

# Métodos espectrais de alta ordem na resolução de equações diferenciais

MONOGRAFIA APRESENTADA  
AO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
DA  
UNIVERSIDADE DE SÃO PAULO  
PARA  
OBTENÇÃO DO TÍTULO  
DE  
BACHAREL EM MATEMÁTICA APLICADA E COMPUTACIONAL

Augusto Carillo Ferrari

Orientador: Prof. Dr. Nelson Mukgayar Kuhl

Coorientador: Dr. Paulo José Saiz Jabardo

São Paulo, dezembro de 2015



# Agradecimientos

A elaborar



# Sumário

<b>Lista de Figuras</b>	<b>vii</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 História . . . . .	1
1.2 Julia . . . . .	1
<b>2 Método espectral e Método dos elementos finitos</b>	<b>3</b>
2.1 sobre . . . . .	3
2.2 Interpolação . . . . .	3
2.2.1 Interpolação polinomial . . . . .	3
2.2.2 Fenômeno de Runge . . . . .	5
2.2.3 Teorema I: Erro de interpolação de Cauchy . . . . .	5
2.2.4 Teorema II: Amplitude mínima . . . . .	6
2.2.5 Integração gaussiana e linhas pseudoespectral . . . . .	7
2.2.6 Teorema: Integração Gauss-Jacobi . . . . .	7
2.2.7 tipos de polinômio . . . . .	7
2.2.8 Derivada . . . . .	8
2.3 método dos elementos finitos . . . . .	8
<b>3 Método dos elementos espectrais</b>	<b>9</b>
3.1 Equação de Helmholtz . . . . .	9
3.2 Condição de contorno Neumann de newton e Dirichlet . . . . .	9
3.3 Discretização . . . . .	9
3.4 Solver linear . . . . .	9
<b>4 Simulações</b>	<b>11</b>
<b>5 Conclusões</b>	<b>13</b>
<b>Referências Bibliográficas</b>	<b>15</b>



# Lista de Figuras

2.1	interpolação simples . . . . .	3
2.2	polinômio base de Lagrange para 6 pontos . . . . .	4
2.3	interpolação com n pontos equidistantes . . . . .	5
2.4	fenômeno Runge . . . . .	5
2.5	método de raízes de chebychev contra raízes de pontos equidistante . . . . .	7





# Lista de Tabelas



# Capítulo 1

## Introdução

### 1.1 História

O método espectral surgiu como uma ferramenta de alto poder computacional em mecânica de fluídos, proposto em **1994 por Blinova O ANO ESTÁ CERTO???**, implementado em 1954 por Sylberman, praticamente abandonado no meio da década de 60 e ressurgindo em 1969-1970 por Orszag e Eliason, Manchenhauer e Rasmussen, foi desenvolvido para aplicações especializadas. No entanto, **somente em 1977 foi formalizado matematicamente por Gottlieb e Orszag em 1980 DECIDA-SE QUAL O ANO. Sugestão PJ: usa o bibtex e coloca as referências. É o correto e não há ambiguidade**

Originalmente o método espectral foi promovido por meteorologistas no estudo de modelos globais de tempo e especialistas em dinâmica de fluídos estudando turbulências isotrópicas. Desde a década de 80 até hoje o estudo na área de CFD (Computational Fluid Dynamics- dinâmica dos fluidos computacional) tem crescido lado a lado ao avanço computacional que o tornou possível.

### 1.2 Julia

Para a implementação do método usaremos como ferramenta de estudo a linguagem de alto nível, *Julia*, que por ser dinâmica e de excelente desempenho computacional será essencial para execução dos cálculos. **Será essencial??? Até hoje não se fazia nada? Porque você acha julia interessante?. Leia o que está na pag. da julia.**

Apesar de nova, a linguagem criada no MIT vem sendo rapidamente acolhida pela comunidade científica e assim, com seu código open-source, ele é diariamente atualizado e possui um número de bibliotecas em crescente ascensão.



## Capítulo 2

# Método espectral e Método dos elementos finitos

### 2.1 sobre

Método espectral é um método poderoso usado para solução de equações diferencial parcial. Diferentemente do método das diferenças finitas, que considera apenas os pontos próximos do ponto que queremos computar chamada de método *local*, o método espectral considera todo o domínio, sendo assim um método *global*. Essa técnica tem mais precisão pois converge exponencialmente diferente do método local. É preferível a utilização desse método quando a solução varia em função do *tempo* e *espaço*.

### 2.2 Interpolação

A interpolação de uma função  $f(x)$  por um polinômio trigonométrico ou não, de grau  $n$ ,  $P_n(x)$  e que satisfaça:

$$P_n(x_i) = f(x_i) \quad i = 1, 2, \dots, n+1 \quad (2.1)$$

Onde  $f(x_i)$  é a função  $f$  pré-calculada nos pontos  $x_i$ . A escolha desses pontos  $x_i$  ainda será explicada.

#### 2.2.1 Interpolação polinomial

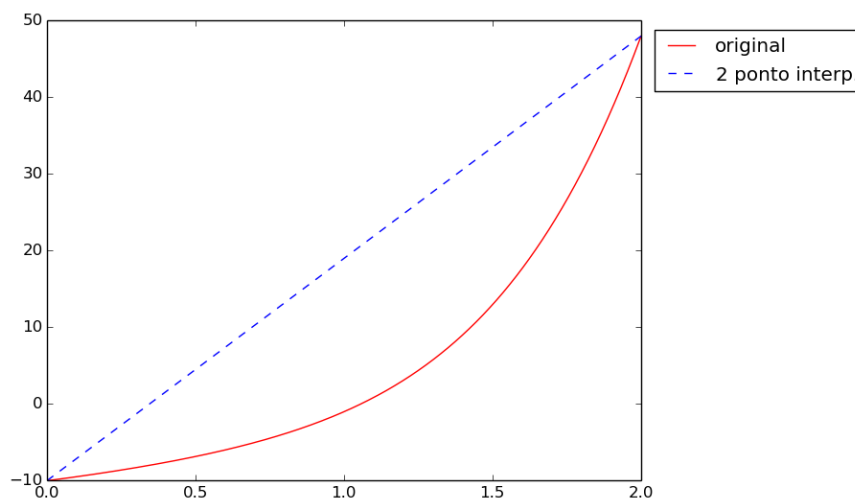


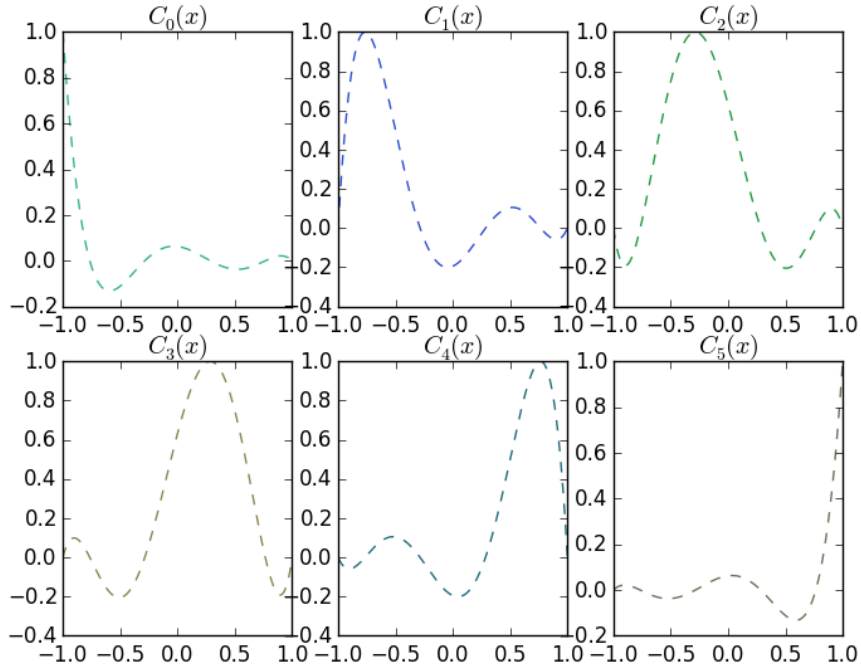
Figura 2.1: *interpolação simples*

Antes do uso de calculadoras e computadores, um método de estimar o valor de uma  $f$  num ponto, eram utilizados tabelas com valores de pré-calculados. A maneira mais simples de entender é a estimação do valor da função em um ponto intermediário entre dois pontos conhecidos é o uso da interpolação *Linear*.

$$f(x) \approx \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) \quad (2.2)$$

Para fazermos essa interpolação para  $n$  pontos conhecidos aproximamos uma função usando o polinômio base de *Lagrange*.

$$C_i(x) = \prod_{j=0, j \neq i}^N \frac{x - x_j}{x_i - x_j} \quad (2.3)$$

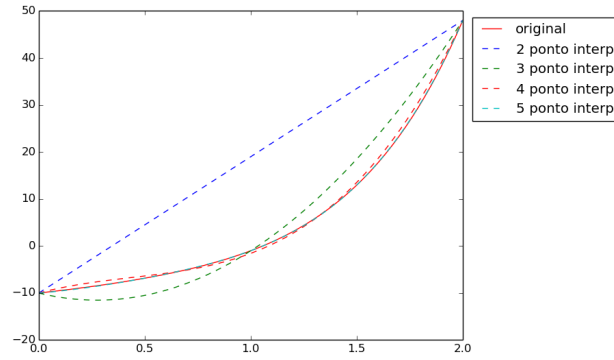


**Figura 2.2:** polinômio base de *Lagrange* para 6 pontos

A interpolação de *Lagrange* é dada por :

$$P_n(x) \equiv \sum_{i=0}^N f(x_i) C_i(x) \quad (2.4)$$

Obedecendo que  $P_n(x_i) = f(x_i)$ . Apesar dos pontos interpoladores equidistante serem comumente utilizados, não há restrições, podendo até mesmo estar fora de ordem.



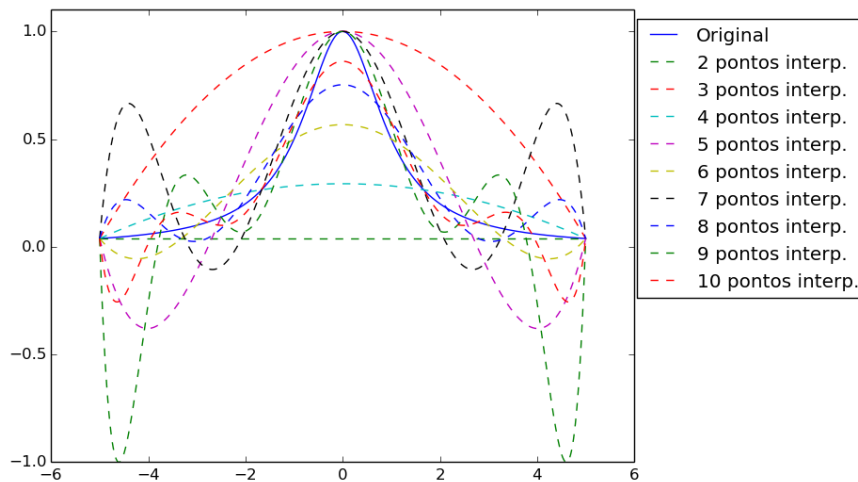
**Figura 2.3:** *interpolação com  $n$  pontos equidistantes*

### 2.2.2 Fenômeno de Runge

Apesar de parecer que uma boa interpolação tenha uma boa aproximação usando pontos igualmente distantes sobre um intervalo  $[a, b]$ ,  $\lim_{n \rightarrow \infty} |f(x) - P_n(x)| = 0$  para qualquer  $f(x)$  diferenciável. No início do século XX, *Carl David Tolmé Runge*, provou que para uma função  $f(x)$ :

$$f(x) = \frac{1}{1+x^2}, x \in [-5, 5] \quad (2.5)$$

que para pontos equidistantes, a interpolação converge apenas no intervalo  $[-3.63, 3.63]$ , e diverge fora do mesmo. Para polinômios de maior grau, esse intervalo de convergência tende a diminuir e perto dos pontos de fronteira diverge bastante (figura abaixo).



**Figura 2.4:** *fenômeno Runge*

Assim, Runge prova que no meio do intervalo temos boas aproximações mas infelizmente perto dos extremos, os valores interpolados oscilam muito, para um polinômio de grau  $n$  com pontos equidistantes. Esse fenômeno sugere que escolhamos pontos diversos que tenham menor concentração do meio do intervalo, onde temos uma maior precisão, e aumentar a densidade de pontos próximos dos extremos. Agora como encontrar uma distribuição de pontos de forma que melhore a interpolação? A resposta pode ser explicada pelos teoremas a seguir.

### 2.2.3 Teorema I: Erro de interpolação de Cauchy

Dado  $f(x)$  com pelo menos  $N + 1$  derivadas no intervalo de interesse e seja  $P_N(x)$  seja o interpolador Lagrangiano de grau  $N$ . Então o erro é dado por:

$$f(x) - P_N(x) = \frac{1}{[N+1]!} f^{(N+1)}(\epsilon) \prod_{i=0}^N (x - x_i) \quad (2.6)$$

Para um  $\epsilon(x) \in [-1, 1]$ .

Logo, para minimizar o erro, não podemos fazer nada quanto o termo  $f^{(N+1)}(\epsilon)$ , pois necessita conhecer a função interpolada. Quanto o polinômio  $\prod_{i=0}^N (x - x_i)$ , sabemos que o coeficiente do termo  $x^N$  é 1, independente da escolha de pontos. Então a pergunta que fica é, qual escolha de pontos nos dá um *polinômio* com coeficiente líder igual a 1 minimiza essa função? Felizmente e coincidentemente, essa resposta foi respondida quase meio século antes do próprio fenômeno de Runge ser descoberto. Veremos no próximo teorema.

#### 2.2.4 Teorema II: Amplitude mínima

De todos os polinômios de grau  $N$  com coeficiente de  $x^N$  igual a 1, o único polinômio que tem o menor máximo no intervalo  $[-1, 1]$  é  $\frac{T_N(x)}{2^{(N-1)}}$ , o polinômio de *Chebyshev* dividido por ideia. Em outras palavras, todos os polinômios de mesmo grau e coeficiente líder unitário, satisfazem a desigualdade:

$$\max_{x \in [-1, 1]} |P_N(x)| \geq \max_{x \in [-1, 1]} \left| \frac{T_N(x)}{2^{(N-1)}} \right| = \frac{1}{2^{N-1}} \quad (2.7)$$

$$T_0(x) = 1 \quad (2.8)$$

$$T_1(x) = x \quad (2.9)$$

$$T_{N+1}(x) = 2xT_N(x) - T_{N-1}(x) \quad (2.10)$$

$$T_N(x) = \cos(n \arccos x) = \cosh(n \operatorname{arccosh} x) \quad (2.11)$$

Agora, qualquer polinômio de grau  $N$  pode ser fatorado na forma de um produtório  $(x - x_i)$ , onde  $x_i$  é uma das raízes do polinômio, em particular:

$$\frac{1}{2^N} T_{N+1}(x) \equiv \prod_{i=1}^{N+1} (x - x_i) \quad (2.12)$$

Temos então que, para minimizar o erro no *Teorema I*, o polinômio deve ser proporcional a  $T_{N+1}(x)$ . Isso implica que os pontos interpoladores que minimizam o erro, são as raízes do próprio polinômio de *Chebyshev* de grau  $N + 1$ .

Usando o fato que esse polinômio pode ser reescrita como uma função trigonométrica, temos que as raízes são:

$$x_i \equiv \cos \left[ \frac{(2i-1)\pi}{2(N+1)} \right], i = 1, 2, \dots, N+1 \quad (2.13)$$

Pela expansão de *Taylor* da função coseno, verificamos a afirmação de acima que o espaçamento da malha é  $O(N^2)$  perto das fronteiras:

$$x_1 \approx -1 + \frac{\pi^2}{8N^2}; \quad x_2 \approx -1 + \frac{9\pi}{8N^2} [N \gg 1] \quad (2.14)$$

Agora temos como conseguir o polinômio de *Chebyshev* que minimiza o erro. Porém essa escolha de polinômio varia para diferentes geometrias ou funções harmônicas ou hermitianas.



### 2.2.5 Integração gaussiana e linhas pseudoespectral

A razão pelo qual o método de colocação é também chamado "pseudoespectral" devido a colocação otimizada dos pontos de interpolação, como o polinômio de *Chebyshev*, faz o método da colocação idêntico ao método de *Galerkin* se o produto interno é calculado por um tipo de integração numérica conhecida como Integração gaussiana. Integração numérica e interpolação Lagrangiana são muito correlacionada pois o método de integração é de aproximar o polinômio pelo integrando de  $f(x)$  e depois integrar  $P_N(X)$ . Como o interpolado pode ser integrado facilmente, o erro é dado como a diferença de  $f(x)$  e  $P_N(x)$ . A fórmula é dada por:

$$\int_a^b f(x) dx \approx \sum_{i=0}^N w_i f(x) \quad (2.15)$$

e a função peso  $w_i$  é dado por:

$$w_i \equiv \int_a^b C_i(x) dx \quad (2.16)$$

Onde  $C_i$  é o polinômio base de *Lagrange*. Quando calculamos, os pontos de *interpolação* são nomeados de *abcissas*, enquanto que a *integral* de um polinômio base é chamado de *quadratura*, no contexto da integração numérica.

A *ordem* de um método numérico é diretamente relacionado a maior polinômio que o aproxima. Por exemplo quando aplicamos a interpolação para equações de primeiro e segundo grau, os erros são respectivamente  $O(h^2)$ ,  $O(h^4)$ . Similarmente, a fórmula de quadratura com  $(N + 1)$  pontos, será exata se o integrando for um polinômio de grau  $N$ .

Gauss fez uma observação que pontos de interpolação equidistantes não são tão bons. Se tivermos pontos interpoladores  $x_i$  e seus pesos  $w_i$  desconhecido, teremos o dobro de parâmetros para escolher que maximizem a precisão do método e assim podemos ter uma aproximação exata para polinômios de grau  $(2N + 1)$ .

### 2.2.6 Teorema: Integração Gauss-Jacobi

Seja  $(N + 1)$  pontos interpoladores  $x_i$  escolhidos e raízes de  $P_N(x)$ , um polinômio de grau  $(N + 1)$  do tipo *ortogonal* no intervalo  $[a, b]$  com respeito a função  $p(x)$ , então a fórmula de quadratura é dado por:

$$\int_b^a f(x) p(x) dx = \sum_{i=0}^N w_i f_i(x) \quad (2.17)$$

é exata para toda  $f(x)$  polinomial de grau máximo  $(2N + 1)$

### 2.2.7 tipos de polinômio

adicionar outros polinômios de base talvez ? Gauss-Jacobi Gauss-Lobatto-Jacobi Gauss-Radau-Jacobi

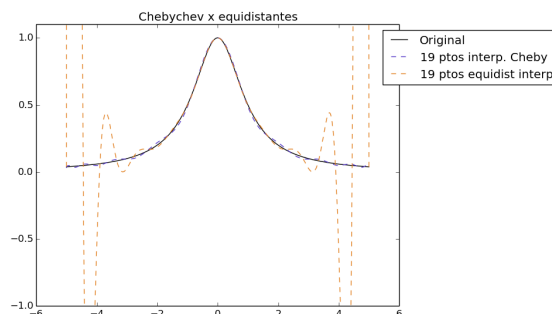


Figura 2.5: método de raízes de chebychev contra raízes de pontos equidistante

### 2.2.8 Derivada

As derivadas podem ser calculadas da seguinte forma:

$$f(x) = \sum_{i=0}^{N-1} f(x_i) C_i(x) \quad (2.18)$$

$$\frac{\partial f(x)}{\partial x} = \sum_{i=0}^N f(x_i) \frac{\partial C_i(x)}{\partial x} \quad (2.19)$$

para calcular as integrais, é necessário conhecer a derivada nos nós da quadratura,  $x_i$  :

## 2.3 método dos elementos finitos

adicionar texto de elementos dinâmicos

## Capítulo 3

## Método dos elementos espectrais

### 3.1 Equação de Helmholtz

[illegible]

### 3.2 Condição de contorno Neumann de newton e Dirichilet

[illegible]

### 3.3 Discretização

[illegible]

### 3.4 Solver linear

[illegible]



## Capítulo 4

# Simulações



## Capítulo 5

# Conclusões

test test test

[Ard14]





# Referências Bibliográficas

[Ard14] Luca Ardito. *Energy-aware Software*. Tese de Doutorado, Politecnico di Torino, 2014. [13](#)