Universidade Federal de Alagoas

Instituto de Matemática

Método de Picard: Implementação Computacional

Experimento 2.6 — Equações Diferenciais Ordinárias

Victor Gabriel Santos de Moura

Bacharelando em Matemática - IM/UFAL

Maceió, 2026

Sumário

1 Fundamentação Teórica **3**
1.1 Notação de Landau 3
1.2 O problema de valor inicial e o operador de Picard 3
1.3 Existência e unicidade 3
1.4 Estimativa quantitativa e cota do erro 4
1.5 Discretização e erro de quadratura 6

2 Objetivos do Experimento **6**

3 Objetivo 1 **7**
3.1 Configuração 7
3.2 Implementação 8

4 Objetivo 2 **9**
4.1 Configuração 10
4.2 Implementação das duas regras 10
4.3 Tabela de erros 12

5 Objetivo 3 **12**
5.1 Configuração 13
5.2 Integração 13
5.3 Piso irredutível do retângulo no extremo esquerdo 14
5.4 Tabela de erros 15

6 Objetivo 4 **16**
6.1 Configuração 16
6.2 Redução de complexidade: $\mathcal{O}(NL^2)$ para $\mathcal{O}(NL)$ 16
6.3 Análise teórica completa para $F(t,x) = x^2$ 18
6.4 Comparação entre os domínios 20

7 Objetivo 5 **21**
7.1 Formulação do problema 21
7.2 Configuração 22
7.3 Adaptações para $d = 2$ 22
7.4 Análise para o sistema bidimensional 24

8 Comparação dos códigos **27**
8.1 Evolução dos algoritmos 27
8.2 Erro de iteração e erro de convergência 28
8.3 Importância de $C(\delta)$ na estruturação do algoritmo 28

9 Conclusões **28**

Referências **29**

1 Fundamentação Teórica

Esta seção fixa a notação e apresenta os resultados teóricos que sustentam toda a análise computacional do experimento, seguindo [1], Capítulo 2.

1.1 Notação de Landau

Dadas funções $f, g : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$, escrevemos $f(h) = \mathcal{O}\big(g(h)\big)$ quando $h \to 0$ se existe $C > 0$ tal que $f(h) \leq C\, g(h)$ para h suficientemente pequeno. Usaremos esta notação exclusivamente para descrever o comportamento dos erros de quadratura conforme o passo $h \to 0$ (equivalentemente, $L \to \infty$ com ε fixo).

1.2 O problema de valor inicial e o operador de Picard

Considera-se o problema de valor inicial

$$x' = F(t,\, x), \qquad x(t_0) = x_0, \tag{1}$$

onde $F : \mathcal{U} \to \mathbb{R}^d$ é contínua num aberto $\mathcal{U} \subset \mathbb{R} \times \mathbb{R}^d$ e localmente Lipschitziana em x. A ideia central do método de Picard é reescrever (1) como equação integral e aplicar o Teorema do Ponto Fixo para Contrações.

Definição 1.1 (Operador de Picard)**.** Dado $\varepsilon > 0$, o *operador de Picard* $\mathcal{L} : Y \to Y$ age em funções contínuas $\gamma : (t_0 - \varepsilon, t_0 + \varepsilon) \to \mathbb{R}^d$, com $\gamma(t_0) = x_0$, por

$$(\mathcal{L}\gamma)(t) \;\coloneqq\; x_0 + \int_{t_0}^{t} F\big(s,\, \gamma(s)\big)\, ds. \tag{2}$$

A sequência de *iterados de Picard* é definida por $\gamma_0 \equiv x_0$ e $\gamma_{n+1} \coloneqq \mathcal{L}\gamma_n$ para $n \geq 0$.

1.3 Existência e unicidade

O resultado de existência e unicidade local, sobre o qual todo o experimento se apoia, é o seguinte.

Teorema 1.2 (Existência e Unicidade, [1] Teorema 2.4)**.** *Suponha que $F : \mathcal{U} \to \mathbb{R}^d$ é contínua e localmente Lipschitziana em x. Então:*

1. *para todo $(t_0, x_0) \in \mathcal{U}$, existem algum intervalo aberto I e alguma solução $\gamma : I \to \mathbb{R}^d$ de (1) tais que $t_0 \in I$ e $\gamma(t_0) = x_0$;*
2. *se $\gamma_1 : I_1 \to \mathbb{R}^d$ e $\gamma_2 : I_2 \to \mathbb{R}^d$ são soluções de (1) e existe $t_0 \in I_1 \cap I_2$ tal que $\gamma_1(t_0) = \gamma_2(t_0)$, então $\gamma_1(t) = \gamma_2(t)$ para todo $t \in I_1 \cap I_2$.*

Nas condições do item (1), dizemos que γ é uma solução de (1) *com condição inicial* $\gamma(t_0) = x_0$. O item (2) afirma que duas soluções que coincidem num ponto coincidem em toda a interseção de seus domínios.

1.4 Estimativa quantitativa e cota do erro

O Teorema 1.2 garante existência e unicidade, mas não fornece uma estimativa explícita para o tamanho do intervalo de definição da solução nem para a taxa de convergência dos iterados. Esses dois pontos são tratados pelo refinamento a seguir ([1], Teorema 2.12).

Teorema 1.3 ([1] Teorema 2.12)**.** *Dado $(t_0, x_0) \in \mathcal{U}$, fixe $\delta > 0$ tal que $\overline{B}_\delta(t_0) \times \overline{B}_\delta(x_0) \subset \mathcal{U}$. Defina*

$$M(\delta) \;\coloneqq\; \sup\Big\{ \|F(t,x)\| : |t - t_0| \le \delta,\ \|x - x_0\| \le \delta \Big\} \tag{3}$$

e seja $C(\delta)$ a constante de Lipschitz de F em x sobre essa região. Para todo $\varepsilon > 0$ satisfazendo

$$\varepsilon \;\le\; \varepsilon(\delta) \;\coloneqq\; \min\left\{ \delta,\ \frac{\delta}{M(\delta)} \right\}, \tag{4}$$

existe solução única $x : (t_0 - \varepsilon,\, t_0 + \varepsilon) \to \mathbb{R}^d$ de (1), e os iterados de Picard convergem uniformemente para ela nesse intervalo.

A condição (4) garante que o operador de Picard leva Y em si mesmo e que $\mathcal{L}^\kappa$ é uma contração para algum $\kappa \ge 1$, conforme estabelecido pelo Lema 1.4 abaixo. A convergência segue então do Teorema do Ponto Fixo para Contrações.

Lema 1.4 ([1] Lema 2.13)**.** *Nas condições do Teorema 1.3, para todo $n \ge 1$ e quaisquer $\gamma_1, \gamma_2 \in Y$,*

$$\|\mathcal{L}^n(\gamma_1)(t) - \mathcal{L}^n(\gamma_2)(t)\| \;\le\; \frac{\big(C(\delta)\big)^n\ |t - t_0|^n}{n!}\, d(\gamma_1, \gamma_2), \qquad |t - t_0| \le \varepsilon, \tag{5}$$

onde $d(\gamma_1, \gamma_2) = \sup_{|t-t_0|\le\varepsilon} \|\gamma_1(t) - \gamma_2(t)\|$. Em particular, existe $\kappa \ge 1$ tal que $\mathcal{L}^\kappa$ é uma contração em Y.

O resultado central para toda a análise numérica do experimento é o seguinte corolário, que quantifica a velocidade de convergência dos iterados.

Corolário 1.5 (Cota do erro, [1] Corolário 2.14)**.** *Sob as condições do Teorema 1.3,*

$$\|\gamma_n - x\|_\infty \;\le\; M(\delta)\ \frac{\big(C(\delta)\,\varepsilon\big)^n}{n!} \qquad \forall\ n \ge 0, \tag{6}$$

onde x é a solução exata de (1) e $\|\cdot\|_\infty \coloneqq \sup_{|t-t_0|\le\varepsilon} \|\cdot\|$.

Demonstração. Aplicamos o Lema 1.4 com $\gamma_1 = \gamma_0 \equiv x_0$ e $\gamma_2 = x$. Como x é ponto fixo de $\mathcal{L}$, tem-se $\mathcal{L}^n(x) = x$ para todo $n \geq 0$, logo $\gamma_n = \mathcal{L}^n(\gamma_0)$ e

$$\|\gamma_n(t) - x(t)\| = \|\mathcal{L}^n(\gamma_0)(t) - \mathcal{L}^n(x)(t)\| \leq \frac{C(\delta)^n \ |t - t_0|^n}{n!} \, d(\gamma_0, \, x). \tag{7}$$

Como $\gamma_0 \equiv x_0$ e x satisfaz a equação integral $x(t) = x_0 + \int_{t_0}^{t} F(s, x(s))\, ds$, temos

$$\|x_0 - x(t)\| = \left\| \int_{t_0}^{t} F(s, x(s))\, ds \right\| \leq M(\delta) \ |t - t_0| \leq M(\delta)\, \varepsilon,$$

logo $d(\gamma_0, x) \leq M(\delta)\, \varepsilon$. Substituindo em (7) e tomando o supremo em t:

$$\|\gamma_n - x\|_\infty \leq \frac{C(\delta)^n \, \varepsilon^n}{n!} \cdot M(\delta)\, \varepsilon = M(\delta) \, \frac{(C(\delta)\, \varepsilon)^n \cdot \varepsilon}{n!}.$$

Por fim, a condição (4) implica $\varepsilon \leq \delta / M(\delta)$, isto é, $M(\delta)\, \varepsilon \leq \delta$. Como $\varepsilon \leq \delta$ e podemos sempre tomar $M(\delta) \geq 1$ (pois F é não-nula em geral), majoramos o fator ε extra usando $M(\delta)\, \varepsilon \leq M(\delta)$, obtendo

$$\|\gamma_n - x\|_\infty \ \leq \ M(\delta) \ \frac{\Big(C(\delta)\, \varepsilon\Big)^n}{n!},$$

que é (6). □

A equação (6) conecta os parâmetros M, C e ε ao número mínimo de iterados necessários para atingir uma tolerância τ:

$$N_{\max}(\tau) \ \coloneqq \ \min\left\{ n \in \mathbb{N} : M(\delta) \, \frac{\Big(C(\delta)\, \varepsilon\Big)^n}{n!} < \tau \right\}. \tag{8}$$

Observação 1.6. O comportamento do lado direito de (6) em função de n depende crucialmente do produto $C(\delta)\, \varepsilon$:

- Se $C(\delta)\, \varepsilon \leq 1$: o fatorial domina a potência desde o primeiro iterado, e a cota decresce monotonamente em n.
- Se $C(\delta)\, \varepsilon > 1$: o termo $(C\varepsilon)^n / n!$ cresce até atingir um máximo em torno de $n \approx C(\delta)\, \varepsilon$ e só então decai via o fatorial. A convergência da sequência é garantida pelo Teorema 1.3, mas $N_{\max}$ pode ser significativamente maior do que no caso $C\varepsilon \leq 1$.

Em particular, o fatorial domina qualquer potência fixa, de modo que a cota (6) tende a zero para todo valor de $C(\delta)\, \varepsilon$ — apenas a taxa varia. Esta distinção é relevante nos Objetivos 4 e 5, onde ε_{num} excede ε^*.

1.5 Discretização e erro de quadratura

Computacionalmente, a curva γ_n é representada por L valores em pontos igualmente espaçados

$$t_k = t_0 - \varepsilon + \frac{k}{L-1} \cdot 2\varepsilon, \quad k = 0, 1, \ldots, L-1, \qquad h \coloneqq \frac{2\varepsilon}{L-1}. \tag{9}$$

Para que $t_0 = 0$ coincida com um índice da malha, é necessário que L seja ímpar; nesse caso $i_0 \coloneqq \lfloor L/2 \rfloor$ satisfaz $t_{i_0} = 0$.

A integral em (2) é aproximada por duas regras clássicas:

$$\text{(Trapézio)} \quad \int_{t_0}^{t_k} f\, ds \;\approx\; \sum_{j=i_0+1}^{k} \frac{f_{j-1} + f_j}{2}\, h, \quad \text{erro global } \mathcal{O}(h^2), \tag{10}$$

$$\text{(Retângulo)} \quad \int_{t_0}^{t_k} f\, ds \;\approx\; \sum_{j=i_0+1}^{k} f_{j-1}\, h, \quad \text{erro global } \mathcal{O}(h), \tag{11}$$

onde o erro global é acumulado sobre $[t_0,\, t_0 + \varepsilon]$ e expresso em termos de $h = 2\varepsilon/(L-1)$. O erro total do n-ésimo iterado discreto γ_n^h satisfaz

$$\left\| \gamma_n^h - x \right\|_\infty \;\leq\; \underbrace{M(\delta)\, \frac{\big(C(\delta)\,\varepsilon\big)^n}{n!}}_{\text{erro de convergência}} \;+\; \underbrace{\mathcal{O}\big(\varepsilon^2 L^{-p}\big)}_{\text{erro de quadratura}}, \tag{12}$$

com $p = 2$ para o trapézio e $p = 1$ para o retângulo. Aumentar n reduz o primeiro termo via o fatorial; aumentar L reduz o segundo como L^{-p}. Para n fixo e suficientemente grande, o retângulo atinge um *piso de erro irredutível* proporcional a ε^2/L, que não pode ser eliminado aumentando o número de iterados — apenas aumentando L ou diminuindo ε.

2 Objetivos do Experimento

O experimento está organizado em cinco objetivos progressivos, correspondentes às instruções da Seção 2.6 de [1]:

1. Escreva o método de Picard em código computacional.
2. Considere a equação diferencial $x' = x^2$ com condição inicial $x(0) = 1$. Calcule os primeiros iterados de Picard num domínio $[0, \varepsilon)$ adequado. Compare com a solução exata $x(t) = 1/(1-t)$ e estime o erro do n-ésimo iterado de Picard. Represente os resultados graficamente.
3. Utilize este método para calcular soluções no domínio $(-\varepsilon, \varepsilon)$. Observe o comportamento do método de cálculo da integral (regra do retângulo, fórmula do

trapézio) no ponto extremo esquerdo da curva e explique o problema observado. Como resolvê-lo?

4. O método encontra soluções em intervalos maiores que $(-\varepsilon, \varepsilon)$? Investigue a convergência dos iterados de Picard em tais domínios maiores, inclusive em todo o intervalo $(-\infty, 1)$ de definição da solução.

5. Repita os passos (2) a (4) para a equação diferencial $x'' = -x + 2\sin t$ com condição inicial $x(0) = 1$, $x'(0) = 0$. A solução exata é $x(t) = (1-t)\cos t + \sin t$. Note que, neste caso, o problema tem dimensão $d = 2$.

Todos os códigos desenvolvidos no texto se basearam na linguagem de programação python e em suas bibliotecas NumPy e MatPlotLib.

3 Objetivo 1

A implementação do Objetivo 1 já incorpora a análise de parâmetros ótimos desenvolvida na Seção 1, antecipando a discussão que percorre todos os objetivos seguintes. O domínio $[0, \varepsilon^*]$ com $\varepsilon^* = 0{,}25$ é escolhido de modo a maximizar o raio garantido pelo Teorema 1.3 para $F(t, x) = x^2$; a justificativa quantitativa é dada na Seção 3.1 abaixo.

3.1 Configuração

Parâmetro	Descrição	Valor
ε	Raio do domínio $[0, \varepsilon]$	0,25
L	Pontos de discretização	500
N	Iterados de Picard	8
x_0	Condição inicial	1,0

Tabela 1: Parâmetros do Objetivo 1.

Justificativa para $\varepsilon = 0,25$: Considere $F(t, x) = x^2$ e $(t_0, x_0) = (0, 1)$, as quantidades de (3) sobre $[-\delta, \delta] \times [1-\delta,\, 1+\delta]$ são:

$$M(\delta) = (1+\delta)^2, \qquad C(\delta) = 2(1+\delta), \qquad \varepsilon(\delta) = \frac{\delta}{(1+\delta)^2}, \tag{13}$$

onde a última igualdade segue de $(1+\delta)^2 > 1$ para $\delta > 0$, que garante que o mínimo em (4) é sempre o segundo argumento. Maximizando $\varepsilon(\delta)$:

$$\frac{d}{d\delta}\left[\frac{\delta}{(1+\delta)^2}\right] = \frac{1-\delta}{(1+\delta)^3} = 0 \implies \delta^* = 1, \quad \varepsilon^* = \tfrac{1}{4} = 0{,}25, \quad M^* = C^* = 4. \tag{14}$$

Com $C(\delta^*)\,\varepsilon^* = 1$, a cota (6) reduz-se a $\|\gamma_n - x\|_\infty \leq 4/n!$, que para $n = 8$ fornece $4/8! \approx 9{,}9 \times 10^{-5}$. A convergência observada é tipicamente melhor do que esta cota superior, uma vez que (6) usa o supremo sobre toda a bola $B_{\delta^*}(x_0)$ e não apenas sobre a trajetória da solução.

3.2 Implementação

```
1  import numpy as np
2
3  eps = 0.25;  L = 500;  N = 8;  x0 = 1.0
4  t = np.linspace(0, eps, L)
5
6  def F(t, x):
7      return x**2
8
9  y = np.zeros((N+1, L))
10 y[0, :] = x0
11
12 for n in range(N):
13     for k in range(L):
14         integrando = F(t[:k+1], y[n, :k+1])
15         y[n+1, k]  = x0 + np.trapezoid(integrando, t[:k+1])
```

Listing 1: EDO1.py — implementação do operador (2).

Na linha 3 está a definição dos parâmetros; na linha 4 geramos a sequência de números t_k com espaçamento uniforme dentro do intervalo $[0, \varepsilon]$. Nas linhas 6 e 7, definimos a função $F(t, x)$ relacionada a EDO do objetivo 2. Na linha 9 estamos gerando uma matriz de tamanho $N + 1 \times L$, na linha 10 implementamos nossa condição inicial. A matriz `y[n, k]` armazena $\gamma_n(t_k)$. A linha 13 discretiza a recorrência da Definição 1.1: para cada t_k, integra $F(s, \gamma_n(s))$ de $t_0 = 0$ a t_k pela regra do trapézio sobre os $k + 1$ pontos calculados. Como os passos do código são construtivos, para melhor fluidez do texto e evitando repetições, nos próximos códigos discutiremos apenas os aspectos novos e as modificações. Todos os códigos completos dos experimentos estarão disponíveis no site do experimento e no GitHub do autor.
GitHub: experimento-picard
Site: https://victormoura22.github.io/

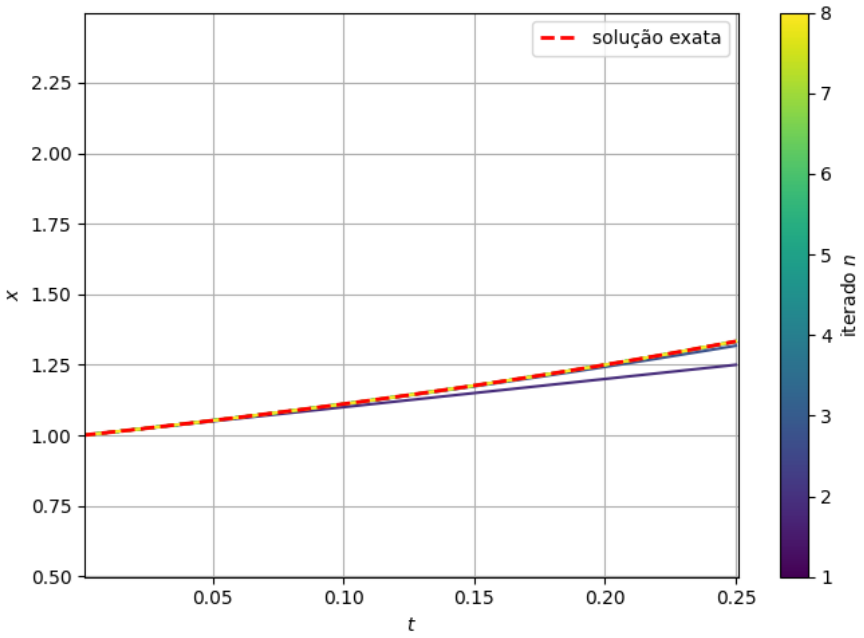

Figura 1: Iterados $\gamma_1, \ldots, \gamma_8$ e solução exata $x(t) = 1/(1-t)$ (tracejado vermelho) em $[0, 0{,}25]$. A convergência é rápida graças a $C\varepsilon^* = 1$ (Observação 1.6).

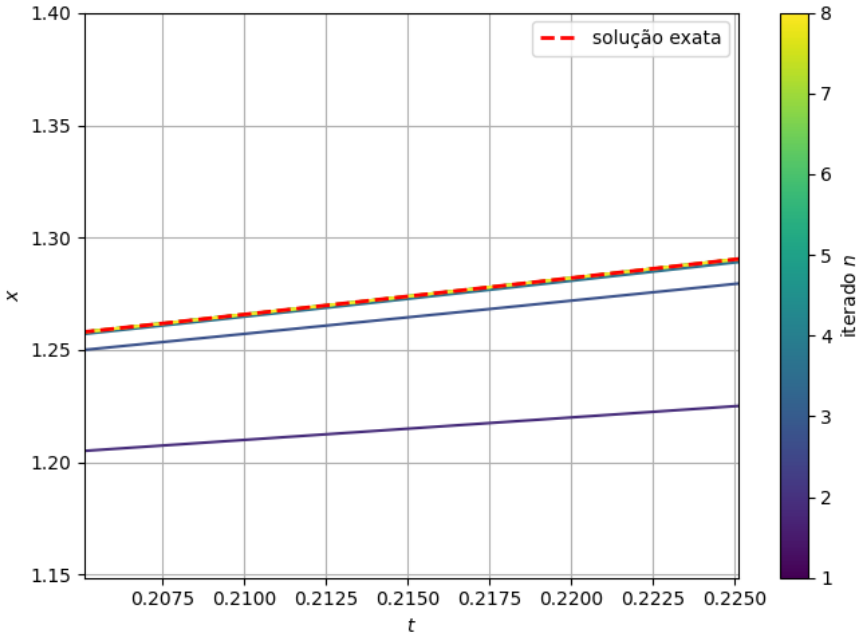

Figura 2: Zoom para observação do comportamento da convergência.

4 Objetivo 2

O Objetivo 2 estende a análise para $\varepsilon = 0{,}5 > \varepsilon^* = 0{,}25$, saindo da região de garantia do Teorema 1.3, e introduz a comparação entre os integradores do trapézio e do retângulo. A convergência fora do raio teórico é justificada pela conexão entre os iterados de Picard

e as somas parciais da série geométrica (15), discutida na Seção 4.1 abaixo; a diferença de ordem entre as duas regras de quadratura é então quantificada pela tabela de erros da Seção 4.3.

4.1 Configuração

Parâmetro	Descrição	Valor
ε	Raio	0,5
L	Pontos	2000
N	Iterados	14

Tabela 2: Parâmetros do Objetivo 2.

Embora $\varepsilon = 0{,}5 > \varepsilon^* = 0{,}25$ ultrapasse a garantia do Teorema 1.3, a convergência numérica ocorre em todo o intervalo $[0,\ 0{,}5]$. Isso decorre de um fato estrutural: os iterados de Picard para $F(t, x) = x^2$ com $x(0) = 1$ coincidem exatamente com as *somas parciais* da série geométrica

$$x(t) = \frac{1}{1-t} = \sum_{k=0}^{\infty} t^k, \tag{15}$$

cujo raio de convergência é $|t| < 1$. Portanto, para qualquer $\varepsilon < 1$, os iterados convergem para a solução exata, ainda que o Teorema 1.3 só garanta isso para $\varepsilon \leq \varepsilon^* = 0{,}25$. Este é um dos pontos matematicamente mais elegantes do experimento, pois o método de Picard constrói, iterado a iterado, os coeficientes da expansão em série de Taylor da solução.

4.2 Implementação das duas regras

```
h = t[1] - t[0]

for n in range(N):
    for k in range(L):
        integrando   = F(t[:k+1], y_tp[n, :k+1])
        y_tp[n+1, k] = x0 + np.trapezoid(integrando, t[:k+1])

for n in range(N):
    for k in range(L):
        integrando   = F(t[:k+1], y_rt[n, :k+1])
        y_rt[n+1, k] = x0 + np.sum(integrando[:-1]) * h
```

Listing 2: EDO2.py — Métodos do trapézio e retângulo.

A regra do trapézio já foi implementada na Código 1, discutiremos apenas a implementação do novo método. Na regra do retângulo, `integrando[:-1]` seleciona os k pontos à esquerda de cada subintervalo, implementando (11).

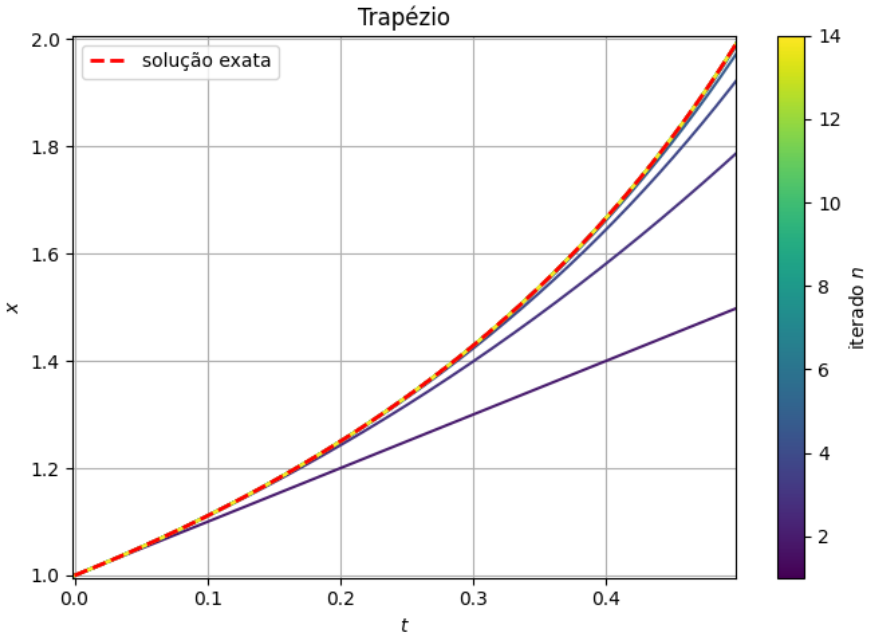

Figura 3: Iterados $\gamma_1, \ldots, \gamma_{14}$ pelo método do trapézio e solução exata (tracejado vermelho).

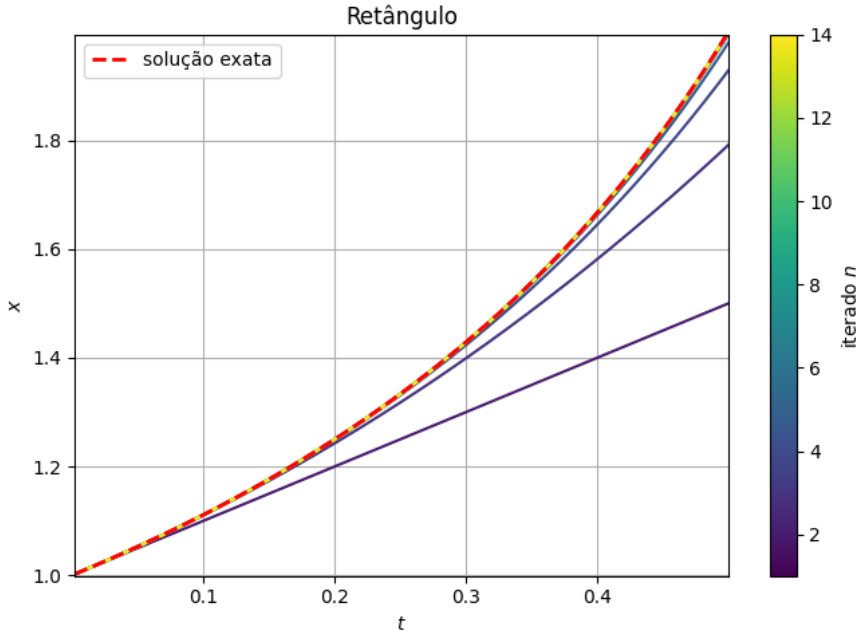

Figura 4: Iterados $\gamma_1, \ldots, \gamma_{14}$ pelo método do retângulo e solução exata (tracejado vermelho).

4.3 Tabela de erros

Tabela 3: Erro máximo $\|\gamma_n - x\|_\infty$ em $[0,\ 0{,}5]$. $L = 2000$, $N = 14$.

n	Trapézio	Retângulo
0	1.00e+00	1.00e+00
2	2.08e-01	2.08e-01
5	4.67e-03	5.32e-03
8	2.53e-05	7.18e-04
11	8.11e-08	6.93e-04
14	1.25e-07	6.93e-04

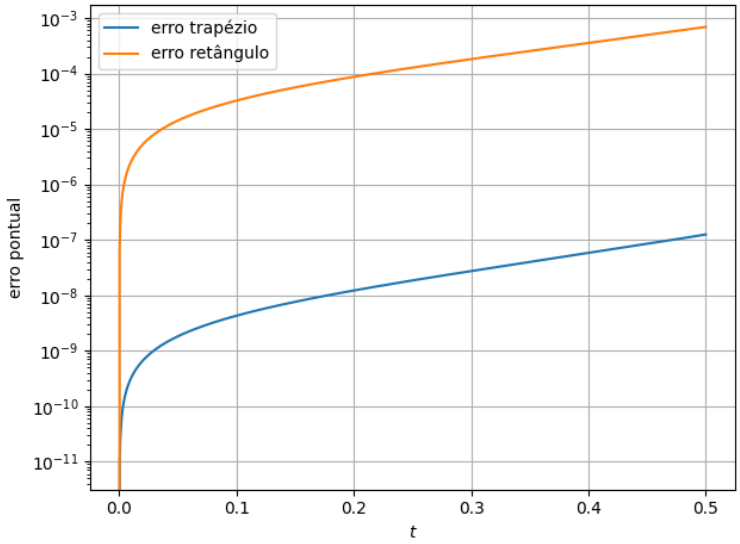

Figura 5: Erro pontual $|\gamma_{14}(t) - x(t)|$ em escala logarítmica. O trapézio apresenta erro sistematicamente menor, consistente com a diferença de ordem em (10)–(11).

5 Objetivo 3

O Objetivo 3 estende o domínio para $(-\varepsilon, \varepsilon)$, exigindo integração bidirecional a partir de $t_0 = 0$. Essa extensão expõe um problema assimétrico entre os dois integradores no extremo esquerdo: enquanto o trapézio mantém convergência em L, o retângulo acumula um piso de erro irredutível que não pode ser reduzido aumentando N. O domínio mais amplo ($\varepsilon = 0{,}85$) também prepara a análise de convergência para domínios maiores desenvolvida no Objetivo 4.

5.1 Configuração

Parâmetro	Descrição	Valor
ε	Raio de $(-\varepsilon, \varepsilon)$	0,85
L	Pontos (ímpar)	10001
N	Iterados	14

Tabela 4: Parâmetros do Objetivo 3.

O valor ímpar de L é exigido para que $i_0 = \lfloor L/2 \rfloor = 5000$ satisfaça $t_{i_0} = 0$ exatamente (cf. (9)).

5.2 Integração

Para $t < t_0 = 0$, a equação integral (2) requer

$$\gamma_{n+1}(t) = x_0 - \int_t^0 F\Big(s,\, \gamma_n(s)\Big)\, ds, \quad t < 0.$$

Assim, podemos dividir o algoritmo de integração nas seguintes etapas:

```
i0 = L // 2    # indice de t = 0

for n in range(N):
    for k in range(L):
        if k > i0:      # t > 0
            intg          = F(t[i0:k+1], y_tp[n, i0:k+1])
            y_tp[n+1, k] = x0 + np.trapezoid(intg, t[i0:k+1])
        elif k < i0:    # t < 0: sinal negativo
            intg          = F(t[k:i0+1], y_tp[n, k:i0+1])
            y_tp[n+1, k] = x0 - np.trapezoid(intg, t[k:i0+1])
        else:
            y_tp[n+1, k] = x0
```

Listing 3: EDO3.py — integração bidirecional.

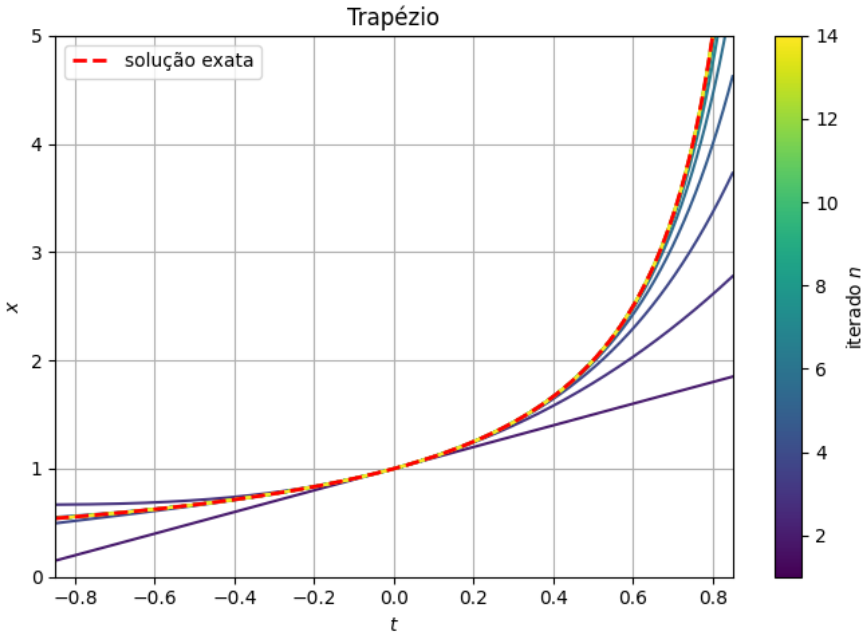

Figura 6: Método do trapézio em EDO3.py .

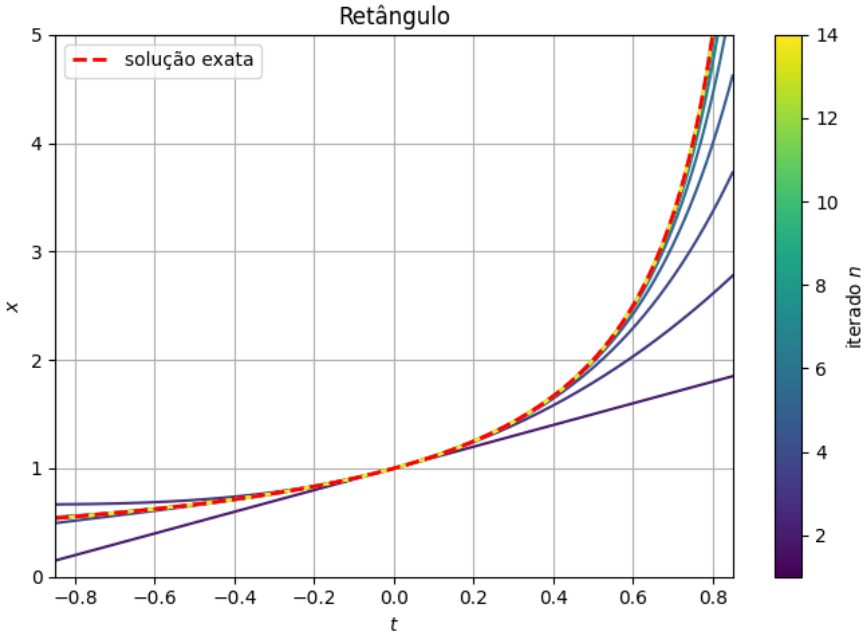

Figura 7: Método do retângulo em EDO3.py .

5.3 Piso irredutível do retângulo no extremo esquerdo

O erro de quadratura global da regra do retângulo ao integrar sobre $[t, 0]$ para $t < 0$ é

$$\text{erro retângulo} = \mathcal{O}(h) \cdot |t| = \mathcal{O}\left(\frac{\varepsilon^2}{L}\right).$$

Para o trapézio, o erro local por subintervalo é $\mathcal{O}(h^3)$, resultando em erro global $\mathcal{O}(\varepsilon^3/L^2)$. Consequentemente, para ε fixo e N suficientemente grande (de modo que o primeiro termo de (12) seja desprezível), o retângulo exibe um erro estacionário proporcional a ε^2/L que não pode ser reduzido aumentando N, apenas aumentando L. O trapézio converge duas ordens a mais em L, tornando-o sempre preferível do ponto de vista da convergência.

5.4 Tabela de erros

Os decaimentos na Figura 8, são pontos onde a curva iterada toca ou se aproxima muito da solução exata. A assimetria entre os lados $t > 0$ e $t < 0$ na Figura 8 deve-se ao acumulo de erro pelo uso de pontos à esquerda.

n	Trapézio	Retângulo
0	5.67e+00	5.67e+00
3	2.94e+00	2.94e+00
6	7.32e-01	7.39e-01
9	6.67e-02	7.97e-02
12	2.36e-03	1.66e-02
14	1.65e-04	1.44e-02

Tabela 5: Erro máximo $\|\gamma_n - x\|_\infty$ em $(-0{,}85,\ 0{,}85)$. $L = 10001$, $N = 14$.

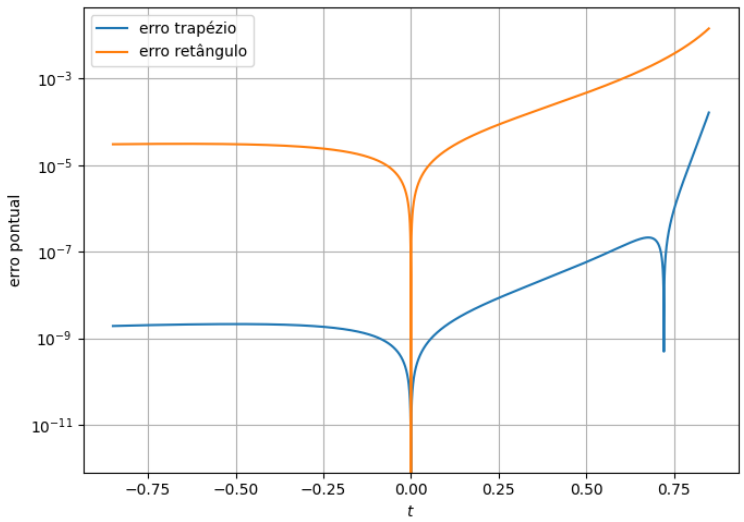

Figura 8: Erro pontual $|\gamma_{14}(t) - x(t)|$ em escala logarítmica.

6 Objetivo 4

O Objetivo 4 concentra as contribuições computacionais e teóricas mais relevantes do trabalho. Do lado algorítmico, a implementação direta de complexidade $\mathcal{O}(NL^2)$ é substituída por uma versão incremental de complexidade $\mathcal{O}(NL)$, tornando viável o uso de $L = 20001$ pontos. Do lado teórico, calculam-se $M(\delta)$, $C(\delta)$ e $\varepsilon(\delta)$ para $F(t, x) = x^2$, determina-se o parâmetro ótimo δ^* e obtém-se $N_{\max}(\tau)$ como função da tolerância admissível. A questão da convergência para $\varepsilon > \varepsilon^*$, inclusive no limite $\varepsilon \to 1^-$, é tratada via a série geométrica (15) discutida no Objetivo 2.

6.1 Configuração

Parâmetro	Descrição	Valor
ε	Raio	0,99
L	Pontos (ímpar)	20001
N	Iterados	30

Tabela 6: Parâmetros do Objetivo 4.

6.2 Redução de complexidade: $\mathcal{O}(NL^2)$ para $\mathcal{O}(NL)$

Na implementação dos Objetivos 1–3, o cálculo de $\gamma_{n+1}(t_k)$ aplica o algoritmo de quadratura no subintervalo $[t_{i_0}, t_k]$, o que exige o recálculo de toda a integral desde o ponto inicial a cada passo, cujo custo é $\Theta(k - i_0)$ operações. Somando sobre todos os pontos à direita de i_0:

$$\sum_{k=i_0}^{L-1} (k - i_0) = \sum_{j=0}^{L-1-i_0} j = \frac{(L - 1 - i_0)(L - i_0)}{2} = \mathcal{O}(L^2), \tag{16}$$

e de forma análoga no lado negativo ($k < i_0$), totalizando $\mathcal{O}(L^2)$ por iterado e $\mathcal{O}(NL^2)$ no total. Para $N = 30$ e $L = 20001$, isso representa $\approx 3 \times 10^9$ operações.

A redução a $\mathcal{O}(NL)$ baseia-se na propriedade de *acumulação incremental*: a regra do trapézio satisfaz

$$I_k^+ = I_{k-1}^+ + \frac{f_{k-1} + f_k}{2}\, h, \quad k = i_0 + 1, \ldots, L - 1, \quad I_{i_0}^+ = 0, \tag{17}$$

e a do retângulo,

$$I_k^+ = I_{k-1}^+ + f_{k-1}\, h. \tag{18}$$

Cada I_k^+ custa $\Theta(1)$ operações, reduzindo o custo por iterado a $\mathcal{O}(L)$. A recorrência análoga no sentido negativo usa $I_k^- = I_{k+1}^- + (f_k + f_{k+1})h/2$.

Versão	Complexidade	Operações (estimativa)	Fator
Direta (Objetivos 1–3)	$\mathcal{O}(NL^2)$	$\approx 3{,}0 \times 10^9$	2500×
Incremental (EDO4_v2)	$\mathcal{O}(NL)$	$\approx 1{,}2 \times 10^6$	1×

Tabela 7: Comparativo de complexidade para $N = 30$, $L = 20001$.

A estimativa da versão direta usa $N \cdot L^2/4$ operações(as contribuições positiva e negativa somam $L^2/4$, obtido de (16)), a versão incremental usa $N \cdot 2L$ operações.

```
for n in range(N):
    f_tp = F(t, y_tp[n])      # avalia F uma vez em todos os
       pontos

    I_tp_pos = np.zeros(L)
    for k in range(i0 + 1, L):           # direcao positiva
        I_tp_pos[k] = I_tp_pos[k-1] + (f_tp[k-1]+f_tp[k])/2.0 *
            h

    I_tp_neg = np.zeros(L)
    for k in range(i0 - 1, -1, -1):      # direcao negativa
        I_tp_neg[k] = I_tp_neg[k+1] + (f_tp[k]+f_tp[k+1])/2.0 *
            h

    y_tp[n+1, i0+1:] = x0 + I_tp_pos[i0+1:]
    y_tp[n+1, :i0]   = x0 - I_tp_neg[:i0]
    y_tp[n+1, i0]    = x0
```

Listing 4: EDO4_v2.py .

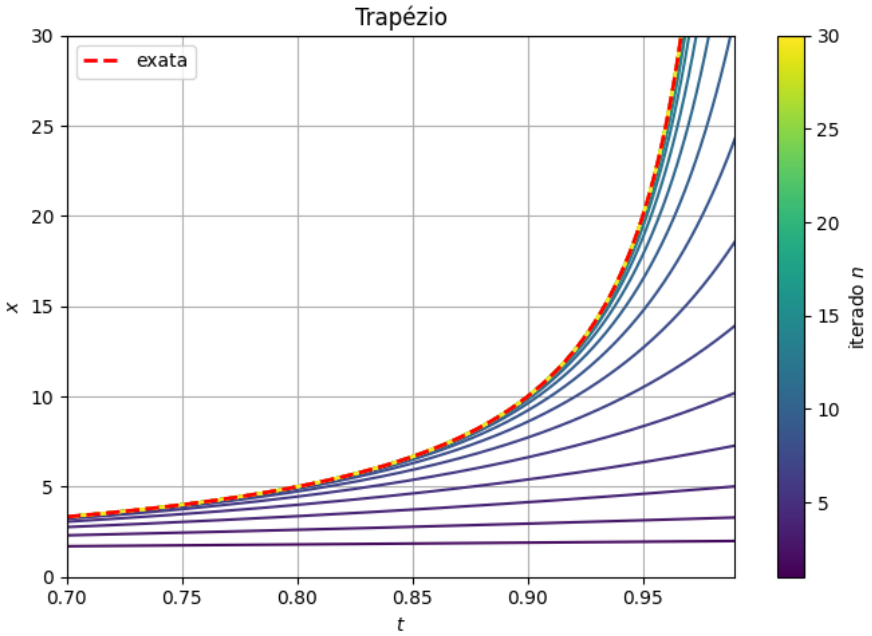

Figura 9: Método do trapézio em EDO4.py .

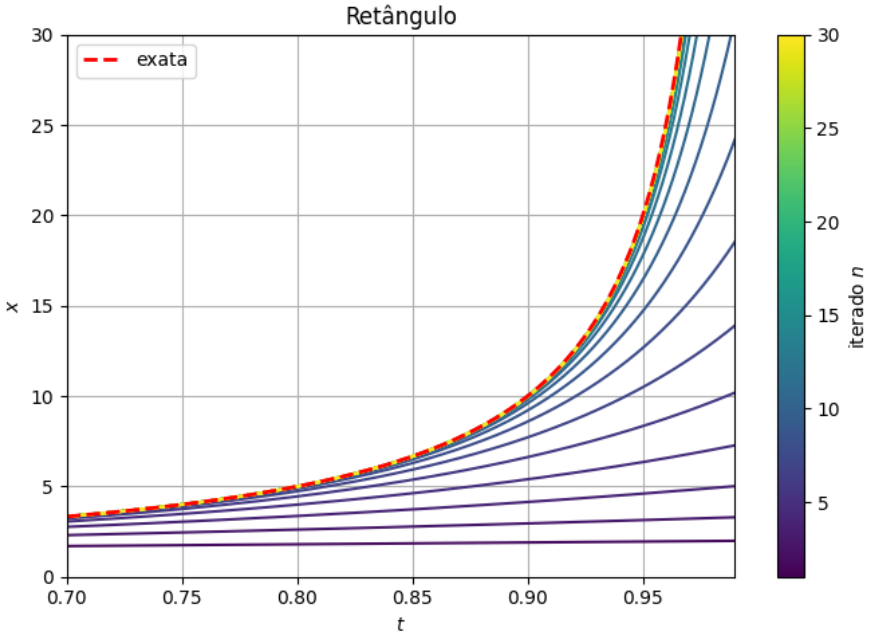

Figura 10: Método do retângulo em EDO4.py .

6.3 Análise teórica completa para $F(t, x) = x^2$

As quantidades (13) foram determinadas em (14). A Tabela 8 lista os valores para escolhas representativas de δ, ilustrando que $\delta^* = 1$ é o único máximo de $\varepsilon(\delta)$.

δ	$M(\delta)$	$C(\delta)$	$\varepsilon(\delta)$	$N_{\max}(\tau = 10^{-6})$
0,25	1,5625	2,5000	0,1600	7
0,50	2,2500	3,0000	0,2222	9
1,00	4,0000	4,0000	0,2500	11
1,50	6,2500	5,0000	0,2400	12
2,00	9,0000	6,0000	0,2222	12

Tabela 8: Quantidades teóricas para $F = x^2$ em função de δ. $\delta^* = 1$ maximiza $\varepsilon(\delta)$ e minimiza $N_{\max}$.

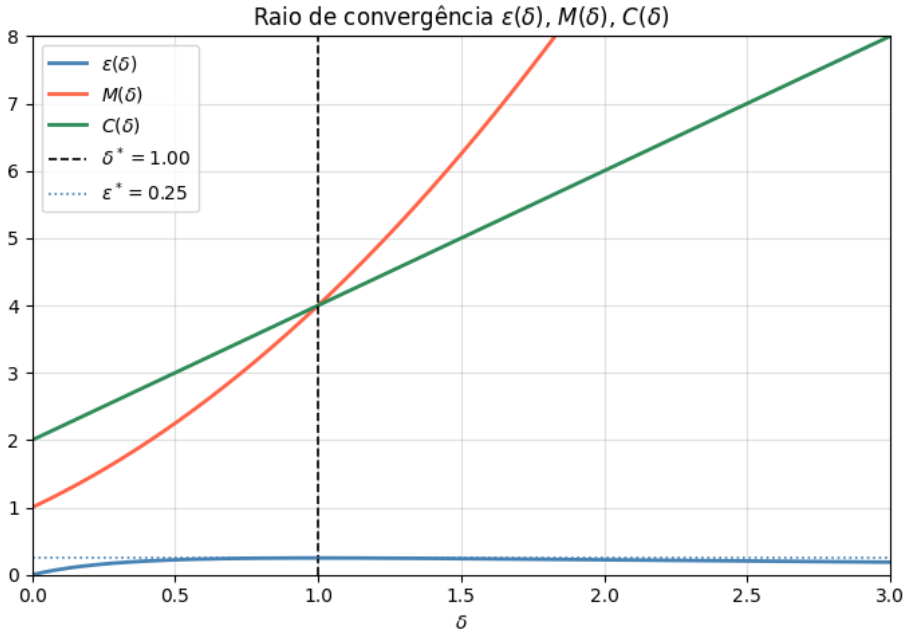

Figura 11: Curvas $\varepsilon(\delta)$, $M(\delta)$, $C(\delta)$ e máximo $\delta^* = 1$ (tracejado vertical).

O valor $N_{\max}(\tau)$ definido em (8) é calculado via a recorrência

$$M\,\frac{(C\varepsilon)^n}{n!} = M\,\frac{(C\varepsilon)^{n-1}}{(n-1)!}\cdot\frac{C\varepsilon}{n},$$

que evita *overflow* numérico ao não calcular as potências e os fatoriais separadamente.

Tolerância τ	$N_{\max}$
10^{-2}	6
10^{-4}	8
10^{-6}	11
10^{-8}	12
10^{-10}	14
10^{-12}	17

Tabela 9: $N_{\max}(\tau)$ para $F = x^2$ com $\delta^* = 1$, $\varepsilon^* = 0{,}25$.

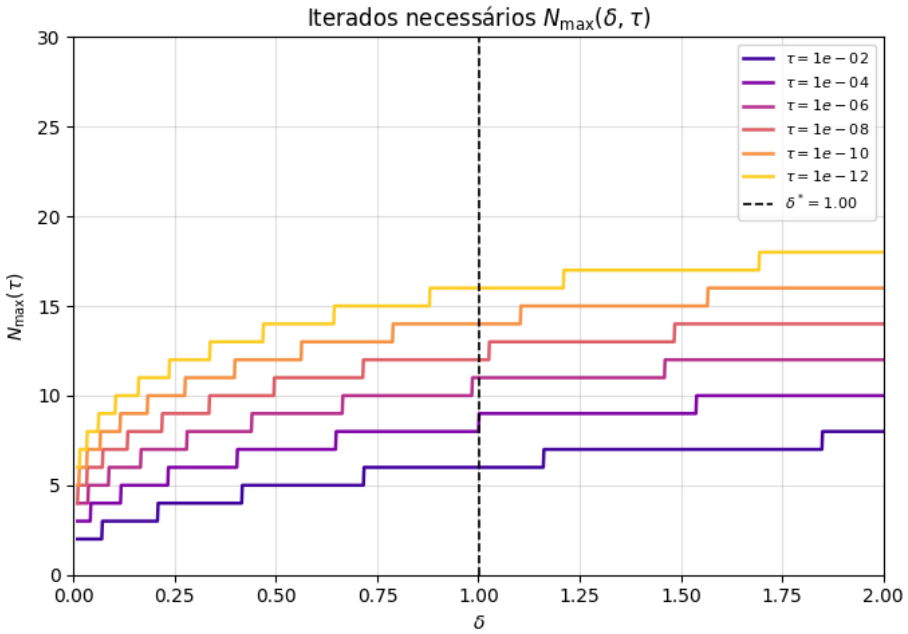

Figura 12: $N_{\max}(\delta, \tau)$ para seis tolerâncias.

6.4 Comparação entre os domínios

O código usa $\varepsilon_{\text{num}} = 0{,}99 \gg \varepsilon^* = 0{,}25$. O Teorema 1.3 não garante convergência para $\varepsilon > \varepsilon(\delta^*)$; contudo, conforme observado na Seção 4, a série (15) converge para $|t| < 1$, de modo que os iterados convergem numericamente em $|t| \leq 0{,}99$. Para $t \in (0{,}99,\ 1)$, a solução cresce rapidamente em direção à singularidade em $t = 1$ e os iterados precisam de $N \gg N_{\max}(\varepsilon^*)$ iterações para acompanhá-la, conforme ilustrado pelo produto $C\varepsilon_{\text{num}} = 4 \times 0{,}99 \approx 3{,}96 > 1$ (Observação 1.6). Isso explica por que, na Figura 13, as barras numéricas são sistematicamente mais altas do que a cota teórica calculada com ε^*.

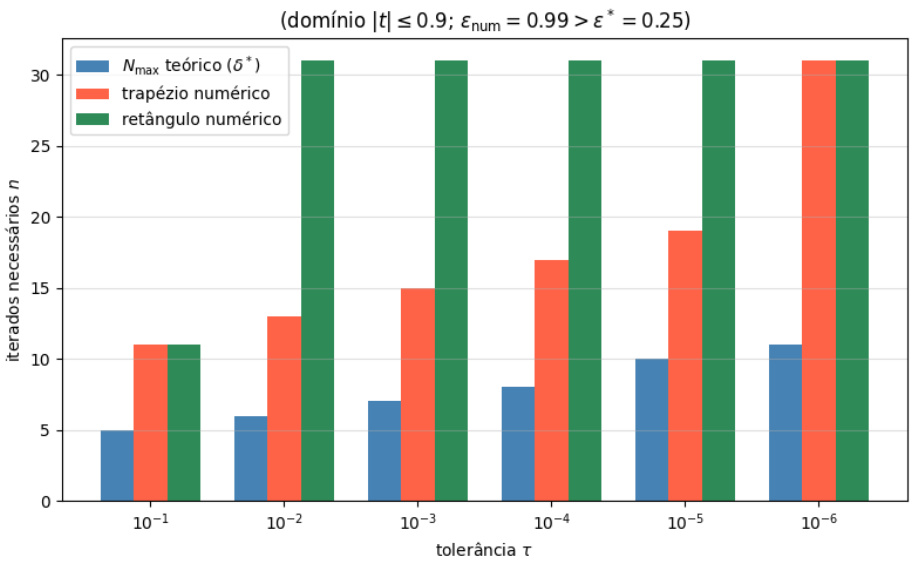

Figura 13: Comparação entre os $N_{\max}$ obtido analiticamente e as iterações aos métodos do trapézio e retângulo. (Seção 6.2).

7 Objetivo 5

7.1 Formulação do problema

A equação $x'' = -x + 2\,\mathrm{sen}\,t$ com condições iniciais $x(0) = 1$, $x'(0) = 0$ é reescrita como sistema de primeira ordem:

$$\begin{pmatrix} x_1' \\ x_2' \end{pmatrix} = \underbrace{\begin{pmatrix} x_2 \\ -x_1 + 2\,\mathrm{sen}\,t \end{pmatrix}}_{F(t,\mathbf{x})}, \qquad \mathbf{x}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{19}$$

A solução exata é

$$x_1(t) = (1-t)\cos t + \mathrm{sen}\,t, \qquad x_2(t) = -(1-t)\,\mathrm{sen}\,t. \tag{20}$$

Ao contrário do Objetivo 4, $\mathbf{x}$ é uma função inteira em $\mathbb{R}$ (sem singularidade real), implicando que os iterados convergem em qualquer domínio compacto.

7.2 Configuração

Parâmetro	Descrição	Valor
ε	Raio	$0,9468$
L	Pontos (ímpar)	20001
N	Iterados	14
$\mathbf{x}_0$	Condição inicial	$(1,\,0)^T$

Tabela 10: Parâmetros do Objetivo 5.

7.3 Adaptações para $d = 2$

O vetor de iterados passa a ter forma $(N+1, L, 2)$ e os acumuladores I^+, I^- têm forma $(L, 2)$; a recorrência (17) é aplicada elemento a elemento pelo NumPy sem alteração de código.

Objeto	EDO4 ($d = 1$)	EDO5 ($d = 2$)	Descrição
`y_tp`	$(N+1,\,L)$	$(N+1,\,L,\,2)$	Iterados
`I_tp_pos`	$(L,)$	$(L,\,2)$	Acumulação positiva
`x0`	`1.0`	`array([1.,0.])`	Condição inicial

Tabela 11: Diferenças de estrutura entre EDO4 e EDO5.

```
def F(t, x):              # x tem estrutura (L, 2)
    dx1 = x[:, 1]
    dx2 = -x[:, 0] + 2.0 * np.sin(t)
    return np.column_stack([dx1, dx2])

y_tp = np.zeros((N+1, L, 2))
y_tp[0, :, :] = x0        # x0 = array([1., 0.])

I_tp_pos = np.zeros((L, 2))
for k in range(i0 + 1, L):
    I_tp_pos[k] = I_tp_pos[k-1] + (f_tp[k-1]+f_tp[k]) / 2.0 * h
```

Listing 5: EDO5.py

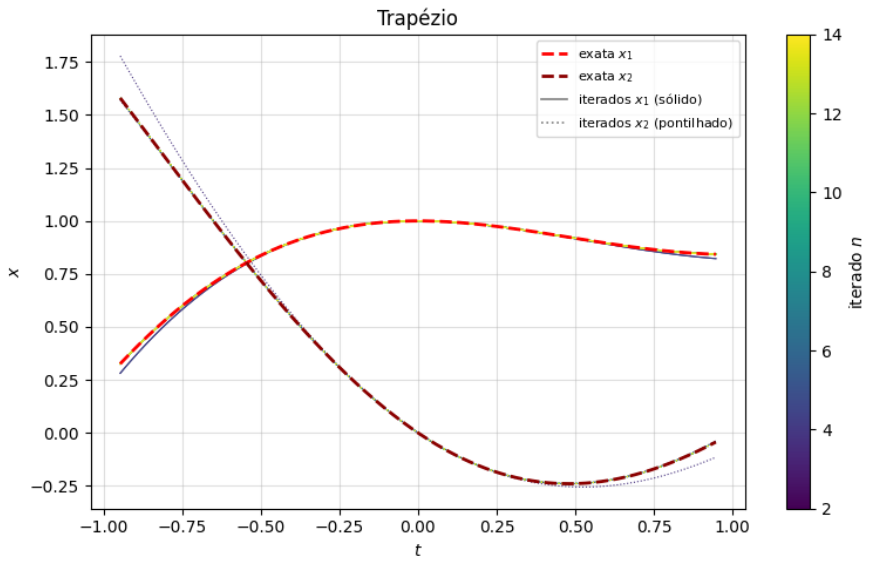

Figura 14: Método do trapézio em EDO5.py .

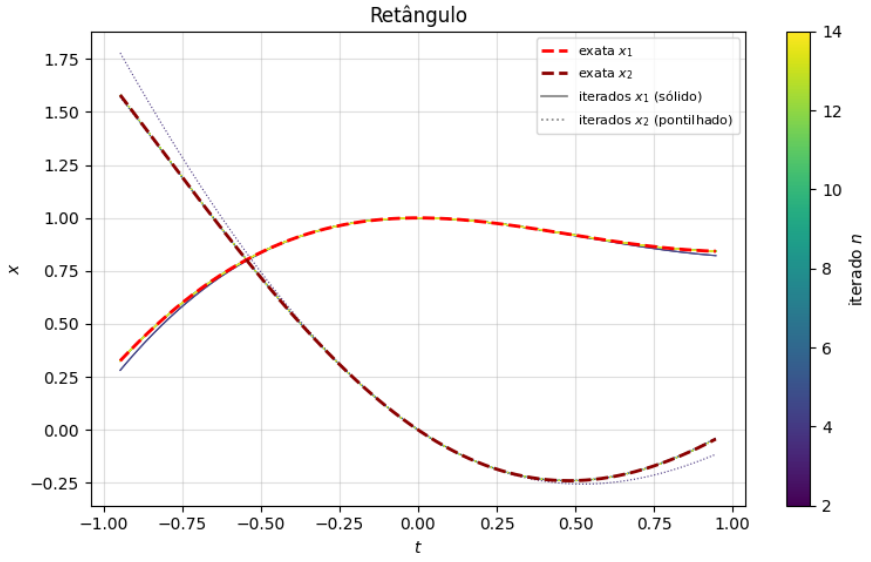

Figura 15: Método do retângulo em EDO5.py .

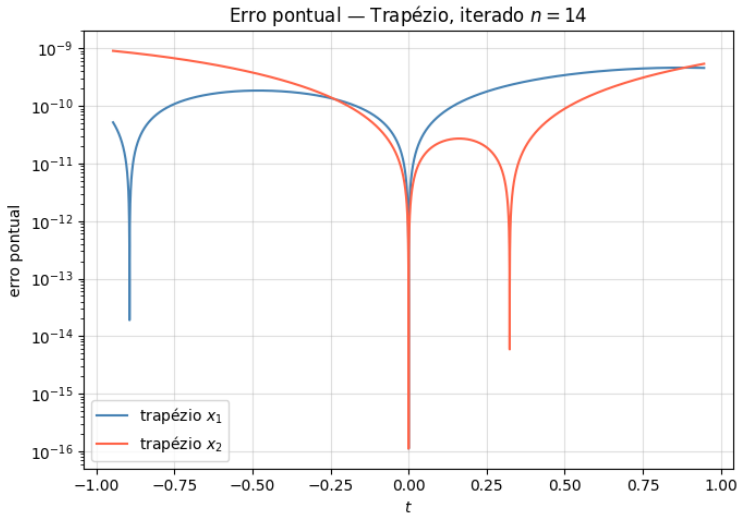

Figura 16: Erro pontual Trapézio em escala logarítmica, iterado $n = 14$.

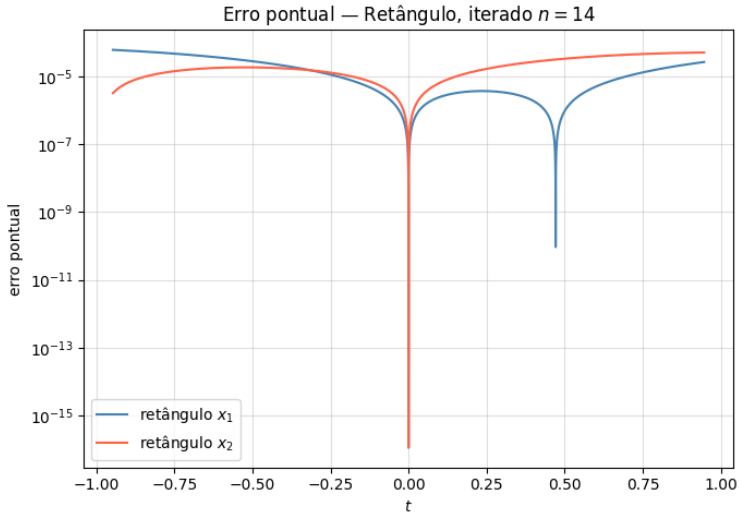

Figura 17: Erro pontual Retângulo em escala logarítmica, iterado $n = 14$.

7.4 Análise para o sistema bidimensional

Constante de Lipschitz: Para $F(t, \mathbf{x}) = (x_2,\ -x_1 + 2\,\mathrm{sen}\,t)$:

$$F(t,\mathbf{x}) - F(t,\mathbf{y}) = \begin{pmatrix} x_2 - y_2 \\ -(x_1 - y_1) \end{pmatrix}, \qquad \|F(t,\mathbf{x}) - F(t,\mathbf{y})\| = \|\mathbf{x} - \mathbf{y}\|\,. \tag{21}$$

Portanto $C(\delta) = 1$ para todo $\delta > 0$. F é afim em $\mathbf{x}$ com parte linear $A\mathbf{x}$, onde $A = \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right)$ é uma rotação e, portanto, uma isometria. A norma espectral de A é $\|A\|_2 = 1$, o que é equivalente a (21).

$M(\delta)$**:** Sobre $|t| \le \delta$, $|x_1 - 1| \le \delta$, $|x_2| \le \delta$, o pior caso ocorre em $x_2 = \delta$, $x_1 = 1 - \delta$, $t = \delta$:

$$M(\delta) = \sqrt{\delta^2 + (\delta - 1 + 2\,\mathrm{sen}\,\delta)^2}\,. \tag{22}$$

Para $\delta \to 0^+$, $M(\delta) \to 1$.

Problema	$C(\delta)$	$M(\delta)$	δ^*	ε^*
$F = x^2$	$2(1+\delta)$, cresce	$(1+\delta)^2$	1,0 (exato)	0,25 (exato)
Sistema 2D	1, constante	(22)	(numérico)	$> 0{,}25$

Tabela 12: Comparativo das quantidades teóricas entre os dois problemas.

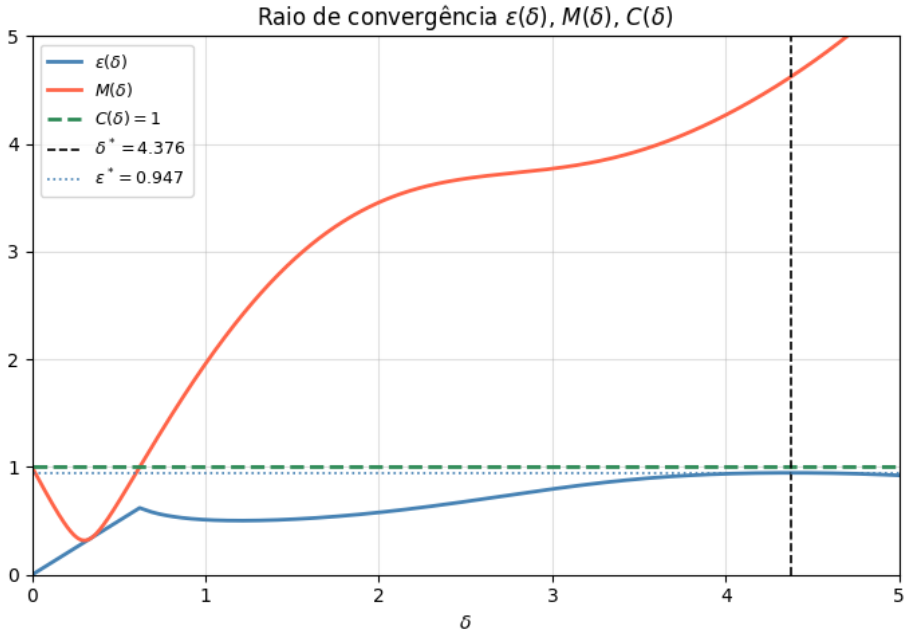

Figura 18: Curvas $\varepsilon(\delta)$, $M(\delta)$, $C(\delta)$ e máximo $\delta^* = 4,376$ (tracejado vertical).

Consequências para $N_{\max}$: Com $C(\delta) = 1$ e $\varepsilon = 1$, o produto $C\varepsilon = 1$, e pela Observação 1.6 a cota (6) reduz-se a $M(\delta^*)/n!$, que decresce monotonamente desde o primeiro iterado. Por outro lado, no Objetivo 4 com $\varepsilon_{\text{num}} = 0{,}99$, o produto $C\varepsilon_{\text{num}} \approx 3{,}96 > 1$ implica que os primeiros iterados pioram antes de convergir (Observação 1.6). O sistema bidimensional é, portanto, intrinsecamente mais favorável ao método de Picard.

Tolerância τ	$N_{\max}$ (EDO4, $F = x^2$)	$N_{\max}$ (EDO5, 2D)
10^{-2}	6	6
10^{-4}	8	8
10^{-6}	11	10
10^{-8}	12	12
10^{-10}	14	14
10^{-12}	16	16

Tabela 13: $N_{\max}(\tau)$ comparativo entre os dois problemas (com δ^* e ε^* de cada um).

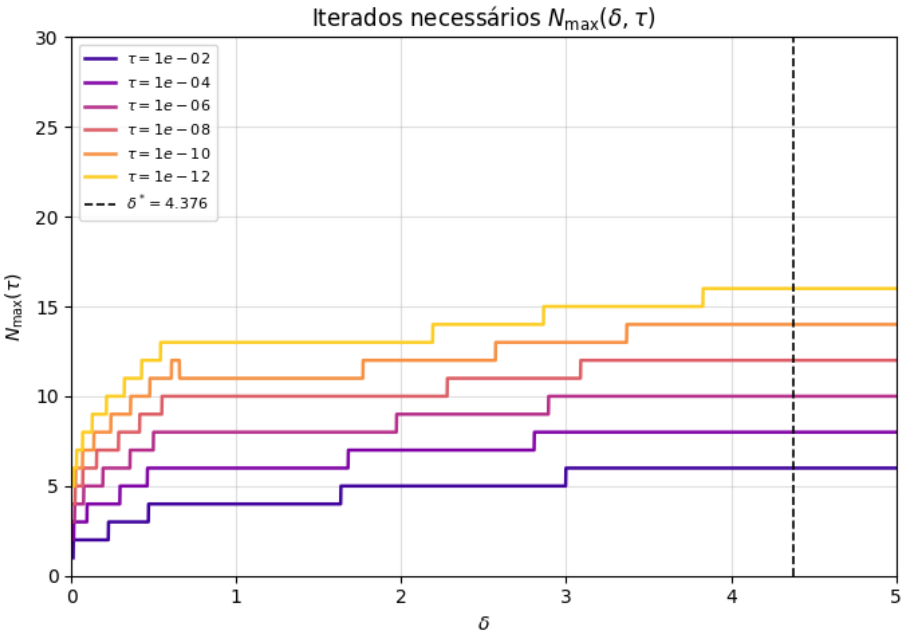

Figura 19: $N_{\max}(\delta, \tau)$ para seis tolerâncias.

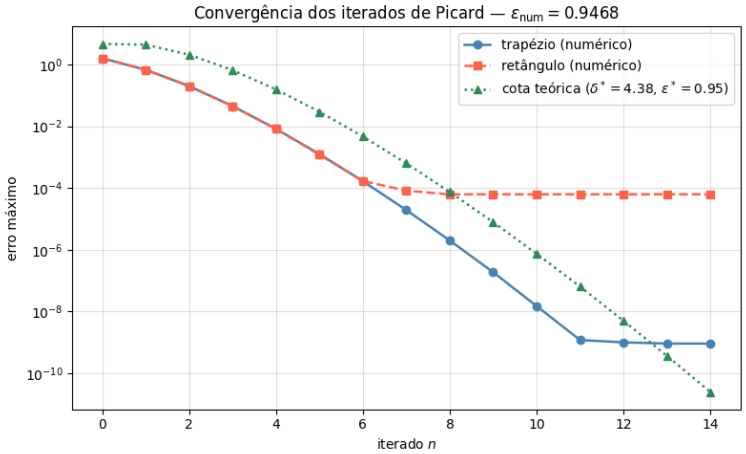

Figura 20: Convergência dos iterados de Picard para $\varepsilon*$. Comparação entre trapézio, retângulo e cota analítica $M(\delta^*)\,(\varepsilon^*)^n/n!$.

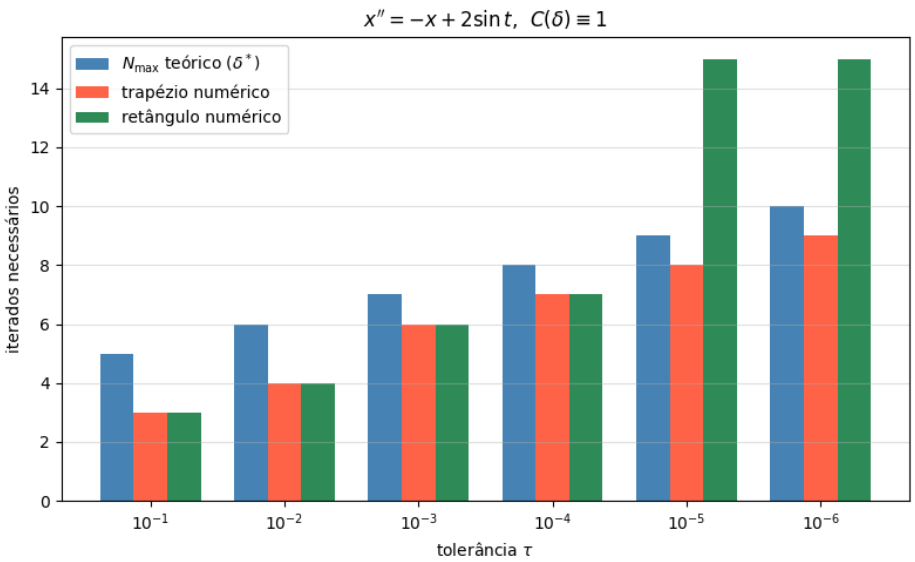

Figura 21: Comparação entre os $N_{\max}$ obtido analiticamente e as iterações aos métodos do trapézio e retângulo.

Um dos pontos que diferenciam a EDO tratada anteriormente nos Objetivos 1-4 e a do Objetivo 5 além do domínio é que a solução (20) é inteira.

8 Comparação dos códigos

Esta seção reúne uma análise comparativa dos cinco códigos, considerando custo computacional, comportamento do erro e o papel da constante $C(\delta)$ na estruturação do método. Os resultados individuais de cada objetivo são retomados aqui sob uma perspectiva unificada.

8.1 Evolução dos algoritmos

Arquivo	Domínio	Complexidade
EDO1.py	[0, 0,25]	$\mathcal{O}(NL^2)$
EDO2.py	[0, 0,5]	$\mathcal{O}(NL^2)$
EDO3.py	$(-0{,}85,\ 0{,}85)$	$\mathcal{O}(NL^2)$
EDO4.py	$(-0{,}99,\ 0{,}99)$	$\mathcal{O}(NL)$
EDO5.py	$(-1,\ 1)$, $d = 2$	$\mathcal{O}(NL)$

Tabela 14: Sumário dos cinco objetivos.

8.2 Erro de iteração e erro de convergência

O erro total (12) decompõe-se em dois termos com dependências opostas em n e L: aumentar n reduz o primeiro exponencialmente (via o fatorial) sem afetar o segundo; aumentar L reduz o segundo como L^{-p} sem afetar o primeiro. O L mínimo efetivo para equilíbrio, dado N fixo, satisfaz

$$\varepsilon^2 L^{-p} \sim M(\delta^*)\,\frac{\left(C\varepsilon^*\right)^N}{N!}\,.$$

8.3 Importância de $C(\delta)$ na estruturação do algoritmo

Problema	$C(\delta)$	Comportamento para $\varepsilon > \varepsilon^*$
$x' = x^2$	$2(1+\delta)$, cresce	Convergência lenta; com teto de $\lvert t\rvert < 1$.
Sistema 2D	1, constante	Convergência global; sem teto pois a solução é inteira.

Tabela 15: Impacto de $C(\delta)$ na estruturação do método de Picard.

Para o caso dos objetivos 1-4, o leitor pode tentar executar o código 4 para $\varepsilon > 1$ e ver a divergência acontecendo para os valores de $|t| > 1$. O fato de $C(\delta) = 1$ para o sistema bidimensional reflete a estrutura algébrica do campo: a parte não-linear de F é apenas a forçante $2\,\mathrm{sen}\,t$ (independente de $\mathbf{x}$), enquanto a parte linear $A\mathbf{x}$ tem $\|A\|_2 = 1$. Em geral, problemas cujo campo F admite uma decomposição $F(t,\mathbf{x}) = A\mathbf{x} + G(t)$ com $\|A\|_2$ pequena e G uniformemente limitada são intrinsecamente mais favoráveis ao método de Picard: a cota (6) decai mais rapidamente e domínios maiores são acessíveis sem aumentar N. Por outro lado, há uma limitação numérica ao aumentar o domínio em ambos os casos, pois existe o acúmulo de erro do método numérico, mas esse erro pode ser controlado com uma otimização dos parâmetros como mostramos.

9 Conclusões

Este projeto implementou o método de Picard para dois problemas distintos e demonstrou que o desempenho do método depende diretamente da estrutura algébrica do campo F.

Para $F(t,x) = x^2$, a constante de Lipschitz $C(\delta) = 2(1+\delta)$ cresce com δ, de modo que o produto $C(\delta)\varepsilon$ determina tanto o tamanho do domínio acessível quanto a velocidade de convergência. O raio ótimo $\varepsilon^* = 0{,}25$ é aquele que equilibra esses dois efeitos; para $\varepsilon > \varepsilon^*$, a convergência ocorre graças à conexão entre os iterados de Picard e as somas parciais da série geométrica (15), que converge para todo $|t| < 1$, mas com custo crescente em N conforme ilustrado nas Tabelas 9 e 13. A otimização

por acumulação incremental (Seção 6.2) foi necessária para tornar viável a exploração desse regime, reduzindo a complexidade de $O(NL^2)$ para $O(NL)$.

Para o sistema bidimensional, $C(\delta) = 1$ para todo δ, reflexo de A ser uma isometria. Com $C\varepsilon = 1$ para $\varepsilon = 1$, a cota (6) decai monotonamente desde o primeiro iterado, sem o período inicial de crescimento observado no Objetivo 4 quando $C\varepsilon_{\text{num}} \approx 3{,}96 > 1$. Além disso, a solução (20) é inteira, de modo que não há limitação de domínio análoga ao teto $|t| < 1$ do caso escalar.

A comparação entre os dois problemas torna concreto o papel de $C(\delta)$: quanto mais próximo de linear for o campo F em x, mais favorável é o método de Picard. Em ambos os casos, o trapézio mostrou-se sempre preferível ao retângulo, com ordem $O(h^2)$ frente ao $O(h)$ do retângulo, implicando um piso de erro mais baixo que não pode ser eliminado aumentando N.

Todos os códigos se encontram em GitHub:
https://github.com/VictorMoura22/experimento-picard

Contato: victor.moura@ctec.ufal.br

Referências

[1] M. Viana e J. Espinar, *Equações Diferenciais Ordinárias*, IMPA, Rio de Janeiro, 2025.