Splines Polinomiais Não Homogêneos na Esfera

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Anamaria Gomide e aprovada pela Banca Examinadora.

Campinas, 23 de Julho de 2000.

Jorge Stolfi (Orientador)

Márcio Luiz de Andrade Neto
(Co-Orientador)

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Engenharia Elétrica.

Splines Polinomiais Não Homogêneos na Esfera

Anamaria Gomide

Tese de Doutorado

Splines Polinomiais Não Homogêneos na Esfera

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Anamaria Gomide e aprovada pela Banca Examinadora.

Campinas, 17 de maio de 1999.

Jorge Stolfi (Orientador)

Márcio Luiz de Andrade Neto
(Co-Orientador)

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Engenharia Elétrica.

Faculdade de Engenharia Elétrica e de Computação
Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas

Splines Polinomiais Não Homogêneos na Esfera

Anamaria Gomide

Abril de 1999

Banca Examinadora:

- Jorge Stolfi (Orientador)
- Carlos Antonio de Moura
  Departamento de Ciência da Computação - UFF
- Saulo Rabello Maciel
  Instituto de Matemática e Estatística - USP
- Wu Shin Ting
  Faculdade de Engenharia Elétrica e de Computação - UNICAMP
- Fulano de Tal
  Faculdade de Engenharia Elétrica e de Computação - UNICAMP
- Sônia Maria Gomes (Suplente)
  Instituto de Matemática Estatística e Computação Científica - UNICAMP
- Fulano de Tal (Suplente)
  Faculdade de Engenharia Elétrica e de Computação - UNICAMP

Agradecimentos

Ao Prof. Dr. Jorge Stolfi o meu mais sincero agradecimento pelo apoio irrestrito desde o início desta jornada. Agradeço a valiosa orientação, o seu profissionalismo, sua prontidão e qualidade superior de suas respostas.

Ao Prof. Dr. Márcio Luiz de Andrade Neto pela confiança e abertura oferecida para que eu pudesse seguir meus próprios caminhos.

À FEEC da Unicamp, através de seu Programa de Pós-Graduação, pelo espaço acadêmico que viabilizou a execução desse trabalho.

Ao Instituto de Computação da Unicamp pelo apoio constante.

Aos mestres e amigos Maria Cristina Cunha, João Frederico Meyer, Sônia Maria Gomes, Renato H. L. Pedrosa e Lucio Tunes dos Santos por suas valiosas contribuições, interesse e sugestões no decorrer desse trabalho.

A todos os amigos e companheiros, que de tantos não cabe nomear, pela palavra amiga, pelo carinho e amizade em todos os momentos. Vocês sabem o quão importante foi a participação de todos neste percurso.

Ao Projeto Pronex-Finep N. 76.97.1022.00 – "Sistemas Avançados de Informação" pelo apoio financeiro na divulgação desse trabalho.

À minha família e, em especial, ao meu filho Gustavo.

Resumo

Estudamos neste trabalho o método de elementos finitos para aproximação de funções, e integração de equações diferenciais parciais sobre a esfera $\mathbf{S}^2$. Tais problemas ocorrem em várias aplicações práticas, incluindo modelagem global do tempo, geofísica, iluminação, etc.

Definimos um *polinômio esférico* como sendo a restrição à esfera $\mathbf{S}^{n-1}$ de um polinômio nas coordenadas cartesianas $(x_1, x_2, \ldots, x_n)$ de $\mathbf{R}^n$. Denotamos por $\mathcal{P}^{d,n}/\mathbf{S}^{n-1}$ o espaço de todos os polinômios esféricos de grau total $\leq d$, e por $\mathcal{H}^{d,n}/\mathbf{S}^{n-1}$ o espaço dos *polinômios esféricos homogêneos* de grau total d.

As funções que investigamos são as funções esféricas $\mathbf{C}_r$ *polinomiais por partes*, ou *splines esféricos* $\mathbf{C}_r$, definidos em relação a uma triangulação esférica T de $\mathbf{S}^{n-1}$. Seja $\mathcal{P}_r^{d,n}[T]/\mathbf{S}^{n-1}$ o espaço de todas as funções f de $\mathbf{S}^{n-1}$ em $\mathbf{R}$ tais que (1) a restrição de f a cada triângulo de T coincide com uma função de $\mathcal{P}^{d,n}/\mathbf{S}^{n-1}$; e (2) a função f tem continuidade de ordem-r através das fronteiras de T. Analogamente, seja $\mathcal{H}_r^{d,n}[T]/\mathbf{S}^{n-1}$ o sub-espaço de $\mathcal{P}_r^{d,n}[T]/\mathbf{S}^{n-1}$ dos *splines esféricos homogêneos*, que consiste das funções que são $\mathcal{H}^{d,n}/\mathbf{S}^{n-1}$ em cada triângulo de T.

Neste trabalho mostramos que $\mathcal{P}^{d,n}/\mathbf{S}^{n-1} = \mathcal{H}^{d-1,n}/\mathbf{S}^{n-1} \oplus \mathcal{H}^{d,n}/\mathbf{S}^{n-1}$, e estendemos esse resultado aos splines esféricos, mostrando que
$\mathcal{P}_r^{d,n}[T]/\mathbf{S}^{n-1} = \mathcal{H}_r^{d-1,n}[T]/\mathbf{S}^{n-1} \oplus \mathcal{H}_r^{d,n}[T]/\mathbf{S}^{n-1}$.

Alfeld, Neamtu e Schumaker propuseram recentemente o espaço $\mathcal{H}_r^d[T]/\mathbf{S}^2$ para aproximação na esfera $\mathbf{S}^2$, e obtiveram uma construção explícita de uma base para o espaço $\mathcal{H}_r^d[T]/\mathbf{S}^2$, quando $d \geq 3r+2$. Combinando esta construção com o nosso resultado, acima descrito, nós obtemos uma base local explícita para o espaço $\mathcal{P}_r^d[T]/\mathbf{S}^2$ quando $d \geq 3r+3$.

Nossa tese é que o $\mathcal{P}_r^d[T]/\mathbf{S}^2$ é um espaço de aproximação mais natural e eficaz do que $\mathcal{H}_r^d[T]/\mathbf{S}^2$. Analisamos, em particular, o uso dos espaços $\mathcal{P}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$ para aproximar funções restritas a esfera $\mathbf{S}^2$, pelo critério dos mínimos quadrados. Analisamos também o uso do espaço $\mathcal{P}_1^6[T]/\mathbf{S}^2$ para resolução numérica de equações diferenciais parciais na esfera, pelo método dos elementos finitos, e descrevemos uma técnica multi-escala para acelerar a convergência em malhas finas.

Abstract

We study in this work the finite element method for function approximation and integration of partial differential equation on the sphere $\mathbf{S}^2$. These problems occur in many pratical applications, including global weather modeling, geophysics, illumination, etc..

A *spherical polynomial* is the restriction to the sphere $\mathbf{S}^{n-1}$ of a polynomial in the coordinates $x_1, x_2, \ldots, x_n$ of $\mathbf{R}^n$. We denote by $\mathcal{P}^{d,n}/\mathbf{S}^{n-1}$ the space of spherical polynomials with total degree $\leq d$, and by $\mathcal{H}^{d,n}/\mathbf{S}^{n-1}$ the space of *homogeneous spherical polynomials* with total degree d.

The functions we investigate are the $\mathbf{C}_r$ *piecewise polynomial functions on* $\mathbf{S}^{n-1}$, or $\mathbf{C}_r$ *spherical splines*, defined relative to a spherical triangulation T on the sphere $\mathbf{S}^{n-1}$. Let $\mathcal{P}_r^{d,n}[T]/\mathbf{S}^{n-1}$ be the space of all functions f from $\mathbf{S}^{n-1}$ to $\mathbf{R}$ such that (1) the restriction of f to each triangle of T matches some function in $\mathcal{P}^{d,n}/\mathbf{S}^{n-1}$; and (2) the function f has order-r continuity across the boundaries of T. Analogously, let $\mathcal{H}_r^{d,n}[T]/\mathbf{S}^{n-1}$ denote the subspace of $\mathcal{P}_r^{d,n}[T]/\mathbf{S}^{n-1}$, *the homogeneous spherical splines*, which consists of the functions that are $\mathcal{H}^{d,n}/\mathbf{S}^{n-1}$ within each triangle of T.

We show that $\mathcal{P}^{d,n}/\mathbf{S}^{n-1} = \mathcal{H}^{d,n}/\mathbf{S}^{n-1} \oplus \mathcal{H}^{d-1,n}/\mathbf{S}^{n-1}$, and generalize this result to spherical splines, showing that $\mathcal{P}_r^{d,n}[T]/\mathbf{S}^{n-1} = \mathcal{H}_r^{d,n}[T]/\mathbf{S}^{n-1} \oplus \mathcal{H}_r^{d-1,n}[T]/\mathbf{S}^{n-1}$.

Alfeld, Neamtu and Shumaker proposed recently the space $\mathcal{H}_r^d[T]/\mathbf{S}^2$ for approximation on the sphere $\mathbf{S}^2$, and they obtained explicit bases for that space, when $d \geq 3r+2$. Combining their construction with our result above, we obtain explicit local bases for the spaces $\mathcal{P}_r^d[T]/\mathbf{S}^2$, when $d \geq 3r+3$.

We argue that the space $\mathcal{P}_r^d[T]/\mathbf{S}^2$ is a more natural and effective tool than $\mathcal{H}_r^d[T]/\mathbf{S}^2$ for approximation on the sphere. We analyze, in particular, the use of spaces $\mathcal{P}_0^3[T]/\mathbf{S}^2$ and $\mathcal{P}_1^6[T]/\mathbf{S}^2$ for least squares function approximation on the sphere $\mathbf{S}^2$. We analyze also the use of the space $\mathcal{P}_1^6[T]/\mathbf{S}^2$ in the numerical integration of partial differential equations; and we describe a multi-scale technique for accelerating the convergence on large grids.

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

Muitos problemas físicos são modelados matematicamente por funções reais ou equações diferenciais parciais (EDP) sobre a esfera $\mathbf{S}^2$. Podemos citar exemplos em geofísica (anomalias gravitacionais), metereologia, (simulação global do tempo, dados climáticos), computação gráfica etc.

Em tais aplicações é comum representar uma função na esfera em termos de coordenadas esféricas ϕ, θ — longitude e latitude, respectivamente. A função pode ser definida por um polinômio em ϕ, θ, ou harmônico esférico de certa ordem. Estas representações tem alguns problemas, entretanto: há descontinuidade nos polos, as geodésicas são representadas por curvas complexas, e a correspondência entre o domínio (ϕ, θ) e a esfera é não uniforme. Além disso, as fórmulas dos operadores diferenciais (gradiente, laplaciano, etc.) em coordenadas esféricas são relativamente complexas. Por estas razões tem havido recentemente interesse na modelagem de funções esféricas como funções das coordenadas cartesianas espaciais (x, y, z), restritas à esfera.

Um *polinômio esférico* é a restrição a $\mathbf{S}^{n-1}$ de um polinômio nas coordenadas $x_1, x_2, \ldots, x_n$ de $\mathbf{R}^n$. Diz-se que uma função f de $\mathbf{R}^n$ para $\mathbf{R}$ é *homogênea de grau d* se $f(ax) = a^d f(x)$, para todo $a \in \mathbf{R}$ e $x \in \mathbf{R}^n$. Seja $\mathcal{P}^{d,n}/\mathbf{S}^{n-1}$ o espaço de polinômios esféricos com grau $\leq d$, em $x_1, x_2, \ldots, x_n$ e $\mathcal{H}^{d,n}/\mathbf{S}^{n-1}$ o sub-espaço dos polinômios *homogêneos* de grau d. Mostramos na seção 2.2, que vale a soma direta

$$\mathcal{P}^{d,n}/\mathbf{S}^{n-1} = \mathcal{H}^{d-1,n}/\mathbf{S}^{n-1} \oplus \mathcal{H}^{d,n}/\mathbf{S}^{n-1} \tag{1.1}$$

Isto quer dizer que todo polinômio esférico geral de grau d é a soma de dois polinômios homogêneos que são únicos.

No capítulo 3, estendemos esse resultado aos *splines* definidos numa decomposição simplicial T sobre a esfera $\mathbf{S}^{n-1}$. Seja $\mathcal{P}_r^{d,n}[T]/\mathbf{S}^{n-1}$ o espaço de todas as funções f de $\mathbf{S}^{n-1}$ em $\mathbf{R}$ tais que (1) a restrição de f a cada triângulo de T pertence a $\mathcal{P}^{d,n}/\mathbf{S}^{n-1}$; e (2) a função f tem continuidade de ordem-r através das fronteiras que separam os simplexos de T. Analogamente, seja $\mathcal{H}_r^{d,n}[T]/\mathbf{S}^{n-1}$ o sub-espaço de $\mathcal{P}_r^{d,n}[T]/\mathbf{S}^{n-1}$ que consiste das funções que são $\mathcal{H}^{d,n}/\mathbf{S}^{n-1}$ em cada triângulo de T. Nós provamos no capítulo 3 que

$$\mathcal{P}_r^{d,n}[T]/\mathbf{S}^{n-1} = \mathcal{H}_r^{d-1,n}[T]/\mathbf{S}^{n-1} \oplus \mathcal{H}_r^{d,n}[T]/\mathbf{S}^{n-1} \tag{1.2}$$

A teoria de *splines* esféricos tem sido desenvolvida principalmente por Alfeld, Neamtu e Schumaker [1, 2, 4]. Eles consideram apenas o sub-espaço $\mathcal{H}_r^d[T]/\mathbf{S}^2$ dos splines esféricos *homogêneos*.

No restante da tese, restringimos nosso estudo às funções definidas sobre $\mathbf{S}^2$. Descrevemos no capítulo 4 a triangulação de Delaunay para a esfera $\mathbf{S}^2$, que utilizamos nesta tese. Descrevemos as funções de percurso e estrutura de dados utilizada na implementação da triangulação.

No capítulo 5 nós descrevemos a representação de um polinômio homogêneo sobre $\mathbf{S}^2$, pelos seus coeficientes de Bézier relativos aos vértices de um triângulo t. Mostramos que, com essa representação, podemos eficientemente calcular o valor do polinômio num ponto p qualquer de $\mathbf{R}^n$, usando o algoritmo de DeCasteljau. Usando a técnica de derivação automática, deduzimos, a partir do algoritmo de DeCasteljau, um algoritmo para calcular o gradiente de um polinômio homogêneo num ponto p. A representação de Bézier é especialmente adequada para *splines* esféricos: para cada face t da triangulação, a função restrita a t é representada pelos seus coeficientes de Bézier relativos aos vértices de t. Descrevemos também algoritmos para localizar o triângulo da triangulação à qual o ponto p pertence.

Para fins de aproximação e modelagem de funções, é desejável que o espaço de funções usado admita uma *base local*: uma base onde cada elemento tenha valor praticamente nulo na esfera toda, exceto numa região pequena. O principal resultado de Alfeld, Neamtu e Schumaker é uma construção explícita para uma base local do espaço $\mathcal{H}_r^d[T]/\mathbf{S}^2$, quando $d \geq 3r+2$. No capítulo 6 nós descrevemos a construção dessa base para $r \leq 1$. Combinando esta construção com o resultado (1.2), nós obtemos uma base local explícita para o espaço $\mathcal{P}_r^d[T]/\mathbf{S}^2$ dos *splines* polinomiais genéricos com $d \geq 3r+3$.

Acreditamos que o espaço $\mathcal{P}_r^d[T]/\mathbf{S}^2$ é uma escolha mais natural do que o espaço $\mathcal{H}_r^d[T]/\mathbf{S}^2$ para aproximar funções definidas na esfera $\mathbf{S}^2$. Uma das razões é que $\mathcal{P}_r^k[T]/\mathbf{S}^2 \subseteq \mathcal{P}_r^d[T]/\mathbf{S}^2$ quando $k \leq d$. Em particular, $\mathcal{P}_r^d[T]/\mathbf{S}^2$ inclui as funções que são constantes sobre $\mathbf{S}^2$, para todo d, enquanto $\mathcal{H}_r^d[T]/\mathbf{S}^2$ só contém tais funções quando d é par.

No capítulo 7 nós analisamos o uso dos espaços $\mathcal{P}_r^d[T]/\mathbf{S}^2$ e $\mathcal{H}_r^d[T]/\mathbf{S}^2$ para aproximar funções restritas à esfera $\mathbf{S}^2$, pelo critério dos mínimos quadrados.

Nós descrevemos no capítulo 8 o uso dos *splines* esféricos na solução de equações diferenciais parciais, pelo método dos elementos finitos. Apresentamos um procedimento iterativo para resolver uma EDP da forma

$$(\mathcal{L}u)(p) = \mathcal{F}(u(p), p) \tag{1.3}$$

onde $\mathcal{L}$ é um operador diferencial linear sobre a esfera, p é a variável independente (um ponto da esfera), u é a função a determinar e $\mathcal{F}$ é uma função (não necessariamente linear) que pode depender de $u(p)$ e p.

Os métodos mais eficientes para integração de equações diferenciais parciais baseiam-se na técnica de *mútiplas escalas* [13]. A idéia é resolver a mesma equação numa hierarquia de várias malhas $M_1, M_2, \ldots, M_k$, com resoluções crescentes em progressão geométrica, sendo que a solução parcial obtida numa malha M_i é usada como estimativa inicial para a malha mais fina M_{i+1}. No capítulo 9 descrevemos o uso da técnica multi-escala para aproximar a solução de uma equação diferencial parcial, com *splines* polinomiais acima descritos. Para tanto, desenvolvemos e implementamos um algoritmo para construção de hierarquias de malhas irregulares, baseado na triangulação de Delaunay [9] esférica para o nível mais fino e no algoritmo de simplificação de Kirkpatrick [12] para a construção das malhas mais grosseiras.

Capítulo 2

Espaços de Polinômios Esféricos

2.1 Funções polinomiais em $\mathbf{R}^n$

Seja $\mathcal{P}^{d,n}$ o espaço dos polinômios em n variáveis de grau $\leq d$. Os elementos desse espaço são funções de $\mathbf{R}^n$ para $\mathbf{R}$. Uma função p pertence a $\mathcal{P}^{d,n}$ se e somente se ela pode ser escrita na forma

$$p(x) = \sum_{0 \leq i_1 + i_2 + \ldots + i_n \leq d} c_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \tag{2.1}$$

para todo $x = (x_1, x_2, \ldots, x_n) \in \mathbf{R}^n$, onde $c_{i_1 i_2 \ldots i_n}$ são coeficientes reais. (Todos os índices são inteiros positivos ou nulos). O conjunto $\mathcal{P}^{d,n}$ é obviamente um um espaço vetorial, de dimensão

$$\dim \mathcal{P}^{d,n} = \binom{d+n}{n}$$

Dizemos que uma função f de $\mathbf{R}^n$ para $\mathbf{R}$ é *homogênea de grau* d se $f(ax) = a^d f(x)$, para todo $a \in \mathbf{R}$ e $x \in \mathbf{R}^n$. Seja $\mathcal{H}^{d,n}$ o subconjunto dos polinômios de $\mathcal{P}^{d,n}$ que são homogêneos de grau d. Uma função h pertence a $\mathcal{H}^{d,n}$ se e somente se ela pode ser escrita na forma

$$h(x) = \sum_{i_1 + i_2 + \ldots + i_n = d} c_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$$

para todo $x = (x_1, x_2, \ldots, x_n) \in \mathbf{R}^n$, onde $c_{i_1 i_2 \ldots i_n}$ são coeficientes reais. Podemos concluir então que

$$\dim \mathcal{H}^{d,n} = \binom{d+n-1}{n-1}$$

É fácil ver que os espaços $\mathcal{H}^{d,n}$ e $\mathcal{H}^{d',n}$ são linearmente independentes, isto é, $\mathcal{H}^{d,n} \cap \mathcal{H}^{d',n} = \{0\}$, quando $d \neq d'$.

2.2 Funções polinomiais em $\mathbf{S}^{n-1}$

Se f é uma função definida no $\mathbf{R}^n$, e $X \subseteq \mathbf{R}^n$, denotamos por f/X a restrição de f ao conjunto X. Por extensão, definimos a restrição de um espaço de funções $\mathcal{F}$ ao conjunto X como sendo

$$\mathcal{F}/X = \{\ f/X : f \in \mathcal{F}\ \}$$

Vamos também utilizar a seguinte notação: se $f/X = g/X$ dizemos que f *é equivalente a* g *módulo* X, e escrevemos $f \equiv g \pmod X$; ou apenas $f \equiv g$, quando X estiver implícito no contexto. É fácil ver que '$\equiv$' é uma relação de equivalência.

Estamos interessados no espaço $\mathcal{P}^{d,n}/\mathbf{S}^{n-1}$, que são as funções polinomiais de $\mathbf{R}^n$ de grau $\leq d$ restritas à esfera unitária $\mathbf{S}^{n-1} = \{\ x \in \mathbf{R}^n : |x| = 1\ \}$. Os elementos pertencentes a este conjunto têm a forma:

$$p(x) = \sum_{0 \leq i_1+i_2+\ldots+i_n \leq d} c_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$$

onde $x = (x_1, x_2, \ldots, x_n) \in \mathbf{S}^{n-1}$ e $c_{i_1 i_2 \ldots i_n} \in \mathbf{R}$.

Observe que dois polinômios que são distintos no $\mathbf{R}^n$ podem ser idênticos quando restritos à esfera $\mathbf{S}^{n-1}$. Portanto, $\mathcal{P}^{d,n}/\mathbf{S}^{n-1}$ tem dimensão menor que $\mathcal{P}^{d,n}$. Os resultados abaixo são fundamentais para a caracterização deste espaço:

Lema 1

$$\mathcal{H}^{d,n}/\mathbf{S}^{n-1} \subseteq \mathcal{H}^{d+2,n}/\mathbf{S}^{n-1}$$

Demonstração:

Seja $h \in \mathcal{H}^{d,n}$,

$$h(x) = \sum_{i_1+i_2+\ldots+i_n = d} c_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$$

Vamos mostrar que os termos de h podem ser transformados em termos de grau $d+2$. Seja $c_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$ um termo de h, então $i_1+i_2+\ldots+i_n = d$. Como $x \in \mathbf{S}^{n-1}$ temos que $x_1^2 + x_2^2 + \ldots + x_n^2 = 1$. Então,

$$c_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} = c_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} (x_1^2 + x_2^2 + \ldots x_n^2)$$

Desta maneira, todos os termos de h de grau d podem ser substituídos por termos de grau $d+2$, obtendo-se assim um polinômio homogêneo h' de grau $d+2$ que coincide com h sobre a esfera $\mathbf{S}^2$. **CQD**

Lema 2 *Se $q \in \mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n}$ e $q \equiv 0 \pmod{\mathbf{S}^{n-1}}$ então $q = 0$.*

Demonstração:

Suponhamos que $q \in \mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n}$ com $q \equiv 0$, isto é

$$q/\mathbf{S}^{n-1} = 0$$

Como $\mathbf{S}^{n-1}$ é uma variedade algébrica [7, 5], a equação mínima de $\mathbf{S}^{n-1}$ deve ser um fator de q, isto é,

$$q = R(x_1^2 + x_2^2 + \ldots + x_n^2 - 1)$$

onde R é algum polinômio em $\mathbf{R}^n$ de grau $n-2$. Como os graus dos termos do polinômio $x_1^2 + x_2^2 + \ldots + x_n^2 - 1$ diferem de 2 e, por outro lado,

$$q \in \mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n}$$

concluímos que $R = 0$, i.e. $q = 0$. **CQD**

Corolário 1

$$\mathcal{H}^{d-1,n}/\mathbf{S}^{n-1} \cap \mathcal{H}^{d,n}/\mathbf{S}^{n-1} = \{0\}$$

Corolário 2 *Se $p, q \in \mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n}$ e $p \equiv q \pmod{\mathbf{S}^{n-1}}$, então $p = q$.*

Teorema 1 *Todo polinômio de $\mathcal{P}^{d,n}$, $n \geq 1$, é equivalente $\pmod{\mathbf{S}^{n-1}}$ a um único polinômio de $\mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n}$.*

Demonstração:

Seja $p \in \mathcal{P}^{d,n}$,

$$p(x) = \sum_{0 \leq i_1 + i_2 + \ldots + i_n \leq d} c_{i_1 i_2 \ldots i_n} x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n}$$

Vamos provar que (1) existe um polinômio $q \in \mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n}$ tal que $p \equiv q$ $\pmod{\mathbf{S}^{n-1}}$; e que (2) este polinômio é único.

(1)*Existência:* Vamos mostrar que os termos de p com grau $\leq d-2$ podem ser transformados em termos de grau d e $d-1$. Um termo de grau $k \leq d-2$

é um polinômio de $\mathcal{H}^{k,n}$. Pelo lema 1, este termo pode ser substituído por um polinômio de $\mathcal{H}^{k+2,n}$, cujos termos têm grau $k+2 \leq d$. Repetindo este processo enquanto houver termos de grau $\leq d-2$ obtemos um polinômio q com termos de grau d e $d-1$.

Portanto, existe um polinômio $q \in \mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n}$ tal que $p \equiv q$ (mod $\mathbf{S}^{n-1}$).

(2) *Unicidade:* Vamos supor que existem q_1 e q_2 em $\mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n}$ com $q_1 \equiv p$ e $q_2 \equiv p$. Como $\equiv$ é transitiva, devemos ter $q_1 \equiv q_2$, isto é $q_1 - q_2 \equiv 0$. Então, pelo lema 2, $q_1 = q_2$. **CQD**

Como consequência do teorema 1, temos o seguinte corolário:

Corolário 3

$$\mathcal{P}^{d,n}/\mathbf{S}^{n-1} = (\mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n})/\mathbf{S}^{n-1} = \mathcal{H}^{d-1,n}/\mathbf{S}^{n-1} \oplus \mathcal{H}^{d,n}/\mathbf{S}^{n-1}$$

Logo,

$$\begin{aligned}
\dim(\mathcal{P}^{d,n})/\mathbf{S}^{n-1} &= \dim(\mathcal{H}^{d-1,n}/\mathbf{S}^{n-1} \oplus \mathcal{H}^{d,n}/\mathbf{S}^{n-1}) && (2.2)\\
&= \dim(\mathcal{H}^{d-1,n} \oplus \mathcal{H}^{d,n}) && (2.3)\\
&= \binom{d+n-1}{n-1} + \binom{d+n-2}{n-1} && (2.4)\\
&= (d+1)^2 && (2.5)
\end{aligned}$$

2.3 Diferenciação de funções esféricas

2.3.1 Gradiente esférico

Se f é uma função de $\mathbf{S}^{n-1}$ para $\mathbf{R}$, denotamos por $\mathbin{\nabla\!\!\!\!\cdot}\, f$ o *gradiente esférico* de f, que é o gradiente de f com respeito às direções tangentes a $\mathbf{S}^{n-1}$. Para fins desse trabalho, definimos $\mathbin{\nabla\!\!\!\!\cdot}\, f$ como um vetor de $\mathbf{R}^n$, tangente a $\mathbf{S}^{n-1}$, tal que a derivada de f em um ponto $p \in \mathbf{S}^{n-1}$, na direção de um vetor unitário v tangente a p, é $v \circ (\mathbin{\nabla\!\!\!\!\cdot}\, f(p))$, onde "$\circ$" denota o produto escalar de vetores de $\mathbf{R}^n$.

Se $f = F/\mathbf{S}^{n-1}$ para alguma função diferenciável F de $\mathbf{R}^n$ para $\mathbf{R}$, então $\mathbin{\nabla\!\!\!\!\cdot}\, f$ é meramente a projeção ortogonal de ∇F (o gradiente ordinário de F) sobre o plano tangente à esfera em p. Isto é, para qualquer ponto $p \in \mathbf{S}^{n-1}$,

$$(\mathbin{\nabla\!\!\!\!\cdot}\, f)(p) = \left[(\nabla F)(p) - ((\nabla F)(p) \circ p)p\right]/\mathbf{S}^{n-1} \tag{2.6}$$

2.3.2 Gradiente de polinômios esféricos

O seguinte teorema mostra que cada componente do gradiente esférico de um polinômio esférico é um polinômio esférico.

Teorema 2 *Se $f \in \mathcal{H}^{d,n}/\mathbf{S}^{n-1}$, então $(\nabla\!\!\!\!\nabla f)_\alpha \in \mathcal{H}^{d+1,n}/\mathbf{S}^{n-1}$, para cada $\alpha \in \{1, \ldots, n\}$.*

Demonstração:

Se f pertence a $\mathcal{H}^{d,n}/\mathbf{S}^{n-1}$, para $d \geq 1$, então $f = F/\mathbf{S}^{n-1}$ para algum polinômio $F \in \mathcal{H}^{d,n}$. É fácil de ver que $(\nabla F)_\alpha = \partial F/\partial x_\alpha$ está em $\mathcal{H}^{d-1,n}$. Portanto, cada componente do lado direito da fórmula (2.6)

$$(\nabla F)(p) - ((\nabla F)(p) \circ p)p$$

está em $\mathcal{H}^{d-1,n} + \mathcal{H}^{d+1,n}$. Pelo lema 1, $\mathcal{H}^{d-1,n}/\mathbf{S}^{n-1}$ é um subespaço de $\mathcal{H}^{d+1,n}/\mathbf{S}^{n-1}$, e o teorema está provado. **CQD**

Corolário 4 *Se $f \in \mathcal{P}^{d,n}/\mathbf{S}^{n-1}$, então $(\nabla\!\!\!\!\nabla f)_\alpha \in \mathcal{P}^{d+1,n}/\mathbf{S}^{n-1}$, para cada $\alpha \in \{1 \ldots n\}$.*

2.3.3 Laplaciano na esfera

Seja $F(x_1, x_2, \ldots, x_n)$ uma função de $\mathbf{R}^n$ para $\mathbf{R}$. O *operador de Laplace ou laplaciano* Δ em $\mathbf{R}^n$ é definido por:

$$\Delta F = \sum_{i=1}^{n} \frac{\partial^2 F}{(\partial x_i)^2}$$

O teorema de Green é uma fórmula importante que relaciona o gradiente e o laplaciano de funções. Sejam Ω uma região compacta de $\mathbf{R}^n$ com interior não vazio, Γ sua fronteira, ν o vetor normal a Γ, F e G funções de $\mathbf{R}^n$ para $\mathbf{R}$ então

$$\int_\Omega F(\Delta G) = -\int_\Omega (\nabla F) \circ (\nabla G) + \int_\Gamma G(\nabla F) \circ \nu$$

se as integrais estiverem definidas.

Para a resolução de equações diferenciais na esfera, precisamos de um operador $\Delta\!\!\!\!\Delta$, análogo ao laplaciano, que satisfaça a versão esférica do teorema de Green [6]:

$$\int_{\mathbf{S}^2} g(\Delta\!\!\!\!\Delta f) = -\int_{\mathbf{S}^2} (\nabla\!\!\!\!\nabla g) \circ (\nabla\!\!\!\!\nabla f) \tag{2.7}$$

Verifica-se que se $f = F/\mathbf{S}^{n-1}$ para alguma função diferenciável F de $\mathbf{R}^n$ para $\mathbf{R}$, o operador desejado é o *laplaciano esférico* $\Delta\!\!\!\!\Delta$ dado pela fómula:

$$\Delta\!\!\!\!\Delta f = \sum_{i=1}^{n} \frac{\partial^2 F}{(\partial x_i)^2}(1 - x_i^2) - 2 \sum_{\substack{1 \leq i,j \leq n \\ i \neq j}} \frac{\partial^2 F}{(\partial x_i)(\partial x_j)} x_i x_j - 2 \sum_{i=1}^{n} \frac{\partial F}{(\partial x_i)} x_i \tag{2.8}$$

2.3.4 Laplaciano de polinômios esféricos

O seguinte teorema mostra que o laplaciano esférico de um polinômio esférico é um polinômio esférico.

Teorema 3 *Se* $f \in \mathcal{H}^{d,n}/\mathbf{S}^{n-1}$, *então* $\mathbin{\triangle} f \in \mathcal{H}^{d,n}/\mathbf{S}^{n-1}$

Demonstração:

Se f pertence a $\mathcal{H}^{d,n}/\mathbf{S}^{n-1}$, para $d \geq 1$, então $f = F/\mathbf{S}^{n-1}$ para algum polinômio $F \in \mathcal{H}^{d,n}$. É facil de ver que $\partial^2 F/(\partial x_i)^2$ e $\partial^2 F/((\partial x_i)(\partial x_j))$ estão em $\mathcal{H}^{d-2,n}$, e $\partial F/\partial x_i$ está em $\mathcal{H}^{d-1,n}$. Lembrando que $1 = \sum_{i=1}^{n} x_i^2$, concluimos que o lado direito da fórmula (2.8) está em $\mathcal{H}^{d,n}/\mathbf{S}^{n-1}$, e o teorema está provado. **CQD**

2.4 Harmônicos esféricos

Há uma relação forte entre os polinômios esféricos, para $n = 3$, e uma classe de funções conhecida como *harmônicos esféricos*. Esta classe é muito usada em física pois surge naturalmente na solução da equação de Laplace $\Delta F = 0$ [15].

Sejam (r, ϕ, θ) as coordenadas esféricas definidas pelas identidades $x = \cos\theta\cos\phi$, $y = \cos\theta\sin\phi$ e $z = \sin\theta$, onde $0 \leq \phi \leq 2\pi$ é longitude e $-\pi/2 \leq \theta \leq \pi/2$ latitude.

Pela definição de homogeneidade, temos que $F(r, \phi, \theta) = r^d G(\phi, \theta)$ onde G é alguma função de (ϕ, θ). Substituindo esta expressão na equação de Laplace, obtemos a seguinte equação diferencial na esfera:

$$\mathbin{\triangle} G + d(d+1)G = 0 \tag{2.9}$$

Demonstra-se [16] que um conjunto máximo linearmente independente de soluções da equação (2.9) são as funções

$$Y_d^m = e^{\mathbf{i}m\phi} P_d^m(\theta), \qquad\qquad m = -d, \ldots, d \tag{2.10}$$

onde $P_d^m(\theta)$ são as funções de Legendre de primeiro tipo de ordem m e grau d [15]. Os elementos desse conjunto são os *harmônicos esféricos de grau d*.

Uma expressão analítica clássica para P_d^m é a fórmula de Rodrigues [24]

$$P_d^m(\theta) = \frac{(1-z^2)^{|m|/2}}{2^d d!} \frac{\partial^{d+|m|}(1-z^2)^d}{\partial z^{d+|m|}} \tag{2.11}$$

onde $z = \sin\theta$.

Denotamos por $\mathcal{Y}^d$ o espaço das funções esféricas reais de $\mathbf{S}^2$ para $\mathbf{R}$ geradas pelos harmônicos esféricos Y_k^m de grau $k \leq d$. O teorema a seguir dá uma caracterização deste espaço.

Teorema 4 *O espaço $\mathcal{Y}^d$ coincide com o espaço $\mathcal{P}^{d,3}/\mathbf{S}^2$.*

Demonstração:

Vamos primeiro mostrar que as partes real e imaginária de todo harmônico esférico Y_d^m de grau d é um elemento de $\mathcal{P}^{d,3}/\mathbf{S}^2$. Pelas fórmulas (2.10) e (2.11) temos

$$Y_d^m = C(d,m)e^{\mathbf{i}m\phi}(1-z^2)^{|m|/2}\,\frac{\partial^{d+|m|}(1-z^2)^d}{\partial z^{d+|m|}}$$

onde $C(d,m)$ é uma constante que depende de m e d. Vamos supor que $m \geq 0$. Lembrando que

$$e^{\mathbf{i}\phi} = \cos\phi + \mathbf{i}\sin\phi = \frac{x+\mathbf{i}y}{\sqrt{x^2+y^2}} = \frac{x+\mathbf{i}y}{(1-z^2)^{1/2}}$$

temos

$$(e^{\mathbf{i}\phi})^m = \frac{(x+\mathbf{i}y)^m}{(1-z^2)^{m/2}}$$

então

$$Y_d^m = C(d,m)(x+\mathbf{i}y)^m\frac{\partial^{d+m}(1-z^2)^d}{\partial z^{d+m}}$$

Agora, observando que $(1-z^2)^d = R(z)$ é um polinômio, na variável z, em que todos os termos têm grau par, entre 0 e $2d$, é fácil mostrar que

$$\frac{\partial^{d+m}(1-z^2)^d}{\partial z^{d+m}}$$

é um polinômio Q(z), na variável z, em que o grau de todos os termos têm a mesma paridade que $d-m$ e grau entre 0 e $d-m$ inclusive.

Então pelo lema 1, a restrição de Q(z) à esfera $\mathbf{S}^2$ é um elemento de $\mathcal{H}^{d-m,3}/\mathbf{S}^2$. Portanto, observando que

$$Y_d^m = C(d,m)(x+\mathbf{i}y)^m Q(z)/\mathbf{S}^2$$

e que $Re[(x+\mathbf{i}y)^m]$ e $Im[(x+\mathbf{i}y)^m]$ são polinômios homogêneos em x e y de grau m, temos que $Re(Y_d^m)$ $Im(Y_d^m)$ são também elementos de $\mathcal{H}^{d,3}/\mathbf{S}^2$.

Como $Y_d^{-m} = (Y_d^m)^*$ [24], o resultado é verdadeiro também para $m < 0$. Com isso concluímos que $\mathcal{Y}^d \subseteq \mathcal{P}^{d,3}/\mathbf{S}^2$.

Vamos agora mostrar que os dois espaços têm a mesma dimensão. Note que para cada k, as funções $Re[Y_k^m]$, $m = 0 \ldots k$ e $Im[Y_k^m]$, $m = 1 \ldots k$ são linearmente independentes [16]. Note também que para $m < 0$ vale $Re[Y_k^{-m}] = Re[Y_k^m]$ e $Im[Y_k^{-m}] = -Im[Y_k^m]$. Portanto, a dimensão do espaço $\mathcal{Y}^d$ é pelo menos $\sum_{k=0}^{d}(2k+1) = (d+1)^2$. Como visto anteriormente $\dim \mathcal{P}^{d,3}/\mathbf{S}^2$ também é $(d+1)^2$, logo os dois espaços coincidem. **CQD**

Capítulo 3

Splines Esféricos

Conforme discutido no capítulo 1, para fins de aproximação e modelagem de funções, é desejável que o espaço de funções usado admita uma *base local*: uma base onde cada elemento tenha valor praticamente nulo na esfera toda, exceto numa região pequena. Um espaço de funções que satisfaz esses requisitos é o dos *splines esféricos*, que definiremos a seguir.

3.1 Splines esféricos

3.1.1 Triangulações da esfera $\mathbf{S}^{n-1}$

Um *cone simplicial* do $\mathbf{R}^n$ é um subconjunto convexo de $\mathbf{R}^n$, com interior não vazio, delimitado por n hiperplanos que passam pela origem. Seja T uma decomposição de $\mathbf{R}^n$ em cones simpliciais $T_1, T_2, \ldots, T_k$, com interiores disjuntos dois a dois. Tal coleção determina uma decomposição de $\mathbf{S}^{n-1}$ em *simplexos esféricos*, que chamaremos de *subdivisão simplicial esférica* induzida por T e denotaremos por $T/\mathbf{S}^{n-1}$. Por abuso de linguagem usaremos os termos *decomposição triedal* para significar uma decomposição em cones simpliciais; e *triangulação esférica* como sinônimo de *subdivisão simplicial esférica.*

3.1.2 Espaços de splines esféricos

Para uma decomposição simplicial T de $\mathbf{R}^n$, definimos os seguintes espaços de funções de $\mathbf{R}^n$ para $\mathbf{R}$:

$$\begin{aligned}
\mathcal{P}^{d,n}[T] &= \{\ p : (\forall i)\ p/T_i \in \mathcal{P}^{d,n}/T_i\ \} \\
\mathcal{H}^{d,n}[T] &= \{\ h : (\forall i)\ h/T_i \in \mathcal{H}^{d,n}/T_i\ \}
\end{aligned}$$

As restrições destas funções à esfera $\mathbf{S}^{n-1}$, ou seja as funções de $\mathcal{P}^{d,n}[T]/\mathbf{S}^{n-1}$ e $\mathcal{H}^{d,n}[T]/\mathbf{S}^{n-1}$ são chamadas de *splines esféricos*, respectivamente *gerais* e *homogêneos*, por analogia com os splines de uma variável [18].

3.1.3 Caracterização dos splines esféricos

O teorema 1 pode ser estendido trivialmente aos splines esféricos.

Corolário 5

$$\mathcal{P}^{d,n}[T]/\mathbf{S}^{n-1} = \mathcal{H}^{d-1,n}[T]/\mathbf{S}^{n-1} + \mathcal{H}^{d,n}[T]/\mathbf{S}^{n-1}$$

Mais ainda, podemos mostrar o seguinte:

Teorema 5 *Se $p \in \mathcal{H}^{d-1,n}[T] + \mathcal{H}^{d,n}[T]$ com $n \geq 2$, e $p \equiv 0 \pmod{\mathbf{S}^{n-1}}$ então $p = 0$.*

Demonstração:

Seja $p \in \mathcal{H}^{d-1,n}[T] + \mathcal{H}^{d,n}[T]$ tal que $n \geq 2$ e $p \equiv 0 \pmod{\mathbf{S}^{n-1}}$. Para todo $i \in \{1 \ldots k\}$, seja $p_i \in \mathcal{H}^{d-1,n} + \mathcal{H}^{d,n}$ tal que $p/T_i = p_i/T_i$. Então, temos que

$$p_i/(\mathbf{S}^{n-1} \cap T_i) = 0$$

Como $T_i \cap \mathbf{S}^{n-1}$ é um subconjunto de $\mathbf{S}^{n-1}$ com dimensão $n-1$, e $\mathbf{S}^{n-1}$ é uma variedade irredutível de $\mathbf{R}^n$ quando $n \geq 2$ [5], concluímos que $p_i/\mathbf{S}^{n-1} = 0$, ou seja p_i é nulo em toda a esfera $\mathbf{S}^{n-1}$. Pelo teorema 1, concluimos que $p_i = 0$. Como esta igualdade vale para todo T_i, concluímos que $p = 0$. **CQD**

Deste teorema seguem:

Corolário 6 *Se $p, q \in \mathcal{H}^{d-1,n}[T] + \mathcal{H}^{d,n}[T]$ e $n \geq 2$, então $p \equiv q \pmod{\mathbf{S}^{n-1}}$ se e somente se $p = q$.*

Corolário 7

$$\mathcal{H}^{d-1,n}[T]/\mathbf{S}^{n-1} \cap \mathcal{H}^{d,n}[T]/\mathbf{S}^{n-1} = \{0\}/\mathbf{S}^{n-1}$$

Corolário 8

$$\mathcal{P}^{d,n}[T]/\mathbf{S}^{n-1} = \mathcal{H}^{d-1,n}[T]/\mathbf{S}^{n-1} \oplus \mathcal{H}^{d,n}[T]/\mathbf{S}^{n-1}$$

3.2 Restrições de continuidade

Vamos agora estender o corolário 8 a splines esféricos sujeitos a restrições de continuidade nas fronteiras entre as partes.

Dizemos que uma função de $\mathbf{S}^{n-1}$ para $\mathbf{R}$ é *contínua de ordem zero* se ela é contínua no sentido ordinário; e é *contínua de ordem* k, para $k > 0$, se ela é contínua, diferenciável, e cada componente de seu gradiente esférico é uma função contínua de ordem $k-1$. Nós denotamos por $\mathbf{C}_k(\mathbf{S}^{n-1})$ o conjunto de todas as funções de $\mathbf{S}^{n-1}$ para $\mathbf{R}$ que são contínuas de ordem k.

Para uma decomposição triedral T de $\mathbf{R}^n$ definimos os espaços:

$$\begin{aligned}\mathcal{P}_k^{d,n}[T]/\mathbf{S}^{n-1} &= \{\ p : p \in \mathcal{P}^{d,n}[T]/\mathbf{S}^{n-1} \ \wedge\ p/\mathbf{S}^{n-1} \in \mathbf{C}_k(\mathbf{S}^{n-1})\ \} \\ \mathcal{H}_k^{d,n}[T]/\mathbf{S}^{n-1} &= \{\ h : h \in \mathcal{H}^{d,n}[T]/\mathbf{S}^{n-1} \ \wedge\ h/\mathbf{S}^{n-1} \in \mathbf{C}_k(\mathbf{S}^{n-1})\ \}\end{aligned}$$

Nosso objetivo é mostrar que $\mathcal{P}_k^{d,n}[T]/\mathbf{S}^{n-1}$ é soma direta de $\mathcal{H}_k^{d-1,n}[T]/\mathbf{S}^{n-1}$ e $\mathcal{H}_k^{d,n}[T]/\mathbf{S}^{n-1}$. Em outras palavras, a imposição de continuidade de ordem k na esfera $\mathbf{S}^{n-1}$ sobre o espaço $\mathcal{P}^{d,n}[T]/\mathbf{S}^{n-1}$ equivale a imposições independentes de continuidade de ordem k sobre cada um dos subespaços $\mathcal{H}^{d-1,n}[T]/\mathbf{S}^{n-1}$ e $\mathcal{H}^{d,n}[T]/\mathbf{S}^{n-1}$. Para tal necessitamos dos seguintes resultados:

Definição 1 *Um* círculo máximo *de* $\mathbf{S}^{n-1}$ *é qualquer subconjunto* $\mathbf{S}^{n-1} \cap H$ *tal que* H *é um hiperplano de* $\mathbf{R}^n$ *que contém a origem.*

Lema 3 *Seja* $P \in \mathcal{P}^{d,n}$ *para* $n \geq 3$, C *um círculo máximo de* $\mathbf{S}^{n-1}$, *e* X *um subconjunto não vazio de* C *com dimensão* $n-2$. *Se* P *é nula em* X, *então* P *é nula em todo* C.

Demonstração:

> Podemos, sem perda de generalidade, supor que C é a intersecção de $\mathbf{S}^{n-1}$ com o hiperplano $x_n = 0$. Se $n \geq 3$ então C é uma variedade irredutível de $\mathbf{R}^{n-1}$, e a tese segue por um resultado clássico de geometria algébrica [7, 5].
> **CQD**

Para provar o resultado principal, começaremos obtendo uma caracterização do espaço $\mathcal{P}_0^{d,n}[T]/\mathbf{S}^{n-1}$.

Teorema 6 *Se $n \geq 3$, então*

$$\mathcal{P}_0^{d,n}[T]/\mathbf{S}^{n-1} = \mathcal{H}_0^{d-1,n}[T]/\mathbf{S}^{n-1} \oplus \mathcal{H}_0^{d,n}[T]/\mathbf{S}^{n-1}$$

Demonstração:

($\supseteq$): Trivial.

($\subseteq$): Seja $p \in \mathcal{P}_0^{d,n}[T]/\mathbf{S}^{n-1}$. Para todo par de cones adjacentes T_i e T_j de T, sejam p_i e p_j funções de $\mathcal{P}^{d,n}$ tais que $p/T_i = p_i/T_i$ e $p/T_j = p_j/T_j$. Pelo corolário 5,

$$p = h^{'} + h^{''}$$

com $h^{'} \in \mathcal{H}^{d-1}[T]/\mathbf{S}^{n-1}$ e $h^{''} \in \mathcal{H}^{d}[T]/\mathbf{S}^{n-1}$. Uma vez que $p/T_i = p_i/T_i$, podemos escrever

$$p_i = h_i^{'} + h_i^{''}$$
$$p_j = h_j^{'} + h_j^{''}$$

onde $h_i^{'}/T_i$ e $h_j^{'}/T_j$ estão em $\mathcal{H}^{d-1,n}/T_j$, e $h_i^{''}/T_i$ e $h_j^{''}/T_j$ estão em $\mathcal{H}^{d,n}/T_j$.

Seja X a fronteira comum aos triângulos esféricos $T_i \cap \mathbf{S}^{n-1}$ e $T_j \cap \mathbf{S}^{n-1}$, e seja C o círculo máximo que contém o conjunto X. Podemos, sem perda de generalidade, supor que C é a esfera $\mathbf{S}^{n-2}$ contida no hiperplano π de equação $x_n = 0$ e definida pela equação $x_1^2 + x_2^2 + \ldots + x_{n-1}^2 = 1$. Uma vez que $p \in \mathbf{C}_0(\mathbf{S}^{n-1})$, temos $p/X = p_i/X = p_j/X$, e portanto $(p_i - p_j)/X = 0$. Como X é um subconjunto de C de dimensão $n-2$, pelo lema 3, concluímos que $p_i - p_j$ é nula em todo C, ou seja $(p_i - p_j)/C = 0$. Uma vez que $(p_i - p_j)/\pi \in \mathcal{P}^{d,n-1}$ e $(p_i - p_j)/C = 0$, e $n-1 \geq 2$ podemos, pelo corolário 8, concluir que

$$\begin{aligned}(h_i^{'} - h_j^{'})/C &= 0 \\ (h_i^{''} - h_j^{''})/C &= 0\end{aligned}$$

Portanto,

$$\begin{aligned}h_i^{'}/C &= h_j^{'}/C \\ h_i^{''}/C &= h_j^{''}/C\end{aligned}$$

Concluímos portanto que $h^{'} \in \mathcal{H}_0^{d-1}[T]/\mathbf{S}^{n-1}$ e $h^{''} \in \mathcal{H}_0^{d}[T]/\mathbf{S}^{n-1}$.

CQD

Observe que o teorema não vale quando $n = 2$, pois é possível obter uma função contínua, polinomial por partes em $\mathbf{S}^1$, cujas componentes homogêneas $h^{'}$ de grau $d-1$ e $h^{''}$ de grau d não são contínuas. Por exemplo

$$P(x,y) = \begin{cases} y & se \quad x>0,\, y>0 \\ 1 & se \quad x<0,\, y>0 \\ -x & se \quad x<0,\, y<0 \\ 0 & se \quad x>0,\, y<0 \end{cases}$$

Ou seja $\mathcal{H}_0^{d-1,2}[T]/\mathbf{S}^1 \oplus \mathcal{H}_0^{d,2}[T]/\mathbf{S}^1 \neq \mathcal{P}_0^{d,2}[T]/\mathbf{S}^1$.
Agora podemos finalmente estender o corolário 8 para splines esféricos com continuidade de ordem k geral:

Teorema 7 *Para $n \geq 3$ e todo $k \geq 0$,*

$$\mathcal{P}_k^{d,n}[T]/\mathbf{S}^{n-1} = \mathcal{H}_k^{d-1,n}[T]/\mathbf{S}^{n-1} \oplus \mathcal{H}_k^{d,n}[T]/\mathbf{S}^{n-1}$$

Demonstração:

($\supseteq$): Trivial.

($\subseteq$): Nós provaremos esta parte por indução em k. Como o caso $k = 0$ é o teorema 6, vamos supor $k > 0$.

Seja p uma função em $\mathcal{P}_k^{d,n}[T]/\mathbf{S}^{n-1}$. Por definição, p é contínua, e ∇p é definido e contínuo de ordem $k-1$. Pelo corolário 4, cada componente $(\nabla p)_\alpha$ de ∇p pertence a $\mathcal{P}^{d+2,n}[T]/\mathbf{S}^{n-1}$, e portanto a $\mathcal{P}_{k-1}^{d+2,n}[T]/\mathbf{S}^{n-1}$. Por indução,

$$(\nabla p)_\alpha \in \mathcal{H}_{k-1}^{d+1,n}[T]/\mathbf{S}^{n-1} \oplus \mathcal{H}_{k-1}^{d+2,n}[T]/\mathbf{S}^{n-1} \tag{3.1}$$

Por outro lado, pelo corolário 5, $p = h^{'} + h^{''}$, onde $h^{'} \in \mathcal{H}^{d-1,n}[T]/\mathbf{S}^{n-1}$ e $h^{''} \in \mathcal{H}^{d,n}[T]/\mathbf{S}^{n-1}$. No interior de cada triângulo de T, o gradiente esférico de p é então

$$\nabla p = \nabla h^{'} + \nabla h^{''}$$

Pelo teorema 2,

$$(\nabla h^{'})_\alpha \in \mathcal{H}^{d+1,n}[T]/\mathbf{S}^{n-1} \tag{3.2}$$
$$(\nabla h^{''})_\alpha \in \mathcal{H}^{d+2,n}[T]/\mathbf{S}^{n-1} \tag{3.3}$$

Comparando a equação (3.1) com as equações (3.2) e (3.3), nós concluímos que

$$
\begin{aligned}
(\nabla h')_\alpha &\in \mathcal{H}_{k-1}^{d+1,n}[T]/\mathbf{S}^{n-1} &\quad (3.4)\\
(\nabla h'')_\alpha &\in \mathcal{H}_{k-1}^{d+2,n}[T]/\mathbf{S}^{n-1} &\quad (3.5)
\end{aligned}
$$

Como p é contínua, o teorema 6 implica que h' e h'' são contínuas também. Com as equações (3.4) – (3.5), nós concluímos que $h' \in \mathcal{H}_k^{d-1,n}[T]/\mathbf{S}^{n-1}$ e $h'' \in \mathcal{H}_k^{d,n}[T]/\mathbf{S}^{n-1}$. Concluímos portanto que

$$\mathcal{P}_k^{d,n}[T]/\mathbf{S}^{n-1} \subseteq \mathcal{H}_k^{d-1,n}[T]/\mathbf{S}^{n-1} + \mathcal{H}_k^{d,n}[T]/\mathbf{S}^{n-1}$$

Por outro lado

$$(\mathcal{H}_k^{d-1,n}[T]/\mathbf{S}^{n-1}) \cap (\mathcal{H}_k^{d,n}[T]/\mathbf{S}^{n-1}) \subseteq (\mathcal{H}^{d-1,n}[T]/\mathbf{S}^{n-1}) \cap (\mathcal{H}^{d,n}[T]/\mathbf{S}^{n-1}) = \{0\}$$

pelo corolário 1. **CQD**

Capítulo 4

Malhas Triangulares Irregulares

Vamos de agora em diante considerar apenas o caso da esfera $\mathbf{S}^2$, e portanto vamos omitir a dimensão $n = 3$ nas notações dos espaços $\mathcal{P}_k^{d,n}[T]/\mathbf{S}^2$ e $\mathcal{H}_k^{d,n}[T]/\mathbf{S}^2$. Neste capítulo estudaremos a topologia e geometria das triangulações simpliciais da esfera $\mathbf{S}^2$.

4.1 Triangulações da esfera

Seja $T_1, T_2, \ldots, T_k$ uma decomposição triedral de $\mathbf{R}^3$ conforme definida na seção 3.1.1. Neste caso tal decomposição é conhecida como *triangulação geodésica*.

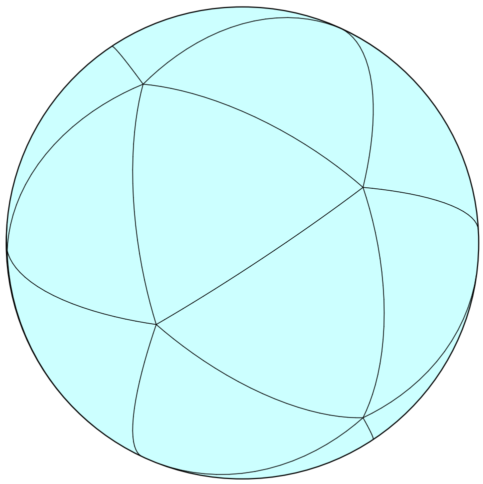


Figura 4.1: Triangulação geodésica.

Cada simplexo esférico $T_i \cap \mathbf{S}^2$ é um *triângulo esférico*, cujos lados são arcos de círculos máximos menores que 180^o. As fronteiras dos triângulos esféricos formam um grafo imerso na esfera [28]. Definimos o *grafo dual* da triangulação como como sendo o grafo que têm um vértice para cada triângulo, e uma aresta ligando dois vértices se os triângulos correspondentes têm um lado em comum.

Podemos construir uma triangulação tomando um conjunto de pontos que definem um poliedro regular (por exemplo, um icosaedro), dividindo regularmente suas faces em triângulos menores, e projetando o resultado sobre a esfera a partir da origem. A triangulação esférica assim obtida é razoavelmente uniforme. Outra maneira é tomar um conjunto qualquer de pontos e construir uma triangulação com vértices nesses pontos. Este segundo método permite construir uma *triangulação adaptativa* com maior concentração de triângulos em regiões de maior necessidade de precisão; basta para tanto usar um gerador de pontos aleatórios com probabilidade não uniforme.

4.2 Triangulação de Delaunay

Dentre as triangulações esféricas com um conjunto dado de vértices V, distingue-se a *triangulação de Delaunay* também conhecida como *diagrama de Delaunay*. Os triângulos dessa triangulação têm a propriedade de serem os mais eqüiláteros possíveis, num certo sentido [11]. Esta é uma característica importante para a interpolação de dados, modelagem de sólidos, cálculo de elementos finitos, e outras áreas da análise numérica, pois os erros cometidos na aproximação geralmente dependem da forma dos triângulos usados – isto é, para triangulações de tamanho fixo, quanto mais eqüiláteros eles forem, menores serão os erros.

Mostra-se que, na esfera, a triangulação de Delaunay de um conjunto de pontos V coincide com a projeção central sobre $\mathbf{S}^2$ do esqueleto da envoltória convexa desses pontos no $\mathbf{R}^3$. A envoltória convexa de um conjunto de pontos do $\mathbf{R}^3$ pode ser calculada em tempo $O(n \log n)$ [11]. (Como o custo da construção da triangulação de Delaunay é insignificante comparado com o custo de integração de uma equação diferencial, não foi necessário implementar um algoritmo ótimo. Em vez disso implementamos um algoritmo incremental, bem mais simples, cujo tempo é $O(n^2)$, descrito por Guibas e Stolfi [9].)

4.3 Funções de percurso

Para manusear uma triangulação, precisamos ter uma representação explícita de sua estrutura topológica, isto é, as relações de incidência entre os vértices, as arestas e as faces. Estes relacionamentos são descritos pelas *funções de percurso* da triangulação.

Seja e uma aresta orientada da triangulação. Denotamos por $eOrg$ e $eDest$ os vértices de origem e destino da aresta e. Analogamente, denotamos por $eLeft$ e $eRight$ as faces esquerda e direita, respectivamente; e por $eSym$ denotamos a mesma aresta tomada com orientação oposta.

As demais funções de percurso relacionam a aresta orientada e às arestas adjacentes. Para definir estas funções, ordenamos no sentido anti-horário as arestas que saem de ou chegam a um mesmo vértice.

O sentido anti-horário de rotação em torno de um ponto p da esfera deve ser entendido do ponto de vista de um observador no infinito, na direção de p, supondo que os eixos estão orientados de maneira convencional. Os conceitos de direita e esquerda devem ser entendidos da mesma forma. Em particular o sentido anti-horário em torno do ponto $(1, 0, 0)$, é o que roda da direção $(0, 1, 0)$ para $(0, 0, 1)$, pelo menor ângulo. O sentido anti-horário nos demais pontos da esfera é obtido por transporte contínuo desta orientação.

Seja e uma aresta com origem u e destino v, face esquerda f e face direita g. Denotaremos por $eOnext$ a próxima aresta com mesma origem u, no sentido anti-horário em torno de u. De mesma forma, denotaremos por $eDnext$ a próxima aresta com o mesmo destino v, no sentido anti-horário em torno de v.

Analogamente, por $eLnext$ a próxima aresta com a mesma face esquerda f, e por $eRnext$ a próxima aresta com a mesma face direita g, sempre no sentido anti-horário em torno dessas faces. Veja figura 4.2(a).

Definimos também as seguintes funções inversas:

$$eOprev = eOnext^{-1}$$

$$eLprev = eLnext^{-1}$$

$$eRprev = eRnext^{-1}$$

$$eDprev = eDnext^{-1}$$

A figura 4.2(b) ilustra estas funções.

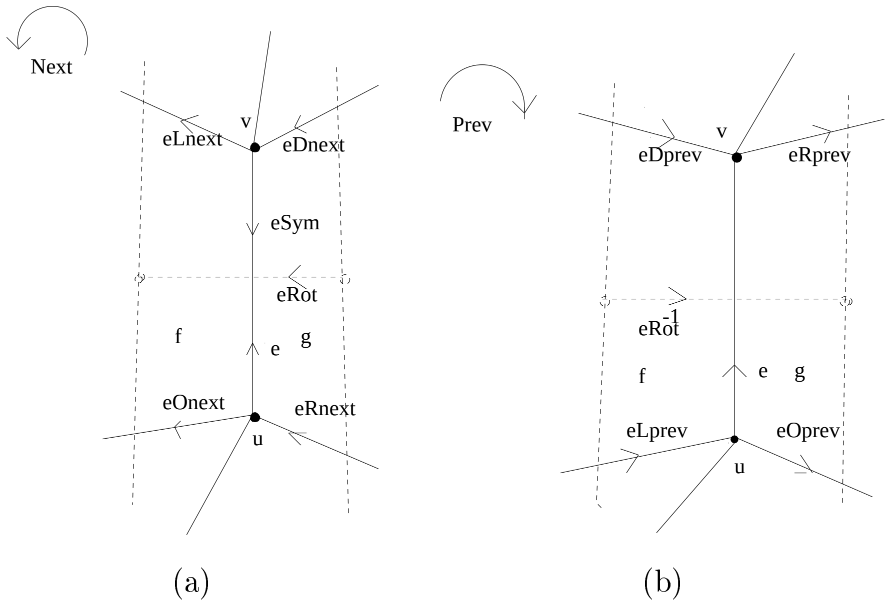


Figura 4.2: Funções de percurso (a) e suas inversas (b).

Pode-se verificar que as funções de percurso e suas inversas satisfazem as seguintes identidades:

$$eOprev = eSymLnext \tag{4.1}$$

$$eDprev = eSymOnext \tag{4.2}$$

$$eLprev = eOnextSym \tag{4.3}$$

$$eRprev = eSymOnext \tag{4.4}$$

$$eDnext = eSymOnextSym \tag{4.5}$$

$$eRnext = eSymLnextSym \tag{4.6}$$

4.4 Estrutura de dados para triangulação

Existem várias estruturas de dados para representar uma triangulação. No nosso trabalho usamos a estrutura *quad-edge* definida por Guibas e Stolfi [9]. Uma vantagem dessa estrutura é a habilidade de representar simultaneamente e uniformemente o grafo primal e o dual. Nesta estrutura, cada aresta orientada e, do grafo primal ou do dual, é representada por um registro com apontadores para os registros que representam a aresta $eOnext$ e a aresta correspondente $eRot$ do dual. A partir desses apontadores, podemos calcular todas as outras funções de percurso, usando as identidades (4.1) – (4.6). A estrutura quad-edge tem operadores topológicos que nos permitem construir e modificar qualquer triangulação.

A estrutura quad-edge básica não contém registros separados para vértices ou faces; topologicamente, estes elementos são definidos como anéis de arestas ligadas pelas funções $eOnext$ ou $eLnext$, respectivamente. Para representar a geometria da triangulação, acrescentamos ao registro de cada aresta e um campo adicional, que aponta para um registro que descreve o vértice de origem $eOrg$, e em particular contém as coordenadas cartesianas do mesmo.

4.5 Otimização da triangulação

A triangulação de Delaunay é em geral a melhor escolha quando o conjunto de vértices V é dado. Entretanto em muitas aplicações a escolha do conjunto V é parte do problema. Uma solução simples é escolher V aleatoriamente.

Para os capítulos que seguem, precisamos entretanto que a triangulação satisfaça certas condições. Uma delas, já mencionada, é que os triângulos sejam os mais eqüiláteros possíveis. Outra condição é que não haja duas arestas colineares, ou quase colineares incidentes no mesmo vértice. Para conseguir uma triangulação com estas características, usamos o seguinte método:

1. Geramos um conjunto V de pontos aleatórios, com a densidade desejada.

2. Construímos a triangulação de Delaunay.

3. Variamos as posições dos vértices, mantendo a topologia da triangulação, procurando minimizar uma função objetivo $U(V)$ que sintetiza todas as violações das condições acima na triangulação.

A função $U(V)$ que usamos é a soma de dois termos:

(a) um termo $U_e(u, v, w)$ para cada par de arestas (u, v), (v, w) incidentes no mesmo vértice v, tal que $U_e(u, v, w)$ tende a infinito quando os três vértices tendem a um mesmo plano passando plela origem.

(b) Um termo $U_t(a, b, c)$ para cada triângulo $t = (a, b, c)$ que é infinito quando a ordem dos três vértices a, b, c é horária, e é proporcional a p^2/A, caso contrário, onde p é o perímetro e A é a área do triângulo.

O conjunto de vértices que minimiza $U(V)$ é tal que a triangulação não possui os defeitos acima mencionados, e representa um certo equilíbrio entre os dois objetivos (ângulos grandes entre arestas incidentes em cada vértice, e triângulos aproximadamente eqüiláteros). Para minimizar a função U, utilizamos um procedimento padrão de otimização não linear [27].

Capítulo 5

Representação de Bézier para Splines

Quando trabalhamos com splines esféricos é conveniente representar o polinômio associado a cada triângulo na forma de Bézier relativa aos vértices do triângulo, descrita a seguir.

5.1 Coordenadas baricêntricas

Seja $v = (v_1, v_2, v_3)$ uma base qualquer de $\mathbf{R}^3$. Qualquer vetor $u \in \mathbf{R}^3$ tem uma única representação na forma

$$u = \hat{v}_1(u) \cdot v_1 + \hat{v}_2(u) \cdot v_2 + \hat{v}_3(u) \cdot v_3$$

Os coeficientes $\hat{v}_1, \hat{v}_2, \hat{v}_3$, são chamados *coordenadas baricêntricas de u relativas* à base v. São funções lineares homogêneas de u que satisfazem $\hat{v}_i(u) > 0$ para todo u no interior do triedro determinado por v_1, v_2 e v_3. Além disso,

$$\hat{v}_i(v_j) = \delta_{ij} \quad para \quad i, j = 1, 2, 3$$

A conversão entre coordenadas cartesianas (u_1, u_2, u_3) de um ponto u e suas coordenadas baricêntricas $(b_1, b_2, b_3) = (\hat{v}_1(u), \hat{v}_2(u), \hat{v}_3(u))$ relativas a uma base v pode ser escrita em forma matricial como

$$(u_1, u_2, u_3) = [b_1, b_2, b_3]M_v \quad \text{e} \quad (b_1, b_2, b_3) = [u_1, u_2, u_3]M_v^{-1}$$

onde M_v é uma matriz 3×3 cujas linhas são as coordenadas cartesianas dos vetores v_1, v_2, v_3.

5.2 Bases de Bézier

Dada uma base $v = (v_1, v_2, v_3)$ de $\mathbf{R}^3$ e um número inteiro não negativo d, considere as funções

$$B^v_{ijk}(u) = \frac{(i+j+k)!}{i!j!k!}\hat{v}^i_1(u)\hat{v}^j_2(u)\hat{v}^k_3(u) \quad i+j+k=d$$

Obviamente, cada $B^v_{ijk}(u)$ é uma função polinomial homogênea de $\mathbf{R}^3$ para $\mathbf{R}$ de grau $d = i+j+k$. Para qualquer base v, estas $\binom{d+2}{2}$ funções são linearmente independentes [8]. Portanto elas constituem uma base do espaço $\mathcal{H}^d$ e, por conseguinte, do espaço $\mathcal{H}^d/\mathbf{S}^2$. Ou seja, todo polinômio homogêneo p de $\mathcal{H}^d$ (ou de $\mathcal{H}^d/\mathbf{S}^2$) pode ser escrito na forma

$$p(u) = \sum_{i+j+k=d} c^v_{ijk} B^v_{ijk}(u) \tag{5.1}$$

com $c^v_{ijk} \in \mathbf{R}$. Os coeficientes c^v_{ijk} são chamados de *coeficientes de Bernstein-Bézier* do polinômio, *relativos* à base v.

5.2.1 Nodos de Bézier

A representação de Bézier pode ser melhor visualizada associando-se cada coeficiente c^v_{ijk} da mesma a um ponto u^v_{ijk} do triângulo plano t, cujos vértices são os vetores v_1, v_2, v_3 da base. O ponto u^v_{ijk} tem por definição coordenadas baricêntricas $(i,j,k)/(i+j+k)$ relativas à base v. A figura 5.1 ilustra esses pontos para $d = 5$. Esses pontos são ditos os *nodos de Bézier* de grau d associados ao triângulo t.

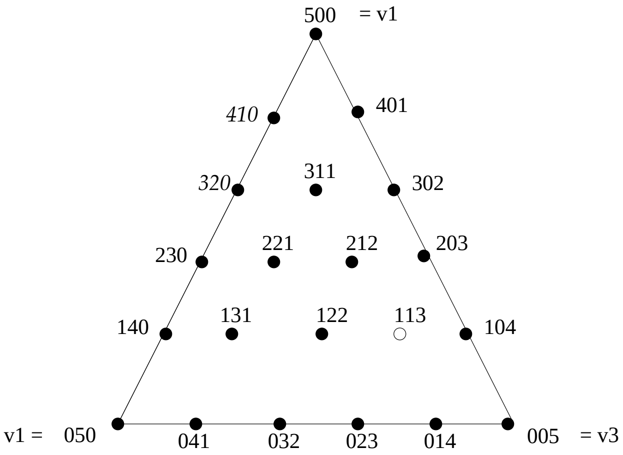


Figura 5.1: Nodos de Bernstein-Bézier u^v_{ijk} para $d = 5$.

A justificativa para esta associação é que o valor máximo de $B^v_{ikj}(u)$ sobre o triângulo t (ou sobre sua projeção central na esfera) está aproximadamente no ponto u^v_{ijk} (ou sobre sua projeção na esfera). Portanto, o efeito do coeficiente de Bézier c^v_{ijk} sobre o valor do polinômio $p(u)$, quando u pertence ao triângulo t, é máximo nas vizinhanças do ponto u^v_{ijk}.

Em particular, é fácil de ver que $p(v_1) = c^v_{d00}$, $p(v_2) = c^v_{0d0}$ e $p(v_3) = c^v_{00d}$.

5.3 Condições de continuidade

As condições sobre os coeficientes de Bézier para que dois polinômios se encontrem suavemente através do plano comum a dois triedros foram estabelecidas por Alfeld, Neamtu e Schumaker [1] usando um resultado clássico devido a de Boor [8].

Teorema 8 *Sejam t_v e t_w triedros com vértices v_1, v_2, v_3 e w_1, w_2, w_3 respectivamente, onde $w_2 = v_2$ e $w_3 = v_3$. Sejam p_v e p_w polinômios homogêneos de grau d, e sejam c^v_{ijk} e c^w_{ijk} seus coeficientes de Bezier relativos a t_v e t_w, respectivamente; ou seja,*

$$p_v(u) = \sum_{i+j+k=d} c^v_{ijk} B^v_{ijk}(u),$$

$$p_w(u) = \sum_{i+j+k=d} c^w_{ijk} B^w_{ijk}(u),$$

Então p_v e p_w e todas suas derivadas espaciais até ordem m se encontram suavemente na face comum aos triedos t_v e t_w se e somente se

$$c^w_{ijk} = \sum_{r+s+t=i} c^v_{r,j+s,k+t} B^v_{rst}(w_1)$$

para todo $i = 0, \ldots, m$ e todos j, k tais que $i + j + k = d$.

Em particular, para continuidade $\mathbf{C}_0$ temos apenas as seguintes condições:

$$c^w_{0jk} = c^v_{0jk}$$

para todo j, k tal que $j + k = d$.

Para continuidade $\mathbf{C}_1$ temos, adicionalmente, as equações:

$$c^w_{1jk} = c^v_{1jk} a_1 + c^v_{0,j+1,k} a_2 + c^v_{0,j,k+1} a_3$$

para todo j, k tal que $j + k = d - 1$, onde a_1, a_2, a_3 são números reais tais que $w_1 = \sum_{i=1}^3 a_i v_i$ (ou seja $a_i = \hat{v}_i(w_1)$). Por exemplo, na figura 5.2, para dois polinômios de grau 6 definidos nos triângulos v_1, v_2, v_3 e w_1, w_2, w_3, as condições de continuidade de ordem zero são:

$$\begin{aligned} c^w_{006} &= c^v_{006} \\ c^w_{015} &= c^v_{015} \\ c^w_{024} &= c^v_{024} \\ c^w_{033} &= c^v_{033} \\ c^w_{042} &= c^v_{042} \\ c^w_{051} &= c^v_{051} \\ c^w_{060} &= c^v_{060} \end{aligned} \tag{5.2}$$

Para continuidade de primeira ordem temos, além das equações de ordem zero (5.2), as equações:

$$\begin{aligned} c^w_{105} &= c^v_{105}a_1 + c^v_{015}a_2 + c^v_{006}a_3 \\ c^w_{114} &= c^v_{114}a_1 + c^v_{024}a_2 + c^v_{015}a_3 \\ c^w_{123} &= c^v_{123}a_1 + c^v_{033}a_2 + c^v_{024}a_3 \\ c^w_{132} &= c^v_{132}a_1 + c^v_{042}a_2 + c^v_{033}a_3 \\ c^w_{141} &= c^v_{141}a_1 + c^v_{051}a_2 + c^v_{042}a_3 \\ c^w_{150} &= c^v_{150}a_1 + c^v_{060}a_2 + c^v_{051}a_3 \end{aligned} \tag{5.3}$$

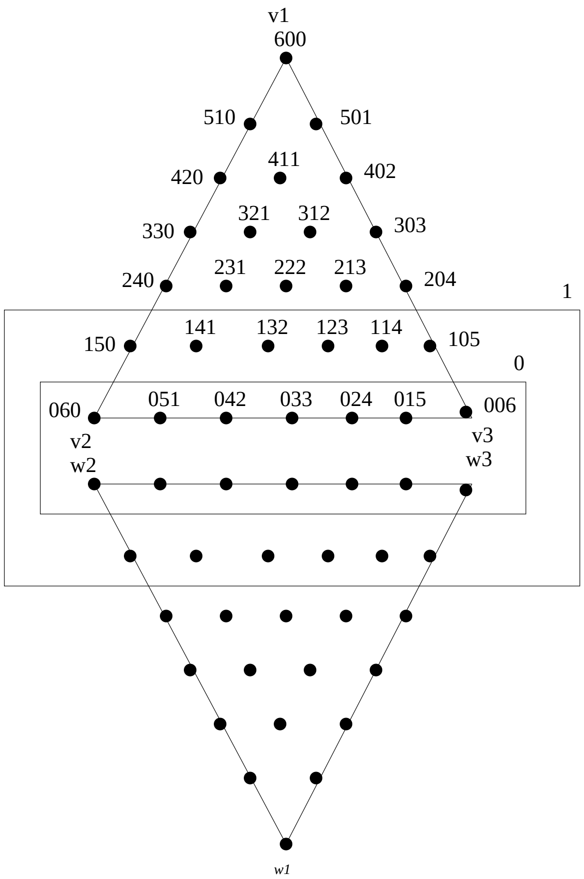


Figura 5.2: Condições de Continuidade $\mathbf{C}_0$ e $\mathbf{C}_1$.

5.4 Cálculo de splines esféricos

5.4.1 Algoritmo de DeCasteljau

Uma grande vantagem da representação de Bézier é que o valor de $p(u)$, para um ponto u qualquer de $\mathbf{R}^n$, pode ser avaliado eficientemente a partir da mesma representação usando o *algoritmo de DeCasteljau*. Este algoritmo recebe como entrada o conjunto $c^v = \{c^v_{ijk} : i+j+k = d\}$ dos coeficientes de Bernstein-Bézier do polinômio, relativos a uma base qualquer $v = (v_1, v_2, v_3)$, e as coordenadas $b = (b_1, b_2, b_3)$ de u relativas a esta mesma base, isto é $b_i = \hat{v}_i(u)$.

Algoritmo 1 *Procedimento* `BezierEval`(c^v, b)

1. `Para todo` i, j, k `com` $i + j + k = d$ `faça` $p_{ijk} \leftarrow c^v_{ijk}$
2. `Para` $m = d - 1, d - 2, \ldots 0$ `faça`

    2.1. `Para todo` i, j, k `com` $i + j + k = m$ `faça`

    2.1.1. $p_{ijk} \leftarrow b_1 p_{i+1,j,k} + b_2 p_{i,j+1,k} + b_3 p_{i,j,k+1}$

3. `devolva` p_{000}.

O passo 2.1.1. é executado $d(d+1)(d+2)/6$ vezes, portanto o custo deste algoritmo é $O(d^3)$. Obviamente este procedimento também pode ser usado para calcular o valor de um polinômio esférico $p \in \mathcal{H}^d$ num ponto u da esfera. Note que neste caso os vetores da base v não precisam ter comprimento unitário, desde que as coordenadas b e os coeficientes c^v_{ijk} sejam consistentes com os mesmos.

5.4.2 Cálculo do gradiente num ponto

Usando as técnicas de derivação automática, deduz-se, a partir do procedimento `BezierEval`, um algoritmo que calcula o vetor gradiente de uma função polinomial homogênea $p : \mathbf{R}^3 \to \mathbf{R}$, de grau d, num ponto u dado.

Este algoritmo recebe as mesmas entradas do procedimento *BezierEval*, ou seja, o conjunto $c^v = \{c^v_{ijk} : i + j + k = d\}$ dos coeficientes de Bernstein-Bézier do polinômio, e as coordenadas baricêntricas $b = (b_1, b_2, b_3)$ de u, ambas relativas à base $v = (v_1, v_2, v_3)$. O algoritmo retorna o vetor gradiente do polinômio p relativo às coordenadas baricêntricas, ou seja

$$(\nabla^v p)(u) = (\frac{\partial q}{\partial b_1} + \frac{\partial q}{\partial b_2} + \frac{\partial q}{\partial b_3})$$

onde $q(b_1, b_2, b_3) = p(b_1 v_1 + b_2 v_2 + b_3 v_3)$.

Algoritmo 2 *Procedimento* `BezierGrad`(c^v, b)

1. `Para todo` i, j, k `com` $i + j + k = d$ `faça` $p_{ijk} \leftarrow c^v_{ijk}$
2. `Para` $r = 1, 2, 3$ `faça` $d^r_{i,j,k} \leftarrow 0$
3. `Para` $m = d - 1, d - 2, \ldots 0$ `faça`

   3.1 `Para todo` i, j, k `com` $i + j + k = m$ `faça`

      3.1.1. $p_{ijk} \leftarrow b_1 p_{i+1,j,k} + b_2 p_{i,j+1,k} + b_3 p_{i,j,k+1}$
      3.1.2. $d^1_{ijk} \leftarrow p_{i+1,j,k} + b_1 d^1_{i+1,j,k} + b_2 d^1_{i,j+1,k} + b_3 d^1_{i,j,k+1}$
      3.1.3. $d^2_{ijk} \leftarrow p_{i,j+1,k} + b_1 d^2_{i+1,j,k} + b_2 d^2_{i,j+1,k} + b_3 d^2_{i,j,k+1}$
      3.1.4. $d^3_{ijk} \leftarrow p_{i,j,k+1} + b_1 d^3_{i+1,j,k} + b_2 d^3_{i,j+1,k} + b_3 d^3_{i,j,k+1}$

4. `devolva` $d^1_{000}, d^2_{000}, d^3_{000}$.

Assim como no algoritmo anterior, os passos 3.1.1 – 3.1.4 são executados $d(d+1)(4d-1)/6$ vezes, e portanto, o custo do algoritmo é $O(d^3)$. Como observado, este algoritmo calcula gradiente $\nabla^v p$ relativo às coordenadas na base v. Para calcular o gradiente ∇p em relação às coordenadas cartesianas de $u = (u_1, u_2, u_3)$, precisamos aplicar a regra da cadeia

$$\frac{\partial p}{\partial u_i} = \sum_{j=1}^{3} \frac{\partial p}{\partial b_j} \frac{\partial b_j}{\partial u_i}$$

Observe que $\partial u_i / \partial b_j$ é a i-ésima coordenada cartesiana do vetor v_j, ou seja, $\partial u_i / \partial b_j = (M_v^\top)_{ij}$. Portanto $\partial b_j / \partial u_i = (M_v^\top)^{-1}_{ij}$. Logo

$$(\nabla p)(u) = (\nabla^v p)(u)(M_v^\top)^{-1}$$

O procedimento `BezierGrad` permite calcular também o gradiente esférico $\bar{\nabla} p$ de um polinômio $p \in \mathcal{H}^d$. Basta calcular o vetor ∇p pela fórmula (5.4.2) e projetá-lo no plano tangente à esfera no ponto u, segundo a fórmula (2.6).

5.4.3 Localização de um ponto numa triangulação

Para calcular o valor de um spline $p \in \mathcal{P}^d[T]$ (ou de um spline esférico $p \in \mathcal{P}^d[T]/\mathbf{S}^2$) num ponto arbitrário u, precisamos localizar primeiro o triângulo t de T que contém u, e em seguida aplicar o algoritmo de *BezierEval* com os coeficientes do polinômio p restrito a t. Uma solução trivial para o primeiro problema é o algoritmo de *localização trivial* abaixo, que faz uma busca exaustiva sobre todos os triângulos.

Algoritmo 3 *Procedimento* `Locate`(u, T)

1. $d_{\min} \leftarrow +\infty$
2. `para cada triângulo` t `de` T `faça`
    2.1. $d \leftarrow bdist(u, t)$;
    2.2. `se` $d \leq d_{\min}$ `então` $t_{\min} \leftarrow t$; $d_{\min} \leftarrow d$;
    2.3. `se` $d_{\min} \leq 0$ `então retorna` $t_{\min}$
3. `retorna` $t_{\min}$

O procedimento `bdist`(u, t) calcula uma medida da distância do ponto u ao triângulo t, se u está fora de t; caso contrário devolve 0.

Algoritmo 4 *Procedimento* `bdist`(u, t)

1. `Sejam` $v = (v_1, v_2, v_3)$ `os vértices de` t;
2. $(b_1, b_2, b_3) \leftarrow (u_1, u_2, u_3) M_v^{-1}$;
3. `retorna max`$\{$`max`$\{0,\ -b_i\}$: $\quad i = 1 \dots 3\}$;

A razão para usar o procedimento *bdist*, em vez de simplesmente testar se u está no triedro definido pelo triângulo t, é que os erros de arredondamento no cálculo deste teste podem fazer com o teste falhe com todos os triângulos de T.

O procedimento `Locate` termina em tempo $O(n_t)$ para uma triangulação de n_t triângulos. Para agilizar os cálculos, é importante pré-calcular a matriz de conversão de base M_v^{-1} para cada triângulo de T. Note que a matriz M_v^{-1} também é necessária para o cálculo do gradiente esférico, como visto na seção 5.4.2.

5.4.4 Algoritmo de localização otimizado

Uma otimização, simples mas extremamente efetiva, que pode ser aplicada ao algoritmo acima é guardar, num campo da triangulação T, o triângulo $t_{\min}$ encontrado na chamada anterior, e testar inicialmente se u pertence a este triângulo. A justificativa para esta otimização é que u geralmente está muito próximo ao ponto procurado na chamada anterior, e portanto frequentemente está no mesmo triângulo.

Algoritmo 5 *Procedimento* `Locate`(u, T)

1. $d_{\min} \leftarrow bdist(u, t_{\min})$;
2. `se` $d_{\min} \leq 0$ `então retorna` $t_{\min}$
3. `para cada triângulo` t `de` T `exceto` $t_{\min}$, `faça`
    - 3.1. $d \leftarrow bdist(u, t)$;
    - 3.2. `se` $d \leq 0$ `então` $t_{\min} \leftarrow t$; $d_{\min} \leftarrow d$;
    - 3.3. `se` $d_{\min} \leq 0$ `então retorna` $t_{\min}$
4. `retorna` $t_{\min}$

Capítulo 6

Bases para Splines Esféricos

Os resultados anteriores mostraram que $\mathcal{P}_k^d[T]/\mathbf{S}^2$, o espaço de funções polinomiais por partes restritas a esfera com continuidade de ordem k, é a soma direta dos espaços $\mathcal{H}_k^d[T]/\mathbf{S}^2$ e $\mathcal{H}_k^{d-1}[T]/\mathbf{S}^2$. Neste capítulo vamos descrever bases explícitas para estes espaços.

6.1 As bases de Alfeld, Neamtu e Schumaker

Alfeld, Neamtu e Schumaker [1, 2, 4] obtiveram recentemente uma base explícita para cada espaço $\mathcal{H}_k^d[T]/\mathbf{S}^2$, quando $d \geq 3k+2$. Estas bases têm *suporte local*, o que significa que cada elemento da base é um spline esférico cujo suporte (região onde f não é identicamente nula) é um número limitado de triângulos: um único triângulo, ou um par de triângulos adjacentes, ou o conjunto dos triângulos incidentes a um único vértice de T (a *estrela* do vértice). A localidade é importante na prática, pois o custo de calcular o produto escalar $\langle f, g\rangle$ para um spline f e uma função arbitrária g é proporcional ao número de triângulos de suporte de f. Conjectura-se que não existem bases com suporte local para $d \leq 3k+1$.

6.1.1 Nodos críticos

Na construção das bases, utilizaremos um método geral, proposto por Alfeld, Neamtu e Schumaker, baseado na representação de Bernstein-Bézier. Do conjunto dos nodos de Bézier de todos os triângulos, escolhemos um subconjunto de *nodos críticos* $U = \{u_1, ..., u_m\}$, um para cada elemento b_i da base. O elemento b_i tem por definição coeficiente de Bezier 1 no nodo u_i, e 0 nos demais nodos críticos u_j, com $j \neq i$. Os coeficientes associados a nodos que não fazem parte do conjunto crítico U são calculados a partir dos coeficientes críticos, de modo a garantir continuidade $\mathbf{C}_1$ através das arestas de T.

6.1.2 Convenções gráficas

Nas figuras que seguem, vamos indicar os nodos críticos por $\bullet$. Os elementos do conjunto não crítico serão divididos em nodos que não participam das condições de continuidade, representados por $\circ$, e os envolvidos nessas condições, representados por $\diamond$.

Na representação das funções da base, as linhas são curvas de nível; o tom da cor indica o sinal da função, (vermelho = positivo, cinza = 0, e azul = negativo) e a intencidade da cor é proporcional ao valor absoluto.

6.2 Bases para $k = 0$

Para garantir continuidade de ordem 0, basta que os coeficientes de Bézier de nodos compartilhados por dois ou mais triângulos sejam iguais. Portanto, o conjunto crítico para os splines de $\mathcal{H}_0^d[T]/\mathbf{S}^2$ é simplesmente o conjunto de todos os nodos de todos os triângulos de T. É importante notar que os nodos da forma u_{0ij}^v, u_{i0j}^v, u_{ij0}^v com $i, j > 0$ são compartilhados entre dois triângulos adjacentes; estes pares de nodos compartilhados contribuem em apenas um elemento para o conjunto crítico. Da mesma maneira, os nodos da forma u_{00i}^v, u_{0i0}^v, u_{i00}^v são compartilhados entre os triângulos incidentes num mesmo vértice; todos estes nodos contribuem com apenas um elemento por vértice para o conjunto U.

Como todos os nodos são críticos, cada elemento da base de $\mathcal{H}_0^d[T]/\mathbf{S}^2$ é obtido atribuindo-se coeficiente de Bézier 1 para um dos nodos, e coeficiente 0 para todos os demais.

6.3 Bases para $k = 1$

Como veremos no capítulo 8, para resolução de equações diferenciais, precisamos de uma base com continuidade $\mathbf{C}_1$. Portanto, os espaços de grau mais baixo, com bases locais conhecidas, que atendem nossas necessidades são $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{H}_1^5[T]/\mathbf{S}^2$. Descreveremos a seguir a construção da bases desses espaços.
Os nodos críticos (e portanto os elementos das bases de $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{H}_1^5[T]/\mathbf{S}^2$) podem ser divididos em:

- nodos associados a triângulos de T;

- nodos associados a arestas de T;
- nodos associados a vértices de T.

6.3.1 Elementos associados a um triângulo de T

Sejam c_{ijk} os coeficientes de Bézier correspondentes a um triângulo t de T. Os nodos críticos associados a t são por definição os nodos u_{ijk} com $i > 1, j > 1, k > 1$. Temos $\binom{d-4}{2}$ elementos nessas condições. Portanto quando $d = 5$ não há nodos críticos nesta classe, e para $d = 6$ temos um único nodo crítico u_{222}. Neste caso fazemos $c_{222} = 1$, e $c_{ijk} = 0$ para todos os outros nodos do triângulo. Veja figuras (6.1) e (6.2).

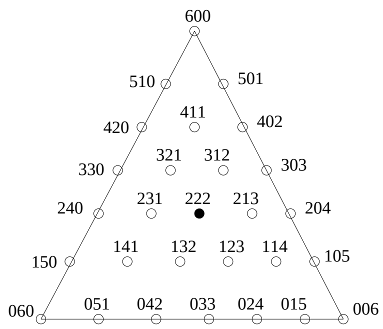


Figura 6.1: Nodo crítico associado a um triângulo (d = 6).

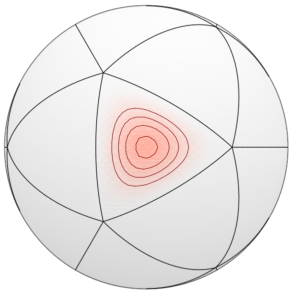
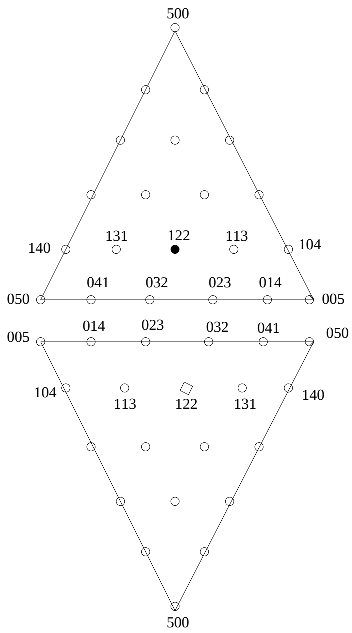


Figura 6.2: Elemento da base associado a um triângulo (d = 6).

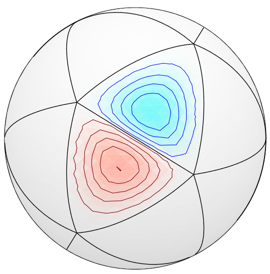
6.3.2 Elementos associados a uma aresta

Para cada aresta e de triangulação, sejam t_v e t_w triângulos adjacentes à aresta com vértices v_1, v_2, v_3 e w_1, w_2, w_3 respectivamente, onde $v_2 = w_3$, $v_3 = w_2$ (Note que esta numeração não coincide com a usada na seção 5.3). Sejam a_1, a_2, a_3 as coordenadas de w_1 relativas à base v_1, v_2, v_3, isto é, $w_1 = \sum_{i=1}^{3} a_i v_i$. Sejam c^v_{ijk} e c^w_{ijk} os coeficientes de Bézier relativos a base (v_1, v_2, v_3) e (w_1, w_2, w_3) respectivamente. Note que $c^v_{0ij} = c^w_{0ji}$ para todo i, j com $i + j = d$.



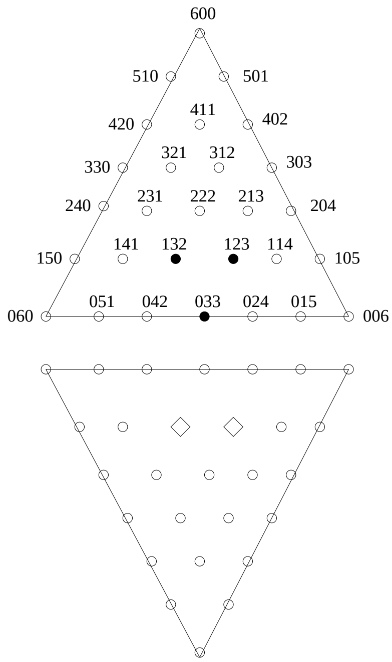
d = 5

Figura 6.3: Nodo crítico associado a uma aresta (d = 5).



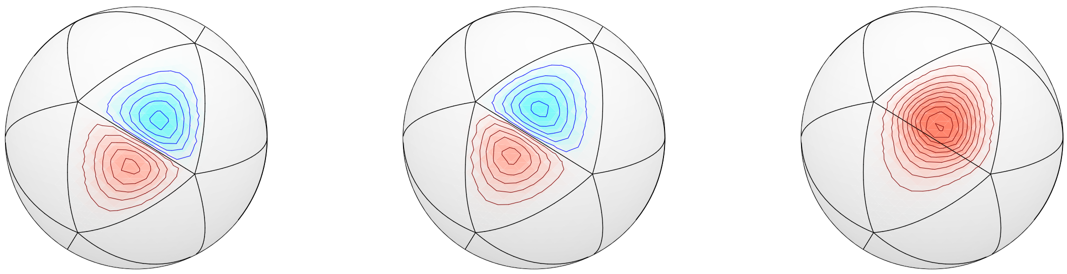
c^v_{122}

Figura 6.4: Elemento da base associado a uma aresta (d = 5).



d = 6

Figura 6.5: Nodos críticos associados a uma aresta (d = 6).



(a) c^v_{123} (b) c^v_{132} (c) c^v_{033}

Figura 6.6: Elementos da base associados a uma aresta (d = 6).

Os nodos críticos associados à aresta são os pontos u^v_{ijk} com $i \leq 1$, $j \leq d-3$ e $k \leq d-3$. Neste caso temos $2d-9$ elementos. Note que os nodos u^v_{0jk} e u^w_{0jk} coincidem e portanto são contados só uma vez. Portanto quando $d = 5$ temos um único nodo crítico u^v_{122} (figura 6.3).

O elemento correspondente da base tem $c^v_{122} = 1$, e $c^v_{ijk} = 0$ para todos os outros nodos $\{ijk\}$ do triângulo t_v. Para satisfazer a condição $\mathbf{C}_1$, precisamos tomar $c^w_{122} = a_1$. Todos os outros coeficientes c^w_{ijk} do triângulo t_w são nulos. Veja figura 6.4.
Quando $d = 6$ temos 3 nodos críticos u^v_{132}, u^v_{132} e u^v_{033} (figura 6.5). Neste caso os coeficientes c^w_{123} e c^w_{132} são determinados pela condição de continuidade $\mathbf{C}_1$. Os elementos da base são:

1. $c^v_{132} = 1$, $c^w_{123} = a_1$.

2. $c^v_{123} = 1$, $c^w_{132} = a_1$.

3. $c^v_{033} = 1$, $c^w_{132} = a_2$, $c^w_{123} = a_3$.

Todos os outros coeficientes c^v_{ijk} e c^w_{ijk} são nulos (figura 6.6).

6.3.3 Elementos associados a um vértice

Para cada vértice v de grau g temos $g+3$ elementos da base. Sejam $w_0, \ldots w_{g-1}$ os vértices vizinhos a v, e sejam $t_m, i = 0 \ldots g-1$ os triângulos incidentes a v enumerados no sentido anti-horário. Sejam u^m_{ijk} os nodos de Bézier associados ao triângulo t_m. Em todos os triângulos os nodos são numerados de tal maneira que os vértices v, w_m, w_{m+1} coincidem com os nodos u^m_{d00}, u^m_{0d0} e u^m_{00d}, respectivamente.

Os nodos críticos associados ao vértice v são u^0_{ijk} onde $j + k \leq 2$, e $u^m_{d-2,0,2}, m = 2 \ldots g-2$. Neste caso a quantidade e arranjo dos nodos críticos é a mesma para $d = 5$ e $d = 6$ (figura 6.7). Como antes, para cada elemento da base, atribuímos o valor 1 ao coeficiente crítico correspondente, e 0 aos demais coeficientes críticos(figura 6.8 e 6.9). Os demais coeficientes não críticos são nulos, exceto os que participam das condições de continuidade $\mathbf{C}_1$, indicados por $\diamond$ na figura 6.7. Neste caso temos um outro subconjunto de nodos que representaremos por $\times$. Cada elemento desse conjunto pode ser um nodo crítico associado à aresta mais próxima. Caso isso ocorra, seu coeficiente deve ser nulo; caso contrário, o coeficiente deve ser calculado pelas condições de continuidade. Felizmente há sempre um número suficiente de nodos '$\times$' para satisfazer todas as restrições de continuidade. Mais precisamente, esses nodos são considerados aos pares: $u^m_{d-3,1,2}$, $u^{m+1}_{d-3,2,1}$. O nodo do par que pertencer ao conjunto crítico terá coeficiente zero, e seu parceiro terá seu coeficiente calculado pela continuidade.

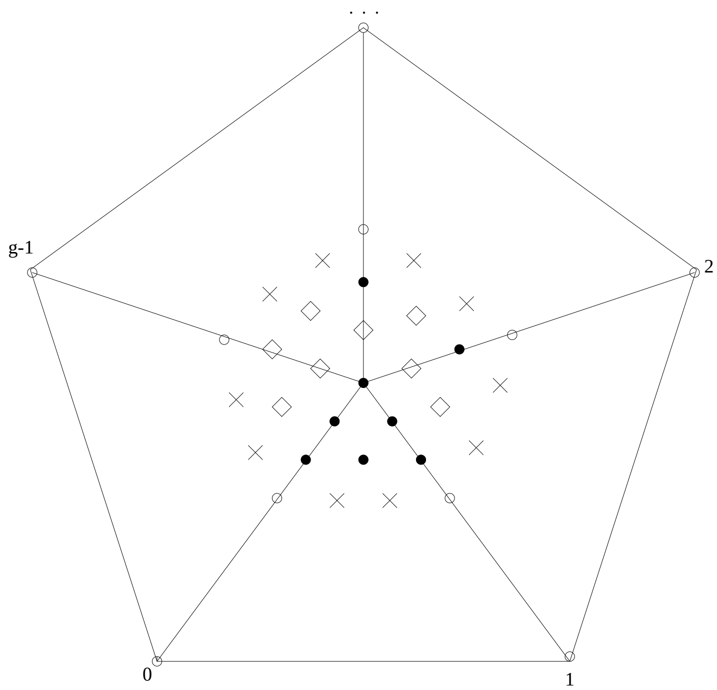


Figura 6.7: Nodos críticos associados a um vértice (d = 5) e (d = 6). Os nodos marcados com $\times$ podem ser nodos críticos associados às arestas, ou nodos calculados pela condição de continuidade(vide texto).

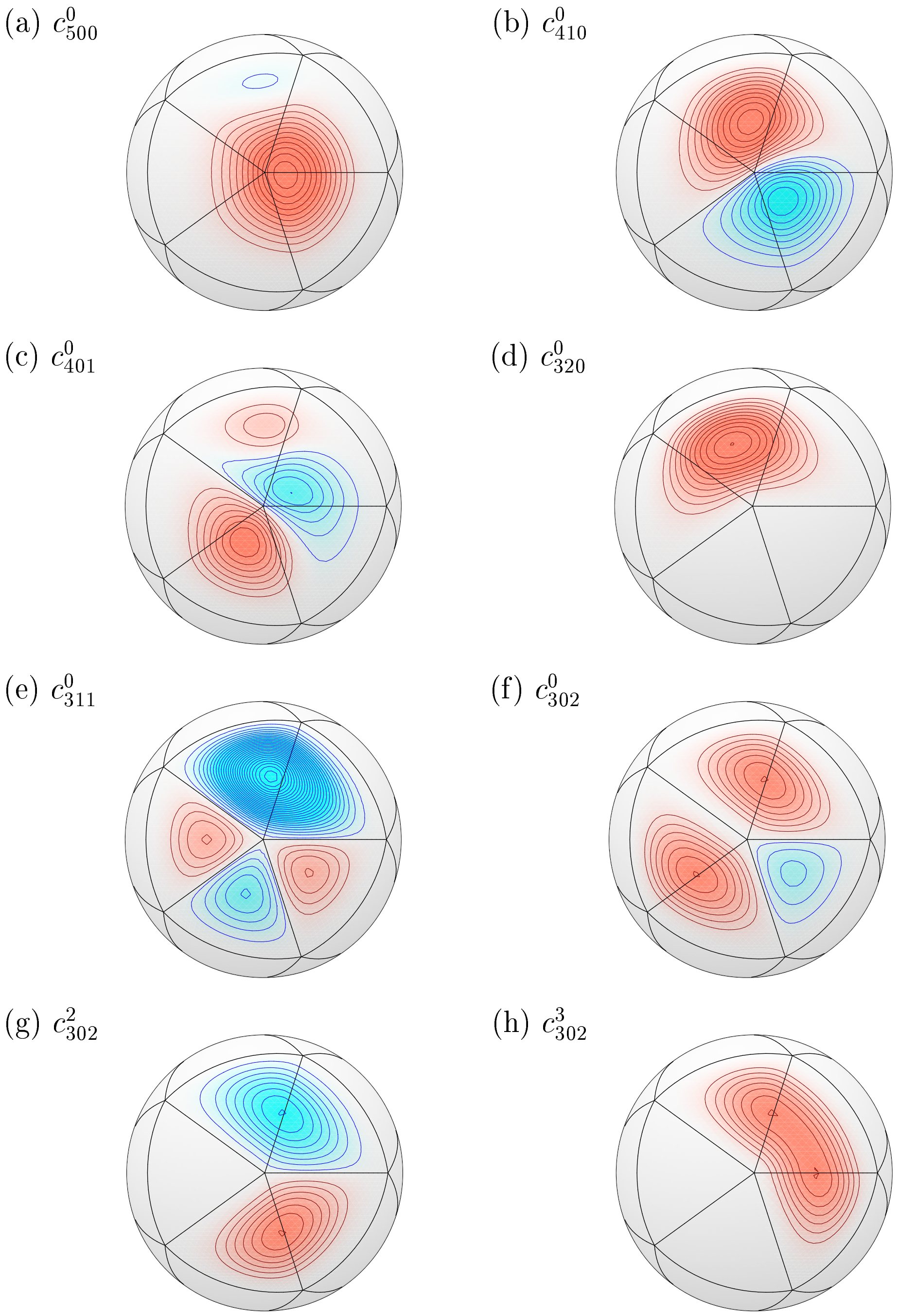
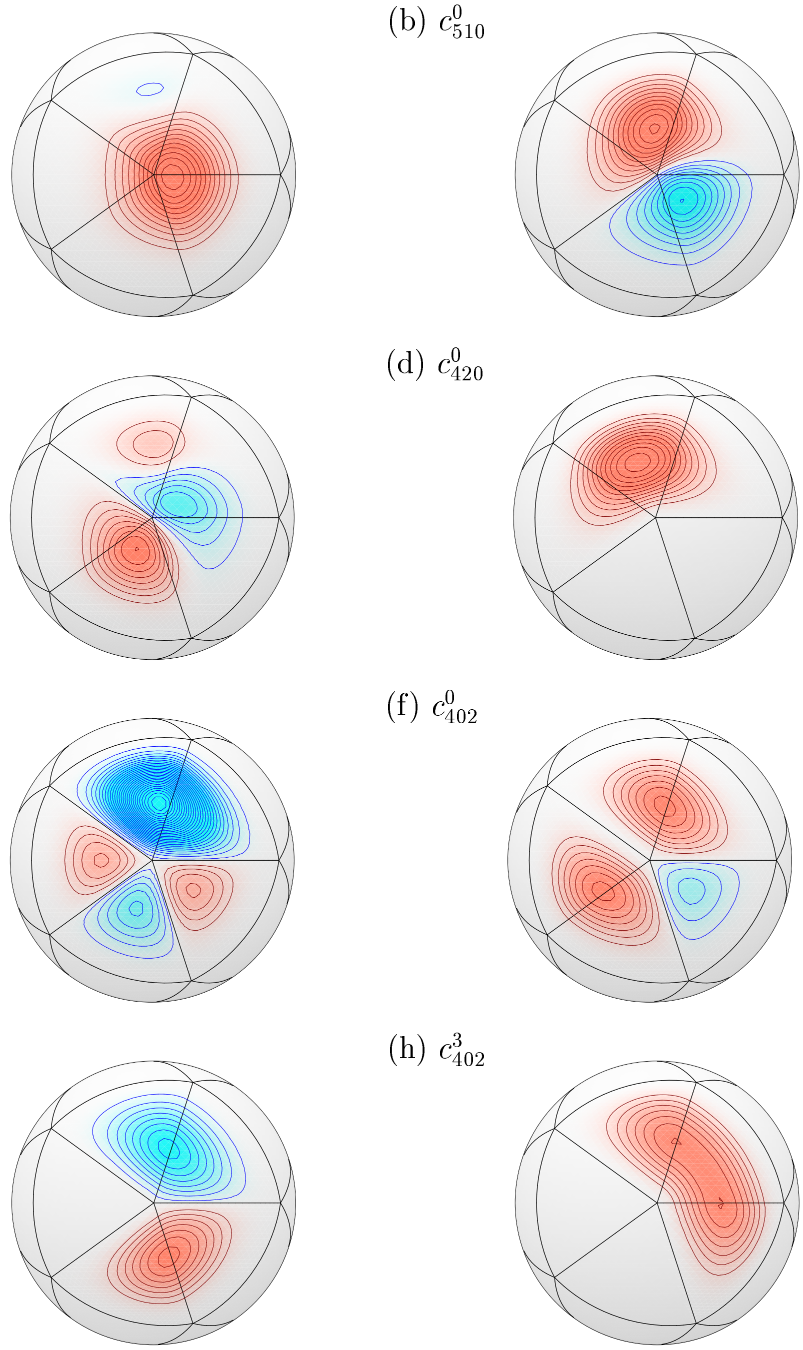


Figura 6.8: Os elementos da base associados a um vértice (d = 5).



(a) c^0_{600} (b) c^0_{510}

(c) c^0_{601} (d) c^0_{420}

(e) c^0_{411} (f) c^0_{402}

(g) c^2_{402} (h) c^3_{402}

Figura 6.9: Os elementos da base associados a um vértice (d = 6).

6.4 Independência dos elementos da base

Examinando os diagramas (6.1), (6.5) e (6.7), podemos verificar que:

- os conjuntos de coeficientes críticos ($\bullet$) indicados em cada diagrama são dois a dois disjuntos;
- todos os coeficientes não nulos calculados por continuidade ($\diamond$) e ($\times$) são disjuntos do conjunto crítico U.

Estas observações nos permitem concluir que cada elemento da base tem apenas um coeficiente crítico não nulo, e portanto eles são linearmente independentes.

6.5 Dimensões das bases de $\mathcal{H}_k^d[T]/\mathbf{S}^2$ e $\mathcal{P}_k^d[T]/\mathbf{S}^2$

Em vista desta construção, concluímos que para triangulações genéricas, a dimensão do espaço de splines homogêneos com continuidade de ordem k e grau $d \geq 3k+2$ é:

$$\begin{aligned} \dim \mathcal{H}_k^d[T]/\mathbf{S}^2 &= (d^2 - 3dk + 2k^2)n_v - 2d^2 + 6dk - 3k^2 + 3k + 2 \\ &= (d^2 - 3dk + 2k^2)n_t/2 + k^2 + 3k + 2 \end{aligned}$$

onde n_v e n_t são os números de vértices e triângulos da triangulação. Em particular, temos

$$\dim \mathcal{H}_0^2[T]/\mathbf{S}^2 \ = \ 4n_v - 6 \ = \ 2n_t + 2$$

$$\dim \mathcal{H}_0^3[T]/\mathbf{S}^2 \ = \ 9n_v - 16 \ = \ 9n_t/2 + 2$$

$$\dim \mathcal{H}_1^5[T]/\mathbf{S}^2 = 12n_v - 18 = 6n_t + 6$$

$$\dim \mathcal{H}_1^6[T]/\mathbf{S}^2 = 20n_v - 34 = 10n_t + 6$$

Em vista do Teorema 7, estes resultados nos dão também a dimensão do espaço $\mathcal{P}_k^d[T]/\mathbf{S}^2$. Note que uma base para esse espaço é simplesmente a concatenação de uma base de $\mathcal{H}_k^d[T]/\mathbf{S}^2$ e uma base de $\mathcal{H}_k^{d-1}[T]/\mathbf{S}^2$.

$$\begin{aligned} \dim \mathcal{P}_k^d[T]/\mathbf{S}^2 &= (2d^2 - 6dk + 4k^2 - 2d + 3k + 1)n_v - 4d^2 + 12dk + 4d - 6k^2 + 2 \\ &= (2d^2 - 6dk + 4k^2 - 2d + 3k + 1)n_t/2 + 2k^2 + 6k + 4 \end{aligned}$$

Em particular,

$$\dim \mathcal{P}_0^3[T]/\mathbf{S}^2 \ = \ 13n_v - 22 \ = \ 13n_t/2 + 4$$

$$\dim \mathcal{P}_1^6[T]/\mathbf{S}^2 = 32n_v - 52 = 16n_t + 12$$

6.6 Tornando as bases mais ortogonais

Um inconveniente da base descrita na seção anterior, é que seus elementos não são ortogonais no produto escalar natural $\langle f, g \rangle$. Isto pode ser observado por exemplo nas figuras 6.6(a)(b); muitos pares de elementos são bastante semelhantes portanto, seu produto é predominantemente positivo. Como veremos no capítulo 8, este fato retarda bastante a convergência de métodos iterativos baseados nestes espaços de splines, pois as matrizes de produtos escalares que surgem nesses problemas não são diagonalmente dominantes.

Podemos, em princípio, resolver este problema ortogonalizando a base descrita neste capítulo, pelo método Gram-Schmidt ou similar. Entretanto, esta ortogonalização destruiria a localidade da base, pois os elementos calculados por último seriam em geral combinações de todos os elementos anteriores.

O ideal seria construir uma base local ortogonal para $\mathcal{P}_k^d[T]/\mathbf{S}^2$. Porém nem sequer sabemos se tal base existe. Entretanto, podemos transformar a base descrita nas seções anteriores numa base um pouco mais próxima da ortogonalidade, sem aumentar o número de triângulos de suporte, aplicando à mesma um *algoritmo de ajuste* que leva em conta a estrutura dos elementos. O algoritmo está descrito a seguir.

No resto desta seção, vamos supor $k = 1$ e $d \geq 5$, e denotaremos por $\phi_1, \phi_2, \ldots, \phi_n$ os elementos da base Φ. Estes elementos podem ser classificados em sub-bases Φ_f, Φ_e e Φ_v, associadas a cada face f, aresta e e vértice v, respectivamente.

O algoritmo de ajuste da base Φ pode ser resumido nos seguintes passos:

- passo "F": Para cada face f, ortogonalize os elementos de Φ_f entre si. Seja Ω_f a base resultante.

- passo "E": Para cada aresta e, sejam f e g as faces adjacentes a e. (Note que os espaços $\langle \Omega_f \rangle$ e $\langle \Omega_g \rangle$ já são ortogonais entre si, pois seus suportes são disjuntos.) Torne os elementos de Φ_e ortogonais a Ω_f e Ω_g, (sem alterar estas últimas), e depois ortogonais entre si. Seja Ω_e a base resultante.

- passo "V": Para cada vértice v, sejam $f_1, ..f_m$ e $e_1, ..e_m$ as faces e arestas incidentes a v. Torne os elementos de Φ_v, ortogonais a todos os espaços Ω_{f_i} e Ω_{e_i} (sem alterar estas base) e em seguida ortogonais entre si. Seja Ω_v a base resultante.

O resultado do algoritmo é a concatenação Ω de todas as bases Ω_f, Ω_e, e Ω_v.

Note que a base Ω ainda não é ortogonal, pois nada garante a ortogonalidade entre os seguintes pares de espaços:

- Ω_u e Ω_v para dois vértices vizinhos u e v;
- Ω_v e Ω_e onde e é uma aresta da fronteira da estrela do vértice v;
- Ω_e e Ω_a onde e e a são arestas da mesma face.

Entretanto, espera-se que o ângulo entre esses espaços (determinado pelo produto escalar $\langle , \rangle$) seja bastante próximo de 90^o.

Observe também que o ajuste acima não muda o conjunto de triângulos-suporte, e portanto não afeta o tamanho da representação da base, nem o custo de calcular produtos escalares envolvendo elementos da mesma.

Os passos "E" e "V" acima podem ser feitos de muitas maneiras. O esquema proposto abaixo se baseia na divisão de Φ_e em sub-classes Φ_e^0 e Φ_e^1, conforme o nodo crítico correspondente ao elemento esteja sobre a aresta e ou fora dela. Note que Φ_e^1 tem d-4 elementos, e Φ_e^0 tem $d-5$ (um elemento a menos).

- passo "E": Para cada aresta e, sejam f e g as faces adjacentes a e. Torne os elementos de Φ_e^0 e Φ_e^1 ortogonais a Ω_f e Ω_g, sem alterar estas últimas, obtendo bases temporárias Θ_e^0 e Θ_e^1. Depois:
  - passo "E1": Torne os elementos de Θ_e^1 ortogonais entre si. Seja Ω_e^1 a base resultante.
  - passo "E0": Torne cada elemento de Θ_e^0 ortogonal a Θ_e^1, sem alterar estas últimas. Feito isso, torne esses elementos ortogonais entre si. Seja Ω_e^0 a base resultante.

Note que os elementos do espaço Φ_e^1 são identicamente nulos sobre a aresta e. Uma vez que os elementos de face Φ_f, Φ_g (e Ω_f, Ω_g) também se anulam sobre essa aresta, o método acima preserva esta propriedade nas bases Ω_e^1. Intuitivamente, os elementos de Ω_e^1 controlam a derivada da função transversalmente à aresta, sem afetar seu valor sobre a mesma.

Na base descrita por Alfeld, Neamtu e Schumaker, o papel dos elementos Φ_e^0 não é fácil de descrever, pois eles afetam tanto o valor da função sobre a aresta e, quanto sua derivada transversal. Entretanto, depois de ortogonalizados em relação a Ω_e^1 no passo "E1", os elementos de Ω_e^0 tendem a ser mais simétricos em relação à aresta, e portanto a controlar o valor da função com relativa independência de sua derivada.

Igualmente, o passo "V" pode ser realizado de muitas maneiras. O esquema abaixo usa o fato que um elemento da classe Φ_v pode ser classificado nas classes Φ_v^0,

Φ_v^1 e Φ_v^2, conforme o maior índice do seu coeficiente crítico seja d, $d-1$, ou $d-2$ respectivamente. Lembremos que a classe Φ_v^0 tem apenas 1 elemento, a classe Φ_v^1 tem dois elementos, e a classe Φ_v^2 tem m elementos, onde m é o número de arestas incidentes no vértice.

- passo "V": Para cada vértice v, sejam $f_1,..f_m$ e $e_1,..e_m$ as faces e arestas incidentes a v. Torne os elementos de Φ_v^0, Φ_v^1, Φ_v^2 ortogonais a todos os espaços Ω_{f_i} e Ω_{e_i}. Sejam Θ_v^0, Θ_v^1, Θ_v^2 as bases resultantes.
    - passo "V2": Torne os elementos de Θ_v^2 ortogonais entre si. Seja Ω_v^2 a base resultante.
    - passo "V1": torne os dois elementos de Θ_v^1 ortogonais a Ω_v^2, e entre si. Seja Ω_v^1 a base resultante.
    - passo "V0": torne o único elemento de Θ_e^0 ortogonal ao espaço gerado por Ω_v^2 e Ω_v^1. Seja Ω_v^0 a base resultante.

Note que os elementos da base Φ_v^2, bem como todos os elementos associados a arestas e vértices, têm valor e gradiente nulos no vértice v. Portanto esta propriedade de Φ_v^2 é preservada pela ortogonalização contra Ω_{f_i} e Ω_{e_i}, e permanece válida para os elementos da base Ω_v^2.

Analogamente, os elementos da base Φ_v^1 têm valor nulo em v, mas gradientes não-nulos e linearmente independentes. Estas propriedades são preservadas pelo passo "V1", e portanto os elementos de Ω_v^1 controlam o gradiente da função em v, sem afetar seu valor.

O elemento Φ_v^0 é mais difícil de explicar, pois ele controla tanto o valor quanto o gradiente em v. Entretanto, a ortogonalização contra Ω_v^1 deve fazer com que ele tenha mais efeito no valor do que no gradiente.

6.7 Triangulações degeneradas

A construção da base para $k = 1$ (seção 6.3) pressupõe que a triangulação não seja geometricamente degenerada. Em particular, se duas das arestas incidentes a um vértice v são coplanares, surge uma dependência linear entre os elementos associados a esse vértice pela construção da secção 6.3 [2]. Para evitar esta situação usamos o método de otimização da triangulação, descrito na seção 4.5.

Capítulo 7

Aproximação de Funções por Splines Esféricos

Neste capítulo, analisaremos o uso dos espaços de splines esféricos, $\mathcal{P}_k^d[T]/\mathbf{S}^2$ e $\mathcal{H}_k^d[T]/\mathbf{S}^2$ para aproximar funções restritas à esfera $\mathbf{S}^2$. Mostraremos que, com o espaço $\mathcal{P}_k^d[T]/\mathbf{S}^2$, podemos obter uma melhor aproximação dessas funções do que com os espaços $\mathcal{H}_k^d[T]/\mathbf{S}^2$.

7.1 Aproximação por mínimos quadrados

Seja f uma função real sobre a esfera, e $\{\phi_i\}_1^n$ uma base de um espaço $\mathcal{F}$ de funções reais sobre $\mathbf{S}^2$. Queremos encontrar uma função $u(p) = \sum_{i=1}^n x_i \phi_i(p)$ que melhor se aproxima da f no sentido que $\| u(p) - f(p) \|^2$ seja mínima. Aqui $\| f \|^2 = \langle f, f \rangle$ sendo $\langle , \rangle$ um produto interno definido no espaço. Este é o *método dos mínimos quadrados* [19] cuja solução recai em um sistema de equações lineares $Gx = b$, chamado *sistema normal*. Neste sistema as incógnitas $x = (x_1, x_2 \dots x_n)$ são os coeficientes da aproximação de $u(p)$ na base $\{\phi_i\}_1^n$; os elementos da matriz são $G_{ij} = \langle \phi_i(p), \phi_j(p) \rangle$, $i, j = 1 \dots n$; e os elementos do vetor independente b são $b_i = \langle f(p), \phi(p) \rangle$, $i = 1 \dots n$.

7.2 Solução do sistema $Gx = b$

Como $\{\phi_i\}_1^n$ é uma base, e $\langle , \rangle$ é um produto interno, a matriz G é simétrica positiva definida. Portanto, como visto em [18], o sistema $Gx = b$ tem solução única. O sistema pode ser resolvido usando métodos diretos ou métodos iterativos. Um método direto eficiente é o *algoritmo de Choleski* [17], que é uma variante do método de decomposição de Gauss especializada para matrizes simétricas e positivas definidas. Neste método, a matriz G é decomposta num produto $LL^\top$, onde L é uma matriz triangular inferior e $L^\top$ é a sua transposta. Dessa forma o sistema $Gx = b$ é transformado no sistema $LL^\top x = b$, cuja solução é encontrada em duas etapas: (1) calculamos y resolvendo o sistema $Ly = b$ e (2) calculamos x resolvendo o sistema $L^\top x = y$. Observe que estes dois sistemas podem ser resolvidos por substituições sucessivas.

O método iterativo mais simples é o método de *Gauss Seidel*, onde a solução x é obtida iterando-se a fórmula

$$x_i \leftarrow x_i + \omega \cdot (b_i - \sum_{j=1}^{n} G_{ij} \cdot x_j)/G_{ii}$$

para $i = 1 \ldots n$, a partir de uma estimativa inicial x arbitrária. A constante ω é um parâmetro que pode ser usado para controlar a velocidade de convergência.

Outro método iterativo é o método do *gradiente conjugado*, que procura minimizar $\| Gx - b \|^2$ usando técnicas de otimização quadrática [27].

7.3 Cálculo do produto escalar

É natural escolher $\langle f, g \rangle = \int_{\mathbf{S}^2} f(p)g(p)dp$ como produto interno para o espaço de funções definidas em $\mathbf{S}^2$. Seja T uma triangulação esférica qualquer, e f uma função esférica. É óbvio que

$$\int_{\mathbf{S}^2} f(p)dp = \sum_{t \in T} \int_t f(p)dp$$

Para o cálculo do produto escalar, precisamos saber integrar, em cada triângulo esférico, uma função definida em $\mathbf{S}^2$. Infelizmente não temos uma fórmula explícita para este cálculo, mesmo para funções simples como polinômios esféricos. Portanto, é necessário usar método de integração numérica.

Em cada triângulo esférico $t \in T$, seja $\bar{t}$ o triângulo plano com os mesmos vértices de t. Então para cada $t \in T$, podemos escrever

$$\int_t f(p)dp = \int_{\bar{t}} f(\frac{u}{|u|})w(u)du \tag{7.1}$$

onde $w(u) = dp/du$ é a *correção esférica*, a razão entre a área de um elemento du de $\bar{t}$ em torno do ponto u, e a área de sua projeção dp na esfera.

A integral à direita da equação (7.1) pode ser calculada por uma fórmula de quadratura de Gauss para um triângulo plano:

$$\int_{\bar{t}} g(u)du \simeq area(\bar{t}) \cdot \sum_{k=1}^{m} \alpha_k \cdot g(u_k) \tag{7.2}$$

onde $u_1 \ldots u_k$ constitui um conjunto de pontos de $\bar{t}$ com coordenadas baricêntricas fixas relativas aos vértices de t, e $\alpha_1 \ldots \alpha_k$ são pesos apropriados.

Demonstra-se que, para qualquer grau d, existe uma coleção finita de pontos fixos u_k tais que a fórmula (7.2) é exata se a função g for qualquer polinômio de grau $\leq d$ [20, 21]. Neste caso, dizemos que a fórmula (7.2) tem *ordem d*. Nos nossos testes, utilizamos o esquema da tabela 7.1, que tem ordem $d = 7$ [22].

Pesos	*Coordenadas Baricêntricas*		
−0.149570044467670	0.333333333333333	0.333333333333333	+0.333333333333333
+0.175615257433204	0.479308067841923	0.260345966079038	+0.260345966079038
+0.175615257433204	0.260345966079038	0.479308067841923	+0.260345966079038
+0.175615257433204	0.260345966079038	0.260345966079038	+0.479308067841923
+0.053347235608839	0.869739794195568	0.065130102902216	+0.065130102902216
+0.053347235608839	0.065130102902216	0.869739794195568	+0.065130102902216
+0.053347235608839	0.065130102902216	0.065130102902216	+0.869739794195568
+0.077113760890257	0.638444188569809	0.312865496004875	+0.048690315425316
+0.077113760890257	0.638444188569809	0.048690315425316	+0.312865496004875
+0.077113760890257	0.312865496004875	0.638444188569809	+0.048690315425316
+0.077113760890257	0.312865496004875	0.048690315425316	+0.638444188569809
+0.077113760890257	0.048690315425316	0.638444188569809	+0.312865496004875
+0.077113760890257	0.048690315425316	0.312865496004875	+0.638444188569809

Tabela 7.1: Integração Gaussiana num triângulo com ordem 7.

Observe que as funções g que temos que integrar são produtos de dois polinômios esféricos de grau 6 (isto é, polinômios de grau 12) multiplicado por um fator de correção não polinomial. Portanto a ordem $d = 7$ pode ser insuficiente para o cálculo de $\langle f, g \rangle$ com precisão necessária. Uma forma de contornar este problema, é dividir cada triângulo t em quatro triângulos, melhorando assim a aproximação obtida.

7.4 Complexidade do cálculo da matriz G

O cálculo da matriz G é a etapa mais trabalhosa do processo todo. Felizmente ela só precisa ser calculada uma vez. Além disso, sua complexidade é proporcional ao tamanho da base (e portanto ao número n_t de triângulos de T), e não quadrática como seria de se esperar. A razão é que os elementos da base têm suporte local, isto é, restrito a um número limitado de triângulos. Além disso, cada triângulo ocorre no suporte de um número limitado de elementos da base. Portanto, a quantidade de pares de elementos cujos suportes têm interseção nao vazia é linear no número de triângulos.

Ou seja a matriz G tem apenas $O(n_t)$ elementos não nulos. A figura 7.1 evidencia este fato.

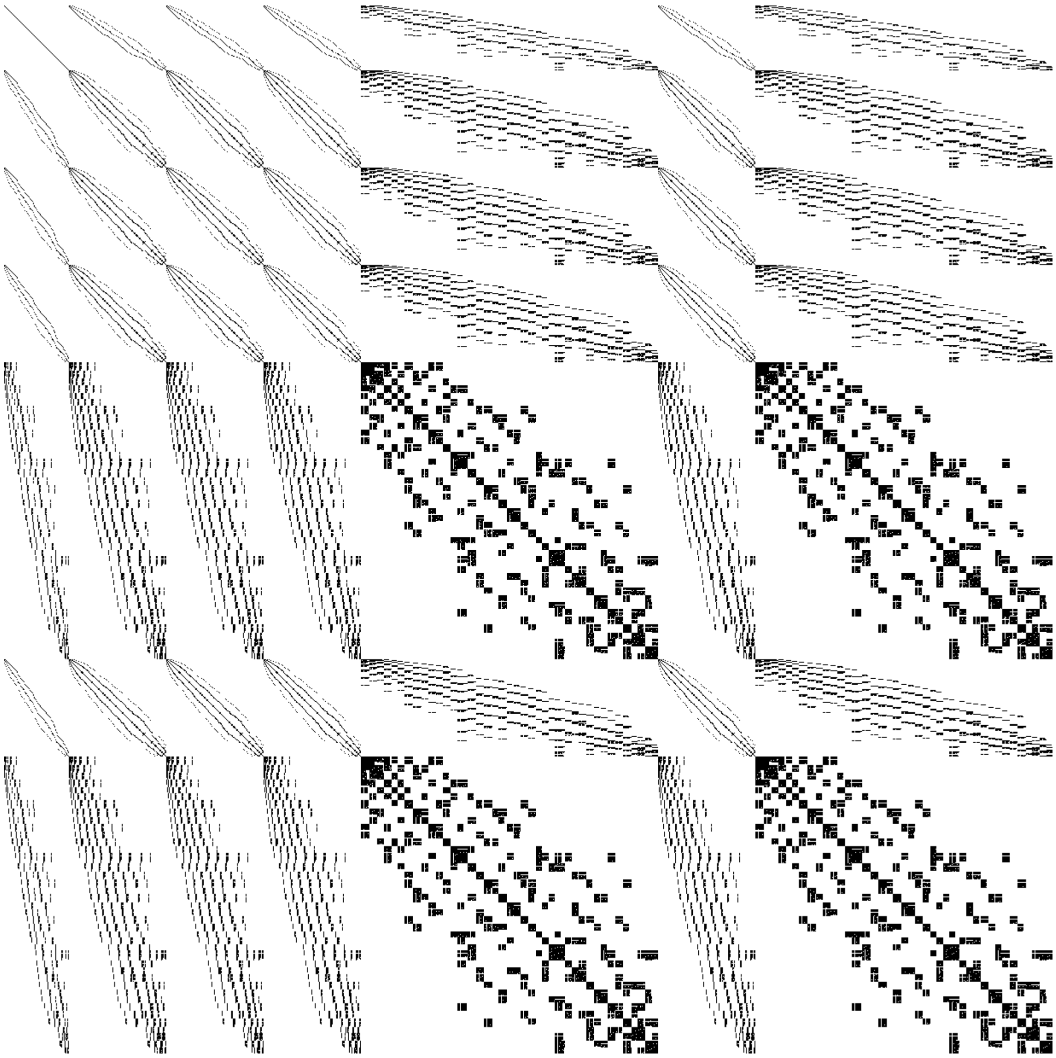


Figura 7.1: Retrato da matriz de produtos escalares G_{ij}, (1228×1228) para o espaço $\mathcal{P}_1^6[T]/\mathbf{S}^2$ onde T é uma triangulação irregular com 76 triângulos. Cada ponto da imagem corresponde a um par ϕ_i, ϕ_j, com $\langle \phi_i, \phi_j \rangle \neq 0$.

A figura 7.2 mostra o retrato da mesma G, obtida com a base semi-ortogonalizada conforme descrito na seção 6.6.

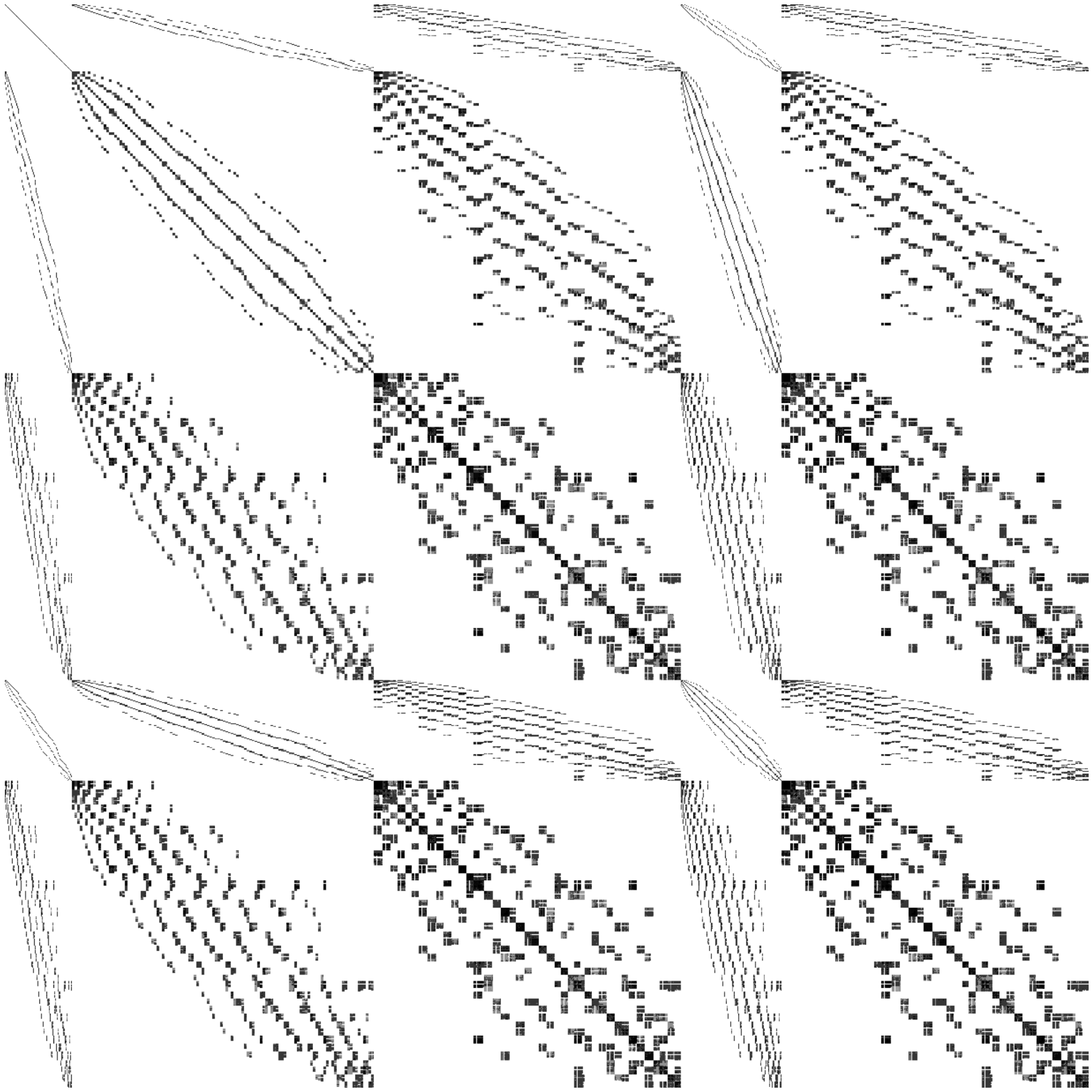


Figura 7.2: Retrato da matriz de produtos escalares G_{ij}, para o espaço $\mathcal{P}_1^6[T]/\mathbf{S}^2$ após a semi-ortogonalização de seus elementos.

7.5 Exemplos de aproximações com continuidade $\mathbf{C}_0$

Testamos, nesta seção, a capacidade de aproximação dos espaços $\mathcal{H}_k^d[T]/\mathbf{S}^2$ e $\mathcal{P}_k^d[T]/\mathbf{S}^2$, pelo método descrito acima.

Em particular aproximamos as funções

$$\begin{aligned} f(x,y,z) &= x^2 \\ g(x,y,z) &= x^3 \\ h(x,y,z) &= \exp(x) \\ m(x,y,z) &= \sin(x) \\ n(x,y,z) &= \cos(x) \end{aligned}$$

nos espaços $\mathcal{H}_0^2[T]/\mathbf{S}^2$, $\mathcal{H}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$. Vale observar que $f(p)$ pertence a $\mathcal{H}_0^2[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$; e $g(p)$ pertence a $\mathcal{H}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$; logo, nestes casos a aproximação deve ser exata.

7.5.1 Aproximação da função x^2

A figura 7.3 ilustra a função $f(p) = x^2$ com $p = (x, y, z)$.

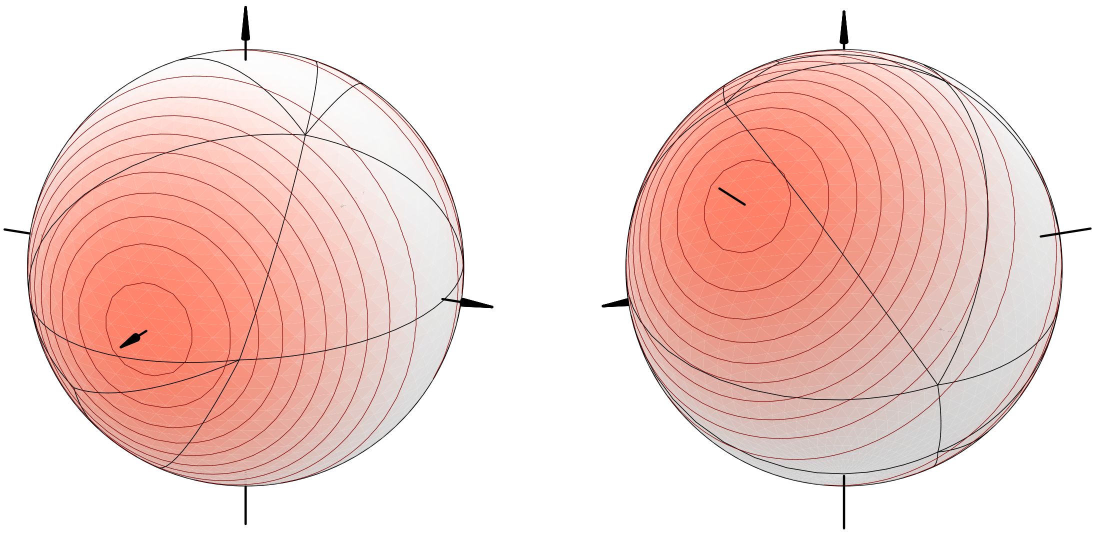


Figura 7.3: Função $u(p) = x^2$. O passo das curvas de nível é $\delta = 0.1$

As figuras 7.4 abaixo mostram o erro de aproximação $e(p) = |u(p) - f(p)|$ para funções aproximadoras $u(p)$, nos espaços $\mathcal{H}_0^2[T]/\mathbf{S}^2$, $\mathcal{H}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$, determinadas pelo método dos mínimos quadrados. Observe que a aproximação nos espaços $\mathcal{H}_0^2[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$ é exata (a menos de erros de arredondamento), enquanto que no espaço $\mathcal{H}_0^3[T]/\mathbf{S}^2$ o erro de aproximação é considerável, da ordem de 10^{-2}.

(a)
$\mathcal{H}_0^2[T]/\mathbf{S}^2$
$e_{\max} = 1.5 \times 10^{-15}$
$\delta = 2.5 \times 10^{-16}$

(b)
$\mathcal{H}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 3.9 \times 10^{-02}$
$\delta = 5.0 \times 10^{-03}$

(c)
$\mathcal{P}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 4.8 \times 10^{-13}$
$\delta = 5.0 \times 10^{-14}$

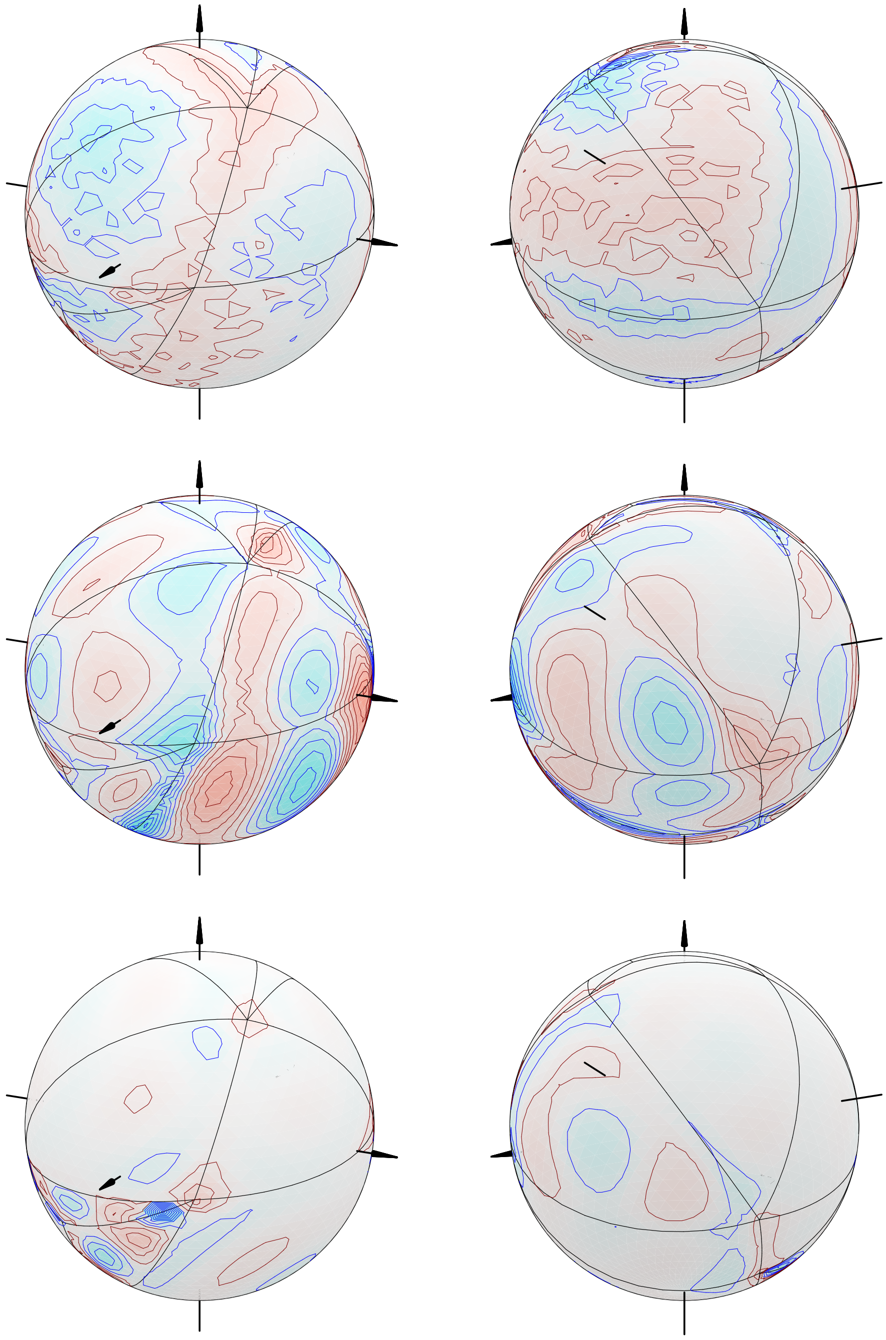


Figura 7.4: Erros de aproximação $e(p) = |u(p) - x^2|$. O parâmetro δ é o passo das curvas de nível

7.5.2 Aproximação da função x^3

Os testes a seguir utilizam a função $g(p) = x^3$, com $p = (x, y, z)$, ilustrada na figura 7.5.

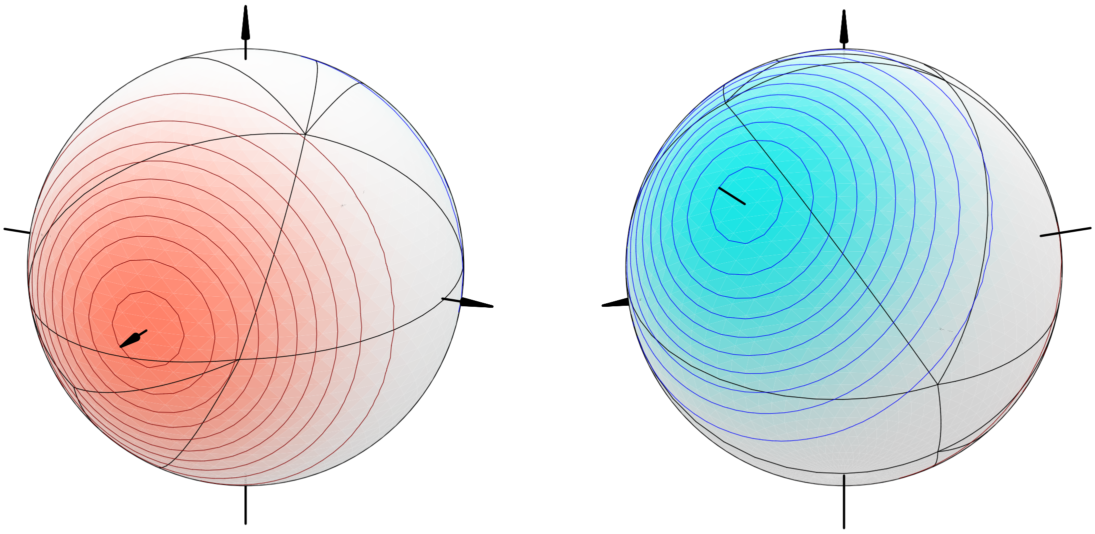


Figura 7.5: Função $u(p) = x^3$. O passo das curvas de nível é $\delta = 0.1$

As figuras 7.6 mostram o erro da aproximação $e(p) = |u(p) - g(p)|$ para funções aproximadoras $u(p)$ nos espaços $\mathcal{H}_0^2[T]/\mathbf{S}^2$, $\mathcal{H}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$. Aqui a aproximação nos espaços $\mathcal{H}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$ é exata (a menos de erros de arredondamento), enquanto que no espaço $\mathcal{H}_1^2[T]/\mathbf{S}^2$ o erro da aproximação é considerável, da ordem de 10^{-01}.

(a)
$\mathcal{H}_0^2[T]/\mathbf{S}^2$
$e_{\max} = 1.0 \times 10^{-01}$
$\delta = 2.5 \times 10^{-02}$

(b)
$\mathcal{H}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 2.0 \times 10^{-15}$
$\delta = 2.5 \times 10^{-16}$

(c)
$\mathcal{P}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 5.8 \times 10^{-13}$
$\delta = 1.0 \times 10^{-13}$

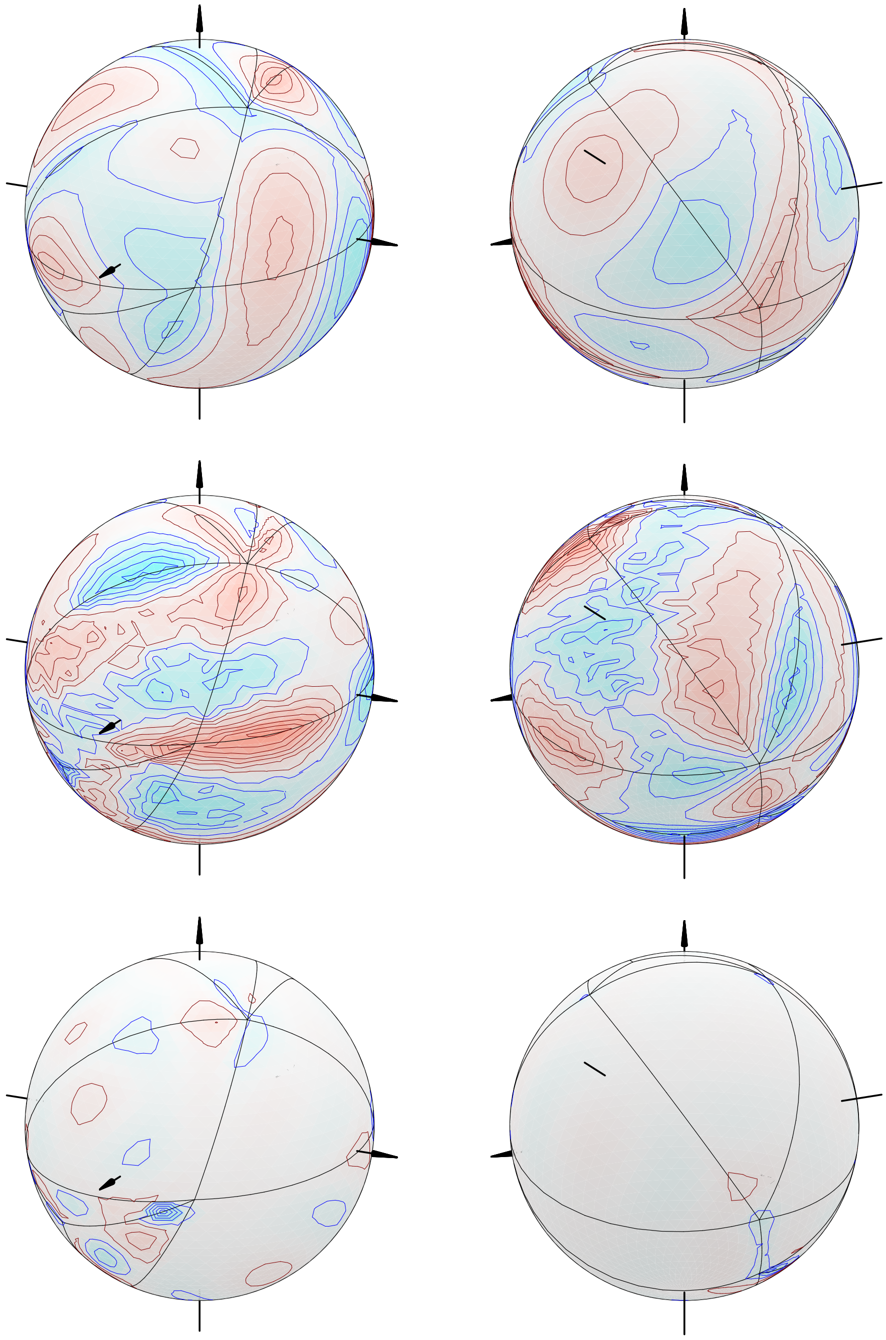


Figura 7.6: Erros de aproximação $e(p) = |u(p) - x^3|$. O parâmetro δ é o passo das curvas de nível

7.5.3 Aproximação da função $\exp(x)$

Os testes seguintes são baseados na função $h(p) = \exp(x)$, com $p = (x, y, z)$, mostrada na figura 7.7.

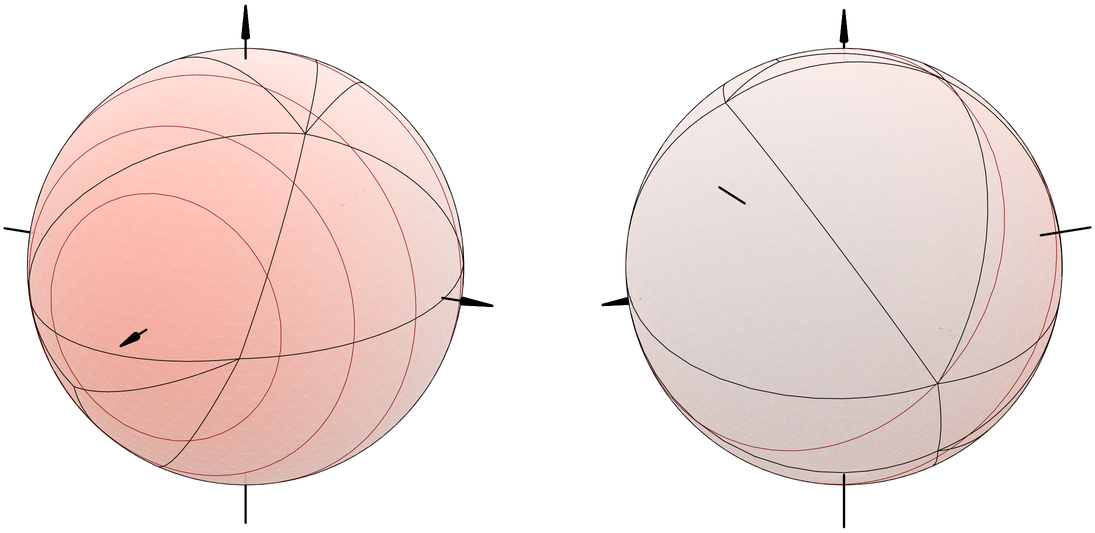
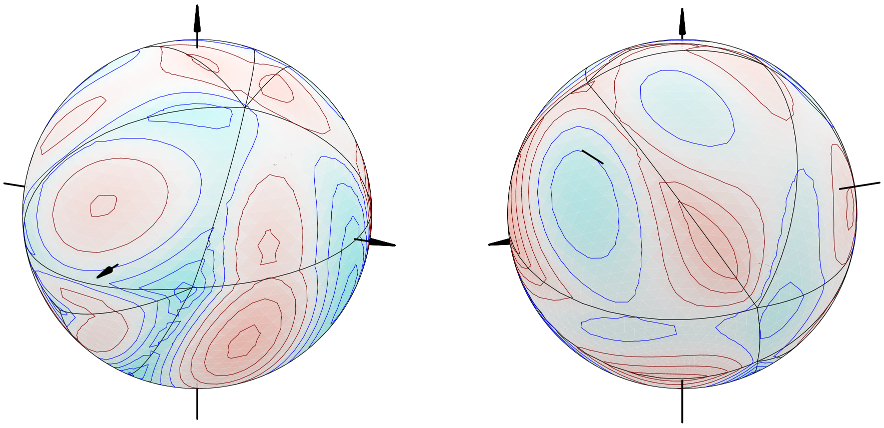


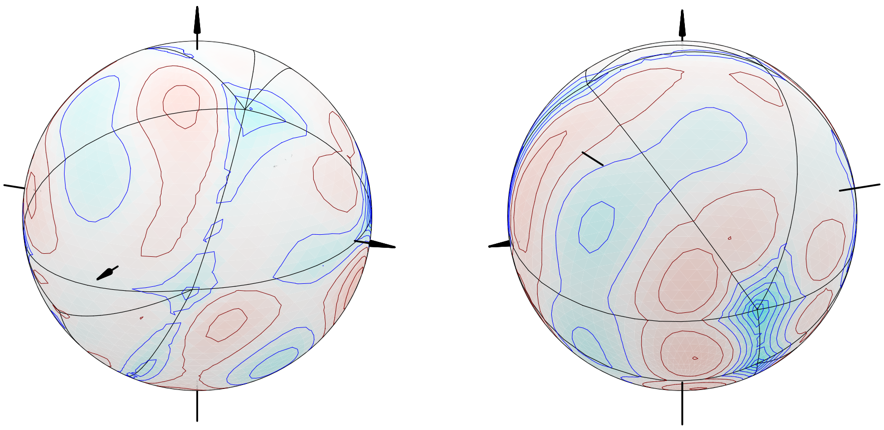
Figura 7.7: Função $u(p) = \exp(x)$. O passo das curvas de nível é $\delta = 0.5$

De maneira análoga, a figura 7.8 mostra os erro de aproximação $e(p) = |u(p) - \exp(x)|$ para funções aproximadoras $u(p)$ nos espaços $\mathcal{H}_0^2[T]/\mathbf{S}^2$, $\mathcal{H}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$. Observe que a função $\exp(x)$ não pertence aos espaços considerados; porém, no espaço $\mathcal{P}_0^3[T]/\mathbf{S}^2$ os erros são da ordem de 10^{-03}, enquanto que no espaço $\mathcal{H}_0^2[T]/\mathbf{S}^2$ e $\mathcal{H}_0^3[T]/\mathbf{S}^2$ os erros são da ordem de 10^{-01}.

(a)
$\mathcal{H}_0^2[T]/\mathbf{S}^2$
$e_{\max} = 1.2 \times 10^{-01}$
$\delta = 2.5 \times 10^{-02}$



(b)
$\mathcal{H}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 1.7 \times 10^{-01}$
$\delta = 2.5 \times 10^{-02}$



(c)
$\mathcal{P}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 1.2 \times 10^{-03}$
$\delta = 2.5 \times 10^{-04}$

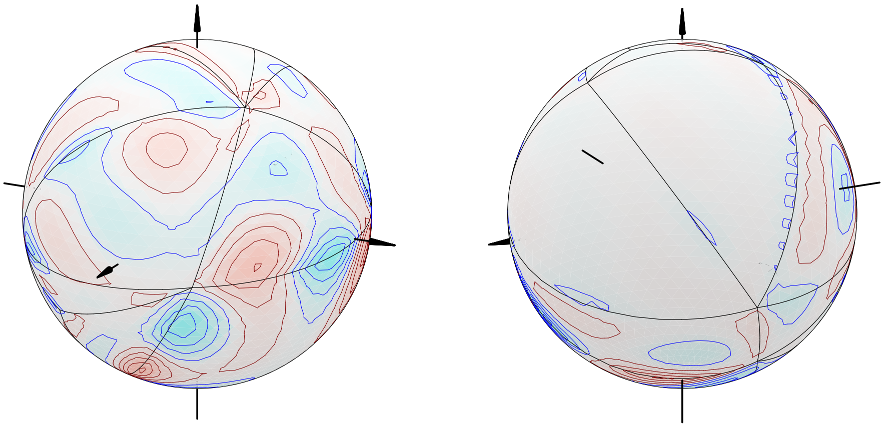


Figura 7.8: Erros de aproximação $e(p) = |u(p) - \exp(x)|$. O parâmetro δ é o passo das curvas de nível

7.5.4 Aproximação da função $\sin(x)$

A figura 7.9 ilustra a aproximação da função $m(p) = \sin(x)$ com $p = (x, y, z)$.

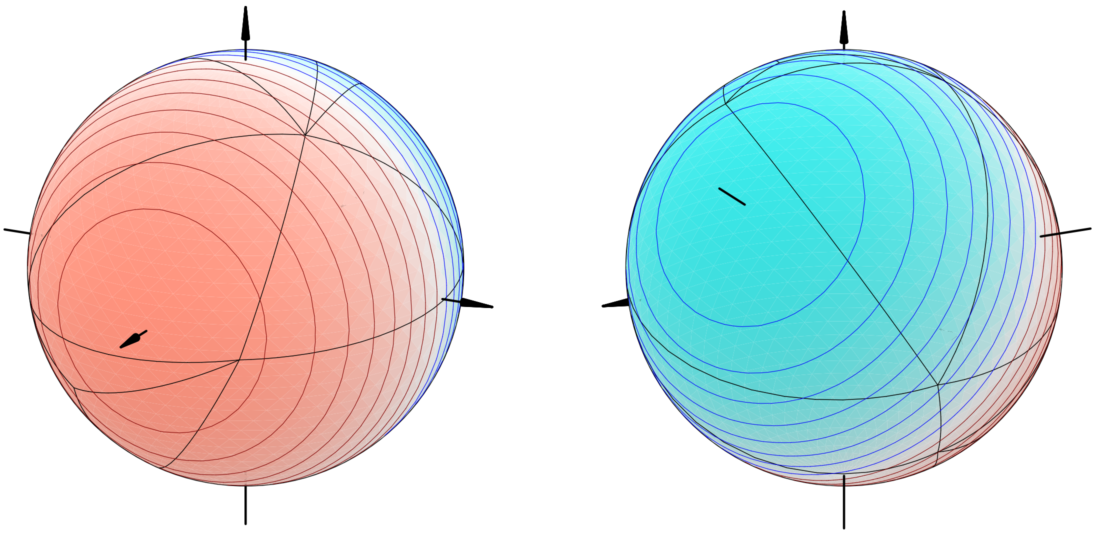


Figura 7.9: Função $u(p) = \sin(x)$. O passo das curvas de nível é $\delta = 0.1$

A figura 7.10(a) mostra os erros de aproximação $e(p) = |u(p) - \sin(x)|$ para funções aproximadoras $u(p)$ nos espaços $\mathcal{H}_0^2[T]/\mathbf{S}^2$, $\mathcal{H}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$. Embora a função $\sin(x)$ não pertença aos espaços considerados, ela é dominada pelo termo de grau 1 da série de Taylor (x), que pertence a $\mathcal{H}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$. Isto explica porque a aproximação nestes espaços foi bem melhor do que no espaço $\mathcal{H}_0^2[T]/\mathbf{S}^2$, onde erros são da ordem de 10^{-01}.

(a)
$\mathcal{H}_0^2[T]/\mathbf{S}^2$
$e_{\max} = 1.2 \times 10^{-01}$
$\delta = 2.5 \times 10^{-02}$

(b)
$\mathcal{H}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 3.7 \times 10^{-04}$
$\delta = 5.0 \times 10^{-05}$

(c)
$\mathcal{P}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 2.8 \times 10^{-04}$
$\delta = 5.0 \times 10^{-05}$

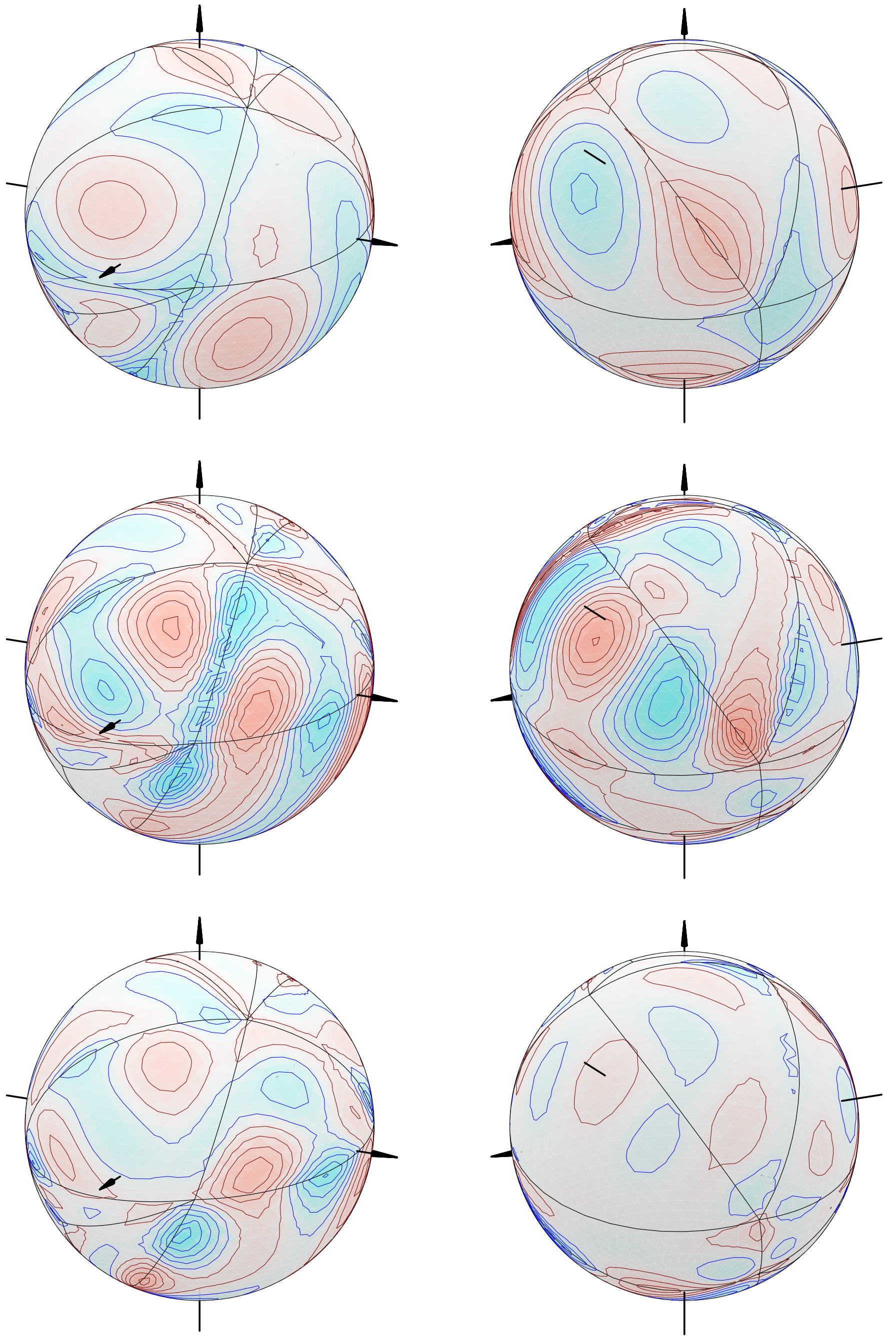


Figura 7.10: Erros de aproximação $e(p) = |u(p) - \sin(x)|$. O parâmetro δ é o passo das curvas de nível

7.5.5 Aproximação da função $\cos(x)$

A figura 7.11 ilustra a função $n(p) = \cos(x)$ com $p = (x, y, z)$.

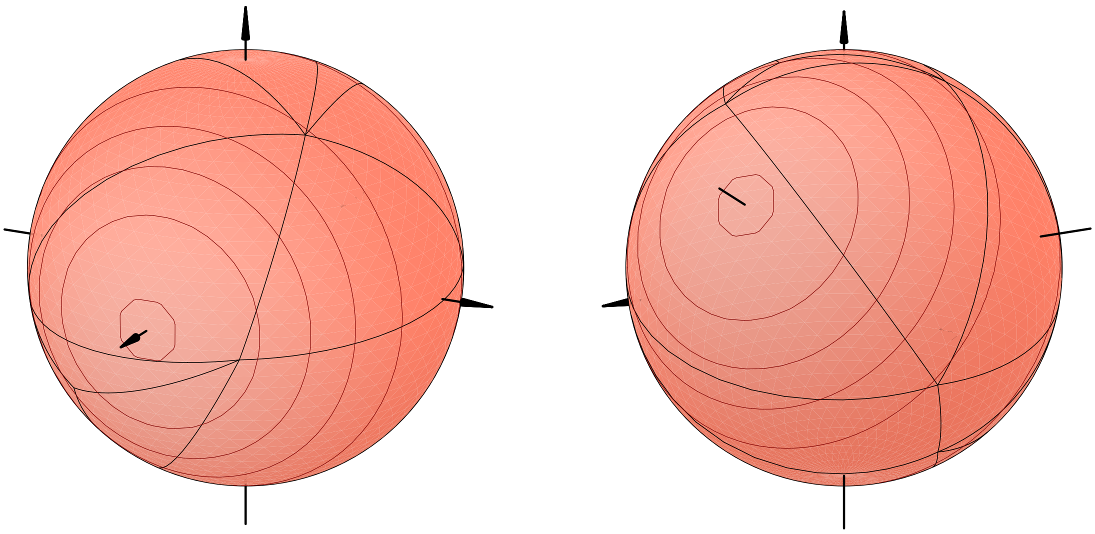


Figura 7.11: Função $u(p) = \cos(x)$. O passo das curvas de nível é $\delta = 0.1$

A figura 7.12 mostra os erros de aproximação $e(p) = |u(p) - \cos(x)|$ para funções aproximadoras $u(p)$ nos espaços $\mathcal{H}_0^2[T]/\mathbf{S}^2$, $\mathcal{H}_0^3[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$. Embora a função $\cos(x)$ não pertença aos espaços considerados, ela é dominada pelo termo de grau 0 da série de Taylor (a constante 1), que pertence a $\mathcal{H}_0^2[T]/\mathbf{S}^2$ e $\mathcal{P}_0^3[T]/\mathbf{S}^2$. Isto explica porque a aproximação nestes espaços foi bem melhor do que no espaço $\mathcal{H}_0^3[T]/\mathbf{S}^2$, onde erros são da ordem de 10^{-01}.

(a)
$\mathcal{H}_0^2[T]/\mathbf{S}^2$
$e_{\max} = 6.6 \times 10^{-03}$
$\delta = 1.0 \times 10^{-03}$

(b)
$\mathcal{H}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 1.8 \times 10^{-01}$
$\delta = 2.5 \times 10^{-02}$

(c)
$\mathcal{P}_0^3[T]/\mathbf{S}^2$
$e_{\max} = 9.0 \times 10^{-04}$
$\delta = 1.0 \times 10^{-04}$

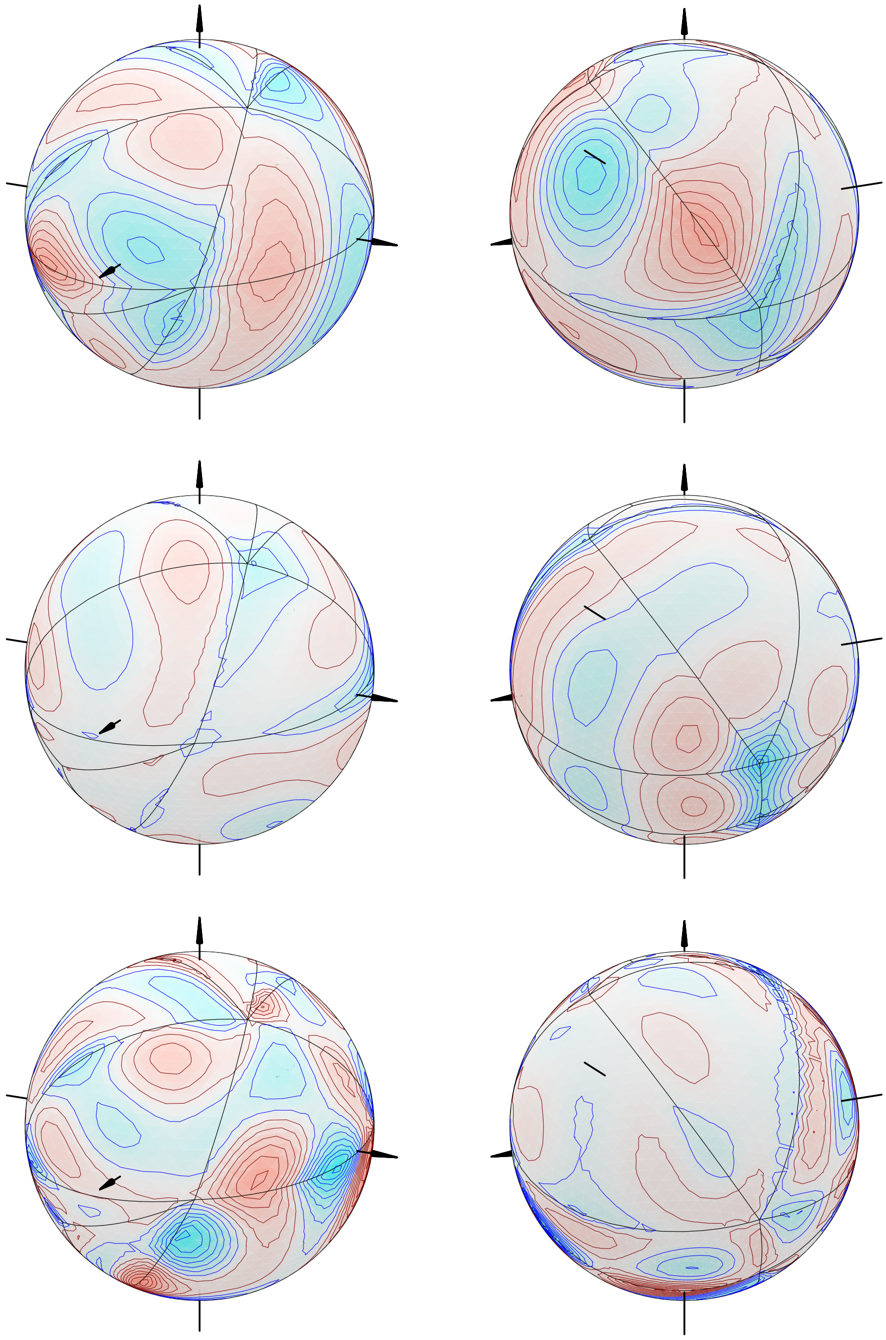


Figura 7.12: Erros de aproximação $e(p) = |u(p) - \cos(x)|$. O parâmetro δ é o passo das curvas de nível

7.5.6 Conclusão

A tabela abaixo resume os erros de aproximação obtidos nestes testes.

	Espaços		
Funções	$\mathcal{H}_0^2[T]/\mathbf{S}^2$	$\mathcal{H}_0^3[T]/\mathbf{S}^2$	$\mathcal{P}_0^3[T]/\mathbf{S}^2$
x^2	1.5×10^{-15}	3.9×10^{-02}	4.8×10^{-13}
x^3	1.0×10^{-01}	2.0×10^{-15}	5.8×10^{-13}
$\exp(x)$	1.2×10^{-01}	1.7×10^{-01}	1.2×10^{-03}
$\sin(x)$	1.2×10^{-01}	3.7×10^{-04}	2.8×10^{-04}
$\cos(x)$	6.6×10^{-03}	1.8×10^{-01}	9.0×10^{-04}

Observa-se que no espaço $\mathcal{P}_0^3[T]/\mathbf{S}^2$ obtemos os erros de aproximação da ordem de 10^{-13} para funções que pertencem a esse espaço, enquanto que nos outros espaços o erro na mesma situação é da ordem de 10^{-15}. A razão disso é que no espaço $\mathcal{P}_0^3[T]/\mathbf{S}^2$ temos mais elementos na base, e consequentemente a matriz do sistema é maior, aumentando assim os erros de arredondamento. Para funções que não pertencem aos espaços, notamos uma considerável melhora do resultado obtido no espaço $\mathcal{P}_0^3[T]/\mathbf{S}^2$.

7.6 Exemplos de aproximações com continuidade $\mathbf{C}_1$

Para testar a capacidade de aproximação dos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$, $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$, calculamos as aproximações das funções

$$\begin{aligned} f(x,y,z) &= x^2 \\ g(x,y,z) &= x^3 \\ h(x,y,z) &= \exp(x) \end{aligned}$$

com $p = (x,y,z)$, nestes espaços. É importante observar que $f(p)$ pertence a $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$, $g(p)$ pertence a $\mathcal{H}_1^5[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$, logo a aproximação dessas funções nesses espaços deve ser exata.

7.6.1 Aproximação da função x^2

A figura 7.13 ilustra a função $f(p) = x^2$ com $p = (x, y, z)$.

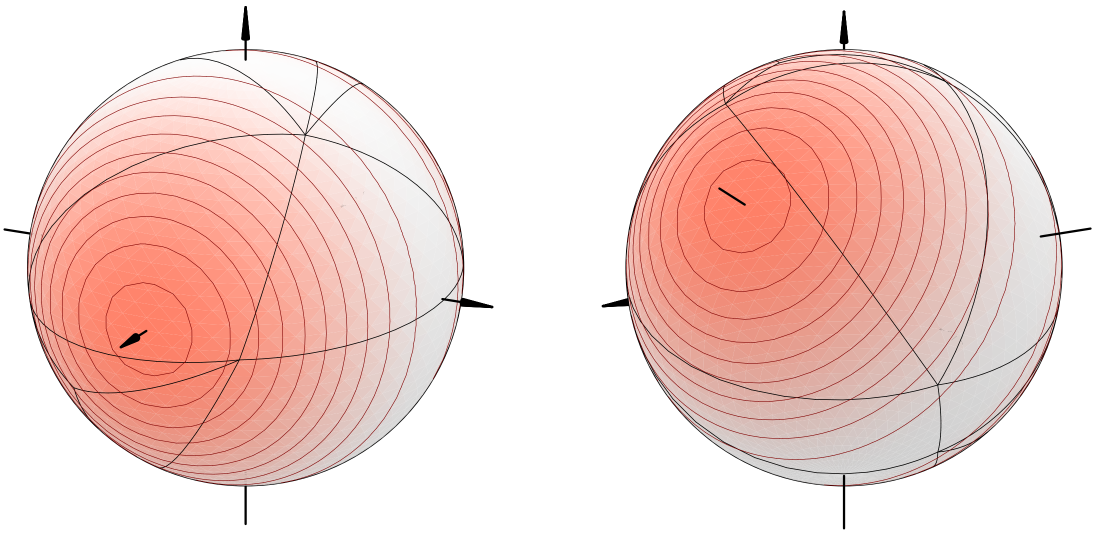


Figura 7.13: Função $u(p) = x^2$. O passo das curvas de nível é $\delta = 0.1$

As figuras 7.14 mostram o erro de aproximação $e(p) = |u(p) - f(p)|$ para funções aproximadoras $u(p)$, nos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$, $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$, determinadas pelo método dos mínimos quadrados. Note que a aproximação nos espaços $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$ é exata (a menos de erros de arredondamento), enquanto que no espaço $\mathcal{H}_1^5[T]/\mathbf{S}^2$ o erro de aproximação é considerável, da ordem de 10^{-02}.

(a)
$\mathcal{H}_1^5[T]/\mathbf{S}^2$
$e_{\max} = 1.1 \times 10^{-02}$
$\delta = 2.5 \times 10^{-03}$

(b)
$\mathcal{H}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 4.4 \times 10^{-12}$
$\delta = 5.0 \times 10^{-13}$

(c)
$\mathcal{P}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 1.7 \times 10^{-08}$
$\delta = 2.5 \times 10^{-09}$

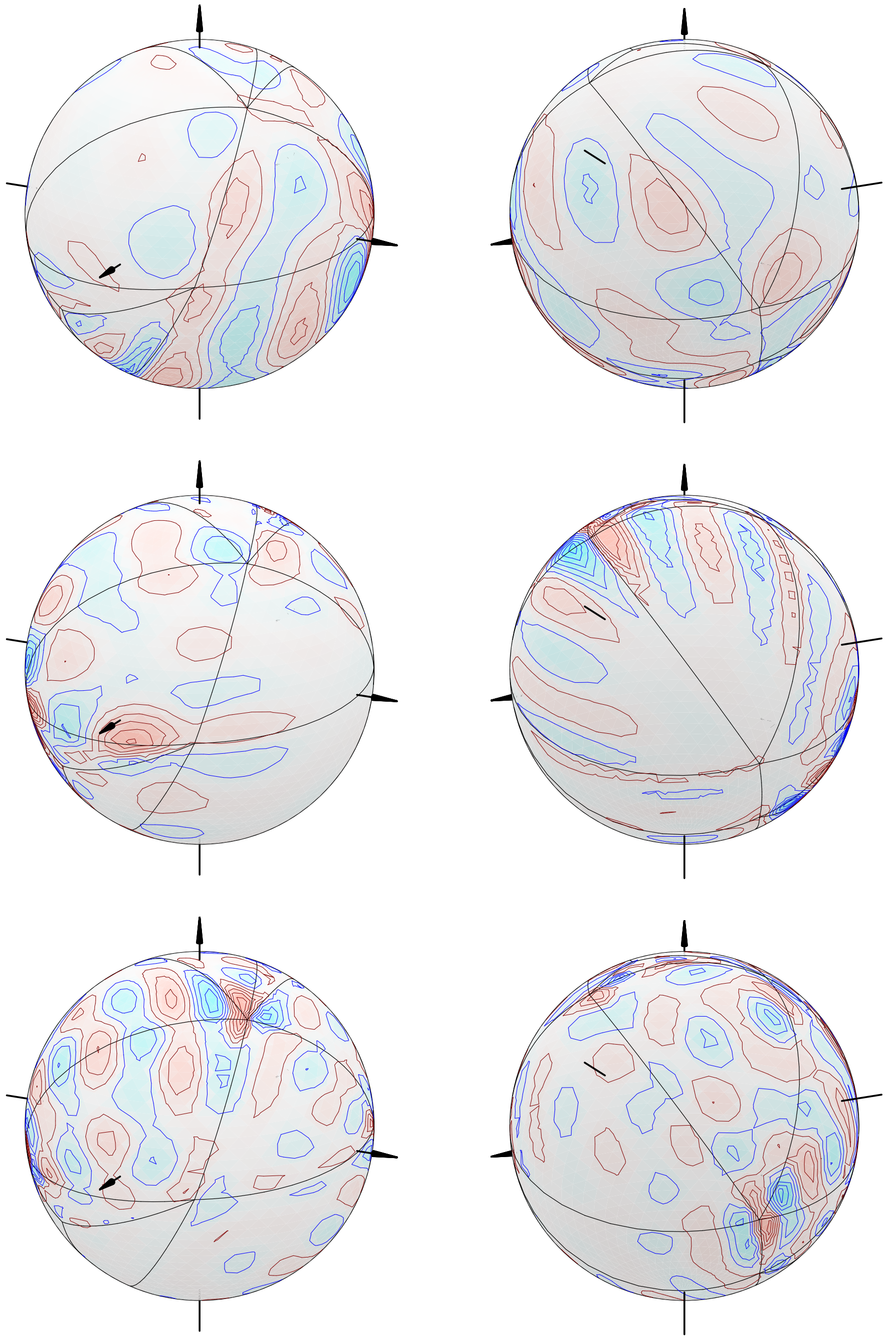


Figura 7.14: Erros de aproximação $e(p) = |u(p) - x^2|$. O parâmetro δ é o passo das curvas de nível

7.6.2 Aproximação da função x^3

Os próximos testes utilizam a função $g(p) = x^3$,com $p = (x, y, z)$, ilustrada na figura 7.15.

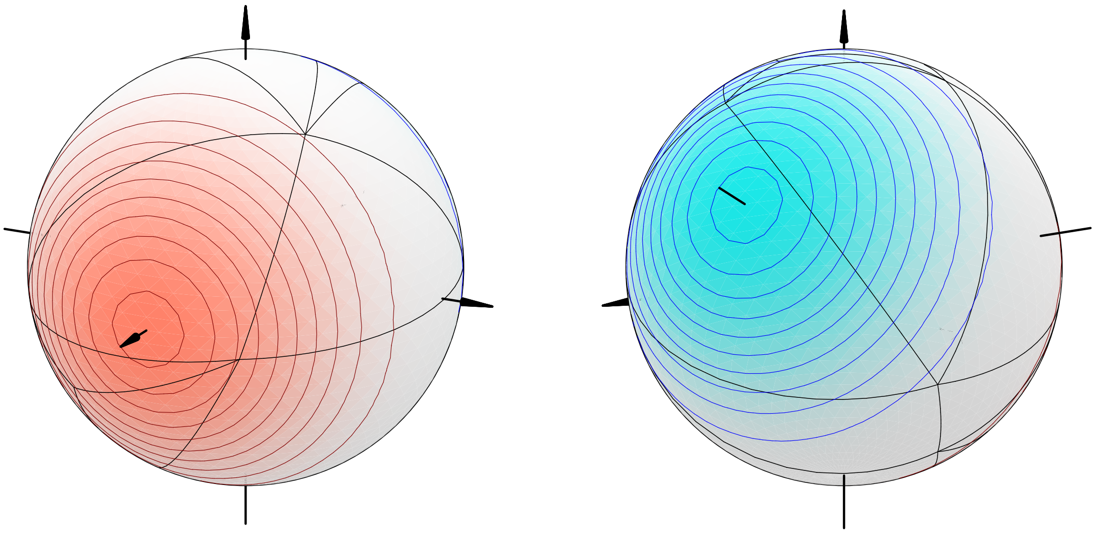


Figura 7.15: Função $u(p) = x^3$. O passo das curvas de nível é $\delta = 0.1$

As figuras 7.16 mostram o erro da aproximação $e(p) = |u(p) - g(p)|$ para funções aproximadoras $u(p)$ nos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$, $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$. Aqui a aproximação nos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$ é exata (a menos de erros de arredondamento), enquanto que no espaço $\mathcal{H}_1^6[T]/\mathbf{S}^2$ o erro da aproximação é considerável, da ordem de 10^{-03}.

(a)
$\mathcal{H}_1^5[T]/\mathbf{S}^2$
$e_{\max} = 1.7 \times 10^{-12}$
$\delta = 2.5 \times 10^{-13}$

(b)
$\mathcal{H}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 3.7 \times 10^{-03}$
$\delta = 5.0 \times 10^{-04}$

(c)
$\mathcal{P}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 1.0 \times 10^{-08}$
$\delta = 2.5 \times 10^{-09}$

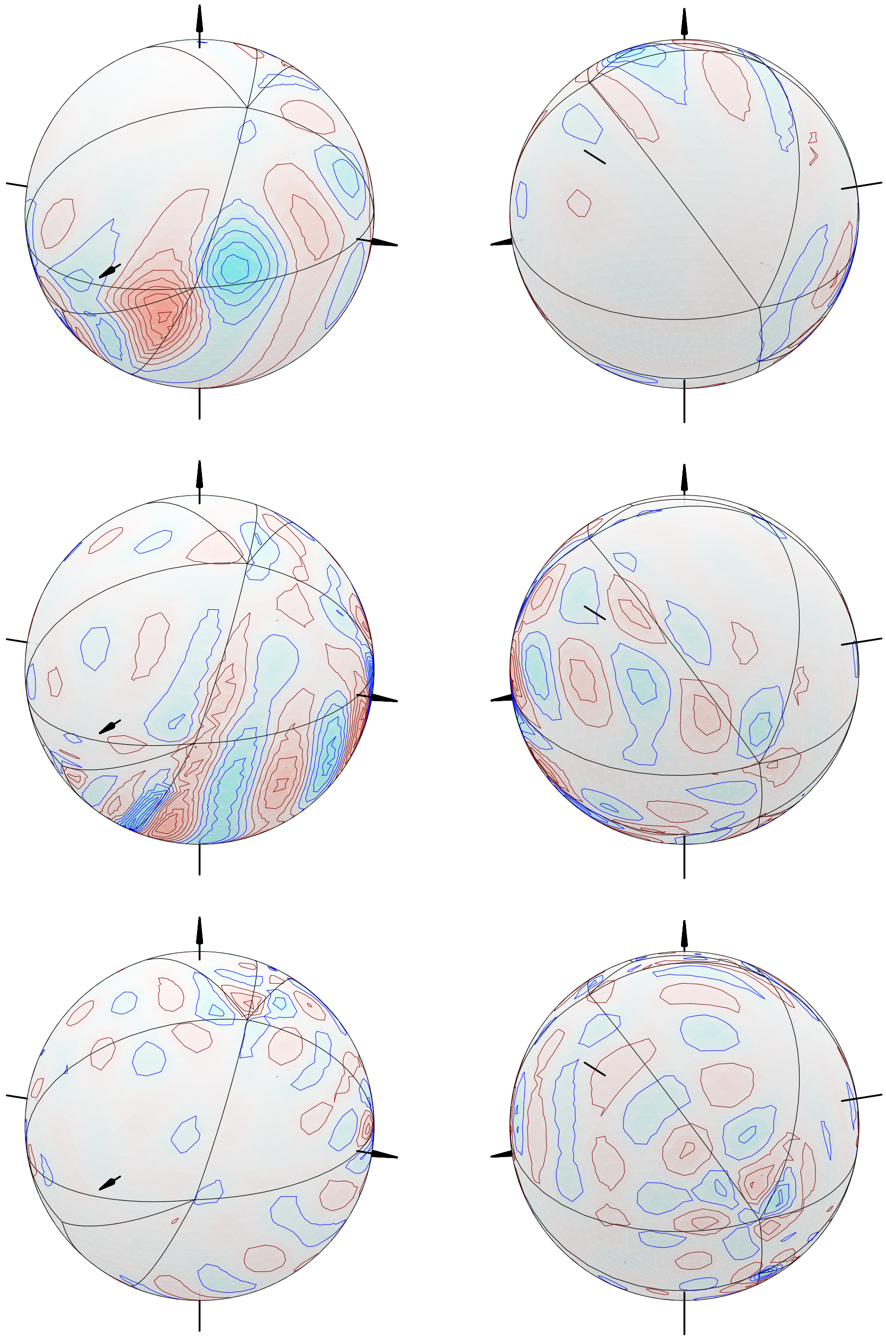


Figura 7.16: Erros de aproximação $e(p) = |u(p) - x^3|$. O parâmetro δ é o passo das curvas de nível

7.6.3 Aproximação da função $\exp(x)$

Os testes seguintes são baseados na função $h(p) = \exp(x)$, com $p = (x, y, z)$, mostrada na figura 7.17.

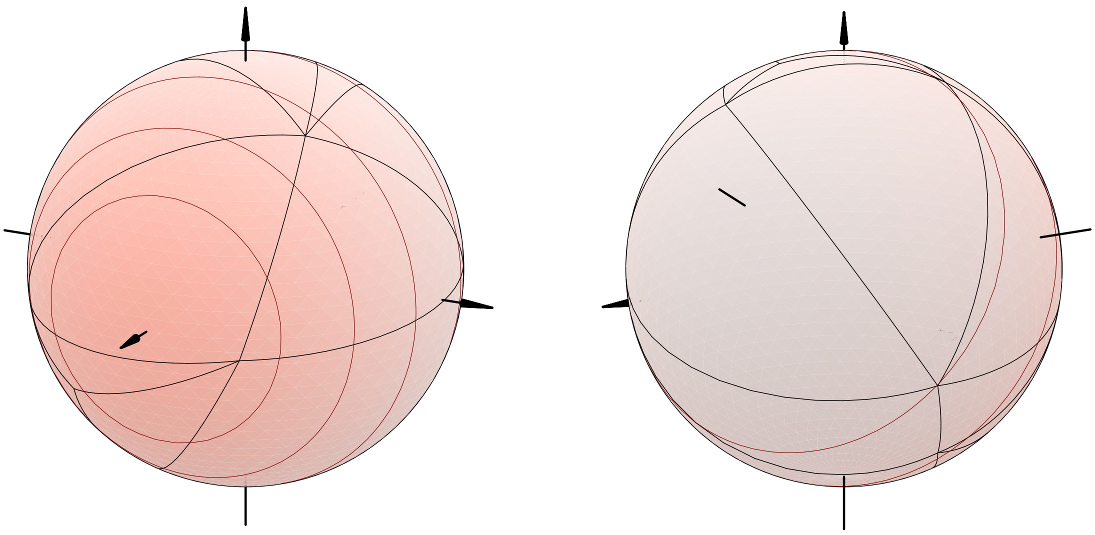


Figura 7.17: Função $u(p) = \exp(x)$. O passo das curvas de nível é $\delta = 0.5$

Como anteriormente, a figura 7.18 mostra os erro de aproximação $e(p) = |u(p) - \exp(x)|$ para funções aproximadoras $u(p)$ nos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$, $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$. Esta função não pertence a nenhum dos espaços considerados; porém, no espaço $\mathcal{P}_1^6[T]/\mathbf{S}^2$ os erros de aproximação são comparáveis aos erros de arredondamento 10^{-07}, enquanto que nos espaços $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{H}_1^5[T]/\mathbf{S}^2$ os erros são da ordem de 10^{-01} e 10^{-02} respectivamente.

(a)
$\mathcal{H}_1^5[T]/\mathbf{S}^2$
$e_{\max} = 6.1 \times 10^{-02}$
$\delta = 1.0 \times 10^{-02}$

(b)
$\mathcal{H}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 9.6 \times 10^{-03}$
$\delta = 1.0 \times 10^{-03}$

(c)
$\mathcal{P}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 1.6 \times 10^{-07}$
$\delta = 2.5 \times 10^{-08}$

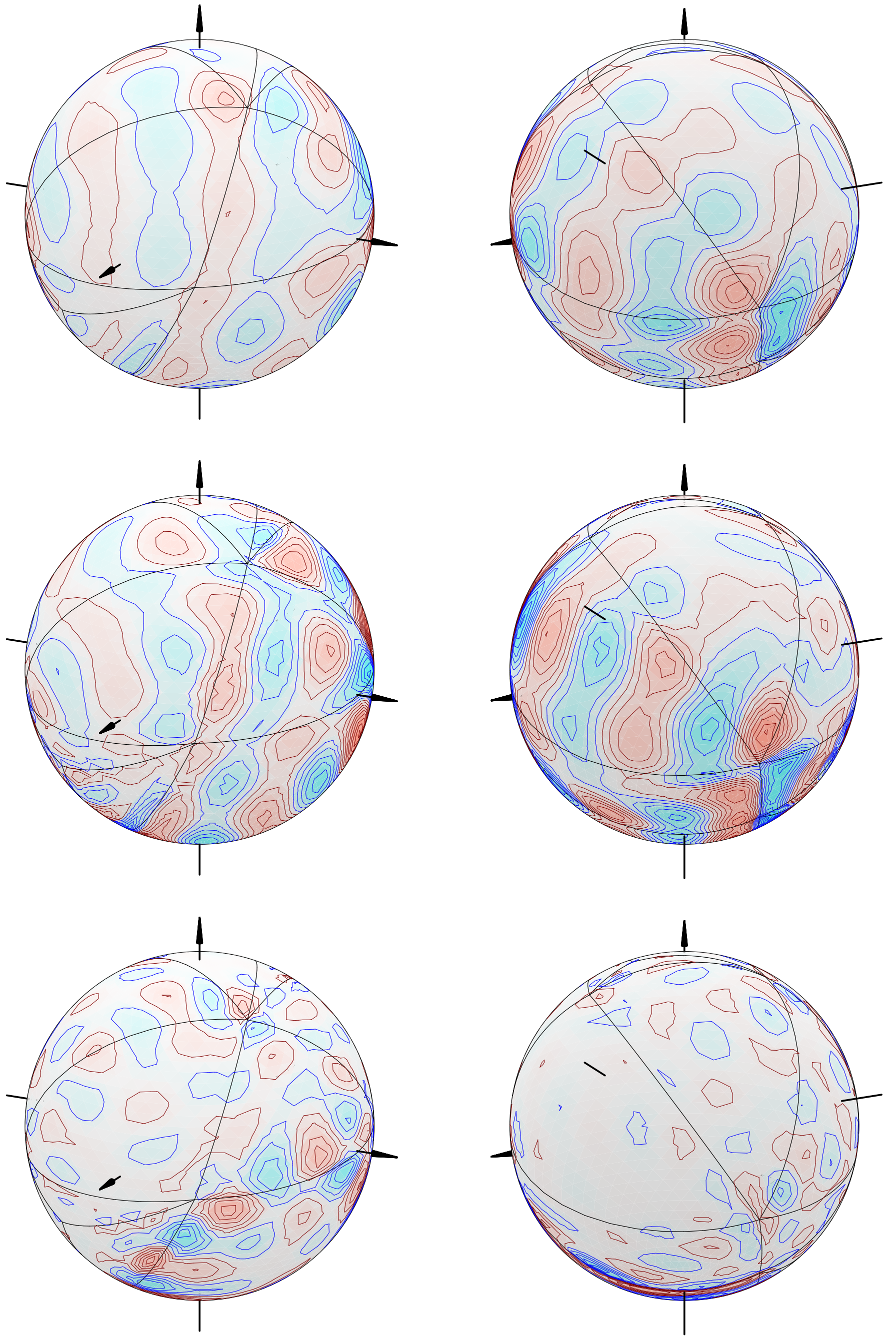


Figura 7.18: Erros de aproximação $e(p) = |u(p) - \exp(x)|$. O parâmetro δ é o passo das curvas de nível

7.6.4 Conclusão

A tabela abaixo resume os erros de aproximação obtidos nestes testes.

	Espaços		
Funções	$\mathcal{H}_1^5[T]/\mathbf{S}^2$	$\mathcal{H}_1^6[T]/\mathbf{S}^2$	$\mathcal{P}_1^6[T]/\mathbf{S}^2$
x^2	1.1×10^{-02}	4.4×10^{-12}	1.7×10^{-08}
x^3	1.7×10^{-12}	3.7×10^{-03}	1.0×10^{-08}
$\exp(x)$	6.1×10^{-02}	9.6×10^{-03}	1.6×10^{-07}

Note-se que no espaço $\mathcal{P}_1^6[T]/\mathbf{S}^2$ obtemos os erros de aproximação da ordem de 10^{-08} para funções que pertencem a esse espaço, enquanto que nos outros espaços o erro na mesma situação é da ordem de 10^{-12}. A razão disso é que no espaço $\mathcal{P}_1^6[T]/\mathbf{S}^2$ temos muito mais elementos na base, e consequentemente a matriz do sistema é bem maior, aumentando assim os erros de arredondamento. Para funções que não pertencem aos espaços, notamos que o resultado obtido no espaço $\mathcal{P}_1^6[T]/\mathbf{S}^2$ é bem mais preciso que nos espaços homogêneos.

Capítulo 8

Solução de Equações Diferenciais Parciais por Elementos Finitos

8.1 Equações diferenciais parciais na esfera

Seja a seguinte equação diferencial sobre um certo domínio $\Omega \subseteq \mathbf{R}^n$

$$\mathcal{L}(u)(p) = \mathcal{F}(u(p), p) \tag{8.1}$$

onde $\mathcal{L}$ é um operador diferencial, $p \in \Omega$ é a variável independente, u é a função a determinar e $\mathcal{F}$ é uma função que pode depender de $u(p)$ e p. Queremos encontrar uma solução, isto é uma função $u(p)$ que satisfaça esta equação diferencial.

8.1.1 Solução de E.D.P. por elementos finitos

A idéia básica de quase todo método numérico para integração de equações diferenciais é a discretização do problema original, com infinitos graus de liberdade, para um problema de aproximação dentro de um espaço de funções de dimensão finita. Desta forma, o problema de integração é reduzido a um sistema de equações com um número finito de incógnitas. Portanto para resolver a equação (8.1), vamos procurar uma aproximação da solução u num espaço de funções V_n de dimensão finita n, definidas sobre Ω com valores reais. Ou seja, dada uma base $\{\phi_i\}_{i=1}^n$ de V_n, procuramos uma aproximação $u^* = \sum_{i=1}^n a_i \phi_i$ tal que $u^*(p) \simeq u(p)$.

Para definir u^* completamente precisamos definir o que é uma aproximação ótima. O mais natural seria minimizar o erro da aproximação $\int_\Omega \parallel u(p) - u^*(p) \parallel^2 \, dp$. Entretanto este critério não pode ser usado pois u não é conhecida. Por esse motivo, a métrica usada em geral é a "discrepância" observada na equação diferencial (8.1) quando substituímos u por u^*. Mais precisamente, definimos o erro como sendo:

$$e^*(p) = \mathcal{L}(\sum_{i=1}^{n} a_i \phi_i)(p) - \mathcal{F}(\sum_{i=1}^{n} a_i \phi_i(p), p)$$

Além disso, em vez de minimizar a integral de $\| \ e^* \ \|^2$, exigimos que o erro e^* seja ortogonal a cada função da base, isto é:

$$\int_\Omega e^*(p) \cdot \phi_i(p) dp = 0 \qquad\qquad i = 1 \ldots n$$

Esta abordagem é conhecida como *método de Galerkin* [21]. Desta forma o problema reduz-se ao sistema de equações:

$$\int_\Omega \left[\mathcal{L}(\sum_{j=1}^{n} a_j \phi_j)(p) - \mathcal{F}(\sum_{j=1}^{n} a_j \phi_j(p), p) \right] \cdot \phi_i(p) dp = 0 \qquad i = 1 \ldots n \tag{8.2}$$

Neste sistema, as incógnitas são os coeficientes a_i. Se a função $\mathcal{F}$ não depende da solução u, e $\mathcal{L}$ é um operador diferencial linear, as equações também são lineares, e podem ser simplificadas para:

$$\sum_{j=1}^{n} a_j \int_\Omega \left[\mathcal{L}(\phi_j)(p) \cdot \phi_i(p) - \mathcal{F}(p) \cdot \phi_i(p) \right] dp = 0 \qquad i = 1 \ldots n \tag{8.3}$$

ou, em forma matricial,

$$Ha = b \tag{8.4}$$

onde

$$H_{ij} = \int_\Omega \mathcal{L}(\phi_j)(p) \cdot \phi_i(p) dp \qquad b_i = \int_\Omega \mathcal{F}(p) \cdot \phi_i(p) dp$$

Caso contrário, o sistema (8.2) é geralmente não linear.

O método dos elementos finitos consiste em procurar aproximações de Galerkin quando as funções ϕ_i são funções polinomiais por partes, como descritas no capítulo 3. As funções ϕ_i devem ter suporte menor possível para que a matriz H do sistema de equações (8.4) resultante da discretização seja bastante esparsa.

8.1.2 Equação de Helmholtz

Em particular estamos interessados em problemas onde o domínio Ω da equação diferencial é a esfera $\mathbf{S}^2$. Um exemplo é a *equação de Helmholtz*

$$- \Delta u(p) + c \cdot u(p) = \mathcal{F}(u(p), p) \tag{8.5}$$

onde $p = (x, y, z)$, com $x^2 + y^2 + z^2 = 1$. Esta equação tem inúmeras aplicações em meteorologia, geociências, etc [23].

Expandindo a equação (8.2) para $\mathcal{L}(u)(p) = -\Delta u + c \cdot u$ temos:

$$-\int_{\mathbf{S}^2} \left[\phi_i(p) \cdot \Delta(\sum_{j=1}^{n} a_j \phi_j)(p) \;+\; c \cdot \phi_i(p) \cdot (\sum_{j=1}^{n} a_j \phi_j)(p) \right] dp = b_i(a) \tag{8.6}$$

onde $b_i(a) = \int_{\mathbf{S}^2} \phi_i(p) \cdot \mathcal{F}(\sum_{j=1}^{n} a_j \phi_j(p), p) dp$ para $i = 1 \ldots n$, ou seja

$$-\sum_{j=1}^{n} a_j \left[\int_{\mathbf{S}^2} \phi_i(p) \cdot \Delta \phi_j(p) dp + c \cdot \int_{\mathbf{S}^2} \phi_i(p) \cdot \phi_j(p) dp \right] = b_i(a) \tag{8.7}$$

para $i = 1 \ldots n$.

A aplicação direta dessa equação exigiria que os splines da base ϕ_i fossem pelo menos de $\mathbf{C}_2$. Entretanto, podemos usar a identidade de Green [6]

$$\int_{\mathbf{S}^2} \phi_i(p) \cdot \Delta \phi_j(p) dp = -\int_{\mathbf{S}^2} \nabla \phi_i(p) \circ \nabla \phi_j(p) dp$$

para reduzir a ordem de diferenciação no integrando, o que permite o uso de espaços de splines $\mathbf{C}_1$. Assim o sistema de equações (8.7) fica

$$\sum_{j=1}^{n} a_j \left[\int_{\mathbf{S}^2} \nabla \phi_i(p) \circ \nabla \phi_j(p) dp + c \cdot \int_{\mathbf{S}^2} \phi_i(p) \cdot \phi_j(p) dp \right] = b_i(a) \tag{8.8}$$

para $i = 1 \ldots n$. Temos portanto um sistema não linear de equações $Ha = b(a)$, onde $H = S + c \cdot G$ com

$$S_{ij} = \int_{\mathbf{S}^2} \nabla \phi_i(p) \circ \nabla \phi_j(p) dp \quad i, j = 1 \ldots n$$

$$G_{ij} = \int_{\mathbf{S}^2} \phi_i(p) \cdot \phi_j(p) dp \quad i, j = 1 \ldots n$$

Note que a não linearidade no sistema (8.8) está limitada ao termo independente $b(a)$.

Apresentamos a seguir um algoritmo iterativo, `UniSolve`, para resolver este sistema. O algoritmo recebe como entrada a base $\phi = \{\phi_i : i = 1 \ldots n\}$, a função $\mathcal{F}$, uma tolerância τ (que determina a condição de término do algoritmo), e um número máximo de iterações $k_{\max}$. O algoritmo devolve um vetor a de coeficientes da aproximação u^* na base ϕ.

Algoritmo 6 *Procedimento* `UniSolve`$(\phi, \mathcal{F}, \tau, k_{\max})$

1. seja $a^{(0)}$ um vetor inicial
2. repita para $k = 0, 1, 2 \ldots k_{\max}$
    2.1. calcule a função $u^{(k)} = \sum_{i=1}^{n} a_i^{(k)} \phi_i$
    2.1. calcule o vetor $b^{(k)}$ onde $b_i^{(k)} = \int_{\mathbf{S}^2} \phi_i . \mathcal{F}(u(p), p) dp$
    2.3. resolva o sistema $Ha^{(k+1)} = b^{(k)}$ obtendo novos coeficientes $a^{(k+1)}$.
    2.4. se $\|a^{(k)} - a^{(k+1)}\| \leq \tau$, retorna $a^{(k+1)}$

No algoritmo acima, se a função $\mathcal{F}$ depender da solução u não precisamos, no passo 2.3, achar a solução exata do sistema. Basta em geral calcular uma solução aproximada, executando uma única iteração de um método iterativo (por exemplo Gauss-Seidel), pois o vetor $b(a)$ terá que ser recalculado de qualquer forma a cada iteração. Por outro lado se a função $\mathcal{F}$ não depende de u, o algoritmo acima precisa executar apenas uma iteração ($k_{max} = 1$), desde que o sistema no passo 2.3 seja resolvido integralmente.

8.2 Exemplos

Implementamos a resolução da equação de Helmholtz 11.1, pelo método descrito acima, considerando diferentes funções para $\mathcal{F}$. Na nossa implementação, os elementos da base ϕ são splines polinomiais por partes na esfera, homogêneos ou gerais, construídos conforme descritos no capítulo 6. A tabela abaixo relaciona as funções $\mathcal{F}$ usadas nos testes e as soluções exatas correspondentes.

Testes	*Equação*	*Solução da equação*
`band`	$-\Delta\, u(p) + \frac{1}{2}u(p) = 6.5x^2 - 2$	$u(p) = 1 - x^2$
`cubx`	$-\Delta\, u(p) + \frac{1}{2}u(p) = 12.5x^3 - 6x$	$u(p) = x^3$
`expx`	$-\Delta\, u(p) + \frac{1}{2}u(p) = \exp(x)(x^2 + 2x - 0.5)$	$u(p) = \exp(x)$
`mexp`	$-\Delta\, u(p) + \frac{1}{2}u(p) = (\frac{1}{2}\exp(x) + \frac{1}{2}u(x))(x^2 + 2x - 0.5)$	$u(p) = \exp(x)$

Para cada uma dessas funções, aplicamos o algoritmo `UniSolve`, usando os espaços de aproximação $\mathcal{H}_1^5[T]/\mathbf{S}^2$, $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$. Observe que a solução do exemplo `band` pertence aos espaços $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$; a solução de `cubx` pertence aos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$, e as soluções de `expx` e `mexp` nao pertencem a nenhum destes espaços.

8.2.1 Teste `band`

A solução exata para a equação `band`

$$-\Delta u(p) + 0.5u(p) = 6.5x^2 - 2$$

é $u(p) = 1 - x^2$,com $p = (x, y, z)$, ilustrada na figura 8.1.

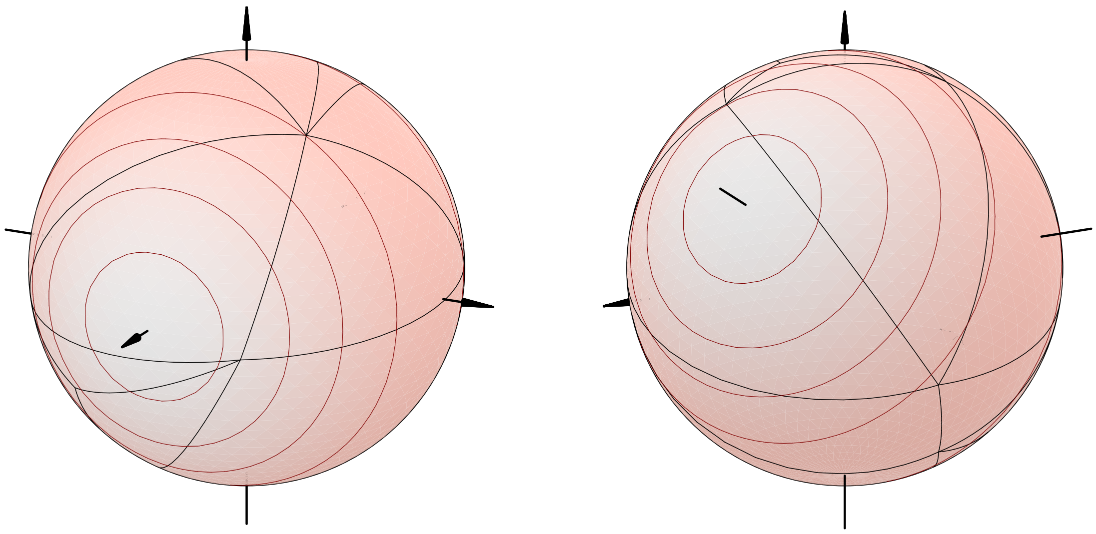


Figura 8.1: Solução exata do teste `band`. O passo das curvas de nivel é $\delta = 0.1$

As figuras 8.2 mostram o erro de aproximação $e^*(p) = |u^*(p) - u(p)|$ para soluções aproximadas $u^*(p)$ nos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$, $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$. Note que a solução pertence aos espaços $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$, o que permitiu obter a aproximação com erro 10^{-04} nestes espaços; enquanto que no espaço $\mathcal{H}_1^5[T]/\mathbf{S}^2$ o erro de aproximação foi da ordem de 10^{-02}.

(a)
$\mathcal{H}_1^5[T]/\mathbf{S}^2$
$e_{\text{max}} = 9.2 \times 10^{-02}$
$\delta = 1.0 \times 10^{-02}$

(b)
$\mathcal{H}_1^6[T]/\mathbf{S}^2$
$e_{\text{max}} = 4.2 \times 10^{-05}$
$\delta = 5.0 \times 10^{-06}$

(c)
$\mathcal{P}_1^6[T]/\mathbf{S}^2$
$e_{\text{max}} = 1.5 \times 10^{-04}$
$\delta = 2.5 \times 10^{-05}$

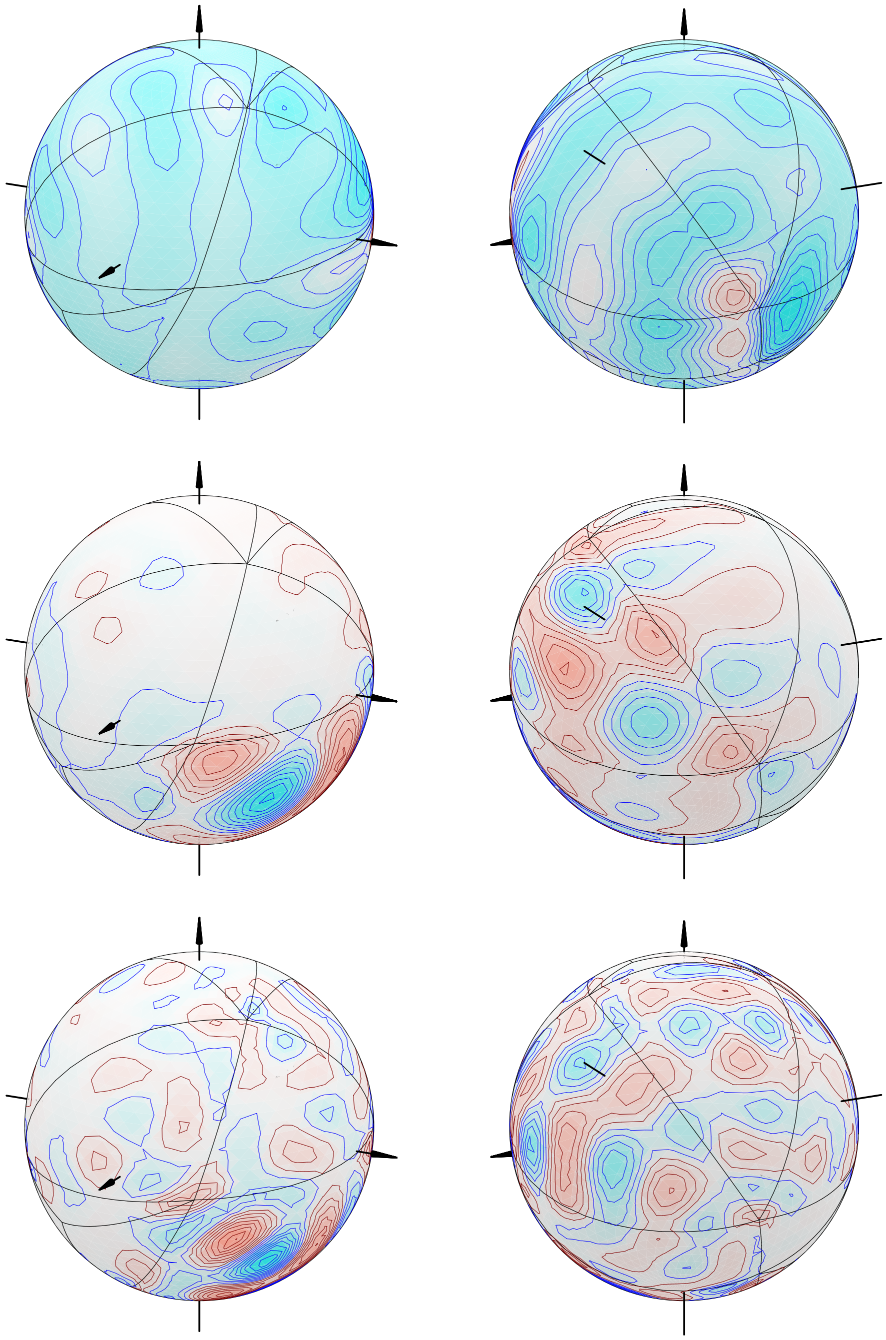


Figura 8.2: Erros de aproximação $e^*(p) = |u(p) - u^*(p)|$. O parâmetro δ é o passo das curvas de nível

8.2.2 Teste cubx

A solução exata da equação `cubx`

$$-\Delta\, u(p) + 0.5u(p) = 12.5x^3 - 6x$$

é $u(p) = x^3$,com $p = (x, y, z)$, ilustrada na figura 8.3.

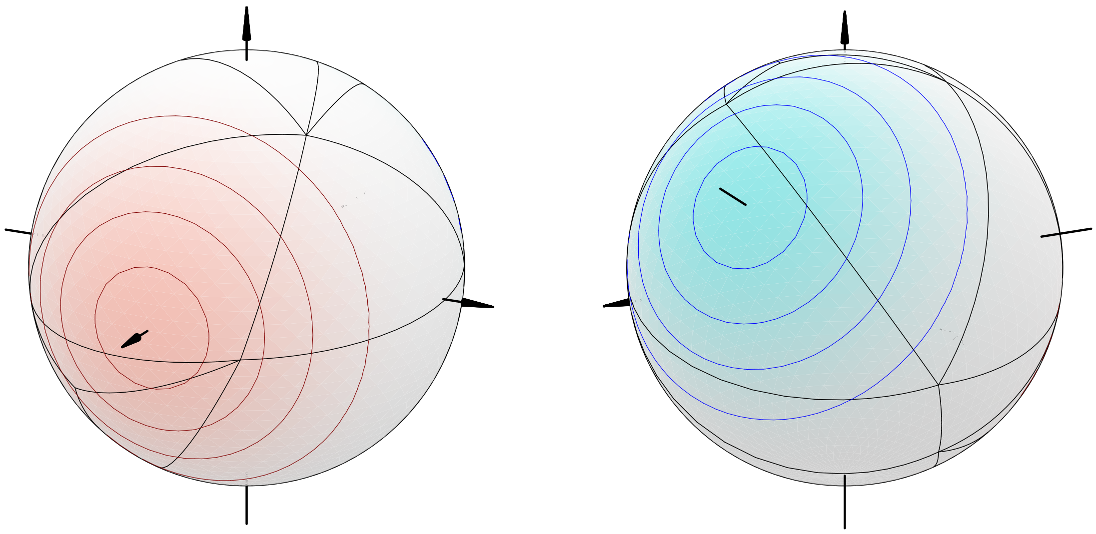


Figura 8.3: Solução exata do teste `cubx`. O passo das curvas de nível é $\delta = 0.1$

As figuras 8.4 mostram o erro $e^*(p) = |u(p) - u^*(p)|$ da solução aproximada $u^*(p)$ nos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$, $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$. Note que a solução x^3 pertence aos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$, o que permitiu obter uma aproximação com erro 10^{-04}; enquanto que no espaço $\mathcal{H}_1^6[T]/\mathbf{S}^2$ o erro da aproximação foi da ordem de 10^{-03}.

(a)
$\mathcal{H}_1^5[T]/\mathbf{S}^2$
$e_{\max} = 6.1 \times 10^{-05}$
$\delta = 1.0 \times 10^{-05}$

(b)
$\mathcal{H}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 9.9 \times 10^{-03}$
$\delta = 5.0 \times 10^{-04}$

(c)
$\mathcal{P}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 1.8 \times 10^{-04}$
$\delta = 2.5 \times 10^{-05}$

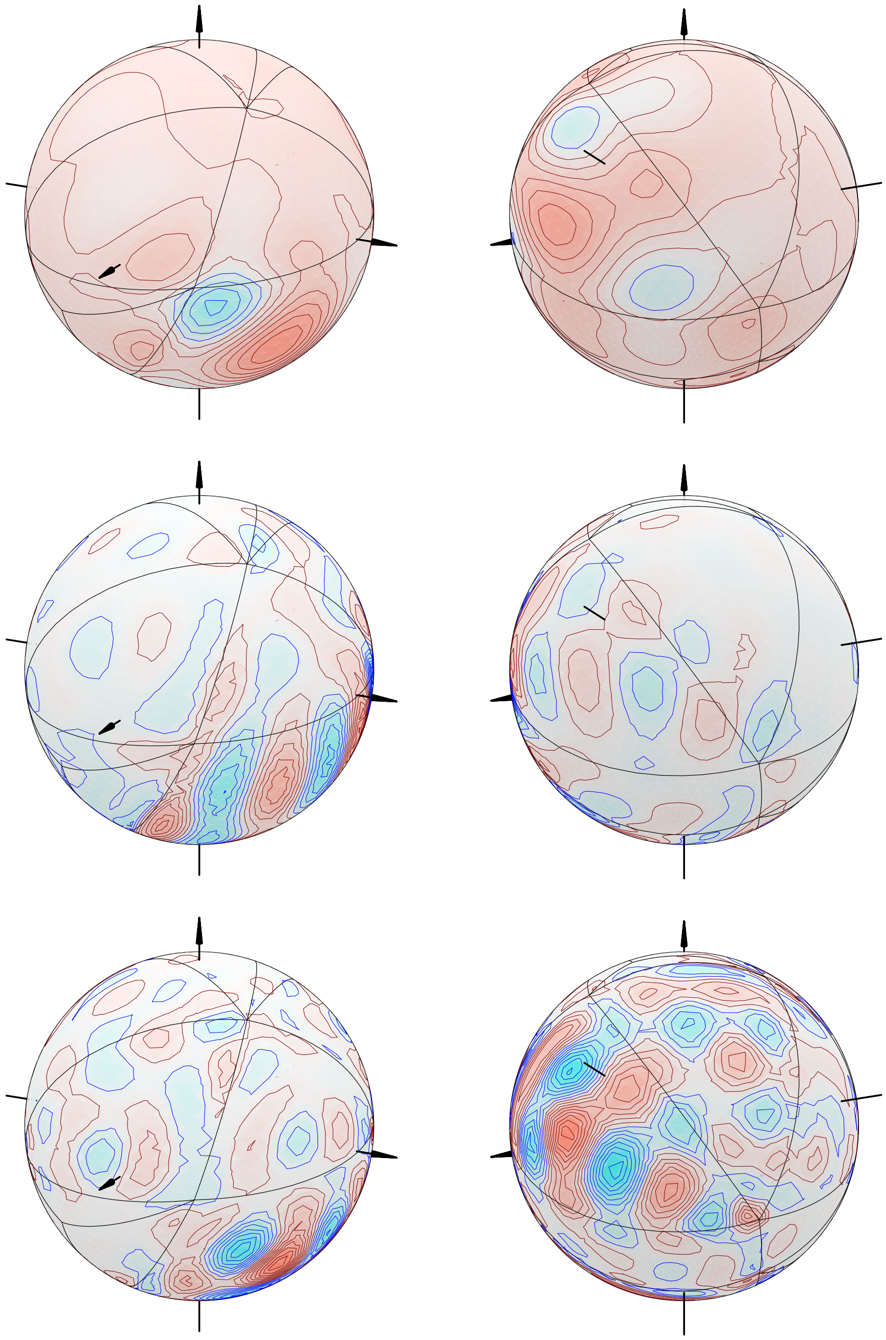


Figura 8.4: Erros de aproximação $e^*(p) = |u(p) - u^*(p)|$. O parâmetro δ é o passo das curvas de nível

8.2.3 Teste `expx`

A figura 8.5 mostra a solução exata da equação

$$-\Delta u(p) + 0.5u(p) = \exp(x)(x^2 + 2x - 0.5)$$

ou seja $u(p) = \exp(x)$ com $p = (x, y, z)$.

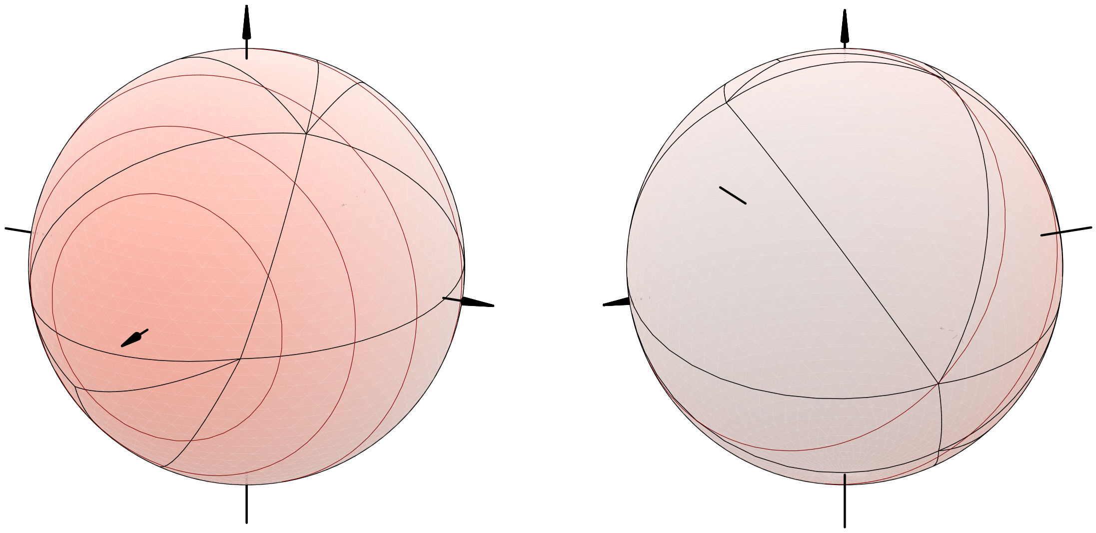


Figura 8.5: Solução exata do teste `expx`. O passo das curvas de nível é $\delta = 0.1$

Como anteriormente, a figura 8.6 mostra os erro $e^*(p) = |u(p) - u^*(p)|$ da solução aproximada $u^*(p)$, nos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$, $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$. Note que a solução exata $\exp(x)$ não pertence a nenhum dos espaços considerados. Como era de se esperar a solução aproximada obtida no espaço $\mathcal{P}_1^6[T]/\mathbf{S}^2$ é melhor (erro máximo 1.8×10^{-04}), enquanto que nos espaços $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{H}_1^5[T]/\mathbf{S}^2$ os erros são da ordem de 10^{-02}.

(a)
$\mathcal{H}_1^5[T]/\mathbf{S}^2$
$e_{\max} = 9.5 \times 10^{-02}$
$\delta = 1.0 \times 10^{-02}$

(b)
$\mathcal{H}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 1.1 \times 10^{-02}$
$\delta = 2.5 \times 10^{-03}$

(c)
$\mathcal{P}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 1.8 \times 10^{-04}$
$\delta = 2.5 \times 10^{-05}$

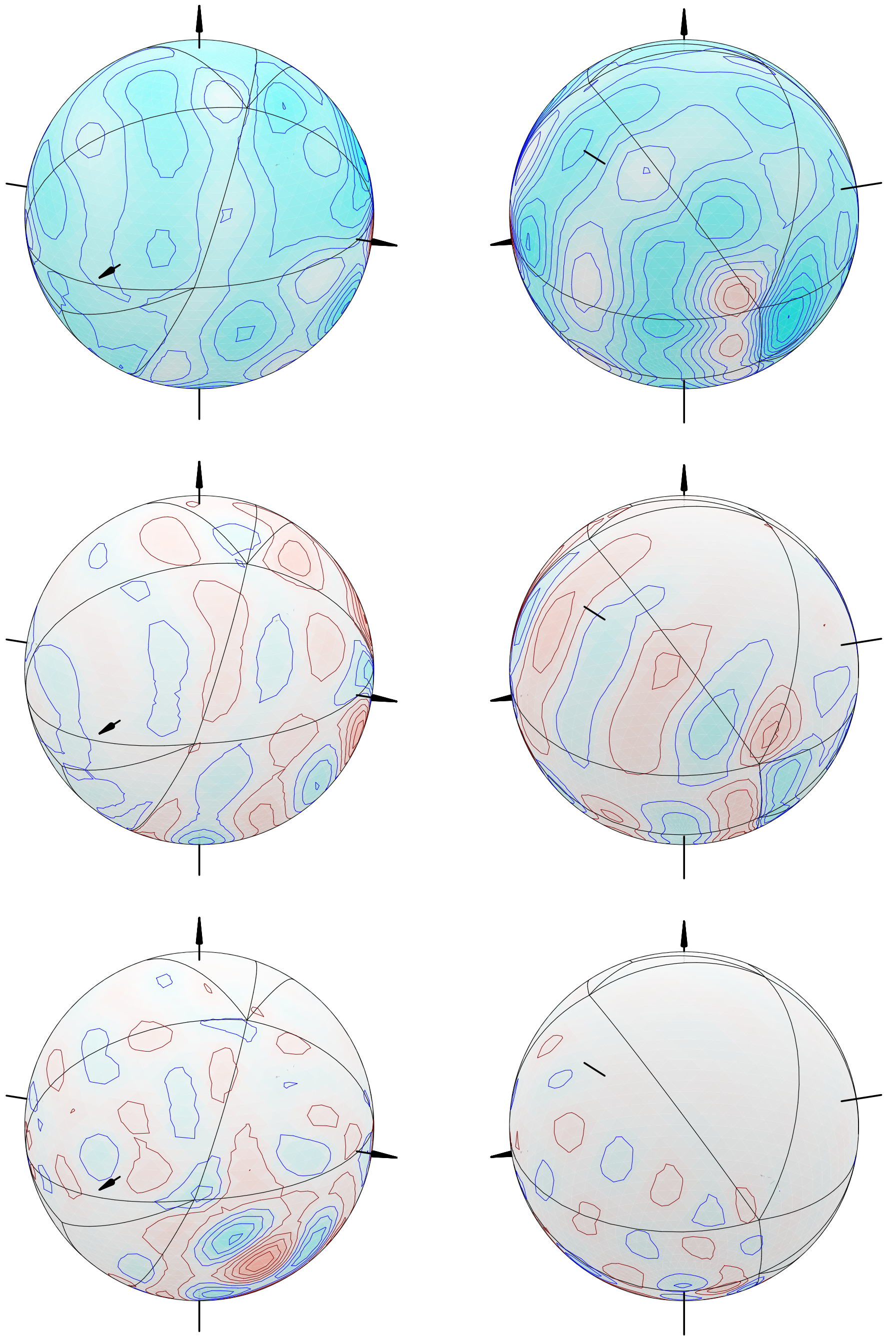


Figura 8.6: Erros de aproximação $e^*(p) = |u(p) - u^*(p)|$. O parâmetro δ é o passo das curvas de nível

8.2.4 Teste `mexp`

Nos exemplos anteriores, consideramos funções $\mathcal{F}$ que não dependem de $u(p)$, permitindo portanto a convergência do procedimento *UniSolve* na segunda iteração. Para ilustrar o caso não linear consideramos o exemplo `mexp` com equação

$$-\Delta u(p) + 0.5u(p) = (0.5\exp(x) + 0.5u(x))(x^2 + 2x - 0.5)$$

com $p = (x, y, z)$. Note que a função $\mathcal{F}$, neste caso, depende de p e da solução $u(p)$. Neste problema, o algoritmo `Unisolve` convergiu após aproximadamente 10 iterações.

A solução exata é $u(p) = \exp(x)$, ilustrada na figura 8.7.

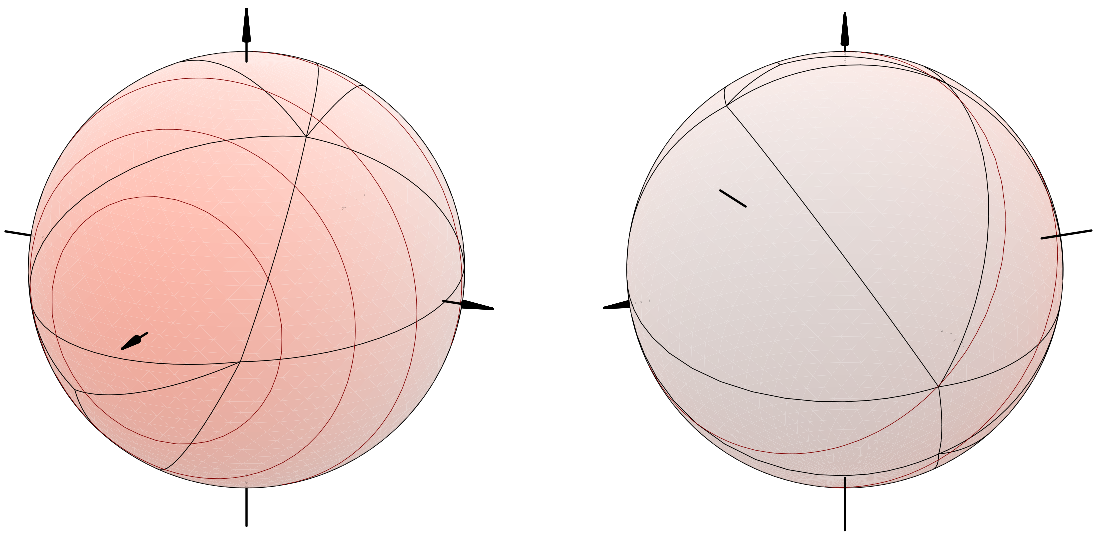


Figura 8.7: Solução exata do teste `mexp`. O passo das curvas de nível é $\delta = 0.1$

A figura 8.8 mostra os erro $e^*(p) = |u(p) - u^*(p)|$ da solução aproximada $u^*(p)$, nos espaços $\mathcal{H}_1^5[T]/\mathbf{S}^2$, $\mathcal{H}_1^6[T]/\mathbf{S}^2$ e $\mathcal{P}_1^6[T]/\mathbf{S}^2$. Observe que, no espaço $\mathcal{P}_1^6[T]/\mathbf{S}^2$, encontramos uma melhor aproximação. Lembramos que a solução exata não pertence a nenhum dos espaços.

(a)
$\mathcal{H}_1^5[T]/\mathbf{S}^2$
$e_{\max} = 9.5 \times 10^{-02}$
$\delta = 2.5 \times 10^{-03}$

(b)
$\mathcal{H}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 1.1 \times 10^{-02}$
$\delta = 2.5 \times 10^{-03}$

(c)
$\mathcal{P}_1^6[T]/\mathbf{S}^2$
$e_{\max} = 1.8 \times 10^{-04}$
$\delta = 2.5 \times 10^{-05}$

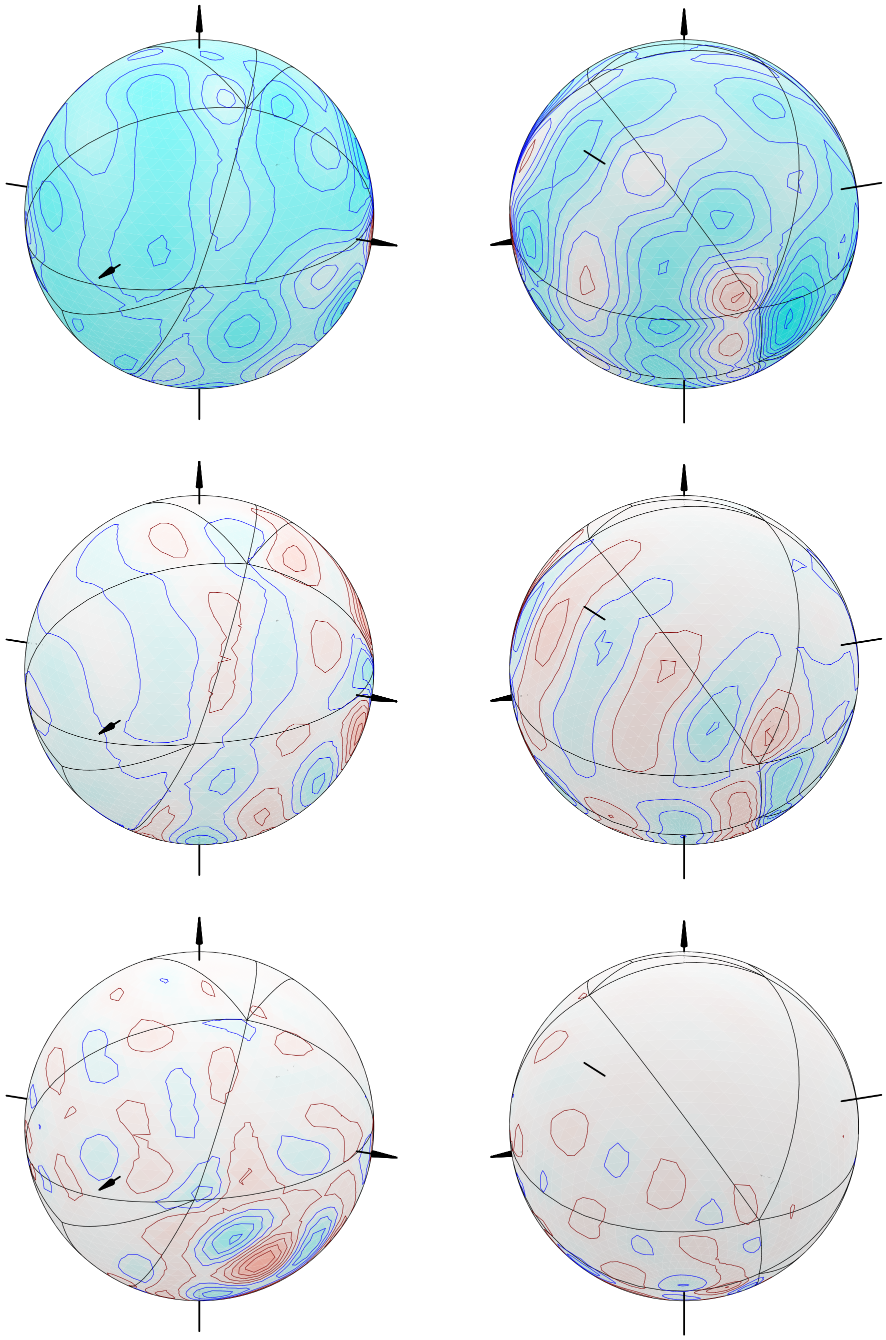


Figura 8.8: Erros de aproximação $e^*(p) = |u(p) - u^*(p)|$. O parâmetro δ é o passo das curvas de nível

A tabela abaixo, resume os erros de aproximação obtidos nestes testes.

	Espaços		
Equação	$\mathcal{H}_1^5[T]/\mathbf{S}^2$	$\mathcal{H}_1^6[T]/\mathbf{S}^2$	$\mathcal{P}_1^6[T]/\mathbf{S}^2$
`band`	9.2×10^{-2}	4.2×10^{-5}	1.5×10^{-4}
`cubx`	6.1×10^{-5}	9.9×10^{-3}	1.3×10^{-4}
`expx`	9.5×10^{-2}	1.1×10^{-2}	1.8×10^{-4}
`mexp`	9.5×10^{-2}	1.2×10^{-2}	1.8×10^{-4}

Capítulo 9

Métodos Multi-Escala

9.1 Técnicas de múltiplas escalas

Como vimos no capítulo 8, a resolução de equações diferenciais parciais na esfera recai num sistema linear cuja ordem é proporcional ao número de elementos da base, ou seja ao número de triângulos da malha. Como observamos, a solução direta do sistema (por fatoração da matriz) é inviável no caso de malhas com centenas de triângulos. Nestes casos, é necessário usar métodos iterativos para resolver o sistema linear. Estes métodos têm convergência bastante lenta, pois os ajustes efetuados em cada iteração propagam-se de triângulo para triângulo num processo análogo ao da difusão do calor. Isto significa que o número de iterações para atingir uma determinada tolerância é aproximadamente proporcional ao quadrado do diâmetro da malha.

Os métodos mais eficientes para integração de equações diferenciais parciais utilizam a técnica de *mútiplas escalas* [13] para acelerar a convergência do processo iterativo. A idéia é resolver a mesma equação numa hierarquia de várias malhas $M_0, M_1, \ldots, M_m$, com resoluções decrescentes, sendo que a solução parcial obtida na malha grosseira M_{i+1} é usada como estimativa inicial para a malha mais fina M_i.

9.2 Algoritmo Multi-Escala

Apresentamos a seguir o procedimento *MultiSolve* que implementa esta técnica. O algoritmo recebe como entrada um conjunto de bases de splines esféricos $\phi = (\phi^r : r = 0 \ldots m)$ associadas a uma hierarquia de triangulações esféricas $T^0, T^1, \ldots, T^m$, onde T^0 é a mais fina e T^m a mais grossa; a função $\mathcal{F}$, uma tolerância τ, que determina a condição de término do algoritmo, e um número máximo de iterações $k_{\max}$. O procedimento devolve uma lista de vetores $(a^r : r = 0, \ldots, m)$ onde a^r é o vetor de coeficientes da solução aproximada da equação, no espaço $\mathcal{V}^r$ gerado pela base ϕ^r.

Algoritmo 7 *Procedimento* `MutiSolve`$(\phi, \mathcal{F}, \tau, k_{\max})$

`1. para` $r \leftarrow m$ `até` 0 `faça`

`1.1. se` $r = m$ `então seja` $a^{(0)}$ `um vetor inicial arbitrário; senão seja` $a^{(0)}$ `o vetor de coeficientes da projeção da função` $\sum_i a_i^{(k)} \phi_i^{r+1}$ `no espaço da base` ϕ^r

`1.2.` $k \leftarrow 0$

`1.3. repita`

`1.3.1. calcule a função` $u^{(k)} = \sum_{i=1}^{n} a_i^{(k)} \phi_i^r$

`1.3.2. calcule o vetor` $b^{(k)}$ `onde` $b_i^{(k)} = \int_{\mathbf{S}^2} \phi_i^r \mathcal{F}(u^{(k)}(p), p) dp$

`1.3.3. resolve o sistema` $H^r a^{(k+1)} = b^{(k)}$ `obtendo novos coeficientes` $a^{(k+1)}$.

`1.3.4. seja` $\delta = \|a^{(k)} - a^{(k+1)}\|$

`1.3.5.` $k \leftarrow k + 1$

`até que` $\delta \leq \tau$ `ou` $k > k_{\max}$

9.3 Transferindo aproximações

No passo 1.1 do procedimento `MultiSolve`, após obter uma solução aproximada u^{r+1} no espaço $\mathcal{V}^{r+1}$ da base ϕ^{r+1}, projetamos esta solução para o espaço $\mathcal{V}^r$ da base ϕ^r, de dimensão maior. Esta operação é feita da seguinte maneira. Sejam ϕ_i^r, $i = 1 \ldots n$ e ϕ_j^{r+1}, $j = 1 \ldots m$ elementos das bases ϕ^r e ϕ^{r+1} respectivamente. Como o espaço $\mathcal{V}^{r+1}$, em geral, não está contido em $\mathcal{V}^r$, precisamos calcular a aproximação $u^r = \sum_{i=1}^{n} a_i^r \phi_i^r$ de u^{r+1} por mínimos quadrados, conforme visto na seção 7.1. Podemos entretanto aproveitar o fato que a função u^{r+1} a ser aproximada é combinação da base ϕ^{r+1} isto é $u^{r+1} = \sum_{i=1}^{m} a_i^{r+1} \phi_i^{r+1}$. O sistema $Gx = b$ do método de mínimos quadrados reduz-se, neste caso, a $G^r a^r = F^r a^{r+1}$, onde $G_{ij}^r = \int_{\mathbf{S}^2} \phi_i^r \cdot \phi_j^r ds$, $i, j = 1 \ldots n$ e $F_{ij}^r = \int_{\mathbf{S}^2} \phi_i^r \cdot \phi_j^{r+1} ds$, $i = 1 \ldots n$, $j = 1 \ldots m$. A solução desse sistema é encontrada calculando-se primeiro o vetor $b = F^r a^{r+1}$ e resolvendo-se o sistema $G^r a^r = b$ como descrito na seção 7.2. Como observado na seção 7.4, essas matrizes são esparsas, e o custo de calculá-las é $O(n_t)$. Note que os vetores coluna da matriz $(G^r)^{-1} F^r$ geram a projeção de $\mathcal{V}_{r+1}$ em $\mathcal{V}_r$ [14].

9.4 Refinamento da Triangulação

Para aplicação do algoritmo precisamos construir uma hierarquia de triangulações esféricas $T^0, T^1, \ldots, T^m$. Para o nível mais fino T^0, usamos a triangulação de Delaunay esférica, descrita no capítulo 4.2. Para a construção das triangulações mais grosseiras, podemos usar o algoritmo de simplificação de Kirkpatrick [12], resumido a seguir.

Dada uma triangulação T^r com vértices V^r, encontramos um *subconjunto independente* X^r de V^r, isto é um subconjunto de vértices, dois a dois não adjacentes. Retiramos em seguida estes vértices e todas as arestas de T^r incidentes a X^r. Como X^r é um conjunto independente, cada vértice retirado dá origem a uma nova face em T^r. Retriangularizamos estas faces, obtendo assim a triangulação T^{r+1}.

O conjunto independente X^r é construído repetindo-se o seguinte algoritmo "guloso": enquanto houver algum vértice v de grau ≤ 6, que não seja adjacente a nenhum outro vértice previamente escolhido, colocamos v no conjunto X^r.

Observe que a quantidade de vizinhos de v é pequena, porque v tem grau pequeno. Isto significa que o tamanho do conjunto X^r é pelo menos $\lfloor n_v/7 \rfloor$ onde n_v é o número de vértices de T^r. Além disso, cada face pode ser retriangularizada em tempo constante. O tempo necessário para construir a triangulação é uma constante vezes o número de vértices retirados – portanto $O(n_v)$. O tamanho de T^{r+1} é no máximo $n_v - \lfloor n_v/7 \rfloor \simeq 6/7 n_v$, portanto o número de triangulações na hierarquia é no máximo $O(\log n_v)$.

Capítulo 10

Conclusões

O principal resultado deste trabalho é o desenvolvimento e análise experimental de uma classe de espaços funcionais para aproximação na esfera, os *splines polinomiais não homogêneos*, $\mathcal{P}_r^d[T]/\mathbf{S}^2$. Esses espaços combinam a simplicidade analítica dos polinômios esféricos com a localidade e adaptabilidade dos *splines* tradicionais. A classe inclui espaços de funções $\mathbf{C}_1$, que podem ser usados para integração de equações diferenciais na esfera.

10.1 Comparações teóricas

Vimos que os espaços de splines polinomiais homogêneos, propostos por Alfeld, Neamtu e Schumaker [2] para estas aplicações, têm certas limitações práticas; por exemplo a função trivial $f(p) = 1$ só pode ser representada em $\mathcal{H}_r^d[T]/\mathbf{S}^2$ quando d é par. Neste trabalho, nós mostramos que o espaço $\mathcal{P}_r^d[T]/\mathbf{S}^2$, dos splines polinomiais não homogêneos, é uma escolha mais natural para estas aplicações. Estes espaços satisfazem a propriedade $\mathcal{P}_r^k[T]/\mathbf{S}^2 \subseteq \mathcal{P}_r^d[T]/\mathbf{S}^2$ quando $k \leq d$. Em particular, as funções que são constantes sobre $\mathbf{S}^2$ estão em $\mathcal{P}_r^d[T]/\mathbf{S}^2$ para todo d e r.

10.2 Construção das bases

Descrevemos no capítulo 6 um algoritmo para construção das bases de $\mathcal{P}_r^d[T]/\mathbf{S}^2$ para $d \geq 3r + 3$ e $r = 0, 1$. O algoritmo é baseado na construção da base para $\mathcal{H}_r^d[T]/\mathbf{S}^2$, de Alfeld, Neamtu e Schumaker. As dimensões dos espaços são dadas pela tabela abaixo.

A representação da base é relativamente compacta, pois seus elementos têm suporte restrito a um pequeno conjunto de triângulos. Para triangulações bem forma-das, em que o grau de cada vértice é limitado por uma constante, o número total de triângulos usados por todos os elementos da base é proporcional ao número de triângulos n_t.

O fato de terem os elementos da base suporte limitado também faz com que as matrizes de produtos escalares $\langle g_i, g_j \rangle$ tenham $O(n_t)$ elementos não nulos, em vez de $O(n_t^2)$. Além disso, o cálculo de $\langle g_i, g_j \rangle$ implica no cálculo da integral $\int g_i(p) \cdot g_j(p) dp$ sobre um número constante de triângulos, independente de n_t. Logo, a construção da base pode ser feita em tempo $O(n_t)$, e portanto é factível mesmo para malhas com milhares de triângulos.

O passo crítico deste processo é o cálculo da integral de uma função sobre um triângulo esférico; qualquer aceleração desse passo implicaria numa aceleração proporcional de todas as aplicações.

10.3 Aproximação de funções

No capítulo 7 analisamos o uso dos espaços $\mathcal{P}_r^d[T]/\mathbf{S}^2$ para aproximar funções esféricas, pelo método dos mínimos quadrados. As soluções obtidas com estes espaços têm precisão mais uniforme do que as obtidas com os espaços $\mathcal{H}_r^d[T]/\mathbf{S}^2$. Parte dessa vantagem é devida ao fato que um spline de $\mathcal{P}_r^d[T]/\mathbf{S}^2$ tem mais graus de liberdade que um spline de $\mathcal{H}_r^d[T]/\mathbf{S}^2$. Porém esta vantagem não parece ser suficiente para explicar a diferença entre os erros de aproximação obtidos em testes típicos, mostrados na tabela abaixo.

Espaços	`dim`	*erro típico*
$\mathcal{H}_0^2[T]/\mathbf{S}^2$	$2n_t + 2$	1.2×10^{-01}
$\mathcal{H}_0^3[T]/\mathbf{S}^2$	$4.5n_t + 2$	1.7×10^{-01}
$\mathcal{P}_0^3[T]/\mathbf{S}^2$	$6.5n_t + 4$	1.2×10^{-03}
$\mathcal{H}_1^5[T]/\mathbf{S}^2$	$6n_t + 6$	6.1×10^{-02}
$\mathcal{H}_1^6[T]/\mathbf{S}^2$	$10n_t + 6$	9.6×10^{-03}
$\mathcal{P}_1^6[T]/\mathbf{S}^2$	$16n_t + 12$	1.5×10^{-07}

Nesta tabela `dim` é a dimensão do espaço (em função de n_t), e *erro típico* é o erro obtido na aproximação da função $\exp(x)$ com $n_t = 16$.

10.4 Integração de equações diferenciais parciais

No capítulo 8 utilizamos o espaço $\mathcal{P}_r^d[T]/\mathbf{S}^2$ na solução de equações diferenciais parciais, definidas em $\mathbf{S}^2$, pelo método de Galerkin. Numa triangulação com 16 triângulos, obtivemos uma aproximação com erro da ordem de 10^{-4} em aproximadamente 10 iterações.

Estes resultados foram obtidos resolvendo-se o sistema de equações (8.8) pelo método direto de Choleski. Infelizmente não pudemos usar este método para triangulações mais finas, pois o algoritmo de construção da base gera muitos elementos semelhantes entre si. Isso faz com que a matriz de produtos escalares seja mal condicionada, impossibilitando o cálculo de sua fatoração de Choleski.

Tentamos contornar esta dificuldade usando os métodos iterativos (Gauss-Seidel e gradiente conjugado) para achar a solução do sistema. Entretanto, pela mesma razão, a matriz não é diagonalmente dominante, o que torna a convergência desses métodos extremamente lenta. A semi-ortogonalização da base descrita na seção 6.6 melhorou um pouco a convergência, mas não o suficiente para tornar o método prático.

10.5 Técnicas de múltiplas escalas

Para contornar o problema da convergência lenta, consideramos a técnica de múltiplas escalas. Como explicado no capítulo 9, a justificativa desta técnica é que as malhas mais grosseiras forneçem um “atalho” entre partes diferentes da malha mais fina, permitindo que os erros se propaguem mais rapidamente para toda a malha. Entretanto, como não conseguimos convergência dos métodos iterativos, mesmo nas malhas mais grosseiras, o mérito desta técnica não pode ser analisado. Por outro lado, para malhas em que foi possível resolver o sistema pelo método de Choleski, encontramos uma boa aproximação da solução com um número de iterações bastante pequeno, portanto não foi necessário utilizar o método multi-escala.

Para construção da hierarquias das malhas, desenvolvemos e implementamos um algoritmo baseado na triangulação de Delaunay esférica para o nível mais fino [9] e no algoritmo de simplificação de Kirkpatrick para a construção das malhas mais grosseiras [12] como descrito na seção 9.4. Um problema que encontramos é que estes métodos geram triangulações que frequentemente têm arestas quase coplanares incidentes ao mesmo vértice, que impedem a construção da base. Para contornar este problema, desenvolvemos um procedimento que aplica uma pertubação aos vértices da malha, mantendo sua topologia (seção 4.5). Este procedimento procura minimizar uma função objetivo que mede as violações de várias propriedades exigidas da triangulação, incluindo arestas coplanares.

10.6 Trabalhos futuros

Uma linha de pesquisa que pode ter um impacto prático significativo é a construção de bases para $\mathcal{P}_r^d[T]/\mathbf{S}^2$ mais ortogonais do que as que utilizamos. Este objetivo implica em utilizar elementos com suportes maiores. Um limite para o que pode ser conseguido nesta direção é o teorema de Alfeld e Schumaker [3], que diz que não existem bases de $\mathcal{H}_r^d[T]/\mathbf{S}^2$, para $r > 0$ e $d \leq 3r + 1$, cujos suportes sejam restritos às estrelas dos vértices.

Outro problema que precisa ser resolvido é o desenvolvimento de algoritmos mais eficientes para resolução do sistema normal (capítulo 7), e do sistema proveniente do método de Galerkin (capítulo 8). Está em aberto encontrar métodos de solução, ou pré-condicionadores, para os métodos de Gauss-Seidel e gradiente conjugado, que nos permitam resolver este sistema em malhas realistas (com 100 ou mais triângulos).

Finalmente, há muito espaço para pesquisa na geração de triangulações adaptativas sobre a esfera. Um desafio é desenvolver algoritmos para gerar subdivisão local de uma malha triangular, sem introduzir arestas coplanares incidentes em um mesmo vértice.

Capítulo 11

Apêndice - Descrição do Software

Os métodos numéricos descritos neste trabalho, foram implementados por nós, em Modula-3 [25, 26], e totalizam 16 mil linhas de código (mais 2000 linhas de programas de testes). Neste capítulo nós descrevemos as principais componentes deste software. Maiores detalhes podem ser encontrados nas interfaces dos módulos de biblioteca, e nos comentários iniciais dos programas.

11.1 Representação de splines esféricos

Para representar os splines esféricos, usamos uma hierarquia de classes de objetos ilustrada na figura 11.1. Cada uma dessas classes é definida num módulo separado da biblioteca.

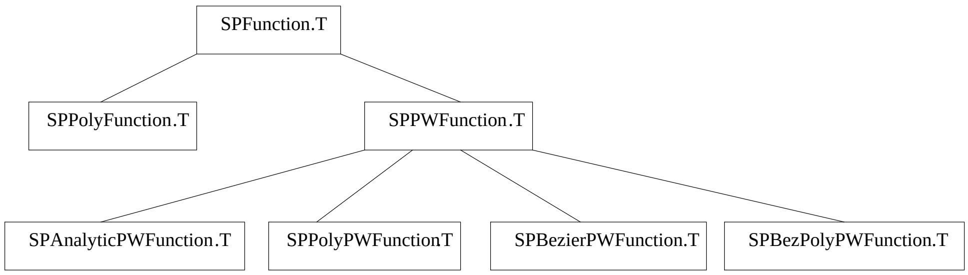


Figura 11.1: Hierarquia de representações de funções esféricas

11.1.1 `SPFunction`

O tipo `SPFunction.T` é uma classe abstrata que representa uma função esférica geral. O tipo define dois métodos, `eval` que avalia a função em um ponto p, e `grad` que avalia o gradiente da função no ponto.

O módulo `SPFunction` define também os seguintes procedimentos:

- `SamplePoints(`n`)` - gera n pontos uniformemente distribuidos na esfera.
- `Sample(`f`,`P`)` - avalia a função f em cada ponto p_i de um conjunto de pontos P e retorna estes resultados em um vetor.
- `DotProduct(`f`,`g`,`k`)` - calcula o produto escalar $\langle f, g\rangle$ das funções f e g sobre toda a esfera, usando a fórmula de quadratura descrita na seção 7.3. A triangulação usada é a divisão da esfera em octantes subdividida em k vezes; portanto, este procedimento usa uma triangulação quase uniforme com 8×4^k triângulos.
- `CustomDotProduct(`f`,`g`,`p`)` - calcula o produto escalar de duas funções f e g definido por
$$\frac{4\pi}{N} \cdot \sum_{k=1}^{N} f(p_k) \cdot g(p_k)$$
onde $p_1, \ldots, p_N$ é um conjunto de pontos na esfera fornecido pelo cliente.

11.1.2 `SPPolyFunction`

O tipo `SPPolyFunction.T` representa um polinômio esférico p. Este objeto tem os seguintes campos visíveis pelos clientes: `deg` que define o grau d do polinômio; `c0`, o vetor dos coeficientes de um polinômio homogêneo p' de grau `deg`; e `c1`, os coeficientes de um polinômio homogêneo p'' de grau `deg`-1, tais que $p = p' + p''$. Os elementos de `c0` e `c1` são os coeficientes c'_{ijk} e c''_{ijk} das formas canônicas

$$p'(x, y, z) = \sum_{i+j+k=d} c'_{ijk} x^i y^j z^k$$

$$p''(x, y, z) = \sum_{i+j+k=d-1} c''_{ijk} x^i y^j z^k$$

armazenados em ordem lexicográfica dos índices i, j, k. (veja 11.1.8)

11.1.3 `SPPWFunction`

O tipo `SPPWFunction.T` representa uma função definida por partes (*PieceWise*) na esfera. Este objeto tem os seguintes campos: uma triangulação `tri`, um plano suporte `supp`, e um vetor `data` contendo a descrição da função sobre os triângulos de `tri` onde ela não é nula. Além dos métodos `eval` e `grad` herdados da classe `SPFunction.T`, o objeto `SPPWFunction.T` tem o método `locate(`p`)` que encontra o triângulo da triangulação que contém o ponto p, como descrito na seção 5.4.3.

Todas as classes de objetos têm também um método `write` que armazena a descrição da função em um arquivo, num formato bem especificado; e um método `read` que lê descrição da função de um arquivo nesse formato.

Cada elemento do vetor `data` é um objeto de tipo `SPPWFunction.TriangleData`. Esta classe abstrata define um campo visível `face`, que guarda o número do triângulo na triangulação, e os seguintes métodos:

- `eval(`p`)` - avalia a função em um ponto p do triângulo;
- `grad(`p`)` - calcula o gradiente da função em um ponto p do triângulo;
- `type()` - devolve um caracter que indica o tipo de função associada a este triângulo;
- `copy()` - faz uma cópia do triângulo e o inicializa com a função nula;
- `scale(`a`)` - substitui a função f do triângulo por $a \cdot f$ onde a é um número real.
- `add(`a`,`t`)` - substitui o função f do triângulo por $f + a \cdot g$ onde g é a função do triângulo t, e a é um número real;

O módulo também define o tipo `SPPWFunction.Basis`, que é simplesmente um vetor de objetos do tipo `SPPWFunction.T`, e os seguintes procedimentos:

- `WriteBasis(`b`)` - escreve a base b em um arquivo.
- `ReadBasis(`b`)` - lê uma base b de um arquivo.
- `ComputeSupportingPlane(`t`)` - recebe um conjunto de triângulos $t_0, t_1, \ldots, t_n$ e retorna um plano suporte para os mesmos (isto é, um plano π tal que todos os triângulos estão do lado positivo de π).
- `DotProduct(`f`,`g`)` - calcula o produto escalar $\langle f, g \rangle$ de duas funções f e g, como descrito na seção 7.3.
- `GradDotProduct(`f`,`g`)` - calcula o produto escalar, $\langle \nabla f, \nabla g \rangle$, do gradiente de duas funções f e g, como descrito na seção 7.3.
- `LinComb(`a`,`ϕ`)` - dado um vetor de pesos $a_0, a_1, \ldots, a_n$ e uma base $\phi_1, \phi_2, \ldots, \phi_n$, retorna a combinação linear das funções ϕ_i com pesos a_i (ou seja, $\sum_{i=1}^{n} a_i \cdot \phi_i$).
- `BuildMatrix(`ϕ`,`ψ`)` - dadas duas bases ϕ e ψ, constrói uma matriz cujo elemento i, j é o produto escalar $\langle \phi_i, \psi_j \rangle$. A matriz é representada como descrito na seção 11.3.

- `BuildMatrixGrad(`ϕ`,`ψ`)` - dadas duas bases ϕ e ψ, constrói uma matriz cujo elemento i, j é o produto escalar $\langle \nabla \phi_i, \nabla \psi_j \rangle$.

- `ChangeCoordsBasis(`a`,`F`,`L`)` - dados os coeficentes $a_0, a_1 \ldots a_n$ de uma função em relação a uma base $\phi_1, \phi_2, \ldots, \phi_n$, este procedimento calcula os coeficientes $b_0, b_1, \ldots, b_n$ dessa função em relação a uma outra base $\psi_1, \psi_2 \ldots, \psi_m$, pelo critério dos mínimos quadrados. O procedimento necessita das matrizes $F_{i,j} = \langle \phi_i, \psi_j \rangle$ e do fator de Choleski L da matriz $G_{i,j} = \langle \psi_i, \psi_j \rangle$.

- `SortTriangles(`t`)` - ordena uma lista $t_0, t_1, \ldots, t_n$ de triângulos em ordem crescente pelo número do triângulo.

11.1.4 `SPAnalyticPWFunction`

O tipo `SPAnalyticPWFunction.T` representa uma função esférica definida proceduralmente por partes. A classe concreta `SPAnalyticPWFunction.TriangleData` (descendente de `SPPWFunction.Triangle.Data`) contém os seguintes campos:

- `evalProc(`p`)` - um procedimento que avalia a função num ponto dado p.

- `gradProc(`p`)`- um procedimento que avalia o gradiente num ponto dado p.

- `descr` - uma cadeia de caracteres que contém a descrição analítica (fórmula) da função, para fins de documentação.

O módulo `SPAnalyticPWFunction` define também os seguintes procedimentos:

- `Uniform(`T`,`$\mathcal{P}$`,`$\mathcal{G}$`,`D`)` - cria um objeto `SPAnalyticPWFunction.T` com uma triangulação dada T, em que todos os objetos `TriangleData` contém os mesmos campos `evalProc` $= \mathcal{P}$, `gradProc` $= \mathcal{G}$ e `descr` $=$ `D`.

- `FromName(`N`,`T`)` - retorna uma função analítica por partes sobre a triangulação T, que, em todos os triângulos, é a mesma função, identificada pelo nome N (uma cadeia de caracteres). Este procedimento reconhece apenas alguns nomes fixos, por exemplo `expx`, para $\exp(x)$, etc.

11.1.5 `SPPolyPWFunction`

O tipo `SPPolyPWFunction.T` é uma função polinomial por partes na esfera, onde cada parte (um polinômio esférico) é representada pelos seus coeficientes na base canônica. O objeto `SPPolyPWFunction.TriangleData` que descreve a parte da função restrita a um triângulo t, contém os seguintes campos:

- `deg` que é grau d do polinômio no triângulo t;
- `c0` e `c1` que são os coeficientes dos polinômios homogêneos, na base canônica, de grau d e $d-1$ respectivamente, como no tipo `SPPolyFunction`.T.

O módulo `SPPolyPWFunction` define também os seguintes procedimentos:

- `TriangleIsNull(`t`)` - retorna verdadeiro se a função polinomial definida no triângulo t (um objeto do tipo `TriangleData`) é nula.

11.1.6 `SPBezierPWFunction`

O tipo `SPBezierPWFunction.T` é um spline esférico homogêneo, onde cada parte (um polinomio homogêneo esférico) está representada pelos seus coeficientes de Bézier, como descrito na seção 5.2. O objeto `SPBezierPWFunction.TriangleData` contém os seguintes campos:

- `deg` que é grau d do polinômio no triângulo;
- `c0` que são os coeficientes de Bézier do polinômio relativos a esse triângulo.

11.1.7 `SPBezPolyPWFunction`

O tipo `SPBezPolyPWFunction.T` é um spline esférico não homogêneo, onde cada parte (um polinômio genérico) está representada pelos coeficientes de Bézier de suas componentes homogêneas, como descrito na seção 5.2.

O objeto `SPBezPolyPWFunction.TriangleData` contém os seguintes campos:

- `deg` que é grau d do polinômio no triângulo;
- `c0` e `c1` que são os coeficientes de Bézier, relativos ao triângulo, dos polinômios de grau d e $d-1$ respectivamente, cuja soma é o polinômio em questão.

O módulo `SPBezPolyPWFunction` define também os seguintes procedimentos:

- `FromHomo(`t`,`d`)` - converte um spline polinomial esférico homogêneo t, de grau d ou $d-1$ de tipo `SPBezierPWFunction.T`, para um spline polinomial esférico não homogêneo de grau d, de tipo `SPBezPolyPWFunction.T`.

11.1.8 Módulos auxiliares

Os módulos que descreveremos a seguir contêm procedimentos auxiliares usados na implementação de alguns dos módulos descritos acima.

O módulo `SPDeCasteljau` fornece procedimentos para calcular o valor e o gradiente de uma função de $\mathbf{R}^n$ para $\mathbf{R}$, dada pelos seus coeficientes de Bézier, usando os algoritmos descritos na seção 5.4.1.

O módulo `HomoLabel` fornece procedimentos que mapeiam os índices i, j, k dos coeficientes de Bézier (como definido na fórmula (5.1)) para um conjunto de inteiros consecutivos entre 0 e $(d+1)(d+2)/2 - 1$, e vice-versa. Estes procedimentos também podem ser usados para mapear os índices dos coeficientes da base canônica (fórmula (2.1)).

O módulo `SPFile` prepara os arquivos para os métodos de `read` e `write` de todos os objetos acima descritos, cuidando da leitura e escrita de cabeçalhos padronizados.

11.2 Representação da triangulação

No módulo `SPQuad` implementamos a estrutura de dados `quad-edge` [9] usada para representar a triangulação e as funções de percurso. Implementamos também procedimentos para enumeração e numeração dos arcos.

O módulo `SPTriang` especializa a estrutura `quad-edge` para representar triangulações geodésicas da esfera. Ele contém procedimentos para enumeração e numeração de vértices e triângulos. Este módulo define também uma representação auxiliar da triangulação, `SPTriang.T`, na forma de três vetores que listam os vértices, arestas e faces. Esta representação inclui, para cada triângulo, matrizes que transformam as coordenadas cartesianas de um ponto em coordenadas baricêntricas relativas aos vértices, e vice-versa. Este módulo define também o tipo `Site`, que representa um vértice da triangulação, e contém o número do vértice e suas coordenadas cartesianas; e o tipo `Triangle`, que representa uma face da triangulação, e contém o número do triângulo. Implementamos também um algoritmo incremental para construir a triangulação de Delaunay de um conjunto de pontos na esfera fornecidos pelo cliente.

11.3 Matrizes esparsas

No módulo `SPMatrix` definimos a estrutura de dados usada para armazenar uma matriz esparsa. Para cada elemento a_{ij} da matriz, diferente de zero, guardamos a linha i, a coluna j, e seu valor a_{ij}. Armazenamos esses elementos em um vetor, ordenados por linha, e, em cada linha, pela coluna.

Este módulo implementa os procedimentos básicos de operações com matrizes como somar e multiplicar duas matrizes, e multiplicar um vetor pela matriz. Implementamos também alguns procedimentos específicos utilizados para resolução de sistemas lineares:

- `DivRow(`y`,`L`,`x`)` multiplica um vetor linha y pela inversa de uma matriz triangular inferior L, e devolve o resultado no vetor x.
- `DivCol(`L`,`y`,`x`)` multiplica a inversa da matriz triangular inferior L pelo vetor coluna y e devolve o resultado no vetor x.
- `Choleski(`A`,`L`)` fatora uma matriz positiva definida A na forma $LL^\top$, onde L é triangular inferior e $L^\top$ é sua transposta.

11.4 Solução de sistemas lineares

O módulo `SPLinearSystem` implementa os procedimentos que resolvem um sistema linear pelos métodos de Choleski (seção 7.2), Gauss-Seidel [27] e gradiente conjugado [27]. Além disso, o módulo inclui um procedimento que compara uma solução aproximada, da forma $\sum_{i=1}^{n} a_i \cdot g_i$ com uma solução exata dada f, e calcula o erro máximo (por amostragem). Outros procedimentos desenham a solução aproximada e a função erro, como nas ilustrações dos capítulos 7 e 8.

11.5 Integração na esfera

No módulo `SPIntegral` implementamos um procedimento, `OnTriangle(`f`,`t`,`k`)`, que calcula a integral de uma função f em um triângulo esférico t. O triângulo é subdividido em 4^k partes, e a integral em cada sub-triângulo é calculada por uma fórmula de quadratura gaussiana com 13 nós (seção 7.3). A função f é dada na forma de um procedimento, e portanto não está presa a nenhuma representação específica.

Outro procedimento `OnTwoTriangles(`t_1`,`t_2`,`f`,`k`)` calcula a integral de uma função f sobre a interseção de dois triângulos arbitrários t_1, t_2 dados. O procedimento decompõe a interseção dos triângulos em um ou mais triângulos disjuntos, e aplica o procedimento `OnTriangle` a cada uma dessas partes. Este procedimento é usado, principalmente, no método multi-escala, no cálculo do produto escalar $\langle f, g\rangle$ onde f e g são splines definidos sobre duas triangulações diferentes da esfera.

O módulo auxiliar `SPOverLapTable` define uma tabela que descreve a interseção de triângulos de duas triangulações. Esta tabela é usada para acelerar o cálculo do produto escalar $\langle f, g\rangle$ entre duas triangulações diferentes, e também pode ser usada para localizar rapidamente um ponto numa triangulação, dada sua posição numa triangulação mais grosseira (algoritmo de localização de Kirkpatrick [12]).

11.6 Construção das bases

No módulo `SPOrthoBasis`, implementamos procedimentos para construção das bases para os espaços de splines homogêneos e não homogêneos, como descrito no capítulo 6. No caso de splines homogêneos, para continuidade 0 implementamos a construção das bases de graus 2 e 3; para continuidade 1 implementamos a construção das bases de graus 5 e 6. Os procedimentos permitem também construir uma base com elementos semi-ortogonalizados (seção 6.6).

O módulo contém também um procedimento que concatena duas bases de splines homogêneos, para os espaços $\mathcal{H}_k^{d-1}[T]/\mathbf{S}^2$ e $\mathcal{H}_k^{d}[T]/\mathbf{S}^2$, produzindo dessa maneira uma base para o espaço $\mathcal{P}_k^{d}[T]/\mathbf{S}^2$. Nesse processo, cada elemento é convertido do tipo `SPBezierPWFunction.T` para o tipo `SPBezPolyPWFunction.T`.

11.7 Representação gráfica das funções

No módulo `SPPlot` estão implementados os procedimentos para desenhar funções esféricas. O módulo fornece procedimentos separados para desenhar a esfera, uma triangulação sobre a mesma, uma função esférica tipo `SPFunction.T` (por meio de tonalização e curvas de nível).

O módulo `SPRange` calcula o valor máximo de uma função na esfera. É geralmente usado antes de `SPPlot`, para determinar o passo apropriado das curvas de nível.

11.8 Programas

Além dos módulos de biblioteca descritos acima, implementamos vários programas executáveis, relacionados a seguir.

11.8.1 Construção de triangulações

O programa `SPMakeTriang` cria uma triangulação regular, ou uma triangulação de Delaunay para pontos aleatoriamente distribuídos na esfera.

O programa `SPFixTriang` corrige a geometria de uma triangulação na esfera, impedindo que a triangulação contenha arestas coplanares incidentes em um mesmo vértice. O programa `SPOptTriang` é uma versão mais eleborada de `SPFixTriang`, que otimiza a geometria da triangulação segundo os critérios descritos na seção 4.5.

O programa `SPReduceTriang` simplifica uma triangulação, segundo o algoritmo de Kirkpatrick (seção 9.4). O programa também constrói uma tabela (`SPOverlapTable.T`) que descreve a interseção dos triângulos das duas triangulações.

11.8.2 Construção das matrizes

O programa `SPAddMatrix` calcula a matriz $C = \alpha \cdot A + \beta \cdot B$, onde α e β são reais e A e B são matrizes dadas. Estas matrizes são lidas e gravadas em formato padronizado (seção 11.3).

O programa `SPGGMatrix` calcula as matrizes G e S com elementos $G_{ij} = \langle \phi_i, \phi_j \rangle$ e $S_{ij} = \langle \nabla \phi_i, \nabla \phi_j \rangle$ onde $\phi_1, \ldots, \phi_n$ é uma base dada.

O programa `SPFGMatrix` calcula a matriz F com elementos $F_{ij} = \langle \phi_i, \psi_j \rangle$, onde $\phi_1, \ldots, \phi_n$ e $\psi_1, \ldots, \psi_m$ são bases dadas.

O programa `SPCholeski` calcula a matriz triangular inferior L da decomposição de Choleski $LL^\top$ de uma matriz simétrica dada.

11.8.3 Construção das bases

O programas `SPFindBasis` e `SPNHFindBasis` encontram uma base para os espaços dos splines esféricos homogêneos e não homogêneos, respectivamente, para uma dada triangulação (vide módulo `SPOrthoBasis` na seção 11.6).

O programa `SPExtractBasis` extrai um elemento de uma base, e grava-o em um arquivo separado.

11.8.4 Aproximação de funções

O programa `SPApprox` calcula uma aproximação por splines esféricos de uma função definida na esfera, pelo método dos mínimos quadrados. O programa tem opções de desenhar a função aproximante e a função erro.

O programa `SPApproxPWFunction` é uma variante desse programa que calcula a aproximação de um spline esférico por um outro spline esférico, possivelmente com triangulação diferente.

11.8.5 Solução de equações diferenciais parciais

O programa `SPUniSolve` resolve numericamente a equação de Helmholtz na esfera

$$- \Delta u(p) + c \cdot u(p) = \mathcal{F}(u(p), p) \tag{11.1}$$

onde a função arbitrária $\mathcal{F}$ e a constante c são especificadas pelo usuário.

O programa procura uma solução aproximada no espaço de splines esféricos, gerado por uma base $\phi_1, \phi_2, \ldots, \phi_n$ dada. Conforme discutido no capítulo 8, a solução dessa equação pode ser encontrada resolvendo-se um sistema linear $Ha = b(a)$. A matriz H deve ser calculada previamente pelo usuário, usando os programas da seção 11.8.2.

O programa `SPMultiSolve` resolve a equação de Helmholtz iterativamente em múltiplas escalas, como descrito no capítulo 9. Este programa exige a matriz H do sistema, como no programa `SPUniSolve`, para cada escala de resolução. Além disso, o programa exige as matrizes F e G para transferência de soluções de uma malha para outra (vide procedimento `SPChangeCoordBasis` na seção 11.1.3).

O módulo `SPNLFunction` tem um procedimento que recebe o nome de uma função $\mathcal{F}$, previamente definida, e retorna um procedimento que calcula o lado direito da equação (11.1).

11.8.6 Programas auxiliares

Os programas `SPDrawTriang`, `SPPWDraw`, `SPPWDrawGrad` e `SPPolyDraw` desenham a triangulação, as curvas de nível de uma função esférica por partes, o campo gradiente de uma função esférica por partes e as curvas de nível de um polinômio esférico por partes, respectivamente, usando os módulos `SPPlot` e `SPRange` (seção 11.7).

Bibliografia

[1] Alfeld, P., Neamtu, M. and Schumaker, L.L. Bernstein-Bézier polynomials on circle, spheres, and sphere-like surfaces. *Computer Aided Geometric Design Journal*, 13:333–349, 1996.

[2] Alfeld, P., Neamtu, M. and Schumaker, L.L. Dimension and local bases of homogeneous spline spaces. *SIAM Journal of Mathematical Analysis*, 27(5):1482–1501, September 1996.

[3] Alfeld, P and Schumaker, L.L. Non-existence of star-supported spline bases. manuscript

[4] Alfeld, P., Neamtu, M., and Schumaker L.L.. Fitting scattered data on sphere-like surfaces using spherical splines. *Journal of Computational and Applied Mathematics*, 73:5–43, 1996.

[5] Fulton, W. *Algebraic Curves: An Introduction to Algebraic Geometry.* W. A. Benjamin, 1969.

[6] Spivak, M. *Calculus on Manifolds - A Modern Approach to Classical Theorems of Advanced Calculus.* W. A. Benjamin, 1965.

[7] Kunz, E. *Introduction to Commutative Algebra and Algebraic Geometry.* Birkhauser, 1993.

[8] de Boor, C. *B-form basics*, in *G. E. Farin (ed.)* Geometric Modeling: Algorithms and New Trends. SIAM 1987, 131-148.

[9] Guibas L. and Stolfi, J. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, Vol.4 No. 2:74-123, 1985.

[10] Green, P. J. e Sibson, R. Computing Dirichlet tesselation in plane. *Comput. J.*, Vol.21 No. 2:168-173, 1977.

[11] Resende, P. J. de e Stolfi, J. *Fundamentos de Geometria Computacional.* IX Escola de Computação, 1994.

[12] O'Rourke, J. *Computational Geometry in C.* Cambridge University Press, 1994.

[13] Briggs, W. L. *A Multigrid Tutorial.* SIAM Publications, 1987.

[14] Trefethen, L. N. and Bau III, D. *Numerical Linear Algebra.* SIAM Publications, 1997.

[15] Müller, C. *Spherical Harmonics.* Lecture Notes in Mathematics, no.17 , Springer, Berlin, 1966.

[16] Courant R, and Hilbert D. *Methods of Mathematical Physics.* Interscience vol. I, New York, 1953.

[17] Burden, R. L. and Faires, J. D. *Numerical Analysis.* PWS-Kent, Boston, 1988.

[18] Schultz Martin H. *Spline Analysis.* Prentice-Hall, New Jersey, 1973.

[19] Hämmerlin, G. and Hoffmann, K.-H. *Numerical Mathematics.* Springer, 1991.

[20] Kaderstuncer H. and Norrier D. H. *Finite Element Handbook.* McGraw-Hill, 1987.

[21] Vichnevetsky R. *Computer Methods For Partial Differential Equations*, Volume I. Prentice-Hall, 1981.

[22] Cowper G, R. *Gaussian quadrature formulas for triangles.* Int. J. Numer. Meth. Eng, Vol. 1, No. 3, pp. 405-408.

[23] Barros, S.R.M. and Kauranne, T. *Spectral and multigrid spherical Helmholtz equation solvers on distributed memory parallel computers.* Proceedings ECMWF pp. 26-30, 1990.

[24] Jarraud, M. and Baede, A. P. M. *The use of spectral techniques in numerical weather prediction.* Lectures in Applied Mathematics, Vol 22, pp. 1-41, 1985.

[25] Nelson, G. *Systems Programming with Modula-3.* Prentice-Hall, 1991.

[26] Harbison, S. P. *Modula-3.* Prentice-Hall.

[27] Press, W.H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T. *Numerical Recipes.* Cambridge University Press, 1986.

[28] Furtado, A.L. *Teoria dos Grafos Algoritmos.* Livros Técnicos e Científicos, 1973.