



# Universidade de Brasília

---

## Uma breve introdução aos problemas de Sturm-Liouville

---

**Matheus Horácio**

Professor: Yuri Dumaresq Sobral

Departamento de Matemática,  
Universidade de Brasília

Trabalho final da disciplina de Introdução a Métodos Computacionais em EDO

Brasília, 14 de Julho de 2025

# Sumário

<b>Introdução</b>	<b>3</b>
<b>1 Resolvendo numericamente o problema de Sturm-Liouville</b>	<b>6</b>
1.1 Problemas regulares . . . . .	6
1.1.1 O método do atirador marchante . . . . .	9
1.1.2 Usando o método de Newton-Raphson . . . . .	13
1.1.3 Resultados numéricos do método do atirador marchante . . . . .	13
1.2 Problemas singulares de tipo 1 . . . . .	18
1.3 Problemas singulares de tipo 2 . . . . .	19
1.4 Resultados numéricos do método do atirador marchante adaptado . . . . .	20
1.5 Resultados numéricos do método de Newton-Raphson . . . . .	29
1.6 Análise da ordem do método do tiro com Newton-Raphson . . . . .	30
<b>Bibliografia</b>	<b>33</b>

# Introdução

Um problema de Sturm-Liouville regular é (em dimensão um) um problema de valores de contorno de segunda ordem da forma

$$\begin{aligned} -\frac{d}{dt} \left( p(t) \frac{dy}{dt} \right) + q(t)y &= \lambda r(t)y, \quad \forall t \in [a, b], \\ g(y(a), y'(a), y(b), y'(b)) &= 0; \end{aligned} \tag{1}$$

onde  $g : \mathbb{R}^4 \longrightarrow \mathbb{R}^2$  é dada por

$$g(u_1, u_2, u_3, u_4) = \begin{pmatrix} \alpha u_1 + \beta u_2 \\ \gamma u_3 + \delta u_4 \end{pmatrix}$$

Aqui, os dados do problema são as funções reais  $p, q, r$  (com hipóteses de continuidade/suavidade apropriadas) e as constantes  $\alpha, \beta, \gamma, \delta$  (que por motivos evidentes pedimos que satisfaçam  $\{\alpha^2 + \beta^2, \gamma^2 + \delta^2\} \neq \{0\}$ ). Encontrar  $\lambda \in \mathbb{R}$  tal que o problema (1) admita uma solução não identicamente nula  $y$  é o que chamamos de *problema de autovalor de Sturm-Liouville*, nomenclatura justificada pelo fato de que se  $0 \neq \lambda$  é tal que o problema (1) admite uma solução não identicamente nula, então  $\lambda$  é um autovalor do operador

$$\mathbb{H} \ni u \mapsto L(u) = -\frac{1}{w(t)} \left( \frac{d}{dt} \left[ p(t) \frac{du}{dt} \right] + q(t)u \right) \in \mathbb{H};$$

onde  $\mathbb{H}$  é um espaço de Hilbert apropriado que não convém definirmos aqui.

O objetivo dessa introdução é convencer o leitor da grande relevância de tais problemas, tanto na matemática (pura ou aplicada) quanto na física. Mais especificamente, ilustraremos concretamente a extrema força do seguinte teorema, que é a joia de ouro da teoria de Sturm-Liouville:

**Teorema (T.0.1) (Propriedades do problema regular de Sturm-Liouville).** (a) *Existe uma quantidade infinita de autovalores reais que podem ser ordenados em ordem crescente:*

$$\lambda_0 < \lambda_1 < \lambda_2 < \cdots < \lambda_n < \cdots$$

*e de forma que  $\lambda_n \rightarrow \infty$  conforme  $n \rightarrow \infty$ .*

(b) *A dimensão de cada autoespaço é um.*

- (c) Autofunções correspondentes a autovalores distintos são linearmente independentes.
- (d) As autofunções são ortogonais com respeito ao produto interno induzido por  $p(t)$  e formam uma base ortonormal.

## A equação de Helmholtz e os autovalores de Dirichlet: é possível ouvir a forma de um tambor?

---

Pode-se mostrar que o deslocamento  $F$  de uma membrana plana  $\Omega$ , mantida fixa ao longo de sua fronteira, satisfaz a equação de onda

$$\frac{\partial^2 F}{\partial t^2} = c^2 \nabla^2 F,$$

onde  $c$  depende das propriedades físicas da membrana e da tensão aplicada. Modelando a membrana de um tambor como um aberto limitado  $\Omega \subset \mathbb{R}^2$ , temos o problema de autovalor

$$\begin{cases} -\Delta u = \lambda u & \text{em } \Omega, \\ u|_{\partial\Omega} = 0. \end{cases} \quad (2)$$

no qual buscamos  $\lambda \in \mathbb{R}$  e  $u \not\equiv 0$ . Note que tal problema pode ser visto como um problema de Sturm-Liouville  $n$ -dimensional. Tais autovalores  $\lambda$  são chamados de *frequências naturais* de vibração da membrana, e as funções  $u$  são chamadas de *modos normais* de vibração. A solução geral do problema (ou, equivalentemente, o som do tambor) é uma combinação dos modos normais.

Mark Kac, em 1966, formulou a seguinte questão: *Can one hear the shape of a drum?*: é possível determinar a forma de  $\Omega$  conhecendo apenas as suas frequências naturais de vibração, i.e, o espectro do laplaciano?

A investigação remonta a 1911, quando Weyl mostrou que a área de um domínio plano é determinada por seu espectro (ou seja, é possível “ouvir a área de um tambor!”). Posteriormente verificou-se que o espectro também determina o perímetro e o número de componentes conexas. Combinando esses fatos à desigualdade isoperimétrica

$$\text{perímetro}(\Omega) > 4\pi \text{ área}(\Omega),$$

A resposta geral da pergunta de Kac é negativa: existem domínios isoespectrais não congruentes. Porém, Kac provou que um domínio isoespectral a um disco de raio  $r$  deve ser congruente ao próprio disco, demonstrando que pelo menos alguns domínios são completamente determinados pelo seu espectro (ou seja, pode-se “ouvir” a forma de *alguns* tambores).

## Motivos físicos e aplicações

---

A equação de Schrödinger autônoma em dimensão um é dada por

$$\frac{-\hbar^2}{2m} \frac{d^2\psi(t)}{dt^2} = [E - V(t)] \psi(t),$$

que é claramente um problema de Sturm-Liouville. A equação de Schrödinger não autônoma, a saber,

$$-\frac{\hbar^2}{2m} \frac{\partial^2 \psi(x, t)}{\partial x^2} + V(t) \psi(x, t) = i\hbar \frac{\partial \psi(x, t)}{\partial t};$$

também pode ser vista como um problema de Sturm-Liouville via separação de variáveis. O teorema **(T.0.1)** pode ser visto como uma validação matemática (com certas ressalvas pertinentes à dimensionalidade, domínios infinitos e singularidades, e com sutilezas que dependem de quão intimamente o problema de Sturm-Liouville se relaciona ao problema original de Schrödinger) dos seguintes fatos:

- Existe uma sequência *discreta* de níveis de energia, rotulados por um número quântico  $n$ . Essa natureza discreta é a razão de a física quântica ser chamada de “quântica”. Os valores de energia dos estados formam a sequência crescente  $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \dots$ , que tende ao infinito. Assim, existe uma noção bem-definida de *estado fundamental* (energia mínima): é o estado com  $n = 1$ .
- As funções de onda de dois níveis de energia com valores distintos podem ser tomadas ortonormais.
- Qualquer função de onda geral  $u$  pode ser escrita como combinação linear  $u = \sum_n c_n u_n$  das funções de onda de nível de energia puro  $u_n$ . Esse é o enunciado de *completude* na teoria de Sturm-Liouville. Na linguagem dos físicos, o estado  $u$  é uma *superposição* dos estados de nível de energia puro.

## Motivos matemáticos

Considere uma EDO linear de segunda ordem qualquer, digamos

$$\psi(t)y'' + \varphi(t)y' + \eta(t)y = f(t), \text{ onde } \psi \neq 0. \quad (3)$$

Definindo

$$\begin{aligned} p(t) &= \exp \left( \int \frac{\varphi(t)}{\psi(t)} dt \right), \\ q(t) &= p(t) \frac{\eta(t)}{\psi(t)}, \\ F(t) &= p(t) \frac{f(t)}{\psi(t)}; \end{aligned} \quad (4)$$

vemos que (3) é equivalente a uma EDO na forma de Sturm-Liouville. De fato, podemos reescrever (3) como

$$\frac{d}{dt} \left( p(t) \frac{dy}{dt} \right) + q(t)y = F(t), \quad (5)$$

Isso prova uma das forças da teoria de Sturm-Liouville: reduzimos o estudo de todas as EDOs lineares de segunda ordem a uma só! Além disso, pode-se mostrar que qualquer solução  $y$  de (5) é uma combinação linear de autofunções  $u_n$  do problema de Sturm-Liouville

$$-\frac{d}{dt} \left( p(t) \frac{du}{dt} \right) + q(t)u = \lambda r(t)u \quad (6)$$

---

# Capítulo 1

## Resolvendo numericamente o problema de Sturm-Liouville

---

Vamos usar o método do tiro para resolver numericamente problemas de Sturm-Liouville, ou seja, determinar numericamente os autovalores e autofunções de tais problemas.

**Observação (O.1).** Os códigos desenvolvidos para resolver numericamente todos os exemplos apresentados aqui estão disponíveis nos repositórios a seguir

- [https://github.com/SaganGromov/SL\\_REGULAR\\_EXEMPLOS](https://github.com/SaganGromov/SL_REGULAR_EXEMPLOS)
- [https://github.com/SaganGromov/SL\\_SINGULAR\\_EXEMPLOS](https://github.com/SaganGromov/SL_SINGULAR_EXEMPLOS)
- [https://github.com/SaganGromov/SL\\_NEWTON\\_RAPHSON](https://github.com/SaganGromov/SL_NEWTON_RAPHSON)

### 1.1 Problemas regulares

---

**Definição (D.1.1).** Uma equação de Sturm-Liouville da forma

$$-(p(x)y'(x))' + q(x)y(x) = \lambda r(x)y(x), \quad \forall x \in [a, b], \quad (1.1)$$

é dita *regular* se as funções  $p$ ,  $q$  e  $r$  são suaves em  $[a, b]$ , com  $p, q$  estritamente positivas em  $[a, b]$ .

**Definição (D.1.2).** Um problema de Sturm-Liouville *regular* é um problema da forma (1.1) sujeito a condições de contorno lineares separadas, a saber,

$$\alpha \cdot y(a) + \beta \cdot y'(a) = 0; \quad (1.2)$$

$$\gamma \cdot y(b) + \delta \cdot y'(b) = 0; \quad (1.3)$$

onde  $\alpha, \beta, \gamma, \delta$  são constantes reais tais que  $(\alpha, \beta) \neq (0, 0)$  e  $(\gamma, \delta) \neq (0, 0)$

Antes de explicitarmos o método do tiro, vamos enunciar vários resultados clássicos da teoria de Sturm-Liouville que serão essenciais para criarmos nosso algoritmo que determina numericamente o

espectro e as autofunções de um problema de Sturm-Liouville regular. A demonstração de tais resultados está fora do escopo desse trabalho, mas pode ser encontrada em [2].

**Teorema (T.1.1).** • *Os autovalores de um problema regular são reais, simples e podem ser ordenados como*

$$\lambda_0 < \lambda_1 < \lambda_2 < \dots$$

- *A autofunção  $y_n$  correspondente a  $\lambda_n$  tem exatamente  $n$  zeros nodais (zeros interiores onde ocorre uma mudança de sinal) e nenhum outro zero interior. Além disso, os nós de  $y_{n-1}$  e  $y_n$  estritamente se entrelaçam - ou seja, entre quaisquer dois zeros consecutivos de  $y_n$ , existe exatamente um zero de  $y_{n-1}$ .*
- *Se  $\alpha\beta \leq 0$ ,  $\gamma\delta \geq 0$ ,  $p > 0$ ,  $q \geq 0$ , e  $r(x) > 0$  em  $[a, b]$ , então todos os autovalores são positivos, exceto no caso em que o problema de autovalores é  $-(py')' = \lambda ry$ ,  $y'(a) = 0$ ,  $y'(b) = 0$ , no qual caso 0 é um autovalor e todos os outros autovalores são positivos. (Ver Teorema 124.)*
- *Se  $\alpha\beta \leq 0$ ,  $\gamma\delta \geq 0$ ,  $p > 0$ ,  $q \geq 0$ , e  $r(x) > 0$  em  $[a, b]$ , então  $\lambda_0 \geq \min_{x \in [a, b]} \frac{q(x)}{r(x)}$  e  $\lambda_0 > 0$  a menos que o problema de autovalores seja  $-(py')' = \lambda ry$ ,  $y'(a) = 0$ ,  $y'(b) = 0$ , no qual caso 0 é um autovalor e todos os outros autovalores são positivos.*

**Observação (O.2).** Observe que estamos começando a contagem em  $n = 0$ . Isso significa que o autovalor  $\lambda_0$  é o menor autovalor, e a autofunção  $y_0$  é a autofunção correspondente a  $\lambda_0$ . A autofunção  $y_0$  não tem zeros nodais, em particular, nunca muda de sinal em  $(a, b)$ .

Consideremos agora o problema regular de autovalor, i.e

$$\begin{cases} -(p(x)y'(x))' + (q(x) - \lambda r(x))y(x) = 0, & \forall x \in [a, b], \\ \alpha y(a) + \beta y'(a) = 0, \\ \gamma y(b) + \delta y'(b) = 0, \end{cases} \quad (1.4)$$

com os coeficientes e constantes satisfazendo as hipóteses já mencionadas. O método que utilizaremos para resolver esse PVC consiste em transformar o PVC em um PVI. Uma vez que os autovalores  $\lambda_i$  de (1.4) são todos simples e as autofunções correspondentes satisfazem  $\alpha y(a) + \beta y'(a) = 0$ , o vetor  $(y(a), y'(a))$  é paralelo a  $(-\beta, \alpha)$ . Segue que existe uma única autofunção normalizada correspondente a cada autofunção, que é exatamente a solução do PVI a seguir:

$$\begin{cases} -(p(x)u'(x))' + (q(x) - \lambda r(x))u(x) = 0, & \forall x \in [a, b], \\ u(a) = -\frac{\beta}{\sqrt{\alpha^2 + \beta^2}}, & u'(a) = \frac{\alpha}{\sqrt{\alpha^2 + \beta^2}}. \end{cases} \quad (1.5)$$

Denotemos então a solução global do PVI por  $u = u(x) = u(x, \lambda)$ , que está definida em  $[a, b]$  seja qual for nossa escolha de  $\lambda$ . Agora, lembremos que  $\lambda$  é um autovalor do problema de Sturm-Liouville (1.4) se e somente se o PVI possuir uma solução  $u(x, \lambda)$  definida em  $[a, b]$  que satisfaz  $F(\lambda) = 0$ , onde

$$F(\lambda) = \gamma u(b, \lambda) + \delta u'(b, \lambda),$$

Nesse caso,  $y(x) = u(x, \lambda)$  é a autofunção correspondente normalizada.

Sabemos então que as raízes de  $F(\lambda)$  são exatamente os autovalores do problema de autovalor de Sturm-Liouville. Isso motiva então usarmos o seguinte algoritmo:

- (i). **Passo 1.** Determinamos um chute inicial para um autovalor de interesse (ou, ainda melhor, um intervalo que contenha um autovalor de interesse).
- (ii). **Passo 2.** Resolvemos o PVI com o chute inicial  $\lambda_k^0$  e obtemos a solução  $u(x, \lambda_k^0)$ .
- (iii). **Passo 3.** Calculamos  $F(\lambda_k^0) = \gamma u(b, \lambda_k^0) + \delta u'(b, \lambda_k^0)$ .
- (iv). **Passo 4.** Se  $|F(\lambda_k^0)| < \varepsilon$ , então  $\lambda_k^0$  é um autovalor (aproximado) do problema de Sturm-Liouville. Caso contrário, usamos algum método de busca de raízes (como o método da bissecção) para atualizar o palpite de  $\lambda_k^0$  e voltamos ao passo 2.

Para usarmos o método da bissecção, precisamos saber a priori que  $F$  muda de sinal em cada autovalor. Isso de fato é verdade, como demonstrado em [2]. Podemos também utilizar o método de Newton-Raphson: de fato, denotando

$$w(x, \lambda) := \frac{\partial u}{\partial \lambda}(x, \lambda);$$

e derivando termo a termo usando o PVI, obtemos

$$-(pw')' + (q - \lambda r)w - ru = 0.$$

Agora, como os dados iniciais *não* dependem de  $\lambda$ , segue que

$$w(a, \lambda) = 0, \quad w'(a, \lambda) = 0.$$

Isso nos dá outro PVI, a saber,

$$\begin{aligned} -(pw')' + (q - \lambda r)w &= ru, \quad \forall x \in [a, b], \\ w(a) &= 0, \quad w'(a) = 0. \end{aligned}$$

Tal PVI nos permite determinar numericamente  $F'(\lambda)$ ,

$$F'(\lambda) = \gamma w(b, \lambda) + \delta w'(b, \lambda).$$

Como mostrado em [2],  $F'(\lambda) \neq 0$ . Assim, podemos usar o método de Newton-Raphson para atualizar o palpite de  $\lambda_k^n$ , fazendo

$$\lambda_k^{(n+1)} = \lambda_k^{(n)} - \frac{F(\lambda_k^{(n)})}{F'(\lambda_k^{(n)})}.$$

Começaremos aplicando essa estratégia (inicialmente usando o método da bissecção) para determinar o primeiro autovalor  $\lambda_0$  e sua autofunção corresponde  $y_0$ . Para muitos problemas que consideraremos, pode-se mostrar a priori que  $\lambda_0 > 0$ , tornando razoável um palpite inicial  $\lambda_0^0 = \varepsilon$  com algum  $\varepsilon > 0$  pequeno.



Determinando um incremento  $\Delta\lambda$  pequeno, podemos então iterar o método da bissecção em intervalos da forma  $[\lambda_k^i + k\Delta\lambda, \lambda_k^i + (k+1)\Delta\lambda]$  até encontrarmos quantos autovalores e autofunções quisermos. Evidentemente, essa abordagem é muito cara e ineficiente, mas possui duas vantagens em comparação com o método de Newton-Raphson: ela não depende de um bom palpite inicial para qualquer  $k$ -ésimo autovalor que queiramos encontrar, e pode ser iterada sucessivamente para determinarmos quaisquer quantidade de autovalores. Note que a mesma abordagem munida do método de Newton-Raphson enfrenta graves problemas - a rapidez/convergência quadrática do método vale somente localmente, tornando possível divergência ou encontro de valores espúrios. A técnica de incrementar  $\lambda$  em pequenos passos fixos  $\Delta\lambda$  e iterar o método de bissecção para percorrer todo o espectro também obviamente não funciona para o método de Newton-Raphson.

## 1.1 O método do atirador marchante

---

Nessa seção explicaremos o passo-a-passo de um programa desenvolvido em Fortran cujo objetivo é resolver numericamente qualquer problema de Sturm-Liouville usando o método do tiro, conforme descrito acima. O programa é capaz de encontrar os autovalores e as autofunções correspondentes para um problema regular de Sturm-Liouville com condições de contorno lineares separadas. Para resolver o PVI associado, usamos o método de Runge-Kuta clássico, de ordem quatro, e para encontrar as raízes de  $F$  usamos o método da bissecção. A seguir, apresentamos uma análise passo a passo do algoritmo implementado.

### PASSO 1. Configuração e Preparação do Problema

---

#### SUBPASSO 1.1. Inicialização do domínio e condições de contorno - `constantes.f90`

Definimos o domínio  $[a, b]$  do problema (na versão atual do código,  $[0, 1]$ ) e inicializamos  $\alpha, \beta, \gamma, \delta$  como constantes reais. No exemplo inicial implementado, usamos  $\alpha = \gamma = 1$  e  $\beta = \delta = 0$ , correspondendo às condições de contorno  $y(0) = y(1) = 0$ .

#### SUBPASSO 1.2. Inicialização de diversos parâmetros numéricos - `constantes.f90`

- `passos = 100000`: Número de passos de integração utilizados para resolver o PVI via RK4
- `tol = 1e-12`: Tolerância para a convergência do método de bissecção
- `n_eigen = 10`: Número de autovalores a ser encontrado
- `verbose_level = 3`: Nível de verbosidade desejado ao executar o programa compilado, serve principalmente para debug

**SUBPASSO 1.3. Definição do problema - problema.f90**

Especificamos qual problema de Sturm-Liouville iremos considerar ao definir as funções  $p(x)$ ,  $q(x)$  e  $r(x)$  como funções Fortran. Por exemplo, para o problema de Dirichlet clássico, definimos:

$$\begin{aligned}p(x) &= 1 \\q(x) &= x \\r(x) &= 0\end{aligned}$$

correspondente à EDO

$$y'' + \lambda y = 0$$

**PASSO 2. EDO de 2ª ordem como um sistema de ordem um - definicao\_ED0.f90**

---

Convertemos a EDO de segunda ordem inicial num sistema de primeira ordem fazendo

$$\begin{cases} y_1(x) = y(x), \\ y_2(x) = p(x)y'(x) \end{cases}$$

de forma que

$$\begin{cases} y_1' = \frac{y_2}{p(x)}, \\ y_2' = (q(x) - \lambda r(x)) y_1, \end{cases} \quad (1.6)$$

**PASSO 3. Definição do método RK4 - integrador\_rk4.f90**

---

O método clássico de Runge-Kutta de 4ª ordem é implementado com tamanho de passo

$$h = \frac{b - a}{\text{passos}} = 10^{-5},$$

onde, para o sistema  $\mathbf{y}' = \mathbf{f}(x, \mathbf{y}, \lambda)$ , temos:

$$\mathbf{k}_1 = \mathbf{f}(x_n, \mathbf{y}_n, \lambda), \quad (1.7)$$

$$\mathbf{k}_2 = \mathbf{f}\left(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_1, \lambda\right), \quad (1.8)$$

$$\mathbf{k}_3 = \mathbf{f}\left(x_n + \frac{h}{2}, \mathbf{y}_n + \frac{h}{2}\mathbf{k}_2, \lambda\right), \quad (1.9)$$

$$\mathbf{k}_4 = \mathbf{f}\left(x_n + h, \mathbf{y}_n + h\mathbf{k}_3, \lambda\right), \quad (1.10)$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \frac{h}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4). \quad (1.11)$$

## PASSO 4. Implementação do Método do Tiro - `atirador.f90`

---

### SUBPASSO 4.1. Resolução do PVI

Agora resolvemos numericamente o PVI (1.5), a saber,

$$\begin{cases} -(p(x)u'(x))' + (q(x) - \lambda r(x))u(x) = 0, & \forall x \in [a, b], \\ u(a) = -\frac{\beta}{\sqrt{\alpha^2 + \beta^2}}, & u'(a) = \frac{\alpha}{\sqrt{\alpha^2 + \beta^2}}. \end{cases}$$

Armazenamos a solução numérica  $y$  e sua derivada  $y'$  em um vetor de tamanho `passos`. Além disso, fazemos a “contabilidade” dos nós, ou seja, contamos quantas vezes a função  $y_1(x)$  muda de sinal no intervalo  $[a, b]$ . O sistema é integrado de  $x = a$  até  $x = b$ , contando as mudanças de sinal em  $y_1(x)$  (nós):

- Inicializamos `nodes = 0`, `yprev = y_1(a)`
- Para cada passo/iteração do loop de integração:
  - Se  $y_1(x_{i+1}) \cdot y_{\text{prev}} < 0$ , incremente `nodes`
  - Atualize  $y_{\text{prev}} = y_1(x_{i+1})$

Finalmente, calculamos o resíduo associado à condição de contorno em  $b$ :

$$F(\lambda) = \gamma y_1(b) + \delta \frac{y_2(b)}{p(b)} = \gamma y(b) + \delta y'(b) \quad (1.12)$$

## PASSO 5. A marcha da busca por autovalores - `espectro_melhorado.f90`

---

### SUBPASSO 5.1. Varrendo valores de $\lambda$

O algoritmo varre valores de  $\lambda$  começando do 0 com um incremento  $\Delta\lambda$  adaptativo:

- $\Delta\lambda$  inicial é definido como 0.01.
- Após encontrar autovalores, refinamos  $\Delta\lambda = 0.01 \cdot (\lambda_n - \lambda_{n-1})$

### SUBPASSO 5.2. Detecção de candidatos a autovalores

Um candidato a autovalor é detectado quando acontece uma das seguintes condições:

1. Mudança de sinal em  $F(\lambda)$ :  $F(\lambda_{\text{prev}}) \cdot F(\lambda) < 0$
2. Incremento de nós:  $\text{nodes}(\lambda) > \text{nodes}(\lambda_{\text{prev}})$

## PASSO 6. Bissecção refinada - `modulo_bisseccao.f90`

---

### Resumo dos Métodos de Bissecção:

- **Método 1: Bissecção por Sinal (`bissec_sig`)**
  - Utilizado quando a função  $F(\lambda)$  apresenta mudança de sinal no intervalo.
  - Realiza a bissecção padrão para encontrar a raiz de  $F(\lambda) = 0$ .
  - Opera com uma tolerância de  $10^{-12}$ .
- **Método 2: Bissecção por Nós (`bissec_nos`)**
  - Empregado quando a contagem de nós aumenta em 1 no intervalo.
  - Divide o intervalo para identificar onde ocorre a mudança na contagem de nós.
  - Emite um aviso se a variação dos nós não for exatamente 1.
- **Método 3: Bissecção Robusta (`bissec_robusta`)**
  - Avalia os dois extremos do intervalo para determinar a condição apropriada.
  - Seleciona o método adequado (por sinal ou por contagem de nós) com base nas condições observadas.
  - Se nenhum dos métodos anteriores se aplicar, utiliza o ponto médio como critério de bissecção.

## PASSO 7. Validação de autovalores - `espectro_melhorado.f90`

---

### SUBPASSO 7.1. Verificação de critérios de validação

Cada candidato a autovalor  $\lambda_i$  é validado ao verificarmos que as duas condições abaixo são satisfeitas:

1.  $|F(\lambda)| < \varepsilon$  (resíduo aproximadamente nulo)
2.  $F(\lambda - \varepsilon) \cdot F(\lambda + \varepsilon) < 0$  (mudança de sinal)

Candidatos que falhem essa validação são rejeitados como espúrios.

## PASSO 8. Salvando a solução - `solucao.f90`

---

Para cada autovalor encontrado, calculamos a autofunção correspondente:

- Reintegramos o PVI de  $x = a$  a  $x = b$  usando o autovalor encontrado
- Salvamos os valores  $(x, y_1(x))$  nos arquivos `solucao_autovalor_N.dat`

## 1.1 Usando o método de Newton-Raphson

---

Desenvolvemos também um programa que implementa o método de Newton-Raphson para encontrar os autovalores e autofunções correspondentes de um problema regular de Sturm-Liouville. Como a implementação nesse caso é muito mais simples, não explicaremos o passo a passo do programa agora. Em contraste com o algoritmo de marcha usando o método da bissecção que descrevemos acima, nesse caso só conseguimos encontrar um autovalor de cada vez, e precisamos de um bom chute inicial para cada autovalor que queremos encontrar.

Lembramos que Newton-Raphson atualiza o chute inicial via

$$\lambda_k^{(n+1)} = \lambda_k^{(n)} - \frac{F(\lambda_k^{(n)})}{F'(\lambda_k^{(n)})}.$$

onde  $F(\lambda)$  é o resíduo associado ao PVI (1.5) e  $F'(\lambda)$  é a derivada de  $F$  em relação a  $\lambda$ , que pode ser obtida resolvendo o PVI

$$\begin{cases} -(pw')' + (q - \lambda r)w = ru, & \forall x \in [a, b], \\ w(a) = 0, & w'(a) = 0. \end{cases}$$

De fato,

$$F'(\lambda) = \gamma w(b, \lambda) + \delta w'(b, \lambda).$$

## 1.1 Resultados numéricos do método do atirador marchante

---

**Exemplo (E.1.1).** Testaremos a abordagem do atirador marchante com o problema de Sturm-Liouville regular mais simples que há, o problema de Dirichlet clássico:

$$\begin{cases} y'' + \lambda y = 0, & \forall x \in [0, 1]; \\ y(0) = y(1) = 0. \end{cases}$$

Pela construção do método, devemos encontrar os autovalores  $\lambda_n = (n+1)^2\pi^2$ , com as autofunções normalizadas correspondentes

$$y_n(x) = \frac{\sin((n+1)\pi x)}{(n+1)\pi}, \quad n = 0, 1, 2, \dots$$

Note que a normalização se deve ao fato de estarmos resolvendo o PVI (1.5), i.e

$$\begin{cases} -(1 \cdot u'(x))' + (0 - \lambda \cdot 1)u(x) = 0, & \forall x \in [0, 1], \\ u(0) = 0, & u'(0) = 1. \end{cases} \quad (1.13)$$

Segue abaixo uma tabela com os autovalores encontrados pelo programa, bem como o número de iterações da bissecção necessárias para encontrá-los. Devido a abordagem cuidadosa do método de

incrementar  $\lambda$  em passos muito pequenos, precisamos buscar por autovalores em milhares de intervalos, sendo a vasta maioria (na verdade, todos exceto um - o último) deles descartados.

$n$	$\lambda_n$	Intervalos buscados	Iterações da bissecção
0	9.86960440	987	34
1	39.47841760	3938	34
2	88.82643961	4105	39
3	157.91367042	4245	39
4	246.74011003	4374	40
5	355.30575844	4497	40
6	483.61061565	4616	40

Plotamos também as autofunções correspondentes  $y_n = \frac{\sin((n+1)\pi x)}{(n+1)\pi}$  e as autofunções determinadas numericamente pelo programa. Conforme veremos abaixo, não há nenhuma diferença humanamente perceptível entre as autofunções determinadas numericamente pelo programa e as autofunções teóricas.

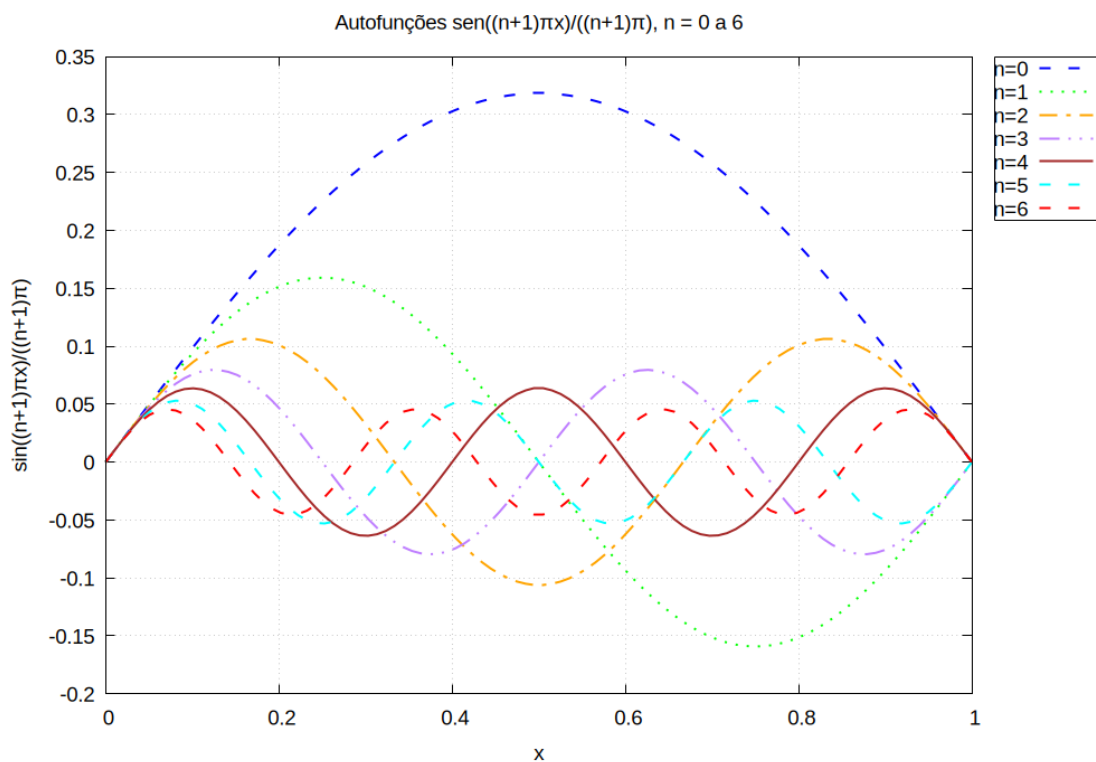


Figura 1.1: Autofunções exatas

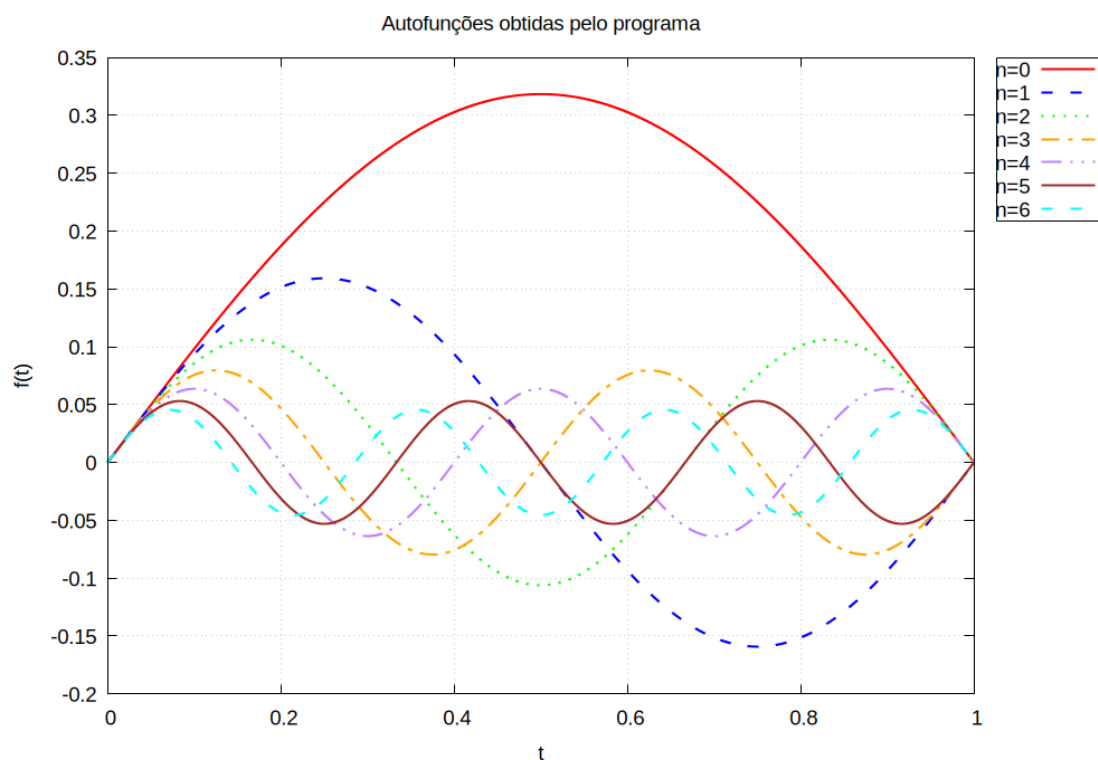


Figura 1.2: Autofunções determinadas numericamente pelo programa

**Exemplo (E.1.2).** Vamos agora determinar numericamente os cinco primeiros autovalores e autofunções do problema de Sturm-Liouville regular

$$\begin{cases} y'' + \lambda y = 0, & 0 \leq x \leq 1, \\ y(0) = 0, \\ y(1) + y'(1) = 0. \end{cases}$$

Numericamente, obtemos os seguintes autovalores:

$n$	$\lambda_n$	Intervalos buscados	Iterações da bissecção
0	4.11585837	412	34
1	24.13934203	2405	34
2	63.65910655	2602	38
3	122.88916176	2752	39
4	201.85125830	2886	40

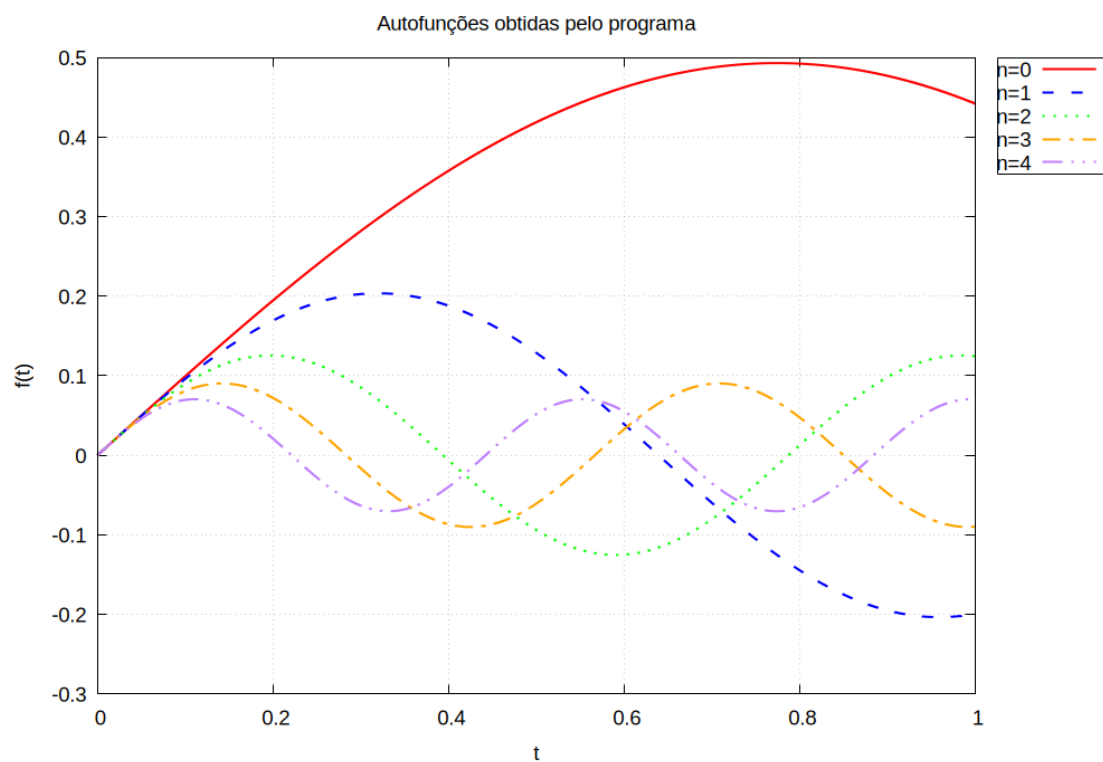


Figura 1.3: Autofunções determinadas numericamente pelo programa

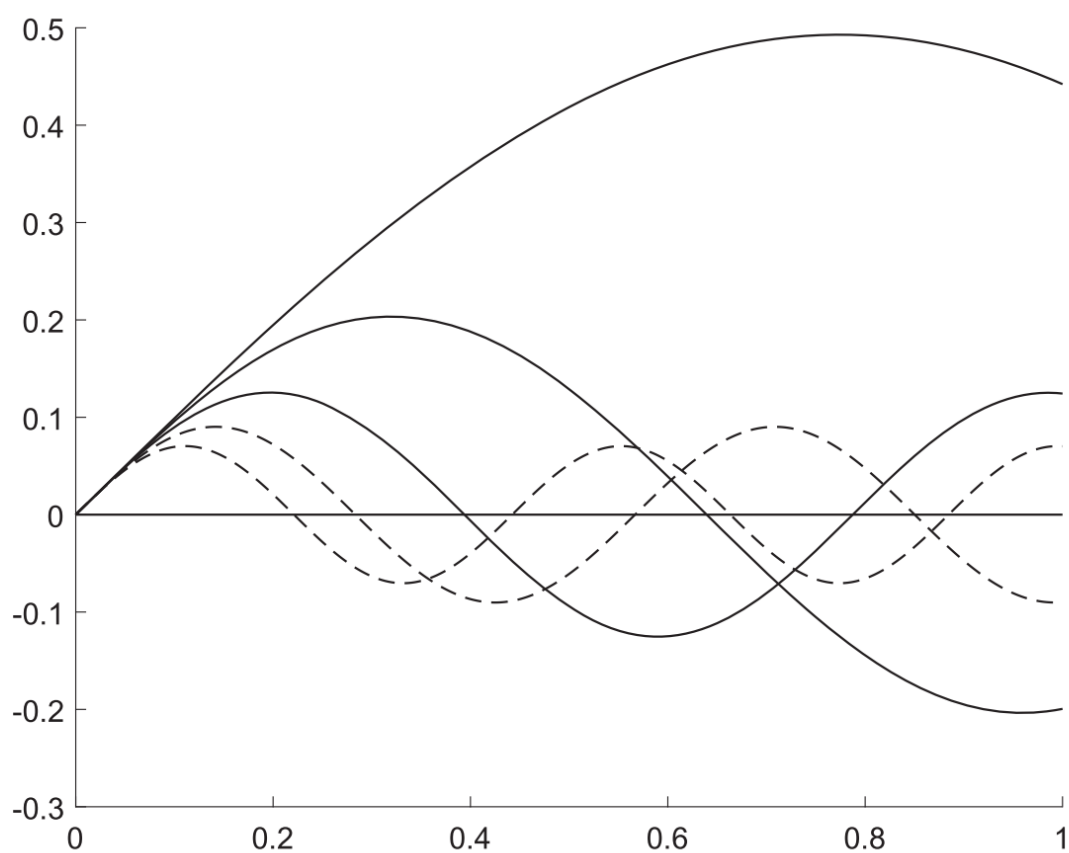


Figura 1.4: Gráficos das autofunções como mostrado no livro [2]



**Exemplo (E.1.3).** Vamos agora determinar numericamente os cinco primeiros autovalores e autofunções do problema de Sturm-Liouville regular

$$\begin{cases} -(1 \cdot u'(x))' + (x - \lambda \cos(x))u(x) = 0, & \forall x \in [0, 1], \\ u'(0) = 0, & u'(1) = 0. \end{cases} \quad (1.14)$$

Numericamente, obtemos os seguintes autovalores:

$n$	$\lambda_n$	Intervalos buscados	Iterações da bissecção
0	0.57824838	58	34
1	13.01659499	1292	34
2	48.48917784	1577	37
3	107.57497472	1744	39
4	190.28375117	1884	40

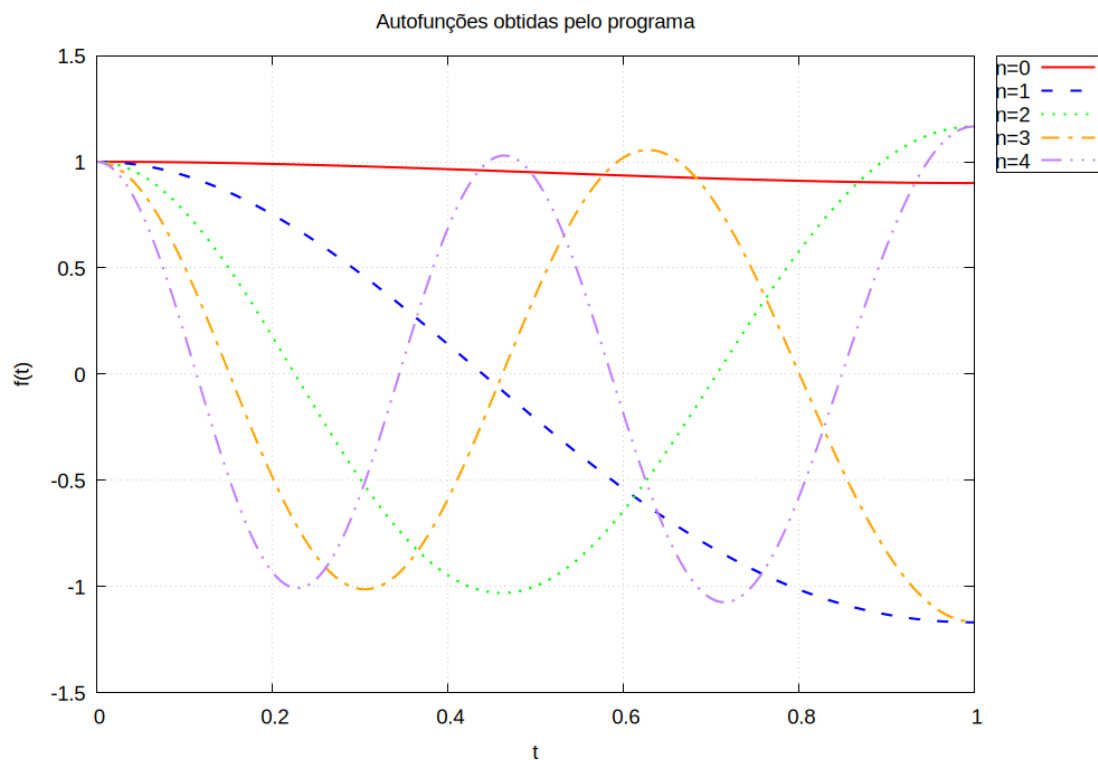


Figura 1.5: Autofunções determinadas numericamente pelo programa

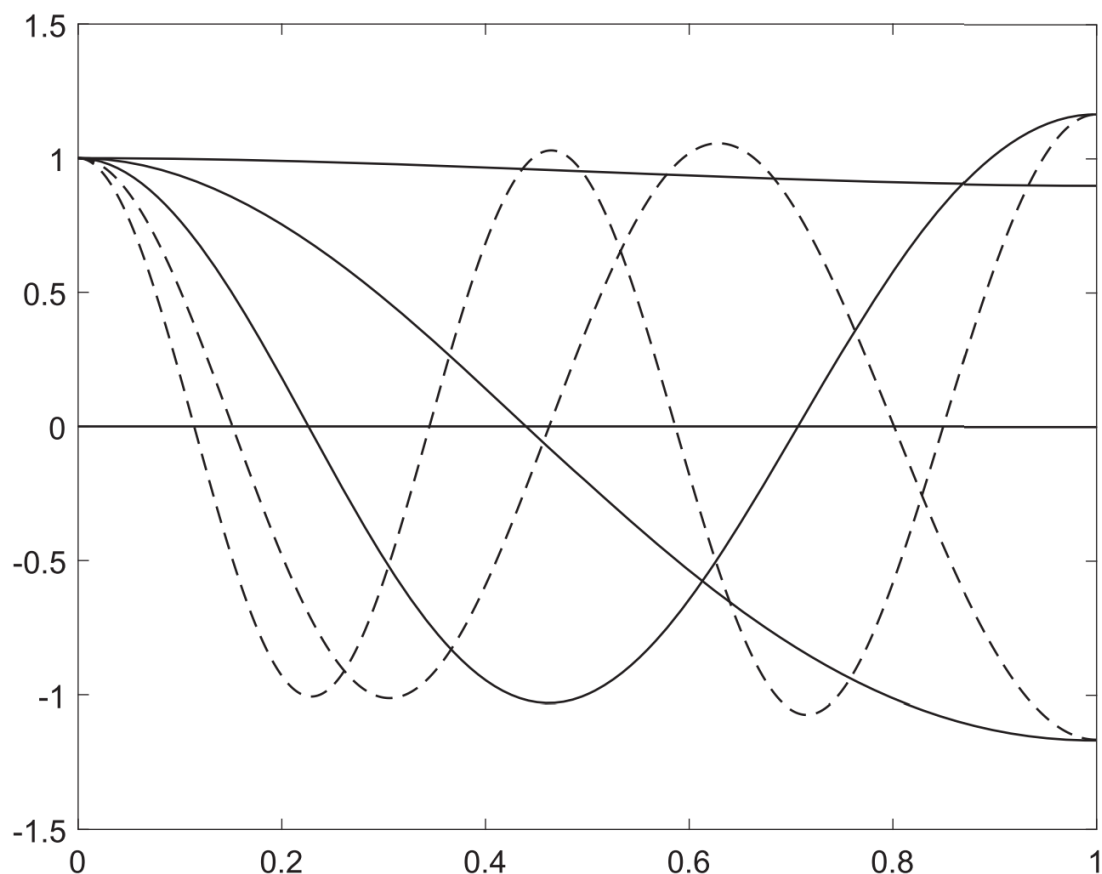


Figura 1.6: Gráficos das autofunções como mostrado no livro [2]

## 1.2 Problemas singulares de tipo 1

Ajustaremos agora a abordagem desenvolvida acima para lidar com problemas de Sturm-Liouville singulares, ou seja, problemas que não são regulares. Um problema de Sturm-Liouville singular de tipo 1 é um problema da forma:

$$\begin{cases} -(p(x)y')' + q(x)y = \lambda r(x)y, & a < x < b, \\ |y(a)| < \infty, \\ \gamma y(b) + \delta y'(b) = 0, & |\gamma| + |\delta| \neq 0, \end{cases} \quad (1.15)$$

onde agora

- (1)  $p(x) \geq 0$  é suave em  $[a, b]$ , não nula em  $a < x \leq b$ , e satisfaz  $p(a) = 0$ ,  $p'(a) \neq 0$ .
- (2)  $q(x)$  é suave em  $[a, b]$ .
- (3)  $p(x)$  e  $q(x)$  são funções reais,  $\gamma$  e  $\delta$  são constantes reais.
- (4)  $r(x)$  é suave em  $[a, b]$ , valendo ou  $r(x) > 0$  em  $[a, b]$  ou  $r(x) = (x - a)^m \rho(x)$  onde  $m > 0$  e  $\rho(x) > 0$  é suave em  $a \leq x \leq b$ .

Observemos agora que o fato de  $p$  se anular em  $a$  condena fatalmente a abordagem definida em (1.6), introduzindo uma divisão por zero em  $x = a$ . Para lidar com isso, começaremos a integrar a partir de  $a + \varepsilon$ ,  $\varepsilon$  sendo um número positivo pequeno. Evidentemente, também é necessário mudar o PVI que estamos resolvendo.

Como mostrado em [2], o PVI e normalização apropriado a ser resolvido é

$$\begin{cases} -(p(x)u'(x))' + (q(x) - \lambda r(x))u(x) = 0, & a \leq x \leq b, \\ u(a) = 1, & u'(a) = \frac{q(a) - \lambda r(a)}{p'(a)}. \end{cases}$$

A condição de contorno em  $b$  permanece a mesma, ou seja,  $\gamma u(b) + \delta u'(b) = 0$ . Além disso, também é possível determinar  $F'(\lambda)$  via outro PVI, como veremos a seguir

**Teorema (T.1.2) (Teorema 190, [2]).** *Se  $\lambda$  é um autovalor do problema de Sturm-Liouville (1.16), então  $F(\lambda) = 0$  e*

$$F'(\lambda) = \gamma w(b, \lambda) + \delta w'(b, \lambda) \neq 0$$

onde  $w = w(x, \lambda) = \frac{\partial u(x, \lambda)}{\partial \lambda}$  e  $y = u(x, \lambda)$  é a autofunção normalizada correspondente.

### 1.3 Problemas singulares de tipo 2

---

Um problema de Sturm-Liouville singular de tipo 2 é um problema da forma:

$$\begin{cases} -(p(x)y')' + q(x)y = \lambda r(x)y, & a < x < b, \\ |y(a)| < \infty, \\ \gamma y(b) + \delta y'(b) = 0, & |\gamma| + |\delta| \neq 0, \end{cases} \quad (1.16)$$

onde agora

- (1)  $p(x) = (x - a)\varphi(x)$ , onde  $\varphi > 0$  e suave em  $[a, b]$ .
- (2)  $q(x) = \frac{q_1(x)}{x - a}$ , onde  $q_1$  assume valores reais e é suave em  $[a, b]$ , além de satisfazer  $q_1(a) > 0$ .
- (3)  $r(x)$  é suave em  $[a, b]$ ,  $\gamma$  e  $\delta$  são constantes reais tais que  $\gamma\delta \geq 0$ .

Notemos que agora também vale  $p(a) = 0$ , e mais ainda, é possível que  $|q(x)| \rightarrow \infty$  conforme  $x \rightarrow a$ . Porém, ainda podemos utilizar a mesma abordagem para problemas singulares de tipo 1. Isso tornará as autofunções que encontrarmos não normalizadas como na convenção adotada em [2], o que evidentemente não é um grande problema, já que as mesmas irão se distinguir das normalizadas tão somente por um fator multiplicativo constante.

### 1.4 Resultados numéricos do método do atirador marchante adaptado

---

Utilizaremos quase exatamente a mesma abordagem que usamos para problemas regulares, as únicas diferenças sendo que agora resolvemos outro PVI ao atirar e integramos de  $x = a + 10^{-6}$  até  $x = b$ .

**Exemplo (E.1.4).** Consideremos o problema de Sturm-Liouville singular de tipo 1

$$\begin{cases} -(xy')' = \lambda xy, & 0 < x < 1, \\ |y(0)| < \infty, \quad y(1) + y'(1) = 0. \end{cases}$$

Numericamente, obtemos os seguintes autovalores:

$n$	$\lambda_n$	Intervalos buscados	Iterações da bissecção
0	1.57699273	158	24
1	16.64213839	1655	24
2	51.20546183	1884	28
3	105.49314030	2041	29
4	179.51705520	2178	30

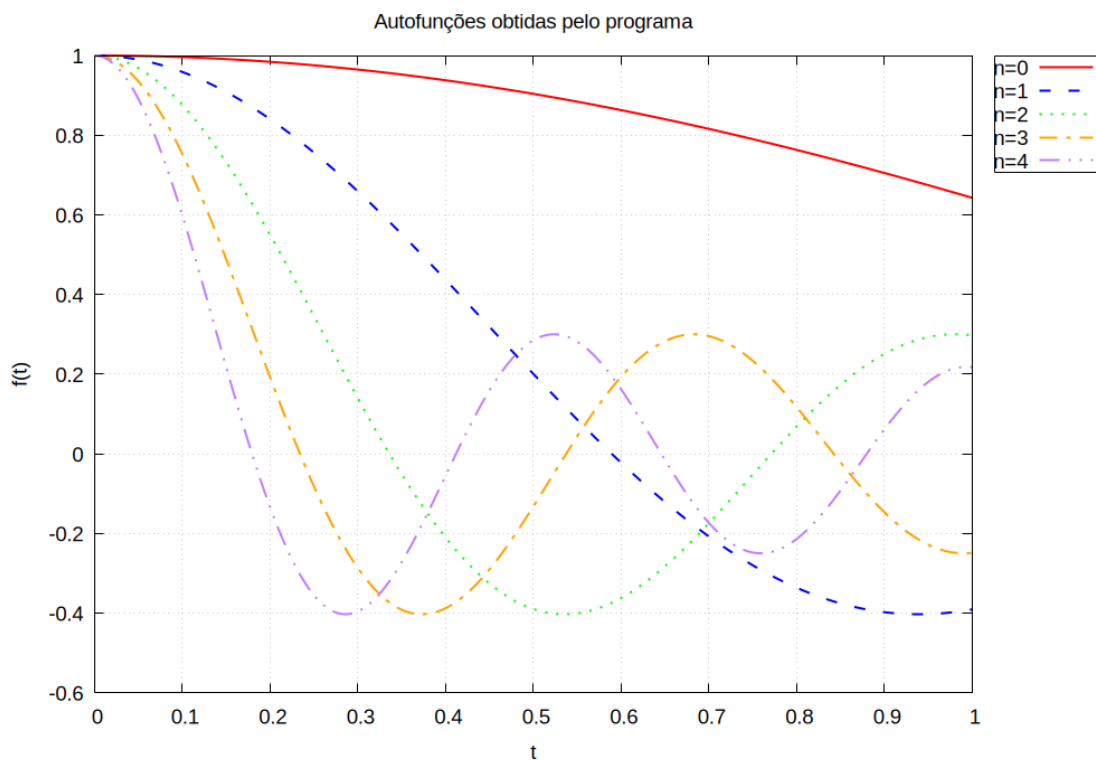


Figura 1.7: Autofunções determinadas numericamente pelo programa

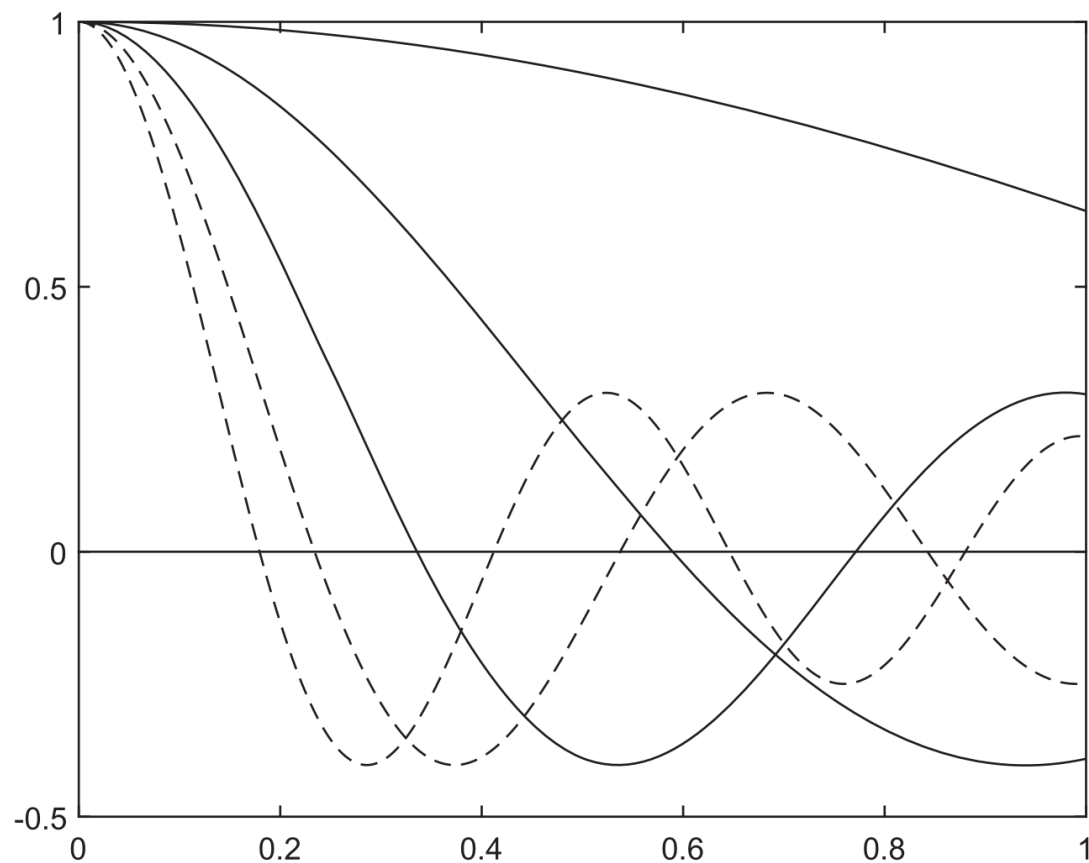


Figura 1.8: Gráficos das autofunções como mostrado no livro [2]

**Exemplo (E.1.5).** Consideremos o problema de Sturm-Liouville singular de tipo 1

$$\begin{cases} -((\sin x)y')' + xy = \lambda(\cos x)y, & 0 < x < 1, \\ |y(0)| < \infty, & y(1) = 0. \end{cases}$$

Numericamente, obtemos os seguintes autovalores:

$n$	$\lambda_n$	Intervalos buscados	Iterações da bissecção
0	1.63556063	164	24
1	8.31310754	822	24
2	20.27440115	1000	26
3	37.55132388	1144	27
4	60.14601837	1275	28

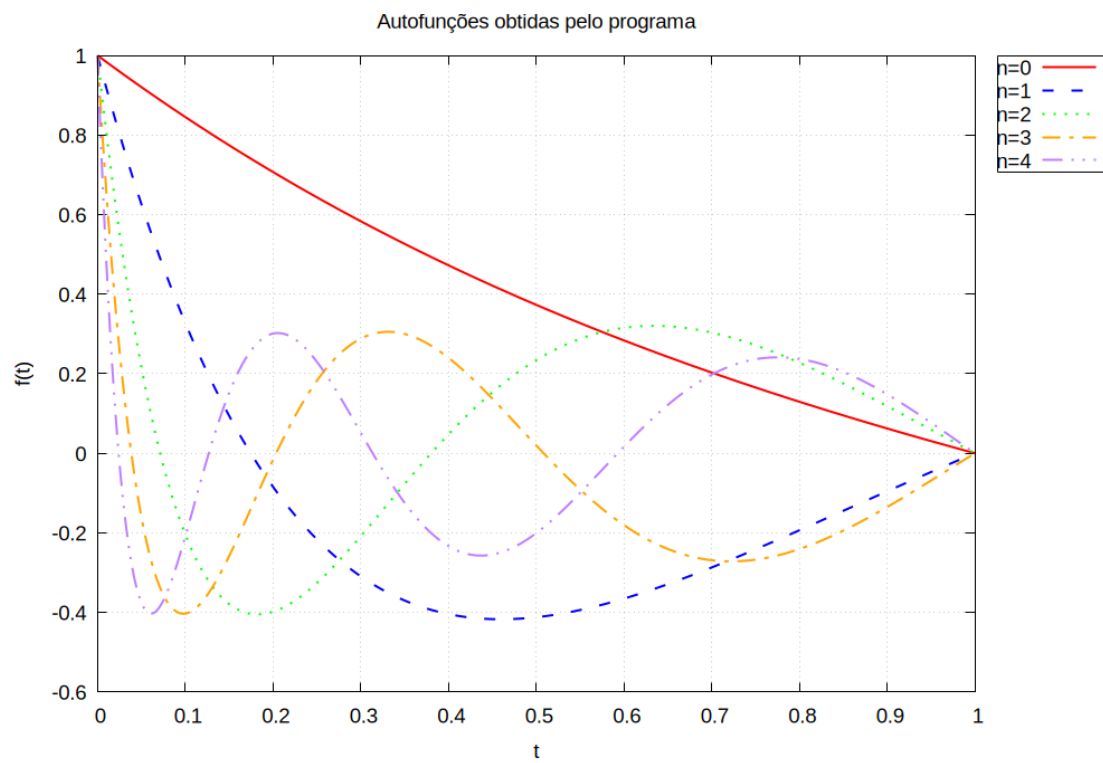


Figura 1.9: Autofunções determinadas numericamente pelo programa

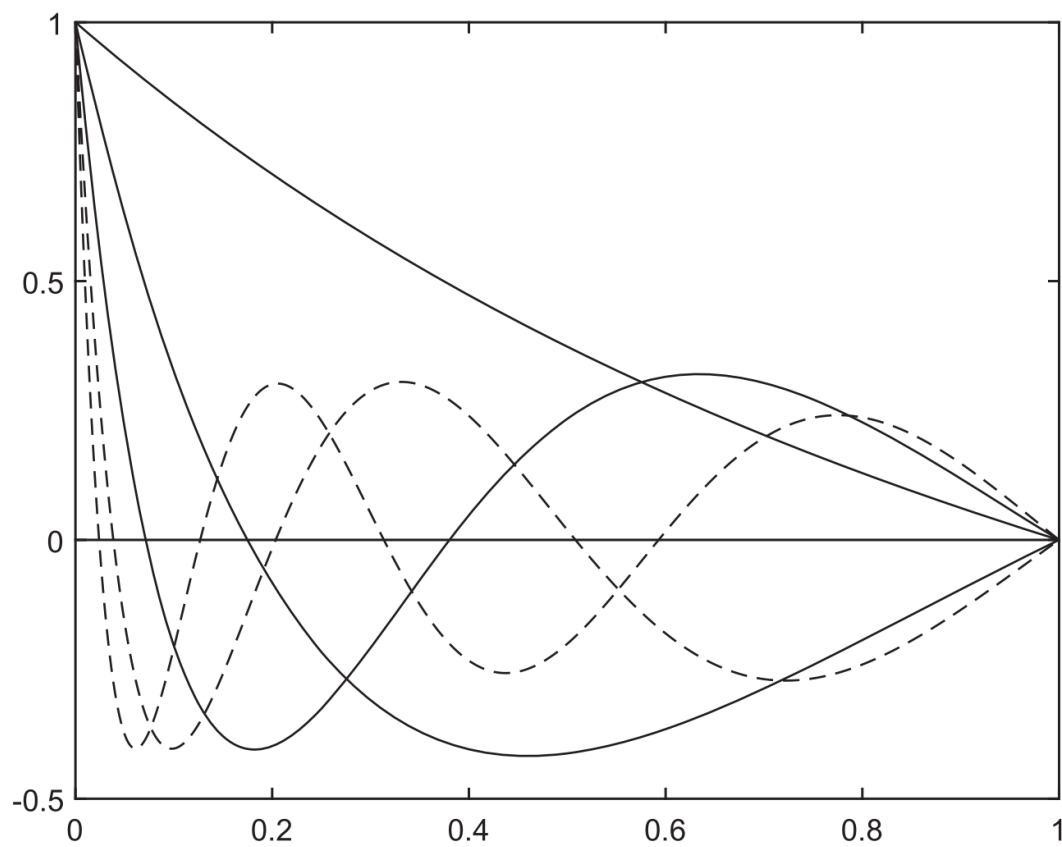


Figura 1.10: Gráficos das autofunções como mostrado no livro [2]

**Exemplo (E.1.6).** Consideremos o problema de Sturm-Liouville singular de tipo 1

$$\begin{cases} -(xy')' + \sin(\pi x)y = \lambda xy, & 0 < x < 1, \\ |y(0)| < \infty, & y'(1) = 0. \end{cases}$$

Numericamente, obtemos os seguintes autovalores:

$n$	$\lambda_n$	Intervalos buscados	Iterações da bissecção
0	1.22123592	123	24
1	16.63599690	1655	24
2	51.09474141	1878	28
3	105.36250987	2036	29
4	179.37909191	2173	30

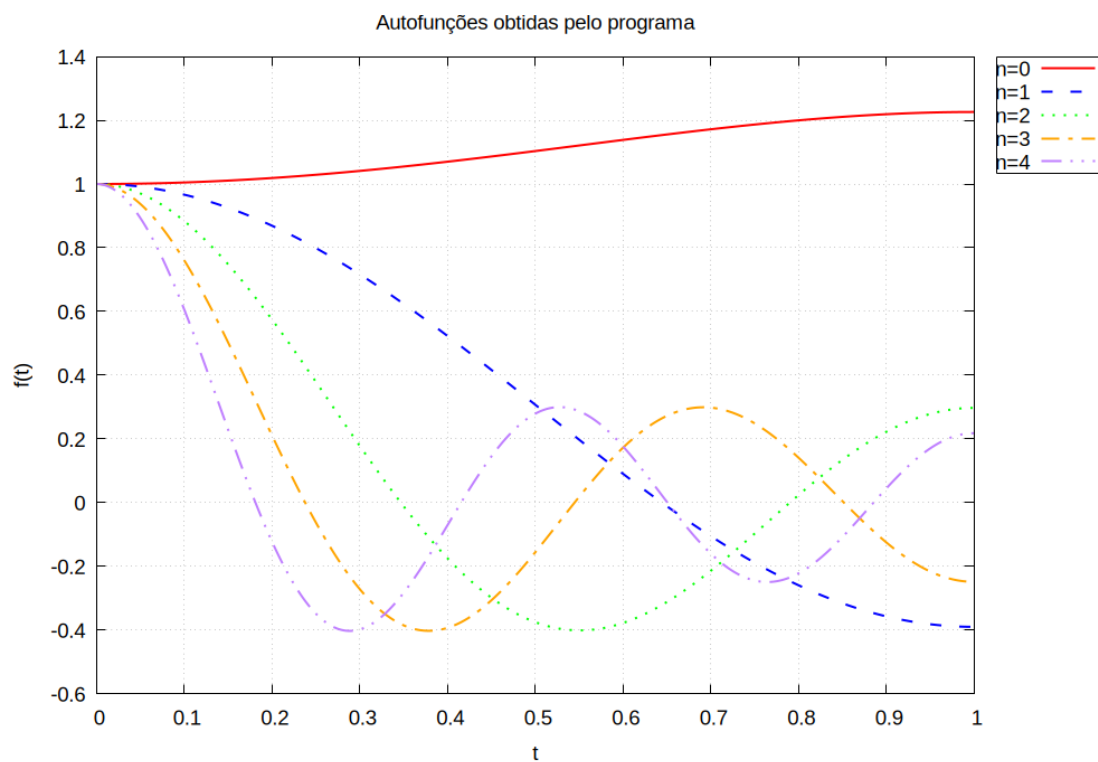


Figura 1.11: Autofunções determinadas numericamente pelo programa

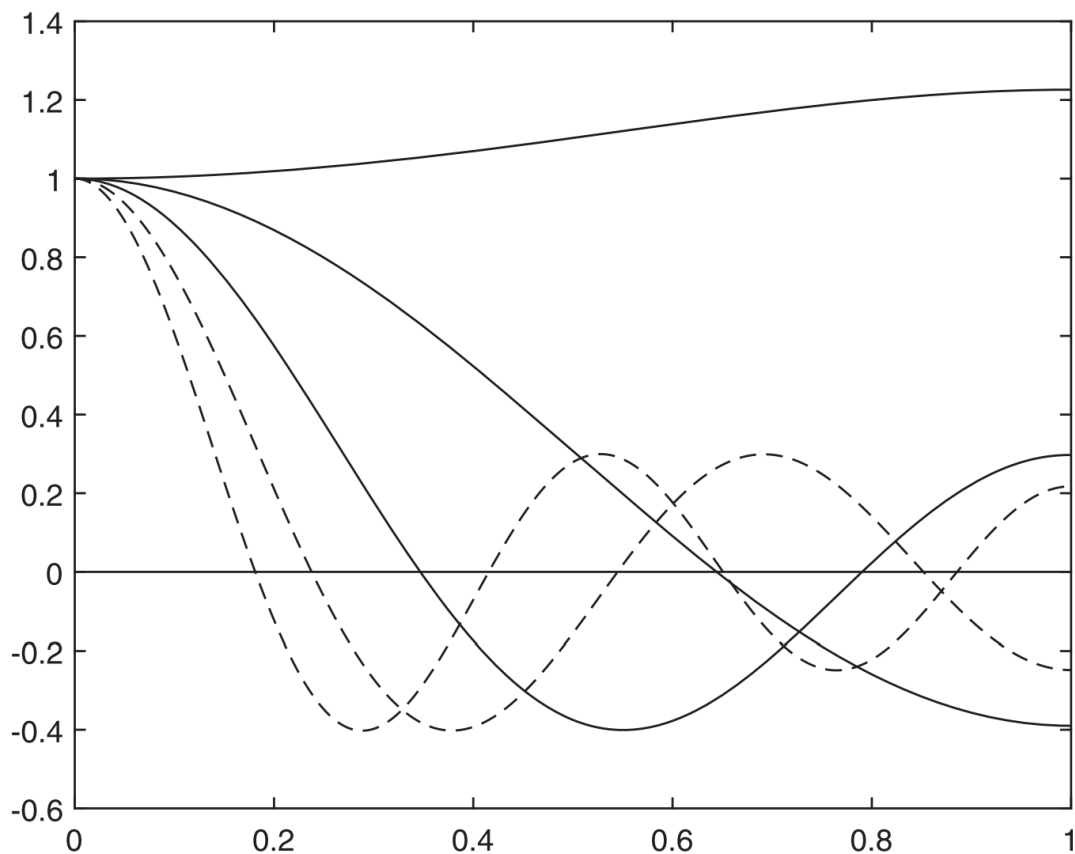


Figura 1.12: Gráficos das autofunções como mostrado no livro [2]

**Exemplo (E.1.7).** Consideremos o problema de Sturm-Liouville singular de tipo 2

$$\begin{cases} -(xy')' + \frac{1}{x}y = \lambda xy, & 0 < x < 1, \\ |y(0)| < \infty, & y(1) + y'(1) = 0. \end{cases}$$

Numericamente, obtemos os seguintes autovalores:

$n$	$\lambda_n$	Intervalos buscados	Iterações da bissecção
0	5.78318596	579	24
1	30.47126234	3038	24
2	74.88700679	3218	28
3	139.04028443	3363	29
4	222.93230362	3494	30

É interessante observarmos aqui que para encontrarmos os autovalores, precisamos baixar a tolerância do resíduo,  $F$ , de  $10^{-6}$  para  $10^{-3}$ , caso contrário o programa descarta autovalores verdadeiros como espúrios. Também foi necessário aumentar significativamente o número de passos de integração para determinarmos as autofunções, além de diminuirmos o valor de  $\varepsilon$  ao integramos partindo de  $x = a + \varepsilon$ .



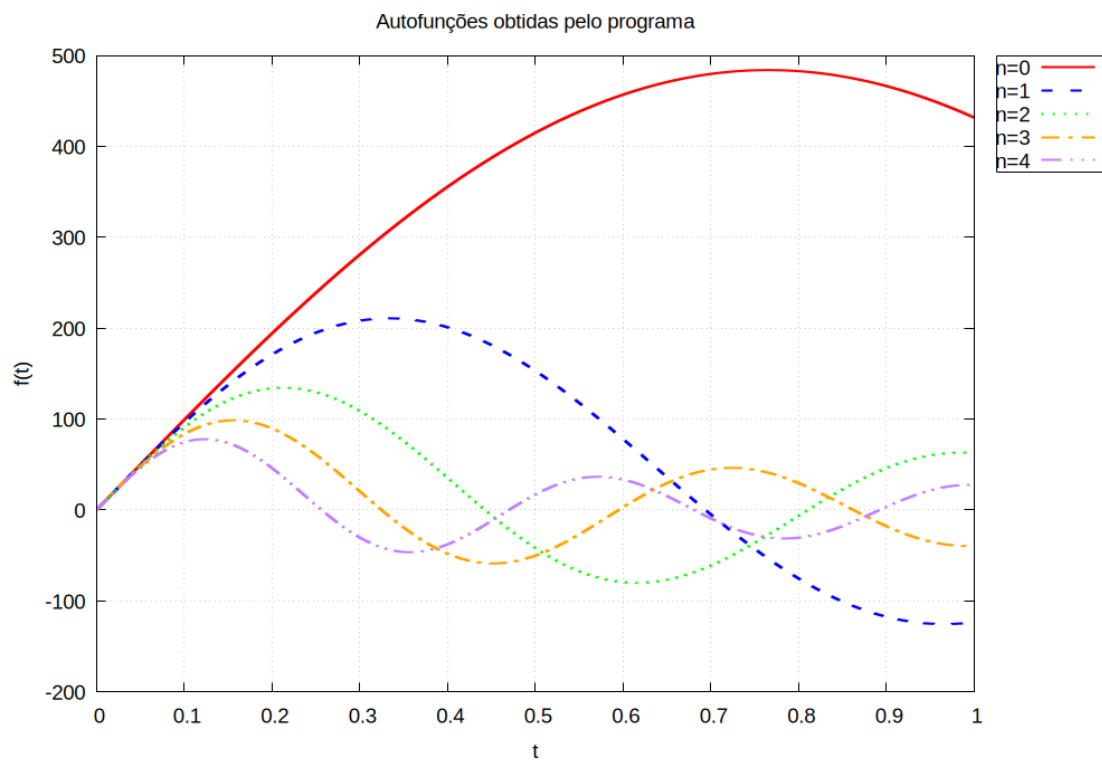


Figura 1.13: Autofunções determinadas numericamente pelo programa

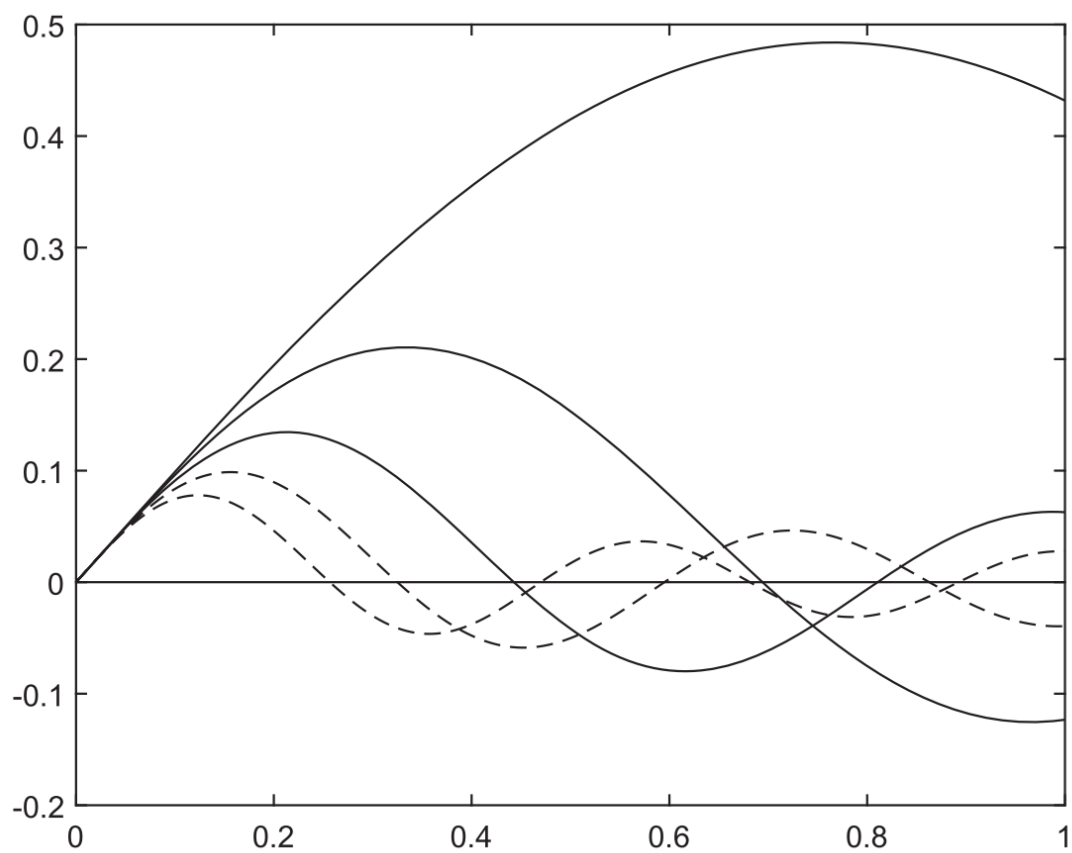


Figura 1.14: Gráficos das autofunções como mostrado no livro [2]

**Exemplo (E.1.8).** Consideremos o problema de Sturm-Liouville singular de tipo 2

$$\begin{cases} -(xy')' + \frac{\cos(x)}{4x}y = \lambda(\sin x)y, & 0 < x < 1, \\ |y(0)| < \infty, & y(1) = 0. \end{cases}$$

Numericamente, obtemos os seguintes autovalores:

$n$	$\lambda_n$	Intervalos buscados	Iterações da bissecção
0	10.21624703	1022	24
1	41.53026555	4144	24
2	93.72409193	4311	29
3	166.79600338	4451	29
4	260.74576040	4580	30

Tabela 1.1: Resultados para o segundo conjunto de autovalores

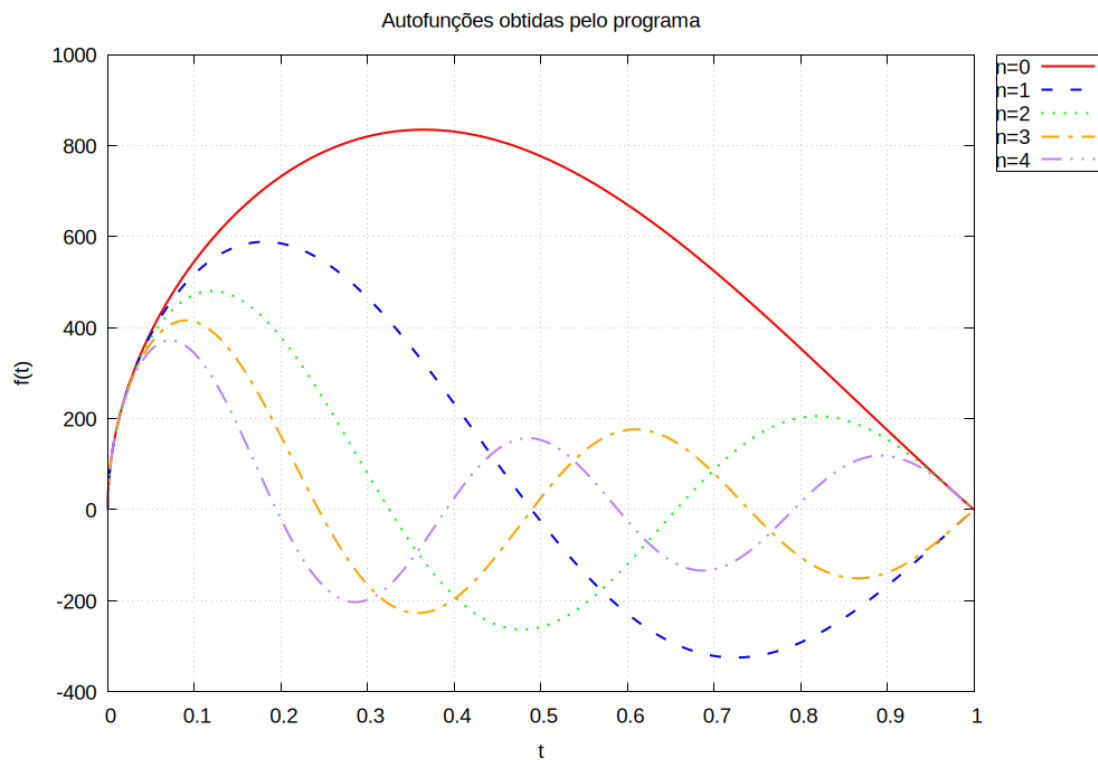


Figura 1.15: Autofunções determinadas numericamente pelo programa

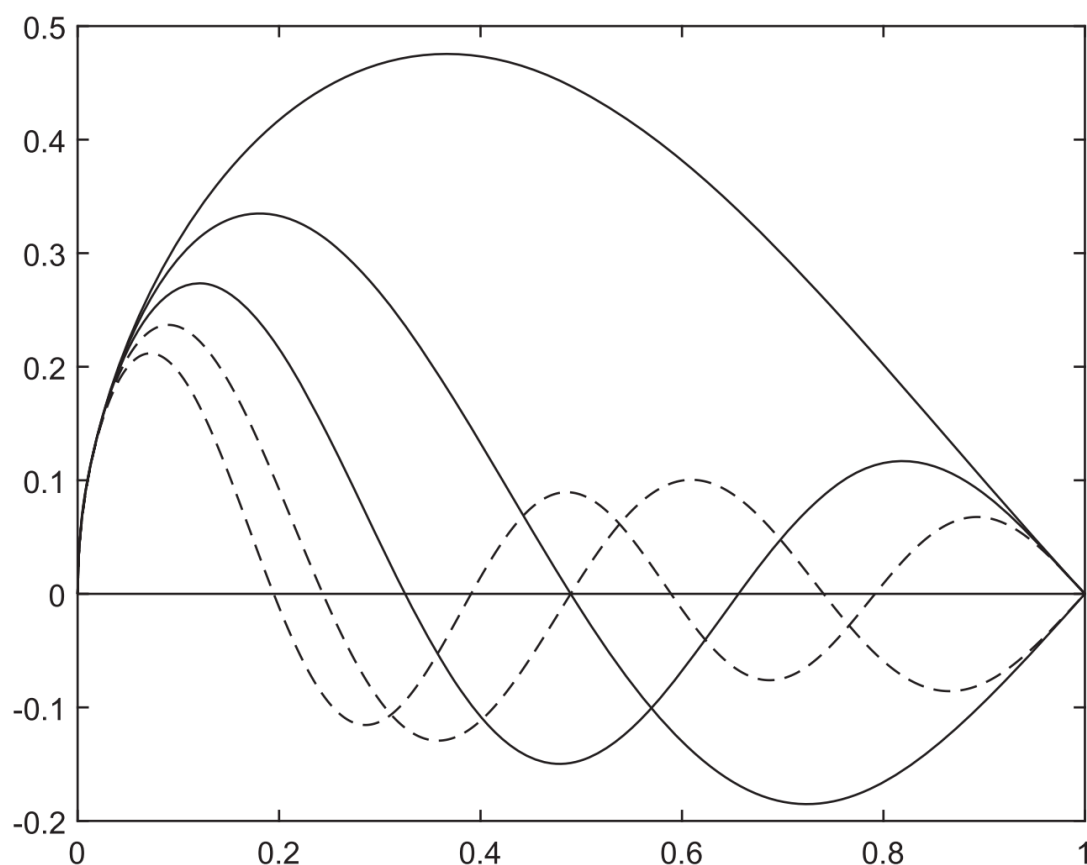


Figura 1.16: Gráficos das autofunções como mostrado no livro [2]

**Exemplo (E.1.9).** A equação de Legendre pode ser vista como um problema de Sturm-Liouville singular:

$$\begin{cases} -\frac{d}{dt} \left( (1-t^2) \frac{dP_n}{dt} \right) = \lambda P_n, & \forall t \in [-1, 1], \\ P(-1) = (-1)^n, & P(1) = 1. \end{cases}$$

Sabe-se que os autovalores são dados por  $\lambda_n = n(n+1)$ . As autofunções correspondentes de  $n=0$  até  $n=4$  são dadas por

n	Polinômio de Legendre
0	1
1	$x$
2	$\frac{1}{2}(3x^2 - 1)$
3	$\frac{1}{2}(5x^3 - 3x)$
4	$\frac{1}{8}(35x^4 - 30x^2 + 3)$

Numericamente, obtemos os seguintes autovalores:

$n$	$\lambda_n$	Intervalos buscados	Iterações da bissecção
0	0.00007518	1	24
1	2.00022555	192	24
2	6.00037594	387	25
3	12.00052635	535	26
4	20.00067675	667	26

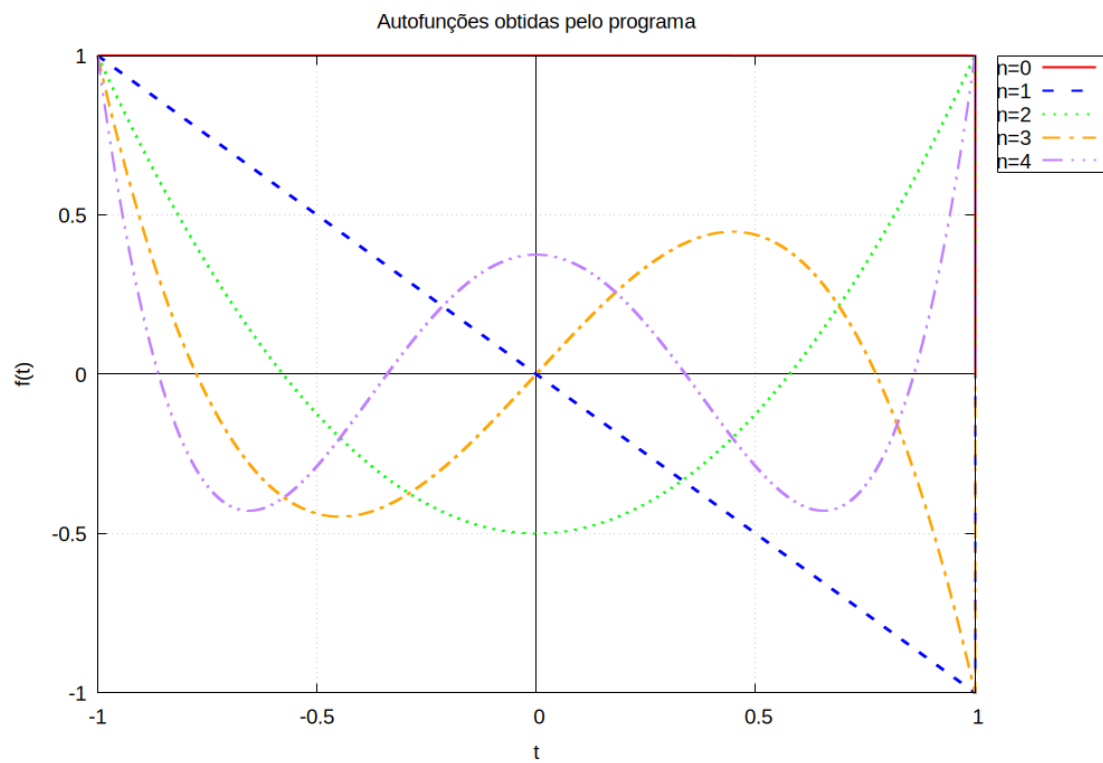


Figura 1.17: Autofunções determinadas numericamente pelo programa

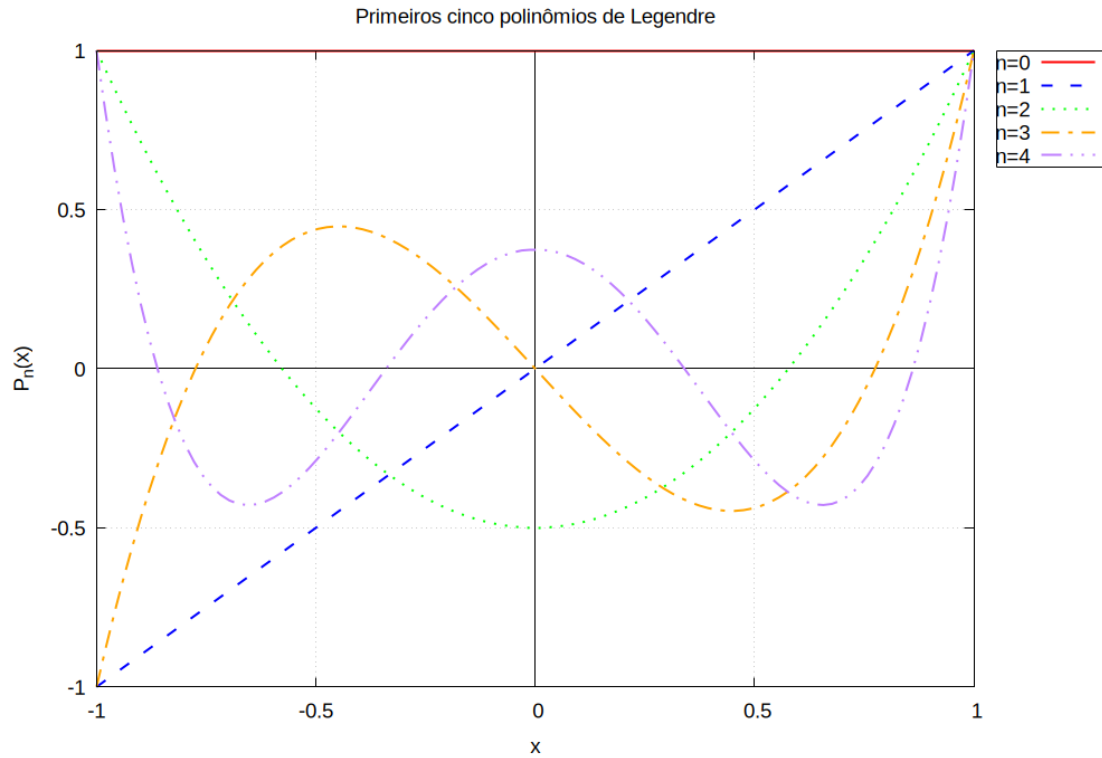


Figura 1.18: Gráficos dos polinômios de Legendre exatos

Observamos que os polinômios de grau ímpar determinados numericamente diferem pela constante multiplicativa  $-1$  dos polinômios de Legendre exatos correspondentes, o que é esperado, já que os polinômios de Legendre satisfazem  $P_n(-1) = (-1)^n$  e  $P_n(1) = 1$ , enquanto que no nosso programa forçamos  $P_n(1) = 1$  sempre.

## 1.5 Resultados numéricos do método de Newton-Raphson

Como mencionamos anteriormente, a abordagem marchante usando o método da bissecção enfrenta grandes dificuldades para ser adaptada com o método de Newton-Raphson. No programa que desenvolvemos, implementamos o método de Newton-Raphson para encontrar um autovalor por vez, com um palpite inicial bom. A implementação é bem mais simples que a do método do atirador marchante, mas não consegue sistematicamente determinar o espectro como anteriormente.

**Exemplo (E.1.10).** Para o problema

$$\begin{cases} -(1 \cdot u'(x))' + (x - \lambda \cos(x))u(x) = 0, & \forall x \in [0, 1], \\ u'(0) = 0, & u'(1) = 0. \end{cases} \quad (1.17)$$

e com o chute inicial  $\lambda_0 = 0$ , Newton-Raphson convergiu em somente 4 iterações:

Iteração	$\lambda$	$F(\lambda)$
1	0.53761681266555983	-0.53403483428546394
2	0.57802783474609609	-0.034945050888573075
3	0.57824837086275171	-0.00018864919931886801
4	0.57824837740937718	-0.000000055997283388719053

Já com o chute inicial  $\lambda_0 = 140$ , Newton-Raphson convergiu em 5 iterações:

Iteração	$\lambda$	$F(\lambda)$
1	60.773415429648637	-9.8908269771386994
2	46.353458505247033	4.6605507929687588
3	48.528804125931671	-0.83760614199634920
4	48.489185568458076	0.015769242002305423
5	48.489177842533614	0.0000030739212154774114

É um pouco surpreendente que o chute inicial  $\lambda_0 = 140$ , mais próximo ao autovalor teórico

$$\lambda_3 = 107.57497472$$

tenha convergido para  $\lambda_2$  ao invés de  $\lambda_3$ .

## 1.6 Análise da ordem do método do tiro com Newton-Raphson

Fizemos uma análise da ordem do método do tiro com Newton-Raphson. Configuramos a tolerância do resíduo  $F(\lambda)$  para  $10^{-16}$  e usamos RK4 novamente. A análise foi feita usando o problema de Dirichlet:

$$\begin{cases} y'' + \lambda y = 0, \forall x \in [0, 1]; \\ y(0) = y(1) = 0. \end{cases}$$

com palpite inicial  $\lambda = 10$ , e o ponto avaliado foi  $x_0 = 0.2$ . Observamos também que quanto mais perto de 0 tomamos  $x_0$

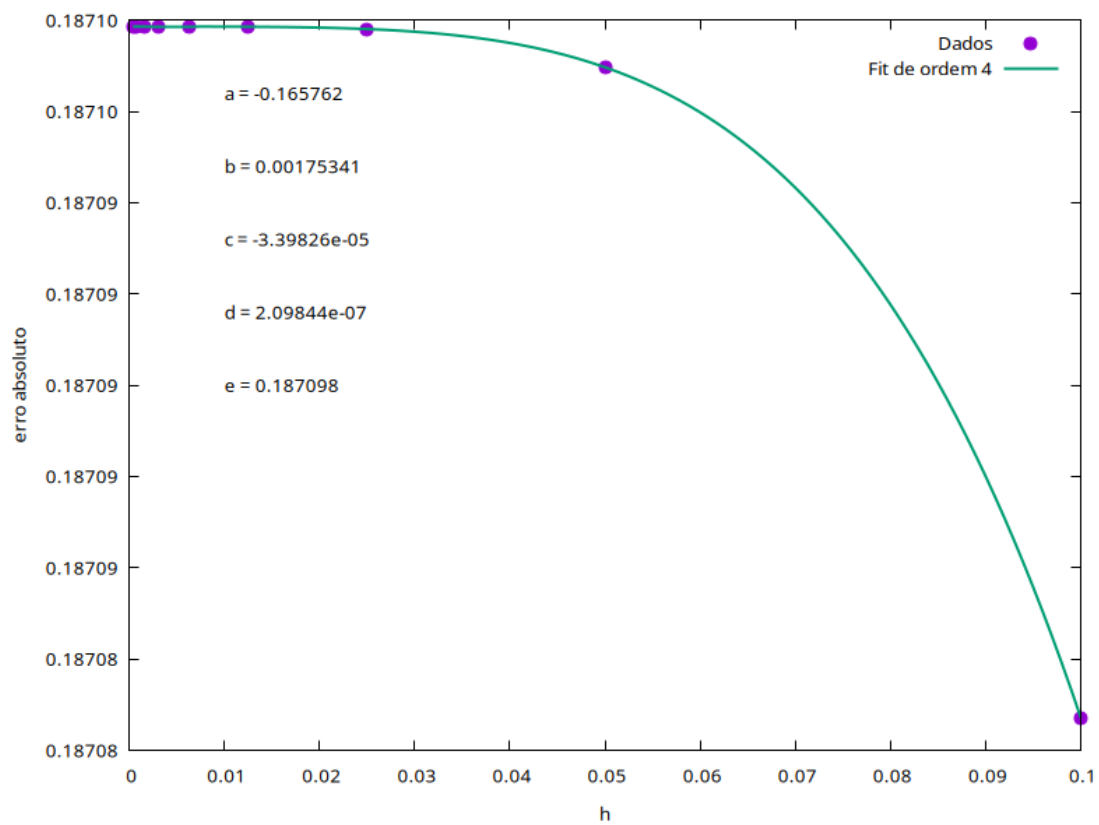


Figura 1.19: Gráfico de convergência,  $h$  vs  $y(0.2)$ . Note que o coeficiente  $e$  é de fato aproximadamente  $y_0(0.2)$ .

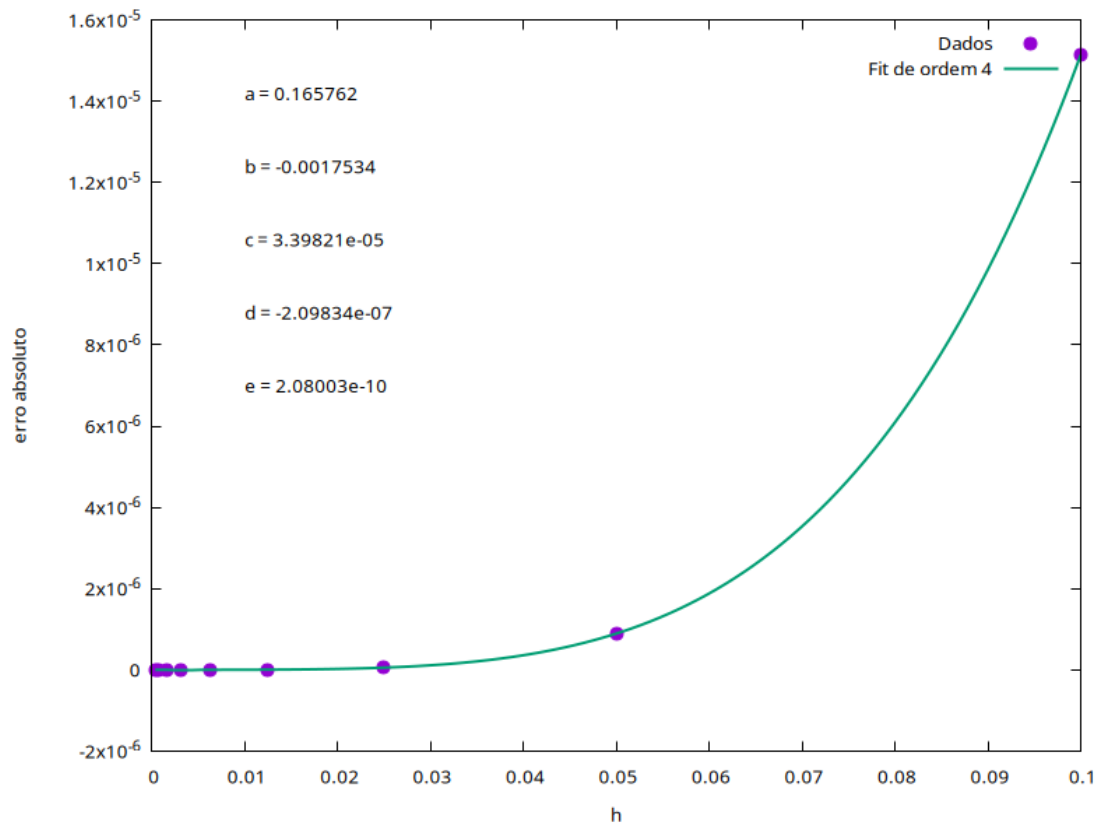


Figura 1.20: Fit do erro de truncamento local.

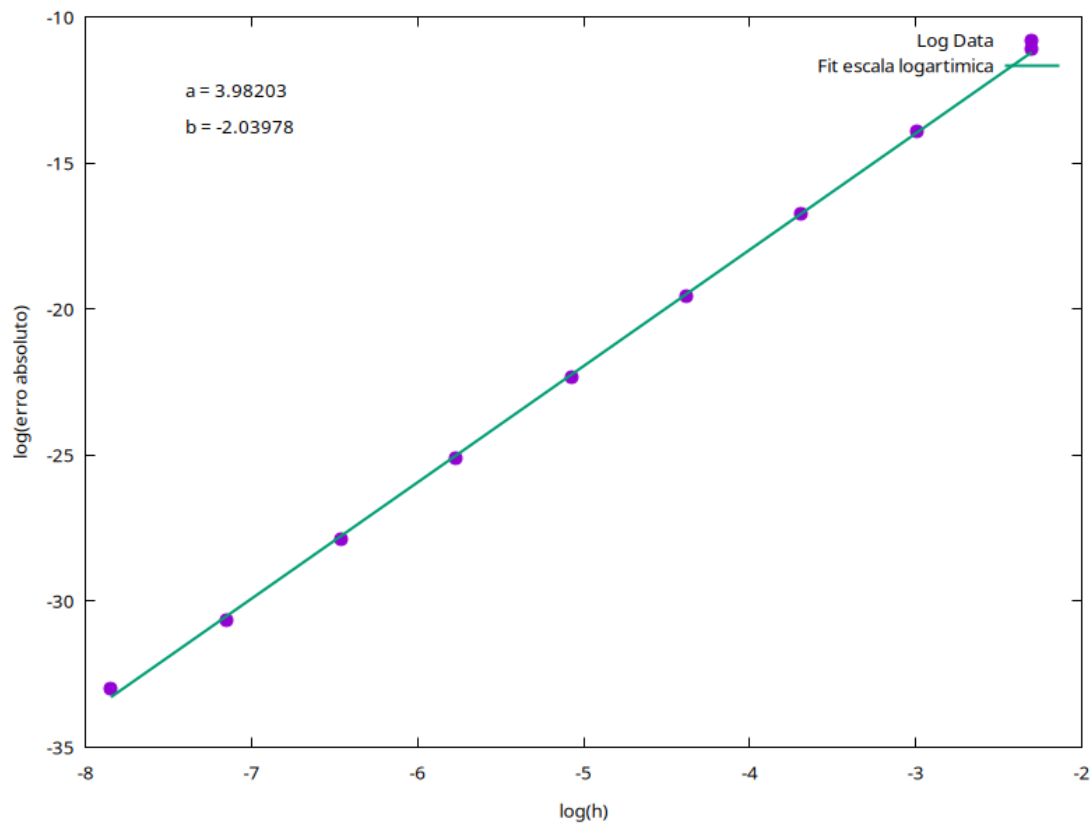


Figura 1.21: Fit linear de  $\log(h)$  vs  $\log$  do erro de truncamento local. Note que a inclinação é de fato aproximadamente 4.



---

# Bibliografia

---

- [1] **Kac, Marc.** “Can One Hear the Shape of a Drum?”. *The American Mathematical Monthly*, vol. 73, no. 4, 1966, pp. 1–23. *JSTOR*, <https://doi.org/10.2307/2313748>.
- [2] **Guenther, R.B; Lee, J.W.** (2019) *Sturm-Liouville problems: Theory and numerical implementation.*, Boca Raton: CRC Press, Taylor & Francis Group.
- [3] <https://math.stackexchange.com/questions/1915313/what-is-the-motivation-for-the-equation-of-the-sturm-liouville-problem>.
- [4] <https://math.stackexchange.com/questions/2268037/can-sturm-liouville-theory-actually-solve-odes>.
- [5] <https://physics.stackexchange.com/questions/176041/the-sturm-liouville-equations-the-schrodinger-equation-and-the-wave-equation>.
- [6] [https://github.com/SaganGromov/SL\\_REGULAR\\_EXEMPLOS](https://github.com/SaganGromov/SL_REGULAR_EXEMPLOS).
- [7] [https://github.com/SaganGromov/SL\\_SINGULAR\\_EXEMPLOS](https://github.com/SaganGromov/SL_SINGULAR_EXEMPLOS).
- [8] [https://github.com/SaganGromov/SL\\_NEWTON\\_RAPHSON](https://github.com/SaganGromov/SL_NEWTON_RAPHSON).