

Semana 10: Aproximação de Funções

Nessa semana vamos aprender como o computador avalia funções como e^x , $\ln(x)$ e $\sin(x)$. A ideia é mostrar que se o computador sabe encontrar valores (aproximados) para $e^{-1,5}$, $\ln(0,5)$ ou $\sin(1)$, nós também podemos encontrar. Até porque, se o computador faz essas contas, faz porque alguém programou para que ele soubesse fazer. E que programa é esse?

10.1 Aproximação para a função $f(x) = e^x$

Resultados de cálculo, que não serão discutidos nesse curso, nos mostram que

$$\lim_{n \rightarrow \infty} \sum_{i=0}^n \frac{x^i}{i!} = e^x \quad , \quad \forall x \in \mathbb{R}. \quad (10.1)$$

Ou seja, para qualquer $x \in \mathbb{R}$, se $n \in \mathbb{N}$ for um número bem grande,

$$\sum_{i=0}^n \frac{x^i}{i!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} \approx e^x.$$

Vale destacar que a convergência é mais rápida para valores de x próximos de zero. Isso significa que, considerando um número fixo de parcelas na equação acima, quanto mais próximo x está de zero, mais perto a soma está de e^x . Mas ela acontece qualquer que seja $x \in \mathbb{R}$, ou seja, mesmo para x distante de zero a soma converge para e^x quando o número de parcelas cresce.

Exemplo 10.1.1 *Vejamos como podemos usar a Equação 10.1 para encontrar uma aproximação para $e^{-1,5}$, sem a ajuda do computador, usando apenas operações de soma e multiplicação.*

Vamos começar encontrando uma aproximação usando $n = 2$:

$$e^{-1,5} = e^{-\frac{3}{2}} \approx 1 + \left(-\frac{3}{2}\right) + \frac{\left(-\frac{3}{2}\right)^2}{2!} = 1 - \frac{3}{2} + \frac{9}{8} = \frac{8 - 12 + 9}{8} = \frac{5}{8} = 0.625$$

E se quisermos uma aproximação mais precisa podemos usar um valor maior de n . Vamos ter que fazer um pouco mais de contas, mas teremos uma aproximação melhor. Por exemplo veja a aproximação para $n = 3$.

$$e^{-1,5} = e^{-\frac{3}{2}} \approx 1 + \left(-\frac{3}{2}\right) + \frac{\left(-\frac{3}{2}\right)^2}{2!} + \frac{\left(-\frac{3}{2}\right)^3}{3!} = \frac{5}{8} - \frac{9}{16} = \frac{10 - 9}{16} = \frac{1}{16} = 0.062$$

Agora para $n = 4$.

$$e^{-1.5} = e^{-\frac{3}{2}} \approx 1 + \left(-\frac{3}{2}\right) + \frac{\left(-\frac{3}{2}\right)^2}{2!} + \frac{\left(-\frac{3}{2}\right)^3}{3!} + \frac{\left(-\frac{3}{2}\right)^4}{4!} = \frac{1}{16} + \frac{27}{128} = \frac{35}{128} \approx 0.2734$$

E para $n = 5$.

$$e^{-1.5} = e^{-\frac{3}{2}} \approx 1 + \left(-\frac{3}{2}\right) + \frac{\left(-\frac{3}{2}\right)^2}{2!} + \frac{\left(-\frac{3}{2}\right)^3}{3!} + \frac{\left(-\frac{3}{2}\right)^4}{4!} + \frac{\left(-\frac{3}{2}\right)^5}{5!} = \frac{35}{128} - \frac{81}{1280} = \frac{269}{1280} \approx 0.2101$$

Se usarmos o computador vamos encontrar a seguinte aproximação para $e^{-1.5} = 0.223130$. Veja que realmente as aproximações calculadas acima melhoraram quando n cresceu.

Se tivermos um computador podemos criar um programa que faça as contas para a gente. Mas como escolher o valor de n ? Podemos verificar o incremento entre duas aproximações consecutivas e parar quando este for relativamente pequeno. Veja como isso será feito no pseudocódigo a seguir.

Entrada: x e *erro*.

Saída: uma aproximação para e^x .

Nome: AproxExp.

- 1 Inicie $i = 1$;
 - 2 Defina $y = x$
 - 3 Inicie $soma = 1 + y$;
 - 4 Se $|y| < erro$, retorne $soma$;
 - 5 Incremente $i = i + 1$;
 - 6 Defina $y = \frac{x^i}{i!}$;
 - 7 Incremente $soma = soma + y$;
 - 8 Volte para a linha 4.
-

Veja que a linha 6 poderia ser alterada para $y = y \frac{x}{i}$. Isso tornaria o código mais eficiente, pois não seria necessário calcular o fatorial em cada iteração do programa.

O pseudocódigo acima simplesmente calcula a série até que o incremento da série, definido por y , seja menor que o erro escolhido pelo usuário e passado como argumento. Garantimos que isso em algum momento acontece uma vez que $\frac{x^n}{n!} \xrightarrow{n \rightarrow \infty} 0$ para todo $x \in \mathbb{R}$.

A escolha do erro é feita pelo usuário. Geralmente se escolhe valores muito pequenos, como por exemplo, $erro = 10^{-3} = 0.001$ ou até menores.

10.2 Aproximação para a função $f(x) = \ln(x)$

Resultados de cálculo, que não serão discutidos nesse curso, nos mostram que

$$\lim_{n \rightarrow \infty} \sum_{i=i}^n (-1)^{i+1} \frac{(x-1)^i}{i} = \ln(x) \quad , \quad \forall 0 < x < 2. \quad (10.2)$$

Ou seja, se $0 < x < 2$, se $n \in \mathbb{N}$ for um número bem grande,

$$\sum_{i=1}^n (-1)^{i+1} \frac{(x-1)^i}{i} = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \dots + (-1)^{n+1} \frac{(x-1)^n}{n} \approx \ln(x).$$

Vale destacar que a convergência é mais rápida para valores de x próximos de 1. Isso significa que, considerando um número fixo de parcelas na equação acima, quanto mais próximo x está de 1, mais perto a soma está de $\ln(x)$. Mas ela acontece qualquer que seja $0 < x < 2$, ou seja, a soma converge para $\ln(x)$ quando o número de parcelas cresce.

Exemplo 10.2.1 *Vejamos como encontrar uma aproximação para $\ln(0.5)$ sem usar calculadora ou computador, usando apenas as operações de soma e multiplicação. Para isso vamos usar a Equação 10.2, atribuindo algum valor para n . Quanto maior o valor de n , melhor será a aproximação. Veja como fica a aproximação para $n = 4$.*

$$\begin{aligned} \ln(0.5) = \ln\left(\frac{1}{2}\right) &\approx \left(\frac{1}{2} - 1\right) - \frac{\left(\frac{1}{2} - 1\right)^2}{2} + \frac{\left(\frac{1}{2} - 1\right)^3}{3} - \frac{\left(\frac{1}{2} - 1\right)^4}{4} = -\frac{1}{2} - \frac{1}{8} - \frac{1}{24} - \frac{1}{64} = \\ &= \frac{-96 - 24 - 8 - 3}{192} = -\frac{130}{192} \approx -0.677 \end{aligned}$$

Se usarmos o computador vamos encontrar o seguinte valor: $\ln(0.5) \approx -0.6931$.

Mas se queremos o valor de $\ln(3)$? Nesse caso não podemos usar a Equação 10.2 substituindo x por 3, pois a convergência só acontece para $0 < x < 2$. Mas podemos usar uma propriedade dos logaritmos para resolver o nosso problema e calcular $\ln(3)$. Veja que

$$\ln\left(\frac{a}{b}\right) = \ln(a) - \ln(b)$$

sempre que $b \neq 0$. Então,

$$\ln\left(\frac{1}{x}\right) = \ln(1) - \ln(x) = -\ln(x) \Rightarrow \ln(x) = -\ln\left(\frac{1}{x}\right)$$

Além disso, se $x > 2 \Rightarrow 0 < \frac{1}{x} < \frac{1}{2} < 2$, logo podemos usar a Equação 10.2 para encontrar uma aproximação para $\ln\left(\frac{1}{x}\right)$. Resumindo, se precisamos encontrar $\ln(x)$ para $x > 2$, então vamos primeiro encontrar uma aproximação para $\ln\left(\frac{1}{x}\right)$ e depois multiplicar por -1, que temos então uma aproximação para $\ln(x)$. Veja um exemplo.

Exemplo 10.2.2 *Encontre uma aproximação para $\ln(3)$ usando a Equação 10.2 com $n = 4$.*

$$\begin{aligned} \ln(3) = -\ln\left(\frac{1}{3}\right) &\approx -\left(\left(\frac{1}{3} - 1\right) - \frac{\left(\frac{1}{3} - 1\right)^2}{2} + \frac{\left(\frac{1}{3} - 1\right)^3}{3} - \frac{\left(\frac{1}{3} - 1\right)^4}{4}\right) = \\ &= -\left(-\frac{2}{3} - \frac{2}{9} - \frac{8}{81} - \frac{16}{324}\right) = \frac{216 + 72 + 32 + 16}{324} = \frac{336}{324} \approx 1.037 \end{aligned}$$

Usando a calculadora chegamos em $\ln(3) \approx 1.0986$.

Vejamos no pseudocódigo a seguir como fica o algoritmo para encontrar uma aproximação para $\ln(x)$. Assim como no caso para a aproximação da exponencial, o algoritmo incrementa n até encontrar uma aproximação boa.

Entrada: x e $erro$.

Saída: uma aproximação para $\ln(x)$.

Nome: AproxLn.

```
1 Se  $x \leq 0$ , pare e retorne mensagem de erro;  
2 Se  $x \geq 2$ , retorne  $-AproxLn(\frac{1}{x}, erro)$ ;  
3 Inicie  $i = 1$ ;  
4 Defina  $y = x - 1$   
5 Inicie  $soma = y$ ;  
6 Se  $|y| < erro$ , retorne  $soma$ ;  
7 Incremente  $i = i + 1$ ;  
8 Defina  $y = \frac{(-1)^{i+1}(x-1)^i}{i}$   
9 Incremente  $soma = soma + y$ ;  
10 Volte para a linha 6.
```

Veja que a linha 8 poderia ser alterada para $y = -y \frac{(x-1)(i-1)}{i}$. Isso tornaria o código mais eficiente.

10.3 Aproximação para a função $f(x) = \sin(x)$

Resultados de cálculo, que não serão discutidos nesse curso, nos mostram que

$$\lim_{n \rightarrow \infty} \sum_{i=0}^n (-1)^i \frac{x^{2i+1}}{(2i+1)!} = \sin(x) \quad , \quad \forall x \in \mathbb{R}. \quad (10.3)$$

Ou seja, para qualquer $x \in \mathbb{R}$, se $n \in \mathbb{N}$ for um número bem grande,

$$(-1)^i \frac{x^{2i+1}}{(2i+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} \approx \sin(x)$$

Vale destacar que a convergência é mais rápida para valores de x próximos de 0. Isso significa que, considerando um número fixo de parcelas na equação acima, quanto mais próximo x está de 0, mais perto a soma está de $\sin(x)$. Mas ela acontece qualquer que seja $x \in \mathbb{R}$, ou seja, a soma sempre converge para $\sin(x)$ quando o número de parcelas cresce.

Atenção: o argumento da função \sin é a medida do ângulo em radianos. Dessa forma, \sin é uma função periódica com período de tamanho 2π .

Exemplo 10.3.1 *Vejamos como encontrar uma aproximação para $\sin(1)$ sem usar calculadora ou computador, usando apenas as operações de soma e multiplicação. Para isso vamos usar a Equação 10.3 com $n = 4$.*

$$\sin(1) \approx 1 - \frac{1^3}{3!} + \frac{1^5}{5!} - \frac{1^7}{7!} + \frac{1^9}{9!} = 1 - \frac{1}{6} + \frac{1}{120} - \frac{1}{5.040} + \frac{1}{362.880} \approx 0,8414710$$

Se usarmos o computador vamos encontrar o seguinte valor: $\sin(1) \approx 0,8414709$. Veja que com $n = 5$ já temos uma aproximação excelente.

No Exemplo 10.3.1 tivemos uma boa aproximação para $\text{sen}(1)$ usando n relativamente baixo, $n = 4$. Isso se deve ao fato de que 1 está razoavelmente próximo de zero. Provavelmente a aproximação não seria tão boa com $n = 4$ para encontrar $\text{sen}(10)$. Mas podemos aproveitar a periodicidade da função seno e aumentar a velocidade de convergência mesmo para valores de x longe de zero. Veja um exemplo.

Exemplo 10.3.2 Usando o computador encontramos $\text{sen}(10) = -0.5440211$. Vamos tentar encontrar esse valor na mão. Primeiro a partir da Equação 10.3 usando $n = 4$ e $x = 10$.

$$\begin{aligned}\text{sen}(10) &\approx 10 - \frac{10^3}{3!} + \frac{10^5}{5!} - \frac{10^7}{7!} + \frac{10^9}{9!} \\ &= 10 - \frac{1.000}{6} + \frac{100.000}{120} - \frac{10.000.000}{5.040} + \frac{1.000.000.000}{362.880} \\ &\approx -1307,46.\end{aligned}$$

Veja que chegamos numa aproximação muito ruim, precisaríamos muito mais parcelas para chegar perto do valor correto de $\text{sen}(10)$.

Veja que, pela periodicidade da função seno, $\text{sen}(10) = \text{sen}(10 - 2\pi) = \text{sen}(3.716815)$. E como 3.716815 está mais perto de zero do que 10, provavelmente a aproximação será melhor usando o mesmo valor de n . Vamos às contas.

$$\begin{aligned}\text{sen}(10) &= \text{sen}(10 - 2\pi) \approx \text{sen}(3.716815) \\ &\approx 3.716815 - \frac{3.716815^3}{3!} + \frac{3.716815^5}{5!} - \frac{3.716815^7}{7!} + \frac{3.716815^9}{9!} \\ &\approx -0,6397173.\end{aligned}$$

Veja como já melhorou! Mas ainda podemos melhorar mais, pois $10 - 2\pi - 2\pi = -2.566371$ está ainda mais perto de zero.

$$\begin{aligned}\text{sen}(10) &= \text{sen}(10 - 2\pi - 2\pi) \approx \text{sen}(-2.566371) \\ &\approx -2.566371 - \frac{-2.566371^3}{3!} + \frac{-2.566371^5}{5!} - \frac{-2.566371^7}{7!} + \frac{-2.566371^9}{9!} \\ &\approx -0,5482006.\end{aligned}$$

O Exemplo 10.3.2 nos mostra que aproveitar a periodicidade da função seno e escolher valores próximos de zero para serem aplicados na Equação 10.3 garante uma boa convergência com menos parcelas. Esse mecanismo será aplicado no pseudocódigo a seguir. Veja no pseudocódigo a seguir como fica o algoritmo para encontrar uma aproximação para $\text{sen}(x)$.

Entrada: x e erro .

Saída: uma aproximação para $\text{sen}(x)$.

Nome: AproxSeno.

- 1 Se $|x| > \pi$, retorne $\text{AproxSeno}(x - 2\pi, \text{erro})$;
- 2 Inicie $i = 0$;
- 3 Defina $y = x$
- 4 Inicie $\text{soma} = y$;

```
5 Se  $|y| < erro$ , retorne  $soma$ ;  
6 Incremente  $i = i + 1$ ;  
7 Defina  $y = \frac{(-1)^i x^{2i+1}}{(2i+1)!}$   
8 Incremente  $soma = soma + y$ ;  
9 Volte para a linha 6.
```

Veja que a linha 8 poderia ser alterada para $y = -y \frac{x^2}{(2i+1) \times 2i}$. Isso tornaria o código mais eficiente.

Exercícios - 10ª Semana

10.1 A partir da Equação 10.1 encontre uma aproximações para e^1 e e^3 usando $n = 4$. Faça as contas na mão.

10.2 a) Implemente uma função que recebe como entrada $x \in \mathbb{R}$ e $n \in \mathbb{N}$ e retorna o polinômio $\sum_{i=0}^n \frac{x^i}{i!}$ avaliado em x . Vamos chamar essa função de `pol_exp(x,n)`.

b) Digite o código a seguir para ver como esse polinômio se aproxima da função exponencial conforme seu grau cresce.

```
> plot(exp,-4,4)
> grid()
> segments(x0=0,y0=0,x1=0,y1=150,lty=2)
> curve(pol_exp(x,n=2),add=T,col="violet")
> curve(pol_exp(x,n=3),add=T,col="red")
> curve(pol_exp(x,n=4),add=T,col="blue")
> curve(pol_exp(x,n=5),add=T,col="green")
```

Veja que a aproximação melhora quanto mais perto de 0 for o ponto avaliado ou quando maior for o valor de n .

10.3 a) Implemente o pseudocódigo visto em sala de aula que recebe como entrada $x \in \mathbb{R}$ e um erro e retorna uma aproximação para e^x .

b) Use a função implementada acima e encontre aproximações para e , e^{-1} , e^3 , \sqrt{e} e $e^{7.3}$. Compare os resultados com os valores fornecidos pela função `exp` do R.

c) Refaça a função implementada em 3a de forma que além da aproximação para e^x ela também retorne o valor de n usado para realizar essa aproximação. Refaça também os testes sugeridos em 3b.

d) Usando a função implementada no item 3a, isto é, a aproximação para e^x , implemente um outra função que recebe como entrada x e um erro e retorna uma aproximação para $e^{-x^2/2}$.

10.4 a) A partir da Equação 10.2 use $n = 3$ e encontre uma aproximações para $\ln(\frac{1}{10})$, $\ln(\frac{3}{5})$ e $\ln(4)$. Faça as contas na mão.

b) Sem olhar os valores exatos quais das aproximações encontradas no item 4a você acha que é mais precisa, a aproximação para $\ln(\frac{1}{10})$ ou para $\ln(\frac{3}{5})$? Por quê?

10.5 a) Implemente uma função que recebe como entrada $x \in \mathbb{R}$ e $n \in \mathbb{N}$ e retorna o polinômio $\sum_{i=0}^n (-1)^{i+1} \frac{(x-1)^i}{i}$ avaliado em x . Vamos chamar essa função de `pol_ln(x,n)`.

b) Digite o código a seguir para ver como esse polinômio se aproxima da função \ln conforme seu grau cresce.

```
> plot(log,0,4)
> grid()
> segments(x0=1,y0=-4,x1=1,y1=10,lty=2)
> curve(pol_ln(x,n=2),add=T,col="violet")
> curve(pol_ln(x,n=3),add=T,col="red")
> curve(pol_ln(x,n=4),add=T,col="blue")
> curve(pol_ln(x,n=5),add=T,col="green")
```

Veja que a aproximação melhora quanto mais perto de 1 for o ponto avaliado ou quando maior for o valor de n .

- 10.6 a) Implemente o pseudocódigo visto em sala de aula que recebe como entrada um número real positivo x e um *erro* e retorna uma aproximação para $\ln(x)$.
- b) Use a função implementada acima e encontre aproximações para $\ln(0.1)$, $\ln(2)$, $\ln(10)$ e $\ln(3.8)$. Compare os resultados com os valores fornecidos pela função `ln` do R.
- c) Implemente agora uma função que retorna o logaritmo de x em qualquer base, ou seja, essa nova função recebe como entrada x , b e *erro* e retorna uma aproximação para $\log_b(x)$. Para isso lembre-se da seguinte propriedade:

$$\log_b(x) = \frac{\log_a(x)}{\log_a(b)} \quad \forall \quad a, b, x \in \mathbb{R}^+.$$

Dica: essa nova função deve chamar a função implementada em 7a e usar a propriedade acima considerando $a = e$.

- 10.7 a) Implemente o pseudocódigo visto em sala de aula que recebe como entrada um número real positivo x e um *erro* e retorna uma aproximação para $\sin(x)$. Aproveite a periodicidade da função e, a partir de uma chamada recursiva, simplifique o argumento até que ele esteja entre $-\pi$ e π .
- b) Use a função implementada acima e encontre aproximações para $\sin(2)$, $\sin(25)$, $\sin(50^\circ)$ e $\sin(\pi/3)$. Compare os resultados com os valores fornecidos pela função `sin` do R.
- c) Implemente agora uma função que recebe como entrada x e *erro* e retorna $\cos(x)$. Para isso perceba que a função cosseno é a função seno deslocada de $\pi/2$, ou seja,

$$\cos(x) = \sin(x + \pi/2).$$