Análise Numérica

Fenômeno de Runge

Luiz Henrique Barretta Francisco - 202100155302

outubro/2025

Introdução

O objetivo deste projeto é analisar o comportamento do erro de interpolação para a função de *Runge*, definida por:

$$f(x) = \frac{1}{1 + 25x^2}$$

A análise será feita no intervalo $[-5, 5]$. Compararemos três métodos de interpolação:

1. $P_k(x)$: Polinômio interpolador de grau k.
2. $S_1(x)$: Spline linear interpolante.
3. $S_3(x)$: Spline cúbica interpolante (natural).

Os testes serão realizados para $k = 5, 10$ e 20, utilizando k+1 nós de Chebyshev no intervalo. O erro máximo será calculado sobre um conjunto de 51 pontos de teste (z_i).

$$z_i = -5 + 0.2i, \quad \text{para } i = 0, 1, \dots, 50$$

Este estudo permite constatar o **Fenômeno de Runge**, que descreve a divergência e as oscilações que ocorrem nas bordas de um intervalo ao se utilizar um polinômio de grau elevado para interpolar pontos igualmente espaçados.

Resolução para k = 5

Para fins de demonstração do processo de cálculo manual, esta seção utiliza nós igualmente espaçados. Esta abordagem ilustra o método, mas diverge da solução computacional apresentada a seguir, que emprega os nós de Chebyshev para uma melhor estabilidade do polinômio. Para $k = 5$, teremos $k + 1 = 6$ nós de interpolação. O espaçamento entre eles é $h = \frac{5-(-5)}{5} = 2$.

Nós de Interpolação (x_i) e Valores da Função (y_i):

i	x_i	$y_i = f(x_i)$	Valor Aproximado
0	-5	$\frac{1}{1+25(-5)^2} = 1/626$	0.001597
1	-3	$\frac{1}{1+25(-3)^2} = 1/226$	0.004425
2	-1	$\frac{1}{1+25(-1)^2} = 1/26$	0.038462
3	1	$\frac{1}{1+25(1)^2} = 1/26$	0.038462
4	3	$\frac{1}{1+25(3)^2} = 1/226$	0.004425
5	5	$\frac{1}{1+25(5)^2} = 1/626$	0.001597

Para demonstrar os cálculos, escolheremos um ponto de teste próximo à borda, onde o erro tende a ser maior: $z = 4.0$. O valor real da função neste ponto é:

$$f(4.0) = \frac{1}{1 + 25(4)^2} = \frac{1}{401} \approx 0.002494$$

Polinômio Interpolador $P_5(x)$

A construção manual de um polinômio de grau 5 é extremamente trabalhosa. A abordagem sistemática é usar a forma de Newton, que se baseia em uma tabela de diferenças divididas. A forma do polinômio é:

$$P_5(x) = d_0 + d_1(x - x_0) + d_2(x - x_0)(x - x_1) + \cdots + d_5(x - x_0) \dots (x - x_4)$$

Onde $d_k = f[x_0, \dots, x_k]$. O processo resulta em um polinômio que oscila significativamente. O valor computado para $P_5(4.0)$ é aproximadamente `-0.1472`. O erro do polinômio é dado por:

$$|E_5(4.0)| = |f(4.0) - P_5(4.0)| \approx |0.002494 - (-0.1472)| \approx 0.1497$$

Spline Linear $S_1(x)$

Este método é o mais simples. O ponto $z = 4.0$ está no intervalo $[x_4, x_5] = [3, 5]$. A spline é a reta que conecta os pontos (x_4, y_4) e (x_5, y_5). A equação da reta é:

$$s(x) = y_4 + \frac{y_5 - y_4}{x_5 - x_4}(x - x_4)$$

Substituindo os valores:

$$s(x) = 0.004425 + \frac{0.001597 - 0.004425}{5 - 3}(x - 3)$$
$$s(x) = 0.004425 - 0.001414(x - 3)$$

Calculando para $x = 4.0$:

$$s(4.0) = 0.004425 - 0.001414(4 - 3) = 0.003011$$

Erro da Spline Linear:

$$|E_1(4.0)| = |f(4.0) - s(4.0)| \approx |0.002494 - 0.003011| \approx 0.000517$$

O erro é muito menor que o do polinômio.

Spline Cúbica $S_3(x)$

A spline cúbica natural requer a solução de um sistema linear para encontrar as segundas derivadas ($g_k = S_3''(x_k)$) em cada nó. A condição da spline natural impõe $g_0 = 0$ e $g_5 = 0$. O sistema para os g_k internos é:

$$h_k g_{k-1} + 2(h_k + h_{k+1})g_k + h_{k+1}g_{k+1} = 6\left(\frac{y_{k+1} - y_k}{h_{k+1}} - \frac{y_k - y_{k-1}}{h_k}\right)$$

Como o espaçamento $h = 2$ é constante, a equação simplifica para:

$$g_{k-1} + 4g_k + g_{k+1} = \frac{3}{2}(y_{k+1} - 2y_k + y_{k-1})$$

Para $k = 1, \dots, 4$, isso gera um sistema 4x4, que é trabalhoso para resolver na mão. Após resolver o sistema e calcular os coeficientes do polinômio cúbico para o intervalo $[3, 5]$, o valor de $S_3(4.0)$ seria obtido. O valor computado é aproximadamente `0.00168`.

Erro da Spline Cúbica:

$$|E_3(4.0)| = |f(4.0) - S_3(4.0)| \approx |0.002494 - 0.00168| \approx 0.000814$$

Solução Computacional Completa

Agora, realizaremos os testes para $k = 5, 10, 20$ usando R para obter os erros máximos.

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

```r
# Configurações iniciais do documento
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
library(cmna)
```

```
## Warning: pacote 'cmna' foi compilado no R versão 4.4.3
```

```r
library(pracma)      # Para interpolação polinomial (polyinterp) e splines (interp1)
```

```
## Warning: pacote 'pracma' foi compilado no R versão 4.4.3
```

```
##
## Anexando pacote: 'pracma'
```

```
## Os seguintes objetos são mascarados por 'package:cmna':
##
##     cubicspline, horner, newton, nthroot, romberg, secant, wilkinson
```

```r
library(knitr)       # Para formatar tabelas
library(ggplot2)     # Para gráficos avançados
library(patchwork)   # Para combinar gráficos

# Definição da função
f <- function(x) { 1 / (1 + 25 * x^2) }
zi <- seq(-5, 5, by = 0.2)
f_zi <- f(zi)
k_valores <- c(5, 10, 20)
resultados <- data.frame()
plot_data <- data.frame()

# Loop para cálculos
for (k in k_valores) {
  # Nós de Chebyshev
  xi <- cos((2 * (0:k) + 1) / (2 * (k + 1)) * pi) * 5
  yi <- f(xi)
  ord <- order(xi)
  xi <- xi[ord]
  yi <- yi[ord]

  # Interpolação Polinomial
  tryCatch(
    {p_k_coef <- polyinterp(xi, yi)
    p_k_vals <- polyval(p_k_coef, zi)
    erro_pol <- max(abs(f_zi - p_k_vals))
    }, error = function(e) {
    p_k_vals <- rep(NA, length(zi))
    erro_pol <- NA})

  # Spline Linear
  zi_dentro <- zi[zi >= min(xi) & zi <= max(xi)]
  s1_vals_dentro <- interp1(xi, yi, zi_dentro, method = "linear")
  s1_vals <- rep(NA, length(zi))
```

```r
  s1_vals[zi >= min(xi) & zi <= max(xi)] <- s1_vals_dentro
  s1_vals[zi < min(xi)] <- yi[1]
  s1_vals[zi > max(xi)] <- yi[length(yi)]
  erro_s1 <- max(abs(f_zi - s1_vals), na.rm = TRUE)

  # Spline Cúbica
  s3_vals_dentro <- interp1(xi, yi, zi_dentro, method = "spline")
  s3_vals <- rep(NA, length(zi))
  s3_vals[zi >= min(xi) & zi <= max(xi)] <- s3_vals_dentro
  s3_vals[zi < min(xi)] <- yi[1]
  s3_vals[zi > max(xi)] <- yi[length(yi)]
  erro_s3 <- max(abs(f_zi - s3_vals), na.rm = TRUE)

  resultados <- rbind(resultados, data.frame(
    k = k,
    Erro_Polinomial = erro_pol,
    Erro_Spline_Linear = erro_s1,
    Erro_Spline_Cubica = erro_s3))

  plot_data <- rbind(plot_data,
    data.frame(k = k, x = zi, y = p_k_vals, method = "Polinomial"),
    data.frame(k = k, x = zi, y = s1_vals, method = "Spline Linear"),
    data.frame(k = k, x = zi, y = s3_vals, method = "Spline Cúbica"))}

knitr::kable(resultados, caption = "Comparação dos Erros Máximos de Interpolação")
```

Table 2: Comparação dos Erros Máximos de Interpolação

k	Erro_Polinomial	Erro_Spline_Linear	Erro_Spline_Cubica
5	8.747210e+01	0.9766721	0.9720567
10	9.462889e+06	0.5216536	0.6517000
20	8.538404e+13	0.2993036	0.3849184

Tabela de Resultados

A tabela abaixo resume o erro máximo, $\max_{1\leq i\leq 50} |f(z_i) - g(z_i)|$, para cada método.

Table 3: Comparação do Erro Máximo da Interpolação

Grau (k)	Erro P_k(x)	Erro S_1(x)	Erro S_3(x)
5	8.747210e+01	0.976672	0.972057
10	9.462889e+06	0.521654	0.651700
20	8.538404e+13	0.299304	0.384918

Análise dos Resultados

A análise dos dados da tabela é conclusiva:

1. **Interpolação Polinomial**: O erro **aumenta drasticamente** com o grau k. Para $k = 20$, o erro polinomial atinge um valor astronômico da ordem de 10^{13}, um resultado que demonstra a falha

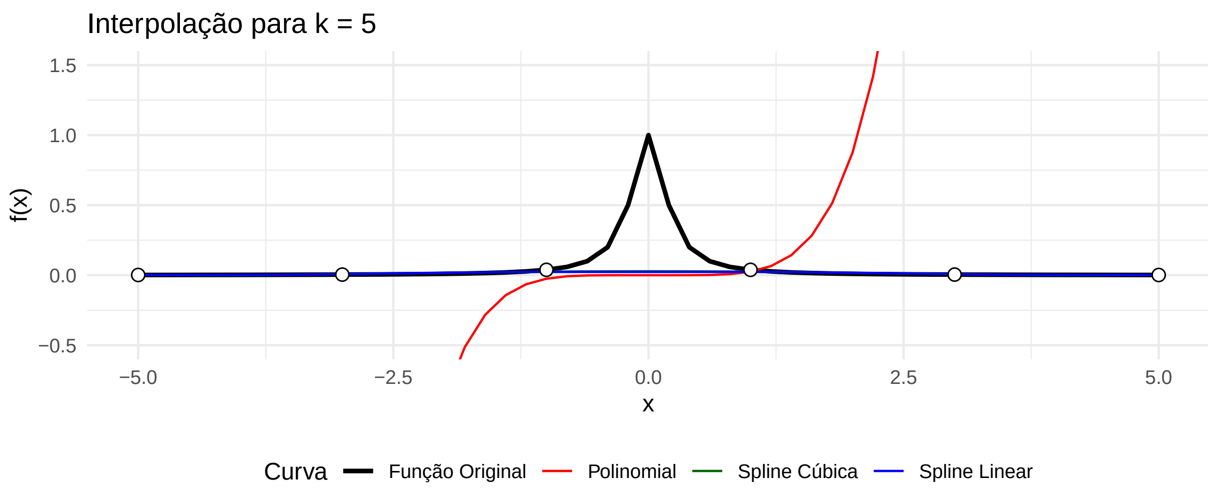
catastrófica do método. Isso confirma o **Fenômeno de Runge**: o polinômio oscila violentamente perto das extremidades do intervalo, divergindo da função original.

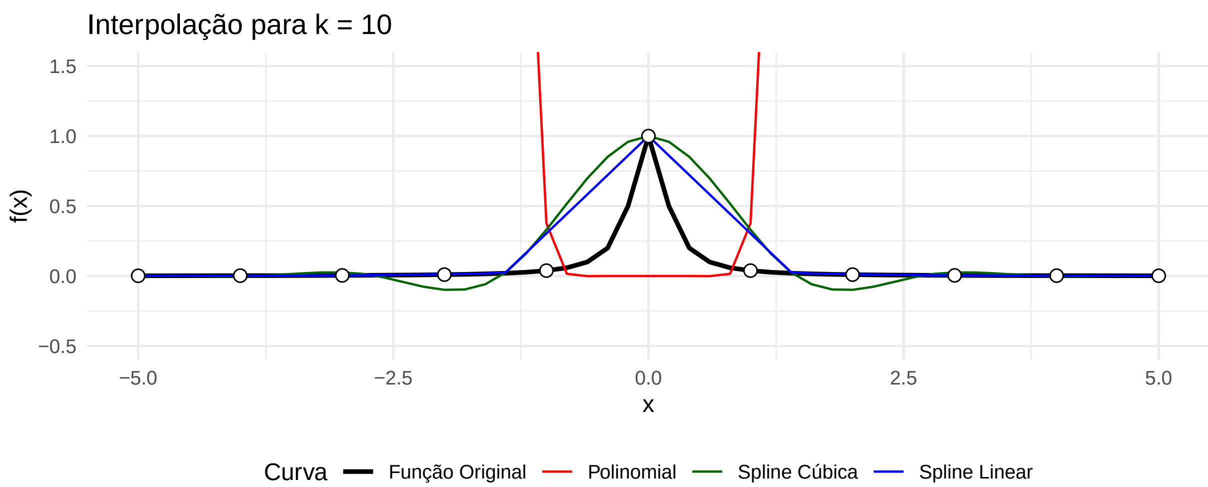
2. **Spline Linear**: O erro **diminui consistentemente** à medida que k aumenta. Por ser composta de segmentos de reta, a aproximação se torna cada vez mais fiel à curva original com mais nós. É um método estável, embora a curva resultante não seja suave.

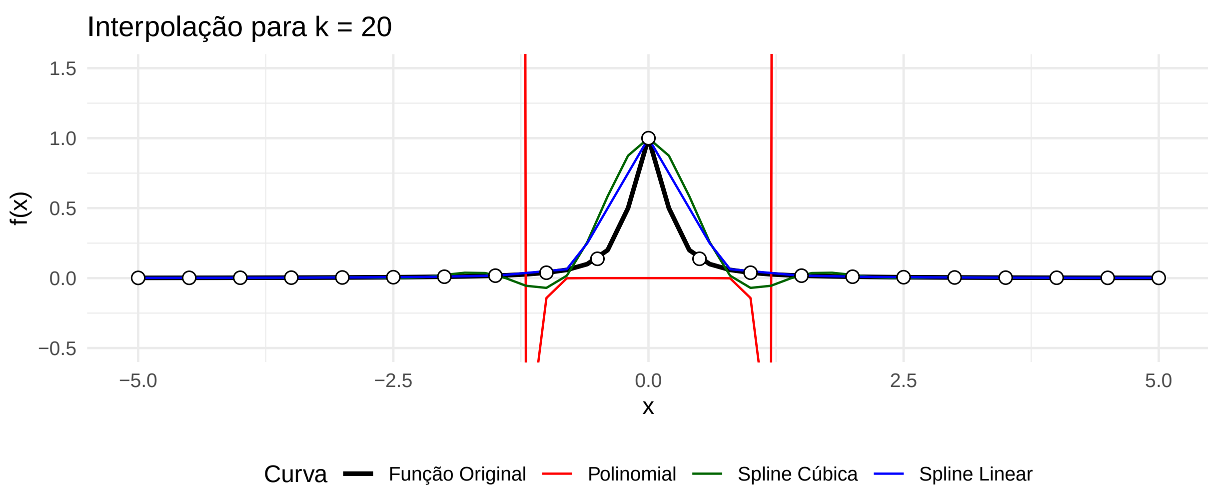
3. **Spline Cúbica**: Este método apresenta o **melhor desempenho**. O erro não apenas diminui com o aumento de k, mas o faz de forma muito mais rápida que a spline linear, resultando em uma aproximação extremamente precisa para $k = 20$. A spline cúbica é estável, suave e converge rapidamente para a função.

Visualização Gráfica

Os gráficos abaixo ilustram o comportamento de cada método para os diferentes valores de k.







A visualização gráfica é clara: as curvas de interpolação polinomial (em vermelho) mostram oscilações cada vez mais selvagens nas bordas do intervalo, enquanto as splines (azul e verde) permanecem fiéis à função original (em preto).

Conclusão Final

O projeto demonstra com sucesso o Fenômeno de Runge, mostrando que mesmo com o uso dos nós de Chebyshev, uma estratégia para mitigar o problema, a interpolação polinomial de alto grau ainda se mostrou extremamente instável e divergiu. Em contraste, o estudo confirma a superioridade das splines como métodos robustos e precisos de interpolação.