Aproximando Equações Diferencial Ordinárias pelo Método de Galerkin

João Victor Lopez Pereira

8 de setembro de 2024

Rio de Janeiro - RJ

Resumo

Este documento apresenta a resolução de um sistema genérico de equações diferenciais ordinárias utilizando o Método de Galerkin, conforme abordado nas aulas da disciplina *Introdução ao Método dos Elementos Finitos*, ministradas pelos professores Dr. Marcello Goulart Teixeira e Dr. Bruno Alves do Carmo na Universidade Federal do Rio de Janeiro, durante o segundo semestre de 2024.

This document presents the solution of a generic system of ordinary differential equations using the Galerkin Method, as covered in the course *Introduction to the Finite Element Method*, taught by professors Dr. Marcello Goulart Teixeira and Dr. Bruno Alves do Carmo at the Federal University of Rio de Janeiro, during the second half of 2024.

Sumário

Capítulo 1

Resolvendo uma EDO pelo Método de Galerkin

1.1 Formulação Forte

Dado uma função $f(x)$ e constantes α, β e γ, queremos encontrar a função $u(x)$ tal que:

$$\begin{cases} -\alpha u_{xx}(x) + \beta u(x) + \gamma u_x(x) = f(x) \\ u(0) = u(1) = 0 \end{cases}$$

1.2 Formulação Fraca

Visto que:

$$-\alpha u_{xx}(x) + \beta u(x) + \gamma u_x(x) = f(x)$$

Podemos multiplicar ambos os lados por uma função

$$v(x), \text{ tal que } v(1) = v(0) = 0$$

que nos ajude a eliminar a segunda derivada em u_{xx}:

$$f(x) = -\alpha u_{xx}(x) + \beta u(x) + \gamma u_x(x)$$

$$f(x)v(x) = \left[-\alpha u_{xx}(x) + \beta u(x) + \gamma u_x(x)\right] v(x)$$

$$f(x)v(x) = -\alpha u_{xx}(x)v(x) + \beta u(x)v(x) + \gamma u_x(x)v(x)$$

$$\int_0^1 f(x)v(x)dx = \int_0^1 \left[-\alpha u_{xx}(x)v(x) + \beta u(x)v(x) + \gamma u_x(x)v(x)\right] dx$$

$$\int_0^1 f(x)v(x)dx = \int_0^1 -\alpha u_{xx}(x)v(x)dx + \int_0^1 \beta u(x)v(x)dx + \int_0^1 \gamma u_x(x)v(x)dx$$

Sabemos, por cálculo, que:

$$\int f(x)g'(x)dx = f(x)g(x) - \int f'(x)g(x)dx$$

Logo, realizando a integração por partes no primeiro termo na equação:

$$\int_0^1 f(x)v(x)dx = -\alpha \left[u_x(x)v(x)\Big|_0^1 - \int_0^1 u_x(x)v_x(x)dx \right] + \int_0^1 \beta u(x)v(x)dx + \int_0^1 \gamma u_x(x)v(x)dx$$

$$\int_0^1 f(x)v(x)dx = -\alpha \left[(u_x(1)v(1) - u_x(0)v(0)) - \int_0^1 u_x(x)v_x(x)dx \right] + \int_0^1 \beta u(x)v(x)dx + \int_0^1 \gamma u_x(x)v(x)dx$$

$$\int_0^1 f(x)v(x)dx = -\alpha \left[\cancel{(u_x(1)v(1) - u_x(0)v(0))} - \int_0^1 u_x(x)v_x(x)dx \right] + \int_0^1 \beta u(x)v(x)dx + \int_0^1 \gamma u_x(x)v(x)dx$$

$$\int_0^1 f(x)v(x)dx = \alpha \int_0^1 u_x(x)v_x(x)dx + \beta \int_0^1 u(x)v(x)dx + \gamma \int_0^1 u_x(x)v(x)dx$$

Seja H um espaço tal que $\forall u \in H$ suficientemente suave, tal que $u(0) = u(1) = 0$, u é solução do sistema. Seja V um espaço finito tal que $\forall v \in V$ suficientemente suave, tal que $v(0) = v(1) = 0$ é espaço das funções de teste. Nesse caso em específico, $H = V$.

Precisamos determinar $u \in H$ tal que:

$$\alpha \int_0^1 u_x(x)v_x(x)dx + \beta \int_0^1 u(x)v(x)dx + \gamma \int_0^1 u_x(x)v(x)dx = \int_0^1 f(x)v(x)dx$$

$$\forall v \in V$$

1.3 Problema Aproximado pelo Método de Galerkin

O método de Galerkin consiste em aproximar o espaço das soluções por um subespaço de dimensão finita.[2] Sendo assim:

Seja H^h um espaço finito tal que $\forall u^h \in H^h$ suficientemente suave, tal que $u(0) = u(1) = 0$, u_h é solução do sistema. Seja V^h um espaço finito tal que $\forall v^h \in V^h$ suficientemente suave, tal que $v(0) = v(1) = 0$ é espaço das funções de teste. Nesse caso em específico, $H^h = V^h$.

Precisamos determinar $u^h \in H^h$ tal que:

$$\forall v^h \in V^h$$

$$\int_0^1 f(x)v^h(x)dx = \alpha \int_0^1 u_x^h(x)v_x^h(x)dx + \beta \int_0^1 u^h(x)v^h(x)dx + \gamma \int_0^1 u_x^h(x)v^h(x)dx$$

Seja

$$u^h(x) \approx \sum_{j=1}^{m} \varphi_j(x)c_j$$

Logo, a sua derivada será

$$u_x^h(x) \approx \sum_{j=1}^{m} \frac{d(\varphi_j(x)c_j)}{dx} = \sum_{j=1}^{m} \frac{d\varphi_j(x)}{dx}c_j + \frac{dc_j}{dx}\varphi_j(x) = \sum_{j=1}^{m} \frac{d\varphi_j(x)}{dx}c_j + \cancel{\frac{dc_j}{dx}\varphi_j(x)} = \sum_{j=1}^{m} \frac{d\varphi_j(x)}{dx}c_j$$

Veja que $\frac{dc_j}{dx}$ é cancelado pois c_j é uma constante.

Substituindo em nossa equação:

$$\int_0^1 f(x)v^h(x)dx = \alpha \int_0^1 \sum_{j=1}^{m} \frac{d\varphi_j(x)}{dx}c_j v_x^h(x)dx + \beta \int_0^1 \sum_{j=1}^{m} \varphi_j(x)c_j v^h(x)dx + \gamma \int_0^1 \sum_{j=1}^{m} \frac{d\varphi_j(x)}{dx}c_j v^h(x)dx$$

$$\int_0^1 f(x)v^h(x)dx = \sum_{j=1}^{m} \left[\alpha \int_0^1 \frac{d\varphi_j(x)}{dx}c_j v_x^h(x)dx + \beta \int_0^1 \varphi_j(x)c_j v^h(x)dx + \gamma \int_0^1 \frac{d\varphi_j(x)}{dx}c_j v^h(x)dx\right]$$

$\forall v^h(x) = \varphi_i(x)$, $i \in [1, m]$, obtemos um sistema linear tal que $\mathbb{K}\mathbb{C} = \mathbb{F}$ tal que $\mathbb{C} \in \mathbb{R}^m$.

1.4 Matriz e Vetor Locais

Para $i, j \in [1, m]$ tais que i e j correspondam aos índices da matriz:

$$
\begin{aligned}
K_{i,j} &= \alpha \int_0^1 \frac{d\varphi_i(x)}{dx}\frac{d\varphi_j(x)}{dx}dx + \beta \int_0^1 \varphi_i(x)\varphi_j(x)dx + \gamma \int_0^1 \varphi_i(x)\frac{d\varphi_j(x)}{dx}dx \\
F_i &= \int_0^1 f(x)\varphi_i(x)dx
\end{aligned}
$$

Definiremos $\varphi_i(x)$ como:

$$
\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h}, & \forall x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{h}, & \forall x \in [x_i, x_{i+1}] \\ 0, & \forall x \notin [x_{i-1}, x_{i+1}] \end{cases}
$$

Observe que a função base $\varphi_i(x)$ é escolhida intencionalmente para facilitar os cálculos. Em particular, utilizamos funções lineares por partes, pois elas são suficientemente simples para nos permitir calcular as integrais de maneira eficiente e precisa, especialmente com métodos numéricos como a quadratura gaussiana.

Além disso, $\varphi_i(x)$ é projetada para ser zero fora do intervalo $[x_{i-1}, x_{i+1}]$, o que significa que cada função base afeta apenas dois ou três pontos consecutivos. Essa escolha reduz significativamente a complexidade do sistema linear, já que a matriz resultante terá muitas entradas nulas. Isso nos ajuda a trabalhar com matrizes esparsas, que são computacionalmente mais eficientes, tanto em termos de armazenamento quanto de tempo de processamento.

1.4.1 Cálculo da Matriz Local por Quadratura Gaussiana

$$
K = \sum_{e=1}^{m+1} K^e
$$

Observe que, pela definição de $\varphi_i(x)$, $K^e_{i,j} \neq 0$ quando $i \in [x_{i-1}, x_{i+1}]$ e $j \in [x_{i-1}, x_{i+1}]$. Logo, podemos limitar nossa integral para esse intervalo visto que restante do intervalo resultará sempre em zero. Faremos isso utilizando a notação local para as funções da base.

$$
\begin{aligned}
K^e_{i,j} &= \alpha \int_{x^{e-1}}^{x^e} \frac{d\varphi_i(x)}{dx}\frac{d\varphi_j(x)}{dx}dx + \beta \int_{x^{e-1}}^{x^e} \varphi_i(x)\varphi_j(x)dx + \gamma \int_{x^{e-1}}^{x^e} \varphi_i(x)\frac{d\varphi_j(x)}{dx}dx \\
K^e_{a,b} &= \alpha \int_{x_1^{e-1}}^{x_2^e} \frac{d\varphi^e_a(x)}{dx}\frac{d\varphi^e_b(x)}{dx}dx + \beta \int_{x_1^{e-1}}^{x_2^e} \varphi^e_a(x)\varphi^e_b(x)dx + \gamma \int_{x_1^{e-1}}^{x_2^e} \varphi^e_a(x)\frac{d\varphi^e_b(x)}{dx}dx
\end{aligned}
$$

Para podermos utilizar a Quadratura Gaussiana, precisaremos modificar nosso intervalo de $[0, 1]$ para $[-1, 1]$ visto que essa é a condição para o uso dessa técnica.

Seja $x(\xi, e)$ a função que converte de ξ para x tal que

$$x(\xi, e) = \frac{h}{2}(\xi + 1) + x_{e-1}$$

$$\text{Seja} \quad \phi_n(\xi) = \begin{cases} \dfrac{1-\xi}{2} & \text{, se } n == 1 \\ \dfrac{1+\xi}{2} & \text{, se } n == 2 \end{cases} \quad \text{e seja} \quad \frac{d\phi_n}{d\xi}(\xi) = \begin{cases} \dfrac{-1}{2} & \text{, se } n == 1 \\ \dfrac{1}{2} & \text{, se } n == 2 \end{cases}$$

tal que:

$$\phi_n(\xi) = \varphi_n^e(x(\xi, e))$$

$$K_{a,b}^e = \alpha \int_{-1}^{1} \frac{d\varphi_a^e(x(\xi,e))}{dx} \frac{d\varphi_b^e(x(\xi,e))}{dx} \frac{h}{2} d\xi + \beta \int_{-1}^{1} \varphi_a^e(x(\xi,e)) \varphi_b^e(x(\xi,e)) \frac{h}{2} d\xi +$$
$$+ \gamma \int_{-1}^{1} \varphi_a^e(x(\xi,e)) \frac{d\varphi_b^e(x(\xi,e))}{dx} \frac{h}{2} d\xi$$

$$K_{a,b}^e = \alpha \int_{-1}^{1} \frac{d\phi_a}{d\xi}(\xi) \frac{d\phi_b}{d\xi}(\xi) \frac{h}{2} \frac{2}{h} \frac{2}{h} d\xi + \beta \int_{-1}^{1} \phi_a(\xi) \phi_b(\xi) \frac{h}{2} d\xi + \gamma \int_{-1}^{1} \phi_a(\xi) \frac{d\phi_b}{d\xi}(\xi) \frac{h}{2} \frac{2}{h} d\xi$$

$$K_{a,b}^e = \alpha \int_{-1}^{1} \frac{d\phi_a}{d\xi}(\xi) \frac{d\phi_b}{d\xi}(\xi) \frac{h}{\cancel{2}} \frac{\cancel{2}}{h} \frac{2}{h} d\xi + \beta \int_{-1}^{1} \phi_a(\xi) \phi_b(\xi) \frac{h}{2} d\xi + \gamma \int_{-1}^{1} \phi_a(\xi) \frac{d\phi_b}{d\xi}(\xi) \frac{h}{\cancel{2}} \frac{\cancel{2}}{h} d\xi$$

$$K_{a,b}^e = \alpha \int_{-1}^{1} \frac{d\phi_a}{d\xi}(\xi) \frac{d\phi_b}{d\xi}(\xi) \frac{2}{h} d\xi + \beta \int_{-1}^{1} \phi_a(\xi) \phi_b(\xi) \frac{h}{2} d\xi + \gamma \int_{-1}^{1} \phi_a(\xi) \frac{d\phi_b}{d\xi}(\xi) d\xi$$

$$K_{a,b}^e = \frac{2\alpha}{h} \int_{-1}^{1} \frac{d\phi_a}{d\xi}(\xi) \frac{d\phi_b}{d\xi}(\xi) d\xi + \frac{\beta h}{2} \int_{-1}^{1} \phi_a(\xi) \phi_b(\xi) d\xi + \gamma \int_{-1}^{1} \phi_a(\xi) \frac{d\phi_b}{d\xi}(\xi) d\xi$$

Por termos definido ϕ como uma função de primeiro grau, sabemos que o produto das ϕs será necessariamente uma função de segundo grau. Por isso, podemos usar a Quadratura Gaussiana como técnica de integração para calcular a integral com 2 pontos visto que dado n pontos, essa técnica consegue integrar polinômios de grau $2n - 1$ sem errors:

$$K_{a,b}^e = \frac{2\alpha}{h} \int_{-1}^{1} \frac{d\phi_a}{d\xi}(\xi) \frac{d\phi_b}{d\xi}(\xi) d\xi + \frac{\beta h}{2} \int_{-1}^{1} \phi_a(\xi) \phi_b(\xi) d\xi + \gamma \int_{-1}^{1} \phi_a(\xi) \frac{d\phi_b}{d\xi}(\xi) d\xi$$

$$K_{a,b}^e = \frac{2\alpha}{h} \sum_{j=1}^{2} w_j \frac{d\phi_a}{d\xi}(p_j) \frac{d\phi_b}{d\xi}(p_j) + \frac{\beta h}{2} \sum_{j=1}^{2} w_j \phi_a(p_j) \phi_b(p_j) + \gamma \sum_{j=1}^{2} w_j \phi_a(p_j) \frac{d\phi_b}{d\xi}(p_j)$$

Para simplificar, podemos definir

$$QG(h, N_{PG}) = \sum_{j=1}^{N_{PG}} w_j h(p_j)$$

Com isso:

$$K_{a,b}^e = \frac{2\alpha}{h} QG\left(\frac{d\phi_a}{d\xi}\frac{d\phi_b}{d\xi}, 2\right) + \frac{\beta h}{2} QG\left(\phi_a \phi_b, 2\right) + \gamma QG\left(\phi_a \frac{d\phi_b}{d\xi}, 2\right)$$

1.4.2 Cálculo do Vetor Local por Quadratura Gaussiana

$$F = \sum_{e=1}^{m+1} F^e$$

Observe que, pela definição de φ, $F_i^e \neq 0$ quando $i \in [e-1, e]$. Logo, assim como fizemos com $K_{i,j}^e$, podemos limitar nossa integral para esse intervalo visto que restante do intervalo resultará sempre em zero. Faremos isso utilizando a notação local para as funções da base.

$$F_a^e = \int_0^1 f(x)\varphi_i(x)dx$$

$$F_a^e = \int_{x_1^e}^{x_2^e} f(x)\varphi_a^e(x)dx$$

$$F_a^e = \int_{-1}^1 f(x(\xi, e))\varphi_a^e(x(\xi, e))\frac{h}{2}d\xi$$

$$F_a^e = \frac{h}{2}\int_{-1}^1 f(x(\xi, e))\phi_a(\xi)d\xi$$

Por fim, podemos aproximar essa integral a partir da Quadratura Gaussiana. Diferente da última vez, não sabemos a definição da função f. Sendo assim, usaremos 5 pontos como aproximação:

$$F_a^e = \frac{h}{2}\sum_{j=1}^{5} w_j f(x(p_j, e))\phi_a(p_j)$$

$$F_a^e = \frac{h}{2} QG(f(x(\xi, e))\phi_a(\xi), 5)$$

1.4.3 Inicialização da Matriz

Visto que sabemos montar a matriz K a partir de $K_{a,b}^e$ e o vetor F a partir de F_a^e, podemos calcular os coeficientes. Tínhamos um código para resolver o sistema:

$$-\alpha u_{xx}(x) + \beta u(x) = f(x)$$

Portanto, com a mudança desse sistema para:

$$-\alpha u_{xx}(x) + \beta u(x) + \gamma u_x(x) = f(x)$$

A única coisa que precisamos fazer é inicializar uma variável `gamma` e também inserir $\left(\gamma QG\left(\phi_a \frac{d\phi_b}{d\xi}, 2\right)\right)$ na inicialização de nossa matrix K_e:

```
function init_Ke_matrix(ne)
  h = 1 / ne
  Ke = zeros(2,2)

  for a in 1:2
    for b in 1:2
      Ke[a,b] = (alpha * 2 / h) * gaussian_quadrature((qsi) -> d_phi(a, qsi) *
          d_phi(b, qsi), 2) + (beta * h / 2) * gaussian_quadrature((qsi) ->
          phi(a, qsi) * phi(b, qsi), 2)  + gamma * gaussian_quadrature((qsi) ->
          d_phi(b, qsi) * phi(a, qsi), 2)
    end
  end

  return Ke
end
```

1.5 Teste Realizado

Para testarmos o procedimento, precisamos primeiro escolher uma função $u(x)$ que satisfaça $u(0) = u(1) = 0$. Também precisamos definir valores para α, β e γ. Nesse caso em específico, selecionamos valores arbitrários somente para testes:

```
# Variables
alpha = 1                                       # constant value
beta = 1                                        # constant value
gamma = 1                                       # constant value
u = (x) -> sin(pi * x)
```

Em seguida, podemos calcular o valor esperado para a função $f(x)$ para que possamos calcular o erro do nosso método. $f(x)$ precisa necessariamente satisfazer:

$$f(x) = -\alpha u_{xx}(x) + \beta u(x) + \gamma u_x(x)$$

Sendo assim:

$$
\begin{aligned}
f(x) &= -\alpha u_{xx}(x) + \beta u(x) + \gamma u_x(x) \\
f(x) &= -\alpha(\sin(\pi x))_{xx} + \beta \sin(\pi x) + \gamma(\sin(\pi x))_x \\
f(x) &= -\alpha(\pi \cos(\pi x))_x + \beta \sin(\pi x) + \gamma \pi \cos(\pi x) \\
f(x) &= -\alpha \pi^2(-\sin(\pi x)) + \beta \sin(\pi x) + \gamma \pi \cos(\pi x) \\
f(x) &= \alpha \pi^2 \sin(\pi x) + \beta \sin(\pi x) + \gamma \pi \cos(\pi x)
\end{aligned}
$$

Logo:

```
f = (x) -> alpha*pi^2 * sin(pi*x) + beta*sin(pi*x) + gamma*pi*cos(pi*x)
```

Sendo assim, ao rodarmos como código com o número de elementos variando de 1 até 131071 ($2^1 - 1$ até $2^{17} - 1$), obtemos os seguintes gráficos:

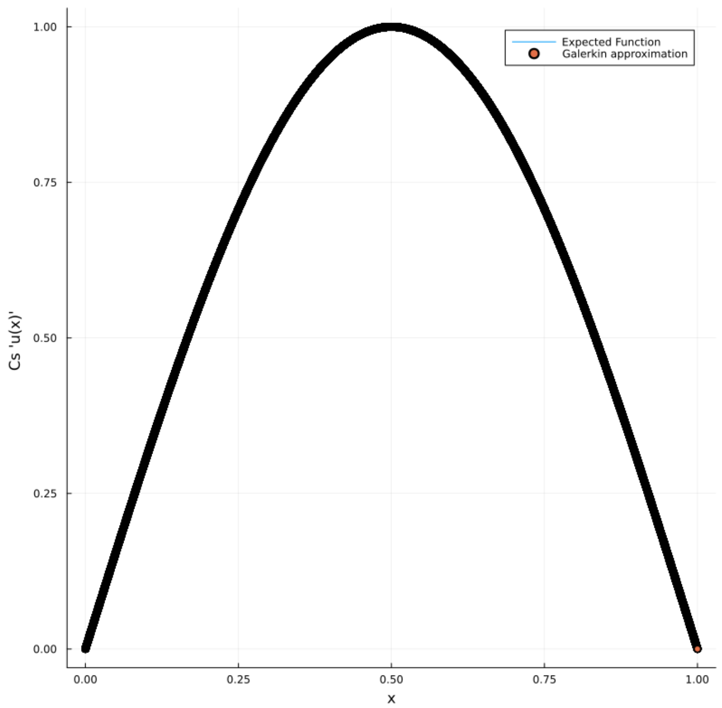

Figura 1.1: Gráfico que mostra a função original e nossos pontos aproximados por cima (Não é possível ver ambos muito bem pois o número de pontos está bem alto).

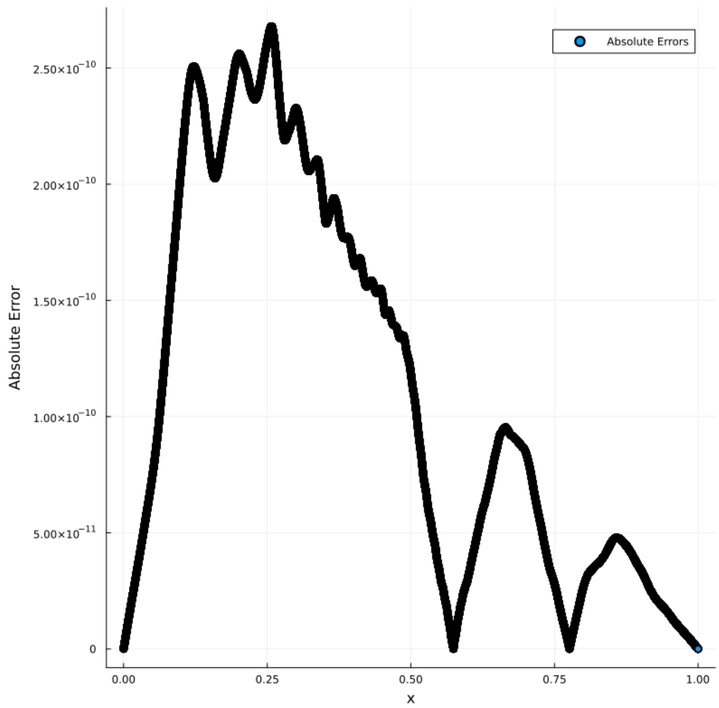

Figura 1.2: Gráfico que mostra a distância no eixo y para cada ponto encontrado pelo nosso método em comparação com a função original.

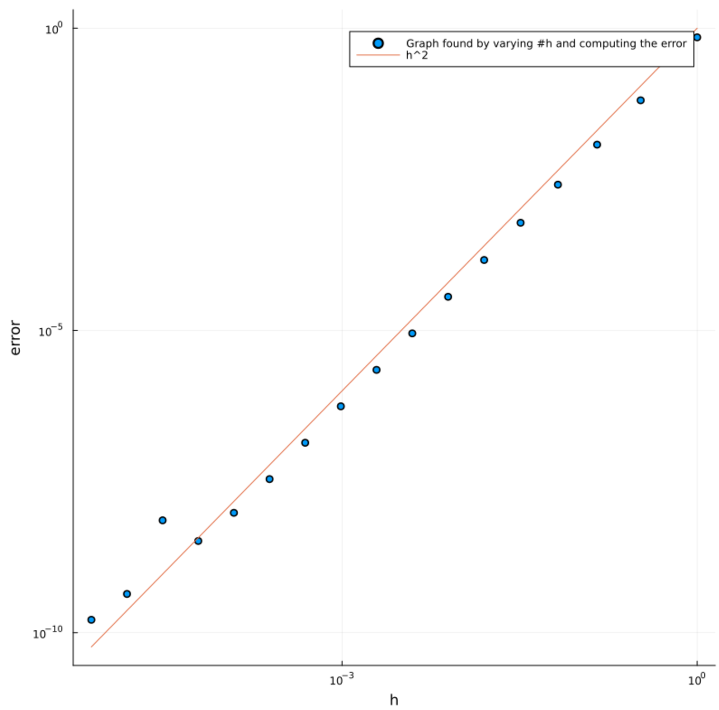

Figura 1.3: Gráfico que mostra os erros conforme o número de elementos varia.

Bibliografia

[1] Bruno Alves do Carmo. *Problema Estacionario Unidimensional*. Acessado em 7 de Setembro de 2024. URL: `https://github.com/bacarmo/Problema-estacionario-unidimensional`.

[2] Mauro A. Rincon e Waldecir Bianchini I-Shih Liu. *Introdução ao Método de Elementos Finitos*. Instituto de Matemática, UFRJ, p. 183. ISBN: 978-65-86502-00-8.