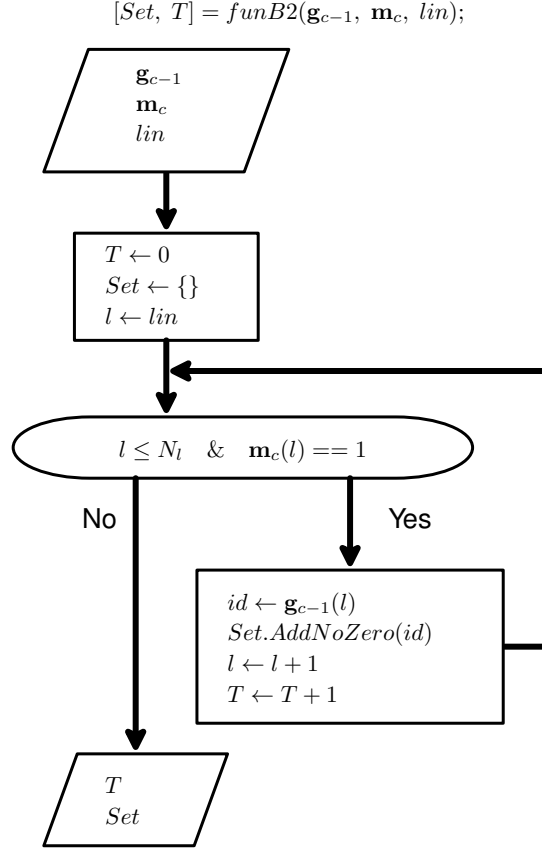
Figure 1.6: Descrição da função $funB$

$$[g_c, lin] = funB1(g_c, m_c, lin);$$

Figure 1.7: Descrição da função $funB1$

da função $Set.AddNoZero(id)$, esta função agrega o índice id ao conjunto Set , se este ainda não existe em Set e se $id \neq 0$.

Algoritmo da função $funB3$: A função recebe como parâmetros de entrada a submatriz $g_{1:c} \in \mathbf{G}$, uma variável com a linha lin que indica a posição do primeiro 1 achado com a função $funB1$, o índice ID que contem o valor do máximo id nos elementos da submatriz $g_{1:(c-1)} \in \mathbf{G}$, um conjunto Set com todos os índice vizinhos ao cúmulo analisado com a função $funB2$, e

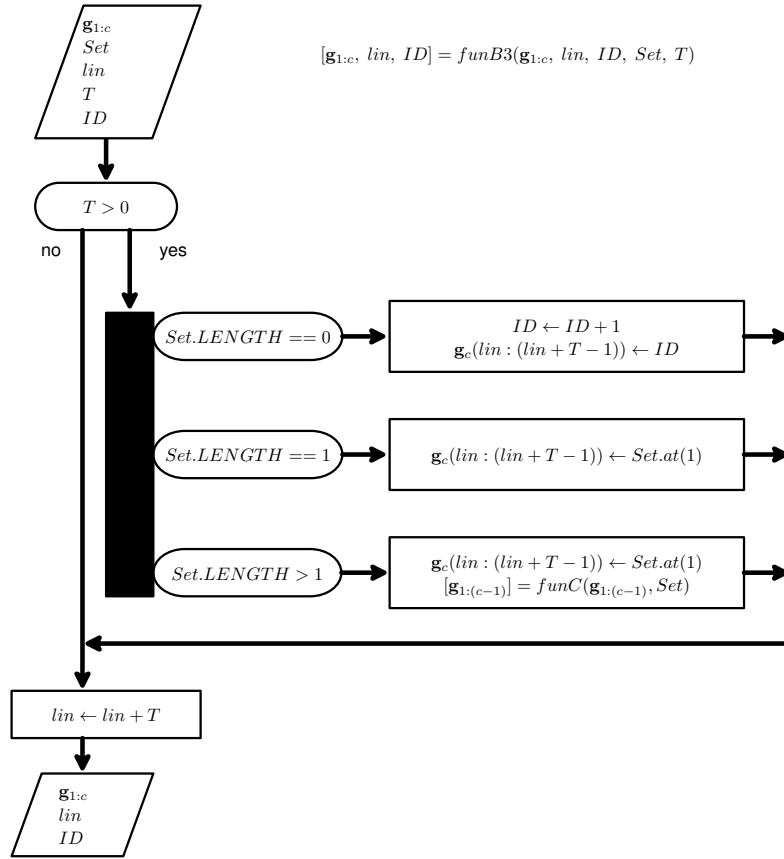
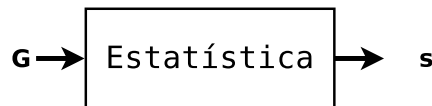
Figure 1.8: Descrição da função *funB2*

finalmente, T , o número de 1's no cúmulo. A função *funB3* se encarrega de escrever no vetor \mathbf{g}_c o índice correspondente, e fusionar índices na submatriz $\mathbf{g}_{1:(c-1)}$ se for necessário; assim, a função retorna a submatriz $\mathbf{g}_{1:c}$, com os índices modificados até a c -ésima coluna e lin -ésima linha, também retorna o atual valor de lin que indica o lugar do primeiro 0 achado, finalmente a função retorna o índice, ID , com maior valor achado na submatriz $\mathbf{g}_{1:c}$. O algoritmo da função *funB3* pode ser visto no diagrama de fluxo da Figura 1.9; nele podemos perceber o uso da função *funC*, esta função troca todos os índices id em $\mathbf{g}_{1:c-1}$, que existam no conjunto Set , pelo valor do primeiro índice no conjunto Set , de modo que todos os elementos achados em $\mathbf{g}_{1:c-1}$ tenham o índice $Set.at(1)$.

1.1.2 Estatística da matriz \mathbf{G}

A informação em \mathbf{G} pode ser processada para retornar um vetor \mathbf{s} com L elementos, um por cada cúmulo, onde $\mathbf{s}(l)$ representa ao l -ésimo cúmulo analisado, ver Figura 1.10. Assim \mathbf{s} é uma estrutura que contém dados e estatística de cada cúmulo como:

- $\mathbf{s}(l).Id$: O índice atribuído ao l -ésimo cúmulo.
- $\mathbf{s}(l).Area$: A área em pixels do l -ésimo cúmulo.

Figure 1.9: Descrição da função $funB3$ Figure 1.10: Diagrama de bloco para obter estatística de \mathbf{G} .

1.2 least-squares fitting of cubic splines

Aqui é mostrado como encaixar a curva $f(x) : \mathbb{R} \rightarrow \mathbb{R}$, sobre um conjunto de N pontos (x_n, y_n) , $\forall n \in \mathbb{Z}$, $0 \leq n < N$; mediante o uso de mínimos quadrados, tendo cada ponto uma importância de w_n . Onde a curva $y = f(x)$ esta composta de um grupo de M splines cúbicos (polinômios de grau 3), como o exemplo da Figura 1.11.

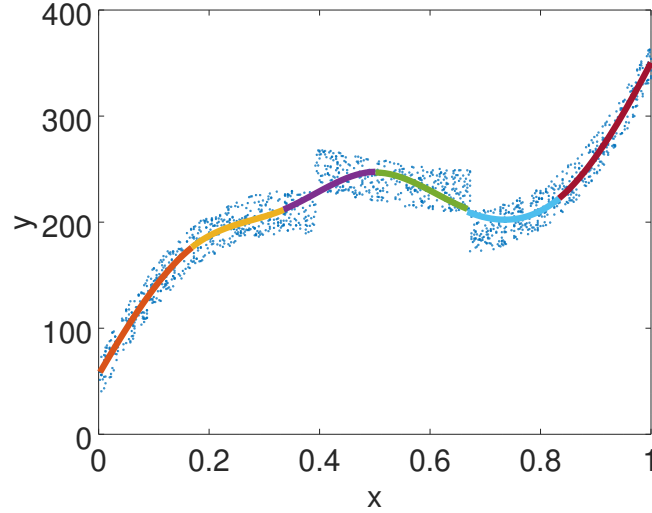


Figure 1.11: Encaixe de $M = 6$ splines cúbicos num conjunto de $N = 500$ pontos $\{x_n, y_n\}$.

Os splines tem seus limites de domínio, nas posições onde x tem valores $d_0, d_1, d_2, \dots, d_M$; assim, podemos definir os vetores $\mathbf{d} = (d_0 \ d_1 \ \dots \ d_M)^T \in \mathbb{R}^{M+1}$ e $\mathbf{p} = (p_0 \ p_1 \ \dots \ p_{4M-1})^T \in \mathbb{R}^{4M}$, de modo que podemos expressar $f(x)$ mediante o uso do polinômio cúbico

$$S_k(x, \mathbf{p}) = P_{4k} x^3 + P_{1+4k} x^2 + P_{2+4k} x + P_{3+4k}, \quad (1.2)$$

na função

$$f(x) \equiv f(x, \mathbf{p}, \mathbf{d}) = \begin{cases} S_0(x, \mathbf{p}) & \text{if } d_0 \leq x \leq d_1 \\ S_1(x, \mathbf{p}) & \text{if } d_1 \leq x \leq d_2 \\ S_2(x, \mathbf{p}) & \text{if } d_2 \leq x \leq d_3 \\ \vdots & \vdots \\ S_{M-1}(x, \mathbf{p}) & \text{if } d_{M-1} \leq x \leq d_M \end{cases} \quad (1.3)$$

1.2.1 Calculando o vetor de parâmetros \mathbf{p}

Agrupando as N amostras $\{x_n, y_n\}$ como mostrado na Seção 1.2.2, podemos definir

$$\mathbf{z} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B}_0 \\ \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix}, \quad \mathbf{R} = \text{diag} \begin{pmatrix} \mathbf{w} \\ \mathbf{w}_a \\ \mathbf{w}_b \\ \mathbf{w}_c \end{pmatrix}; \quad (1.4)$$

onde os vetores \mathbf{w}_a , \mathbf{w}_b e \mathbf{w}_c são escolhido por nós, e indicam o peso que desejamos dar ao cálculo do erro na continuidade de polinômios adjacentes, na suas derivadas de ordem 0, 1 e 2, respetivamente. Assim, para calcula o vetor \mathbf{p} , aplicamos *LMS* (Least Mean Square) definindo a regra de minimização $e(\mathbf{p}) : \mathbb{R}^{4M} \rightarrow \mathbb{R}$,

$$e(\mathbf{p}) = \|\mathbf{z} - \mathbf{Q}\mathbf{p}\|_{\mathbf{R}}^2 + \alpha \|\mathbf{p} - \mathbf{p}_*\|. \quad (1.5)$$

Onde \mathbf{p}_* pode ser entendido como \mathbf{p} numa iteração anterior; é dizer $\mathbf{p}_* \equiv \mathbf{p}_{k-1}$. Assim, aplicando *LMS* e a regularização de Tikhonov, o mínimo valor de $\mathbf{p} = \hat{\mathbf{p}}$ que minimiza $e(\mathbf{p})$ se obtêm iterativamente usando a equação

$$\mathbf{p}_k = \mathbf{p}_{k-1} + [\mathbf{Q}^T \mathbf{R} \mathbf{Q} + \alpha \mathbf{I}]^{-1} \mathbf{Q}^T \mathbf{R} [\mathbf{z} - \mathbf{Q} \mathbf{p}_{k-1}] \quad (1.6)$$

desde um \mathbf{p}_0 sabiamente escolhido, ate que os vetores \mathbf{p}_k e \mathbf{p}_{k-1} sejam muito próximos, onde se declara que $\hat{\mathbf{p}} = \mathbf{p}_k$.

1.2.2 Modelando a solução

Para poder calcular o vetor $\mathbf{p} = (p_0 \ p_1 \ p_2 \ \dots \ p_{4M-1})^T$, do problema spline descrito na Seção 1.2, a partir dos dados $(x_n, y_n, w_n); \forall n \in \mathbb{Z}, 0 \leq n < N$, é necessário escolher os valores $\mathbf{d} = (d_0 \ d_1 \ \dots \ d_M)^T$; ordenamos nossos dados nos seguintes vetores e sub-vetores coluna:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_m \\ \vdots \\ \mathbf{x}_{M-1} \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_m \\ \vdots \\ \mathbf{y}_{M-1} \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_m \\ \vdots \\ \mathbf{w}_{M-1} \end{pmatrix}, \quad (1.7)$$

onde \mathbf{x}, \mathbf{y} e $\mathbf{z} \in \mathbb{R}^N$, e os sub-vetores $\mathbf{x}_m, \mathbf{y}_m$ e \mathbf{w}_m , incluem informação de $\{x_n, y_n, w_n\}$ que tem domínio em $d_m \leq x_n \leq d_{m+1}$.

Minimização do erro $\|\mathbf{y} - \mathbf{A}\mathbf{p}\|_{\mathbf{W}}^2$

Os dados (x_n, y_n) são agrupados de forma que se procura minimizar $\|\mathbf{y} - \mathbf{A}\mathbf{p}\|_{\mathbf{W}}^2$, onde a matriz diagonal $\mathbf{W} = \text{diag}(\mathbf{w})$,

$$\mathbf{A}_m = \begin{pmatrix} \mathbf{x}_m^3 & \mathbf{x}_m^2 & \mathbf{x}_m & 1 \end{pmatrix} \quad (1.8)$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_2 & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}_{M-1} \end{pmatrix} \in \mathbb{R}^{N \times 4M} \quad (1.9)$$

Minimização do erro $\|\mathbf{B}_0 \mathbf{p}\|_{\mathbf{W}_a}^2$ na derivada de ordem 0 em splines adjacentes

Dados os polinômios $S_{k-1}(x, \mathbf{p})$ e $S_k(x, \mathbf{p})$ do $(k-1)$ -ésimo e k -ésimo spline, respetivamente. Podemos igualar ambos polinômios para garantir a continuidade dos splines em $x = d_k$, de modo que,

$$S_{k-1}(d_k, \mathbf{p}) = S_k(d_k, \mathbf{p}), \quad (1.10)$$

Se agrupamos estas equações para os valores de $0 < k < M$, obtemos o sistema $\mathbf{0} = \mathbf{B}_0 \mathbf{p}$, onde $\mathbf{B}_0 \in \mathbb{R}^{M-1 \times 4M}$ é definido como

$$\mathbf{B}_0 = \begin{pmatrix} \mathbf{e}_1 & -\mathbf{e}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{e}_2 & -\mathbf{e}_2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{e}_3 & -\mathbf{e}_3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{e}_{M-1} & -\mathbf{e}_{M-1} \end{pmatrix}, \quad (1.11)$$

usando como variável auxiliar,

$$\mathbf{e}_k = \begin{pmatrix} d_k^3 & d_k^2 & d_k & 1 \end{pmatrix}. \quad (1.12)$$

Equação de continuidade da 1ra derivada dos splines consecutivos

Dados os polinômios $S_{k-1}(x, \mathbf{p})$ e $S_k(x, \mathbf{p})$ do $(k-1)$ -ésimo e k -ésimo spline, respetivamente. Podemos igualar a primeira derivada de ambos polinômios para garantir a continuidade dos splines em $x = d_k$, de modo que,

$$\left. \frac{\partial S_{k-1}(x, \mathbf{p})}{\partial x} \right|_{x=d_k} = \left. \frac{\partial S_k(x, \mathbf{p})}{\partial x} \right|_{x=d_k} \quad (1.13)$$

Assim, agrupando as equações dos polinômios de $0 < k < M$, obtemos $\mathbf{0} = \mathbf{B}_1 \mathbf{p}$, onde,

$$\mathbf{B}_1 = \begin{pmatrix} \partial \mathbf{e}_1 & -\partial \mathbf{e}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \partial \mathbf{e}_2 & -\partial \mathbf{e}_2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \partial \mathbf{e}_3 & -\partial \mathbf{e}_3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \partial \mathbf{e}_{M-1} & -\partial \mathbf{e}_{M-1} \end{pmatrix}, \quad (1.14)$$

usando como variável auxiliar,

$$\partial \mathbf{e}_k = \begin{pmatrix} 3d_k^2 & 2d_k & 1 & 0 \end{pmatrix}. \quad (1.15)$$

Equação de continuidade da 2da derivada dos splines consecutivos

Dados os polinômios $S_{k-1}(x, \mathbf{p})$ e $S_k(x, \mathbf{p})$ do $(k-1)$ -ésimo e k -ésimo spline, respetivamente. Podemos igualar a segunda derivada de ambos polinômios para garantir a continuidade dos splines em $x = d_k$, de modo que,

$$\left. \frac{\partial^2 S_{k-1}(x, \mathbf{p})}{\partial x^2} \right|_{x=d_k} = \left. \frac{\partial^2 S_k(x, \mathbf{p})}{\partial x^2} \right|_{x=d_k} \quad (1.16)$$

Assim, agrupando as equações dos polinômios de $0 < k < M$, obtemos $\mathbf{0} = \mathbf{B}_2 \mathbf{p}$, onde,

$$\mathbf{B}_2 = \begin{pmatrix} \partial^2 \mathbf{e}_1 & -\partial^2 \mathbf{e}_1 & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \partial^2 \mathbf{e}_2 & -\partial^2 \mathbf{e}_2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \partial^2 \mathbf{e}_3 & -\partial^2 \mathbf{e}_3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \partial^2 \mathbf{e}_{M-1} & -\partial^2 \mathbf{e}_{M-1} \end{pmatrix}, \quad (1.17)$$

usando como variável auxiliar,

$$\partial^2 \mathbf{e}_k = \begin{pmatrix} 6d_k & 2 & 0 & 0 \end{pmatrix}. \quad (1.18)$$