

DEDALUS - Acervo - IME

Procedimentos para calculos com splines /



31000010688

RT-MAP-8401-A

PROCEDIMENTOS PARA
CÁLCULOS COM SPLINES
Parte A - Resumos Teóricos

Dirceu Douglas Salvetti

JANEIRO 1984

PROCEDIMENTOS PARA CALCULOS COM SPLINES

INDICE DE CAPITULOS

CAPITULO 0 - RESULTADOS BASICOS

Introdução Teórica aos Splines Polinomiais

CAPITULO 1 - APLICACOES

Ainda em desenvolvimento

PROCEDIMENTOS PARA CALCULOS COM SPLINES

MANUAL DE REFERENCIA

Este manual contém uma exposição detalhada de algoritmos para cálculos com splines e a coleção correspondente de procedimentos codificados em ALGOL.

Encontra-se estruturado em três partes divididas por Capítulos:

- PARTE A - Resumos Teóricos
- PARTE B - Descrição dos Procedimentos
- PARTE C - Listagens de Testes

PARTE A

Na parte A apresenta-se, resumidamente, a justificação teórica dos algoritmos descritos na Parte B, bem como definições e notações utilizadas em suas descrições. Cada algoritmo terá como referência o nome do respectivo procedimento codificado em ALGOL.

A parte A começa por um prefácio e índice de conteúdo do capítulo.

PARTE B

Os procedimentos encontram-se gravados em fita magnética à disposição dos usuários do CCE da Universidade de São Paulo. A parte B principia dando a receita dos comandos de controle para o B-6900 que torna acessível ao usuário os procedimentos splines desejados.

Apresenta em seguida uma breve definição de cada procedimento, relacionando-se de duas formas diferentes:

- (i) em ordem alfabética de seus nomes
- (ii) em ordem alfabética de suas finalidades.

As descrições dos procedimentos encontram-se em ordem alfabética de seus nomes.

Cada descrição possui as seguintes características:

Título: capítulo, finalidade e nome.

Problema: dados e resultados.

Declaração do Procedimento: nome e parâmetros.

Entrada e Saída: parâmetros de entrada e de saída.

Método: breve descrição do método utilizado.

Procedimentos Auxiliares: relação de seus nomes; quando um procedimento auxiliar chamar outro, o nome deste último virá escrito entre parentesis após o nome do primeiro.

Multiplicações ou Divisões: (opcional)

Algoritmo: descrição em linguagem LEAL ([1] , [2]);

Codificação: linguagem ALGOL.

Apresentamos a seguir um ÍNDICE DE CAPÍTULOS. A numeração das páginas é independente para cada capítulo e cada parte.

[1] Setzer, V.W., Sanches, M.M; A Linguagem LEAL para Ensino Básico de Computação, Instituto de Matemática e Estatística da USP, RT-MAP-7704, 1977.

[2] Freund, G., Homem de Melo, I.S., Yoshida, L.R.; LEAL, Manual de Referência - 1980 - Publicação interna do DMA do IME da USP.

PREFACIO

Esta introdução foi compilada de SPLINE FUNCTIONS I ; de Larry L. Schumacker, conservando-se as definições e notações adotadas pelo seu autor.

A finalidade do resumo apresentado nesta introdução é servir de apoio para uma melhor compreensão dos algoritmos implantados nos procedimentos codificados em ALGOL e descritos na Parte B.

A maior parte das demonstrações foram omitidas, ou parcialmente omitidas, podendo ser obtidas diretamente da obra acima citada, Cap.4 e Cap.5 .

As referências bibliográficas para os procedimentos implantados são as seguintes:

- (a) Schumacker, Larry L.; SPLINE FUNCTIØNS I, preprint , 1979
- (b) Cox, M.G.; THE NUMERICAL EVALUATIØN of B-SPLINES, J.Inst. Maths Applics (1972)10,134-139.
- (c) Boor, Carl de; ON CALCULATING WITH B-SPLINES, Journal of Aproximation Theory 6, 50-62 (1972).
- (d) Boor, Carl de; PACKAGE FOR CALCULATING WITH B-SPLINES, SIAM J. Numer. Anal., Vol. 14, No.3, June 1977.

Os procedimentos deste capítulo referem-se unicamente a avaliação de splines, de suas derivadas e integrais e à conversão da expansão em B-splines para a representação polinomial por partes.

Os nomes dos procedimentos que começam por BS envolvem B-splines normalizados associados a nós quaisquer, de multiplicidades

arbitrárias, definidos numa partição estendida, Os nomes que começam por BS e terminam por H indicam B-splines associados com nós simples (multiplicidade 1) de uma partição uniforme de passo h .

O material deste capítulo é suficiente para sua utilização em inúmeras aplicações, tais como: interpolação, aproximação pelo Método dos Mínimos Quadrados, solução numérica de problemas em equações diferenciais, etc.

1983

INDICE DO CAPITULO 0

	PAG.
1. PROBLEMAS DA APROXIMACAO	
1.1 - Polinômios -----	A-02
1.2 - Polinômios por partes -----	A-03
1.3 - Polinômios splines -----	A-05
2. SPLINES	
2.1 - Partição Δ -----	A-05
2.2 - Vetor de multiplicidade m -----	A-05
2.3 - Espaço $\mathcal{S}(\mathcal{P}_m; m; \Delta)$ -----	A-06
2.4 - Dimensão de \mathcal{S} -----	A-06
2.5 - Base de um-so-lado -----	A-07
2.6 - Uma base local e simétrica para \mathcal{S} -----	A-08
2.6.1 - Matrizes e determinantes -----	A-09
2.6.2 - Matriz e determinante de Vandermond -----	A-10
2.6.3 - Diferenças Divididas -----	A-10
2.6.4 - Construção de uma função com suporte finito -----	A-13
2.7 - Partição estendida $\bar{\Delta}$ -----	A-14
2.8 - B-splines -----	A-14
2.9 - B-splines normalizados -----	A-16
3. ALGORITMOS	
3.1 - Armazenamento e avaliação de um spline -----	A-17
3.1.1 - Avaliação de um spline através de sua expansão em B-splines(LBISEC-INTERV_BSMQ-BSX) -----	A-17
3.1.2 - Avaliação de um spline através de sua expansão em B-splines(BSX2) -----	A-19
3.1.3 - Partição simples e partição estendida -----	A-21
3.2 - Derivadas(BSMCD) -----	A-22
3.2.1 - Determinação dos coeficientes de $D_+^{j-1} s(x)$. \in $\mathcal{S}(\mathcal{P}_{m-j+1}^{m'}; \Delta)$ (BSCDJ) -----	A-27

3.2.2 - Avaliação de $D^{j-1}s(x)$ (BSDJX) -----	A-34
3.2.3 - Avaliação de $\{D^{j-1}s(x)\}_1^m$ (BSDX2) -----	A-35
3.2.4 - Avaliação de $\{D^{j-1}s(x)\}_1^m$ (BSDX) -----	A-35
3.3 - Integrais (BSMCI) -----	A-36
3.3.1 - Integral definida (BSID) -----	A-38
3.3.2 - Produtos internos de B-splines normalizados -----	A-44
3.3.3 - Fórmula de integração numérica de Gauss -----	A-44
3.3.4 - Matriz de Gram (BSMGRM) -----	A-45
3.4 - Representação Polinomial por partes (RPPSX, RPPSDX) -----	A-46
3.4.1 - Conversão da expansão em B-splines para a representação polinomial por partes (BSRPP) -----	A-47

4. B-SPLINES PARA NO IGUALMENTE ESPACADOS

4.1 - Partição uniforme de passo h -----	A-49
4.2 - Partição estendida associada com $\mathcal{S}_m(\Delta h)$ -----	A-49
4.3 - Representação de $s \in \mathcal{S}_m(\Delta h)$ -----	A-51
4.4 - Avaliação de $s \in \mathcal{S}_m(\Delta h)$ (BSX2H, BSMQH, BSXH) -----	A-51
4.5 - Derivadas de $s \in \mathcal{S}_m(\Delta h)$ (BSMCDH, BSDXH) -----	A-51
4.6 - Integrais de $s \in \mathcal{S}_m(\Delta h)$ (BSMCIH, BSIDH) -----	A-53
4.7 - Produtos internos -----	A-53
4.8 - Conversão de uma expansão em B-splines de $\mathcal{S}_m(\Delta h)$ para sua representação polinomial por partes (BSRPPH) -----	A-53

1. PROBLEMAS DE APROXIMAÇÃO

Funções constituem a ferramenta matemática básica para descrição e análise de processos físicos. Somente em alguns casos tais funções são conhecidas explicitamente, necessitando-se, na maior parte das vezes, construir-se aproximações baseadas apenas em limitadas informações sobre os processos físicos que representam. Tais problemas de aproximação constituem a parte central da Teoria da Aproximação.

Há duas grandes categorias de problemas de aproximação. A primeira delas engloba os problemas em que toda informação sobre a função desconhecida está restrita a um número finito de valores da função, em geral obtidos experimentalmente, e, portanto, sujeitos aos erros iniciais de observação. Estes problemas são conhecidos como *Ajustamento de Dados*.

A segunda grande categoria envolve modelos matemáticos que representam inúmeros processos físicos. Estes modelos usualmente descrevem equações operacionais que definem implicitamente as respectivas funções. Por equações operacionais referimo-nos aos problemas com valores iniciais ou de contorno em equações diferenciais ordinárias, em equações diferenciais a derivadas parciais, em equações integro-diferenciais, aos problemas de controle ótimo, etc. Estes problemas de aproximação são denominados *Problemas em Equações Operacionais*.

A resolução dos problemas de aproximação envolve usualmente a seguinte estratégia:

- (i) Escolha de uma classe de funções com certo poder de aproximação.
- (ii) Aplicação de um processo de seleção que permite separar uma determinada função da classe como solução aproximada de um problema específico.

O estudo das classes de funções de aproximação é feito na *Teoria da Aproximação* enquanto que a pesquisa e a análise de algoritmos eficientes, utilizando tais classes, se fazem em *Análise Numérica*.

PROCEDIMENTOS PARA CALCULOS COM SPLINES

INDICE DE CAPITULOS

CAPITULO 0 - RESULTADOS BASICOS

Introdução Teórica aos Splines Polinomiais

CAPITULO 1 - APLICACOES

Ainda em desenvolvimento

Os polinômios, durante muitos anos, ocuparam a parte central da Teoria da Aproximação e da Análise Numérica.

As funções splines, entretanto, vêm substituindo com vantagem os polinômios como classe de funções de aproximação.

A teoria das funções splines apresentou um rápido desenvolvimento a partir de 1960 devido principalmente à sua grande utilidade nas aplicações.

As classes de funções splines possuem ótimas propriedades estruturais e grande poder de aproximação. Devido a facilidade com que são armazenadas, avaliadas e manipuladas em computador digital acabaram sendo utilizadas na resolução numérica de inúmeros problemas que ocorrem nos mais variados ramos da Matemática Aplicada.

1.1 POLINÔMIOS

O sucesso da resolução de um problema de aproximação depende da existência de uma conveniente classe de funções aproximadoras. Uma classe A de funções aproximadoras deve possuir pelo menos as seguintes propriedades básicas:

- 1) As funções de A devem ser relativamente suaves.
(As funções que representam processos físicos são suaves)
- 2) As funções de A devem ser fáceis de armazenar e manipular em computador digital.
(Muitos problemas que ocorrem no mundo real não podem ser resolvidos sem a ajuda de um computador digital de alta velocidade)
- 3) As funções de A devem ser fáceis de avaliar num computador digital, bem como suas derivadas e integrais.
- 4) A classe A deve ser suficientemente ampla a fim de poder aproximar bem uma função suave qualquer.

Seja $\mathcal{P}_m = \{p(x) : p(x) = \sum_{i=1}^m c_i x^{i-1}, c_1, \dots, c_m, x \text{ reais}\}$ o espaço dos polinômios de ordem m (grau $m-1$).

Polinômios foram escolhidos como classe de funções aproximadoras porque o espaço \mathcal{P}_m possui as seguintes características:

- 1) \mathcal{P}_m é um espaço vetorial de dimensão m com uma base conveniente.
- 2) Polinômios são funções suaves.
- 3) Polinômios são fáceis de armazenar, avaliar e manipular em computadores digitais.
- 4) Derivadas e integrais de polinômios são polinômios fáceis de serem obtidos algebricamente ou computavelmente.
- 5) O número de zeros de um polinômio de ordem m não excede $m-1$.
- 6) Matrizes que ocorrem em interpolação ou ajustamento por polinômios são não singulares.
- 7) A estrutura de sinal e curva de um polinômio está intimamente relacionada com a estrutura de sinal de seus coeficientes.
- 8) Dada uma função contínua em $[a, b]$ existe um polinômio que está uniformemente próximo dela.
- 9) Podem ser obtidas medidas precisas da convergência das aproximações polinomiais às funções suaves.

Embora estas propriedades apontem para os polinômios como ideais aos propósitos de aproximação, na prática foi observado que possuem um comportamento indesejável, o que leva a conjecturar sobre a existência de outras classes melhores de funções de aproximação. Este comportamento revela uma espécie de inflexibilidade da classe \mathcal{P}_m e constitui sua próxima característica:

- 10) Muitos processos de aproximação envolvendo polinômios tendem a construir polinômios de aproximação que oscilam descontroladamente.

1.2 POLINÔMIOS POR PARTES

A maior objeção para uso de \mathcal{P}_m como classe de funções de aproximação

ção é sua relativa inflexibilidade. Entretanto, para intervalos suficientemente pequenos e polinômios de ordem menor ou igual a 3 ou 4, as oscilações quase desaparecem.

Este comportamento nos leva a pensar numa classe de funções de aproximação com maior flexibilidade que \mathcal{P}_m porém mantendo suas boas propriedades. A idéia inicial para a obtenção dessa classe é considerar polinômios de ordem relativamente baixa e particionar o intervalo de aproximação em subintervalos menores. Isto nos leva à seguinte consideração:

Dado o intervalo fechado $I = [a, b]$ seja $\Delta = \{x_i\}_1^K$ um conjunto de pontos tais que $a = x_0 < x_1 < \dots < x_K < x_{K+1} = b$.

O conjunto Δ , chamado partição de I , decompõe I em $K+1$ subintervalos $I_i = [x_i, x_{i+1}]$, $i=0, 1, \dots, K-1$ e $I_K = [x_K, x_{K+1}]$.

Dado um inteiro positivo m definimos

$\mathcal{PP}_m(\Delta) = \{f: \text{existem polinômios } P_0, P_1, \dots, P_K \text{ em } \mathcal{P}_m \text{ tais que } f(x) = P_i(x) \text{ se } x \in I_i, i=0, \dots, K\}$, denominado espaço dos polinômios por partes de ordem m com nós x_1, \dots, x_K .

Se $f \in \mathcal{PP}_m$ é claro que f é formado por $K+1$ polinômios, cada um deles definido em cada um dos $K+1$ subintervalos. Embora \mathcal{PP}_m seja mais flexível que \mathcal{P}_m não mantém, necessariamente, a desejada suavidade, pois pode ocorrer que $f \in \mathcal{PP}_m$ não seja nem mesmo contínua. Isto desclassifica \mathcal{PP}_m como classe de funções aproximadoras e é provavelmente o motivo pelo qual até 1960 \mathcal{PP}_m desempenhou pouca importância em Análise Numérica e Teoria da Aproximação. Apesar disso polinômios por partes foram utilizados em fórmulas de integração numérica (Newton-Cotes, Gauss), na resolução numérica de problemas com valores iniciais em equações diferenciais ordinárias (Método de Euler) e em análise na demonstração do Teorema da Aproximação Uniforme de Weierstrass.

1.3 POLINÔMIOS SPLINES

A fim de conservar a flexibilidade conquistada em \mathcal{PP}_m e manter a suavidade de \mathcal{P}_m definimos a seguinte classe de funções, chamada espaço dos polinômios splines de ordem m com nós simples nos pontos x_1, x_2, \dots, x_K :

$$\mathcal{S}_m(\Delta) = \mathcal{PP}_m(\Delta) \cap C^{m-2}[a,b]$$

Variando-se o grau de suavidade entre os subintervalos pode-se definir diferentes espaços de polinômios splines.

Além dos polinômios splines ou splines polinomiais existem os *splines não-polinomiais* e os *splines multidimensionais*.

Neste texto vamos nos restringir aos splines polinomiais e por isso muitas vezes referir-nos-emos a eles simplesmente por splines.

Verifica-se que os splines conservam praticamente as 9 primeiras características de \mathcal{P}_m acrescida da seguinte:

- 10) splines de ordem baixa são bastante flexíveis e não exibem as oscilações usualmente apresentadas pelos polinômios.

2. SPLINES

2.1 PARTIÇÃO Δ

Seja $I = [a,b]$ e $\Delta = \{x_i\}_1^K$ tal que $a=x_0 < x_1 < \dots < x_K < x_{K+1} = b$ uma partição de I em $K+1$ subintervalos $I_i = [x_i, x_{i+1})$, $i=0, \dots, K-1$ e $I_K = [x_K, x_{K+1}]$.

2.2 VETOR DE MULTIPLICIDADE m

Seja m um inteiro positivo e $m = (m_1, \dots, m_K)$ um vetor de inteiros que satisfazem $1 \leq m_i \leq m$, $i=1, 2, \dots, K$.

2.3 SPLINES POLINOMIAIS OU SIMPLEMENTE SPLINES

Seja $\mathcal{S}(\mathcal{P}_m; \mathbf{m}; \Delta) = \{s: \text{existem polinômios } p_0, p_1, \dots, p_k \text{ em } \mathcal{P}_m \text{ tais que } s(x) = p_i(x) \text{ para } x \in I_i, i=0,1,\dots,k \text{ e, além disso, } D^j p_{i-1}(x_i) = D^j p_i(x_i), j=0,1,\dots,m-1-m_i, i=1,2,\dots,k\}$

Denominamos $\mathcal{S}(\mathcal{P}_m; \mathbf{m}; \Delta)$ espaço dos polinômios splines de ordem m com nós x_1, \dots, x_k de multiplicidades m_1, \dots, m_k , respectivamente.

Como as multiplicidades podem ser diferentes os polinômios por partes em $\mathcal{S}(\mathcal{P}_m; \mathbf{m}; \Delta)$ geralmente possuem graus diferentes de suavidade em cada nó $x_i, i=1, \dots, k$.

Se $m_i = m, i=1, \dots, k$, dois polinômios splines p_{i-1} e p_i definidos em intervalos vizinhos ao nó x_i podem não estar relacionados entre si e apresentar em x_i um ponto de descontinuidade. Concluimos que neste caso

$$\mathcal{S}(\mathcal{P}_m; \mathbf{m}; \Delta) = \mathcal{P}_m(\Delta)$$

Se $\mathbf{m} = (1, 1, \dots, 1)$ então $\mathcal{S}(\mathcal{P}_m; \mathbf{m}; \Delta) = \mathcal{S}_m(\Delta)$, isto é, corresponde ao espaço spline mais suave possível.

Em muitos problemas práticos usualmente se formula o problema de aproximação sobre $I = [a, b]$ e, neste caso, utiliza-se o espaço $\mathcal{S}(\mathcal{P}_m; \mathbf{m}; \Delta)$ definido sobre I .

Entretanto, cada splines $s \in \mathcal{S}(\mathcal{P}_m; \mathbf{m}; \Delta)$ possui uma extensão natural em \mathbb{R} definida por $s(x) = p_0(x)$ para $x < a$ e $s(x) = p_k(x)$ para $x > b$.

2.4 DIMENSÃO DE \mathcal{S}

$\mathcal{S}(\mathcal{P}_m; \mathbf{m}; \Delta)$ é um espaço vetorial de dimensão $n = m + \sum_{i=1}^k m_i$

A estrutura de espaço vetorial decorre da própria definição de $\mathcal{S}(\mathcal{P}_m; \mathbf{m}; \Delta)$.

Seja $s \in \mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ e $P_i(x) = \sum_{j=1}^m c_{ij} x^j$, $i=0,1,\dots,k$ os polinômios que representam s em cada subintervalo I_i , $i=0,1,\dots,k$, respectivamente.

A dimensão de $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ igual a n decorre, intuitivamente, da seguinte consideração:

- (i) número de coeficientes livres $(k+1)m$
- (ii) número de equações entre os coeficientes impostas pelas condições de continuidade em cada nó de $\Delta = \{x_i\}_1^k$: $\sum_{i=1}^k (m-m_i)$
- (iii) número de coeficientes independentes:

$$(k+1)m - \sum_{i=1}^k (m-m_i) = km+m-km + \sum_{i=1}^k m_i = m + \sum_{i=1}^k m_i = n$$

A demonstração rigorosa deve verificar que as condições (ii) são de fato independentes entre si.

2.5 BASE DE UM - SÔ - LADO de $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$

Como $\mathcal{P}_m \in \mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$, uma base de $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ deve conter uma base de \mathcal{P}_m . Portanto, m das n funções de uma base de $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ podem ser

$$1, (x-a), (x-a)^2, \dots, (x-a)^{m-1}$$

Os outros $n-m$ elementos da base são obtidos a partir da seguinte consideração:

$$\text{Seja } (x-y)_+^j = (x-y)^j (x-y)_+^0, \quad j > 0, \quad (x-y)_+^0 = \begin{cases} 0, & x < y \\ 1, & x \geq y \end{cases}$$

Definindo-se $x_0 = a$ e $m_0 = m$, demonstra-se que

$$\left\{ \rho_{ij}(x) = (x-x_i)_+^{m-j} \right\}_{i=1, j=0}^{m_i, K}$$

é uma base de $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$, chamada base de um-sô-lado (one-sided-basis)

Note que $(x-x_i)_+^j = 0$ para $x < x_i$ enquanto que para $x \geq x_i$

$(x-x_i)_+^j = (x-x_i)^j$ é um polinômio de grau j .

Em relação a esta base $s \in \mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ é expressa por

$$s(x) = \sum_{i=0}^k \sum_{j=1}^{m_i} c_{ij} \rho_{ij}(x) = \sum_{i=0}^k \sum_{j=1}^{m_i} c_{ij} (x-x_i)_+^{m-j}, \quad x \in [a, b]$$

Embora útil para considerações teóricas esta base não é conveniente para aplicações numéricas. Por exemplo, a avaliação de $s(x)$ para x próximo de b exigirá a avaliação de todos os elementos da base. Além disso, verificou-se que o processo de avaliação é muito instável do ponto de vista numérico.

2.6 UMA BASE LOCAL E SIMÉTRICA PARA $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$

Devido a instabilidade numérica no processo de avaliação de um spline representado na base de um-só-lado, um grande número de métodos que envolvem computação com splines utilizaram alguma forma de representação de splines através de polinômios por partes.

O estudo de funções simétricas dotadas de suportes adequados, chamados B-Splines, permitiu a introdução de uma base com propriedades locais extremamente útil para aplicações numéricas. No início não houve muito uso dessa base formada por funções B-Splines. Greville em 1964 fez uma das primeiras aplicações em problemas de interpolação. Apesar das vantagens numéricas dos B-Splines o uso de polinômios por partes tornou-se mais popular. Talvez a utilização de diferenças divididas na definição de uma função B-Spline tenha contribuído para seu pouco uso no início. Entretanto, a partir de 1972, a descoberta de uma relação recursiva (Cox e de Boor) por meio do qual se pode gerar B-Splines de uma certa ordem a partir de B-Splines de ordem imediatamente inferior, levou à formulação de algoritmos de avaliação numericamente estáveis. O uso dos B-Splines a partir de então passou a predominar nas aplicações.

Apresentaremos a seguir resumos de resultados úteis para a definição de funções B-Splines.

2.6.1 MATRIZES E DETERMINANTES

Seja $\{u_i\}_i^m$ um conjunto de funções definidas em $I = [a, b]$ e t_1, t_2, \dots, t_m pontos de I tais que $t_1 < t_2 < \dots < t_m$.

Chama-se matriz associada com $\{u_i\}_i^m$ e $\{t_i\}_i^m$ à

$$M \begin{pmatrix} t_1, t_2, \dots, t_m \\ u_1, u_2, \dots, u_m \end{pmatrix} = \begin{bmatrix} u_1(t_1) & u_2(t_1) & \dots & u_m(t_1) \\ u_1(t_2) & u_2(t_2) & \dots & u_m(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(t_m) & u_2(t_m) & \dots & u_m(t_m) \end{bmatrix}$$

cujo determinante indicamos por $D \begin{pmatrix} t_1, \dots, t_m \\ u_1, \dots, u_m \end{pmatrix}$.

Seja $t_1 \leq t_2 \leq \dots \leq t_m = \underbrace{\tau_1, \dots, \tau_1}_{\ell_1} < \underbrace{\tau_2, \dots, \tau_2}_{\ell_2} < \dots < \underbrace{\tau_d, \dots, \tau_d}_{\ell_d}$ is

to é, um conjunto de pontos de I onde alguns τ 's são iguais a outros. Nesta notação o ponto τ_i aparece repetido ℓ_i vezes e $\sum_{i=1}^d \ell_i = m$

Supondo que as funções u_1, \dots, u_m sejam suficientemente diferenciáveis, definimos:

$$M \begin{pmatrix} t_1, \dots, t_m \\ u_1, \dots, u_m \end{pmatrix} = \begin{bmatrix} u_1(\tau_1) & u_2(\tau_1) & \dots & u_m(\tau_1) \\ Du_1(\tau_1) & Du_2(\tau_1) & \dots & Du_m(\tau_1) \\ \vdots & \vdots & \ddots & \vdots \\ D^{\ell_1-1}u_1(\tau_1) & D^{\ell_1-1}u_2(\tau_1) & \dots & D^{\ell_1-1}u_m(\tau_1) \\ \vdots & \vdots & \ddots & \vdots \\ u_1(\tau_d) & u_2(\tau_d) & \dots & u_m(\tau_d) \\ Du_1(\tau_d) & Du_2(\tau_d) & \dots & Du_m(\tau_d) \\ \vdots & \vdots & \ddots & \vdots \\ D^{\ell_d-1}u_1(\tau_d) & D^{\ell_d-1}u_2(\tau_d) & \dots & D^{\ell_d-1}u_m(\tau_d) \end{bmatrix}$$

$\left. \begin{matrix} \dots \\ \dots \\ \dots \end{matrix} \right\} \ell_1$
 $\left. \begin{matrix} \dots \\ \dots \\ \dots \end{matrix} \right\} \ell_d$

2.6.2 MATRIZ E DETERMINANTE DE VANDERMOND

Seja $\{u_i\}_1^m = \{x^{i-1}\}_1^m$.. Neste caso

$$VM(t_1, t_2, \dots, t_m) = M \begin{pmatrix} t_1, t_2, \dots, t_m \\ 1, x, \dots, x^{m-1} \end{pmatrix}$$

$$e \quad V(t_1, t_2, \dots, t_m) = D \begin{pmatrix} t_1, t_2, \dots, t_m \\ 1, x, \dots, x^{m-1} \end{pmatrix}$$

Se os pontos $\{t_i\}_1^m$ são todos distintos, $V(t_1, \dots, t_m) = \prod_{1 \leq i < j \leq m} (t_j - t_i)$
 $1 \leq i < j \leq m$.

Se $t_1 \leq t_2 \leq \dots \leq t_m = \underbrace{\tau_1 = \dots = \tau_1}_{\ell_1} < \underbrace{\tau_2 = \dots = \tau_2}_{\ell_2} < \dots < \underbrace{\tau_d = \dots = \tau_d}_{\ell_d}$ então

$$V(t_1, t_2, \dots, t_m) = \prod_{1 \leq i < j \leq d} (\tau_j - \tau_i)^{\ell_j \ell_i} \prod_{i=1}^d \prod_{v=1}^{\ell_i-1} v!$$

Se $t_1 = t_2 = \dots = t_m$ então $V(t_1, t_2, \dots, t_m) = \prod_{v=1}^{m-1} v!$

2.6.3 DIFERENÇAS DIVIDIDAS

Sejam t_1, t_2, \dots, t_{r+1} pontos do campo de definição de uma função f .

Chama-se diferença dividida de ordem r de f a

$$[t_1, t_2, \dots, t_{r+1}] f = \frac{D \begin{pmatrix} t_1, t_2, \dots, t_{r+1} \\ 1, x, \dots, f \end{pmatrix}}{D \begin{pmatrix} t_1, t_2, \dots, t_{r+1} \\ 1, x, \dots, x^r \end{pmatrix}}$$

Esta definição permanece válida para qualquer ordem dos pontos $\{t_i\}_1^{r+1}$, isto é, se $\tilde{t}_1 \leq \tilde{t}_2 \leq \dots \leq \tilde{t}_{r+1}$ são pontos em ordem crescente do conjunto

$\{t_i\}_1^{r+1}$ então $[\tilde{t}_1, \dots, \tilde{t}_{r+1}] f = [t_1, \dots, t_{r+1}] f$

Se os t 's forem distintos entre si a diferença dividida estará definida para qualquer função que assuma valores finitos naqueles pontos.

Entretanto, se alguns t 's ocorrerem repetidos então o determinante do numerador utilizado na definição da diferença dividida irá depender

de certas derivadas de f e a diferença dividida pode não estar definida.

PROPRIEDADES

1. Se t_1, \dots, t_{r+1} são pontos distintos então

$$[t_1, \dots, t_{r+1}]f = \sum_{i=1}^{r+1} [f(t_i) / \prod_{\substack{j=1 \\ j \neq i}}^{r+1} (t_i - t_j)]$$

Se $t_1, t_2, \dots, t_{r+1} = \underbrace{\tau_1, \dots, \tau_1}_{\ell_1}, \underbrace{\tau_2, \dots, \tau_2}_{\ell_2}, \dots, \underbrace{\tau_d, \dots, \tau_d}_{\ell_d}, \sum_{i=1}^d \ell_i = r+1$

sendo $\tau_1 < \tau_2 < \dots < \tau_d$ então

$$[t_1, \dots, t_{r+1}]f = \sum_{i=1}^d \prod_{j=1}^{\ell_i} d_{ij} D^{j-1}f(\tau_i), \quad d_{i\ell_i} \neq 0, \quad i=1, \dots, d$$

CONSEQUÊNCIAS

(i) A ordem dos pontos t 's é irrelevante na definição de diferença dividida.

(ii) A diferença dividida é linear.

2. Dados t_1, t_2, \dots, t_{r+1} não necessariamente em ordem e uma função f suficientemente diferenciável então

se $t_1 \neq t_{r+1}$, $[t_1, \dots, t_{r+1}]f = \frac{[t_2, \dots, t_{r+1}]f - [t_1, \dots, t_r]f}{t_{r+1} - t_1}$

se $t_1 = t_2 = \dots = t_{r+1}$, $[t_1, \dots, t_{r+1}]f = D^r f(t_1)/r!$

se $f \in C^r[a, b]$, $a = \min\{t_i\}_1^{r+1}$ e $b = \max\{t_i\}_1^{r+1}$ então existe θ ,

$a \leq \theta \leq b$, tal que

$$[t_1, \dots, t_{r+1}]f = \frac{D^r f(\theta)}{r!}$$

se $f(x) = x^j$ então $[t_1, \dots, t_{r+1}]x^j = \begin{cases} 0, & 0 \leq j < r \\ \rho_{j-r}(t_1, \dots, t_{r+1}), & j \geq r \end{cases}$

$$\rho_0(t_1, \dots, t_{r+1}) = 1$$

$$\rho_\ell(t_1, \dots, t_{r+1}) = \sum (t_{i_1} t_{i_2} \dots t_{i_\ell})$$

$$1 \leq i_1 \leq i_2 \leq \dots \leq i_\ell \leq r+1$$

sendo a soma estendida para todas escolhas de ℓ inteiros, em que ℓ indica repetição.

Por exemplo

$$\rho_1(t_1, t_2, t_3) = t_1 + t_2 + t_3$$

$$\rho_2(t_1, t_2) = t_1^2 + t_1 t_2 + t_2^2$$

Existem $(r+\ell)!/(r!\ell!)$ parcelas.

3. Regra de Liebniz

Se t_1, t_2, \dots, t_{r+1} são pontos do campo de definição de duas funções f e g suficientemente diferenciáveis então

$$[t_1, \dots, t_{r+1}]fg = \sum_{i=1}^{r+1} [t_1, \dots, t_i]f [t_i, \dots, t_{r+1}]g$$

4. Seja $t_1 \leq t_2 \leq \dots \leq t_{r+1}$ e j , entre 1 e $r+1$, fixado.

Se t_j não aparece repetido e f é suficientemente diferenciável então

$$\frac{\partial}{\partial t_j} [t_1, \dots, t_{r+1}]f = [t_1, \dots, t_{j-1}, t_j, t_j, t_{j+1}, \dots, t_{r+1}]f$$

5. Seja $t_1, \dots, t_{r+1} = \underbrace{\tau_1, \dots, \tau_1}_{\ell_1}, \underbrace{\tau_2, \dots, \tau_2}_{\ell_2}, \dots, \underbrace{\tau_d, \dots, \tau_d}_{\ell_d}, \sum_{i=1}^d \ell_i = r+1$

Consideremos o caso em que todo bloco τ_i, \dots, τ_i se move.

$$\text{Se } \tau_i < \tau_{i+1} \text{ então } \frac{\partial^+}{\partial \tau_i} [t_1, \dots, t_{r+1}]f = \ell_i [\tau_1, \dots, \tau_i^{\ell_{i+1}}, \dots, \tau_d^{\ell_d}]f$$

sendo $\frac{\partial^+}{\partial \tau_i}$ a derivada a direita. Se $\tau_{i-1} < \tau_i$ a mesma expressão vale pa

ra a derivada a esquerda $\frac{\partial^-}{\partial \tau_i}$.

6. Se t_1, \dots, t_{r+1} são pontos igualmente espaçados com passo h , isto é,

respectivamente iguais a $t, t+h, \dots, t+rh$, chama-se diferença simples de ordem r a

$$\Delta_h^r f(t) = \Delta(\Delta^{r-1} f(t)) = \dots = \underbrace{\Delta_h(\Delta_h(\dots \Delta_h f(t)))}_r$$

sendo $\Delta_h f(t) = f(t+h) - f(t)$

São válidas as seguintes propriedades:

- $\Delta_h^r f(t) = r! h^r [t, t+h, \dots, t+rh] f$
- $\Delta_h^r f(t) = \sum_{i=0}^r (-1)^{r-i} \binom{r}{i} f(t+ih)$
- $\Delta_h^r x^i = r! h^r \delta_{ir}, i=0, 1, \dots, r$ ($\delta_{ir} = \begin{cases} 0, & i \neq r \\ 1, & i = r \end{cases}$)
- $|\Delta_h^r f(t)| \leq 2^r \|f\|_\infty$
- Se $f \in C^r [t, t+rh]$ então $\Delta_h^r f(t) = h^r D^r f(\theta), t \leq \theta \leq t + rh$

2.6.4 CONSTRUÇÃO DE UMA FUNÇÃO COM SUPORTE FINITO

Seja $\tau_1 < \tau_2 < \dots < \tau_d, 1 \leq \ell_i \leq m, i=1, 2, \dots, d$ e

$$B(x) = \sum_{i=1}^d \sum_{j=1}^{\ell_i} \alpha_{ij} \frac{(x - \tau_i)^{m-j}}{(m-j)!} \quad \text{onde} \quad \frac{(x - \tau_i)^{m-j}}{(m-j)!} \quad \text{são as fun-}$$

ções de um-só-lado utilizadas na construção de uma base de $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$

Pode-se demonstrar que

- (i) Se $\sum_{i=1}^d \ell_i > m$ então existe $\{\alpha_{ij}\}_{i=1}^d, j=1, \dots, \ell_i$, não simultaneamente nulos, tais que $B(x)=0$ para $x \notin [\tau_1, \tau_d]$. O intervalo $[\tau_1, \tau_d]$ chama-se suporte de $B(x)$.
- (ii) Se $\sum_{i=1}^d \ell_i = m+1$ então $B(x) = c \underbrace{[\tau_1, \dots, \tau_1]}_{\ell_1} \underbrace{[\tau_2, \dots, \tau_2]}_{\ell_2} \dots \underbrace{[\tau_d, \dots, \tau_d]}_{\ell_d} (x-y)_+^{m-1}$

onde c é uma constante não nula, ou seja $B(x)$, a menos de um fator de multiplicidade, é a diferença dividida de ordem $m+1$ da função de Green $(x-y)_+^{m-1}$

Se desejamos construir uma base de $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ tal que a cada elemento esteja associado um suporte local, pelos resultados anteriores deveremos considerar diferenças divididas da função $(x-y)_+^{m-1}$.

2.7 PARTIÇÃO ESTENDIDA $\tilde{\Delta}$ ASSOCIADA COM $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$

Seja $\Delta = \{x_i\}_1^k$ uma partição de $I = [a, b]$, $1 \leq m_i \leq m$, $i=1, 2, \dots, k$ e

$$n = m + \sum_{i=1}^k m_i.$$

Se $y_1 \leq y_2 \leq \dots \leq y_{n+m}$ são definidos por:

$$y_1 \leq y_2 \leq \dots \leq y_m = a$$

$$b \leq y_{n+1} \leq \dots \leq y_{n+m}$$

$$y_{m+1} \leq y_{m+2} \leq \dots \leq y_n = \underbrace{x_1, \dots, x_1}_{m_1}, \dots, \underbrace{x_k, \dots, x_k}_{m_k}$$

chamamos $\tilde{\Delta} = \{y_i\}_1^{m+n}$ partição estendida associada com $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$

2.8 B-SPLINES

Seja $\dots \leq y_{-i} \leq y_0 \leq y_1 \leq y_2 \leq \dots$ uma seqüência de números reais.

Dados os inteiros i e m , $m > 0$, definimos

$$Q_i^m(x) = \begin{cases} (-1)^m [y_i, \dots, y_{i+m}] (x-y)_+^{m-1} & \text{se } y_i < y_{i+m} \\ 0 & \text{em caso contrário} \end{cases}$$

para todo x real.

$Q_i^m(x)$ chama-se *B-splines de ordem m associado com os nós y_i, \dots, y_{i+m}*

Demonstram-se as seguintes propriedades:

$$1. \text{ Se } y_i < y_{i+1} = \dots = y_{i+m} \text{ então } Q_i^m(x) = \begin{cases} (x-y_i)^{m-1} / (y_{i+m}-y_i)^m & \text{se } x \in [y_i, y_{i+m}) \\ 0 & \text{em caso contrário} \end{cases}$$

Se $y_i = \dots = y_{i+m-1} < y_{i+m}$ então $Q_i^m(x) = \begin{cases} (y_{i+m}-x)^{m-1}/(y_{i+m}-y_i)^m & \text{se } x \in [y_i, y_{i+m}) \\ 0 & \text{em caso contrário} \end{cases}$

2. Se $y_i < y_{i+m}$ e $y_i \leq \dots \leq y_{i+m} = \underbrace{\tau_1, \dots, \tau_1}_{\ell_1}, \dots, \underbrace{\tau_d, \dots, \tau_d}_{\ell_d}$ então

$$Q_i^m(x) = \sum_{j=1}^d \sum_{k=1}^{\ell_j} \alpha_{jk} (x - \tau_{jk})^{m-k} \quad \text{com } \alpha_{j\ell_j} \neq 0, j=1,2,\dots,d$$

e $D_-^k Q_i^m(\tau_j) = D_+^k Q_i^m(\tau_j), k=0,1,\dots,m-\ell_j-1, j=1,2,\dots,d$

Estes resultados mostram que $Q_i^m(x)$ é um spline polinomial de ordem m com nós em τ_1, \dots, τ_d de multiplicidades, respectivamente, ℓ_1, \dots, ℓ_d .

3. Seja $m \geq 2$ e $y_i < y_{i+m}$ então

$$Q_i^m(x) = \frac{(x-y_i)Q_i^{m-1}(x) + (y_{i+m}-x)Q_{i+1}^{m-1}(x)}{(y_{i+m}-y_i)}, \quad \text{para todo } x \text{ real}$$

Esta relação recursiva que relaciona B-splines de ordem m com B-splines de ordem $m-1$ proporciona algoritmos numericamente estáveis para a avaliação de B-splines e se estende para suas derivadas conforme veremos na próxima propriedade.

4. $D_+ Q_i^m(x) = \frac{(m-1)(Q_i^{m-1}(x) - Q_{i+1}^{m-1}(x))}{y_{i+m} - y_i}, \quad \text{para todo } x \text{ real}$

5. Se $m > 1$ e $y_i < y_{i+m}$ então

- $Q_i^m(x) > 0$ para $x \in (y_i, y_{i+m})$

- $Q_i^m(x) = 0$ para $x \notin (y_i, y_{i+m})$

- Seja α_i o número de repetições de y_i em $y_i \leq \dots \leq y_{i+m}$

$$Q_i^m(y_i) = D_+ Q_i^m(y_i) = \dots = D_+^{m-\alpha_i-1} Q_i^m(y_i) = 0$$

$$(-1)^{m+K-\alpha_i} D_+^K Q_i^m(y_i) > 0, K=m-\alpha_i, \dots, m-1$$

- Seja β_{i+m} o número de repetições de y_{i+m} em $y_i \leq \dots \leq y_{i+m}$

$$Q_i^m(y_{i+m}) = D_+ Q_i^m(y_{i+m}) = \dots = D_+^{m-\beta_{i+m}-1} Q_i^m(y_{i+m}) = 0$$

$$(-1)^{m-\beta_{i+m}} D_+^K Q_i^m(y_{i+m}) > 0, K=m-\beta_{i+m}, \dots, m-1$$

6. Se $y_\ell < y_{\ell+1}$ então os B-splines $\{Q_i^m\}_{i=\ell+1-m}^\ell$ geram o espaço dos polinômios de ordem m , \mathcal{P}_m , sobre $[y_\ell, y_{\ell+1})$.

Com maior generalidade: se $\ell < r$ e $y_{r-1} < y_r$ então $\{Q_i^m\}_{i=\ell-m+1}^{r-1}$ são linearmente independentes sobre $[y_\ell, y_r)$.

2.9 B-SPLINES NORMALIZADOS

As propriedades anteriores mostram que $Q_i^m(x)$ é um spline polinomial de ordem m com nós em $[y_i, \dots, y_{i+m})$. Entretanto, sua magnitude varia amplamente dependendo da localização dos nós. Por exemplo, em $[y_i, y_{i+1})$, $Q_i^1 = 1/(y_{i+1} - y_i)$ pode ser um número extremamente grande ou extremamente pequeno, dependendo do espaçamento entre os y 's. Do ponto de vista computacional é indesejável manipular funções que assumem valores muito pequenos ou muito grandes. A fim de evitar este tipo de ocorrência introduzimos um fator de normalização nos B-splines, definindo os B-splines normalizados, indicados por $N_i^m(x)$.

B-splines normalizados associados com os nós y_i, \dots, y_{i+m}

Seja $y_i < y_{i+m}$ e $y_i \leq \dots \leq y_{i+m}$

Por definição, $N_i^m(x) = (y_{i+m} - y_i) Q_i^m(x) = (-1)^m (y_{i+m} - y_i) [y_i, \dots, y_{i+m}]_+(x-y)^{m-1}$

Propriedades

$$1. \sum_{i=j+1-m}^j N_i^m(x) = 1 \text{ para todo } x \in [y_j, y_{j+1})$$

Isto é, os B-splines normalizados formam uma partição da unidade e para $m \geq 1$, $0 \leq N_i^m(x) \leq 1$ para todo x real.

$$2. N_i^m(x) = (x - y_i) Q_i^{m-1}(x) + (y_{i+m} - x) Q_{i+1}^{m-1}(x) \text{ para } x \in [a, b]$$

3. Seja $\bar{\Delta} = \{y_i\}_1^{n+m}$, $n = m + \sum_{i=1}^d m_i$, uma partição estendida associada

com $\mathcal{S}(\mathcal{P}_m; m; \Delta)$. Teremos:

$\{N_i\}_1^n$ é uma base de $S(\mathcal{P}_m; m; \Delta)$

$$N_i^m(x) > 0 \text{ para } x \in (y_i, y_{i+m})$$

$$N_i^m(x) = 0 \text{ para } x \notin (y_i, y_{i+m})$$

$$\sum_{i=1}^m N_i^m(x) = 1 \text{ para } x \in [a, b]$$

3. ALGORITMOS

3.1 ARMAZENAMENTO E AVALIAÇÃO DE UM SPLINE

Seja $s \in S(\mathcal{P}_m; m; \Delta)$ e $\{N_i^m\}_1^n$ B-splines normalizados associados com uma partição estendida $\bar{\Delta} = \{y_i\}_1^{m+n}$. Então $s = \sum_{i=1}^n c_i N_i^m(x)$, $\forall x \in [y_m, y_{n+1})$ sendo c_1, \dots, c_n determinados de modo único. Esta combinação linear chama-se expansão B-splines de s .

Devido a unicidade de representação de um spline s através de seus coeficientes numa expansão B-spline, para armazenar s num computador basta armazenar o vetor de coeficientes $c = (c_1, \dots, c_n)$

3.1.1 AVALIAÇÃO DE UM SPLINE s ATRAVÉS DE SUA EXPANSÃO EM B-SPLINES - I

Devido às propriedades locais de suporte dos B-splines, para computar $s(x)$ necessitamos computar apenas uma soma envolvendo m dos n B-splines da base.

$$\text{Seja } s \in S(\mathcal{P}_m; m; \Delta) \text{ e } s = \sum_{i=1}^n c_i N_i^m(x)$$

$$\text{Se } x \in [y_\ell, y_{\ell+1}) \text{ sendo } m \leq \ell < n \text{ então } s(x) = \sum_{i=\ell+1-m}^{\ell} c_i N_i^m(x)$$

Os valores dos B-splines para computar $s(x)$ podem ser obtidos através da geração da seguinte matriz triangular:

$$\begin{matrix}
 & & & & Q_{\ell}^1(x) \\
 & & & & \vdots \\
 & & & Q_{\ell-1}^2(x) & Q_{\ell}^2(x) \\
 & & & \vdots & \vdots \\
 & & Q_{\ell+2}^{m-1}(x) & \cdots & Q_{\ell-1}^{m-1}(x) & Q_{\ell}^{m-1}(x) \\
 N_{\ell+1}^m(x) & N_{\ell+2}^m(x) & \cdots & N_{\ell-1}^m(x) & N_{\ell}^m(x)
 \end{matrix}$$

Os valores desta matriz são computadas de forma eficiente e estável a partir de $Q_{\ell}^1(x) = \frac{1}{y_{\ell+1} - y_{\ell}}$ e da relação recursiva dos B-splines. A última linha obtem-se a partir de $N_m^i(x) = (x-y_i)Q_i^{m-1}(x) + (y_{i+m} - x)Q_{i+1}^{m-1}(x)$.

As vezes necessita-se calcular o valor de $s(y_{\ell+1}^-) = \lim_{x \uparrow y_{\ell+1}} s(x)$.

Se $y_{\ell+1}$ possui multiplicidade m , em geral $s(y_{\ell+1}^-) \neq s(y_{\ell+1}^+)$.

Na prática esta situação ocorre quando se utiliza um partição estendida com $y_1 = y_2 = \dots = y_m = a < b = y_{n+1} = \dots = y_{n+m}$ e se deseja o valor de $s(x)$ para $x = b$.

Se $s(x) = \sum_{i=1}^n c_i N_i^m(x)$ e $y_{\ell} < y_{\ell+1}$ então $s(y_{\ell+1}^-) = \sum_{i=\ell+1-m}^{\ell} c_i N_i^m(y_{\ell+1}^-)$

sendo os valores de $N_i^m(y_{\ell+1}^-)$ calculados através da matriz triangular gerada a partir de $x = y_{\ell+1}$.

Os valores da última linha da matriz triangular são calculados através do procedimento BSMQ (B-spline Matriz Q)

Dados x em $[a, b]$ e o spline s através de sua expansão em B-splines, a avaliação de $s(x)$ exige que antes se determine o intervalo $[y_{\ell}, y_{\ell+1})$. A pesquisa de ℓ tal que $y_{\ell} \leq x < y_{\ell+1}$ pode ser feita através do método da bissecção do intervalo original ao qual x pertença.

O valor de ℓ é obtido através do procedimento inteiro LBISEC. No caso em que se deseja construir uma tabela de $s(x)$ o procedimento anterior

pode ser otimizado uma vez que cada x considerado na tabela está próximo do anterior. Utilizando-se uma estimativa ℓx de ℓ e LBISEC construímos o procedimento inteiro INTERV que fornece mais rapidamente o valor de ℓ .

A avaliação de um spline $s \in \mathcal{S}(\mathcal{P}_m; m; \Delta)$, através de sua expansão em B-splines num ponto x de $[a, b]$ obtém-se através do procedimento BSX (B-Spline X).

3.1.2 AVALIAÇÃO DE UM SPLINE s ATRAVÉS DE SUA EXPANSÃO EM B-SPLINES-II.

Existe um algoritmo alternativo para avaliar $s(x)$ mais eficiente que o anterior e que dispensa os cálculos dos B-splines normalizados. Baseia-se na seguinte estratégia:

$s(x) = \sum_{i=1}^n c_i N_i^m(x) = \sum_{i=1}^n c_i^{[1]} \{ (x-y_i) Q_i^{m-1}(x) + (y_{i+m} - x) Q_{i+1}^{m-1}(x) \}$ utilizando-se $c_i = c_i^{[1]}$ e a relação recursiva para $N_i^m(x)$. Como $N_i^m(x) = (y_{i+m} - y_i) Q_i^m(x)$

segue-se

$$s(x) = \sum_{i=1}^n c_i^{[1]} \left\{ \frac{(x - y_i)}{y_{i+m-1} - y_i} N_i^{m-1}(x) + \frac{(y_{i+m} - x)}{y_{i+m} - y_{i+1}} N_{i+1}^{m-1}(x) \right\}$$

$$s(x) = \sum_{i=1}^n c_i^{[1]} \left[\frac{x - y_i}{y_{i+m-1} - y_i} N_i^{m-1}(x) + \sum_{i=1}^n c_i^{[1]} \left[\frac{y_{i+m} - x}{y_{i+m} - y_{i+1}} N_{i+1}^{m-1}(x) \right] \right]$$

$$s(x) = c_1^{[1]} \left[\frac{x - y_1}{y_m - y_1} N^{m-1}(x) + \sum_{i=2}^n c_i^{[1]} \left[\frac{x - y_i}{y_{i+m-1} - y_i} N_i^{m-1}(x) + \sum_{i=2}^n c_{i-1}^{[1]} \left[\frac{y_{i+m-1} - x}{y_{i+m-1} - y_i} N_i^{m-1}(x) + \right. \right. \right. \\ \left. \left. \left. + c_n^{[1]} \left[\frac{y_{n+m} - x}{y_{n+m} - y_{n+1}} N_{n+1}^{m-1}(x) \right] \right] \right]$$

$$s(x) = c_1^{[1]} \left[\frac{x - y_1}{y_m - y_1} N^{m-1}(x) + \sum_{i=2}^n \frac{(x-y_i)c_i^{[1]} + (y_{i+m-1}-x)c_{i-1}^{[1]}}{y_{i+m} - y_i} N_i^{m-1} + c_n^{[1]} \left[\frac{y_{n+m} - x}{y_{n+m} - y_{n+1}} N_{n+1}^{m-1}(x) \right] \right]$$

Introduzindo-se $c_0^{[1]} = c_{n+1}^{[1]} = 0$ e definindo-se

$$c_i^{[2]} = \frac{(x-y_i)c_i^{[1]} + (y_{i+m-1} - x)c_{i-1}^{[1]}}{y_{i+m-1} - y_i} \quad \text{ou } c_i^{[2]} = 0 \text{ se } y_{i+m-1} - y_i = 0$$

podemos escrever

$$s(x) = \sum_{i=1}^{n+1} c_i^{[2]} N_i^{m-2}.$$

Repetindo-se o mesmo tipo de transformação em relação aos B-splines normalizados obtemos genericamente para todo $1 \leq j \leq m$

$$s(x) = \sum_{i=1}^{n+j-1} c_i^{[j]}(x) N_i^{m-j+1}(x)$$

com os $c_i^{[j]}$ calculados recursivamente a partir de

$$c_i^{[1]} = c_i, \quad i = 1, 2, \dots, n$$

$$c_0^{[j]} = c_{n+j}^{[j]} = 0, \quad j = 1, 2, \dots, m$$

$$c_i^{[j+1]}(x) = \begin{cases} \frac{(x-y_i)c_i^{[j]}(x) + (y_{i+m-j} - x)c_{i-1}^{[j]}(x)}{y_{i+m-j} - y_i}, & \text{se } (y_{i+m-j} - y_i) \neq 0 \\ 0, & \text{em caso contrário} \end{cases}$$

para $i = 1, 2, \dots, n+j$ e $j = 1, 2, \dots, m-1$.

Em particular, se $y_\ell \leq x < y_{\ell+1}$ teremos:

$$s(x) = \sum_{i=\ell+1-m}^{\ell} c_i^{[1]} N_i^{m-1}(x) = \sum_{i=\ell-m}^{\ell} c_i^{[2]} N_i^{m-2}(x) = \dots = \sum_{i=\ell}^{\ell} c_i^{[m]} N_i^1(x) = c_\ell^{[m]} N_\ell^1(x) = c_\ell^{[m]}$$

pois $N_\ell^1 = (y_{\ell+1} - y_\ell) Q_\ell^1 = \frac{(y_{\ell+1} - y_\ell)}{(y_{\ell+1} - y_\ell)} = 1.$

O valor de $s(x) = c_\ell^{[m]}$ pode ser obtido pela geração da seguinte matriz:

$$\begin{array}{cccc}
 c_{\ell+1-m}^{[1]} & c_{\ell+2-m}^{[1]} & \dots & c_{\ell}^{[1]} \\
 & c_{\ell+2-m}^{[2]} & \dots & c_{\ell}^{[2]} \\
 & & \dots & c_{\ell}^{[m]}
 \end{array}$$

Se a matriz for gerada a partir de $x = y_{\ell+1}$ obtem-se

$$s(y_{\ell+1} -) = \lim_{x \rightarrow y_{\ell+1}} s(x) = c_{\ell}^{[m]}$$

O valor de $s(x)$ através deste algoritmo é calculado por meio do procedimento BSX2.

3.1.3 - PARTIÇÃO SIMPLES E PARTIÇÃO ESTENDIDA

A avaliação de um spline pertencente a $\mathcal{S}(\mathcal{P}_m; \mathcal{m}; \Delta)$ supõe como dados do problema, além do vetor dos coeficientes, a identificação de \mathcal{S} . Essa identificação é dada por m =ordem dos polinômios por parte, pelo vetor de multiplicidade $\mathcal{m} = (\ell_1, \ell_2, \dots, \ell_k)$, pelos nós da partição $\Delta = \{x_i\}_{i=1}^k$ e pelos extremos a e b do intervalo I subdividido por Δ .

Pode-se também identificar \mathcal{S} por m =ordem dos polinômios por parte, n =dimensão de \mathcal{S} e pelos nós $\{y_i\}_{i=1}^{m+n}$ da partição estendida $\bar{\Delta}$ associada a $\mathcal{S}(\mathcal{P}_m; \mathcal{m}; \Delta)$.

Recordemos que

$$\Delta = \{x_i\}_{i=1}^k \text{ com } a < x_1 < \dots < x_k < b$$

$$\mathcal{m} = (\ell_1, \dots, \ell_k), \text{ sendo } 1 \leq \ell_i \leq m, \quad i=1, 2, \dots, k$$

$$n = m + \sum_{i=1}^k \ell_i$$

$$\bar{\Delta} = \{y_i\}_{i=1}^{m+n} \text{ satisfazendo a}$$

$$y_1 \leq y_2 \leq \dots \leq y_m = a$$

$$\underbrace{y_{m+1} \leq \dots \leq y_{m+\ell_1}}_{\ell_1} < \underbrace{y_{m+\ell_1+1} \leq \dots \leq y_{m+\ell_1+\ell_2}}_{\ell_2} < \dots < \underbrace{y_{m+\sum_{i=1}^{k-1} \ell_i} \leq \dots \leq y_{m+\sum_{i=1}^k \ell_i}}_{\ell_k} < b$$

$$b = y_{n+1} \leq y_{n+2} \leq \dots \leq y_{n+m}$$

A fim de facilitar a conversão de um para outro grupo de dados desenvolvemos os procedimentos PTE (Partição Estendida) e INVPTTE (Inverte Partição Estendida) que, respectivamente, definem $\tilde{\Delta}$ a partir de Δ e vice-versa:

PTE(m,n,y,k,a,b,l,x)

Entra: m,k,a,b,{ ℓ_i }₁^k, { x_i }₁^k . Sai: n,{ y_i }₁^{m+n}

INVPTTE(m,n,y,k,a,b,l,x)

Entra: m,n,{ y_i }₁^{m+n} . Sai: k,a,b,{ ℓ_i }₁^k, { x_i }₁^k

Os respectivos algoritmos encontram-se descritos na parte B deste capítulo.

3.2 - DERIVADAS

Se $s \in \mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ então $D_+s(x)$ é uma função contínua à direita de x em R .

Alem disso $D_+s \in \mathcal{S}(\mathcal{P}_{m-1}; \mathcal{M}'; \Delta)$, sendo $\mathcal{M}' = (m'_1, \dots, m'_k)$ e $m'_i = \min(m-1, m_i)$, $i=1,2,\dots,k$.

A existência de $D_+s(x)$ para todo $x \notin \Delta$ decorre de ser $s(x)$ um polinômio por partes enquanto que em nós de Δ a própria definição de spline assegura a existência das derivadas pela direita e pela esquerda.

A continuidade de D_+s pela direita resulta do fato de ser s um polinômio em cada intervalo $[x_i, x_{i+1})$, $i=1,2,\dots,k$, fechado à esquerda.

É evidente que D_+s é também um polinômio por partes de ordem $m-1$. Se, por exemplo, $s, D_+s, \dots, D_+^r s$ são contínuas em x_i , então são contínuas $D_+s, D_+(D_+s), \dots, D_+^{r-1}(D_+s)$; resulta daqui que $m_i = \min(m-1, m_i)$, $i=1,2,\dots,k$.

Considerando-se a expansão de s numa base de B-splines, podemos escrever

$$s(x) = \sum_{i=1}^n c_i N_i^m(x),$$

onde $n = \text{dimensão de } \mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta) = m + \sum_{i=1}^k m_i$.

$$\text{Daí resulta } D_+s(x) = \sum_{i=1}^n c_i D_+ N_i^m(x).$$

$$\text{De } N_i^m(x) = (y_{i+m} - y_i)Q_i^m(x) \text{ e de } D_+Q_i^m(x) = (m-1) \frac{Q_i^{m-1}(x) - Q_{i+1}^{m-1}(x)}{y_{i+m} - y_i} \quad (3.2.1)$$

obtemos

$$D_+s(x) = \sum_{i=1}^n c_i (y_{i+m} - y_i) D_+Q_i^m(x) = \sum_{i=1}^n c_i (y_{i+m} - y_i) (m-1) \frac{[Q_i^{m-1}(x) - Q_{i+1}^{m-1}(x)]}{y_{i+m} - y_i}$$

$$D_+s(x) = \sum_{i=1}^n (m-1)c_i [Q_i^{m-1}(x) - Q_{i+1}^{m-1}(x)] = \sum_{i=1}^n (m-1)c_i Q_i^{m-1}(x) + \sum_{i=2}^{n+1} (m-1)c_{i-1} Q_i^{m-1}(x)$$

Tendo em vista que para $x \in [y_m, y_n)$

$$Q_1^{m-1}(x) = (-1)^{m-1} [y_1, \dots, y_m] (x-y)_+^{m-2} = (-1)^{m-1} [y_1, \dots, y_m] (x-y)^{m-2} \equiv 0$$

$$Q_{n+1}^{m-1}(x) = (-1)^{m-1} [y_{n+1}, \dots, y_{n+m}] (x-y)_+^{m-2} = (-1)^{m-1} [y'_{n+1}, \dots, y_{n+m}] (x-y)^{m-2} \equiv 0, \text{ vem}$$

$$D_+s(x) = \sum_{i=2}^n (m-1)(c_i - c_{i-1})Q_i^{m-1}(x) = \sum_{i=2}^n (m-1) \frac{(c_i - c_{i-1})}{y_{i+m-1} - y_i} (y_{i+m-1} - y_i)Q_i^{m-1}(x)$$

Definindo-se

$$c_i^{[2]} = \begin{cases} (m-1) \frac{c_i - c_{i-1}}{y_{i+m-1} - y_i} & \text{se } y_{i+m-1} - y_i \neq 0 \\ 0, & \text{em caso contrário} \end{cases}$$

$i=2,3,\dots,n$ e substituindo-se $(y_{i+m-1} - y_i)Q_i^{m-1}$ por N_i^{m-1} obtemos finalmente:

$$D_+s(x) = \sum_{i=2}^n c_i^{[2]} N_i^{m-1}(x) \quad (3.2.2)$$

Observações:

1. Se $\mathcal{M} = (m_1, m_2, \dots, m_k) = (1, 1, \dots, 1)$ então $\mathcal{M}' = \mathcal{M}$ e $\tilde{n} = n-1$, sendo \tilde{n} e n , respectivamente, as dimensões dos espaços $\mathcal{S}(\mathcal{P}_{m-1}; \mathcal{M}'; \Delta)$ e $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$.

Pela definição (3.2.2) observamos que nenhum $c_i^{[2]}$, $i=2,\dots,n$ é teoricamente nulo porque o denominador $y_{i+m-1} - y_i \neq 0$ para todo $i=2,\dots,n$. Como a dimensão de $\mathcal{S}(\mathcal{P}_{m-1}; \mathcal{M}'; \Delta)$ é $\tilde{n} = n-1$, concluímos que os B-splines $\{N_i^{m-1}\}_2^n$ constituem uma base de $\mathcal{S}(\mathcal{P}_{m-1}; \mathcal{M}'; \Delta)$,

uma vez que são linearmente independentes.

Na relação de recorrência (3.2.1) utilizada para obter (3.2.2) as funções B-splines $Q_i^{m-1}(x) = (-1)^{m-1} [y_i, \dots, y_{i+m-1}] (x-y)_+^{m-2}$ são definidas em nós $\{y_i\}_1^{m+n}$ da partição estendida associada a $\mathcal{S}(\mathcal{P}_m; m; \Delta)$.

Indicando $\{\tilde{y}_j\}_1^{m-1+\tilde{n}}$ os nós da partição estendida associada a $\mathcal{S}(\mathcal{P}_{m-1}; m'; \Delta)$ teremos as seguintes relações entre as duas partições:

$$(i) \quad \{y_i\}_1^{m+n} = y_1 \cup \{\tilde{y}_j\}_1^{m-1+\tilde{n}} \cup y_{m+n}$$

$$(ii) \quad \tilde{y}_j = y_{j+1}, \quad j=1, 2, \dots, n-1$$

Substituindo (ii) em $N_i^{m-1}(x) = (-1)^{m-1} (y_{i+m-1} - y_i) [y_i, \dots, y_{i+m-1}] (x-y)_+^{m-2}$

obtemos $N_i^{m-1}(x) = (-1)^{m-1} (\tilde{y}_{i+m-2} - \tilde{y}_{i-1}) [\tilde{y}_{i-1}, \dots, \tilde{y}_{i-1+m-1}] (x-y)_+^{m-2}(x)$, onde \tilde{N}_i^{m-1} indica os B-splines definidos sobre nós da partição estendida associada a $\mathcal{S}(\mathcal{P}_{m-1}; m'; \Delta)$.

Portanto, de (3.2.2) vem

$$D_+s(x) = \sum_{i=2}^n c_i^{[2]} N_i^{m-1}(x) = \sum_{i=2}^n c_i^{[2]} \tilde{N}_{i-1}^{m-1}(x) = \sum_{j=1}^{n-1} c_{j+1}^{[2]} \tilde{N}_j^{m-1}(x) = \sum_{j=1}^{\tilde{n}} \tilde{c}_j \tilde{N}_j^{m-1}(x) \text{ sendo } \tilde{c}_j = c_{j+1}^{[2]}, \quad j=1, 2, \dots, \tilde{n}.$$

2. Se $m = (m_1, m_2, \dots, m_k) = (m, m, \dots, m)$ então $m' = (m-1, m-1, \dots, m-1)$ e $\tilde{n} = n-1-k$, como decorrência das definições de m' e dimensão de espaços splines.

Aparentemente a definição de $D_+s(x)$ por (3.2.2) encerra uma contradição, pois expressa uma combinação linear de $n-1$ B-splines, que são linearmente independentes, num espaço de dimensão $\tilde{n} = n-1-k$.

Devemos mostrar que existem k coeficientes $c_i^{[2]}$ teoricamente nulos e que os B-splines dos quais são coeficientes, são B-splines identicamente nulos, restando assim em (3.2.2) uma combinação linear de no máximo $n-1-k$ coeficientes não nulos, os quais formarão então uma base de $\mathcal{S}(\mathcal{P}_{m-1}; m; \Delta)$.

Indicando por $\{\tilde{c}_i\}_1^{\tilde{n}}$ os primeiros \tilde{n} coeficientes não teoricamente nulos podemos escrever

$$D_+ s(x) = \sum_{i=1}^{\tilde{n}} \tilde{c}_i \tilde{N}_i^{m-1}(x) \tag{3.2.3}$$

onde $\tilde{N}_i^{m-1}(x)$, $i=1,2,\dots,\tilde{n}$ são os B-splines definidos sobre nós da partição estendida $\{\tilde{y}_i\}_1^{m-1+\tilde{n}}$ associada com $\mathcal{S}(\mathcal{P}_{m-1}; m'; \Delta)$.

De fato, o denominador de (3.2.2), $(y_{i+m-1} - y_i)$, anula-se k vezes, introduzindo, por definição, k coeficientes nulos. Como cada nó de Δ possui multiplicidade m , $(y_{i+m-1} - y_i) = 0$ para $i = m+1, 2m+1, \dots, km+1$, conforme ilustramos abaixo com os nós da partição estendida associada a

$$\mathcal{S}(\mathcal{P}_m; m; \Delta):$$

$$y_1 = y_2 = \dots = y_m = a$$

$$a < y_{m+1} = y_{m+2} = \dots = y_{2m} << y_{2m+1} = \dots = y_{3m} < \dots < y_{km+1} = \dots = y_{(k+1)m} < b$$

$$b = y_{n+1} = y_{n+2} = \dots = y_{n+m}$$

Estes k coeficientes teoricamente nulos correspondem aos B-splines normalizados $N_j^{m-1}(x) = (-1)^{m-1} (y_{j+m-1} - y_j) [y_j, y_{j+1}, \dots, y_{j+m-1}] (x-j)_+^{m-2}$, $j = m+1, 2m+1, \dots, km+1$. Na expressão da diferença dividida,

$$y_j = y_{j+1} = \dots = y_{j+m-1} \quad \text{e pelo parágrafo 2.6.3}$$

$$[y_j, \dots, y_{j+m-1}] (x-y)_+^{m-2} = \frac{D^{m-1} (x-y)_+^{m-2}}{(m-1)!} \equiv 0. \quad \text{Portanto}$$

$$N_j^{m-1}(x) \equiv 0, \quad j = m+1, 2m+1, \dots, km+1.$$

Uma demonstração rigorosa de (3.2.3) exigiria explicitar as mudanças de variáveis que mostrariam como $N_i^{m-1}(x)$ e $\tilde{N}_i^{m-1}(x)$ se encontram relacionados, conforme foi feito na observação 1.

3. No caso de nós de Δ com multiplicidades arbitrárias, $1 \leq m_i \leq m$, $i=1,2,\dots,k$, ocorrem situações intermediárias às que correspondem a $\mathcal{M} = (1,1,\dots,1)$ e $\mathcal{M} = (m,m,\dots,m)$. A conclusão anterior é estendida para estes casos: "Os coeficientes $\{\tilde{c}_i\}_1^{\tilde{n}}$ da expansão em B-splines de $D_+^j s(x) \in \mathcal{S}(\mathcal{P}_{m-1}; \mathcal{M}; \Delta)$ são obtidos respectivamente como os \tilde{n} primeiros coeficientes $\{c_i^{[2]}\}_2^{\tilde{n}}$ não teoricamente nulos". Esta estratégia será utilizada para definir o procedimento BSCDJ, conforme veremos no parágrafo 3.2.1..

Definição 2.

$$D_+^{j-1} s(x) = \sum_{i=j}^{\tilde{n}} c_i^{[j]} N_i^{m-j+1}(x), \text{ onde } c_i^{[1]} = c_i, i=1,2,\dots,n \text{ e}$$

$$c_i^{[j]} = \begin{cases} (m-j+1) \frac{c_i^{[j-1]} - c_{i-1}^{[j-1]}}{y_{i+m-j+1} - y_i} & \text{se } y_{i+m-j+1} - y_i \neq 0 \\ 0, & \text{em caso contrário} \end{cases} \quad (3.2.2)$$

para $j=2,\dots,m$ e $i=j,\dots,n$.

Utilizando-se a notação

$$D_+^{j-1} s(x) = \sum_{i=j}^{\tilde{n}} cd(j,i) N_i^{m-j+1}(x), \quad a \leq x \leq b$$

facilmente se computa a matriz $\{cd(j,i)\}_{j=1}^m, i=j: \tilde{n}$:

$$\begin{matrix} cd(1,1) & cd(1,2) & \dots & cd(1, n-1) & cd(1,n) \\ & cd(2,2) & \dots & cd(2, n-1) & cd(2,n) \\ & & \dots & & \\ & & & cd(m, n-1) & cd(m,n) \end{matrix}$$

A j -ésima linha dessa matriz contém os coeficientes da expansão em B-splines de $D_+^{j-1} s(x)$, porém referidos a expansão estendida associada a $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$.

Para avaliar $D_+^{j-1} s(x)$, $x \in [a, b]$ basta executar o procedimento BSX2(\tilde{m} , n , y , \tilde{c} , ℓx , ℓx) definindo-se, respectivamente, $\tilde{m} = m-j+1$ e $\tilde{c}_i = cd(j, i)$, $i=j, \dots, n$.

O procedimento BSMCD (B-Spline Matriz CD) calcula a matriz CD através do algoritmo (3.2.2).

3.2.1 DETERMINAÇÃO DOS COEFICIENTES DE $D_+^{j-1} s(x) \in \mathcal{S}(\mathcal{P}_{m-j+1}; m'; \Delta)$

O procedimento BSCDJ($m, n, y, c, j, mtil, ntil, ctil$) determina os coeficientes $\{ctil(i)\}_{i=1}^{ntil}$ da expansão em B-splines da $D_+^{j-1} s(x)$, considerando-se as funções B-splines definidas em nós da partição estendida $\{ytil(i)\}_{i=1}^{mtil+ntil}$ associada com $\mathcal{S}(\mathcal{P}_{m-j+1}; m'; \Delta)$.

O algoritmo baseado nas observações 3 e 4 do parágrafo 3.2 apresenta as seguintes etapas principais:

- 1) Determinação da matriz $\{cd(j, i)\}_{j=1}^m, i=j}^n$ através do procedimento BSMCD. A j -ésima linha da matriz cd contém coeficientes da expansão em B-splines de $D_+^{j-1} s(x)$, com as funções B-splines definidas em nós da partição estendida associada com $\mathcal{S}(\mathcal{P}_m; m; \Delta)$.
- 2) Determinação da partição simples Δ por meio do procedimento INVPT. Obtemos $k, a, b, \{x_i\}_1^k$ e $\{\ell m_i\}_1^k$.
- 3) Definição do vetor de multiplicidade m' utilizado em $\mathcal{S}(\mathcal{P}_{m-j+1}; m'; \Delta)$. Construimos $\{n \ell m_i\}_1^k$ tais que $n \ell m_i = \min(m-j+1, \ell m_i)$, $i=1, 2, \dots, k$.
- 4) Elimina de $\{cd(j, i)\}_{i=j}^n$ os coeficientes teoricamente nulos de tal modo que em $\{cd(j, i)\}_{i=j}^{ntil}$ fiquem armazenados respectivamente os primeiros $ntil$ coeficientes não teoricamente nulos, sendo $ntil$ a dimensão de $\mathcal{S}(\mathcal{P}_{m-j+1}; m'; \Delta)$.

5) Finalmente define $ctil(i-j+1) = cd(j,i), i=j, \dots, ntl, mtil=m-j+1$ e através de PTE, $ntil$ e $\{ytil_i\}_{i=1}^{ntil}$

A utilização de BSCDJ permite avaliar $D_+^{j-1}s(x)$, bem como, construir o procedimento BSMCD2($m,n,y,c,mtil,ntil,ytil,cd$), abaixo descrito, que define em $\{cd(j,i)\}_{j=1}^m, i=j, \dots, ntil(j)$ os coeficientes das expansões em B-splines das derivadas $D_+^{j-1}s(x)$, com as funções B-splines definidas em nós da partição estendida associada a $\mathcal{S}(\mathcal{P}_{m-j+1}; m'; \Delta)$.

```

procedimento BSMCD2(m,n,y,c,mtil,ntil,ytil,cd);
inteiro m,n,mtil; inteiro ntil[1:n-m];
real y[*], ytil[1:m+n], cd[1:m,1:n];
[ inteiro j, ntl ; real ctil[1:n];
para j de 1 passo 1 até m faça
[ BSCDJ(m,n,y,c,j,mtil,nt,ctil);
ntil[j] ← nt;
para i de 1 passo 1 até nt faça cd[j,i] ← ctil[i] ] ] ] ;

```

Apresentamos a seguir listagens de processamento de programas que testam o procedimento BSCDJ no cálculo de derivadas à direita de $s(x)$, comparando-o com outros algoritmos equivalentes no cálculo dessas derivadas, descritos nos parágrafos 3.2.3 e 3.2.4.

```

=====
PROCEDURE BSDJX(M,N,Y,C,J,MTIL,NTIL,YTIL,CC);
INTEGER M,N,J,MTIL,NTIL; REAL ARRAY YI(1:Y),YTILI(1:Y),CCI(1:Y),CI(1:Y);
=====
% UTILIZA BSMCD-PTE-INVPTI
=====
BEGIN
INTEGER ARRAY LMI(1:N-M), NLM(1:N-M); REAL ARRAY XI(1:N-M),CDI(1:M,1:N);

INTEGER K, I, MI, KD, NTIL, IK; REAL A,B;
BSMCD(M,N,Y,C,CD); INVPTI(M,N,Y,K,A,B,LH,X);
%
FOR I:=1 STEP 1 UNTIL K DO
BEGIN NLM(I):= LMI(I);
IF LMI(I) > M-J+1 THEN NLM(I):= M-J+1
END;
MI:= M; KD:= 0;
FOR I:= 1 STEP 1 UNTIL K DO
BEGIN KD:= KD + LMI(I) - NLM(I);
NTIL:= MI + NLM(I);
IF KD > 0 THEN
FOR IK:= MI+1 STEP 1 UNTIL NTIL DO CDI(J,IK):= CDI(J,IK+KD);
MI:=NTIL
END;
FOR I:= J STEP 1 UNTIL NTIL DO CCI(I-J+1):=CDI(J,I);
NTIL:=M-J+1; PTE(NTIL,NTIL,YTIL,K,A,B,NLM,X)
END;
=====
REAL PROCEDURE BSDX(M,N,Y,C,J,X);
INTEGER M,N,J; REAL X; REAL ARRAY YI(1:Y), CI(1:Y);
=====
% UTILIZA BSX2(INTERV(LRISEC)),BSDJ(BSMCD,PTE,INVPTI)
=====
BEGIN
INTEGER MTIL,NTIL; REAL ARRAY YTILI(1:N+M),CTILI(1:N); INTEGER LTIL;
BSDJX(M,N,Y,C,J,MTIL,NTIL,YTIL,CTIL); LTIL:=(MTIL+NTIL)/2;
BSDX:= BSX2(MTIL,NTIL,YTIL,CTIL,X,LTIL)
END;
=====
%-----TESTE DOS PROCEDIMENTOS BSDJX, BSDX, BSDX2 E RPPSDX-----
% UTILIZA BSRPP-BSMCD-INTERV-LRISEC-BSX2-INVPTI-BSM0
=====
%
BEGIN
INTEGER M,N,I,J,K; REAL A,B,Z;

INTEGER L;
FOR L:=1 STEP 1 UNTIL 7 DO BEGIN
READ(CR,/,/,M,K);
BEGIN
REAL ARRAY DX1(1:M),DX2(1:M),DX3(1:M); INTEGER ARRAY LMI(1:M);
REAL ARRAY YI(1:N+M),CI(1:N),XI(1:M),CPTC(N,1:M);
%
WRITE(IMP,SKIP(1));
WRITE(IMP,</,/,100(="">>);
WRITE(IMP,</,/"TESTE DOS PROCEDIMENTOS BSDJX, BSDX, BSDX2 E RPPSDX">>);
WRITE(IMP,</,/"UTILIZA BSRPP-INTERV-LRISEC-BSX2-INVPTI-BSM0">>);
READ(CR,/,/,FOR I:=1 STEP 1 UNTIL N+M DO YI(I));

```

```

READ(CR, //, FOR I:=1 STEP 1 UNTIL N DO C(I));
WRITECIMP, <"POLINOMIOS DE ORDEM M=", I3, X3, "S-SPLINES LINEARMENTE INDEPE
NDENTES, N=", I3, X3, M, N);
WRITECIMP, <//, "PARTICAO ESTENDIDA E COEFICIENTES DA EXPANSAO EM B-SPLIN
LS">);
WRITECIMP, <//, " I", *(I4)>, N+M, FOR I:=1 STEP 1 UNTIL N+M DO I);
WRITECIMP, <//, "Y(I)", *(F4.1)>, N+M, FOR I:=1 STEP 1 UNTIL N+M DO Y(I));
WRITECIMP, <"C(I)", *(F4.1)>, N, FOR I:=1 STEP 1 UNTIL N DO C(I));
WRITECIMP, <//, "COEFICIENTES DOS POLINOMIOS POR PARTES">);
BSRPP(M, N, Y, C, K, X, CP);
WRITECIMP, <//, " K ", *( " C(", I1, ")")>, M, FOR I:=0 STEP 1 UNTIL M-1 DO I)
; WRITECIMP(SPACE 11);
FOR I:=0 STEP 1 UNTIL K DO
    WRITECIMP, <I2, X1, *F7.2>, I, M, FOR J:=1 STEP 1 UNTIL M DO C(I, J));
INVTIC(M, N, Y, K, A, B, LM, X);
WRITECIMP, <//, "PARTICAO SIMPLES E VETOR DE MULTIPLICIDADE">);
WRITECIMP, <//, "EXTREMOS DO INTERVALO A=", F4.1, " B=", F4.1, X, A, B);
WRITECIMP, <//, " K", *(I4)>, K, FOR I:=1 STEP 1 UNTIL K DO I);
WRITECIMP, <//, "X(K)", *(F4.1)>, K, FOR I:=1 STEP 1 UNTIL K DO X(I));
WRITECIMP, <"L(K)", I3, *I4>, LM(I), K-1, FOR I:=2 STEP 1 UNTIL K DO L(I));
WRITECIMP(SKIP 11);
WRITECIMP, <//, "AVALIACAO DA (J-1)-ESIMA DERIVADA DO SPLINE S EM X">);
WRITECIMP, <//, " X J BSDJX BSDX BSDX2 RPP
DX">);
WRITECIMP, <59(" *")>);
Z:=Y(I);
DO BEGIN FOR J:=1 STEP 1 UNTIL M DO DX(I, J):=BSDJX(M, N, Y, C, J, Z);
    BSDX(M, N, Y, C, Z, DX1); BSDX2(M, N, Y, C, Z, DX2); RPPSDX(K, N, X, CP, Z, DX
3);
    WRITECIMP(SPACE 11);
    FOR I:=1 STEP 1 UNTIL M DO
        WRITECIMP, <F4.1, I3, 4E13.4>, Z, I, DX(I), DX1(I), DX2(I), DX3(I)
;
        Z:=Z+0.5 END
UNTIL Z > Y(N+1)
END; END; END;

```

=====

END.

=====

TESTE DOS PROCEDIMENTOS BSDJX, BSDX, BSDX2 E RPPSDX

UTILIZA BSRPP-INTERV-LBISEC-BSX2-INVPT-BSM0

POLINOMIOS DE ORDEN M= 4 B-SPLINES LINEARMENTE INDEPENDENTES, N= 14

PARTICAO ESTENDIDA E COEFICIENTES DA EXPANSAO EM B-SPLINES

I	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Y(I)	0.0	0.0	0.0	0.0	1.0	3.0	3.0	4.0	4.0	4.0	7.0	7.0	7.0	7.0	8.0	8.0	8.0	8.0
C(I)	1.0	3.0	2.0	5.0	7.0	5.0	3.0	2.0	1.0	4.0	5.0	7.0	3.0	2.0				

COEFICIENTES DOS POLINOMIOS POR PARTES

K	C(0)	C(1)	C(2)	C(3)
0	2.78	0.33	1.33	2.78
1	2.78	0.33	1.33	-0.31
2	6.33	2.00	-5.00	-0.33
3	3.00	-1.00	0.00	0.15
4	5.00	6.00	-18.00	9.00

PARTICAO SIMPLES E VETOR DE MULTIPLICIDADE

EXTREMOS DO INTERVALO A= 0.0 B= .8.0

K	1	2	3	4
X(K)	1.0	3.0	4.0	7.0
L(K)	1	2	3	4

AVALIACAO DA (J-1)-ESIMA DERIVADA DO SPLINE S EM X

X	J	BSDJX	BSDX	BSDX2	RPPDX
0.0	1	1.0000E+00	1.0000E+00	1.0000E+00	1.0000E+00
0.0	2	6.0000E+00	6.0000E+00	6.0000E+00	6.0000E+00
0.0	3	-1.4000E+01	-1.4000E+01	-1.4000E+01	-1.4000E+01
0.0	4	1.6667E+01	1.6667E+01	1.6667E+01	1.6667E+01
0.5	1	2.5972E+00	2.5972E+00	2.5972E+00	2.5972E+00
0.5	2	1.0833E+00	1.0833E+00	1.0833E+00	1.0833E+00
0.5	3	-5.6667E+00	-5.6667E+00	-5.6667E+00	-5.6667E+00
0.5	4	1.6667E+01	1.6667E+01	1.6667E+01	1.6667E+01
1.0	1	2.7778E+00	2.7778E+00	2.7778E+00	2.7778E+00
1.0	2	3.3333E-01	3.3333E-01	3.3333E-01	3.3333E-01
1.0	3	2.6667E+00	2.6667E+00	2.6667E+00	2.6667E+00
1.0	4	-1.8333E+00	-1.8333E+00	-1.8333E+00	-1.8333E+00
1.5	1	3.2396E+00	3.2396E+00	3.2396E+00	3.2396E+00
1.5	2	1.4375E+00	1.4375E+00	1.4375E+00	1.4375E+00
1.5	3	1.7500E+00	1.7500E+00	1.7500E+00	1.7500E+00
1.5	4	-1.8333E+00	-1.8333E+00	-1.8333E+00	-1.8333E+00
2.0	1	4.1389E+00	4.1389E+00	4.1389E+00	4.1389E+00
2.0	2	2.0833E+00	2.0833E+00	2.0833E+00	2.0833E+00
2.0	3	8.3333E-01	8.3333E-01	8.3333E-01	8.3333E-01
2.0	4	-1.8333E+00	-1.8333E+00	-1.8333E+00	-1.8333E+00
2.5	1	5.2465E+00	5.2465E+00	5.2465E+00	5.2465E+00
2.5	2	2.2708E+00	2.2708E+00	2.2708E+00	2.2708E+00
2.5	3	-8.3333E-02	-8.3333E-02	-8.3333E-02	-8.3333E-02
2.5	4	-1.8333E+00	-1.8333E+00	-1.8333E+00	-1.8333E+00
3.0	1	6.3333E+00	6.3333E+00	6.3333E+00	6.3333E+00
3.0	2	2.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00
3.0	3	-1.0000E+01	-1.0000E+01	-1.0000E+01	-1.0000E+01
3.0	4	-2.0000E+00	-2.0000E+00	-2.0000E+00	-2.0000E+00
3.5	1	6.0417E+00	6.0417E+00	6.0417E+00	6.0417E+00
3.5	2	-3.2500E+00	-3.2500E+00	-3.2500E+00	-3.2500E+00
3.5	3	-1.1000E+01	-1.1000E+01	-1.1000E+01	-1.1000E+01
3.5	4	-2.0000E+00	-2.0000E+00	-2.0000E+00	-2.0000E+00
4.0	1	3.0000E+00	3.0000E+00	3.0000E+00	3.0000E+00
4.0	2	-1.0000E+00	-1.0000E+00	-1.0000E+00	-1.0000E+00
4.0	3	0.	0.	0.	0.
4.0	4	8.8889E-01	8.8889E-01	8.8889E-01	8.8889E-01
4.5	1	2.5185E+00	2.5185E+00	2.5185E+00	2.5185E+00
4.5	2	-8.8889E-01	-8.8889E-01	-8.8889E-01	-8.8889E-01
4.5	3	4.4444E-01	4.4444E-01	4.4444E-01	4.4444E-01
4.5	4	8.8889E-01	8.8889E-01	8.8889E-01	8.8889E-01
5.0	1	2.1481E+00	2.1481E+00	2.1481E+00	2.1481E+00
5.0	2	-5.5556E-01	-5.5556E-01	-5.5556E-01	-5.5556E-01
5.0	3	8.8889E-01	8.8889E-01	8.8889E-01	8.8889E-01

5.0	4	8.8889E-01	8.8889E-01	8.8889E-01	8.8889E-01
5.5	1	2.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00
5.5	2	0.	0.	0.	1.8190E-12
5.5	3	1.3333E+00	1.3333E+00	1.3333E+00	1.3333E+00
5.5	4	8.8889E-01	8.8889E-01	8.8889E-01	8.8889E-01
6.0	1	2.1852E+00	2.1852E+00	2.1852E+00	2.1852E+00
6.0	2	7.7778E-01	7.7778E-01	7.7778E-01	7.7778E-01
6.0	3	1.7778E+00	1.7778E+00	1.7778E+00	1.7778E+00
6.0	4	8.8889E-01	8.8889E-01	8.8889E-01	8.8889E-01
6.5	1	2.8148E+00	2.8148E+00	2.8148E+00	2.8148E+00
6.5	2	1.7778E+00	1.7778E+00	1.7778E+00	1.7778E+00
6.5	3	2.2222E+00	2.2222E+00	2.2222E+00	2.2222E+00
6.5	4	8.8889E-01	8.8889E-01	8.8889E-01	8.8889E-01
7.0	1	5.0000E+00	5.0000E+00	5.0000E+00	5.0000E+00
7.0	2	6.0000E+00	6.0000E+00	6.0000E+00	6.0000E+00
7.0	3	-3.6000E+01	-3.6000E+01	-3.6000E+01	-3.6000E+01
7.0	4	5.4000E+01	5.4000E+01	5.4000E+01	5.4000E+01
7.5	1	4.6250E+00	4.6250E+00	4.6250E+00	4.6250E+00
7.5	2	-5.2500E+00	-5.2500E+00	-5.2500E+00	-5.2500E+00
7.5	3	-9.0000E+00	-9.0000E+00	-9.0000E+00	-9.0000E+00
7.5	4	5.4000E+01	5.4000E+01	5.4000E+01	5.4000E+01
8.0	1	2.0000E+00	2.0000E+00	2.0000E+00	2.0000E+00
8.0	2	-3.0000E+00	-3.0000E+00	-3.0000E+00	-3.0000E+00
8.0	3	1.8000E+01	1.8000E+01	1.8000E+01	1.8000E+01
8.0	4	5.4000E+01	5.4000E+01	5.4000E+01	5.4000E+01

3.2.2 AVALIAÇÃO DE $D_+^{j-1}s(x)$ NUM PONTO x DE $[a,b]$, $j=1,\dots,m$

Suponhamos que $s \in \mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ seja conhecido através de sua expansão em B-splines, ou seja, que sejam conhecidos os coeficientes $\{c_i\}_1^n$ tais que $s(x) = \sum_{i=1}^n c_i N_i^m(x)$.

Recordemos que o espaço $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ fica determinado através de sua partição estendida $\{y_i\}_1^{m+n}$

Dado $x \in [a,b]$ o valor de $s(x)$ é obtido pela execução do procedimento tipo função $BSX2(m,n,y,c,x,\&x)$ (Vide parágrafo 3.1.2)

Como $D_+^{j-1}s(x) \in \mathcal{S}(\mathcal{P}_{m-j+1}; \mathcal{M}'; \Delta)$, sendo $\mathcal{M}' = (m'_1, \dots, m'_k)$ e $m'_i = \min(m-j+1, m)$, $i=1,2,\dots,k$, o procedimento listado no parágrafo anterior permite obter o valor de $D_+^{j-1}s$ num ponto $x \in [a,b]$, $j=1,2,\dots,m$, através de $BSCDJ$ e $BSX2$.

Entretanto, de acordo com a observação 4 do parágrafo 3.2 podemos utilizar diretamente os coeficientes da matriz CD gerada através do procedimento $BSMCD$, obtendo um algoritmo mais simples para a avaliação de $D_+^{j-1}s(x)$:

```

Função real BSDJX(m,n,y,c,j,x);
inteiro m,n,j; real x; real y[*], c[*];
[ inteiro i, mtil, l; real cd[1:m, 1:m], cc[1:n]; real s;
  BSMCD(m,n,y,c,cd);
  se x = y[n+1] então l ← n senão l ← INTERV(1,m+n,(m+n)/2,x,y);
  se j = m então s ← cd[m,l]
  senão [ para i de j passo 1 até n faça
          cc[i] ← cd[j,i];
          mtil ← m-j+1;
          s ← BSX2(mtil,n,y,cc,x,&x) ] ] ;
BSDJX ← s ]

```

Portanto, a avaliação de $D_+^{j-1}s(x)$ para j entre 1 e m obtém-se pela execução da função acima, isto é, de $BSDJX(\underline{B}\text{-spline } \underline{D}\text{erivada } \underline{J}\text{-ésima no ponto } \underline{X})$.

Na descrição do algoritmo fizemos uso do seguinte resultado: "se $j=m$ então $D_+^{j-1}s(x) = cd[m, \ell]$ sendo ℓ um índice tal que $y_\ell \leq x < y_{\ell+1}$ " cuja justificação apresentamos em seguida:

$$D_+^{j-1}s(x) = \sum_{i=\ell+1-\tilde{m}}^{\ell} cd(j,i) N_i^{\tilde{m}}(x) \quad \text{com } \tilde{m} = m-j+1 \text{ e } \ell \text{ tal que } y_\ell \leq x < y_{\ell+1}$$

$$\text{Para } j=m \text{ obtemos } \tilde{m}=1 \text{ e } D_+^{m-1}s(x) = \sum_{i=\ell}^{\ell} cd(m,i) N_i^1(x) = cd(m, \ell) N_\ell^1(x)$$

$$\text{Como } N_\ell^1(x) = \frac{(-1)(y_{\ell+1} - y_\ell)}{(y_\ell - y_{\ell+1})} = 1, \quad D_+^{m-1}s(x) = cd(m, \ell).$$

3.2.3 AVALIAÇÃO DE $s(x), Ds(x), \dots, D^{m-1}s(x) - I$

Dado $x \in [a, b]$ e $s \in \mathcal{S}(\mathcal{P}_m; m; \Delta)$ através de sua expansão em B-splines, os valores de $D^{j-1}s(x)$, $j=1, 2, \dots, m$ são obtidos pela aplicação do mesmo algoritmo utilizado em BSDJX descrito no parágrafo anterior.

O procedimento BSDX2 (B-Spline Derivadas no ponto X) implementa esse procedimento e calcula $s(x), Ds(x), \dots, D^{m-1}s(x)$, armazenando-os, respectivamente, em $dx[1], \dots, dx[m]$.

3.2.4 AVALIAÇÃO DE $s(x), Ds(x), \dots, D^{m-1}s(x) - II$

Para avaliação das derivadas sucessivas de um spline $s \in \mathcal{S}(\mathcal{P}_m; m; \Delta)$ existe uma abordagem diferente da empregada em BSDX2, mais eficiente para $m \geq 6$.

$$\text{Vimo que } D_+^{j-1}s(x) = \sum_{i=1}^n cd(j,i) N_i^{m-j+1}(x), \quad j=1, \dots, m$$

$$\text{Para } x \in [y_\ell, y_{\ell+1}), \quad D_+^{j-1}s(x) = \sum_{i=\ell-(m-j+1)+1}^{\ell} cd(j,i) N_i^{m-j+1}(x), \quad j=1, \dots, m$$

Descrivendo-se $D_+^{m-1}s(x), \dots, s(x)$ em vez de $s(x), \dots, D_+^{m-1}s(x)$, isto é, ordenando-se a soma pelo índice $m-j+1 = 1, 2, \dots, m$, obtemos:

$$D_+^{m-j} s(x) = \sum_{i=\ell-j+1}^{\ell} cd(m-j+1,i) N_i^j(x), \quad j=1,2,\dots,m$$

Como $N_i^j(x) = (y_{i+j} - y_i) Q_i^j(x)$ obtemos

$$D_+^{m-j} s(x) = \sum_{i=\ell-j+1}^{\ell} cd(m-j+1,i) (y_{i+j} - y_i) Q_i^j(x), \quad j=1,2,\dots,m \quad (3.2.4.1)$$

Os valores dos B-splines $\{Q_i^j\}_{j=1,i=\ell-j+1}^m$ estão todos contidos na matriz triangular Q gerada por meio do procedimento BSMQ, conforme vimos no parágrafo 3.1.1.

O procedimento BSDX(m,n,y,c,x,dx) calcula, através de (3.2.4.1), os valores de $s(x), Ds(x), \dots$ e $D^{m-1}s(x)$, armazenando-os respectivamente, em $dx[1], dx[2], \dots, dx[m]$.

3.3 INTEGRAIS

Inicialmente observamos que se $s \in \mathcal{S}(\mathcal{P}_m; m; \Delta)$ então sua integral indefinida $D_a^{-1}s(x) = \int_a^x s(t) dt \in \mathcal{S}(\mathcal{P}_{m+1}; m'; \Delta)$.

De fato, $D_a^{-1}s(x)$ é obviamente um polinômio por partes de ordem $m+1$. Por definição, em um nó x_i de Δ , $s, Ds, \dots, D^{m-1-m_i}s$ são contínuas em x_i . Portanto, $D(D_a^{-1}s), D^2(D_a^{-1}s), \dots, D^{m-m_i}(D_a^{-1}s)$ são contínuas em x_i . Como $D_a^{-1}s$ é contínua em x_i vemos que $D^j(D_a^{-1}s), j=0,1,\dots,m-m_i$ são contínuas em x_i . Consequentemente, em $\mathcal{S}(\mathcal{P}_{m+1}; m; \Delta)$, a multiplicidade m_i de x_i se mantém, pois a ordem da maior derivada se expressa por $(m+1)-1-m_i$. Concluimos que $D_a^{-1}s \in \mathcal{S}(\mathcal{P}_{m+1}; m; \Delta)$ e que a dimensão do espaço é $n+1$.

Suponhamos que $s \in \mathcal{S}(\mathcal{P}_m; m; \Delta)$ seja conhecido através de sua expansão em B-splines, isto é, são conhecidos os coeficientes $\{c_i\}_1^n$ tais que $s(x) = \sum_{i=1}^n c_i N_i^m(x)$. Consideramos o espaço $\mathcal{S}(\mathcal{P}_m; m; \Delta)$ determinado por m =ordem dos polinômios por partes, n =sua dimensão e $\{y_i\}_1^{m+n}$ nós da perspectiva partição estendida.

Como $D_{y_1}^{-1}s(x) = \int_{y_1}^x s(t)dt \in \mathcal{S}(\mathcal{P}_{m+1}; \mathcal{M}^{\Delta}; \Delta)$ existem coeficientes $\{c_i^{-1}\}_0^n$ de sua expansão em B-splines, considerando-se os B-splines definidos sobre os nós da partição estendida associada a $\mathcal{S}(\mathcal{P}_{m+1}; \mathcal{M}; \Delta)$, a qual podemos representar por $y_0 \cup \{y_i\}_1^{m+n} \cup y_{m+n+1}$, com $y_0 \leq y_1$ e $y_{m+n} \leq y_{m+n+1}$.

Portanto, para todo $x \in [a, b)$ ou seja $x \in [y_m, y_{n+1})$, podemos escrever:

$$D_{y_1}^{-1}s(x) = \sum_{i=0}^n c_i^{-1} N_i^{m+1}(x)$$

No ponto $x = y_1$, $D_{y_1}^{-1}s(y_1) \equiv 0$.

$$\text{Portanto, } c_0 N_0^{m+1}(y_1) + c_1 N_1^{m+1}(y_1) + \dots + c_n N_n^{m+1}(y_1) = 0 \quad (3.3.1)$$

Tendo em vista que $N_i^{m+1}(x) = (-1)^{m+1} (y_{i+m+1} - y_i) [y_i, \dots, y_{i+m+1}] (x - y)_+^m$ observamos que $N_i^{m+1}(y_1) = 0$ para $i=2, 3, \dots, n$ em virtude de y_1 não pertencer aos respectivos suportes. Pelas condições de continuidade, nos extremos do suporte, $N_i^{m+1}(x)$ também se anula; logo $N_1^{m+1}(y_1) = 0$.

Em (3.3.1) restou apenas a primeira parcela, isto é, $c_0 N_0^{m+1}(y_1) = 0$.

Impondo $y_0 < y_1$ resulta $N_0^{m+1}(y_1) > 0$ pois $y_1 \in (y_0, y_{m+1})$. Portanto $c_0^{-1} = 0$.

Tendo em vista que $D(D_y^{-1}s(x)) = s(x)$, os respectivos coeficientes estão relacionados entre si, conforme vimos no parágrafo 3.2, por

$$c_i = \begin{cases} m \frac{c_i^{-1} - c_{i-1}^{-1}}{y_{i+m} - y_i} & \text{se } y_{i+m} - y_i \neq 0 \\ 0, & \text{em caso contrário} \end{cases} \quad (3.3.2)$$

$$i = 1, 2, \dots, n$$

Observamos que $y_{i+m} - y_i \neq 0$ para $i=1, 2, \dots, n$ em virtude de se manter a multiplicidade de cada nó x_i e de $1 \leq m_i \leq m$.

Resolvendo (3.3.2) em relação a c_i vem $c_i^{-1} = \frac{(y_{i+m} - y_i)}{m} c_i$. Daqui segue:

$$c_1^{-1} = \frac{1}{m} (y_{m+1} - y_1) c_1 + c_0^{-1}, \quad c_0^{-1} = 0$$

$$c_2^{-1} = \frac{1}{m} (y_{m+2} - y_2) c_2 + c_1^{-1}$$

⋮

$$c_n^{-1} = \frac{1}{m} (y_{m+n} - y_n) c_n + c_{n-1}^{-1}$$

Por substituições sucessivas obtemos finalmente

$$c_i^{-1} = \frac{1}{m} \sum_{j=1}^i (y_{m+j} - y_j) c_j, \quad i=1,2,\dots,n \quad (3.3.3)$$

O procedimento BSMCI(B-Spline Matriz dos Coeficientes da Integral indefinida) fornece, a menos do fator m , os coeficientes $c_0^{-1}, c_1^{-1}, \dots, c_n^{-1}$, calculados de acordo com o algoritmo descrito em (3.3.3). As divisões por m de cada coeficiente foram omitidas para diminuir o esforço computacional de (3.3.3).

Em BSMCI(m, n, y, c, ci) os coeficientes $mc_0^{-1}, \dots, mc_n^{-1}$ são armazenados respectivamente em $ci[1], \dots, ci[n+1]$.

3.3.1 INTEGRAL DEFINIDA

O procedimento BSMCI(m, n, y, c, ci) fornece através de $\{ci_j\}_{j=1}^{n+1}$ os coeficientes da expansão em B-splines de $D_{y_1}^{-1} s(x) = \int_{y_1}^x s(t) dt \in$

$$\mathcal{S}(\mathcal{P}_m; m; \Delta).$$

Quando se utiliza a integral indefinida para cálculo da integral definida $ID = \int_a^d s(t) dt = \int_{y_1}^d s(t) dt - \int_{y_1}^c s(t) dt$, $a \leq c < d \leq b$, as constantes de integração se compensam e obtemos o valor correto de ID.

Como a condição $D_{y_1}^{-1} s(x) = s(x)$ foi utilizada para relacionar os coeficientes de $s(x) \in \mathcal{S}(\mathcal{P}_m; m; \Delta)$ e $D_{y_1}^{-1} s(x) \in \mathcal{S}(\mathcal{P}_{m+1}; m; \Delta)$,

os coeficientes de $s(x)$ podem ser obtidos a partir dos coeficientes de $D_{y_1}^{-1}s(x)$ através do procedimento BSMCD($\bar{m}, \bar{n}, \bar{y}, ci, cd$), ficando os mesmos armazenados em $\{cd[2, i]\}_{i=2}^{\bar{n}}$; \bar{m} , \bar{n} , e \bar{y} referem-se a identificação de $\mathcal{S}(\mathcal{P}_{m+1}; \mathcal{M}; \Delta)$.

Entretanto, como era de se esperar, a integral indefinida da derivada de $s(x)$ não reproduz $s(x)$; os coeficientes de $s(x)$ e $D_{y_1}^{-1}(Ds(x))$ são diferentes e $s(x)$ e $D_{y_1}^{-1}s(x)$ diferem por uma constante.

Exemplo

Seja $\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$ identificada por $m=2$, $n=4$ e $\{y_i\}_{i=1}^{m+n} = \{1, 1, 2, 3, 4, 4\}$.

Suponhamos s determinado por $s(x) = N_1^2(x) + 2N_2^2(x) + 3N_3^2(x) + 4N_4^2(x)$

(i) Cálculo de $D_{y_1}^{-1}s(x)$

Através de BSMCI obtemos $D_{y_1}^{-1}s(x) = 0N_1^3(x) + \frac{1}{2}N_2^3(x) + \frac{5}{2}N_3^3(x) + \frac{11}{2}N_4^3(x) + \frac{15}{2}N_5^3(x)$

Temos $m=3$ e $n=5$

(ii) Determinação de $D(D_{y_1}^{-1}s(x))$

Aplicando BSMCD determinamos os coeficientes de $D(D_{y_1}^{-1}s(x))$ e

$\mathcal{S}(\mathcal{P}_m; \mathcal{M}; \Delta)$

$$CD = \begin{bmatrix} 0 & 1/2 & 5/2 & 11/2 & 15/2 \\ - & 1 & 2 & 3 & 4 \\ - & - & x & x & x \end{bmatrix} \quad m=2 \quad e \quad n=4$$

$$D(D_{y_1}^{-1}s(x)) = N_1^2(x) + 2N_2^2(x) + 3N_3^2(x) + 4N_4^2(x)$$

(iii) Cálculo de $Ds(x)$

Por meio de BSMCD obtemos

$$CD = \begin{bmatrix} 1 & 2 & 3 & 4 \\ - & 1 & 1 & 1 \end{bmatrix}$$

Temos $m=1$, $n=3$ e $Ds(x) = N_1^1(x) + N_2^1(x) + N_3^1(x)$

(iv) *Determinação de $D_{y_1}^{-1}(Ds(x))$*

Aplicando-se BSMCI vem $\{ci_j\}_{j=1}^3 = \{0, 1, 2, 3\}$

Temos $m=2$, $n=4$ e $D_{y_1}^{-1}(Ds(x)) = 0N_1^2(x) + N_2^2(x) + 2N_3^2(x) + 3N_4^2(x)$

Note que $s(x) - D_{y_1}^{-1}(Ds(x)) = N_1^2(x) + N_2^2(x) + N_3^2(x) + N_4^2(x) = 1$ para todo x real.

Algoritmos BSID, BSID1, BSID2 para cálculo da integral definida

As etapas principais para cálculo de $ID = \int_c^d s(t)dt$ são as seguintes:

- 1) Cálculo dos coeficientes da expansão em B-splines de $\int_{y_1}^x s(t)dt$ através de BSMCI.
- 2) Determinação da partição estendida $\{\bar{y}_i\}_1^{m+n+2}$ associada com $S(\mathcal{P}_{m+1}; m; \Delta)$
- 3) Avaliação de $\int_{y_1}^d s(t)dt$ e $\int_{y_1}^c s(t)dt$ através de BSX2.
- 4) Cálculo de $ID = \frac{1}{m} \left[\int_{y_1}^d s(t)dt - \int_{y_1}^c s(t)dt \right]$

Apresentamos a seguir a listagem e teste de três algoritmos equivalentes codificadas em ALGOL. Por simplicidade implementamos BSID(m, n, y, c, ei, es) onde utilizamos os B-splines definidos em nós da partição estendida associado a $S(\mathcal{P}_m; m; \Delta)$, corrigindo, por mudança de variável, os coeficientes da expansão em B-splines da integral definida fornecidos através do procedimento BSMCI.

BEGIN

FILE CR(KIND=READER),IMP(KIND=PRINTER);

```

=====
REAL PROCEDURE BSID(M,N,Y,C,EI,ES);
INTEGER M,N; REAL ARRAY Y[*],CI[*]; REAL EI,ES;
=====

```

UTILIZA BSMCI,BSX2(INTERV(LBISEC))

```

=====
BEGIN
INTEGER LX; REAL IEI,IES; REAL ARRAY CII1:N+1];

```

```

BSMCI(M,N,Y,C,CI);
FOR LX:=1 STEP 1 UNTIL N DO CII(LX):=CII(LX+1); LX:=(M+N)/2;
IEI:=BSX2(M+1,N,Y,CI,EI,LX);
IES:=BSX2(M+1,N,Y,CI,ES,LX);
BSID:=(IES-IEI)/M
END;

```

```

=====
REAL PROCEDURE BSID2(M,N,Y,C,EI,ES);
INTEGER M,N; REAL ARRAY Y[*],CI[*]; REAL EI,ES;
=====

```

UTILIZA BSMCI,BSX2(INTERV(LBISEC))

```

=====
BEGIN
INTEGER I,MTIL,NTIL,LX; REAL IEI,IES; REAL ARRAY CII1:N+1];

```

```

REAL ARRAY YIIL1:N+M+2]; BSMCI(M,N,Y,C,CI); MTIL:=M+1; NTIL:=N+1;
FOR I:=N+M+1 STEP -1 UNTIL 2 DO YIIL[I]:=YIIL[I-1];
YIIL[1]:=Y[1]-1; YIIL(N+M+2):=YIIL(N+M+1)+1; LX:=(MTIL+NTIL)/2;
IEI:=BSX2(MTIL,NTIL,YIIL,CI,EI,LX); IES:=BSX2(MTIL,NTIL,YIIL,CI,ES,LX);
BSID2:=(IES-IEI)/M
END;

```

```

=====
REAL PROCEDURE BSID1(M,N,Y,C,EI,ES);
INTEGER M,N; REAL ARRAY Y[*],CI[*]; REAL EI,ES;
=====

```

UTILIZA BSMCI-INVPTI-PTE-BSX2(INTERV(LBISEC))

```

=====
BEGIN
INTEGER K,LX; INTEGER ARRAY LMI1:N-M]; REAL ARRAY XI1:N-M]; REAL A,B;

```

```

INTEGER NTL;
REAL ARRAY CII1:N+1], YIIL1:N+M+2]; REAL IEI,IES;
BSMCI(M,N,Y,C,CI); INVPTI(M,N,Y,K,A,B,LM,X);
PTE(M+1,NTL,YIIL,K,A,B,LM,X);
LX:=(M+N+2)/2; IEI:=BSX2(M+1,N+1,YIIL,CI,EI,LX);
IES:=BSX2(M+1,N+1,YIIL,CI,ES,LX);
BSID1:=(IES-IEI)/M
END;

```

=====

```

=====
X  TESTE BSID-BSID1-BSID2
X  UTILIZA BSMCI,BSX2,INTERV(1B1SEC),INVPT E PTE
=====
BEGIN
INTEGER M,N,I,K;          REAL EI,ES,INT,INT1,INT2,A,B;

READ(CR,/,/,M,N);
BEGIN
REAL ARRAY Y(1:N+M), C(1:N);

REAL ARRAY XI(1:N-M); INTEGER ARRAY L(1:N-M);
WRITE(IMP,<100("=")>);
WRITE(IMP,</,/,"TESTE DOS PROCEDIMENTOS BSID-BSID1-BSID2">);
WRITE(IMP,</,/,"UTILIZA BSMCI,BSX2,INTER(LB1SEC),INVPT E PTE">);
WRITE(IMP,</,/,"ENTRADA:M,N,Y,C">);
WRITE(IMP,</,/,"SAIDA: INTEGRAIS DEFINIDAS CALCULADAS ATRAVES">);
WRITE(IMP,<"          DOS PROCEDIMENTOS BSID, BSID1 E BSID2">);
WRITE(IMP,</,/,"PCLINOMICOS DE GRAU M=",I3>,M);
WRITE(IMP,<"B-SPLINES LINEARMENTE INDEPENDENTES,N=",I3>,N);
READ(CR,/,/,FOR I:=1 STEP 1 UNTIL N+M DO Y(I));
READ(CR,/,/,FOR I:=1 STEP 1 UNTIL N DO C(I));
WRITE(IMP,</,/," I ",*F4.0>,N+M,FOR I:=1 STEP 1 UNTIL N+M DO Y);
WRITE(IMP,<"Y(I)",*F4.0>,N+M,FOR I:=1 STEP 1 UNTIL N+M DO Y(I));
WRITE(IMP,<"C(I)",*F4.0>,N,FOR I:=1 STEP 1 UNTIL N DO C(I));
WRITE(IMP,</,/,"INTEGRAIS DEFINIDAS DE EI A ES">);
WRITE(IMP,</,X4,"EI",X4,"ES",X6,"BSID",X10,"BSID1",X9,"BSID2">);
WRITE(IMP,SPACE(1)); INVPT(M,N,Y,K,A,B,LM,X);
EI:=X(1); ES:=EI; I:=2;
DO BEGIN INT:=BSID(M,N,Y,C,EI,ES); INT1:=BSID1(M,N,Y,C,EI,ES);
          INT2:=BSID2(M,N,Y,C,EI,ES);
          WRITE(IMP,<2F6.1,3E14.5>,EI,ES,INT,INT1,INT2);
          EI:=ES; ES:=X(I); I:=I+1
        END UNTIL I > K ;
EI:=X(I); ES:=X(K); INT:=BSID(M,N,Y,C,EI,ES);
          INT1:=BSID1(M,N,Y,C,EI,ES);
          INT2:=BSID2(M,N,Y,C,EI,ES);
WRITE(IMP,<2F6.1,3E14.5>,EI,ES,INT,INT1,INT2);
END;

END;
=====
END.

```

=====

TESTE DOS PROCEDIMENTOS BSID-ESID1-BSID2

UTILIZA BSMCI,BSX2,INTER(LBISEC),INVPTE E PTE

ENTRADA=M,N,Y,C

SAIDA: INTEGRAIS DEFINIDAS CALCULADAS ATRAVES
DOS PROCEDIMENTOS BSID, BSID1 E BSID2

POLINOMIOS DE GRAU M= 4

B-SPLINES LINEARMENTE INDEPENDENTES,N= 12

I	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
Y(I)	0.	0.	0.	0.	2.	2.	3.	3.	3.	5.	6.	6.	9.	9.	9.	9.
C(I)	3.	5.	1.	4.	7.	6.	3.	5.	7.	8.	9.	5.				

INTEGRAIS DEFINIDAS DE EI A ES

EI	ES	BSID	BSID1	BSID2
2.0	2.0	0.	0.	0.
2.0	3.0	5.00000E+00	5.00000E+00	5.00000E+00
3.0	5.0	9.50000E+00	9.50000E+00	9.50000E+00
2.0	6.0	2.10625E+01	2.10625E+01	2.10625E+01

3.3.2 PRODUTOS INTERNOS DE B-SPLINES NORMALIZADOS

Faz-se uso da fórmula de integração numérica de Gauss que permite o cálculo sem erro de truncamento.

3.3.3 FÓRMULA DE GAUSS

$\int_{-1}^1 f(t) dt \sim \sum_{j=1}^m w_j f(z_j)$, onde $\{z_i\}_1^m$, $\{w_i\}_1^m$ e f satisfazem às seguintes condições:

- (i) $-1 < z_1 < z_2 < \dots < z_m < 1$; os pontos z_i são simétricos em relação a origem.
- (ii) w_1, w_2, \dots, w_m são números positivos, chamados pesos, também são simétricos.
- (iii) Se $f \in \mathcal{P}_{2m}$ então $\int_{-1}^1 f(t) dt = \sum_{j=1}^m w_j f(z_j)$

Observações

- (i) Se f está definida em $a \leq x \leq b$ efetua-se antes a mudança de variável linear $\phi : [a, b] \rightarrow [-1, 1]$, $x \mapsto \left(\frac{b-a}{2}\right)(t+1) + \left(\frac{b+a}{2}\right)$ e depois aplica-se a fórmula de Gauss:

$$\int_a^b f(x) dx = \left(\frac{b-a}{2}\right) \int_{-1}^1 f\left[\left(\frac{b-a}{2}\right)t + \left(\frac{b+a}{2}\right)\right] dt \sim \left(\frac{b-a}{2}\right) \sum_{j=1}^m w_j f\left[\left(\frac{b-a}{2}\right)z_j + \left(\frac{b+a}{2}\right)\right]$$

- (ii) Os valores de z_j e w_j , $j=1, \dots, m$ encontram-se tabelados para diferentes valores de m .

m	z ₂	z ₃	z ₄	z ₅
2	0,577 350 269 2	-	-	-
3	0	0,774 596 669 2	-	-
4	-	0,339 981 043 6	0,861 136 311 6	-
5	-	0	0,538 469 310 1	0,906 117 981 043 6

m	w ₂	w ₃	w ₄	w ₅
2	1	-	-	-
3	0,888 888 888 9	0,555 555 555 6	-	-
4	-	0,652 145 154 9	0,347 854 845 1	-
5	-	0,568 888 888 9	0,478 628 670 5	0,236 926 885 1

3.3.4 MATRIZ DE GRAM

Seja $G_{ij} = \int_{y_i}^{y_{i+m}} N_i^m(x) N_j^m(x) dx$

A matriz $G = \{G_{ij}\}_{i=1, j=1}^n$ chama-se matriz de Gram associada com os B-splines $\{N_i^m\}_1^n$

Como $G_{ij} = \int_{y_i}^{y_{i+m}} N_i^m(x) N_j^m(x) dx = \int_{y_v}^{y_{v+1}} N_i^m(x) N_j^m(x) dx$, o cálculo de cada uma das m integrais da soma pela fórmula de Gauss permite obter o valor de G_{ij} sem erro de truncamento.

Por definição $N_i^m(x) = (-1)^m (y_{i+m} - y_i) [y_i, \dots, y_{i+m}]_+(x-y)^{m-1}$ e

$N_j^m(x) = (-1)^m (y_{j+m} - y_j) [y_j, \dots, y_{j+m}]_+(x-y)^{m-1}$.

Daí, se $x \in (y_i, y_{i+m})$ existem $2m-1$ B-splines não nulos:

$N_{i+1-m}^m(x), \dots, N_i^m(x), \dots, N_{m+i-1}^m(x)$

Os $N_j^m(x)$ não nulos devem, portanto, satisfazer à condição $i+1-m \leq j \leq i+m-1$.

Como decorrência $G_{ij} = 0$ se $j \notin [i-m+1, i+m-1]$. Ilustrando com $m=3$ e

n=7 obtemos a seguinte configuração para a matriz de Gram:

$$\begin{bmatrix} G_{11} & G_{12} & G_{13} & 0 & 0 & 0 & 0 \\ G_{21} & G_{22} & G_{23} & G_{24} & 0 & 0 & 0 \\ G_{31} & G_{32} & G_{33} & G_{34} & G_{35} & 0 & 0 \\ 0 & G_{42} & G_{43} & G_{44} & G_{45} & G_{46} & 0 \\ 0 & 0 & G_{53} & G_{54} & G_{55} & G_{56} & G_{57} \\ 0 & 0 & 0 & G_{64} & G_{65} & G_{66} & G_{67} \\ 0 & 0 & 0 & 0 & G_{75} & G_{76} & G_{77} \end{bmatrix} \quad \text{com } G_{ij} >> 0 \text{ e } G_{ij} = G_{ji}$$

Concluimos que a matriz de Gram é uma matriz de banda 2m-1, definida positiva e simétrica.

Para cálculo de $\{G_{ij}\}_{i=1, j=1}^n$ procedemos por etapas:

(i) Inicialmente zeramos todos elementos de G.

(ii) Como $G_{ij} = \int_{y_v}^{y_{v+1}} N_i^m(x) N_j^m(x) dx = \int_{y_v}^{y_{v+1}} \left(\frac{y_{v+1} - y_v}{2} \right) \sum_{u=1}^m w_u N_i^m(t_{uv}) N_j^m(t_{uv})$,

sendo $t_u = \left(\frac{y_{v+1} - y_v}{2} \right) z_u + \left(\frac{y_{v+1} + y_v}{2} \right)$, para todo v variando de m a n verificamos se $y_{v+1} - y_v > 0$. Em caso afirmativo para cada valor inteiro de u entre 1 e m:

(a) Calculamos através de BSMQ todos B-splines não nulos no ponto

$$t_{uv} = \left(\frac{y_{v+1} - y_v}{2} \right) z_u + \left(\frac{y_{v+1} + y_v}{2} \right)$$

(b) Para todo i e j tais que $N_i^m(t_{uv})$ e $N_j^m(t_{uv})$ não são nulos adicionamos à G_{ij} a parcela $\left(\frac{y_{v+1} - y_v}{2} \right) w_u N_i^m(t_{uv}) N_j^m(t_{uv})$.

Os valores de $\{G_{ij}\}_{i=1, j=1}^n$ são calculados por meio do procedimento BSMGRM (B-Splines Matriz de GRaM).

3.4 REPRESENTAÇÃO POLINOMIAL POR PARTES DE UM SPLINE S

Embora a expansão em B-splines de um spline s seja preferível para

manipulação em computador há algumas situações em que pode ser preferível trabalhar com sua representação polinomial por partes.

É o caso, por exemplo, em que se deseja avaliar o spline em um grande número de pontos.

Seja $s \in \mathcal{S}(\mathcal{P}_m; m; \Delta)$, sendo $\Delta = \{x_1 < x_2 < \dots < x_k\}$ uma partição do intervalo $[a, b]$ nos subintervalos:

$$I_0 = [a, x_1), I_1 = [x_1, x_2), \dots, I_{k-1} = [x_{k-1}, x_k) \text{ e } I_k = [x_k, b]$$

Sejam $P_0(x), P_1(x), \dots, P_k(x)$ os polinômios que representam o spline s , respectivamente, sobre os subintervalos I_0, I_1, \dots, I_k . Estes polinômios podem ser expressos em relação a base $1, (x-x_1), \dots, (x-x_i)^{m-1}$ em cada intervalo I_i por:

$$P_0(x) = \sum_{j=0}^{m-1} cw_{0j} (x - x_1)^j, \quad x \in I_0$$

$$P_i(x) = \sum_{j=0}^{m-1} cw_{ij} (x - x_i)^j, \quad x \in I_i, \quad i=1, \dots, k$$

e chamam-se representação polinomial por partes de s .

A matriz $\{cw_{ij}\}_{i=0, j=1}^{k, m}$ representa s .

Observamos que na representação polinomial por partes o número de coeficientes é igual a $(k+1)m$ enquanto que na expansão em B-splines o número de coeficientes é $n = m + \sum_{i=1}^k \ell_i$ com $m + k \leq n \leq (k+1)m$.

3.4.1 CONVERSÃO DA EXPANSÃO EM B-SPLINES PARA A REPRESENTAÇÃO POLINOMIAL POR PARTES.

Supondo $s(x)$ conhecido através de sua expansão em B-splines teremos para $x \in I_i$

$$D_+ s(x) = D_+ \left(\sum_{j=1}^m cw_{ij} (x-x_i)^{j-1} \right) = \sum_{j=2}^m cw_{ij} (j-1) (x-x_i)^{j-2}$$

$$D_+^2 s(x) = D_+ \left(\sum_{j=2}^m cw_{ij} (j-1) (x-x_i)^{j-2} \right) = \sum_{j=3}^m cw_{ij} (j-1)(j-2) (x-x_i)^{j-3}$$

⋮

$$D_+^{m-1} s(x) = D_+ \left(\sum_{j=m-1}^m cw_{ij} (j-1)(j-2)\dots(j-m+2) (x-x_i)^{j-m+1} \right) = cw_{im} (m-1)(m-2)\dots(m-m+1)$$

Daí $D_+^{j-1} s(x_i) = cw_{ij} (j-1)!$, $i=1,2,\dots,k$, $j=1,2,\dots,m$,

Portanto, $cw_{ij} = \frac{D_+^{j-1} s(x_i)}{(j-1)!}$, $i=1,\dots,k$, $j=1,\dots,m$.

Analogamente

$$cw_{0j} = \frac{D_-^{j-1} s(x_1)}{(j-1)!}, \quad j=1,\dots,m$$

Para a conversão desejada utiliza-se o procedimento BSDX ou BSDX2 para calcular $\{D_+^{j-1} s(x_i)\}_{i=1}^k$, $j=1,\dots,m$. Os cálculos de $\{D_-^{j-1} s(x_1)\}_{j=1}^m$ deverão ser feitos pelo mesmo algoritmo utilizado em BSDX ou BSDX2 considerando-se $x = x_1$.

Utilizando-se o dispositivo prático de Briot-Ruffini para avaliar $s(x)$ através de sua representação polinomial por partes serão executadas apenas $(m-1)$ multiplicações enquanto que BSX2 exige $3m(m-1)/2$. Esta diferença torna-se relevante quando for grande o número de pontos onde se de seja avaliar $s(x)$.

As derivadas de $s(x)$ também podem ser obtidas mais rapidamente através do esquema de Horner.

Os valores da matriz $\{cw_{ij}\}_{i=0}^k$, $j=1,\dots,m$ são calculados por meio do procedimento BSRPP (B-Spline Representação Polinomial por Partes) sendo armazenados, respectivamente, em $\{cp(i,j)\}_{i=0}^k$, $j=1,\dots,m$.

O valor de $s(x)$ avaliado por intermédio do dispositivo prático de Briot-Ruffini e da representação de $s(x)$ por polinômios por partes é obtido por meio do procedimento RPPSX (Representação Polinomial por Partes

para avaliar o Spline s no ponto X).

Através do procedimento RPPSDX (Representação Polinomial por Partes na avaliação de um Spline s e suas Derivadas no ponto X) calculamos os valores de $s(x)$, $D_+s(x)$, $D_+^2s(x)$, ..., $D_+^{m-1}s(x)$ utilizando a representação polinomial por partes e o esquema de Horner, armazenando-os, respectivamente em $dx[1]$, $dx[2]$, ..., $dx[m]$.

4. B-SPLINES COM NÓS IGUALMENTE ESPAÇADOS

No caso em que os nós de uma partição de $[a, b]$ forem igualmente espaçados ocorrem muitas simplificações em fórmulas que envolvem expansões em B-splines permitindo a formulação de algoritmos mais eficientes. Os nós são considerados com multiplicidade 1.

Os procedimentos que envolvem nós igualmente espaçados e que correspondem aos que já foram introduzidos no caso geral terão seus nomes iguais aos adotados acrescidos da letra H.

Assim sendo descreveremos os seguintes procedimentos cujos algoritmos tornaram-se mais eficientes devido às simplificações introduzidas : BSMQH, BSMCDH, BSXH, BSX2H, BSDXH, BSRPPH, BSMCIH e BSIDH.

4.1 PARTIÇÃO UNIFORME DE PASSO H

Dado um intervalo $[a, b]$ e um inteiro $k \geq 1$ seja

$$x_i = a + ih, \quad i=0, 1, \dots, k+1, \quad h=(b-a)/(k+1)$$

Dizemos que $\Delta h = \{x_i\}_1^k$ define uma PARTIÇÃO UNIFORME DE $[a, b]$ COM PASSO H.

4.2 PARTIÇÃO ESTENDIDA ASSOCIADA COM $S_m(\Delta h)$

Numa partição uniforme com nós de multiplicidade 1, o spline s será

considerado como elemento do espaço $\mathcal{S}_m(\Delta h)$, ou seja, do espaço dos splines com nós simples (Vide 1.3) em pontos de uma partição uniforme.

Como a soma das multiplicidades coincide com o número de nós distintos de Δh valem as seguintes relações:

$$k = (b-a)/h - 1$$

$$n = m + k$$

$$y_i = a + (i-m)h, \quad i=1, 2, \dots, n+m$$

Uma base de $\mathcal{S}_m(\Delta h)$ é dada pelos B-splines normalizados $\{N_i^m\}_{i=1}^n$ associados com a partição estendida $\{y_i\}_{i=1}^{m+n}$:

← m →			┌───────────┐	← m →			└───────────┘			
			x ₁	...			x _k			
i	1	...	m-1	m	m+1	...	m+k	n+1	...	n+m
y _i	a-(m-1)h	...	a-h	a	a+h	...	a+kh	b	...	a+mh

Como os nós $\{y_i\}_1^{n+m}$ são crescentes e possuem multiplicidade 1, dado $x \in [a, b]$ a determinação de ℓ tal que $y_\ell \leq x < y_{\ell+1}$ dispensa um processo de pesquisa sendo dado diretamente por:

"se $x=b$ então $\ell = n$ senão $\ell = (x-a)/h + m$ "

Outra vantagem da escolha desta partição estendida é que todos B-splines normalizados são trasladados de $N_i^m(x) = N^m\left(\frac{x - y_i}{h}\right)$.

De fato, $Q_i^m(x) = (-1)^m [y_i, \dots, y_{i+m}] (x-y)_+^{m-1}$

Como os nós $\{y_j\}_i^{i+m}$ são igualmente espaçados com passo h

$$Q_i^m(x) = (-1)^m [y_i, \dots, y_{i+m}] (x-y)_+^{m-1} = \frac{(-1)^m \Delta h^m (x-y)_+^{m-1}}{h^m m!} = \frac{\sum_{i=0}^m (-1)^i \binom{m}{i} (x-y_i)_+^{m-1}}{h^m m!}$$

$$Q_i^m(x) = \frac{\sum_{i=0}^m (-1)^i \binom{m}{i} \left(\frac{x-y_i}{h}\right)_+^{m-1}}{h^m m!} = \frac{1}{h} Q^m\left(\frac{x-y_i}{h}\right)$$

Logo, $N_i^m(x) = (y_{i+m} - y_i) Q_i^m(x) = h Q_i^m(x) = Q^m\left(\frac{x-y_i}{h}\right) = N^m\left(\frac{x-y_i}{h}\right)$

Portanto, para todo $s \in \mathcal{S}_m(\Delta h)$, $s = \sum_{i=1}^m c_i N^m\left(\frac{x-y_i}{h}\right)$

4.3 REPRESENTAÇÃO DE UM SPLINE $s \in \mathcal{S}_m(\Delta h)$ ATRAVÉS DE SUA EXPANSÃO EM B-SPLINES.

Devido a unicidade de representação de um vetor em relação a base, para armazenar um spline s num computador é suficiente armazenar seus coeficientes $\{c_i\}_1^n$ tais que $s = \sum_{i=1}^n c_i N^m\left(\frac{x-y_i}{h}\right)$.

4.4 AVALIAÇÃO DE UMA EXPANSÃO EM B-SPLINES DE UM SPLINE $s \in \mathcal{S}_m(\Delta h)$

Será feita através dos procedimentos BSX2H e BSXH(BSMQH).

BSXH foi desenvolvido apenas para conferência de algoritmos equivalentes utilizados em BSXH e BSX2H. Entretanto, BSX2H deve ser preferido por ser mais eficiente. Por exemplo, para $m=4$, BSXH e BSX2H envolvem, respectivamente, 27 e 14 operações de multiplicações ou divisões.

4.5 DERIVADAS DE UMA EXPANSÃO EM B-SPLINES DE UM SPLINE $s \in \mathcal{S}_m(\Delta h)$.

A partir de $s(x) = \sum_{i=1}^n c_i N^m\left(\frac{x-y_i}{h}\right)$ e da relação recursiva

$D_+ N^m(x) = N^{m-1}(x) - N^{m-1}(x-1)$ obtemos para qualquer x de $I = [a, b]$:

$$D_+^j s(x) = h^{-j} \sum_{i=j}^n \nabla^j c_i N^{m-j}\left(\frac{x-y_i}{h}\right), \quad j=0, 1, \dots, m-1$$

onde: (i) $\nabla^j f(x) = \sum_{i=0}^j (-1)^{j-i} \binom{j}{i} f(x-i)$

(ii) convencionamos $c_{-1} = \dots = c_{-j} = 0$

Utilizando a notação $CD(j, i) = \nabla^{j-1} c_i$ definidos na matriz

$\{CD(j,i)\}_{j=1, i=j}^m, n$ os coeficientes das derivadas de $s(x)$ a menos do fator h^{-j} . A j -ésima linha desta matriz corresponde aos coeficientes de $D_+^{j-1}s(x)$ a menos de h^{-j} .

Como $\nabla f(x) = f(x) - f(x-1)$ os cálculos de $\nabla^{j-1}c_i$ são gerados facilmente na construção da matriz abaixo a partir de $cd(1,i) = c_i, i=1, \dots, n$, ilustrada para $m=4$ e $n=7$:

i	$\nabla^0 cd$	∇cd	$\nabla^2 cd$	$\nabla^3 cd$
1	cd(1,1)			
2		cd(2,2)		
3			cd(3,3)	
4				cd(4,4)
5				
6				
7				

$$\{cd(j,i) = cd(j-1,i) - cd(j-1,i-1)\}_{j=2, i=j}^{4, 7}$$

Os valores de $\{cd(j,i)\}_{j=1, i=j}^m, n$ são calculados através do procedimento BSMCDH (B-Spline Matriz CD, passo H)

Finalmente, combinando os procedimentos BSMCDH e BSMQH em BSDXH, dado $x \in [a,b]$, avaliamos $D_+^{j-1}s(x), j=1, \dots, m$, armazenando estes valores respectivamente em $dx[1], \dots, dx[m]$.

Entretanto, pode-se desenvolver um algoritmo mais eficiente que o utilizado em BSDXH utilizando-se a versão simplificada de BSDX2 através dos procedimentos BSMCDH e BSX2H, dando origem a BSDX2H.

4.6 INTEGRAIS DE $s(x) \in \mathcal{S}_m(\Delta h)$ ATRAVÉS DE SUA EXPANSÃO EM B-SPLINES

O cálculo de integrais indefinidas ou definidas de expansão em B-splines associados com uma partição uniforme com nós de multiplicidade 1 pode ser feito exatamente como no caso geral. Em BSMCIH geramos a partição estendida associada com $\mathcal{S}_m(\Delta)$ e utilizamos o mesmo algoritmo considerado em BSMCI. Desprezando-se as operações para cálculos dos nós da partição estendida, o número de multiplicações é o mesmo em BSMCI e BSMCIH.

Os valores de $\{mci_i\}_1^{m+1}$ gerados em BSMCIH correspondem aos coeficientes da integral indefinida $\int_{y_1}^x s(t)dt = \sum_{i=1}^{n+1} ci_i N^{m+1}\left(\frac{x-y_i}{h}\right)$, multiplicados por m .

Em BSIDH inicialmente executamos BSMCIH e em seguida BSX2H com $\tilde{m} = m+1$ para $x = ei$ e $x = es$. A execução de BSX2H com $\tilde{m}=m+1$ automaticamente gera a partição estendida associada a $\mathcal{S}_{m+1}(\Delta)$. Finalmente, obtém-se $BSIDH = \int_{ei}^{es} s(t)dt = \frac{1}{m} \left[\int_{y_1}^{es} s(t)dt - \int_{y_1}^{ei} s(t)dt \right]$. BSIDH é mais eficiente que BSID; por exemplo, para $m=4$ enquanto BSID executa 37 multiplicações ou divisões, BSIDH executa apenas 29.

4.7 PRODUTOS INTERNOS

Os produtos internos de B-splines normalizados associados com uma partição uniforme podem ser calculados explicitamente.

Em geral, $I_j^{m,n} = \int_0^m N^m(x) N^n(x+j)dx$, $j=0,1,\dots,m-1$.

Se $\{N_i^m\}$ são B-splines associados com uma partição uniforme de passo h , então para todo i e j , $\int_0^{mh} N_i^m(x) N_{i+j}^n(x)dx = h I_j^{m,n}$, $j=0,1,\dots,m-1$.

4.8 CONVERSÃO DE UMA EXPANSÃO EM B-SPLINES DE $s \in \mathcal{S}_m(\Delta h)$ PARA SUA REPRESENTAÇÃO POLINOMIAL POR PARTES.

Consideremos $s \in \mathcal{S}_m(\Delta h)$ e $s(x) = \sum_{i=1}^n c_i N^m\left(\frac{x-y_i}{h}\right)$;

$\Delta h = \{x_1 < x_2 < \dots < x_k\}$ particiona $I = [a, b]$ em $k+1$ subintervalos:

$I_0 = [a, x_1)$, $I_1 = [x_1, x_2)$, ..., $I_{k-1} = [x_{k-1}, x_k)$ e $I_k = [x_k, b]$. Sejam

$p_0(x), p_1(x), \dots, p_k(x)$ os polinômios de ordem m que representam o spline

$s \in \mathcal{E}_m(\Delta h)$ respectivamente em I_0, I_1, \dots, I_k , definidos por:

$$p_0(x) = \sum_{j=1}^m cp_{0j} (x-x_1)^{j-1}, \quad x \in I_0$$

$$p_i(x) = \sum_{j=1}^m cp_{ij} (x-x_i)^{j-1}, \quad x \in I_i, \quad i=1, \dots, k$$

Conforme vimos no caso geral:

$$\{cp_{0j}\}_{j=1}^m = \left\{ \frac{D_-^{j-1} s(x_1)}{(j-1)!} \right\}_{j=1}^m$$

$$\{cp_{ij}\}_{i=1, j=1}^{k, m} = \left\{ \frac{D_+^{j-1} s(x_i)}{(j-1)!} \right\}_{i=1, j=1}^{k, m}$$

Os coeficientes dos polinômios por partes são obtidos através do procedimento BSRPPH.

"RELATÓRIO TÉCNICO"
DEPARTAMENTO DE MATEMÁTICA APLICADA
TÍTULOS PUBLICADOS

- RI-MAP-7701 - Ivan de Queiroz Barros
On equivalence and reducibility of Generating Matrices
of LR-Procedures - Agosto 1977
- RI-MAP-7702 - V.W. Setzer
A Note on a Recursive Top-Down Analyzer of N.Wirth - Dezembro 1977
- RI-MAP-7703 - Ivan de Queiroz Barros
Introdução a Aproximação Ótima - Dezembro 1977
- RI-MAP-7704 - V.W. Setzer, M.M. Sanches
A linguagem "LEAL" para Ensino básico de Computação - Dezembro 1977
- RI-MAP-7801 - Ivan de Queiroz Barros
Proof of two lemmas of interest in connection with discretization
of Ordinary Differential Equations - Janeiro 1978
- RI-MAP-7802 - Silvio Ursic, Cyro Patarra
Exact solution of Systems of Linear Equations with Iterative Methods
Fevereiro 1978
- RI-MAP-7803 - Martin Grötschel, Yoshiko Wakabayashi
Hypohamiltonian Digraphs - Março 1978
- RI-MAP-7804 - Martin Grötschel, Yoshiko Wakabayashi
Hypotractable Digraphs - Maio 1978
- RI-MAP-7805 - W. Hessa, V.W. Setzer
The Line-Justifier: an example of program development by transformations
Junho 1978
- RI-MAP-7806 - Ivan de Queiroz Barros
Discretização
Capítulo I - Tópicos Introdutórios
Capítulo II - Discretização
Julho 1978
- RI-MAP-7807 - Ivan de Queiroz Barros
($n, 7$) - Estabilidade e Métodos Preditores-Corretores - Setembro 1978
- RI-MAP-7808 - Ivan de Queiroz Barros
Discretização
Capítulo III - Métodos de passo progressivo para Eq. Dif. Ord. com
condições iniciais - Setembro 1978
- RI-MAP-7809 - V.W. Setzer
Program development by transformations applied to relational Data-Base
queries - Novembro 1978
- RI-MAP-7810 - Ngwiffo B. Boyom, Paulo Beulos
Homogeneity of Cartan-Killing spheres and singularities of vector
fields - Novembro 1978

TÍTULOS PUBLICADOS

- RT-MAP-7811 - D.T. Fernandes e C. Patarra
Sistemas Lineares Esparsos, um Método Exato de Solução - Novembro 1978
- RT-MAP-7812 - V.W. Setzer e G. Bressan
Desenvolvimento de Programas por Transformações: uma Comparação entre dois Métodos - Novembro 1978
- RT-MAP-7813 - Ivan de Queiroz Barros
Variação do Passo na Discretização de Eq. Dif. Ord. com Condições Iniciais - Novembro 1978
- RT-MAP-7814 - Martin Grötschel e Yoshiko Wakabayashi
On the Complexity of the Monotone Asymmetric Travelling Salesman Polytope I: HIPOHAMILTONIAN FACETS - Dezembro 1978
- RT-MAP-7815 - Ana F. Humes e E.I. Jury
Stability of Multidimensional Discrete Systems: State-Space Representation Approach - Dezembro 1978
- RT-MAP-7901 - Martin Grötschel, Yoshiko Wakabayashi
On the complexity of the Monotone Asymmetric Travelling Salesman Polytope II: HYPOTRACEABLE FACETS - Fevereiro 1979
- RT-MAP-7902 - M.M. Sanches e V.W. Setzer
A portabilidade do Compilador para a Linguagem LEAL - Junho 1979
- RT-MAP-7903 - Martin Grötschel, Carsten Thomassen, Yoshiko Wakabayashi
Hypotractable Digraphs - Julho 1979
- RT-MAP-7904 - N'Guiffo B. Boyom
Translations non triviales dans les groupes (transitifs) des transformations affines - Novembro 1979
- RT-MAP-8001 - Ângelo Barone Netto
Extremos detectáveis por jatos - Junho 1980
- RT-MAP-8002 - Ivan de Queiroz Barros
Medida e Integração
Cap. I - Medida e Integração Abstrata - Julho 1980
- RT-MAP-8003 - Routo Terada
Fast Algorithms for NP-Hard Problems which are Optimal or Near-Optimal with Probability one - Setembro 1980
- RT-MAP-8004 - V.W. Setzer e R. Lapyda
Uma Metodologia de Projeto de Bancos de Dados para o Sistema ADABAS
Setembro 1980
- RT-MAP-8005 - Imre Simon
On Brzozowski's Problem: $(l\cup A)^m = A^*$ - Outubro 1980
- RT-MAP-8006 - Ivan de Queiroz Barros
Medida e Integração
Cap. II - Espaços L_p - Outubro 1980

TÍTULOS PUBLICADOS

- RT-MAP-8101 - Luzia Kazuko Yoshida e Gabriel Richard Bitran
Um algoritmo para Problemas de Programação Vetorial com Variáveis Zerradas - Fevereiro 1981
- RT-MAP-8102 - Ivan de Queiroz Barros
Medida e Integração
Cap. III - Medidas em Espaços Topológicos - Março 1981
- RT-MAP-8103 - V.W. Setzer, R. Lapyda
Design of Data Models for the ADABAS System using the Entity-Relationship Approach - Abril 1981
- RT-MAP-8104 - Ivan de Queiroz Barros
Medida e Integração
Cap. IV - Medidas e Integração Vetoriais - Abril 1981
- RT-MAP-8105 - U.S.R. Murty
Projective Geometries and Their Truncations - Maio 1981
- RT-MAP-8106 - V.W. Setzer, R. Lapyda
Projeto de Bancos de Dados, Usando Modelos Conceituais
Este relatório Técnico complementa o RT-MAP-8103. Ambos substituem o RT-MAP-8004 ampliando os conceitos ali expostos. - Junho 1981
- RT-MAP-8107 - Maria Angela Gurgel, Yoshiko Wakabayashi
Embedding of Trees - August 1981
- RT-MAP-8108 - Ivan de Queiroz Barros
Mecânica Analítica Clássica - Outubro 1981
- RT-MAP-8109 - Ivan de Queiroz Barros
Equações Integrais de Fredholm no Espaço das Funções A-Uniformemente Contínuas
- Novembro 1981
- RT-MAP-8110 - Ivan de Queiroz Barros
Dois Teoremas sobre Equações Integrais de Fredholm - Novembro 1981
- RT-MAP-8201 - Siang Mun Song
On a High-Performance VLSI Solution to Database Problems - Janeiro 1982
- RT-MAP-8202 - Maria Angela Gurgel, Yoshiko Wakabayashi
A Result on Hamilton-Connected Graphs - Junho 1982
- RT-MAP-8203 - Jörg Blatter, Larry Schumaker
The Set of Continuous Selections of a Metric Projection in $C(X)$
- Outubro 1981
- RT-MAP-8204 - Jörg Blatter, Larry Schumaker
Continuous Selections and Maximal Alternators for Spline Approximation
- Dezembro 1981
- RT-MAP-8205 - Arnaldo Mandel
Topology of Oriented Matroids - Junho 1982
- RT-MAP-8206 - Erich J. Neuhold
Database Management Systems; A General Introduction - Novembro 1982
- RT-MAP-8207 - Béla Bollobás
The Evolution of Random Graphs - Novembro 1982

- RT-MAP-8208 - V.W. Setzer
Um Grafo Sintático para a Linguagem PL/M-80 - Novembro 1982
- RT-MAP-8209 - Jayme Luiz Szwarcfiter
A Sufficient Condition for Hamilton Cycles - Novembro 1982
- RT-MAP-8301 - W.M. Oliva
Stability of Morse-Smale Maps - Janeiro 1983
- RT-MAP-8302 - Belá Bollobás, Istvan Simon
Repeated Random Insertion into a Priority Queue - Fevereiro 1983
- RT-MAP-8303 - V.W. Setzer, P.C.D. Freitas e B.C.A. Cunha
Um Banco de Dados de Medicamentos - Julho 1983
- RT-MAP-8304 - Ivan de Queiroz Barros
O Teorema de Stokes em Variedades Celuláveis - Julho 1983
- RT-MAP-8305 - Arnaldo Mandel
The 1-Skeleton of Polytopes, oriented Matroids and some other lattices -
 - Julho 1983
- RT-MAP-8306 - Arnaldo Mandel
Alguns Problemas de Enumeração em Geometria - Agosto 1983
- RT-MAP-8307 - Siang Wun Song
Complexidade de E/S e Projetos Optimais de Dispositivos para Ordenação -
 - Agosto 1983
- RT-MAP-8401-A - Dirceu Douglas Salvetti
Procedimentos para Cálculos com Splines
 Parte A - Resumos Teóricos - Janeiro 1984
- RT-MAP-8401-B
 Parte B - Descrição de Procedimentos - Janeiro 1984
- RT-MAP-8401-C
 Parte C - Listagem de Testes - Janeiro 1984