Curso de Cálculo

P. Kantek

17 de dezembro de 2021

Sumário

1 Revisão **7**
1.1 Fundamentos 7
1.2 Demonstração da fórmula de Baskara 8
1.3 Da álgebra e da geometria 9
1.4 Estratégia para extrair funções 9
1.5 Problemas geométricos de duas dimensões 9
1.6 Problemas geométricos tridimensionais 12
1.7 Problemas de economia e administração 15
1.8 Problemas suplementares 17
1.9 Da Geometria Analítica 17
1.9.1 Localização no plano 17
1.9.2 Distância entre 2 pontos 17
1.9.3 Círcunferência 18
1.9.4 Reta 18
1.9.5 Paralelismo e perpendicularidade 18

2 Instrumentos **19**
2.1 Maple 19
2.1.1 Álgebra em maple 20
2.1.2 Programação em Maple 22
2.2 Python 24
2.2.1 Versão 25
2.2.2 sob Windows 25
2.2.3 Alo Mundo 26
2.2.4 Winpython 27
2.2.5 Ipython 27
2.2.6 Python na WEB 27
2.2.7 Variáveis, sua entrada e saída 28
2.2.8 Tipos 29
2.2.9 Nomes 29
2.2.10 Expressões aritméticas 30
2.2.11 Saída de Dados 31
2.2.12 Entrada de dados 31
2.2.13 Um programa completo usando input e print 32
2.2.14 Variáveis Lógicas 32
2.2.15 Expressões Lógicas 33
2.2.16 Comandos condicionais 33
2.2.17 Repetições 34
2.2.18 while 34
2.2.19 break 35
2.2.20 Repetição aninhada 35
2.2.21 Continue 36
2.2.22 For 36
2.3 sympy 36
2.4 APL 43
2.4.1 Dyalog APL: Qualidades e novidades 43
2.4.2 Primitivas 44
2.5 www.mathway.com 85

2.6 Tudo junto . . . 86

3 Cálculo **87**
3.1 Problema da área . . . 87
3.2 Problema da tangente . . . 87
3.3 Velocidade . . . 88

4 Limites **91**
4.1 Limite de uma sequência . . . 91
4.2 Soma de uma série . . . 92
4.3 Limites . . . 92
4.4 Limites Laterais . . . 93
4.5 Definição precisa . . . 93
4.6 Continuidade . . . 94

5 Derivadas **95**
5.1 Derivada e taxa de variação . . . 95
5.1.1 Mais taxas de variação . . . 96

6 Regras de derivação **97**
6.1 A derivada como função . . . 97
6.2 Derivada do seno . . . 97
6.3 Derivada de ordem superior . . . 99
6.4 Movimento em linha reta . . . 101
6.5 Aplicações à ciência e engenharia . . . 103
6.6 Administração e economia . . . 104
6.7 Problemas suplementares . . . 105
6.8 Função constante . . . 106
6.9 Função potência . . . 106
6.10 Derivada de função multiplicada por constante . . . 107
6.11 Derivada de soma de funções . . . 107
6.12 Derivada de função exponencial . . . 107
6.13 Regra do produto . . . 108
6.14 Regra do quociente . . . 108
6.15 Derivadas de funções trigonométricas . . . 108
6.16 Regra da Cadeia . . . 108
6.17 Derivação implícita . . . 109
6.18 Derivadas de funções logarítmicas . . . 109
6.19 Número e como um limite . . . 109
6.20 Em resumo . . . 110

7 Derivadas Relacionadas **111**
7.1 Problemas suplementares . . . 117

8 Aplicações **119**
8.1 Problemas de otimização . . . 119
8.2 O que as derivadas dizem de uma função . . . 120
8.3 Formas indeterminadas e a regra de L'Hospital . . . 121
8.4 Problemas de otimização . . . 122
8.5 Método de Newton-Raphson . . . 123

9 Integrais **127**
9.1 Cálculo de áreas e distâncias . . . 127
9.2 Integral definida . . . 127
9.2.1 Relembrando somatórias . . . 127
9.2.2 Ponto Médio . . . 128
9.3 Propriedades da integral definida . . . 128
9.4 Teorema Fundamental do Cálculo . . . 128
9.4.1 Integrais indefinidas . . . 128
9.5 Regra da substituição . . . 129
9.6 Integração por partes . . . 129
9.7 Integração aproximada . . . 130
9.7.1 Regra do Trapézio . . . 130
9.8 Regra de Simpson . . . 131

9.9 Aplicações: problemas de área 131
9.10 Volumes de sólidos de revolução 135
9.11 Problemas Suplementares 139
9.12 Integrais Impróprias 140
9.13 Logaritmo natural como integral 140

10 Máximos e mínimos **141**
10.1 Problemas suplementares 156

11 Funções Trigonométricas **159**
11.1 Problemas de Variação 160
11.2 Problemas de máximos e mínimos 162
11.3 Problemas suplementares 165

12 Funções exponenciais **167**
12.1 Crescimento e decaimento exponenciais 167
12.1.1 Carbono 14 170
12.1.2 Juros compostos contínuos 170
12.1.3 Modelos exponenciais adicionais 171
12.2 Problemas Suplementares 172

13 Equações Diferenciais **175**
13.1 Equação diferencial linear 176

14 Derivadas parciais **177**
14.1 Interpretação geométrica da derivada parcial 178
14.2 Derivadas de ordem superior 178
14.2.1 Teorema de Clairaut 178

15 Integrais Duplas **179**
15.1 Coordenadas esféricas 180

16 Cálculo vetorial **183**
16.1 Gradiente 183
16.2 Rotacional 183

17 Bibliografia **185**

Capítulo 1

Revisão

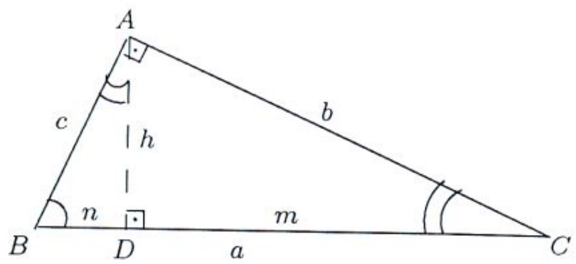
1.1 Fundamentos

Começa-se demonstrando o Teorema de Pitágoras, pelo protagonismo que ele exerce sobre tudo o que virá.

Teorema de Pitágoras

Há dezenas (centenas ?) de demonstrações, gráficas, geométricas, trigonométricas, basta procurar na Internet por elas. Vai-se mostrar aqui, talvez a mais simples e completa.

Suponha-se um triângulo $\triangle ABC$ reto em A e defina-se sua altura que intercepta a hipotenusa no ponto D conforme a figura



Tal altura acaba de criar 2 novos triângulos, a saber $\triangle ABD$ e $\triangle ADC$ também retos, pela definição da altura de um triângulo.

Estes 3 triângulos $\triangle ABC$, $\triangle ABD$ e $\triangle ADC$ são semelhantes, uma vez que os 3 são retângulos e compartilham 1 ângulo 2 a 2. Agora

Da semelhança entre $\triangle ABD$ e $\triangle ABC$ tem-se

$$\frac{c}{a} = \frac{n}{c}$$

Da semelhança entre $\triangle ADC$ e $\triangle ABC$ tem-se

$$\frac{b}{a} = \frac{m}{b}$$

Multiplicando em cruz ambas expressões fica

$$c^2 = an \qquad e \qquad b^2 = am$$

Somando essas duas expressões fica

$$c^2 + b^2 = an + am$$

Pondo em evidência a tem-se

$$c^2 + b^2 = a(n + m)$$

Mas, como $a = n + m$ pode-se substituir e fica

$$c^2 + b^2 = a.a = a^2$$

CQD

1.2 Demonstração da fórmula de Baskara

O objetivo aqui é resolver uma equação de segundo grau, cuja forma é $ax^2 + bx + c = 0$. Note, que devido a presença do x^2 não se consegue facilmente isolar o x como se faria em uma equação linear.

Começando Com a equação acima descrita $ax^2 + bx + c = 0$

Colocando a em evidência para as duas primeiras parcelas, fica

$$a\left(x^2 + \frac{bx}{a}\right) + c = 0$$

Somando e subtraindo a mesma coisa Como se sabe, a equação não se altera. O objetivo é construir um quadrado perfeito. O termo a somar é $(\frac{b}{2a})^2$ e fica

$$a\left(x^2 + \frac{bx}{a} + (\frac{b}{2a})^2 - (\frac{b}{2a})^2\right) + c = 0$$

Fatorando Lembrando que $(a+b)^2 = a^2 + 2ab + b^2$ faz-se

$$a\left((x + \frac{b}{2a})^2 - (\frac{b}{2a})^2\right) + c = 0$$

aplicando o a na segunda parcela , fica

$$a\left((x + \frac{b}{2a})\right)^2 - \frac{b^2}{4a} + c = 0$$

Passando para o segundo membro , fica

$$a\left(x + \frac{b}{2a}\right)^2 = \frac{b^2}{4a} - c$$

dividindo tudo por a , fica

$$\left(x + \frac{b}{2a}\right)^2 = \frac{b^2}{4a^2} - \frac{c}{a}$$

mínimo múltiplo comum no segundo termo, que é $4a^2$

$$\left(x + \frac{b}{2a}\right)^2 = \frac{b^2 - 4ac}{4a^2}$$

extraindo a raiz

$$x + \frac{b}{2a} = \sqrt{\frac{b^2 - 4ac}{4a^2}}$$

e agora

$$x + \frac{b}{2a} = \frac{\sqrt{b^2 - 4ac}}{\sqrt{4a^2}}$$

e

$$x + \frac{b}{2a} = \frac{\sqrt{b^2 - 4ac}}{2a}$$

passando a segunda parcela do primeiro termo para o segundo termo da igualdade fica

$$x = \frac{\sqrt{b^2 - 4ac}}{2a} - \frac{b}{2a}$$

mmc

$$x = \frac{\sqrt{b^2 - 4ac} - b}{2a}$$

Reorganizando e lembrando que ao extrair uma raiz quadrada deve-se considerar tanto o resultado positivo quanto o negativo fica

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

CQD

1.3 Da álgebra e da geometria

1.4 Estratégia para extrair funções

A parte mais importante para extrair funções é ler e entender o problema. Depois de entendido o problema surgem 3 etapas, a saber

Etapa 1 Desenhe um diagrama. Rotule as quantidades conhecidas e as desconhecidas.
Etapa 2 Escreva as esquações representando as quantidades a serem expressas como funções.
Etapa 3 Use as restrições dadas no problema para eliminar todas as quantidades EXCETO uma.

Exemplo 3 A soma de 2 números é 40. Expresse seu produto como uma função de um dos números.

Etapa 1 Em muitos problemas numéricos um diagrama não é esperado. Vamos rotular os dois números como x e y.
Etapa 2 Pretende-se expressar o produto como uma função. Então $P = xy$.
Etapa 3 Já que $x + y = 40$ então $y = x - 40$ substuindo-se isto na função do produto fica

$$P(x) = x(40 - x)$$

$$P(x) = 40x - x^2$$

Exemplo 4 O produto de dois números é 32. Ache a função que representa a soma de seus quadrados.

Etapa 1 Vamos identificar os dois números como x e y.
Etapa 2

$$S = x^2 + y^2$$

Etapa 3 Já que $xy = 32$ então $y = \frac{32}{x}$ substuindo-se isto na função do produto fica

$$S(x) = x^2 + \left(\frac{32}{x}\right)^2$$

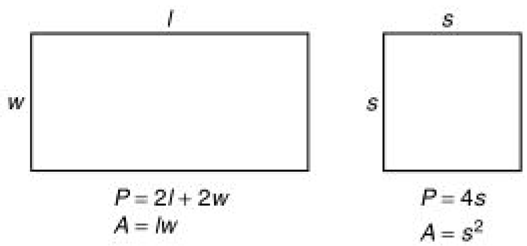
1.5 Problemas geométricos de duas dimensões

Muitos problemas geométricos são compostos de retângulos, triângulos e círculos. Portanto é adequado relembrar as fórmulas de perímetros e áreas destas figuras fundamentais. Um retângulo de comprimento l e largura w tem um perímetro que é a soma do comprimento de seus quatro lados. A área é o produto de seu comprimento pela sua largura.

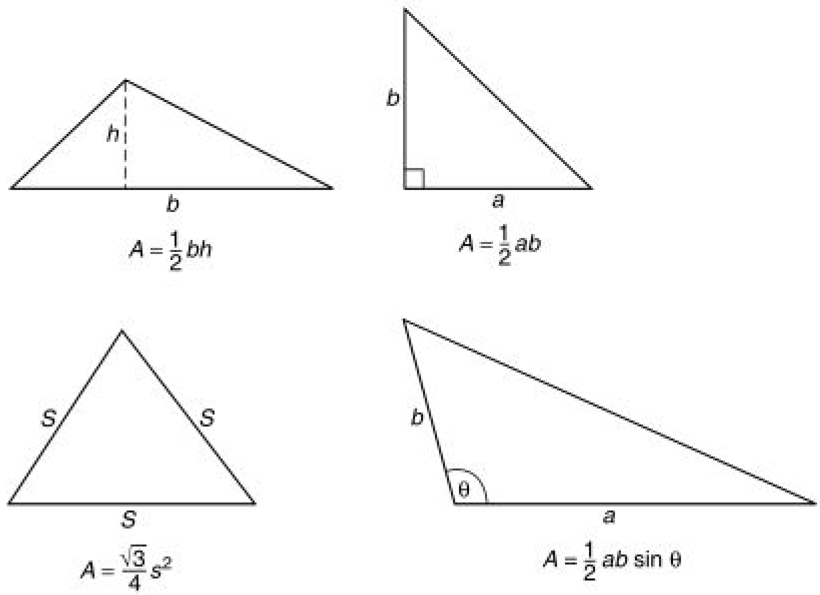
$$\boxed{\begin{array}{l} P = 2l + 2w \\ A = lw \end{array}}$$

Um caso especial ocorre quando l e w são iguais. A figura geométrica resultante é o quadrado, cujo lado é s.

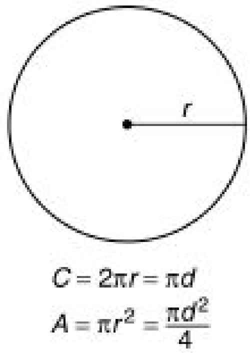
$$\boxed{\begin{array}{l} P = 4s \\ A = s^2 \end{array}}$$



Um triângulo de base b e altura h tem uma área de $\frac{1}{2}bh$. Casos especiais incluem triângulos retos e triângulos equiláteros. O perímetro de um triângulo é a soma de seus 3 lados.

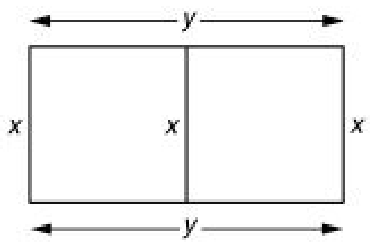


Um círculo é medido pelo seu raio r. O perímetro do círculo é conhecido como sua circunferência. As vezes o círculo é identificado pelo seu diâmetro que vem a ser o dobro do raio ($d = 2r$).



Exemplo 4 Um fazendeiro tem 1500 pés de cerca guardado em seu celeiro. Ele pretende cercar um pasto retangular dividido em 2 regiões por uma cerca perpendicular a um lado e paralela ao outro no meio do pasto. Expresse a área do pasto cercado como uma função de seu comprimento w. Qual o domínio da função ?

Etapa 1 Desenhando um diagrama simples e identificando as dimensões do retângulo



Etapa 2 Expressemos a área em função das variáveis x e y. Note que a área é afetada pelas cercas externas apenas. A cerca interna não influência nada.

Etapa 3 Usa-se a restrição dos 1500 pés de cerca para obter a relação entre x e y.

$$3x + 2y = 1500$$

$$2y = 1500 - 3x$$

$$y = 750 - \frac{3}{2}x$$

Finalmente, substitui-se esta expressão para y na equação da área obtida na etapa 2

$$A = xy$$

$$A = x\left(750 - \frac{3}{23}x\right)$$

$$A(x) = 750x - \frac{3}{2}x^2$$

Matematicamente o domínio de $A(x)$ é o conjunto dos reais. Entretanto, neste problema, como em todos os geométricos, dimensões negativas não são aceitas. Ainda que $x = 0$ possa parecer irrealista, nós geralmente permitimos que um retângulo possa ter a dimensão 0 em uma de seus lados, o que acarretará área também zero. Este caso é chamado retângulo degenerado. Já que o perímetro é fixo, y cresce quando x diminui e vice-versa. Assim, o maior valor de x será obtido quando y tiver o menor valor ou $y = 0$.

$$3x + 2y = 1500$$

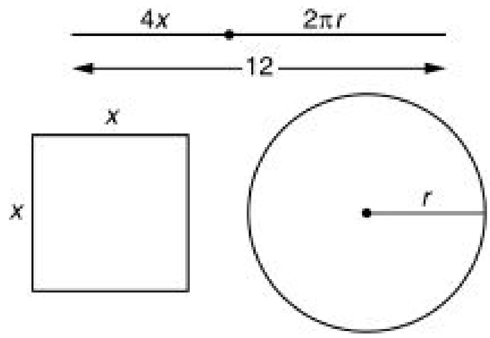
$$3x = 1500 \quad (y = 0)$$

$$x = 500$$

A função descrevendo o pasto é $A(x) = 750x - \frac{3}{2}x^2$ com $0 \leq x \leq 500$.

Exemplo 6 Um rolo de arame tem 12 polegadas e deve ser usado para formar um quadrado e/ou um círculo. Determine a função que expressa a área combinada das duas figuras. Qual é o domínio ?

Etapa 1 Seja x o lado do quadrado e r o raio do círculo. Podemos expressar a área como



Etapa 2

$$A = x^2 + \pi r^2$$

Etapa 3 Considerando que as 2 figuras devem ter perímetro igual a 12 polegadas

$$4x + 2\pi r = 12$$

$$2\pi r = 12 - 4x$$

$$r = \frac{12 - 4x}{2\pi} = \frac{6 - 2x}{\pi}$$

Substituindo r em termos de x na etapa 2 fica

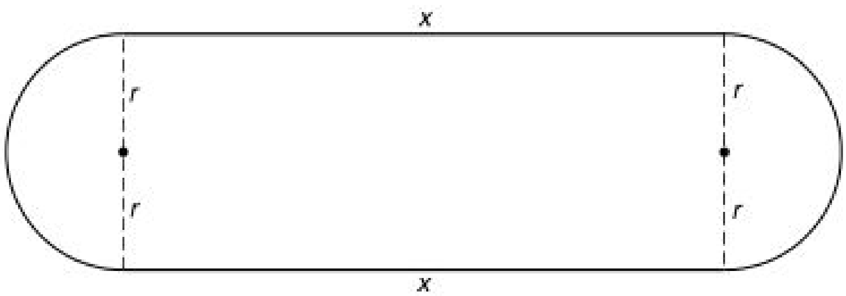
$$A(x) = x^2 + \pi\left(\frac{6 - 2x}{\pi}\right)^2$$

$$= x^2 + \frac{(6 - 2x)^2}{\pi}$$

Se todo o arame for usado para formar o círculo, então $x = 0$. Se todo o arame for usado para o quadrado, $4x = 12$ e $x = 3$. Daqui o domínio é $0 \leq x \leq 3$.

Exemplo 7 Uma pista de corrida de 1 milha tem dois finais semicirculares contectados por linhas retas. Expresse a área fechada pela trilha como uma função do raio semicircular. Determine o domínio.

Etapa 1



Etapa 2 A área procurada é formada por um retângulo de lados x e $2r$ e dois semicírculos de raio r cujas áreas combinadas são πr^2. Então

$$2x + 2\pi r = 1$$
$$2x = 1 - 2\pi r$$
$$x = \frac{1 - 2\pi r}{2}$$

Etapa 3 Note que cada arco semicírcular tem comprimento πr. Juntos eles formam um circulo completo cuja circunferência é $2\pi r$ Substituindo este valor na equação da etapa 2

$$A(r) = 2r\left(\frac{1 - 2\pi r}{2}\right) + \pi r^2$$
$$= r - 2\pi r^2 + \pi r^2$$
$$A(r) = r - \pi r^2$$

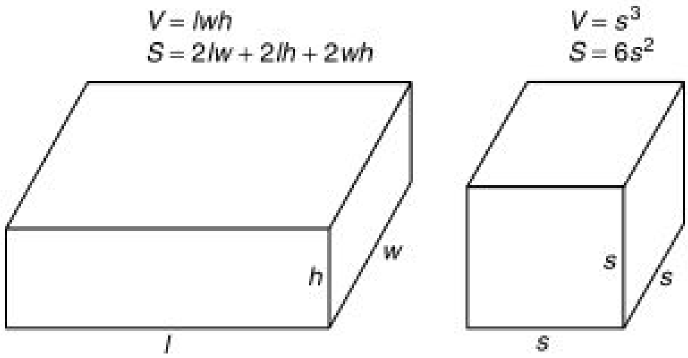
Já que r não pode ser negativo $r \geq 0$. Já o perímetro da trilha está fixado, então o valor máximo de r ocorre quando $x = 0$. Então

$$2x + 2\pi r = 1$$
$$2\pi r = 1 \quad (x = 0)$$
$$r = \frac{1}{2\pi}$$

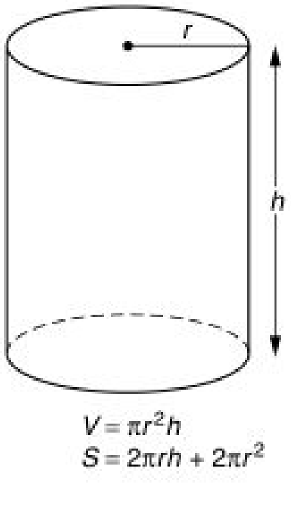
e o domínio é $0 \leq r \leq \frac{1}{2\pi}$

1.6 Problemas geométricos tridimensionais

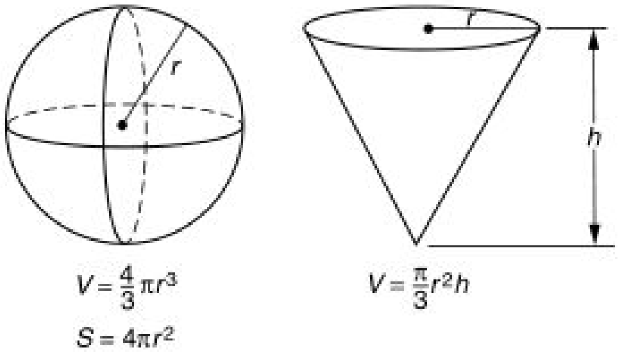
Muitos problemas envolvem caixas, cilindros, esferas e cones. Uma caixa tem um volume igual ao produto de suas 3 dimensões. A área de uma caixa fechada é a soma das áreas de seus 6 lados Uma caixa aberta não tem tampa, seu volume é o mesmo da caixa fechada, mas sua superfície envolve apenas 5 áreas. Um cubo é uma caixa com as arestas de mesmo comprimento



Um cilindro de altura h e raio r tem volume de $\pi r^2 h$ e superfície lateral $2\pi rh$. Um jeito fácil de lembrar disto é multiplicar a área e a circunferência de um círculo por h. A área total de um cilindro é $2\pi rh + 2\pi r^2$

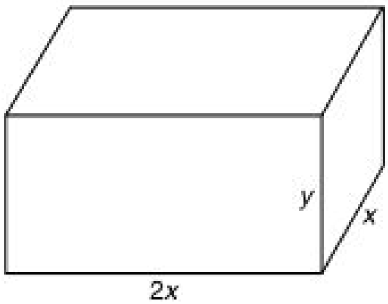


Já as esferas e cones tem a seguinte formulação



Exemplo 8 Uma caixa fechada tem base duas vezes mais comprida que a largura. Se o seu volume é de $100cm^3$ expresse a sua área de superfície como uma função de sua largura.

Etapa 1 Já que a base da caixa é um retângulo com comprimento igual a duas vezes a largura, pode-se representar suas dimensões por x e $2x$. A altura, entretanto deve ser representada por outra variável.



Etapa 2 A área da caixa S é a soma das áreas de seus 6 lados. A parte superior e inferior tem área $2x^2$ cada uma , a frente e o fundo tem área $2xy$ cada uma. Os lados esquerda e direita tem cada um área de xy

$$S = 2x^2 + 2x^2 + 2xy + 2xy + xy + xy$$

$$S = 4x^2 + 6xy$$

Etapa 3 O volume da caixa ($l \times w \times h = 2x^2y$) é de $100cm^3$

$$2x^2y = 100$$

$$y = \frac{50}{x^2}$$

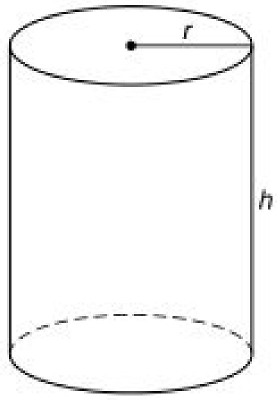
Substituindo este resultado na etapa 2

$$S(x) = 4x^2 + 6x\left(\frac{50}{x^2}\right)$$

$$S(x) = 4x^2 + \frac{300}{x}$$

Exemplo 9 Um cilindro com uma base circular tem uma área total de $64m^2$. Expresse seu volume como uma função de seu raio r.

Etapa 1



Etapa 2

$$V = \pi r^2 h$$

Etapa 3 A área do cilindro é 64. Resolvendo h em termos de r

$$2\pi rh + 2\pi r^2 = 64$$

$$2\pi rh = 64 - 2\pi r^2$$

$$h = \frac{64 - 2\pi r^2}{2\pi r}$$

$$h = \frac{32 - \pi r^2}{\pi r}$$

Substituindo o resultado cima na etapa 2

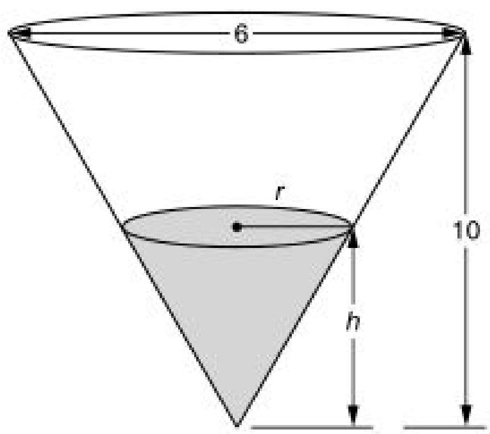
$$V(r) = \pi r^2 \left(\frac{32 - \pi r^2}{\pi r} \right)$$

$$= r(32 - \pi r^2)$$

$$V(r) = 32r - \pi r^3$$

Exemplo 10 Água é armazenada em um tanque com a forma de um cone invertido de altura $10m$ e diâmetro de $6m$. Expresse o volume de água no tanque como uma função da altura h do nível de água.

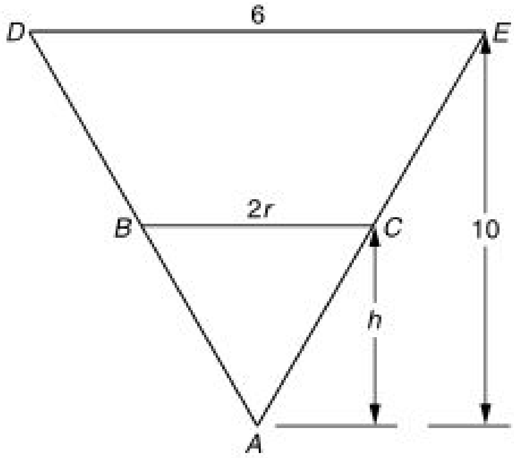
Etapa 1



Etapa 2 A água tem a forma de um cone dentro deste tanque. Seu volume V é

$$V = \frac{\pi}{3} r^2 h$$

Etapa 3 Para obter a relação entre r e h usa-se a geometria básica. Vendo o problema de uma perspectiva bidimensional pode-se observar 2 triângulos similares $\triangle ABC$ e $\triangle ADE$



$$\frac{2r}{h} = \frac{6}{10}$$
$$20r = 6h$$
$$r = \frac{3h}{10}$$

Substituindo a equação do volume na etapa 2, obtém-se

$$V(h) = \frac{\pi}{3}\left(\frac{3h}{10}\right)^2 h$$
$$= \frac{\pi}{3} . \frac{9h^2}{100} . h$$
$$V(h) = \frac{3\pi}{100} h^3$$

1.7 Problemas de economia e administração

Geralmente estes problemas lidam com dinheiro. *Receita* é o dinheiro obtido por uma empresa quando vende seus produtos. *Custo* é quanto a empresa precisa pagar por salários, material, aluguéis e assim por diante. *Lucro* é a diferença entre receita e custo. Valor negativo aqui implica em prejuizo.

Exemplo 11 Uma máquina pode produzir 12 potes de argila por hora. Custa \$ 750 para preparar a máquina e \$ 6 por hora de uso da máquina. Cada pote de argila exige \$ 2 de argila. Se cada pote pode ser vendido por \$ 10 expresse a receita, custos e lucro produzindo x potes como uma função do tempo t.

Etapa 1 Seja x o número de potes produzidos e t o número de horas necessárias para produzir x potes. Sejam R, C e L a receita, custo e lucro respectivamente.

Etapa 2 Já que cada figura vende-se por 10, tem-se

$$R = 10x$$

O custo consiste de 3 partes: Um custo fixo de 750, o custo de usar a máquina por t horas é de $6t$ e o custo da argila é de $2x$, então

$$C = 750 + 6t + 2x$$

Já que o lucro é $R - C$

$$L = 10x - (750 + 6t + 2x)$$
$$= 8x - 6t - 750$$

Etapa 3 Já que a cada hora podem ser produzidos 12 potes, então $x = 12t$ e substituindo

$$R = 10x = 10(12t)$$
$$R(t) = 120t$$
$$C = 750 + 6t + 2x$$
$$= 750 + 6t + 2(12t)$$
$$= 750 + 6t + 24t$$

$$C(t) = 750 + 30t$$

$$L = 8x - 6t - 750$$

$$= 8(12t) - 6t - 750$$

$$= 96t - 6t - 750$$

$$L(t) = 90t - 750$$

Exemplo 12 Um ônibus de turismo tem 80 lugares. A experiência mostra que quando o passeio custa 300 $ por vaga, todos os lugares do ônibus são vendidos. Para cada incremento de 10 $ 2 lugares deixam de ser vendidos. Ache a função que representa a receita de um único passeio.

Etapa 1 Neste tipo de problema é conveniente sugerir x como o número de incrementos de 10 sobre o preço básico de 300. Então, por exemplo, se $x = 2$ o preço é 320. Usar-se-á n para representar o número de lugares vendidos e p o preço por assento.
Etapa 2 A receita R é o produto do número de assentos vendidos e o preço por assento ou vaga.

$$R = np$$

Etapa 3 Para cada unidade de incremento x, n diminui de 2 e p incrementa de 10.

$$n = 80 - 2x$$

$$p = 300 + 10x$$

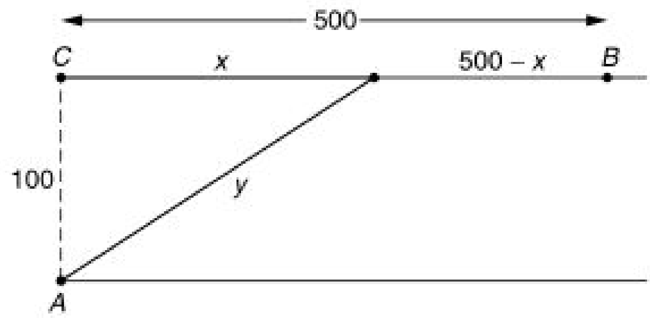
substituindo na etapa 2

$$R = np = (80 - 2x)(300 + 10x)$$

$$R(x) = 24000 + 200x - 20x^2$$

Exemplo 13 Um rio tem $100m$ de largura. A companhia local de telefonia precisa passar um cabo entre os pontos A e B. A está em um lado do rio e B do outro lado, a uma distância de $500m$ conforme a figura. Custa $3/m$ extender o cabo sob o rio enquanto custa $2/m$ extender o cabo em terra firme. Determine o custo de extender o cabo entre A e B.

Etapa 1 Seja x o número de metros entre C (diretamente em frente a A) onde o cabo emerge da água enquanto y indica a distância em metros que o cabo ficará submerso



Etapa 2 O custo total é igual a soma entre o custo embaixo do rio e o custo em terra firme

$$C = C_{terra} + C_{rio}$$

$$= 2(500 - x) + 3y$$

$$= 1000 - 2x + 3y$$

Etapa 3 Pelo Teorema de Pitágoras, $y = \sqrt{x^2 + 100^2}$ então o custo é

$$C(x) = 1000 - 2x + 3\sqrt{x^2 + 10.000} \quad 0 \leq x \leq 500$$

1.8 Problemas suplementares

- 1. A diferença entre 2 números é 15. Expresse o seu produto como uma função do menor número, x.
- 2. Um retângulo tem área de 200 m^2. Expresse o seu perímetro como uma função de sua largura.
- 3. Carina quer cercar um jardim retangular de vegetais e subdividí-lo em 3 regiões usando 2 cercas adicionais, paralelas ao lado x do retângulo. A área total cercada é de $1000m^2$. Expresse a quantidade total de cerca como uma função de x.
- 4. Um retângulo deve ser inscrito em um semicírculo de raio 10, com a base do retângulo repousando ao longo da base do semicírculo. Expresse a área do retângulo como uma função de sua largura e determine o seu domínio.
- 5. Um caixa aberta deve ser construída a partir de uma peça retangular de metal de $8 \times 12m$, cortando-se peças quadradas de lado xm de cada um dos quatro cantos e dobrando os lados. Expresse o volume da caixa resultante em função de x.
- 6. Uma janela de igreja deve ter a forma de um retângulo encimada por um semicírculo. Se o perímetro da janela é de 100 polegadas, expresse a área como uma função do raio semircicular r.
- 7. Uma caixa aberta tem uma base quadrada. Se a sua superfície é de $200cm^2$ expresse seu volume como função da areasta da base x.
- 8. Um cilindro é inscrito em uma esfera de raio 10. Expresse o volume e a superfície como uma função de sua altura h.
- 9. Se 500 macieiras estão plantadas em um pomar, cada macieira produz 800 maçãs. Para cada árvore adicional plantada, o número de frutas produzidas por árvore diminui em 20 unidades. Ache a função que representa o número total de maçãs produzidas no pomar.
- 10. Custa 800\$produzir um certo modelo de computador pessoal. Os custos fixos da companhia são de 2000\$ por semana. O preço de venda do computador é de 1500\$, mas como incentivo a companhia aceita diminuir o preço de cada máquina em 10\$ para cada computador vendido acima de um limite de 10. (Assim, se 13 computadores são vendidos, cada um custará 1470\$). Expresse o lucro semanal da companhia como uma função do número de computadores vendidos.

As soluções estão no livro original, às págs. 21 a 26.

1.9 Da Geometria Analítica

1.9.1 Localização no plano

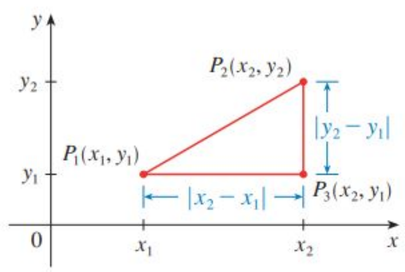
Todos os pontos de um plano podem ser localizados por 2 números reais ordenados. Define-se um ponto de origem e nele desenham-se 2 eixos. Eles – em tese – não precisam ser ortogonais, mas fica mais fácil assim. O ponto individualizado pelas medidas (x, y) é localizado traçando-se a $reta_1$ paralela ao eixo Y e passando pelo ponto x marcado no eixo X. Traça-se a $reta_2$ paralela ao eixo X e passando pelo ponto y marcado no eixo Y. Na intersecção das duas retas está o ponto caracterizado por (x, y).

A primeira medida é conhecida como coordenada-x ou ordenada do ponto. A segunda é conhecida como coordenada-y ou abcissa do ponto. Este sistema de coordenadas é conhecido como cartesiano em homenagem a René Descartes (1596-1650), ainda que outro francês, Pierre de Fermat (1601-1665), andou desenvolvendo os princípios da geometria analítica antes do que Descartes.

Os eixos X e Y dividem o plano cartesiano em 4 quadrantes. O primeiro quadrante é aquele nos quais as ordenadas e as abcissas são positivas.

1.9.2 Distância entre 2 pontos

Para achar a distância entre 2 pontos P_1 e P_2 usa-se o Teorema de Pitágoras, como se pode ver em



Aqui, o triângulo vermelho (retângulo por construção), tem como dois catetos os valores $|x_2 - x_1|$ e $|y_1 - y_2|$, a hipotenusa é a distância procurada, então
$d^2 = (x_2 - x_1^2) + (y_1 - y_2)^2$ Note que o módulo foi retirado, já que a expressão está elevada ao quadrado. Extraindo as raizes, fica

$$d = \sqrt{(x_2 - x_1)^2 + (y_1 - y_2)^2}$$

que é a fórmula da distância entre 2 pontos.

1.9.3 Círcunferência

Uma circunferência é o lugar geométrico de todos os pontos que estão equidistantes de um ponto dado. Este ponto dado é conhecido como o centro da circunferência e a menor distância entre qualquer ponto e o centro é conhecida como raio.

Generalizando a fórmula da distância entre 2 pontos, e usando um ponto como centro (x_r, y_r) e um ponto genérico como (x, y), a equação da circunferência é $r = \sqrt{(x - x_r)^2 + (y - y_r)^2}$ e elevando esta equação ao quadrado, obtém-se a fórmula final:

$$(x - x_r)^2 + (y - y_r)^2 = r^2$$

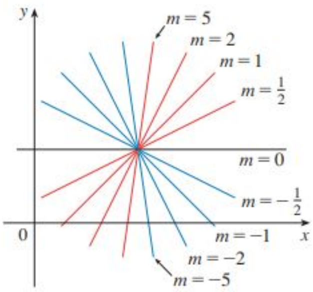
Em particular, a equação da circunferência com centro na origem (o ponto 0,0) e raio r é $x^2 + y^2 = r^2$.

1.9.4 Reta

Uma reta é definida por 2 de seus pontos. Supondo que esses dois pontos sejam $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$ define-se inclinação de uma reta ao valor m cuja fórmula é

$$m = \frac{\Delta x}{\Delta y} = \frac{y_2 - y_1}{x_2 - x_1}$$

Note-se que a inclinação não é definida para retas verticais, porque neste caso o denominador da fórmula acima seria infinito. A inclinação da uma medida da variação de x em relação a y. A seguir, alguns exemplos de retas com suas inclinações



Considerando uma reta que passe pelo ponto $P_1(x_1, x_2)$ e que tenha inclinação m, ela seria $\frac{y-y_1}{x-x_1} = m$ e reescrevendo a equação da reta que passa por (x_1, y_1) e tem inclinação m:

$$y - y_1 = m(c - x_1)$$

Outra manifestação da reta é dada pela reta que intercepta o eixo Y com o valor b e tem inclinação m é

$$y = mx + b$$

Em particular se uma reta tem inclinação zero ela é horizontal e vale $y = b$. Se a reta é vertical ela não tem inclinação, mas sua fórmula pode ser escrita como $x = a$.

1.9.5 Paralelismo e perpendicularidade

Duas retas de inclinações m_1 e m_2 são paralelas se $m_1 = m_2$ e são perpendiculares se $m_1.m_2 = -1$, ou dito de outra maneira, duas retas são perpendiculares se e somente se $m_2 = -\frac{1}{m_1}$.

Capítulo 2

Instrumentos

2.1 Maple

O MAPLE é um pacote de software para manipulação matemática, que se originou na Universidade de Waterloo no Canadá, a partir de 1981. Em 1988 passou a ser desenvolvido e comercializado pela Maplesoft. Em 2009, a Maplesoft foi comprada pela Cybernet Systems do Japão.

Uma das mais famosas versões do Maple é a 15, disponível desde 2011. A licença *full* do produto custa mais de 2000 dólares, enquanto a versão estudantil custa cerca de 100 dólares. Existe uma versão liberada, que é a 5, liberada em 1990, que pode ser facilmente carregada na Internet.

O produto integra em um mesmo pacote a computação algébrica (manipulação de expressões), a computação numérica (a solução de tais expressões), a computação gráfica (gerando gráficos) e um adequado ambiente de programação de propósito geral (computação simbólica), além de um competente editor de textos matemáticos.

Iniciando Após ter instalado o produto em seu computador, clique no ícone ou inicie-o através do comando Iniciar. O local de interação entre Maple e usuário é a *worksheet*. Aqui tudo é muito parecido com um editor convencional de textos como o Word (novo, salvar, abrir, ...) O símbolo usado pelo maple para indicar que está esperando um novo comando é >, que aparece logo após um [. Após escrever o comando, o mesmo precisa ser encerrado por um ;. Opcionalmente, o comando pode ser encerrado por um :, quando então o maple não vai responder nada. Uma região do tipo *input* pode ser modificada para uma região do tipo texto, mediante o uso de F5 ou do ícone "T". Se você apertar o ENTER ao final de um comando sem colocar o terminador (; ou :) o maple vai dar erro.

Exercício Entre no maple e efetue as seguintes computações:

```
> 3+2;
> (33*2 + 8/11)^3;
> evalf(%);
```

A primeira é óbvia, está-se pedindo ao maple que calcule quanto é 3+2, e ele responde 5. A segunda, já é mais complexa, trata-se da fração que é $\frac{395446904}{1331}$, que foi obtida segundo as regras usuais da álgebra. Para obter o resultado final da divisão, pediu-se o comando $evalf$ (evaluate as floating point), que recebeu como parâmetro o sinal %, que deve ser entendido como *a última coisa calculada.* Obs: No Maple V, este sinal é a aspa dupla ("). A ante-penúltima é %% e ante-antepenúltima é %%%.

Note que a resposta veio com 10 dígitos. Para pedir com 30 dígitos, o $evalf$ deveria ser `evalf[30](%%)`. Outra maneira de modificar a quantidade de dígitos padrão é alterar a variável Digits, que normalmente tem 10, para outro valor:

```
> Digits := 20;
```

Outra maneira simples de obter o resultado diretamente em forma decimal é pedir a expressão com pelo menos um operando em ponto flutuante, ou seja (note que não é preciso fazer 3.0, basta fazer 3.):

```
> 2/3.;
0.6666666667
```

Em uma expressão aritmética, existe a seguinte regra de precedência entre as operações: Primeiro a potência, depois a divisão, multiplicação, adição e subtração. Esta ordem pode ser modificada com o uso de parênteses, como é comum. Dois ou mais comandos podem ser colocados na mesma linha, por exemplo

```
>3+2; 4+6;
```

Nesse, caso, as duas operações vão ser feitas, uma depois da outra. Também pode ocorrer o contrário, a saber: uma expressão maior pode ser escrita em mais de uma linha, desde que agrupada dentro de colchetes, e só será executada ao final quando o usuário digitar o ; seguido de ENTER. Para acionar o sistema de ajuda do maple, digite ?*palavra* onde palavra deve ser substituído pelo tópico onde você quer ajuda.

Ao encontrar uma falha na digitação, o maple dá uma mensagem de erro, inidcando qual foi o erro percebido por ele (erro de síntaxe, erro de falta de ;, erro de estouro de variável...).

Existem mais de 10 versões do Maple em funcionamento no mundo. Assim, esta descrição vai se manter no nível das generalidades para poder ser usada em qualquer uma delas.

Tudo o que é calculado na parte branca da tela compõe a folha de trabalho que pode ser salvada e recuperada mais tarde. O nome do arquivo termina por *.mw* (na versão 15).

Na parte superior da tela, existem os comandos usuais neste tipo de programa. Na linha superior está o menú, com os comandos Arquivo, Editar, Visualizar, Inserir, Formatar, Tabela, Desenho, Gráfico, Planilha, Ferramentas, Janela e Ajuda.

Abaixo, está uma barra de ferramentas, que são: operações usuais de arquivos, cortar e colar, desfazer refazer, operações com seções, navegar nas diversas folhas de trabalho, execuções e depuração e ajuda.

esquerda, existem palhestas que servem para auxiliar na entrada de dados.

Funções A seguir as principais funções do maple

raiz quadrada	sqrt(x)	sqrt(36) é 6
fatores	ifactor (x)	ifactor(12) é (2) 2 (3)
máximo divisor comum	igcd (x,y)	igcd (33,45) é 3
divisão inteira	iquo (x,y)	iquo (12,7) é 1
resto de resto de	a mod b irem (x,y)	10 mod 3 é 1 irem (10,3) é 1
é primo ?	isprime (x)	isprime (12) é false
absoluto	abs (x)	abs (-4) é 4
conversão em flutuante	convert (x,float)	
conversão em fração	convert (x,fraction)	
conversão em binário	convert (x,binary)	
coseno	cos (x)	cos (Pi/3) é 1/2
seno	sin (x)	sen (Pi/3) é $\frac{1}{2}\sqrt{3}$
tan	tan (x)	tan(Pi) é 0
exponencial	exp(x)	devolve $e\ x$
logaritmo neperiano	ln(x)	devolve $log_e x$
parte real	Re(x), parte real de a+bI	Re(1+I) é 1
parte imaginária	Im(x)	Im(1+3I) é 3
atribuição	a := x;	a := 3.33;

2.1.1 Álgebra em maple

Como programa de álgebra, o maple é muito bom em operar sobre expressões algébricas, simplificando-as, fatorando-as, expandindo-as e finalmente resolvendo-as. Antes de partir para as equações é importante distinguir entre 3 conceitos importantes em maple: sequências, conjuntos e listas. Uma sequência é uma série de constantes ou expressões separadas por vírgulas. Uma lista é uma sequência dentro de COLCHETES e um conjunto é uma sequência dentro de CHAVES. Exemplos:

sequência: 4,7,x,$sin(x)$
lista: [4,7,x,$sin(x)$]
conjunto: $\{$4,7,x,$sin(x)\}$

As diferenças entre estas 3 entidades são:

- no conjunto, não importa a ordem dos elementos. Nas listas e sequências a ordem é importante.
- as listas e sequências podem ter elementos repetidos. Nos conjuntos, o maple elimina os duplos.
- uma sequência de sequencias é uma sequência
- a quantidade de elementos em lista ou conjunto é dada por $nops(x)$

Para os conjuntos estão definidas as operações de *union* (união), *intersect* (intersecção), *minus* (diferença) e *subset* (subconjunto).

Para recuperar o i-ésimo elemento de uma lista ou conjunto X, escreve-se $X[i]$. No lugar do i pode-se escrever um intervalo de valores inteiros no formato $i..j$.

Antes de começar a escrever expressões, note-se um detalhe importante: Ao entrar uma expressão no modo texto, a multiplicação deve ser explicitamente colocada (fazer `18 * x` ao invés de `18x`) e a linha deve ser explicitamente terminada por ; ou :. Já no modo matemática, A multiplicação pode ser feita usando $*$ (`18 * x`), ou deixando um espaço entre os dois fatores (`18 x`), ou em certos casos sem deixar espaço nenhum (`18x`). Deve-se ter alguma disciplina aqui, sob pena de pagar muito sofrimento depois. Um exemplo real: ao digitar `18xy`, o maple considerou uma variável de nome `xy`, que nada tinha a ver com x nem com y.

Isto tem a ver com a modalidade de entrada de comandos em maple: Há 2 opções disponíveis, chamadas 1-D (ou maple) e 2-D. A segunda tornou-se padrão em versões maiores do maple. A decisão de como trabalhar deve-se dar no comando Ferramenta-Opções-Exibir-Exibição da entrada.

Veja a seguir uma lista das operações algébricas mais importantes do maple:

factor Fatora uma expressão algébrica, transformando-a em fatores primos. Exemplo: seja `eq1:=8*x^3 - 18*x*y^2 + 20*x`
O $factor(eq1)$ é `(3y+2x)(5y+2x)(23y+2x)`

expand Expande uma expressão algébrica. Exemplo `expand((x+1)^4);` dá como resposta $x\ 4+4\,x\ 3+6\,x\ 2+4\,x+1$

simplify Simplifica uma expressão. Nem sempre é fácil definir o que é uma expressão mais simples. O comando possui um segundo parâmetro opcional que permite fazer suposições sobre as variáveis envolvidas. Exemplo: `y:=sqrt(a^2 * (b + 1)^2);` Dá como resposta $y := \sqrt{a\ 2(b+1)\ 2}$. Escrevendo `simplify(y);` fica $y := \sqrt{a\ 2(b+1)\ 2}$. Mas, fazendo `simplify(y, assume=nonnegative);` a resposta é $a(b+1)$.

subs Permite substituir variáveis por valores ou por outras variáveis em uma expressão. Seu formato é `subs(x=exp1, y=exp2,`
Por exemplo `subs(x=2,x^2 + y^2 + z^3);` fica $4 + y\ 2 + z\ 3$.

normal Transforma a expressão em $numerador/denominador$, onde ambos são polinômios relativamente primos com coeficientes inteiros.

solve Resolve uma equação, que é uma expressão com um sinal de igual no meio. Se não tem o igual, o maple assume que a expressão dada é igual a zero. A resposta é uma sequência de raizes. Exemplo `solve(4*x - 1 = 7*x + 9);` A resposta é $-\frac{10}{3}$.

fsolve Igual ao anterior, mas dando o resultado na forma de números em ponto flutuante.

isolve resolve equações diofantinas (i=inteiro)

msolve resolve equações com congruência do tipo $f(x) \equiv\ b\ mod\ p$

combine Quando possível combina expressões (sobretudo expressões trigonométricas envolvendo seno e cosseno).

convert Converte uma expressão na outra (veja as opções em ?$convert$?. Exemplo: `> convert(x^3/(x^2-1),parfrac,x);` dá como resposta $x + 1/2\ (x-1)\ -1 + 1/2\ (x+1)\ -1$

collect Converte a expressão em um polinômio geral. Exemplo: `> collect((x+1)^3*(x+2)^2,x);` dá $4 + x\ 5 + 7\,x\ 4 + 19\,x\ 3 + 25\,x\ 2 + 16\,x$

rhs Lado direito de uma equação

lhs Lado esquerdo de uma equação

numer Extrai o numerador de uma expressão

denom Extrai o denominador de uma expressão

evalf Avalia uma expressão usando ponto flutuante

evalc Avalia uma expressão usando aritmética complexa e responde com o formato `a+I*b`.

evalb Avalia uma expressao booleana (retorna $true$, $false$ ou $FAIL$.

Diferença entre expressão e função Suponha a expressão $x\ 2+y\ 2$. Definindo-a como expressão, fica `> e:=x^2+y^2;`. Já a definição como função, fica: `> f := (x,y) -> x^2+y^2;`. Para avaliar a expressão no ponto $x = 1$ e $y = 2$, faz-se: `> eval(e,[x=1, y=2]);` cuja resposta é 5. A função é `> f(1,2);` que também dá 5.

2.1.2 Programação em Maple

O maple, embora tenha poucos comandos de programação, é um pacote completo, no qual se pode fazer praticamente tudo em termos de programas de computador. O manual do maple Programming Guide, tem mais de 600 páginas, o que demonstra, que há bastante por estudar.

A propósito, aqui vem uma sugestão de alguém que tem bastantes quilômetros de caminhada na estrada da programação de computadores: (exatamente 47 anos até 2021) muito difícil, quase impossível programar sem uma boa referência: nenhuma memória humana consegue guardar os milhares de detalhes que a maldita máquina exige. Então vai o conselho: comprem um pendrive específico para o MAPLE. Coloquem nele todos os livros, cartelas, resumos e principalmente o programa MAPLE. Guardem nele todos os programas que vierem a fazer, pois um programa já escrito (e funcionando) sempre é uma excelente referência. E, se fizerem isso, não esqueçam de fazer um backup por semana do pendrive.

Uma diferença importante aqui é o surgimento do comando `print`, cujo formato é `print(v1, v2, ...);` e cuja função é mostrar os valores das variáveis citadas. No maple interativo este comando não é necessário, basta escrever o nome da variável, mas dentro de um programa não dá para fazer assim, logo ele é necessário.

Escrita do programa

Há várias maneiras de fazer isso, mas só vai-se descrever a mais funcional delas:

Usar um editor ASCII convencional (notepad, SC1, Notepad+, entre outros) para criar um arquivo de texto contendo o programa. Dentro do Maple basta emitir o comando `read`, cujo formato é `read "nome-do-arquivo";`. Nessa hora o conteúdo do arquivo é lido e tudo se dá como se tudo o que estiver lá fosse digitado no Maple. (esta facilidade funciona no V e no 15).

Embora o nome do arquivo possa ser qualquer, vamos combinar aqui que ele sempre vai ter a extensão MPL, isto vai ajudar a recuperar programas maple (onde está aquele programa ?????).

Em outras palavras, para escrever o programa P1, o arquivo que o contém sempre vai se chamar P1.MPL.

Tome cuidado, se usar o notepad do windows, que ele tem a mania iritante de incluir a extensão TXT.

Comando Condicional Trata-se do comando IF, cujo formato é

```
if <condição> then
   comando1
   comando2
   ...
[ else
   comando k
   ... ]
fi;
```

Ele serve para determinar caminhos que dependem de alguma condição. No caso, se <condição> é verdadeira os comandos que seguem o then são executados. Se a condição for falsa e a cláusula else existir (ela é opcional), os comandos que a seguem é que serão executados. Tudo se encerra com a cláusula end if, devidamente seguida por um ponto-e-vírgula.

Relembre que quando um caminho é seguido, o oposto (se existir) é convenientemente saltado, sem ser executado. Veja-se um exemplo:

```
if soma > 250 then
   print('limite ultrapassado')
end if;
```

Outro exemplo:

```
if (x>=0) then
   r1:=sqrt(x);
else
   print('raizes negativas, saindo');
fi;
```

Embora o comando todo possa ser escrito em uma única linha, isso NÃO É UMA BOA IDÉIA. O capricho ao escrever código é que distingue o programador responsável. Lembre do velho ditado válido entre programadores: "Quando eu escrevo um programa, só eu e Deus sabemos o que aquilo faz. Uma semana mais tarde, só Deus..."

Outra possibilidade é um comando completo formado por um if, vários elif, um else e um end if. Este comando é irrelevante, já que pode perfeitamente ser feito com vários if e elses.

Comandos de repetição Existem dois comandos de repetição. São eles o `for` e o `while`. Ambos resolvem todas as necessidades em termos de desvio e repetição. O comando `for` tem o seguinte formato

```
for <variável> from <v1> to <v2> do
   comando 1,  comando 2,
   ...
od;
```

A variável citada no comando, é inicializada com o valor 1. A seguir, a lista de comandos é executada. A variável é incrementada em 1 e tem o seu valor testado com o valor 2. Se a variável for menor ou igual, a lista de comandos é reexecutada. Quando a variável ultrapassar o valor2, o comando for se encerra.

Se o incremento for diferente de um, a primeira linha do comando passa a ser:

```
for <variável> from <v1> to <v2>
   by <incremento> do ...
```

Assim como nos demais comandos, nada impede que um for seja construído dentro do outro. Daí cada um deles controla sua variável e tudo funciona legal.

s vezes, a repetição não deve ser feita um número determinado de vezes, e sim deve depender da verificação de uma dada condição. Nesses casos, o for não pode ser usado e sim o comando while, cujo formato é

```
while <condição> do
   comando1    comando2
   ...
od;
```

A <condição> é avaliada. Se ela for verdadeira, a sequencia de comandos é executada. Quando o end do é encontrado, ocorre um desvio para o while, quando a condição é reavaliada, e assim por diante.

Dentro de ambos os laços podem ser incluídos os comandos `break` e `next`. O comando `break` encerra prematuramente o laço, independente de outras condições. Quando break é executado, o próximo comando é o seguinte ao end do.

Quando `next` é executado dentro de um `for`, a variável é incrementada e a condição testada. Quando dentro de um `while`, apenas a condição é reavaliada.

Procedimentos É o que em outras linguagens recebe o nome de programa. O formato de um procedimento (programa) em maple é

```
<nome-procedimento> := proc()
   comando1    comando2
   ...
end;
```

Para chamar (executar) o programa acima, basta fornecer o comando nome-procedimento();

É comum, o programa receber variáveis externas, que serão conhecidas apenas em tempo de execução. Tais variáveis são citadas dentro do parênteses e são conhecidas pelo programa através de seu valor (passagem por valor). Também é comum que o procedimento precise devolver um determinado valor. Usa-se o comando opcional return <valor>. Se este comando não for citado o procedimento devolve a última coisa que for calculada. Note que esta definição de procedimento é muito parecida com a definição de uma função.

Se a variável de chamada deve ser de um determinado tipo (sob pena de erro), seu nome pode ser citado como (<variavel>::<tipo>). Veja no exemplo:

```
P := proc(x::posint)
   if isprime(x) then
  RETURN true;
   else
  RETURN false;
   fi;
end;
```

Todas as variáveis citadas em um procedimento são locais, a menos que a declaração global <variável> seja emitida dentro do procedimento.

Deve-se ressaltar que um procedimento pode receber outro procedimento como parâmetro em sua chamada. A quantidade de parâmetros passados para um procedimento pode ser obtida através da variável nargs. A sequencia de parâmetros passados para o procedimento é guardada na variável args, que é naturalmente indexada.

Maple possui o comando printf, muito similar ao seu homônimo em C, e que permite fazer impressão formatada.

2.2 Python

A pergunta: **Porque Python ?**

- É uma linguagem moderna (nasceu em 1990) e herdou tudo de bom que as demais linguagens já tinham nessa época. Uma coisa que ela não herdou foi a preocupação exigente de desempenho: Em 90, já havia computação abundante. Com isso a linguagem é redonda e não impõe trancos e barrancos para funcionar. [1]
- Multiplataforma: Unix, Windows, Apple, celular android, raspberry e outros ambientes menos famosos rodam exatamente o mesmo programa.
- Freeware: pode copiar, baixar, instalar, vender, ceder... pode fazer qualquer coisa sem infringir a lei e sem dever explicações a ninguém.
- Uso no mainstream: Olhem o que diz o Philip Guo, um blogueiro da CACM "oito dos top-10 Departamentos de Ciência da Computação e 27 dos top-39 Departamentos de Ciência da Computação das principais universidades nos EUA usam Python nos cursos introdutórios de CS0 e CS1" (CS0 e CS1 são os cursos introdutórios de Computer Science lá). As 39 universidades pesquisadas por ele incluem: MIT, Berkeley, Stanford, Un. Columbia, UCLA, UIUC, Cornell, Caltech, Un. Michigan, Carnegie Mellon, entre outras [2]
- Muito Usada: o índice TIOBE [3] lista o Python como a quinta linguagem mais usada no mundo em 2017. . Em junho de 2018, Python já é a quarta linguagem mais usada. Perde apenas para java, c e c++. E, corre por fora...

  Se você olhar o estudo da TIOBE https://www.tiobe.com/tiobe-index/ verá o bonito papel desempenhado pelo Python.

Linguagem	17	12	07	02	97	92
Java	1	1	1	1	12	-
C	2	2	2	2	1	1
C++	3	3	3	3	2	2
C#	4	4	7	17	-	-
Python	5	7	6	11	27	-
VB.NET	6	19	-	-	-	-
PHP	7	6	4	5	-	-
JavaScript	8	9	8	8	19	-
COBOL	25	28	17	9	3	10
Lisp	32	12	14	12	9	5
Prolog	33	32	26	15	20	12
Pascal	106	15	19	97	8	3

  Consulta em 07/2017

  Uma pequena explicação desta tabela: ela lista de 5 em 5 anos (o que no mundo da computação é quase uma eternidade) a lista dos top-10 ou as linguagens de programação mais usadas. O começo é 1992, e daí de 5 em 5 anos, por 30 anos ou 6 ciclos, até o último em 2017. Diversos fenômenos podem ser observados, eis alguns:

  – A performance notável de Java, cuja explicação parece estar na palavra *portabilidade* ou seja o mesmo programa fonte rodando em inúmeras plataformas distintas. Isso é conseguido pelo conceito de java machine. Foge ao escopo deste texto descrever o conceito, mais em resumo é uma capa ou camada de compatibilidade que se adiciona ao ambiente de maneira a tornar ou permitir a mesma coisa sobre coisas diferentes.

  – A derrocada do Pascal, nascido como uma excelente ferramenta de aprendizado e que em algum momento se perde completamente.

  – A consistência de C e seus agregados (c++ e c#), seu sucesso pode ser descrito como eficiência e o possível alto nível de portabilidade.

  – A morte anunciada e posteriormente executada do COBOL. Ele só permanece com alguma relevância graças ao legado ou aos milhões de linha de código que ainda precisam ser mantidos vivos.

  – Quase a mesma coisa pode ser dita das linguagens da inteligência artificial: Lisp e Prolog. Elas migram para a generalidade: Java, C e Python.

  – A irrupção da Internet: PHP, Javascript e Python, por trás dos panos. Inclusive da Internet das Coisas (aqui com Java).

[1] Em compensação, quando se organiza uma maratona de programação, para um determinado problema, impõe-se 1 segundo de CPU quando ele é resolvido em C++ e pelo menos 10 (10 vezes mais) para quando ele é resolvido usando Python

[2] https://cacm.acm.org/ blogs/blog-cacm/176450 -pyhon-is-now-the -most-popular-introductory- teaching-language-at- top-u-s-universities/ fulltext, acessado em agosto/2017

[3] www.tiobe.com, acessado em agosto/17

- Pacotes, pacotes, pacotes. A comunidade Python mantém o PYPI (Python Package Index) que é um repositório freeware com pacotes de programação: hoje ele tem mais de 110.000 pacotes. Só para ter uma idéia, um pacote destes, o NUMPY (numerical computation in Python) tem um manual de referência de mais de 1500 páginas.

- Inteligência Artificial: Quando o Google resolveu aposentar o DistBelief e criar uma nova ferramenta de Machine Learning, escolheu a linguagem Python para desenvolvê-la: nasceu o Tensor Flow, que está por trás de inúmeras iniciativas google: tradução, classificação de imagens, elaboração de perfis etc. [4] Logo na entrada do produto o aviso: pode usar em Java, C++ ou Go. Mas, se quiser usar a funcionalidade completa do produto, use-o em Python, que ele está feito nessa linguagem.

- Fácil: parece portugol, ou vá lá: inglêsgol. Não tem caracteres sobrando: sem ponto e vírgula, sem chaves nem colchetes. O que se escreve, ele executa e raramente dá erro.

Python é freeware, não custa nada, você pode copiar e usar para o que desejar, sem pedir licença a ninguém, e pode inclusive cobrar pelo software que você desenvolver usando Python. A partícula *free* em freeware significa liberdade para fazer o que você quiser sem nenhum tipo de restrição e não gratuidade, como alguns pensam.

Existem versões de Python para quase todos os sistemas operacionais conhecidos, eis alguns: AIX (da IBM), AROS (Amiga), AS/400, HPUnix, Linux, MacOS, MS-DOS, OS/2, PalmOS, Playstation, Solaris, VMS, Windows (32 e 64), WindowsCE, entre outros.

Esta multiplicidade de versões acrescenta uma qualidade imensa aos programas Python: desde que eles não usem nenhum recurso específico de uma dessas arquiteturas, serão 100% (ou quase isso) portáveis podendo rodar o mesmo programa em todas essas plataformas.

2.2.1 Versão

As versões de Python são numeradas em 3 níveis: O primeiro número identifica a versão global que atualmente pode ser 2 ou 3. Além das versões 0 e 1, quase experimentais, a linguagem começou sua carreira global com a versão 2, que chegou a ser bastante usada, muito software foi construído com ela, e esse fato impede agora que ela seja totalmente substituída pela versão 3, o que seria o melhor dos mundos. Se você está começando agora, não há o que pensar: deve-se escolher a versão 3. O problema é que na mudança de 2 para 3 perdeu-se alguma compatibilidade o que impede a livre transição de uma a outra. A versão 2 já está com sua sentença de morte estabelecida, ela só será suportada até 2020. Desse ponto em diante quem quiser usá-la o fará por sua conta e risco.

O segundo número indica a sub-versão, que diferencia algum nível maior de correção e sobretudo de lançamento de novidades. Finalmente o terceiro número vai sendo incrementado à medida em que pequenas correções e melhoramentos vão sendo construídos. Eis uma pequena versão das modificações:

Python 3.6.5	liberado em 28 Março 2018
Python 3.6.4	19 Dezembro 2017
Python 3.6.0	23 Dezembro 2016
Python 3.5.1	07 Dezembro 2015
Python 3.3.7	19 Setembro 2017
Python 3.2.6	11 Outubro 2014
Python 3.0	3 Dezembro 2008
Python 2.7.8	1 Julho 2014
Python 2.6.2	14 Abril 2009
Python 2.1.3	8 Abril 2002
Python 1.4	25 Outubro 1996

Existe tamanha diversidade no mundo Python que é impossível tratar todas as alternativas de uso. Primeiro, vai-se escolher o Python 3 e depois vai-se olhar 3 delas: a mais tradicional possível, sob Windows (nas 2 versões, sob 32 e sob 64 bits) e também uma instalação alternativa que permite instalar o Python em um pendrive para usá-lo em qualquer computador, mesmo sem os direitos de administrador nesse computador e o Python interativo.

2.2.2 sob Windows

A primeira decisão é sob qual ambiente de windows você está. Uma maneira simples de descobrir é solicitar o *Painel de Controle*, e aqui solicitar a aba *System*. Isto mostra um texto com algumas características do sistema e lá no meio do texto, vem a frase "System type". O que aparecer aí deve servir de guia para a escolha do Python a baixar e instalar: 32 ou 64 bits.

Daí, deve-se ir a `https://www.python.org/downloads/` e escolher a última versão de distribuição. Usualmente há versões mais novas que a última, mas são conhecidas como experimentais e não são recomendáveis, pois ainda não

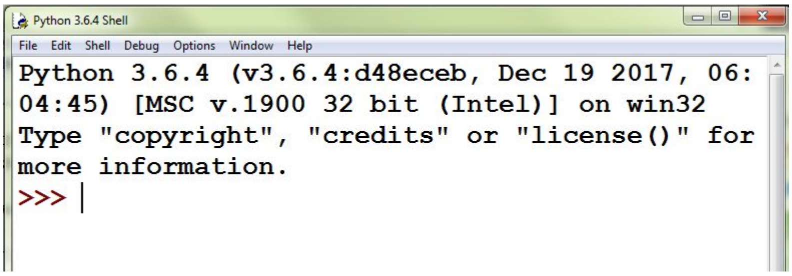
[4] www.tensorflow.org

estão cristalizadas ou consolidadas. Deve-se escolher sempre o Python 3 e não o 2, que este vai desaparecer nos próximos anos e só está nessa lista para quem precisa usar e suportar software antigo.

No processo de instalação, especial preocupação deve ser tida ao verificar em qual diretório a instalação vai ser feita. Eu recomendo criar um diretório novo (não usar a sugestão do instalador) no primeiro nível, logo abaixo do nível raiz (como em `c:/python3`). Sob este diretório, o instalador vai criar diversos sub-diretórios (DLL, doc, etc, include, LIB, libs, Scripts, tcl e tools). É importante olhar isto, pois para instalar novos pacotes é preciso entrar no diretório Scripts, como se verá adiante. Logo é preciso saber onde está.

Depois de instalar o Python, deverá aparecer no seu menu inicial um ícone com 2 cobrinhas (uma azul e outra amarela)

o desenho padrão do Python. O nome deve identificar qual a versão instalada e quando se clicar nele deve-se abrir uma tela inicial do python, como em



Nessa tela minimalista, surgem algumas informações como versão, data e hora da liberação desta versão, tipo de sistema sob qual ela está rodando. Depois algumas opções disponíveis e finalmente o símbolo

```
>>>
```

Ele é o *prompt* (prontidão) que é um conceito usado por praticamente todos os softwares interativos para sinalizar ao operador que o software está *pronto* para aceitar e executar comandos. Enquanto este símbolo não aparecer, isso significa que o processador está ocupado fazendo alguma coisa e não pode ser interrompido pelo operador.

Aqui já se podem escrever comandos Python como em

```
>>> 1+2
3
>>>
```

Aqui pediu-se ao Python a execução de $1 + 2$ e ele respondeu 3, seguido de um novo *prompt*. Esta modalidade de uso do Python e conhecida como interativa e neste caso ele funciona como se fosse uma calculadora. Basta comandar alguma coisa e ele a executará.

2.2.3 Alo Mundo

Tradicionalmente o primeiro programa que um novato deve construir em qualquer novo ambiente é chamado `Alo Mundo`. Sua função é apenas mostrar esta mensagem e sua execução com sucesso sinaliza que o processo de criar um programa fonte, salvá-lo, compilá-lo e depois executá-lo foi bem sucedido. Vamos a isso, então:

1. na tela do Python, escolha `File`
2. no menu que vai aparecer, escolha `New File` e lembre que estes dois passos podem ser comandados apertando `Ctrl N`
3. Vai-se abrir uma nova tela em branco, muito parecida com a tela do Python. Parece Python, mas não é Python e sim o bloco de notas do Python[5]
4. Na tela do bloco de notas do Python escreva o seguinte

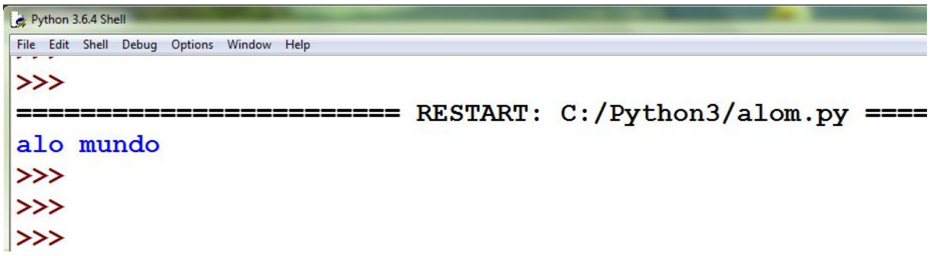
   ```
   print("alo mundo")
   ```

5. Agora você vai comandar a compilação (interpretação) e execução desse script (tecnicamente isto não é um programa e sim um script: é uma diferença semântica meio irrelevante, mas será estudada depois). Quem faz isto é a tecla `F5`. Ou também o comando `Run`, já destacado.

[5]Uma maneira enviezada de identificar qual tela é qual é esta: o bloco de notas do python tem o comando Run enquanto a tela do Python não tem este comando.

6. O Python antes de qualquer coisa vai pedir para você um nome sob o qual o programa (ou script) vai ser salvo. Esta é uma excelente exigência do Python: garante que aconteça o que acontecer, o programa que você escreveu e está tentando executar, será salvo em disco, sem risco de que ele se perca, caso o pior (tela azul da morte, por exemplo) aconteça. O nome que você der terá agregado o sufixo `.py` e será salvo no lugar onde você informar [6]

7. Tendo sido salvo e compilado com sucesso a tela do Python vai mostrar



   Esta tela mostra a execução do script que você criou e a mensagem sendo mostrada.

8. Isto significa que você teve sucesso na instalação do python e na criação e execução do seu primeiro programa.

2.2.4 Winpython

Esta implementação aparece aqui por algumas razões, sendo a principal a **portabilidade** do ambiente de desenvolvimento:

- Permite usar Python em máquinas nas quais você não é o administrador, coisa que o Python padrão exige para ser instalado.
- Como consequência, este Python é portável, e pode ser carregado em um pendrive ou similar, ainda que haja alguma demora no processo de carga: o winpython é enorme.
- Winpython é um ambiente científico, pelo que já vem com SciPi e Numpy devidamente instalados, além de eventualmente outros pacotes importantes.

2.2.5 Ipython

É um python interativo, no qual fica mais evidente o diálogo entre usuário e Python. Na sua versão original (Python) é uma linguagem de programação que conta com um enfoque interativo também. No Ipython, ao contrário, é um ambiente interativo, que por acaso também dá acesso a uma linguagem de programação. Muitas vezes associado ao Jupyter, que é um ambiente interativo construído originalmente para as linguagens JUlia, PYThon E (and) R. Posteriormente o ambiente foi migrado para muitas outras linguagens e hoje o menú é bem extenso. O ambiente é bom pois permite desenvolver notebooks que são arquivos simulando um caderno de notas mesclando texto, figuras e código.

Ele aparece aqui nesta lista pois é a versão disponibilizada nos laboratórios do curso de engenharia elétrica da UFPR.

Neste ambiente, eis o caminho das pedras:

1. Usar seu login para entrar no sistema.
2. Na tela de atividades, digitar IP e aguardar. Escolher Ipython.
3. Carregar também o editor GEDIT. Manter ambos abertos. (Ipython e GEDIT)
4. Escrever o código python no GEDIT e salvar o programa e sua execução no diretório /eletrica, com o nome de aa.py
5. Dentro do Ipython executar `%run aa.py`
6. Para corrigir, edite no GEDIT, salve e reexecute dentro do Ipython.

2.2.6 Python na WEB

Uma alternativa disponível a quem tenha acesso à web e não pretenda (ou não possa) instalar nada em sua máquina é usar um ambiente python disponível na web. Há vários, mas um funcional é `https://www.pythonanywhere.com`. Aqui você precisa criar um login e depois disso passa a acessar um ambiente python completo. Já estão instalados nele vários pacotes importantes (teste: numpy, sympy, matplotlib, django). Este ambiente é perfeito para uma consulta rápida.

[6] Sugestão para quem está começando: compre e use um pendrive para guardar seus programas: eles irão com você. E não esqueça de fazer cópias deles – o famoso backup.

2.2.7 Variáveis, sua entrada e saída

Uma variável é um objeto do mundo do software que pode ter algumas abordagens. A primeira, é aquela da matemática, quando se escreve $y = f(x)$ e se quer dizer que y é uma variável que depende do valor de x. O que existe aí, nessa expressão, é uma função que associa o valor de y ao valor de x. Tanto x quanto y nesse enfoque são variáveis: isto é são valores eventualmente desconhecidos, mas que têm um valor qualquer. Outra abordagem é da ciência da computação: nela, uma variável é um pedacinho de memória dentro de um programa (que está devidamente registrado nessa memória) e que armazena um determinado valor. Cada variável precisa ter um nome, para que possa ser referenciada. Esse nome precisa ser único (não se pode ter duas variáveis com o mesmo nome), e depois que ela foi criada, usar essa variável num comando qualquer é exatamente a mesma coisa que usar o valor que a variável tem. Por exemplo, se a variável A tem o valor 10, tanto faz escrever $A+1$ como $10+1$, só que no primeiro caso, a generalidade é maior, pois independentemente do valor que ela tiver, depois de executar $A+1$, ter-se-á um valor uma unidade maior do que o valor anterior de A.

Atribuição

O comando importante aqui é o chamado **atribuição** . É ele que permite:

- criar uma variável
- alterar o valor de uma variável

As duas funções só se diferenciam pelo existência prévia da variável. Se ela não existe, usar a atribuição significa criar a variável. Se ela já existia (e portanto tinha um valor qualquer) usar a atribuição nela vai alterar o valor dela para este, que é citado agora na atribuição.

O formato do comando é

```
variável = valor
```

Onde `variável` é o nome da variável, `=` identifica o comando de atribuição e `valor` é o valor que a variável passará a ter depois desta atribuição. Por exemplo:

`A=10`	A variável de nome A é criada (ou alterada) e passa a ter o valor 10.
`X=-33.4`	Agora o valor de X é o número negativo -33.4
`NM='alfa'`	A variável agora se chama NM e seu valor é uma palavra, que é 'alfa'. Atente para a presença das aspas, elas se

Algumas observações importantes. O caráter de atribuição é = e ele é lido como `recebe`. Então, o comando $A = 10$ deve ser lido como `A recebe 10`. Outra observação é que este sinal tem outras funções no mundo. Por exemplo, na matemática o sinal de igual é isso mesmo: para indicar uma igualdade. Igualdade não é a mesma coisa que atribuição. O problema é que se não se usar o igual, que outro símbolo pode-se usar ? Outras linguagens resolveram esse problema, usando dois caracteres (por exemplo, Pascal usa `:=`) enquanto outras usaram caracteres novos, como por exemplo APL, que usa `←`.

A discussão de qual símbolo usar para a atribuição (que lembrando é um conceito que não existe na mátemática) é antiga e poderia ser aberta aqui. Mas, pegando carona no `C`, linguagem avô de boa parte do leque de linguagens usadas no século XXI, (tais como C++, Java, PHP, entre outras), o Python também associou o sinal de = à atribuição. A pergunta é: o que usar quando se falar sobre igualdade entre dois valores ? Neste caso, usam-se 2 caracteres iguais. Então, para perguntar se as variáveis A e B têm o mesmo valor, escreve-se $A == B$, já que se se escrever $A = B$, o que se estará mandando fazer é atribuir o valor da variável B para a variável A.

Voltando ao formato do comando de atribuição, acima descrito, vamos estudar o que pode ser `valor` citado no formato do comando. Como o nome sugere, `valor` pode ser um número como 13, 2780.4, -33 e assim por diante. Mas, também pode ser outra variável, como em `A=X`. Aqui se está mandando copiar o valor de `X` para a variável `A`. Note que não se sabe que valor é esse. Não importa, ele é copiado de `X` para `A`.

Finalmente, esse valor pode ser uma expressão aritmética como em `A=11+8`, ou em `A=V-3`, ou finalmente `A=cos(theta)`. No primeiro caso, `A` vale 19, no segundo vale o valor de `V` menos 3 e no terceiro caso `A` vale o coseno do angulo `theta`.

Agora pode-se ver a importância da presença das aspas no comando de atribuição: acompanhe no exemplo `A='alfa'` e `A=alfa`. No primeiro caso, é o conteúdo `alfa` que está sendo atribuído à variável A. No segundo pressupõe-se a existência de uma variável de nome `alfa` e é esta variável que tem seu valor (seja ele qual for) copiado para a variável `A`.

As aspas denotam um conteúdo alfanumérico, que estruturalmente é diferente de um conteúdo puramente numérico. Há muitas diferenças que ainda serão estudadas, mas uma importante é que conteúdos (e variáveis) numéricas podem ser usadas em expressões aritméticas enquanto variáveis alfanuméricas causam erros de execução quando se tenta envolvê-las em expressões numérico-aritméticas.

`>>> A=5.5`	Cria-se a variável A, flutuante, com o valor 5.5
`>>> B='oba'`	Cria-se B alfanumérica contendo 'oba'
`>>> A+B`	Tenta-se somar as duas variáveis. Há um erro

O erro é sinalizado pela mensagem `TypeError: unsupported operand type(s) for +: 'float' and 'str'` cuja tradução é: Erro de tipo: não se pode somar (+) um tipo float com um tipo str.

Swap de variáveis

Em programação muitas vezes é necessário inverter o conteúdo de 2 variáveis. Em situação normal, isso exige o uso de uma variável auxiliar como em

```
>>> aux=a
>>> a=b
>>> b=aux
```

Python possui um mecanismo simples para realizar a troca sem usar variáveis adicionais. O mesmo exemplo acima pode ser feito assim

```
a,b=b,a
```

2.2.8 Tipos

Graças a essa distinção vista acima, pode-se pontuar a respeito de uma característica importante das variáveis. Cada variável terá um tipo que sinalizará como ela será manipulada depois de criada. Os principais tipos são: numérico inteiro, numérico real, alfanumérico e lógica.

Ao contrário de outras linguagens (como o C, por exemplo), as variáveis em Python não têm um tipo explicitamente declarado. É o seu valor que determina qual é o seu tipo. [7] Então,

`A=10.3`	A é uma variável do tipo real, que em Pythons é conhecida como `float`
`B=5`	A variável B é do tipo inteiro.
`C=10.0`	A variável C é tipo float
`HAL='ivo viu a uva'`	A variável HAL contém a frase `ivo viu a uva`
`GA=True`	GA é uma variável do tipo lógico que tem o valor de verdadeiro
`J=False`	J é uma variável do tipo lógico e tem o valor Falso

A variável do tipo lógico só admite dois valores `True` e `False`. Obviamente, estes dois nomes não servem para dar nomes a variáveis. É isto que se chama de palavra reservada .

Os tipos numéricos podem ser inteiros (`int`) ou reais, que no Python recebem o nome de `float`. Poder-se-ia argumentar que o tipo inteiro é desnecessário (já que o conjunto dos reais contém o conjunto dos inteiros ou $\mathbb{R} \subset \mathbb{Z}$), mas isto não é verdade, já que:

- Dentro do computador, a matemática inteira é dezena de vezes mais rápida que a matemática real. Então quando aplicável, sempre vale a pena ser mais eficiente.
- Mas, principalmente, o conjunto dos inteiros tem uma característica que o real não tem: a enumerabilidade. Tal característica é muito importante (por exemplo, na indexação) e ela exige que o tipo seja inteiro em certos momentos.

Feita esta distinção, os conteúdos que uma variável numérica pode ter são:

inteiro : os 10 dígitos `0,1,2,3,4,5,6,7,8 e 9`, um sinal opcional `+ ou -`.

flutuante : os mesmos dígitos e sinal e além destes, o sinal de ponto decimal. Note que usa-se em Python o ponto decimal e não a vírgula decimal, herança dos países de língua inglesa. Opcionalmente pode-se ter a letra `E` ou `e` significando exponencial. E, se presente, indica o expoente de 10, que deve ser multiplicado pelo número antes do E. O expoente também pode ter opcionalmente um sinal que pode `+` ou `-`. Exemplos:

`3E4`	Significa 3×10^4 ou 3000
`-4E2`	-4×10^2 ou -400
`5E-3`	5×10^{-3} ou $5 \times \frac{1}{10^3}$ ou ainda 5×0.001 que é 0.005.

2.2.9 Nomes

A regra para dar nomes em Python (regras essas que se aplicam à nomeação de variáveis) são simples:

- uma única palavra, ou seja não se admitem espaços em branco dentro dos nomes.
- pode-se usar o caracter sublinha (`_`) para separar palavras no nome
- o primeiro caractere deve ser uma letra

[7] Esta regra nem sempre é verdadeira, por exemplo, ao usar o pacote Numpy. Mas, por enquanto pode-se aceitá-la como verdadeira.

- depois do primeiro pode-se usar letras ou números
- Letras maiúsculas são diferentes das letras minúsculas (ou seja `OBA` é diferente de `Oba` que não é a mesma coisa que `oba`).

O leitor não deve se preocupar muito com estas regras, substituindo-as por algum bom senso: os nomes devem ser autodeclarativos (sobre para que serve a variável), com nomes nem muito grandes (provocam erros de digitação) nem muito pequenos (para que serve mesmo a variável B ?). Uma regra muito usada por aí sugere usar várias palavras ou abreviaturas no nome, separando umas das outras pelo uso de maiúsculas. Usando esta regra pode-se ter

`SaldoConta`	uma variável para conter o saldo da conta
`LucroAntesIR`	auto-explicativo

Os detratores da regra acima dizem que não se deve misturar maiúsculas e minúsculas no nome, pois isto acaba sendo uma fonte de erros de digitação. Estes dizem que as variáveis devem ser todas minúsculas ou todas maiúsculas.

Enfim, a sugestão aqui é que cada leitor deve definir a sua regra de formação de nomes e se ater a ela, sempre visando a minimização dos erros de digitação.

A partir da versão 3, o Python pode aceitar caracteres acentuados na formação de nomes, pois aqui por padrão usa-se o conjunto de caracteres Unicode, através do Unicode Transformate Format-8 (UTF-8). Mas, pensando francamente, não parece ser uma boa idéia: se você definir uma variável de nome `SAÍDA` e depois chamá-la de `SAIDA`, para o Python serão duas variáveis distintas, e portanto aqui está uma nascente fonte de problemas. Tente não usar caracteres acentuados ao programar. Você não vai se arrepender. Conteúdos dentro de aspas (constantes) podem e devem usar caracteres acentuados: vamos homenagear a Língua Portuguesa, e dentro das aspas não se corre nenhum perigo. Como em `ESTADO='Paraná'`.

2.2.10 Expressões aritméticas

As variáveis e constantes numéricas podem ser operadas e interligadas usando-se as operações aritméticas usuais (e algumas não tão usuais) da matemática convencional. Para isso são usados os símbolos a seguir descritos, que buscam seguir – tanto quanto possível – as convenções usuais da escrita matemática. Por exemplo

```
>>> a = 5
>>> b = 3.6
>>> a + 4
9
>>> b * 3
10.8
>>> (a + b) / 2
4.3
>>> a - 6
-1
>>> a - -8
13
```

Note que a adição é representada pelo sinal de "mais" (+), a subtração por "menos" (−), a multiplicação pelo asterísco ($*$) e a divisão real pela barra (/).

Note também que a prioridade é determinada pelo uso de parênteses, que as vezes é redundante, mas nem sempre, pois se o comando fosse escrito como $2 + 8/2$ sem parênteses, o Python primeiro faria a divisão de $8/2$ dando 4 e depois a soma de 2 com 4 resultando 6. Ao se escrever $(2 + 8)/2$ primeiro a soma dando 10 e depois a divisão por 2, que dará 5 de resposta final. A seguir, uma tabela de algumas operações aritméticas:

Símb	Nome	Como funciona
$+$	adição	em $a + b$ o resultado é a adição entre a e b.
$-$	subtração	em $a - b$ o resultado é $a - b$.
$*$	multiplicação	Em $a * b$ o resultado é a multiplicação de a e b.
$/$	divisão real	Em a/b o resultado é um `float` contendo a divisão de a por b.
$//$	divisão inteira	Em $a//b$ o resultado é um inteiro contendo a divisão inteira de a por b.
$\%$	Resto	Em $a\%b$ o resultado é o resto da divisão inteira de a por b, sendo que é um valor $0 \leq resto \leq b - 1$.
$**$	Potência	Em $a * *b$ o resultado de a^b.
abs	Módulo	Em $abs(a)$ o resultado é $-a$ se a é negativo e a se a é positivo.

Note a diferença entre as duas divisões

```
>>> 10/3
```

```
3.3333333333333335
>>> 10//3
3
>>> 10%3
1
```

2.2.11 Saída de Dados

Todo programa durante sua execução deve em princípio divulgar alguma coisa. Afinal para algo ele está sendo executado. Ainda que haja quase infinitas possíveis aplicações de programas, grande parte deles vai precisar dizer algo. Uma diferença importante entre Python 2 e Python 3 é que no 2, os parâmetros de print seguem o nome da função sem parênteses enquanto que os parênteses são obrigatórios no Python 3. Aliás este é um dos mecanismos para identificar se um programa originalmente foi escrito em 2 ou 3.

A maneira mais simples de obter o conteúdo de uma variável é apenas digitando seu nome seguido de ENTER. Veja

```
>>> a = 5
>>> b = a+2
>>> b
7
```

Uma maneira mais elaborada de saída é com o comando `print()`. É muito rico, com múltiplas opções, mas vamos começar lentamente. Na sua versão mais simples, o print admite uma lista de variáveis e eventualmente constantes a serem escritas na saída padrão quando o programa for executado. A saída padrão é um conceito importante em computação pessoal: trata-se do dispositivo que o computador usará para todas as saídas, quando nada for afirmado sobre ONDE fazer a saída. Em computação usa-se o conceito de *default* nestes casos. Então a saída *default* é conhecida como saída padrão e quase sempre é o monitor de vídeo. Vejam-se alguns exemplos

`print(a)`	Imprime o valor da variável a.
`print("o valor de a: ",a)`	Agora antes do valor da variável a, imprime-se a mensagem entre aspas. Ajuda a identificar as saídas
`print("a: ",a,"b: ",b)`	Imprime a e b identificando cada uma delas
`print("Desligue agora")`	Imprime apenas a mensagem
`print("Resultado ",x+22)`	Imprime a mensagem e depois o resultado de x+22. Note que o valor de x permanece inalterado

2.2.12 Entrada de dados

A grande maioria dos algoritmos (programas) precisam obter algum dado do operador antes de realizar o processamento. O comando básico aqui é `input`. Seu formato sempre será

```
variavel = input("mensagem")
```

O que acontece aqui:

- A mensagem citada no comando é impressa na saída padrão
- O programa é interrompido neste ponto
- depois que o usuário digitar alguma coisa e teclar `ENTER`, o que foi digitado é colocado na variável citada no comando
- O processamento segue sem mais interrupções.

Um detalhe importante é que os conteúdos SEMPRE serão carregados no formato de string. Sem transformações, estas variáveis não podem ser usadas, por exemplo, em computações numéricas. Para tanto elas deverão ser transformadas usando (entre outras) as funções: `int` converte algo para inteiro, `float` converte algo para flutuante, e já que estamos estudando conversões, tem-se também `str` que converte algo para string.   Veja alguns exemplos

```
>>> idade=input("entre com a idade ")
entre com a idade 33
>>> idade
'33'
>>> idade=int(input("entre com a idade "))
entre com a idade 33
>>> idade
33
```

Perceba que no primeiro caso (sem a transformação `int`) a variável idade é do tipo string, o que se pode perceber nas aspas que são mostradas ('33'). No segundo caso, em que há a transformação de string para inteiro (representada pela chamada à função `int`, o conteúdo da variável é um número e portanto pode ser operada aritmeticamente.

2.2.13 Um programa completo usando input e print

Seja um programa para pedir o raio de um círculo e imprimir a área do mesmo usando a fórmula da geometria

$$S = \pi \times R^2$$

```
def area():
  a=float(input("Informe raio"))
  s=3.1415 * a
  print("a área é: ",s)
area()
```

Mais um exemplo

```
def prim2(n):
  import math
  if n==1:
    return False
  if n<4:
    return True
  if n%2==0:
    return False
  if n<9:
    return True
  if n%3==0:
    return False
  r=math.floor(math.sqrt(n))
  f=5
  while f<=r:
    if n%f==0:
      return False
    if (n%(f+2))==0:
      return False
    f=f+6
  return True

def enesprim(qual):
  count=1
  candidato=1
  while count<qual:
    candidato=candidato+2
    if prim2(candidato):
      count=count+1
  return(candidato)

aa=int(input("Informe qual a ordem do primo "))
bb=enesprim(aa)
print(bb)
```

Neste exemplo primeiro define-se a função prim2 (bastante otimizada) que recebendo um valor qualquer informa se ele é ou não é primo. Depois define-se a função enesprim que recebe um número inteiro x e devolve o x-ésimo número primo. Finalmente, o script pede um número qualquer em aa, e com ele calcula o aa-ésimo número primo, e o joga em bb. Depois este número é impresso.

2.2.14 Variáveis Lógicas

Uma outra família de variáveis são as variáveis lógicas, geralmente obtidas como resultado de uma comparação ou pergunta. Essas variáveis só podem ter 2 valores que são Verdadeiro (em Python `True`) e Falso (`False`).

Usualmente, estas variáveis são obtidas ao operar condições. As principais condições em Python são

Símb	Nome	Como funciona
==	Igual	Em $a == b$, a resposta será True se a for igual a b e False em caso contrário.
$>$	Maior	Em $a > b$ a resposta será True se a for maior do que b. Será False em caso contrário.
$<$	Menor	Em $a < b$ a resposta será True se a for menor do que b. Será False em caso contrário.
$>=$	Maior ou igual	Em $a >= b$ a resposta será True se a for maior do que ou for igual a b. Será False em caso contrário.
$<$	Menor ou igual	Em $a <= b$ a resposta será True se a for menor do que ou igual a b. Será False em caso contrário.
$! =$	Diferente	Também conhecida como não igual, ela responde o contrário da operação igual ==.

2.2.15 Expressões Lógicas

Quando há necessidade de expressar condições compostas, as operações acima podem ser conectadas usando-se 3 conectores lógicos que são:

Símb	Nome	Como funciona
and	E	Em $c1$ *and* $c2$, o resultado será verdadeiro se ambas, $c1$ e $c2$ forem verdadeiras e Falso, senão
or	OU	Em $c1$ *or* $c2$, o resultado será verdadeiro se $c1$ for verdadeiro, ou se $c2$ for verdadeiro ou se ambas forem verdadeiras. Será falso em caso contrário.
not	NÃO	Em *not* $c1$ o resultado será verdadeiro se $c1$ for falsa e será falsa se $c1$ for verdadeiro. Note que este operador é unário (só se aplica a uma condição).

Deve-se lembrar que estas operações e conectores estão definidas na matemática, particularmente na lógica de predicados. Lá seus símbolos são: $\wedge$ para o conectivo E, $\vee$ para o conecttivo OU e finalmente o símbolo $\sim$ para a negação lógica.

2.2.16 Comandos condicionais

Uma característica onipresente em linguagens de programação é a capacidade de executar ou não um conjunto de instruções a depender de uma variável lógica. Dizendo de outra maneira, pode-se comandar uma condição qualquer e se ela for verdadeira (que no Python é `True`) então um conjunto de instruções será executado. Se a condição for falsa (em Python `False`), ou nada é executado, ou também pode ocorrer, um segundo conjunto alternativo de instruções é executado.

O comando condicional por excelência em Python é `if` e ele têm o seguinte formato

```
if condição:
   comando_1
   comando_2
   ...
comando_n
```

Para escrever o comando, deve-se substituir a palavra condição por alguma condição relacional ou lógica, que ao final informe um valor `True` ou `False`. A regra é que os comandos 1, 2, ... só serão executados se a condição for `True`. Já ao final do bloco, o comando n será executado sempre independentemente da condição. Quem determina se o comando está ou não vinculado ao if é o espacejamento no início da linha, chamada em programação de indentação.

Em outras linguagens, usam-se caracteres delimitadores para os blocos de comando ({ e } em C, Java e C++; *begin* e *end* em Pascal e Delphi), mas em Python consoante com a proposta de limpeza da linguagem, usam-se as margens para determinar as condições.

Como tudo na vida isto tem vantagens e desvantagens. Começando pelas primeiras, o código fica limpo e legível. Quem sempre exigia dos programadores que respeitassem as margens, agora ganha um aliado importante: se as margens forem bagunçadas o programa não funcionará bem. Além disso economizam-se caracteres dispensáveis.

Já a desvantagem decorre principalmente da inabilidade de muitos ambientes para manusear este tipo de arquivo. Muitos editores se arvoram em trocar um certo número de brancos por um ou mais caracteres de tabulação, e fazer isto é receita quase certa para problemas de operação dos programas. Além disso, muitos gerenciadores de download deixam de respeitar essas margens, fazendo com o que o recebedor de um programa fonte Python tenha sempre que revisar com cuidado o resultado de descargas de código.

Veja alguns exemplos

```
a = 5
if a==6:
   print('ufa')
print('agora')
```

Primeiro a recebe 5. Daí 5 é comparado para igual com 6. A resposta é `False`. Com isso, os comandos identados abaixo não são executados (no caso print('ufa')). Finalmente é impresso 'agora' já que esta impressão independe do resultado da comparação.

Quem inicia um bloco indentado é a presença dos dois pontos : após a condição. Ele sinaliza ao editor em uso que a seguir deve-se respeitar a nova margem. Quando o programador quiser encerrar o bloco (e por conseguinte, a condição) ele terá que recuar manualmente o cursor até o início da linha.

Um segundo formato para o comando condicional, conhecido como condicional composta é aquele que apresenta uma condição e dois caminhos: um para o True e outro para o resultado False. Agora a separação entre um e outro é sinalizado pela palavra `else` que também deve ser seguida por dois pontos. Acompanhe no exemplo

```
a = 7
if a>3:
   print('oba')
else:
   print('ufa')
print('foi')
```

Acompanhe o raciocínio: Primeiro a é comparado com maior para 3. A resposta é `True` já que $7 > 3$ e com isso a impressão de 'oba' é feita. Como o resultado foi True, salta-se o bloco referente ao else:. Encerra o comando a impressão de 'foi' que independe do resultado da comparação. Veja agora um trecho parecido

```
a = 2
if a>3:
   print('oba')
else:
   print('ufa')
print('foi')
```

Agora a comparação resulta False (já que $2 > 3$ é falso) e com isso a impressão de 'oba' é saltada e depois a impressão de 'ufa' acontece. Novamente 'foi' é impresso independentemente da comparação inicial.

2.2.17 Repetições

As repetições são estruturas da linguagem que asseguram a execução repetida de trechos de comandos, até que uma determinada condição for satisfeita. Note que este conceito faz parte do nosso dia a dia.

Por exemplo, uma caixa dágua posta a encher o fará até que atinja a capacidade total. Um forno assando um bolo deverá operar até que o bolo esteja assado, uma esteira carregadeira de um navio opera até que o depósito do navio esteja cheio ou enquanto o silo abastacedor tiver conteúdo, parando na condição que primeiro for atingida, um pneu de carro receberá ar comprimido até que uma certa pressão seja atingida e a água do café vai esquentar até ferver ou um pouco antes.

O que não falta na nossa vida são operações repetitivas que ocorrem até que uma condição seja atendida, ou o que praticamente é a mesma coisa, enquanto certa condição perdurar.

2.2.18 while

Os comandos básicos em Python são `while` que pode ser traduzido por enquanto. Seu formato é

```
while <condição>:
   comando_1
   comando_2
   ...
   comando_i
comando_n
```

Quando ele for encontrado será assim interpretado: primeiro a condição associada ao while será avaliada. Se ela for falsa, ocorre um desvio desprezando todo o bloco indentado abaixo. No exemplo acima, se a condição for falsa, o próximo comando a ser executado é o $comando_n$. Agora, se a condição é verdadeira, o bloco indentado é executado. Quando ele acabar, há um retorno (daí a repetição) ao início do bloco e a reavliação da condição. Enquanto ela for verdadeira, o bloco fica em loop.

Logo, aqui vale uma advertência: quando usar um *while* garanta que em algum momento do bloco, a condição seja tornada falsa, sob pena do seu programa nunca mais sair daí.

Por exemplo, seja somar os 33 primeiros números ímpares:

```
i=1
q=0
p=1
```

```
while i<34:
   q=q+p
   p=p+2
   i=i+1
print(q)
```

O programa acima vai imprimir 1089, que é a soma procurada.

2.2.19 break

Um comando que permite modificar o funcionamento do while é o break. Normalmente ele é colocado dentro de uma condição, embora isto não seja obrigatório. Quando break é interpretado ele faz com que o ciclo do while, na verdade qualquer ciclo, seja interrompido, forçando uma saída do bloco indentado.

Exemplo, seja um programa para calcular o triplo de um número digitado. O programa deve operar até ser digitado um número negativo, que significará fim de processamento.

```
a=1
while 1==1:
  if a<0:
    break
  a=int(input("informe o valor "))
  print(3*a)
print('adeus')
```

Note que como vai-se usar o *break* para a saída do laço, escreve-se uma condição que sempre é verdadeira (conhecida na lógica como uma tautologia) , que neste caso é $1 == 1$, que, por óbvio, é sempre verdadeira.

Eis um trecho da execução desse programa:

```
========== RESTART: C:/Python3/teste.py ===========
informe o valor 2
6
informe o valor 33
99
informe o valor 8
24
informe o valor -1
-3
adeus
>>>
```

2.2.20 Repetição aninhada

Quando se cria um bloco de repetição, dentro do bloco, o Python admite qualquer comando, inclusive um outro comando de repetição. Então surgem os blocos aninhados que são muito importantes em programação. Acompanhe o exemplo

```
i=1
while i<=3:
   j=10
   while j<=30:
      j=j+10
      print (i,j)
   i=i+1
print('acabou')
```

que vai gerar a seguinte saída

```
(1, 20)
(1, 30)
(1, 40)
(2, 20)
(2, 30)
(2, 40)
(3, 20)
(3, 30)
(3, 40)
acabou
```

2.2.21 Continue

Este comando também serve para interromper um ciclo de processamento, mas diferentemente de `break` que encerra inapelavelmente o laço, este comando `continue` encerra o processamento de uma interação e avança para a próxima interação. Acompanhe a diferença entre os dois comandos

```
i=3
while i<10:
   print(i,' ')
   if i%7==0:
      break
   i=i+1
print('acabou')
```

Esta execução teve o seguinte resultado:

```
3
4
5
6
7
acabou
```

Agora acompanhe o seguinte exemplo usando continue

```
i=3
while i<10:
   print(i,' ')
   if i%7==0:
      continue
   i=i+1
print('acabou')
```

Neste caso, não ponha para rodar, pois o programa vai entrar em laço infinito. Veja porque: O processamento começa com i=3, 4, 5 r 6. Quando o valor 7 for atingido, o continue vai comandar um retoino à próxima interação, mas sem alterar o valor de i (já que esta alteração está depois do `continue`. Aí ela vai imprimir 7,7,7,7,.... e não vai parar nunca mais.

2.2.22 For

O For é o grande interador do Python. Ele é uma expansão do comando `for` do C++. Lá no C++ o for funciona como se fosse uma simplificação do while, mas aqui ele é mais do que isso. A mudança se deve ao conceito de interador. Numa rápida explicação um interador é uma generalização do conceito de índice. Ele é uma generalização já que o índexador só se aplica a conjuntos homogêneos indexados (conhecidos como arrays) e o interador se aplica a qualquer coleção de objetos Python. Pode ser um array, mas pode ser também registros em um arquivo, itens em um conjunto, etc etc.

Veja um exemplo radical disso

```
s="Curitiba - Paraná - Brasil"
qtd=0
for c in s:
  if c=="a":
    qtd=qtd+1
print("achei ",qtd," a")
```

Ao executar o código acima, ele vai responder `achei 4 a"`. Note que são 4 letras ’a’, sendo que o ’a’ acentuado de Paraná, não conta, já que é outro caracter.

2.3 sympy

O Symbolic Python é um pacote Python que implementa um sistema de computação simbólica (CAS=Computer Algebra System) no ambiente Python. Para entender o que é um sistema deste tipo, acompanhe:

```
comput. numérica: sqrt(8)=2.82842712
comput. simbólica: sqrt(8)=2*sqrt(2)
```

A vantagem da segunda é que é fácil passar da segunda para a primeira, mas não é possível o sentido oposto. Acompanhe:

```
>>> print(sp.sqrt(8))
2*sqrt(2)
>>> print(sp.sqrt(8).n())
2.82842712474619
>>> print(sp.sqrt(8).n(30))
2.82842712474619009760337744842
```

O paradigma de computação simbólica é o Maple. As vantagens de usar o sympy versus o Maple (por exemplo) são pelo menos duas: sympy é de graça (Maple custa 1.000 dólares) e por ser em python admite que você use tudo o que já sabe de python e também permite usar os mais de 135.000 pacotes também de graça.

Instalação e uso O sympy precisa ser instalado. Na versão para windows, você deve localizar o diretório scripts e nele fazer `pip install sympy` em um computador que tenha acesso à Internet. Se quiser usar winpython, nele o sympy já está pré-instalado. Num ou noutro, precisa importar o pacote. Faça `from sympy import *` ou algum comando equivalente. Este pacote exige como dependência o pacote NUMPY. A documentação completa do sympy ocupa hoje 2.127 páginas A4. Sua última versão deve ser buscada na Internet (procure `sympy doc`). Este é um produto relativamente recente na história de Python (2013), e portanto alguma instabilidade pode aparecer. Paciência, nem tudo nasce pronto neste planeta.

Começo

```
>>> import sympy as sp
>>> sp.init_session()
```

Importação Neste material, vai-se executar sempre o comando de importação

```
import sympy as sp
init_session() # estabelece ambiente
```

Iniciando Uma novidade neste pacote é o surgimento de variáveis simbólicas que terão um comportamento diferente das variáveis convencionais do Python. Quem cria variáveis simbólicas é o comando `symbols()`, veja no exemplo

```
>>> x,y,z = sp.symbols('x,y,z')
```

A partir deste ponto as variáveis x, y e z estarão disponíveis para armazenar expressões simbólicas. Uma definição alternativa, usando ferramentas do Python é

```
>>> a0,a1,a2,a3 = sp.symbols('a0:4')
```

Note que falando estritamente, não é obrigatório que os símbolos tenham o mesmo nome das variáveis, mas para evitar fortes emoções, vamos usar sempre o mesmo nome.

Expressões Pode-se definir expressões em sympy usando símbolos junto com as operações básicas da matemática além de outras funções. Para escrever expressões, deve-se obedecer ao padrão Python (por exemplo $2x$ deve ser escrito como `2*x`, a potência x^2 como `x**2`). Há uma diferença importante em que a igualdade em sympy deve ser escrita como `Eq(a,b)` para representar $a = b$. Ou melhor ainda, se $a = b$, então $a - b = 0$ e daí pode-se escrever em sympy `a-b`.

Há um método em sympy chamado `equals` que testa 2 expressões em pontos aleatórios e retorna True se todos resultarem iguais.

```
>>> a,b,x=symbols('a,b,x')
>>> a=cos(x)**2-sin(x)**2
>>> b=cos(2*x)
>>> a.equals(b)
True
```

Substituição A substituição é usada para avaliar uma equação num ponto

```
>>> expr
cos(x) + 1
>>> expr.subs(x,pi/2)
1
>>> (x**2+2*x+10).subs(x,3)
25
```

Também é usada para trocar uma subexpressão por outra, como em

```
>>> (x**2+2*x+10).subs(x,y**2)
y**4 + 2*y**2 + 10
```

Às vezes, o uso de métodos primitivos pode levar a resultados indesejados, como em

```
>>> expr=sin(2*x)+cos(2*x)
>>> expand_trig(expr)
2*sin(x)*cos(x) + 2*cos(x)**2 - 1
```

Ao querermos a simplificação apenas do seno duplo, faz-se

```
expr.subs(sin(2*x),2*sin(x)*cos(x))
2*sin(x)*cos(x) + cos(2*x)
```

Note-se que o método subs devolve outra instância de expressão já que a expressão original permanece imutável. A substituição pode ser de mais de uma variável/expressão

```
>>> expr=3*x**3+4*y**2-2*z
>>> expr.subs([(x,3),(y,5),(z,1)])
179
```

Simpificação Não confundir com simplificação. Trata-se de transformar uma string em uma expressão sympy

```
>>> a="x**2+3"
>>> a
'x**2+3'
>>> b=sympify(a)
>>> b
x**2 + 3
>>> b.subs(x,1)
4
```

Impressão Há vários dispositivos em sympy. `pprint()` imprime a saída na tela. Há `latex` e `mathml`, entre outros.

```
>>> pprint(Integral(sqrt(1/x),x))
```

$$\int \sqrt{\frac{1}{x}} dx$$

```
>>> latex(Integral(sqrt(1/x),x))
'\\int \\sqrt{\\frac{1}{x}}\\, dx'
```

Simplificação Sympy tem dúzias de funções para fazer diversos tipos de simplificação. Há uma função principal chamada `simplify()` que aplica todas estas simplificações escolhendo depois a a mais simples. Isto nem sempre funciona bem, já que a expressão *a mais simples* não está claramente definida.

```
>>> simplify(sin(x)**2+cos(x)**2)
1
>>> simplify((x**3+x**2-x-1)/(x-1))
x**2 + 2*x + 1
>>> simplify(x**2+2*x+1)
x**2 + 2*x + 1
```

expand() e factor() Para este último caso, pode-se escolher `expand()` ou `factor()` que são dois simplificadores importantes em sympy.

```
>>> expand((x+1)**3)
x**3 + 3*x**2 + 3*x + 1
>>> factor(x**3+3*x**2+3*x+1)
(x + 1)**3
```

A expansão de um polinômio coloca-o na forma de uma soma de monômios. Já fatoração transforma a soma de monômios em uma multiplicação de fatores irredutíveis. Note-se que a entrada de ambos não precisa ser um polinômio estrito senso.

```
>>> init_printing()
>>> expand((cos(x)+sin(x))**2)
sin(x)**2 + 2*sin(x)*cos(x) + cos(x)**2
```

collect() Outro simplificador é `collect()` que junta potências comuns dentro da expressão.

```
collect(x*y+x-3+2*x**2-z*x**2+x**3,x)
x**3  + x**2 *(2 - z) + x*(y + 1) - 3
```

cancel() `cancel()` coloca qualquer função racional na forma canônica $\frac{p}{q}$ onde p e q são polinômios expandidos sem fator comum e com fatores inteiros.

```
>>> e=symbols('e')
>>> e=1/x+(3*x/2-2)/(x-4)
>>> cancel(e)
```

$$\frac{3.x^2 - 2.x - 8}{2.x^2 - 8.x}$$

apart() `apart()` executa uma decomposição de frações parciais de uma função racional.

```
>>> e=(x**3+x**2+1)/(x**4+x**2)
```

$$\frac{x}{x^2+1} + \frac{1}{x^2}$$

trigsimp() Simplifica usando identidades trigonométricas circulares e hiperbólicas.

```
>>> trigsimp(sinh(x)/tanh(x))
cosh(x)
```

expand_trig() Expande funções trigonométricas, ou seja, aplica as identidades de ângulos duplos ou soma de ângulos

```
>>> expand_trig(sin(x+y))
sin(x)*cos(y) + sin(y)*cos(x)
>>> expand_trig(tan(2*x))
```

$$\frac{2.tan(x)}{1 - tan(x)^2}$$

Potência As potências obedecem às seguintes identidades:

1. $x^a x^b = x^{a+b}$
2. $x^a y^a = (xy)^a$ só é verdade quando $x, y \geq 0$ e $a \in \mathbb{R}$
3. $(x^a)^b = x^{ab}$ só é verdade quando $b \in \mathbb{Z}$

Veja-se para as regras 2 e 3: $(-1)^{1/2}(-1)^{1/2} \neq (-1. - 1)^{1/2}$ e $((-1)^2)^{1/2} \neq (-1)^{2.1/2}$ Esta observação é importante pois o sympy só aplica simplificações quando elas são sempre verdadeiras. Assim, as regras 2 e 3 só serão aplicadas se expressamente suas pré-condições forem explicitamente declaradas.

powsimp() Aplica as identidades 1 e 2 da esquerda para a direita

```
>>> powsimp(x**a*x**b)
 a + x
b
```

expand_power_exp(), expand_power_base() e powdenest() Aplicam as identidades acima.

Logaritmos Têm tratamento similar às potências. Há 2 identidades:

1. $\log(xy) = \log(x) + \log(y)$
2. $\log(x^n) = n.\log(x)$

Nenhuma dessas 2 é verdadeira para $x, y \in \mathbb{C}$. Para serem verdadeiras, $x, y \geq 0$ e $n \in \mathbb{R}$.

expand_log() Aplica as identidades 1 e 2 da esquerda para a direita.

```
>>> expand_log(log(x*y))
log(x*y)
>>> expand_log(log(x*y),force=True)
log(x) + log(y)
```

logcombine() aplica as identidades 1 e 2 da direita para a esquerda.

```
>>> logcombine(log(x)+log(y))
log(x) + log(y)
>>> logcombine(log(x)+log(y),
     force=True)
log(x*y)
```

factorial(n) e binomial(n,k)

rewrite() Permite reescrever uma função em termos de outra

```
>>> tan(x).rewrite(cos)
```

$$\frac{cos(x - \frac{\pi}{2})}{cos(x)}$$

diff() Acha a derivada de uma função. Seu formato `diff(função,variável)`.

```
>>> diff(exp(x**2),x)
```

$$2.x.e^{x^2}$$

```
>>> diff(tan(x),x)
   2
tan (x) + 1
```

Para conseguir derivadas múltiplas, passe a variável diversas vezes, ou a variável seguida da ordem da derivada.

```
>>> diff(x**4,x,x,x)
24.x
>>> diff(x**4,x,3)
24.x
```

Derivative() estabelece literalmente a derivada, tendo o mesmo formato de diff(). Para forçar o cálculo posterior, chame `doit()`.

```
>>> e=exp(x)+sin(x)
>>> e
```

$$e^x + sin(x)$$

```
>>> diff(e,x)
```

$$e^x + cos(x)$$

```
>>> Derivative(e)
```

$$\frac{d}{dx}e^x + sin(x)$$

```
>>> Derivative(e).doit()
```

$$e^x + cos(x)$$

integrate() Há 2 tipos de integrais: a definida e a indefinida. Para a integral indefinida é `integrate(função, variável)`.

```
>>> integrate(cos(x))
sin(x)
```

Note-se que não aparece a constante aditiva. Se ela for necessária deve ser manualmente incluída ou então o problema deve ser redesenhado para ser resolvido como uma equação diferencial.

Para computar uma integral definida, o formato é `integrate(funç, (var, linf, lsup))`

```
>>> integrate(exp(x),(x,0,oo))
oo
>>> integrate(exp(-x),(x,0,oo))
1
>>> integrate(exp(-x**2-y**2),
    (x,-oo,oo),(y,-oo,oo))
pi
```

Também há `Integral()` e depois para implementá-lo, chame `doit()`.

```
>>> e=log(x)**2
>>> Integral(e,x)
```

$$\int \log(x)^2 dx$$

```
>>> Integral(e,x).doit()
     2
x*log (x) - 2*x*log(x) + 2*x
```

limit() Seja o limite $lim_{x \to x_0} f(x)$

```
>>> limit(sin(x)/x,x,0)
1
```

$$\lim_{x \to 0+} \frac{cos(x) - 1}{x}$$

```
>>> Limit((cos(x)-1)/x,x,0).doit()
0
```

Note-se que limites à esquerda e a direita, exigem um quarto parâmetro que pode ser `'+'` ou `'-'` respectivamente.

series() Permite a expansão de séries em torno a um ponto. Para computar a expansão de $f(x)$ em torno do ponto $x = x_0$ em termos de x^n, usa-se `f(x).series(x,x0,n)`. `x0` e `n` podem ser omitidos, no caso em que serão assumidos $x_0 = 0$ e $n = 6$.

```
>>> exp(sin(x)).series(x,0,4)
```

$$1 + x + \frac{x^2}{2} + O(x^4)$$

solveset() A função principal para solução de equações. Sua síntaxe é `solveset(equação, variável, domínio)`. A equação $x = y$ deve ser escrita usando `Eq(x,y)` ou mais facilmente como `x-y`.

```
>>> solveset(x**2-x,x)
{0, 1}
>>> solveset(sin(x)-1,x,domain=Reals)
```

$2.n.\pi + \frac{\pi}{2} / \in \mathbb{Z}$

linsolve() Resolve sistemas de equações lineares. Seu formato `linsolve([eq1,eq2,...],(v1,v2,...))`. Seja o sistema: $2x + 3y - z = 4$, $x + y + z = 7$ e $x - 2y + 3z = 9$

```
>>> linsolve([2*x+3*y-z-4,x+y+z-7,
    x-2*y+3*z-9],(x,y,z))
{(1, 2, 4)}
>>> linsolve(Matrix([[2,3,-1,4],
    [1,1,1,7],[1,-2,3,9]]),(x,y,z))
{(1, 2, 4)}
```

nonlinsolve() Resolve sistemas não lineares.

```
>>> nonlinsolve([x**2+y-z+1,2*x+y+z-8,
    x-2*y+3*z-9],(x,y,z))
{(-1, 4, 6), (1, 2, 4)}
```

dsolve() Resolve equações diferenciais. Para tanto deve-se criar funções

```
>>> f,g=symbols('f,g',cls=Function)
>>> f(x)
f(x)
>>> f(x).diff(x)
d
--(f(x))
dx
```

Supondo a equação diferencial $f''(x) - 2f'(x) + f(x) = sin(x)$ deve-se fazer

```
>>> dx=(f(x).diff(x,x)-
    2*f(x).diff(x)+f(x)-sin(x))
>>> dx
```

$$f(x) - sin(x) - 2.\frac{d}{dx}(f(x)) + \frac{d^2}{dx^2}(f(x))$$

```
>>> dsolve(dx,f(x))
```

$$f(x) = (C_1 + C_2.x).e^x + \frac{cos(x)}{2}$$

Matrizes Criação de matrizes podem ser feitas com `Matrix([[l1],[l2],...])`, com `eye(ordem)`, `zeros(lin,col)`, `ones(lin,col)` e `(diag(vetor))` em que o vetor é colocado na diagonal.

Matrizes podem ser operadas como em `M+N`, `3*M`, `M**2` e `M**-1` (esta última traz a matriz inversa).

```
>>> M=Matrix([[1,2,3],[1,2,1],[1,0,6]])
>>> M
[1  2  3]
[1  2  1]
[1  0  6]
>>> M**-1
[-3    3     1  ]
[5/4  -3/4  -1/2]
[1/2  -1/2   0  ]
```

det() calcula o determinante de uma matriz

```
>>> M.det()
-4
```

rref() Calcula a forma reduzida por linhas, retornando uma tupla de 2 elementos. O primeiro é a matriz reduzida por linhas e o segundo os índices das colunas pivot.

```
>>> N=Matrix([[2,3,-1,4],[1,1,1,7],
    [1,-2,3,9]])
>>> N
[2  3   -1  4]
[1  1   1   7]
[1  -2  3   9]
>>> N.rref()
 [1  0  0  1]
([0  1  0  2], (0, 1, 2))
 [0  0  1  4]
```

init_session()

```
from sympy import *
x, y, z, t = symbols('x y z t')
k, m, n = symbols('k m n', integer=True)
f, g, h = symbols('f g h', cls=Function)
init_printing() # doctest: +SKIP
```

2.4 APL

O APL é uma linguagem à frente do seu tempo por excelência. Hoje, graças às interfaces orientadas a gráfico, rompeu-se o grande represor antigo que era a dificuldade de representar os caracteres APL. (Ajuda também do UNICODE). Prova de estar à frente de seu tempo: O APL é de 1964, O Windows de 1988 e o Unicode de 1995. Hoje, em 2020, temos as seguintes alternativas:

APL2 da IBM. Legítimo sucessor do APL que começou quando o Ken Inverson juntou-se a uma turma boa (Adin Falkoff entre outros) na IBM para gerar a primeira versão de APL operacional. Teve como sucessor o APL2, que ainda hoje é comercializado pela IBM. Grande software: estável e com enorme base instalada. A versão que uso é para 32 bits, obtida em 2004 através de acordo universitário. A versão 64 bits existe, mas precisa ser comprada.

NARS2000 Uma tentativa de escrever um APL freeware. Disponível em `http://www.nars2000.org/`. Parece bem completa, tenho instalada no meu computador, mas nunca fui muito longe com ela.

APLX Uma iniciativa da empresa inglesa microapl (`http://microapl.com/APL`). Era um excelente APL, cheguei a usá-lo por alguns anos. Comprei uma licença pagando 99 libras em 2009. Anos mais tarde, a empresa decidiu descontinuar o produto. A notícia boa é que ao fazer isso disponibilizou ao mundo a versão que até então era paga. Já adaptada para vigorar em 32 ou 64 bits. Seu sítio migrou para o sítio do Dyalog APL. Gera excelentes produtos compilados. Para usar a versão liberada: *Enter a Licence number of 70702153 and a Serial number of W7CBS4CX and click OK.*

Dyalog É esta que vai ser objeto de estudo aqui. Não é freeware, mas inteligentemente pode ser baixada e usada enquanto prova de conceito. Não pode gerar software comercial. Quando isto ocorrer, as licenças precisam ser compradas. Veja mais em `https://www.dyalog.com/`.

outras Tem algumas outras versões/opções. Lembro de cabeça: `VSAPL` , `APLPLUS` e `Sharp APL` entre outras. Não vão ser tratadas aqui, tem cheiro de mofo. Por exemplo a APLPLUS, que era muito boa, mas só conseguia acessar 640KB de memória, já que operava em uma janela DOS. Para o APL isso é um traque apenas.

Então, daqui para a frente, está-se tratando apenas do Dyalog APL, também tratado aqui como DAPL2. O 2 no nome indica ser versão compatível com arrays aninhados, a grande mudança que a IBM fez ao passar do APL $\rightarrow$ APL2.

2.4.1 Dyalog APL: Qualidades e novidades

Eis uma lista incompleta e simples das novidades que ela apresenta:

- 100 % de compatibilidade reversa. Isso é importante porque permite reinflar software feito em APL em anos passados. A compatibilidade só se perde quando o software antigo usava características nativas do ambiente antigo. Por exemplo, ao usar variáveis e funções de sistema. Dando um exemplo mais específico: ao converter software feito em APL2 e que – lastimavelmente – usava a função `⎕FMT` daquele ambiente.
- Novos operadores. Lembrando que o APL original tinha 4 ou 5 operadores (produtos interno e externo, redução e scan além de eixo). Agora são muitos, e precisam ser estudados. Tem um capítulo inteiro, dedicado a eles, a seguir.
- Muitas variáveis e sobretudo funções de sistema novas. Algumas francamente inúteis, mas deve haver o que se salva.
- Suporte pleno a Unicode. Se o Unicode já é importante no software em geral, aqui no APL ele é fundamental sob pena de não haver comunicação com nada e ninguém.
- Suporte à programação estruturada. Este realmente era o calcanhar de Aquiles do APL e igualmente do APL2. Embora o APL já menosvalorizasse o conceito de laço pela sua natural vocação a substituí-los, ainda assim, é muito difícil programar sem estas ferramentas.
- Comandos de usuário. Uma sacada genial, que permite escrever código APL para rodar standalone (isto é, sem estar fisicamente no ws onde é executado). Outra sacada genial é identificar estes comandos por um `]`, numa óbvia analogia com os comandos de sistema. Aparece a seguir no capítulo SALT.
- DFN. Uma iniciativa bem interessante. Equivalem às funções *lambda* de LISP. Permitem definir funções de maneira muito simples e rápida para uso instantâneo.

2.4.2 Primitivas

A idéia aqui é destacar as principais mudanças em relação ao APL *old fashioned*, já que se o objetivo fosse descrever completamente o APL este texto deveria ter centenas de páginas. Menos, menos.

Axis Operator - Operador Eixo Usado em inúmeras situações, sendo algumas novas. Acompanhe

```
      1 0 1/[1]3 2⍴⍳6
1 2
5 6
      1 2 3+[2]2 3⍴10 20 30
11 22 33
11 22 33
```

Um número fracionário indica um novo eixo.

```
      'NOMES',[0.5]'-'
NOMES
-----
```

Obviamente este operador depende do valor de `⎕IO`.

Abort - Cancelamento A flecha à direita sozinha aborta uma interrupção e limpa o state indicator.

Adição Operação de adição convencional. A possível novidade fica por conta dos complexos

```
      1J3 20+3J3
4J6 23J3
```

And / MMC - And / Mínimo Múltiplo Comum Se ambos operandos são booleanos, a função é AND (∧). Se um dos operandos não é booleano, a função retorna o mínimo múltiplo comum.

```
      1 0 1 ∧ 0 0 1
0 0 1
      22 33 44 ∧ 4 5 6
44 165 132
```

Assignement - Assinalamento Cria ou altera uma ou mais variáveis

```
      a←4.5
      a
4.5
      b←2 3 4
      b
2 3 4
      (a b)←2
      a,b
2 2
      (a x y)←'alfa' 4 (4 5 6)
      a
alfa
      x
4
      y
4 5 6
      a ← b ← c ← 0
```

A novidade é o assinalamento de funções. Note que não há aspas.

```
      somatorio←+/
      somatorio 1 2 3 4
10
      media←{(+/⍵)÷⍴⍵}
      media 1 2
1.5
```

Pode-se assinalar uma referência a um namespace em uma variável, que terá classe 9. Um array contendo referências terá classe 2. Um nome já existente pode receber um novo valor desde que não se altere a sua classe. Veja à pág 15 para mais detalhes.

O assinalamento pode ser indexado, com 3 possibilidades. i. Assinalamento com índices simples

```
      a←ι5
      a[2 3]←50
      a
1 50 50 4 5
```

Em um array, índices podem ser omitidos, significando a extensão total. Veja

```
      a←3 4ριl2
      a[2;]←50
      a
 1  2  3  4
50 50 50 50
 9 10 11 12
```

ii. Assinalamento a índices escolhidos

```
      c
11 12 13 14
21 22 23 24
      c[(1 2) ( 2 3)]←102 103
      c
11 102  13 14
21  22 103 24
```

iii. Assinalamento para índice encontrado: Quando a especificação do índice I não é um array de inteiros simples

```
      E←('gato' 'cachorro' 'cavalo')
      E
 gato  cachorro  cavalo
      E[⊂2 1]←'X'
      E
 gato  Xachorro  cavalo
```

Note que para qualquer array A, a quantidade `A[⊂⍬]` representa uma quantidade escalar, então

```
      a←ι6
      a[⊂⍬]←1
      a
1
```

Selective Assignement - Assinalamento seletivo Tem como formato

```
      (exp X) ← Y
```

A expressão `exp` seleciona elementos de X. A expressão Y é atribuída aos elementos selecionados de X por *exp*. A seguir uma lista de funções para assinalamento seletivo

```
↑ (take), ↓ (drop), , (ravel), ⍪ (tabela), ⌽ ⊖ (reverso, rotação), ρ (reformatação)
⊃ (disclose, pick), ⍉ (transposta monádica e diádica), / ⌿ (replicação),
\ ⍀ (expansão), ⌷ (indexação), ∊ (enlist)
```

Eis alguns exemplos

```
      x←'Curitiba'
      ((x∊'aeiou')/x)←'#'
      x
C#r#t#b#
      y←3 4ριl2
      (7↑,y)←111
      y
111 111 111 111
111 111 111   8
  9  10  11  12
```

```
     mat←3 3 ρι9
      1 1⍉mat
1 5 9
      (1 1⍉mat)←0
      mat
0 2 3
4 0 6
7 8 0
```

Tudo isto pode ser usado junto com o operador each (¨)

```
       x←'olha' 'a' 'pandemia'
      (2↑¨x)←'*'
      x
 **ha * **ndemia
      a←'hello world'
      ((a='o')/a)←'*'
      a
hell* w*rld
```

Binomial Em `R←X!Y`, X e Y podem ser quaisquer números. A exceção é que se Y é negativo, X deve ser inteiro. Para ambos inteiros `X!Y=(!Y)÷((!X)×(!Y-X))`. Para outros argumentos é a função beta `beta(X,Y)=÷Y×(X-1)!X+Y-1`. Para argumentos inteiros, o resultado é o número de seleções de X tomados de Y.

Branch - Desvio É a flecha à direita, com alguma expressão após. Tem inúmeras formas, mas vamos ficar com as duas consagradas: `→label` significa desviar incondicionalmente para o label. Já na expressão `→(condição)/label` só haverá desvio se a condição for verdadeira ou igual a 1.

Catenate/Laminate - Concatenação e Laminação Concatenar significa tomar junto e laminar tomar em uma nova dimensão. Lembrar que quando não há especificação de eixo, vale sempre o último. Já o símbolo `⍪` é usado para o primeiro eixo. Veja

```
     (ι3),ι2
1 2 3 1 2
      (2 3ρι6),0
1 2 3 0
4 5 6 0
      (2 3ρι6)⍪0
1 2 3
4 5 6
0 0 0
     (2 3ρι6),[0.5]0
1 2 3
4 5 6

0 0 0
0 0 0
     (2 3ρι6),[1.5]0
1 2 3
0 0 0

4 5 6
0 0 0
```

Ceiling - Teto No formato `R←⌈Y`. Y deve ser numérico, e se for complexo, a operação será aplicada às partes real e imaginária do número.

```
      ⌈1 1.2 1.9 ¯3.4
1 2 2 ¯3
      ⌈1.2J3.9
2J4
```

Circular No formato `R←X○Y`, onde Y deve ser numérico e X estar entre 12 positivo e negativo (`¯12 ≤ X ≤ 12`). Veja-se a tabela, na qual a e b são `a+bi` e θ é a fase.

X	X negativo: `(-X)○Y`	X positivo: `X○Y`
0	igual à direita (não existe -0)	`(1-Y*2)*.5`
1	`arcsen Y`	`sen Y`
2	`arccos Y`	`cos Y`
3	`arctan Y`	`tan Y`
4	`Y=¯1:0` `Y≠¯1:(Y+1)×((Y-1)÷Y+1)*0.5`	`(1+Y*2)*.5`
5	`arcsenh Y`	`senh Y`
6	`arccosh Y`	`cosh Y`
7	`arctanh Y`	`tanh Y`
8	`-8○Y`	`(-1+Y*2)*.5`
9	`Y`	`a`
10	`+Y`	`|Y`
11	`Y×0J1`	`b`
12	`*Y×0J1`	θ

Conjugate - Conjugado Em `R←+Y`, se Y é real ou não numérico devolve Y sem alteração, como `⊣`, mas este é mais rápido. Se Y é complexo, nega a parte imaginária

```
     a←'alfa'
      +a
alfa
      +3J4
3J¯4
```

Deal - Aleatórios sem repetição Em `R←X?Y`, Y deve ser escalar não negativo, idem para X, porém `X≤Y`. R é um vetor de seleções randômicas de `⍳Y` sem repetição.

```
      5?8
8 6 5 3 4
```

Decode - Decodificação No formato `R←X⊥Y`, onde X e Y devem ser arrays numéricos simples. R resulta no número obtido pela avaliação do número Y na base de numeração X

```
      60 60⊥ 2 27
147
      2⊥1 0 1 1
11
```

Este comando também permite avaliar polinômios como em $y_1.x^2 + y_2.x + y_3$ devendo-se neste caso escrever `x⊥y1,y2,y3` como em `2⊥1 2 3 4 = 26` que corresponde ao polinômio $y_1.x^3 + y_2.x^2 + y_3.x + y_4 =$ o que é $1.2^3 + 2.2^2 + 3.2 + 4 = 8 + 8 + 6 + 4 = 26$

Depth - Profundidade Cujo formato é `R←≡Y` e Y pode ser qualquer array. R é o número máximo de aninhamento de Y. Alguém escalar tem depth de 0. Qualquer array simples, sem aninhamento tem depth de 1. Qualquer array cujos elementos não são simples é aninhado e seu depth é uma unidade maior do que o elemento mais aninhado que o formar. Y tem depth uniforme se todos os seus elementos têm o mesmo depth uniforme. Se Y não tem depth uniforme a resposta é negativa.

```
        ≡55
0
      ≡2 3⍴⍳8
1
      ≡⊂1 2 3
2
      ≡( 1 2 (⊂1 2))
¯3
```

Direction (Signal) - Direção (sinal) No formato `R←×Y`, Y deve ser um array numérico. Para Y real, informa o sinal de Y, a saber: 0 se Y é zero, 1 se Y é positivo e -1 se Y é negativo. Se Y é complexo a resposta é um complexo com a mesma fase e com magnitude igual a 1.

```
    ×10 0 ¯11
1 0 ¯1
      ×3J4
0.6J0.8
```

Disclose - Desempacotamento No formato `R←⊃Y` ou `R←↑Y`. Aqui há um problema: 2 símbolos a depender do nível de migração. Pode ser `⊃` ou `↑`. Fiquemos com o primeiro. Devolve o primeiro elemento do ravel de Y ou se este é vazio devolve o protótipo de Y.

```
      ⊃2 3⍴5+⍳5
6
      ⊃⍬
0
```

Divide - Divisão No formato `R←X÷Y`, aqui R é a divisão de X por Y. Se `⎕DVI` é 0 e Y é 0 então se X é 0 a divisão é 1. Se X é diferente de 0 a divisão dá um erro de domínio. Se `⎕DVI` é 1 e Y é 0, a divisão dá 0 para qualquer valor de X.

```
      ⎕DIV
0
      1 2 3 ÷ 2 3 4
0.5 0.6666666667 0.75
      4J6÷ 2 3J1
2J3 1.8J1.4
```

Drop No formato `R←X↓Y` elimina as extremidades selecionadas

```
       3↓⍳10
4 5 6 7 8 9 10
      ¯2↓⍳10
1 2 3 4 5 6 7 8
      1 ¯1↓3 3⍴⍳9
4 5
7 8
```

Enclose - Empacotamento Cujo formato é `R←⊂Y`, onde Y pode ser qualquer array. R é um array escalar formado por Y. Se Y é um escalar, R fica escalar e não muda. Caso contrário a profundidade de R é uma unidade maior do que a profundidade de Y

```
      ≡⊂4
0
      ≡4
0
      ⊂4 2⍴⍳8
 1 2
 3 4
 5 6
 7 8
      ⍴⊂4 2⍴⍳8

      ≡⍴⊂4 2⍴⍳8
1
```

O enclosed pode ser feito com índices k. Este é um vetor de zero ou mais eixos em Y.

Encode - Codificação No formato `R←X⊤Y`, devolve a representação de Y no sistema numérico definido por X

```
    2 2 2 2 ⊤7
0 1 1 1
      60 60⊤193
3 13
```

Enlist - Pertence à lista No formaro `R←∊Y` devolve os elementos como um simples vetor. Veja

```
      M←1 (2 2⍴2 3 4 5) (6(7 8))
      M
1   2 3   6   7 8
    4 5
      ,M
1   2 3   6   7 8
    4 5
      ⍴,M
3
      ∊M
1 2 3 4 5 6 7 8
      ⍴∊M
8
```

Equal - Igual No formato `R←X=Y`, retorna 1 se os operandos forem iguais e 0 senão. Depende de `⎕CT`. Não tem versão monádica.

```
      'aqui'=⍳4
0 0 0 0
          1=1.0000000000001
0
          1=1.00000000000001
1
         ⎕CT
1E¯14
```

Exclude - Subtração de conjunto No formato `R←X~Y`, devolve um vetor com os elementos de X que não estão em Y na ordem em que eles ocorrem em X. Na linguagem de conjuntos, $a\tilde{\ }b \Leftrightarrow a - b$

```
         'Curitiba'~'Parana'
Cuitib
      3 5 7 7 9~⍳6
7 7 9
```

Execute No formato `R←{X}⍎Y`, Y deve ser um escalar ou vetor de caracteres contendo uma expressão válida de APL. A expressão pode conter subexpressões separadas por `⋄`. X se presente deve ser um namespace no qual a expressão seja executada.

Tudo ocorre como se a expressão tivesse sido recém digitada: ela é executada

```
      ⍎'2+5'
7
```

Expand - Expansão No formato `R←X\[K]Y`. O eixo K é opcional e se ausenta implica o último eixo de Y

```
       1 ¯2 3 ¯4 5\1
1 0 0 1 1 1 0 0 0 0 1 1 1 1 1
      1 2 3\4 5 6
4 5 5 6 6 6
      1 0 1 1 0 1\'faca'
f ac a
```

No primeiro eixo, pode-se usar o símbolo `⍀`.

Exponential - Exponencial No formato `R←*Y`, Y deve ser numérico. R é numérico e corresponde à $Y-sima$ potência de e, a base dos logaritmos naturais

```
      *¯1 0 1 2
0.3678794412 1 2.718281828 7.389056099
```

Atentemos para a brilhante fórmula de Euler, aquela que engloba os 5 principais símbolos da matemática ($e^{\pi i}+1=0$)

```
      1+*○0j1
0
```

Factorial - Fatorial `R←!Y`. Quando Y é inteiro. Quando não é, equivale à função gamma de $Y+1$.

```
      !1 2 3 4 5
1 2 6 24 120
```

Find No formato `R←X⍷Y`, o resultado é um booleano com o formato de Y que identifica ocorrências de X em Y. Note que ao contrário de `∊` a ocorrência deve ser do X completo. Acompanhe

```
      3 4 5 ∊ ⍳10
1 1 1
      3 4 5 ⍷ ⍳10
0 0 1 0 0 0 0 0 0 0
      3 4 6 ⍷ ⍳10
0 0 0 0 0 0 0 0 0 0
```

Floor - Piso No formato `R←⌊Y` devolve o maior inteiro menor ou igual a Y. Para complexos, o resultado depende do relacionamento entre as partes reais e imaginárias do número

```
     ⌊ 1J0.1 2.9J¯1.1 3.3J3.3
1 2J¯1 3J3
     ⌊ 1 1.1 1.9 ¯1.1 ¯1.9 2
1 1 1 ¯2 ¯2 2
```

Format - Formatação No formato monádico `R←⍕Y` transforma Y de numérico para caracter. Números flutuantes dependem de `⎕PP`. A representação é igual à da tela.

Já o format diádico no formato `R←X⍕Y` permite controlar (via X) a apresentação da formatação de Y. Se qualquer elemento de Y é complexo, a função retorna erro de domínio.

```
      ⍕1234.567
1234.567
      ⍴⍕1234.567
8
      2 ⍕1234.567
#1234.57       ⍝ represento espaço por #
      12 1⍕1234.567
######1234.6
      12 ¯1⍕1234.567
########1.E3
```

Grade Down - Índices decrescentes No formato `R←⍒Y`, retorna os índices que ordenam Y em ordem decrescente. No formato diádico, `R←X⍒Y`, a ordenação base é aquela dada por X. Veja

```
      a←3 5 12 88 7 ¯2
      ⍒a
4 3 5 2 1 6
      a[⍒a]
88 12 7 5 3 ¯2
     b←'agora Vai MESMO'
      ⍒b
4 3 9 2 1 5 8 7 13 15 11 14 12 6 10
      b[⍒b]
roigaaaVSOMME
      'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'⍒b
6 10 4 3 9 2 1 5 8 7 13 15 11 14 12
      'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'⍒b
6 10 7 13 15 11 14 12 4 3 9 2 1 5 8
```

Grade Up - Índices crescentes Não vale a pena gastar muita pena aqui, mas é a mesma coisa do anterior só que agora em ordem crescente.

Greater - Maior No formato `R←X>Y` retorna 1 se $X > Y$ e 0 senão. Não tem versão monádica.

```
      1 5 77 > 2
0 1 1
      6>2 4⍴⍳8
1 1 1 1
1 0 0 0
```

Greater or equal - Maior ou igual No formato `R←X≥Y` retorna 1 se $X \geq Y$ e 0 senão. Não tem versão monádica.

```
      1 5 77 ≥ 2
0 1 1
      6≥2 4⍴⍳8
1 1 1 1
1 1 0 0
```

Index - Índice Esta é uma função mandrake no sentido de que ganhou símbolo apenas para coerência com o resto do APL. Na versão clássica era indicada por elementos entre colchetes e aqui o pecado mortal: usar 2 símbolos para uma única função, fugindo ao esquema geral da linguagem. O formato é `R←X⌷Y` e isto deve ser assim interpretado

```
(I J ... ⌷ Y) ≡ Y[I;J;...]
```

Note que este índice pode ser usado em especificações seletivas. Depende de `⎕IO`. Pode ser usado com o operador eixo

```
     a←'CuritibaParaná'
      a[2 5 8]
uta
      2 5 8⌷a    ⍝ dará erro: veja a definição acima
LENGTH ERROR
      2 5 8⌷a
           ^
      (⊂2 5 8)⌷a
uta
      a←3 4⍴⍳12
      a
1  2  3  4
5  6  7  8
9 10 11 12
      2⌷a
5 6 7 8
      1⌷[1]a
1 2 3 4
      1⌷[2]a
1 5 9
      2⌷[1]a
5 6 7 8
      2⌷[2]a
2 6 10
```

Index Generator - Gerador de índices No formato `R←⍳Y`. Y deve ser escalar ou vetor numérico inteiro positivo. Se Y é escalar a resposta é o vetor que começa em `⎕IO` e tem X elementos, com incremento de 1.

Quando Y é um vetor, a resposta é um array numérico composto pelo conjunto de todas as coordenadas possíveis em um array de forma Y. Veja

```
      ⎕IO←0
      ⍳5
0 1 2 3 4
      ⎕IO←1
      ⍳5
1 2 3 4 5
      ⍳2 3
 1 1  1 2  1 3
 2 1  2 2  2 3
```

Index of - Índice de No formato `R←X⍳Y`. Retorna a primeira ocorrência de Y em X. Se o elemento não pode ser achado, a resposta é `⎕IO+⊃⍴X`. O formato da resposta é o de Y.

```
      'uxih' ⍳ 'Curitiba'
5 1 5 3 5 3 5 5
      'Curitiba' ⍳ 'uxih'
2 9 4 9
      a←3 4 ⍴ ⍳ 12
      a
1  2  3  4
5  6  7  8
9 10 11 12
      a ⍳ 1 2 3 4
1
      a ⍳ 4 5 6 7
4
      a ⍳ 5 6 7 8
2
```

Indexing - Indexação No formato `R←X[Y]`, usa 2 caracteres para uma única função. Paciência. Índices são separados por `;`. Depende de `⎕IO`.

```
      a←10 20 30
      a[2 2⍴⍳3]
10 20
30 10
      'um' 'dois' 'tres' [2 1]
 dois  um
      c←3 4⍴⍳12
      c[1;3]
3
      c[;3]
3 7 11
```

Há um tipo de indexação chamado *choose indexing* no qual Y é um array não simples. Cada elemento de Y identifica um elemento de X por um conjunto de índices com um elemento por eixo de X na ordem principal.

```
      c←3 4⍴⍳12
      c[⊂1 3]
3
      c[(2 1) (2 2) (1 2)]
5 6 2
      s←5
      s[4⍴⊂⍳0]  ⍝ um escalar é indexado por vetor vazio empacotado (⊂⍳0)
5 5 5 5
```

Intersection - Intersecção No formato `R←X∩Y`. Ambos devem ser escalares ou vetores. Se forem escalares serão tratados como vetores unitários. A resposta é o conjunto de elementos que aparecem em X e Y na ordem determinada por X. Se o elemento é repetido em X e ocorre em Y aparecerá repetido na saída

```
      ¯2 11 3 66 7 9 11∩⍳12
11 3 7 9 11
      1 2  'mais' 'agora' ∩ 1 2 3 4
1 2
```

Interval Index - Intervalo de índices No formato `R←X⍸Y`. X é um array não escalar ordenado que representa conjuntos de intervalos. O resultado é um array inteiro que identifica em qual intervalo o correspondente valor de Y cai. Neste exemplo

```
      1 10 100 1000 ⍸ 5 50 50000
1 2 4
```

Neste exemplo, pode-se criar a seguinte tabela de intervalos

0	`<1`
1	`≥1 ∧ <10`
2	`≥10 ∧ <100`
3	`≥100 ∧ <1000`
4	`≥1000`

Se X não estiver em ordem crescente, há um erro de domínio.

```
      'aeiou'⍸'Curitiba'
0 5 4 3 4 3 1 1
      'aeiou'⍸'Curitiba' 1 2
0 5 5
```

Left - Esquerda No formato `R←X⊣Y` devolve X

```
      2⊣3
2
```

Less - Menor No formato `R←X<Y` retorna 1 se $X < Y$ e 0 senão. Não tem versão monádica.

```
      1 5 77 < 2
1 0 0
      6<2 4⍴⍳8
0 0 0 0
0 1 1 1
```

Less or equal - Menor ou igual No formato `R←X≤Y` retorna 1 se $X \leq Y$ e 0 senão. Não tem versão monádica.

```
      1 5 77 ≤ 2
1 0 0
      6≤2 4⍴⍳8
0 0 0 0
0 1 1 1
```

Logarithm - Logaritmo No formato `R←X⍟Y` devolve o logaritmo de Y na base X

```
      2 ⍟ 10 200
3.321928095 7.64385619
      10 ⍟ 1 10 200
0 1 2.301029996
      10 ⍟ 2J3
0.5569716762J0.4268218909
```

Note que o logaritmo diádico é definido em termos do logaritmo monádico como segue $\log_x Y \Leftrightarrow \frac{\log_e Y}{\log_e X}$ veja-se o exemplo: $\log_{10} 2 = 0.3010 = \frac{\log_e 2}{\log_e 10} = \frac{0.6931}{2.3025} = 0.3020$ a menos de erros de arredondamento. Em APL, portanto, `X⍟Y ≡(⍟Y)÷⍟X`.

Magnitude No formato `R←|Y`, se Y é real devolve o valor absoluto de Y e se é complexo, $(a + bi) = \sqrt{a^2 + b^2}$

```
      |1 ¯2 3 4
1 2 3 4
      |2J2
2.828427125
```

Match No formato `R←X≡Y` devolve 1 se X e Y são idênticos, 0 senão. Dois arrays não vazios são idênticos se têm a mesma estrutura e os mesmos elementos nos lugares equivalentes. Arrays nulos só são iguais se têm o mesmo protótipo.

```
      ⍬≡⍳0
1
      ⍬≡''
0
      (2 4⍴⍳8)≡'alfabeto'
0
```

Materialize No formato `R←⎕Y`, onde Y é uma referência a uma instância de classe com propriedade default. Se não for, retorna Y. Se for, retorna as propriedades da classe.

Matrix Divide - Divisão Matricial No formato `R←X⌹Y`, onde X e Y devem ser arrays numéricos simples de rank menor ou igual a 2. Y deve ser não singular, isto é, ter determinante diferente de 0. Um escalar é tratado como matriz unitária. Se é um vetor, é tratado como matriz coluna. O número de linhas de ambos deve ser igual. R é a divisão matricial de X por Y ou seja, `Y+.×R` é X. R é calculado de forma a minimizar `(X-Y+.×R)*2`

Dada a característica meio exotérica desta função, vamos a um exemplo um pouco mais elaborado. Seja o sistema

$$\begin{array}{llll} 3x & +4y & +2z & = & 25 \\ 9x & -y & -z & = & 1 \\ 5x & & +z & = & 10 \end{array}$$

que pode ser resolvido em APL com facilidade fazendo

```
      a←3 3⍴3 4 2 9 ¯1 ¯1 5 0 1
      25 1 10⌹a
1 3 5
      a+.×1 3 5
25 1 10
```

a interpretação é que $x = 1$, $y = 3$ e $z = 5$ é a solução do sistema.

Se o número de linhas é maior do que as colunas, a resposta usa o método dos mínimos quadrados para minimizar a solução. Se, no exemplo acima acrescentarmos outra equação, por exemplo $6x + 2y + z = 20$ a resposta agora é

```
      a←4 3⍴3 4 2 9 ¯1 ¯1 5 0 1  6 2 1
      a
3  4  2
9 ¯1 ¯1
5  0  1
6  2  1
      25 1 10 20⌹a
1.097888676 3.282149712 4.873320537
```

Obviamente, esta função aceita e devolve complexos, quando for o caso.

Matrix Inverse - Matriz Inversa No formato `R←⌹Y` devolve a matriz inversa de Y desde que esta seja não singular ($det(Y) \neq 0$). No mundo do cálculo numérico, a inversa tem inúmeras aplicações, que ficam a cargo do leitor. Este texto é sobre APL, não sobre cálculo numérico.

```
      a←3 3⍴3 4 2 9 ¯1 ¯1 5 0 1
      b←⌹a
      b
 0.02040816327  0.08163265306  0.04081632653
 0.2857142857   0.1428571429  ¯0.4285714286
¯0.1020408163  ¯0.4081632653   0.7959183673
      b+.×25 1 10    ⍝ cuidado que esta multiplicação NÃO é comutativa
1 3 5
```

Maximum - Máximo No formato `R←X⌈Y`, retorna o maior dos dois valores X ou Y

```
     1 2 3 4 ⌈ 4 3 2 1
4 3 3 4
     4⌈⍳8
4 4 4 4 5 6 7 8
```

Membership - Membro No formato `R←X∊Y` devolve um booleano no formato de X informando 1 se o elemento de X pertence a Y e 0 senão.

```
      1 6 10∊⍳7
1 1 0
      'Curitiba' ∊ 'Paraná'
0 0 1 0 0 0 0 1
      'Curitiba' ∊ ⍳10
0 0 0 0 0 0 0 0
```

Minimum - Mínimo No formato `R←X⌊Y`, retorna o menor dos dois valores X ou Y

```
      1 2 3 4 ⌊ 4 3 2 1
1 2 2 1
      4⌊⍳8
1 2 3 4 4 4 4 4
```

Minus - Menos Em `R←X-Y` devolve a subtração de X menos Y. Obviamente obedece à regra dos sinais. Veja

```
      1 2 ¯3 4 5 6 - 3
¯2 ¯1 ¯6 1 2 3
      3 - 1 2 ¯3 4 5 6
2 1 6 ¯1 ¯2 ¯3
      1 5 10 - 3 2 1
¯2 3 9
```

Mix Nos formatos `R←↑[K]Y` ou `R←⊃[K]Y`. Aqui neste Dyalog APL cujo `⎕ML` é 1, vale o primeiro formato. Então em `R←↑[K]Y`

```
      ⎕ML
1
      ↑(1 2)(3 4)(5 6)
1 2
3 4
5 6
      ↑(1 2)(3 4 5)(6 7)
1 2 0
3 4 5
6 7 0
```

Uma aplicação importante desta função é na leitura de um aquivo nativo via `⎕NGET`. O retorno é um array de vetores (cada linha é um vetor). Para transformar isso em uma matriz de caracteres (como é mais usual operar), usa-se função `↑`, como pode-se ver em

```
[0] r←{a}FMD arq;LF;aux;fi;ncol;nlin
[1] LF←⎕UCS 10
[2] →(0=⎕NC'a')/leitura
[3] gravacao:(,a,LF)⎕NPUT(arq)1
[4] r←a
[5] →fim
[6] leitura:aux←⎕NGET(arq)1
[7] r←↑⊃aux[1]
[8] fim:
```

Nesta função, a chamada monádica implica leitura e a diádica gravação. Note o uso de mix na linha 7 logo após a leitura.

```
      ↑'juan' 'alois' 'maria jose'
juan
alois
maria jose
      ⍴↑'juan' 'alois' 'maria jose'
3 10
```

Multiply - Multiplicação Em `R←X×Y`, o resultado é a multiplicação de X por Y.

```
      2 3 4 × 1 ¯2 5
2 ¯6 20
      3J5 × 0J1 1 3J6
¯5J3 3J5 ¯21J33
```

Nand - Not and Com formato `R←X⍲Y`, devolve a resposta conforme a tabela verdade

X	Y	R
0	0	1
0	1	1
1	0	1
1	1	0

Possivelmente uma das mais inúteis e menos usadas funções do APL, até porque basta negar a função and para obter este resultado.

Natural Logarith - Logaritmo natural Em `R←⍟Y` devolve o logaritmo de Y na base natural (número $e = 2.718281828$).

```
      ⍟1 2 3 4
0 0.6931471806 1.098612289 1.386294361
      ⍟1J1 2J2 3J3 4J0
0.3465735903J0.7853981634   1.039720771J0.7853981634   1.445185879J0.7853981634
 1.386294361            ⍝   quebrei em 2 para ficar mais fácil de ler...
```

Negative - Negativo Em `R←-Y` devolve o valor negativo de Y. Para complexos nega-se a parte real e a parte imaginária.

```
      - 1 2 ¯3 10 20
¯1 ¯2 3 ¯10 ¯20
      -1J1
¯1J¯1
      -1J¯1
¯1J1
```

Nest - Aninha cujo formato é `R←⊆Y`, se Y é simples, R é um escalar com o Y aninhado. Se não é simples, R é o Y inalterado.

```
      ]DISPLAY 1 2 3
┌→────┐
│1 2 3│
└~────┘
      ]DISPLAY ⊆ 1 2 3
┌─────────┐
│ ┌→────┐ │
│ │1 2 3│ │
│ └~────┘ │
└∊────────┘
      ]DISPLAY ⊆⊆ 1 2 3
┌─────────┐
│ ┌→────┐ │
│ │1 2 3│ │
│ └~────┘ │
└∊────────┘
```

Nor - Not or Com formato `R←X⍱Y`, devolve a resposta conforme a tabela verdade

X	Y	R
0	0	1
0	1	0
1	0	0
1	1	0

Vale o mesmo comentário do nand, vide acima.

Not - Negação No formato `R←~Y`, exige que Y seja booleano, respondendo 1 se Y é 0, e 0 senão.

```
      ~0 1
1 0
      ~~0
0
```

Not Equal - Diferente No formato `R←X≠Y`, devolve 1 se X é diferente de Y e 0 senão.

```
a←2 4 ⍴1 1 1 0 1
      a
1 1 1 0
1 1 1 1
      ~a
0 0 0 1
0 0 0 0
```

Not Match - Não coincidência No formato `R←X≢Y`. X e Y são quaisquer. R é 1 se X é idêntico a Y, 0 senão.

```
      (⍳5) ≢ 3 3⍴⍳6
1
      33 ≢ 'aa'
1
```

Or / mdc - Ou / Máximo divisor comum No formato `R←X∨Y`. Aqui se X e Y são booleanos, R é a função lógica OU (∨) cuja tabela verdade é

X	Y	R
0	0	0
0	1	1
1	0	1
1	1	1

Se um dos operandos X ou Y não são booleanos, a função é o máximo divisor comum.

```
      1 0 0 1 ∨ 1 1 0 0
1 1 0 1
      22 35 40 ∨ 50 50 50
2 5 10
```

Uma dica para relembrar o mmc e o mdc de acordo com os símbolos. O de máximo divisor comum – mdc, é o de um número menor que os operandos. Daí o seu símbolo aponta para baixo (∨). Já o mínimo múltiplo comum – mmc, é um número maior que X e Y. O seu símbolo aponta para cima (∧).

Partition - Partição No formato `R←X⊆[K]Y`, devolve os elementos de Y particionados de acordo com X. X deve ser um vetor de inteiros.

```
      ]display 1 1 1 2 2 3 3 3 ⊆ 'Curitiba'
┌→──────────────────┐
│ ┌→──┐ ┌→─┐ ┌→──┐ │
│ │Cur│ │it│ │iba│ │
│ └───┘ └──┘ └───┘ │
└∊──────────────────┘
      ]display 1 1 1 0 0 0 1 1    ⊆ 'Curitiba'
┌→───────────┐
│ ┌→──┐ ┌→─┐ │
│ │Cur│ │ba│ │
│ └───┘ └──┘ │
└∊───────────┘


      ]display 1 1 1 0 0 0 5 6    ⊆ 'Curitiba'
┌→──────────────┐
│ ┌→──┐ ┌→┐ ┌→┐ │
│ │Cur│ │b│ │a│ │
│ └───┘ └─┘ └─┘ │
└∊──────────────┘
```

Partitioned Enclose - Partição anexada No formato `R←X⊂[K]Y`. X deve ser um vetor de inteiros. R é um vetor de itens selecionados de Y pela inserção de 0 ou mais divisores, especificados por X entre as células principais.

```
      ]display 0 0 1 0 0 0 0 1 0 0 ⊂ ⍳9
┌→──────────────────────┐
│ ┌→────────┐ ┌→──┐ │
│ │3 4 5 6 7│ │8 9│ │
│ └~────────┘ └~──┘ │
└∊──────────────────────┘
      ]display 1 0 1⊂[1] 3 5⍴⍳15
┌→──────────────────────────────────┐
│ ┌→─────────┐ ┌→─────────────┐ │
│ ↓1 2 3 4  5│ ↓11 12 13 14 15│ │
│ │6 7 8 9 10│ └~──────────────┘ │
│ └~─────────┘                    │
└∊──────────────────────────────────┘
```

Pi Times - Vezes Pi No formato `R←○Y`, devolve o valor de Y multiplicado por $\pi = 3.141592654$.

```
      180÷○1   ⍝ quantos graus vale 1 radiano ?
57.29577951
```

Pick - Escolha No formato `R←X⊃Y`, Y pode ser qualquer e X é um vetor de índices de Y. R é um item selecionado da estrutura de Y de acordo com X.

Plus - Adição Cujo formato é `R←X+Y` é a operação de adição, sem maiores comentários.

Power - Potência O formato é `R←X*Y` e o valor de R é X elevado à potência Y. Se Y é 0, R é 1. Se X é 0, Y deve ser não negativo. Se X é negativo, o resultado R provavelmente é complexo.

```
      1 2 0 ¯2 ¯3 4 * 0 ¯1 2 ¯3 1.8 ¯5
1  0.5  0  ¯0.125  5.84488409J¯4.246556863  0.0009765625
```

Ravel - Vetorização No formato `R←,Y` ou `R←,[K]Y` devolve um vetor no primeiro caso e um array no segundo com os elementos de Y

```
      m←3 4⍴⍳12
      ,m
1 2 3 4 5 6 7 8 9 10 11 12
      ⍴,[0.5]m
1 3 4
      ⍴,[1.5]m
3 1 4
      ⍴,[2.5]m
3 4 1
```

Recíprocal - Recíproco No formato `R←÷Y`. R é o resultado de `1÷R`. Depende de `⎕DIV`

```
      ÷ 4 5 2
0.25 0.2 0.5
```

Replicate - Replicação No formato `R←X/[K]Y`, Y pode ser qualquer. X é um vetor.

```
      1 0 1/'ABC'
AC
      1 ¯2 3 4/⍳4
1 0 0 3 3 3 4 4 4 4
      m←3 4⍴⍳12
      1 0 1/[1]m
1  2  3  4
9 10 11 12
      1 0 0 1/[2]m
1  4
5  8
9 12
```

Reshape - Reformatação No formato `R←X⍴Y`, R tem os elementos do ravel de Y com a forma de X. Os elementos de Y são tomados ciclicamente ou desprezados, conforme o caso.

```
      2 4⍴⍳5
1 2 3 4
5 1 2 3
      2 4⍴⍳20
1 2 3 4
5 6 7 8
      2 3⍴0
0 0 0
0 0 0
```

Residue - Resto No formato `R←X|Y`. Para argumentos positivos, R é o resto quando Y é dividido por X. Se $X = 0$, R é Y.

```
     4 4 ¯4 1 | 10 17 10 13
2 1 ¯2 0
     1j2|5j3
¯1J1
```

Reverse - Reverso No formato `R←⌽[K]Y`, Y pode ser qualquer. O eixo é opcional e se ausente, vale o último. Para o primeiro eixo, o símbolo pode ser `⊖`.

```
      ⌽⍳5
5 4 3 2 1
      m ← 3 4 ⍴ ⍳ 12
      ⌽[1]m
9 10 11 12
5  6  7  8
1  2  3  4
      ⌽m
 4  3  2 1
 8  7  6 5
12 11 10 9
      ⊖m
9 10 11 12
5  6  7  8
1  2  3  4
```

Right - Direito No formato `R←X⊢Y`. O resultado R é Y.

```
      1 2 3 ⊢ 'alfa' 123
 alfa  123
```

Roll - Rolo No formato `R←?Y`. Gera um aleatório dentro do `⍳` do elemento correspondente de Y. A forma de R é a mesma de Y. Há um caso particular quando Y vale 0. Neste caso a resposta é um real entre 0 e 1 uniformemente distribuído.

```
      ?3 4 5
3 1 2
      ?3 3 ⍴50
 8 28 39
43 31 41
11  1 18
      ?0
0.3787086806
      ?4⍴0
0.3128095179 0.8912442253 0.5662684266 0.8561391733
```

Rotate - Rotação No formato `R←X⌽[K]Y`. Y pode ser qualquer array. X deve ser um array simples de inteiros. R tem os elementos rotacionados.

```
      3⌽⍳10
4 5 6 7 8 9 10 1 2 3
      ¯2⌽⍳10
9 10 1 2 3 4 5 6 7 8
      m←3 4⍴⍳12
      1 2 3⌽m    ⍝ equivale a 1 2 3⌽[2]m
 2 3  4  1
 7 8  5  6
12 9 10 11
      1 2 3 4⌽[1]m
5 10  3  8
9  2  7 12
1  6 11  4
```

Same - Mesmo Nos formatos `R←⊣Y` ou `R←⊢Y`, não faz nada e devolve Y

```
      ⊣1 2 3
1 2 3
      ⊢'abc'
abc
```

Shape - Forma No formato `R←⍴Y` devolve a forma de R que é um vetor de inteiros positivos. Cada elemento de R informa uma dimensão de Y. Se Y é um escalar, R é um vetor vazio. O rank de Y é dado por `⍴⍴Y`.

```
      m←3 4 ⍴ ⍳12
      ⍴m
3 4
      ⍴⍴m
2
```

Split - Cisão No formato `R←↓[K]Y`, Y pode ser qualquer. K é opcional e se ausente implica no último. Os ítens de R são os sub-arrays de Y. O rank diminuir uma unidade

```
      ]display a← 3 4⍴⍳12
┌→──────────┐
↓1  2  3  4│
│5  6  7  8│
│9 10 11 12│
└~──────────┘
      ↓a
 1 2 3 4   5 6 7 8   9 10 11 12
      ⍴↓a
3
      ]display ↓a
┌→────────────────────────────────────┐
│ ┌→──────┐ ┌→──────┐ ┌→─────────┐ │
│ │1 2 3 4│ │5 6 7 8│ │9 10 11 12│ │
│ └~──────┘ └~──────┘ └~─────────┘ │
└∊────────────────────────────────────┘
    ]display ↓[1]a
┌→──────────────────────────────────────┐
│ ┌→────┐ ┌→─────┐ ┌→─────┐ ┌→─────┐ │
│ │1 5 9│ │2 6 10│ │3 7 11│ │4 8 12│ │
│ └~────┘ └~─────┘ └~─────┘ └~─────┘ │
└∊──────────────────────────────────────┘
```

Subtract - Subtração No formato `R←X-Y`, R é a diferença entre X e Y.

Table - Tabela No formato `R←⍪Y`, R é matriz de 2 dimensões, com os elementos de Y tomados na ordem maior (a primeira), preservando as formas da primeira dimensão de Y se existir

```
      a ← 3 4 ⍴ ⍳ 12
      a⍪⍳4
1  2  3  4
5  6  7  8
9 10 11 12
1  2  3  4
      a⍪[2]⍳3
1  2  3  4 1
5  6  7  8 2
9 10 11 12 3
```

Take - Tomar No formato `R←X↑Y` toma pela frente (se X positivo) ou por trás (se X negativo) os elementos de Y. Pode ter uma especificação de eixo (pág. 132)

```
      3↑⍳8
1 2 3
      ¯2↑⍳8
```

```
7 8
      5↑⍳3
1 2 3 0 0
      ¯5↑⍳3
0 0 1 2 3
     2 3↑3 4 ⍴ ⍳ 12
1 2 3
5 6 7
```

Tally - Contador No formato `R←≢Y` retorna o número de células maiores em Y

```
    ≢3 4⍴⍳12
3
      ≢⍳8
8
```

Times - Multiplicação No formato `R←X×Y` devolve a multiplicação entre X e Y.

Transpose - Transposta No formato `R←⍉Y` devolve a transposta de Y

```
     ⍉3 4⍴⍳12
1 5  9
2 6 10
3 7 11
4 8 12
     ⍴ ⍉3 4⍴⍳12
4 3
```

Tem também a versão diádica no formato `R←X⍉Y` cuja descrição é bastante complicada. Veja na pág 135 e 136.

Type - Tipo No formato `R←∊Y` devolve um array de mesmo formato de Y onde valores numéricos são trocados por 0 e valores alfa por ' '.

Union - União No formato `R←X∪Y` exige que X e Y sejam vetores. Devolve os elementos de X conactenados com os elementos de Y que não estão em X

```
      (⍳6)∪1 5 10 20
1 2 3 4 5 6 10 20
      'Curitiba'∪'Paraná'
CuritibaPná
```

Unique - Único No formato `R←∪Y` retorna os elementos principais únicos de Y. Elementos no caso de vetor, linhas no caso de matriz e assim por diante, na ordem em que eles aparecem.

```
      ∪ 1 3 5 6 3 2 4 5 6 4
1 3 5 6 2 4
      ∪ 3 4⍴1 2 3 7 0 0 9 8 1 2 3 7
1 2 3 7
0 0 9 8
```

Unique Mask - Máscara Única No formato `R←≠Y` retorna um vetor booleano cujo comprimento equivale ao número de células majoritárias (elementos principais) de R. O valor é 1 na sua primeira ocorrência e 0 nas demais.

```
      ≠⍳5
1 1 1 1 1
      ≠1 2 1 2 3 1 2 3 4
1 1 0 0 1 0 0 0 1
      ≠ 3 4⍴1 2 3 7 0 0 9 8 1 2 3 7
1 1 0
```

Where - Onde No formato `R←⍸Y`. *Y* deve ser array numérico de inteiros não negativos e a resposta é `{(,⍵)/,⍳⍴⍵}`. Simples assim

```
      ⍸3 4 5 6 7
1 1 1 2 2 2 2 3 3 3 3 3 4 4 4 4 4 4 5 5 5 5 5 5 5
      ⍸1 0 1 0
1 3
      ⍸3 1
1 1 1 2
```

Resumo das funções NOVAS em Dyalog APL Veja a seguir

função	o que faz	exemplo
`a ∧ b`	Devolve o mínimo múltiplo comum, quando a e b não booleanos	`   22 33 44 ∧ 4 5 6` `44 165 132`

`∊b`	devolve elementos de b como vetor simples	`   M←1 (2 2⍴2 3 4 5) (6(7 8))`
		`   ∊M`
		`1 2 3 4 5 6 7 8`

`a~b`	subtração de conjuntos	`'   Curitiba'~'Parana'` `Cuitib`

`⍳a`	a como vetor	`   ⍳2 3`
		`1 1   1 2   1 3`
		`2 1   2 2   2 3`

`a\b`	a não booleana	`   1 2 3 \ 4 5 6` `4 5 5 6 6 6`
`a⍷b`	ocorrência completa de a em b	`   3 4 5⍷⍳10` `0 0 1 0 0 0 0 0 0 0`
`a∩b`	ambos vetores: conjunto união	
`a⍸b`	intervalor de índices	`   1 10 100 1000 ⍸ 5 50 50000` `1 2 4`
`a⊣b`	esquerdo: retorna a	

`↑a`	devolve matriz de vetor de vetores	`   ↑(1 2)(3 4 5)`
		`1 2 0`
		`3 4 5`

`⊆a`	se a é simples, retorna `⊂a`. Senão, não faz nada	
`a∨b`	devolve o máximo divisor comum, quando a e b não booleanos	`   22 35 40 ∨ 50 50 50` `2 5 10`
`a⊆b`	partição	`   1 1 1 2 2 3 3 3 ⊆ 'curitiba'` ` cur  it  iba`
`a⊂b`	partição anexada	`   1 0 0 1 0 1 0 0 ⊂ 'curitiba'` ` cur  it  iba`
`a/b`	replicação	`   1 0 ¯2 3/⍳4` `1 0 0 4 4 4`
`a⊢b`	direito: retorna b	
`?0`	retorna real entre 0 e 1	
`↓a`	split: diminui o rank uma unidade	`   a←3 4⍴⍳12 ⋄ ↓a` `1 2 3 4   5 6 7 8   9 10 11 12`
`a∪b`	união: exige ambos vetores.	
`∪a`	únicos: retorna elementos principais únicos	
`≠a`	máscara única: retorna booleano com 1 na primeira ocorrência e 0 nas demais	
`⍸a`	where> `{(,⍵)/,⍳⍴⍵}`	

Operadores novos

Each Este operador monádico evita a necessidade de processar os itens de um array, um após o outro em um loop explícito. Traduzido por "cada", e representado pelo sinal dieresis (¨). Pode ser usado com funções de maneira monádica ou diádica.

Each com operando monádico Seu formato é

`{r}←f¨Y`

`f` pode ser qualquer função monádica. `Y` pode ser qualquer array. Cada um dos itens de Y recebe a ação da função f. Veja alguns exemplos

```
      ⍳5
1 2 3 4 5
      ⍳¨⍳5
 1  1 2  1 2 3  1 2 3 4  1 2 3 4 5
      G←('ola' (⍳3)) ('fulano' (⍳4)('ciclano'(⍳5)))
      G
```

```
  ola  1 2 3    fulano  1 2 3 4   ciclano  1 2 3 4 5
      ]display G
┌→──────────────────────────────────────────────────────────────────────┐
│ ┌→──────────────┐ ┌→──────────────────────────────────────────────┐ │
│ │ ┌→──┐ ┌→────┐ │ │ ┌→─────┐ ┌→──────┐ ┌→────────────────────────┐ │ │
│ │ │ola│ │1 2 3│ │ │ │fulano│ │1 2 3 4│ │ ┌→──────┐ ┌→────────┐ │ │ │
│ │ └───┘ └~────┘ │ │ └──────┘ └~──────┘ │ │ciclano│ │1 2 3 4 5│ │ │ │
│ └∊──────────────┘ │                    │ └───────┘ └~────────┘ │ │ │
│                   │                    └∊────────────────────────┘ │ │
│                   └∊──────────────────────────────────────────────┘ │
└∊──────────────────────────────────────────────────────────────────────┘

      ⍴G
2
      ⍴¨G
 2  3
      ⍴¨¨G
  3  3    6  4  2

      z← ¨G
      z
 fulano  1 2 3 4   ciclano  1 2 3 4 5
      ⍴z
3
      ]display z
┌→───────────────────────────────────────────────┐
│ ┌→─────┐ ┌→──────┐ ┌→────────────────────────┐ │
│ │fulano│ │1 2 3 4│ │ ┌→──────┐ ┌→────────┐ │ │
│ └──────┘ └~──────┘ │ │ciclano│ │1 2 3 4 5│ │ │
│                    │ └───────┘ └~────────┘ │ │
│                    └∊────────────────────────┘ │
└∊───────────────────────────────────────────────┘
       w←¨¨G
      w
 ciclano  1 2 3 4 5
      ⍴w
2
      ]display w
┌→─────────────────────────┐
│ ┌→──────┐ ┌→────────┐ │
│ │ciclano│ │1 2 3 4 5│ │
│ └───────┘ └~────────┘ │
└∊─────────────────────────┘
```

Each com operando diádico Seu formato é

```
{r}←Xf¨Y
```

`f` pode ser qualquer função diádica. `X` e `Y` são quaisquer arrays conformáveis. A função é aplicada aos pares formados entre X e Y. Veja alguns exemplos:

```
      1 2 3 4÷3 4 5 6
0.3333333333 0.5 0.6 0.6666666667
      1 2 3 4 ÷¨3 4 5 6
0.3333333333 0.5 0.6 0.6666666667
      'ABC','xyz'
ABCxyz
      'ABC',¨'xyz'
 Ax  By  Cz
      G←(1 (2 3))(4 (5 6))(8 9) 10
      G
 1  2 3   4  5 6   8 9  10
      1⌽G
 4  5 6   8 9  10  1  2 3
```

```
      1⌽¨G
 2 3  1   5 6  4  9 8  10
      1⌽¨¨G
 1  3 2   4  6 5   8 9  10
      1 2 3 4↑¨G
 1  4  5 6   8 9 0  10 0 0 0
```

At Substitui iens selecionados em Y por novos valores ou aplica uma função para modificar itens selecionados em Y. Seu formato

```
r←{X}(f@g)Y
```

O operando g identifica quais itens do array Y são substituídos ou modificados. O g é, de duas, uma

- um array que especifica um conjunto de índices em Y. Se ele for um escalar ou um vetor especifica células majoritárias (linhas em matrizes, p. exemplo) em Y.
- ou uma função que quando aplicada a Y retorna um array booleano de mesmo formato de Y (uma máscara) na qual um valor 1 indica que o elemento vai ser substituído ou modificado. Note que o ravel da máscara indica algo sobre o ravel de Y.

O operando esquerdo f, de duas uma

- Um array que contém os valores a substituir os itens em Y identificados por g
- ou uma função que se aplica a tais itens, cujo resultado os substitui. Se a função é diádica o argumento esquerdo é X

O resultado r é o mesmo Y com os itens especificados em g substituidos ou modificados por f. Exemplos

```
      (10 20@2 4)⍳5 ⍝ substitui o seg. e quarto em ⍳5 por 10 e 20
1 10 3 20 5
       10 20@2 4⍳5 ⍝ o parênteses não é necessário
1 10 3 20 5
      (2 3⍴10 11)(@2 4)4 3⍴⍳12
 1  2  3   ⍝ substitua as linhas (células
10 11 10   ⍝ majoritárias da matriz 4 3
 7  8  9   ⍝ por linhas (circulares) contendo
11 10 11   ⍝ 10 e 11
      (0@(1 1)(4 3))4 3⍴⍳12
 0  2 3   ⍝ substitua o primeiro (1 1)
 4  5 6   ⍝ e o último (4 3) elementos
 7  8 9   ⍝ na matriz 4x3 por zero
10 11 0
      ÷@2 4⍳5 ⍝ substitua o 2. e o 4. em ⍳5
1 0.5 3 0.25 5  ⍝ pelos seus inversos
      10×@2 4⍳5
1 20 3 40 5 ⍝ multiplique por 10 o 2. e o 4. em ⍳5
      0@(2∘|)⍳5 ⍝ troque os impares por 0
0 2 0 4 0
      ÷@(2∘|)⍳5 ⍝ troque os impares pelo recíproco
1 2 0.3333333333 4 0.2
```

Atop Seu símbolo é ⍤ e seu formato

```
{r}←{X}f⍤gY
```

A função derivada é equivalente a `fgY` ou `fXgY` e não precisa retornar resultado. Acompanhe

```
    3 1 4 1 5 ~∊1 2 3
4 5
    3 1 4 1 5 ~⍤∊1 2 3
0 0 1 0 1
```

Beside Cuja tradução é *ao lado*. Usa o símbolo ∘ e tem formato

```
{r}←{X}f∘gY
```

g pode ser qualquer função monádica que retorne resultado. Y deve ser apropriada para a função g, com gY adequada ao argumento direito de f.

Se X for omitido, f deve ser monádico. A função derivada é equivalente a fgY ou $XfgY$ e não precisa retornar.

```
      ⍳¨2 4 6
 1 2  1 2 3 4  1 2 3 4 5 6
      +/∘⍳¨2 4 6
3 10 21
      33,⍳¨⍳3
33  1  1 2  1 2 3
      33,∘⍳¨⍳3
33 1  33 1 2  33 1 2 3
```

Bind Cuja tradução é *ligar, ligação*. Seu formato é

```
{r}←A∘fY ou {r}←(f∘B)Y
```

Ele conecta um array A ou B a uma função dinâmica f ou como seu argumento esquerdo ou direito respectivamente. O primeiro pode ser descrito como carregamento à esquerda e o segundo como carregamento à direita. A, B e Y podem ser arrays quaisquer cujos itens sejam apropriados à função f. No caso de B ser usado como argumento direito de f os parênteses são necessários para distinguir entre B e Y. Este exemplo usa ambas as formas de bind para listar as funções do workspace

```
      ⎕∘←∘⎕VR¨↓⎕NL 3
...
      (*∘0.5)4 16 25
2 4 5
```

Commute Tem formato

```
{r}←{X}f⍨Y
```

f pode ser qualquer função diádica. X e Y quaiquer arrays que sejam apropriados para f. A função derivada é equivalente a YfX. Se X é omitido, o Y é duplicado e ocupa seu lugar. MEIO INÚTIL, NÃO ?????

```
      N
3 2 5 4 6 1 3
      N/⍨2|N     ↔   (2|N)/N
3 5 1 3
      ⍴⍨3   ↔   3⍴3
3 3 3
```

Produto Interno É uma generalização da multiplicação matricial. Esta, tem como formato

```
r←X+.×Y  →  generalização  r←Xf.gY
```

A generalização acima sugere que ao inves de usar + e × pode-se usar quaisquer funções diádicas. Veja-se por exemplo

```
      (⍳10)+.×⍳10
385
      nomes←↑'pedro' 'paula' 'felipe' 'ana'
pedro
paula
felipe
ana
      ⍴nomes
4 6
      nomes∧.=6↑'pedro'
1 0 0 0
```

Key Traduzido por chave. Funciona de maneira similar à cláusula GROUP BY do SQL. Em Dyalog chama-se com CTRL-SHIFT-K. Seu formato.

`r←{X}f⌸Y`

Aplica a função f a cada chave única em X e as células majoritárias de Y que tem tal chave. Se X é omitido, Y é um array cujas células majoritárias representam chaves. Neste caso f é aplicado a cada chave única e aos elementos que têm tal chave. Os elementos em r aparecem na ordem que primeiro aparecem em Y. Veja o exemplo

```
      car←'2' 'J' 'A' '4' '6'
      nai←'paus' 'espadas' 'paus' 'ouros' 'copas'
      nai,[1.5]car
 paus    2
 espadas J
 paus    A
 ouros   4
 copas   6
      nai {⍺':'⍵} car
  paus  espadas  paus  ouros  copas  : 2JA46
      nai {⍺':'⍵} ⌸ car
  paus     : 2A
  espadas  : J
  ouros    : 4
  copas    : 6
      {⍺ ⍵} ⌸ nai
  paus      1 3    ⍝ índices dos naipes
  espadas   2
  ouros     4
  copas     5
      {⍺,≢⍵}⌸nai
 paus      2   ⍝ quantas cartas de cada naipe
 espadas   1
 ouros     1
 copas     1
```

Produto Externo Tem o formato

`{r}←X∘.gY`

g é aplicada a cada par de elementos de X e Y. Veja

```
      (⍳4)∘.×⍳4
1 2  3  4
2 4  6  8
3 6  9 12
4 8 12 16
      1 2 3∘.⍴'ABC'
 A    B    C
 AA   BB   CC
 AAA  BBB  CCC
```

Over - Sobre No formato `{R}←{X}f⍥gY`. Gera a função `fgY` ou `(gX)f(gY)`. Este operador permite que as operações sejam coladas para criar funções mais complexas.

```
      ]display 2 3,⊂'texto'
┌→──────────────┐
│      ┌→────┐  │
│ 2 3  │texto│  │
│      └─────┘  │
└∊──────────────┘
      ]display 2 3,⍥⊂ 'texto'
┌→────────────────┐
│ ┌→──┐  ┌→────┐  │
│ │2 3│  │texto│  │
│ └~──┘  └─────┘  │
└∊────────────────┘
```

Power - Potência No formato `{R}←{X}(f⍣g)Y`. Um valor de g negativo aplica a inversa `|g` vezes.

```
      f←(32∘+)∘(×∘1.8)
      f 0 100
32 212
      c←f⍣¯1
      c 32 212
0 100
      invs←{(⍺⍺⍣¯1)⍵} ⍝ operador inverso
      +\invs 1 3 6 10
1 2 3 4
      2 ∘⊥invs 9
1 0 0 1
```

Rank No formato `R←{X}(f⍤B)Y`. Aplica f sucessivamente aos sub arrays de Y ou a operação diádica F entre os sub arrays de X e Y. Sub-arrays são selecionados pelo operador B.

```
      y←2 3 4⍴24?24
      y
19 18  3  2
22  7 10 16
14  8 21 17
         15 24  1  4
20 13 11 12
 5 23  6  9
      ]display ⊂⍤2⊢y
┌→──────────────────────────────┐
│ ┌→───────────┐ ┌→───────────┐ │
│ ↓19 18  3  2│ ↓15 24  1  4│ │
│ │22  7 10 16│ │20 13 11 12│ │
│ │14  8 21 17│ │ 5 23  6  9│ │
│ └~───────────┘ └~───────────┘ │
└∊──────────────────────────────┘
      ]display ⊂⍤1⊢y
┌→──────────────────────────────────────────┐
↓ ┌→──────────┐ ┌→──────────┐ ┌→──────────┐ │
│ │19 18 3 2│ │22 7 10 16│ │14 8 21 17│ │
│ └~──────────┘ └~──────────┘ └~──────────┘ │
│ ┌→──────────┐ ┌→──────────┐ ┌→──────────┐ │
│ │15 24 1 4│ │20 13 11 12│ │5 23 6 9│ │
│ └~──────────┘ └~──────────┘ └~──────────┘ │
└∊──────────────────────────────────────────┘
```

Reduce - Redução No formato `R←f/[K]Y`, é a redução tradicional do APL

```
      +/1 2 3 4
10
      z←3 4⍴⍳12
      +/[1]z
15 18 21 24
      +/[2]z
10 26 42
```

Reduce N-wise No formato `R←Xf/[K]Y`. R é o array formado pela aplicação da função F entre itens de sub vetores de comprimento X de Y.

```
      3+/⍳4 ⍝ (1+2+3) (2+3+4)
6 9
      2+/⍳4 ⍝ (1+2) (2+3) (3+4)
3 5 7
      1+/⍳4 ⍝ (1) (2) (3) (4)
1 2 3 4
      0+/⍳4 ⍝ elemento identidade para +
0 0 0 0
```

```
      0×/⍳4 ⍝ elemento identidade para ×
1 1 1 1
      2,/⍳4 ⍝ (1,2) (2,3) (3,4)
1 2   2 3   3 4
      ¯2,/⍳4 ⍝ (2,1) (3,2) (4,3)
2 1   3 2   4 3
```

Scan No formato `R←f\[K]Y`. Versão tradicional do scan

```
      +\⍳4
1 3 6 10
      +\3 4⍴⍳12
1  3  6 10
5 11 18 26
9 19 30 42
      +\[1]3 4⍴⍳12
 1  2  3  4
 6  8 10 12
15 18 21 24
      ,\'abcde'
a ab  abc  abcd  abcde
```

Spawn - Desovar No formato `{R}←{X}f&Y`. Este operador gera uma nova thread na qual f é aplicada a Y ou entre os argumentos X e Y. O resultado é o número da thread.

```
      ⎕←÷&4
1
0.25
```

Stencil - Estêncil No formato `R←(f⌺g)Y`. Conhecido como movimento de janela. Aplica f a uma série de retângulos (possivelmente overpostos) em Y.

```
      ]display {⊂⍺ ⍵}⌺3 3⊢3 3⍴⍳12
┌→───────────────────────────────────────────────────────┐
↓ ┌→──────────────┐ ┌→──────────────┐ ┌→───────────────┐ │
│ │ ┌→──┐ ┌→────┐ │ │ ┌→──┐ ┌→────┐ │ │ ┌→───┐ ┌→────┐ │ │
│ │ │1 1│ ↓0 0 0│ │ │ │1 0│ ↓0 0 0│ │ │ │1 ¯1│ ↓0 0 0│ │ │
│ │ └~──┘ │0 1 2│ │ │ └~──┘ │1 2 3│ │ │ └~───┘ │2 3 0│ │ │
│ │       │0 4 5│ │ │       │4 5 6│ │ │        │5 6 0│ │ │
│ │       └~────┘ │ │       └~────┘ │ │        └~────┘ │ │
│ └∊──────────────┘ └∊──────────────┘ └∊───────────────┘ │
│ ┌→──────────────┐ ┌→──────────────┐ ┌→───────────────┐ │
│ │ ┌→──┐ ┌→────┐ │ │ ┌→──┐ ┌→────┐ │ │ ┌→───┐ ┌→────┐ │ │
│ │ │0 1│ ↓0 1 2│ │ │ │0 0│ ↓1 2 3│ │ │ │0 ¯1│ ↓2 3 0│ │ │
│ │ └~──┘ │0 4 5│ │ │ └~──┘ │4 5 6│ │ │ └~───┘ │5 6 0│ │ │
│ │       │0 7 8│ │ │       │7 8 9│ │ │        │8 9 0│ │ │
│ │       └~────┘ │ │       └~────┘ │ │        └~────┘ │ │
│ └∊──────────────┘ └∊──────────────┘ └∊───────────────┘ │
│ ┌→───────────────┐ ┌→───────────────┐ ┌→────────────────┐ │
│ │ ┌→───┐ ┌→────┐ │ │ ┌→───┐ ┌→────┐ │ │ ┌→────┐ ┌→────┐ │ │
│ │ │¯1 1│ ↓0 4 5│ │ │ │¯1 0│ ↓4 5 6│ │ │ │¯1 ¯1│ ↓5 6 0│ │ │
│ │ └~───┘ │0 7 8│ │ │ └~───┘ │7 8 9│ │ │ └~────┘ │8 9 0│ │ │
│ │        │0 0 0│ │ │        │0 0 0│ │ │         │0 0 0│ │ │
│ │        └~────┘ │ │        └~────┘ │ │         └~────┘ │ │
│ └∊───────────────┘ └∊───────────────┘ └∊────────────────┘ │
└∊───────────────────────────────────────────────────────┘
```

Variant - Variante Não consegui decobrir como gerar o quad com 2 pontos dentro.

Um resumo das novidades no capítulo operadores Veja a seguir

<table>
<tr><td>operador</td><td>o que faz</td><td>exemplo</td></tr>
<tr><td><code>{|s}f/Y</code></td><td>redução: f entre todos os itens de Y em grupos de <code>|s</code> no último eixo</td><td><code>   2+/⍳5</code>
<code>3 5 7 9</code></td></tr>
<tr><td><code>f¨Y</code></td><td>each monádico</td><td><code>   ⍳¨1 2 3 4 5</code>
<code>1 1 2 1 2 3 1 2 3 4 1 2 3 4 5</code></td></tr>
<tr><td><code>Xf¨Y</code></td><td>each diádico</td><td><code>   'ABC','xyz'</code>
<code>'ABCxyz'</code>
<code>   'ABC',¨'xyz'</code>
<code>Ax By Cz</code></td></tr>
<tr><td><code>{X}(f@g)Y</code></td><td>At: itens em g modificados por f</td><td><code>   ÷@2 3⍳5</code>
<code>1 0.5 0.3333333333 4 5</code>
<code>   10×@2 4⍳5</code>
<code>1 20 3 40 5</code></td></tr>
<tr><td><code>{X}f⍤gY</code></td><td>Atop: <code>fgy</code> ou <code>fXgY</code></td><td><code>   1 5 9-⍤+2 4 6</code>
<code>¯3 ¯9 ¯15</code></td></tr>
<tr><td><code>Xf[K]Y</code></td><td>eixo com operação diádica</td><td><code>   mat←10×2 3⍴⍳6</code>
<code>   mat+[1]1 2</code>
<code>11 21 31</code>
<code>42 52 62</code></td></tr>
<tr><td><code>{X}f∘gY</code></td><td>compose: <code>fgy</code> ou <code>XfgY</code> (beside)</td><td></td></tr>
<tr><td><code>X∘gY</code></td><td>compose: g entre X e Y ou seja, <code>XgY</code></td><td></td></tr>
<tr><td><code>(f∘Y2)Y1</code></td><td>compose: f entre Y_1 e Y_2 ou seja <code>Y1 f Y2</code></td><td></td></tr>
<tr><td><code>{X}f⌸Y</code></td><td>Key:f sobre os itens de Y agrupados por valores únicos de Y</td><td><code>   val←5 1 10 20 3 ⋄ dia←2 3 2 4 3</code>
<code>   dia{(+/⍵)÷⍴⍵}⌸val</code>
<code>7.5 2 20 ⍝ medias diarias</code></td></tr>
<tr><td><code>{X}f⍨Y</code></td><td>Commute: o mesmo que <code>YfX</code> ou <code>YfY</code> (se monádico)</td><td><code>   N←3 2 5 4 6 1 3</code>
<code>   N/⍨2|N ←→ (2|N)/N</code>
<code>3 5 1 3</code>
<code>   ⍴⍨3   ←→   3⍴3</code>
<code>3 3 3</code></td></tr>
<tr><td><code>{X}f&Y</code></td><td>spawn: f em Y em uma nova thread</td><td></td></tr>
<tr><td><code>{X}(f⍤B)Y</code></td><td>rank: f sobre ou entre subarrays de rank B</td><td></td></tr>
<tr><td><code>(f⌺g)Y</code></td><td>Stencil: movimento de janela
No exemplo ao lado, cria-se uma janela 2x2 que passeia pela matriz somando-se os elementos de cada sub-matriz</td><td><code>   a←3 3⍴⍳12</code>
<code>   ({+/,⍵}⌺ 2 2)a</code>
<code>12 16</code>
<code>24 28</code></td></tr>
<tr><td><code>{X}(f⍣g)Y</code></td><td>power: itera <code>F</code> (ou <code>X∘f</code>) sobre Y até a condição YgfY (ou YgXfY) ser verdadeira</td><td></td></tr>
<tr><td><code>{X}(f⍣Js)Y</code></td><td>power: <code>f (ou X∘f)</code> sobre Y, Js vezes</td><td></td></tr>
</table>

Operador I-beam Existe um capítulo imenso de operações mistas (funções, operadores, funções de sistema, uma miscelânea) todos eles sob operação do operador I-beam (`⌶`).

Índice em tabela invertida

`r ← X (8⌶) Y`

Retorna os índices de Y em X, sendo que ambos são estruturas compatíveis. Veja no manual APL Language, páginas 198 e 199.

Execute

`r ← X (85⌶) Y`

Executa a expressão Y como a primitiva "execute" (`⍎`) mas maneja os resultados tímidos da expressão de maneira diferente. *X* pode ser 0 ou 1. Se for 1 e a expressão em Y retorna resultado explicito, ele vai para *r*. Se *Y* não retorna resultado ou retorna resultado tímido, a função sinaliza erro 85.

Reescreve áreas livres

`r ← 127 ⌶ Y`

Normalmente o workspace mantém dados prévios, mesmo após a eliminação de variáveis. Em aplicações sensíveis, a execução desta expressão garante que as áreas livres serão reescritas. *Y* é um array vazio, preferivelmente zilde (`⍬`). *r* é sempre 1.

Representação canônica

```
r ← 180 ⌶ Y
```

Funciona como ⎕CR, exceto que pode ser usada para obter a representação canônica de métodos em classes. É usada pelo comando]PROFILE.

Tipo do array

```
r ← 181 ⌶ Y
```

Funciona como o ⎕DR monádico e retorna o tipo de Y. Veja o manual APL Language, pág. 202.

Coloreando a síntaxe

```
r ← 200 ⌶ Y
```

Este sujeito retorna uma especificação estrutural de uma função. Y é um vetor de caracteres contendo a ⎕NR de uma função. Veja

```
 r←isnum x
 :If (83=⎕DR x) ◇ r←1
     →0
 :Else ◇
     r←0
 :EndIf
```

Agora o ⎕NR

```
      ⎕NR'isnum'
  r←isnum x   :If (83=⎕DR x) ◇ r←1       →0   :Else ◇       r←0   :EndIf
```

E finalmente o 200 ⌶ Y

```
      200⌶⎕NR'isnum'
 3 34 19 21 21 21 21 21 3 34  3 119 119 119 3 19 5 5 19 13 13 13 3
 34 19 3 25 3 34 19 5  3 3 3 3 3 19 5  3 123 123 123 123
 123 3 25  3 3 3 3 3 34 19 5  3 124 124 124 124 124 124
```

Aqui, neste exemplo, cada número vale:

21	constante de caracter
19	primitiva
3	espaço em branco
34	nome local
7	nome global
23	idioma
119	:If
...	...

Tokens da coloração de síntaxe

```
r ← 201 ⌶ Y
```

Obtém os valores completos da tabela acima. Y é zilde. r é uma matriz de 4 colunas, a saber: tipo do token, valor numérico dele, nome interno e a quarta coluna é usada para terminais tipo tty. Vale a pena olhar.

Compressão de vetor de inteiros short

```
r ← X (219 ⌶) Y
```

O vetor de inteiros deve ser formado por valores ¯128 a 127 (tipados como 83 pelo ⎕DR. (Veja mais à pag. 354 do APL Language). Em muitos casos, esta funcionalidade é usada junto com a 220 ⌶ (serializar / desserializar arrays).

X especifica a operação, a biblioteca e algum parâmetro opcional. Y contém os dados a serem operados.

Compressão Y deve ser um vetor de inteiros short. r é um vetor de 2 ítens. ambos são short ints. $r[1]$ descreve a compressão e $r[2]$ contém os dados resultados da aplicação da compressão sobre Y. Eis os valores de X

`X[1]`	`X[2]`	Biblioteca
1	não se aplica	LZ4
2	0..9	zlib
3	0..9	gzip

Se usado LZ4, X deve ser um escalar ou um vetor de 1 elemento. De outra forma, se X[2] é presente ele indica o nível de compressão. Números maiores indicam compressão maior, mas mais demorada.

Descompressão r é um vetor de inteiros short, contendo o resultado da descompressão aos dados Y.

Os valores de X são

`X[1]`	Biblioteca
`¯1`	LZ4
`¯2`	zlib
`¯3`	gzip

Vamos ver exemplos

```
      oba ← {⍵-256×⍵>127}
```

A função dinâmica `oba` acerta o vetor de inteiros para o intervalo `¯128..127`

```
      utf8←'UTF-8' ∘ ⎕UCS
      utf8 'Pedro'
80 101 100 114 111
       ⎕AV[81]  ⍝ nao esqueça que aqui ⎕IO=0
P
```

O operador utf8, converte números em caracteres UNICODE e vice-versa. Se quiser olhar a definição de `⎕UCS`, veja a pág. 624 do APL Language (vol 2).

```
      nome←'Curitiba Paraná Pedro ⌹'
      ⊣ v ← oba utf8 nome      ⍝ o ⊣ mostra o resultado
67 117 114 105 116 105 98 97 32 80 97 114 97 110 ¯61 ¯95 32 80 101 100 114
111 32 ¯30 ¯116 ¯71
      ⍴v
26
      ⍴nome
23
```

E agora, a mágica

```
       ⊣comp←1 (219⌶) v
 8 ¯55 1 0 0 0 0 26  ¯16 11 67 117 114 105 116 105 98 97 32 80 97 114 97
110 ¯61 ¯95 32 80 101 100 114 111 32 ¯30 ¯116 ¯71
      utf8 256|0 (219⌶) comp
Curitiba Paraná Pedro ⌹
       ⍴comp
2
      ⍴⊃comp[1]
8
      ⍴⊃comp[2]
28
      ⍴nome
23
      comp[1]
 8 ¯55 1 0 0 0 0 26
      comp[2]
 ¯16 11 67 117 114 105 116 105 98 97 32 80 97 114 97 110 ¯61 ¯95 32 80 101
100 114 111 32 ¯30 ¯116 ¯71
```

Serialização/desserialização de vetor

```
r ← X (220 ⌶) Y
```

usado em conjunto com a funcionalidade anterior. X deve ser 0 (desserializa) ou 1 (serializa). Veja pág. 207 de APL Language.

Controle do compilador

```
r ← {X} (400 ⌶) Y
```

Veja o *Compiler User Guide*. A compilação em Dyalog serve apenas para acelerar a execução de funções (e não para distribuir software como em outras encarnações de APL). Uma função em Dyalog APL gasta 2 fatias de tempo para ser executada:

- A interpretação do comando da função
- Sua aplicação sobre os dados

A compilação opera apenas sobre a primeira dessas parcelas. E, só tem sentido quando aplicada a escalares ou arrays pequenos. Quando um comando se aplica a um array enorme, a parcela da interpretação é muito pequena, e portanto otimizá-la não tem grande significado. O código compilado é traduzido para um byte code otimizado e guardado junto com a função original no workspace. Ele é manipulado junto quando a função é copiada entre workspaces. Veja como a coisa funciona

```
⎕FX 'r←funcao y' ... ⍝ define a funcao
funcao 100           ⍝ executa interpretadamente
2(400⌶)'funcao'      ⍝ compila a função
funcao 100           ⍝ executa a versão compilada
```

Para perguntar se a função foi compilada adequadamente faça

```
1(400⌶)'funcao'
```

Responde 1 se houve sucesso na compilação.

Para compilar uma função faça

```
2(400⌶)'funcao'
```

Esta função devolve uma matriz de diagnósticos. Se a matriz tiver 0 linhas, deu tudo certo. Senão cada linha descreve um problema havido na compilação. Os itens da linha: o erro APL; o número da linha; o número da coluna (presentemente é 0) e a mensagem de erro. Eis aqui um exemplo Seja a função

```
       ∇media[⎕]∇
[0]   r←media x;a
[1]   a←+/x
[2]   r←a÷ρ,x

        2(400⌶)'media'
       1(400⌶)'media'
1
       4(400⌶)'media'
Dump of bytecode for media:
  0000: 00000012 // version 18
  0001: 00000000 // localised system variables: none
  0002: 00000201 // 2 slots
  0003: 00000002 // 0 uslots
  0004: 00009BC5 cpy PFUNCTION, rawlst[4]
  0005: 00000F46 cpy slot[0], Rarg
  0006: 00000024 eval
  0007: 00002F11 tokoff 002F
  0008: 00000E45 mov Rarg, slot[0]
  0009: 00002E66 mov slot[1], Rslt
  000A: 00002124 eval 0x21 // ,
  000B: 00003711 tokoff 0037
  000C: 00006045 mov Rarg, Rslt
  000D: 00002024 eval 0x20 // ρ
  000E: 00003611 tokoff 0036
  000F: 00002E25 mov Larg, slot[1]
  0010: 00006045 mov Rarg, Rslt
  0011: 00000524 eval 0x05 // ÷
  0012: 00003511 tokoff 0035
  0013: 00000003 ret
```

Veja mais detalhes no APL Language Reference, vol-1, pág 208 e seguintes.

Controle de trap

```
r ← 600 ⌶ Y
```

Usada para temporariamente desabilitar os mecanismos de traping de erro usados por `:Trap` e por `⎕TRAP`. Isto pode ser importante em depuração de aplicações. O Y determina o que fazer: 0=habilita todos os traps; 1=desabilita todos os traps e 2=desabilita em funções suspensas.

Conversão de caixa

```
r ← {X} (819 ⌶) Y
```

Converte os caracters de Y para maiúscula ou minúscula. Esta função foi depreciada (deprecate) e substituída pela função de sistema `⎕C`. Se X é 0, converte para minúscula e se x é 1 converte para maíuscula.

Chamada monádica Identifica como a função corrente foi chamada.

```
r ← 900 ⌶ Y
```

R é booleano. Se a última função na pilha foi monádica, r é 1. Se não há função na pilha ou se ela não foi monádica r é 0.

Diretório temporário retorna o nome do diretório temporário para arquivos de usuário. Y é 0.

```
      739⌶0
C:/Users/Pedro/AppData/Local/Temp
```

Bibliotecas carregadas

```
r ← 950 ⌶ Y
```

reporta o nome da dll que esta correntemente carregada como resultado da execução de `⎕NA` (name association). Veja página 214 e depois páginas 436 do APL reference guide.

Número de threads Especifica o número de threads na execução paralela.

```
r ← 1111 ⌶ Y
```

Se Y vale `⍬` r contém o número de threads. De outra forma, Y indica o número desejado de threads. O limite é 64. Aqui em casa, eu fiz

```
      1111⌶⍬
12
```

Limite para execução em paralelo

```
r ← 1112 ⌶ Y
```

Y é um inteiro que estabelece o tamanho limite para arrays. Se uma função que permite processamento em paralelo é chamada e o número de elementos do array é maior do que este limite, o processamento ocorre em paralelo. Se o valor não é alterado seu *default* é 32.768. r é o valor prévio.

Atualização do carimbo de tempo

```
{R} ← X ( 1159 ⌶) Y
```

Y é um array de nomes de funções como o argumento direito de `⎕AT`. X é um array de atributos no mesmo formato da saída de `⎕AT`. r informa se a troca foi feita ou não: 0=não foi feita, o nome não é uma função ou ela está trancada (locked). 1=o carimbo de tempo e de usuário foi atualizado.

Formato de data-tempo

```
r ← X ( 1200 ⌶ ) Y
```

Y é um array de qualquer forma, onde cada elemento contém uma data Dyalog entre 1 de janeiro de 0001 e 28 de fevereiro de 4000 no calendário Proléptico Gregoriano. X é um escalar ou vetor de caracteres especificando o padrão pelo qual Y deve ser formatado. Tem páginas e páginas de explicação (veja pags. 217 a 225 do APL reference (vol 1)). Um exemplo

```
      t←1 ⎕DT ⊂ 2020 7 17 11 10 0
      t
44028.46528
      'Dddd DDoo Mmmm YYYY; hh:mm:ss' (1200⌶) t
 Friday 17th July 2020; 11:10:00
```

Array de hash Cria um array de hash, retorna uma cópia *unhashed* de um array ou relata o estado de hash de um array.

```
r ← {X} 1500 ⌶ T
```

Y pode ser qualquer array. Se X é omitido, r é uma cópia de Y invisivelmente marcada como hash. r funciona como Y em qualquer aspecto. A única diferença é que `⍳` diádica e relacionados funciona mais rápido em r do que em Y. Se X é 1, r informa: $r = 0$ Y não foi marcada para hashing; $r = 1$ Y foi marcada para hashing, mas a tabela hash ainda não foi criada. $r = 2$ informa que Y é uma tabela hash.

Estatísticas de gerenciamento de memória

```
r ← {X} ( 2000 ⌶ ) Y
```

Retorna informações sobre alocação de memória. Veja na pág. 228. Eis um exemplo

```
      2000⌶0    ⍝ pergunta o ⎕WA
208252184
      2000⌶1    ⍝ pergunta quanto já foi usado no ws
1463016
```

Especifica o espaço disponível

```
r ← 2002 ⌶ Y
```

Funciona como `⎕WA`, mas é o mescanismo para especificar a quantidade de memória *committed.* Veja a pág. 228 e 231.

Desabilitar gatilhos globais

```
r ← 2007 ⌶ Y
```

Temporariamente desabilita e re-habilita os gatilhos globais. Se $Y = 0$, desabilita e se $Y = 1$ habilita-os.

Demais I-beam codes No manual tem uma série deles, mas acho meio irrelevante aqui. A maioria é para código .NET, veja-os lá.

Funções do usuário Existem pelo menos 5 maneiras de criar uma função do usuário dentro de um workspace DAPL, a saber:

1. Estabelecendo `⎕FX` de uma matriz de caracteres que seja adequada para conter uma função, possivelmente vindo de uma chamada a `⎕CR`.

2. Usando o editor de caracteres. Este uso está bem descontinuado e só aparece aqui por compatibilidade com o passado, já que usa um protocolo caractere a caractere, herança dos antigos terminais de teletipo. Lembrando, durante muitos anos esta foi a única maneira de criar e alterar programas. Ainda hoje, às vezes é usada (talvez por saudosismo) para fazer uma pequena alteração em um comando. Entra-se (e saí-se) nesta modalidade teclando `∇`.

3. Método MINI: Define-se a função escrevendo as operações sem nenhum operando na ordem em que elas vão aparecer. Veja

   ```
         media2←(+/)÷⍴
         media2 1 2 3 4
   2.5
   ```

4. Método MIDI: Este método recebe o nome de **dfn**=função dinâmica dyalog. É uma expressão APL entre chaves podendo ou não ter nome e possivelmente usando `⍺` como argumento esquerdo e `⍵` como argumento direito. veja no primeiro caso sem nome e no segundo com nome (=media2):

   ```
         {(+/⍵)÷⍴⍵} 1 2 3 4
   2.5
         media2←{(+/⍵)÷⍴⍵}
         media2 1 2 3 4
   2.5
   ```

   Uma dfn pode ser definida:

   - para execução imediata: no meio da sessão escreve-se abre e fecha chave e segue o baile

- dentro de uma função ou operador
- dentro do operador `each` (¨)
- dentro de outra dfn, o que permite funções locais aninhadas.

5. Método MAXI: o jeito padrão por excelência. Chamando o editor de funções e definindo nele a função completa e por extenso. A chamada se dá por `)ed nome` ou clicando duas vezes no nome da função, ou ainda chamando `⎕ed 'nome'`. Veja o mesmo exemplo

```
    ∇ r←media x
[1] r←(+/x)÷⍴x ∇
```

Mais sobre dfn Uma função direta (dfn, pronunciada como *dee fun*) é uma maneira alternativa de definir uma função. A mesma idéia se aplica à definição de um operador, aqui chamado de dop (pronunciado como *dee op*). Foram inventados por John Scholes em 1996.

São funções escritas entre chaves ({ e }) e separadas por nova-linha ou eventualmente por `⋄` (diamante). Nesta especificação alfa (`⍺`), indica o operando esquerdo da dfn enquanto ômega (`⍵`), indica o operando direito da dfn. Auto-referências, necessárias para chamar a si próprio em casos de recursividade usam o símbolo del (`∇`) como o nome desta dfn. Veja-se um exemplo interessante de dfn

```
   PT ←  { ( + / ⍵ * 2 ) = 2 × ( ⌈ / ⍵ ) * 2 }
   PT  3  4  5
1
   x
 4   5   3
 3  11   6
 5  13  12
17  16   8
11  12   4
17  15   8
   PT  x
1  0  1  0  0  1
```

Esta dfn acima verifica se 3 números fazem parte de um trio pitagórico (lados de um triângulo retângulo). A lógica é meio óbvia, mas por seu sincretismo vale alguma explicação. O código começa elevando os 3 números ao quadrado e somando este resultado, isto tudo dentro do parêntesis. A seguir, a função determina o maior valor (supostamente a hipotenusa) e eleva-a ao quadrado. Como este valor foi igualmente somado ao primeiro termo, aqui ele é multiplicado por 2 para equilibrar. Havendo igualdade a trinca é pitagórica. Senão não. Como as 2 reduções não têm indicação de eixo, elas se aplicam ao último eixo. Isto permite chamar esta função com um array, como no segundo caso do exemplo.

Mais um exemplo, agora recursivo, do cálculo do fatorial

```
   fact← {0=⍵:1 ⋄ ⍵×∇ ⍵-1}
   fact 5
120
   fact¨ ⍳10     ⍝ fact é aplicado a cada elemento de 0 a 9
1 1 2 6 24 120 720 5040 40320 362880
```

Uma dfn é formada por um conjunto de expressões, expressões guardadas ou apenas guardas. Uma guarda é uma condição que retorna 0 ou 1. A guarda é avaliada e se o resultado for 1, a expressão associada é executada. A dfn termina após a primeira expressão não guardada que não seja um assinalamento (`←`) OU após a primeira expressão guardada que tenha respondido 1 OU quando não há mais expressões.

O resultado de uma dfn é o resultado da última expressão avaliada. Se a última expressão avaliada termina em um assinalamento, o resultado é tímido (*shy*) ou seja não é mostrado na sessão. (Mas vai para uma variável se isto estiver no código).

Todas as variáveis citadas na dfn são locais a ela, e não há risco de conflito de nomes com variáveis globais. Há uma atribuição especial `⍺←expressão` que só é executada se a dfn for chamada monadicamente. Se ela for chamada com 2 operandos, este comando não é executado. Este caso sinaliza o valor *default* para a variável esquerda.

Operador direto Usa mais ou menos as mesmas regras, mas agora o operando esquerdo é alfa-alfa (`⍺⍺`) enquanto o direito é ômega-ômega (`⍵⍵`). Já a auto-chamada na recursão é representada por del-del (`∇∇`). Isto tudo nos permite escrever o seguinte resumo

{`⍺` função `⍵`}	{`⍺⍺` operador `⍵⍵`}	:guarda
`⍺` argumento esquerdo	`⍺⍺` operando esquerdo	:: erro-guarda
`⍵` argumento direito	`⍵⍵` operando direito	`⍺←`arg esq default
`∇` auto-referencia	`∇∇` auto-referencia	`s←`resultado tímido

Guarda de erro O exemplo seguinte mostra como manipular os códigos de erro associados. Veja o exemplo

```
mais←{
  tx←'catch all' ⋄  0::tx
  tx←'domain'    ⋄ 11::tx
  tx←'length'    ⋄  5::tx
  ⍺+⍵
}
   2 mais 3              ⍝ sem erro
5
   2 3 4 5 mais 'treis'  ⍝ tamanhos dos argumentos não combinam
length
   2 3 4 5 mais 'tres'   ⍝ não se somam letras a números
domain
   2 3 mais 3 4⍴5        ⍝ não dá para somar vetor e matriz
catch all
```

Em APL o erro número 5 é "length error". O erro número 11 é "domain error"e o erro número 0 é qualquer erro entre 1 e 999.

O exemplo mostra como estabelecer o valor de um erro de guarda. O nome `tx` é local e apenas retêm o erro correto.

Funções trad Este é o nome de funções definidas as usual, usando um editor. Eu chamei-as há pouco de funções MAXI. As diferenças que interessam no caso:

- Uma dfn pode ser anônima, uma tradfn sempre tem nome.
- Uma dfn é nomeada pelo assinalamento. Uma tradfn têm seu nome estabelecido na primeira linha de sua definição.
- Os nomes em uma dfn são locais. Em uma tradfn são globais, a menos que tenham sido citados no header (primeira linha) da definição.
- Uma dfn pode chamar uma tradfn e vice-versa. Uma dfn pode ser definida em uma tradfn e vice-versa.

O Dyalog APL chama estas funções de funções procedurais ou funções tradicionais, ou abreviadamente trad-funs.

Editor O Dyalog tem um editor built-in usado para editar objetos. Sua chamada em

```
)ed coisa
```

edita a FUNÇÃO de nome `coisa`. Mas, o mesmo editor pode ser chamado para editar outras coisas, a depender de um caracter escrito entre `)ED` e o nome da coisa. Acompanhe na tabela

caracter	edita o que ?
nada	função
`∇`	função
`-`	matriz de texto
`→`	vetor de texto
`∊`	vetor de vetores de texto (1 sub-vetor por linha)

Funções do sistema Este capítulo é extenso, vamos abreviar.

função	significado	detalhes e exemplo
`⍞`	entrada e saída de caracteres	`      ⍞←'2+2' ⋄ ⍞←'=' ⋄ ⍞←4` `2+2=4`
`⎕`	entrada e saída do APL	ao contrário do APL2 serve para alfa.
`⎕A`	caracteres alfabéticos	`      ⎕A` `ABCDEFGHIJKLMNOPQRSTUVWXYZ`
`⎕AI`	informações de contabilização	[1]=identificação do usuário; [2]=tempo de computador em milésimos de segundo; [3]=tempo de conexão; [4]=tempo de digitação. Aqui em casa: `⎕AI` `0 7203 9682597 9378999`
`⎕AN`	nome do usuário	`      ⎕AN` `Pedro`
`⎕ARBIN`	Input arbitrário	pág. 298
`⎕ARBOUT`	Output arbitrário	pág. 300

`⎕AT`	atributos	Informa atributos do objeto (tipo de resultado e valência)
`⎕AV`	Vetor atômico	Depreciado e trocado por `⎕UCS`
`⎕AVU`	Vetor atômico Unicode	Vetor de inteiros com 256 elementos com os codepoints dos caracteres em `⎕AV`
`⎕BASE.Y`	Base Class	Informa a classe base do nome Y
`{X}⎕C Y`	Conversão de caso	Se $X = 1$ devolve Y em maiúsculo. Se $X = ^{-}1$ em minúsculo.
`{X}⎕CLASS Y`	Referências a classes e interfaces	Se monádico retorna a hierarquia de classes de Y. Se diádico, Y é uma classe e X é uma referência a uma interface
`⎕CLEAR`	Clear	Limpa o workspace e dá-lhe o nome de `CLEAR WS`

`{R}←⎕CMD Y`	Executa comando	`Z←⎕CMD'dir'`
		`↑Z'`
		...segue o dir...

`{R}←X ⎕CMD Y`	Inicia processador auxiliar	Sinônimo de `⎕SH`. X é o nome do processador auxiliar. Y é ignorado em windows.
`⎕CR Y`	Representação canônica	da função Y
`{R}←{X}⎕CS Y`	Change Space	Y é a referencia a um nomespace. X é o conjunto de objetos a exportar para o namespace.
`{R}←{X}⎕CSV Y`	Valores separados por vírgula	Importa e exporta valores separados por vírgula. Se monádico Y especifica a origem dos valores. Se diádico X contém os valores a converter a CSV. Y neste caso é o destino.
`⎕CT`	Tolerância de comparação	Default é 10^{-14}
`{R}←{X}⎕CY Y`	Copia de workspace	Y é o nome do workspace salvo. X é opcional e pode identificar nomes APL do workspace.
`⎕D`	Dígitos	vetor de caracteres: '0123456789'
`⎕DCT`	Tolerância de comparação decimal	Usado quando `⎕FR` é 1287. Nos demais casos usa-se `⎕CT`
`{R}←⎕DF Y`	Forma de display	Estabelece como mostrar um namespace, objeto GUI, instância ou classe
`⎕DIV`	Tratamento da divisão por zero	Se 0, a divisão por zero produz erro de domínio, exceto em `0÷0` que vale 1. Se igual a 1, a divisão por 0 retorna 0. O default é 0.
`{R}←⎕DL Y`	Atraso	Pausa de Y segundos.
`⎕DM`	Mensagem de diagnóstico	vetor de 3 elementos: a mensagem, a linha errada e a posição do caret
`⎕DMX`	Mensagem de diagnóstico extendido	Inclui informações de threads
`{R}←⎕DQ Y`	aguarda e processa eventos.	Uso em objetos GUI
`{X}⎕DR Y`	Representação do dado	Se monádico, retorna o tipo do argumento Y. Se diádico converte Y no tipo especuificado por X
`X ⎕DT Y`	Data e tempo	Valida ou converte data e tempo entre formatos
`{R}←{X} ⎕ED Y`	Edita objeto	Invoca o editor. Y é o objeto a editar e o opcional X é o que o editado deve virar.

`⎕EM Y`	Mensagem de evento	Converte um código em mensagem de erro
		`⎕EM 11`
		`DOMAIN ERROR`

`⎕EN`	Evento número	Retorna o número do último erro ou interrupção
`⎕EXCEPTION`	Exception	Retorna a última exceção de um objeto NET
`{R}←⎕EX Y`	Expunge	Elimina o objeto Y
`{R}←{X}⎕EXPORT Y`	Exportação de objeto	Exporta funções ou operadores
`{R}←X ⎕APPEND Y`	Component file append	Acrescenta X no arquivo de número (tie) Y
`⎕FAVAIL`	File system available	Se retornar 1 é porque o componente de File System está disponível. Se voltar 0, não está
`{X} ⎕FCHK Y`	File check & repair	valida e repara arquivos de componentes de número Y
`X ⎕FCOPY Y`	Cópia de arquivo	Cria uma cópia do arquivo de número Y com o nome X
`{R}←X ⎕CREATE Y`	Cria o arquivo de nome X com o número Y	
`{R}←⎕FDROP Y`	Elimina arquivo	de número Y

`{R}←X ⎕FERASE Y`	Elimina arquivo	de número Y
`⎕FHIST Y`	História do arquivo	Indica data e hora dos principais eventos do arquivo
`{R}←⎕FHOLD Y`	Hold de arquivo ou variável externa	Para processamento sincronizado
`{R}←{X}⎕FIX Y`	Fixa script	Estabelece namespaces, interfaces e funções do script Y
`⎕FLIB Y`	Biblioteca de arquivos componentes	No diretório Y, mostra os arquivos
`{X}⎕FMT Y`	Formatação	Transforma numérico em caracter. págs 388-394
`⎕FNAMES`	File names	Retorna os nomes dos arquivos abertos (tied)
`⎕FNUMS`	File numbers	Retorna os números dos arquivos abertos (tied)
`X ⎕FPROPS Y`	File properties	Retorna as propriedades X do arquivo número Y
`⎕FR`	Representação em ponto flutuante	O valor de `⎕FR` determina como são executadas as oprações de ponto flutuante
`⎕FRDAC Y`	Acesso de leitura a arquivo	Devolve a matriz de acesso do arquivo de número Y
`⎕FRDCI Y`	Informação de arquivo de leitura	Informa tamanho, último leitor e última leitura do arquivo
`⎕FREAD Y`	Leitura	Lê o registro $Y[2]$ do arquivo de número $Y[1]$
`X ⎕FRENAME Y`	Renomear	Renomeia o arquivo de número Y (tied) para o nome X
`{R}←X ⎕FREPLACE Y`	Substituição de registro	Troca o registro $Y[2]$ do arquivo de número $Y[1]$ para o conteúdo X
`{R}←{X} ⎕FRSIZE Y`	Troca de tamanho	Troca o tamanho do arquivo de número Y. X determina o tamanho máximo em bytes. O valor 0 indica o máximo possível. Esta operação também compacta o arquivo
`⎕FSIZE Y`	Dados do arquivo	Informa: o número do primeiro componente; o número (+1) do último, o tamanho atual e o limite do arquivo
`X ⎕FSTAC Y`	Estabelece Acesso	Usando uma matriz válida em X estabelece o acesso a Y
`{R}←X ⎕FSTIE Y`	Compartilhamento e abertura	Associa o arquivo de nome X ao número Y. Exemplo `'SALES' ⎕FSTIE 1`
`{R}←X ⎕FTIE Y`	Abertura exclusiva	Abre e associa o arquivo X ao número Y. Exemplo `'SALES' ⎕FTIE 1`
`{R}←⎕FUNTIE Y`	Fechamento	Fecha o arquivo. Se Y =`⍬`, fecha tudo
`{R}←⎕FX Y`	Estabelece (fixa)	o objeto Y
`⎕INSTANCES Y`	Instâncias	Retorna uma lista das instâncias da classe Y
`⎕IO`	Origem dos índices	Pode ser 0 ou 1
`{X}⎕JSON Y`	Dados JSON	Importa e exporta dados neste formato
`⎕LC`	Line count	Vetor de números informando o indicador de estado
`⎕LOAD Y`	Load de ws	Carrega o workspace salvo como Y
`{R}←{X}⎕LOCK Y`	Lock	Tranca a função ou operador. X indica em que condições...
`⎕LX`	Expressão latente	O que é executado automaticamente quando o workspace é carregado
`{X}⎕MAP Y`	Map File	Associa um arquivo mapeado com um array APL
`{R}←{X} ⎕MKDIR Y`	Make directory	Cria um diretório de nome Y de acordo com indicações de X
`⎕ML`	Migartion level	Determina o grau de migração entre este APL e o APL2 da IBM
`{R}←X ⎕MONITOR Y`	Monitoramento	X identifica as linhas da função/operador Y que devem ser monitoradas. A resposta é a acumulação de estatísticas de tempo para estas linhas. Se X =`⍬`, cancela tudo. Se monádico, a resposta é para toda a função. Pág. 435
`{R}←{X}⎕NA Y`	Associação de nome	Permite o APL acessar funções compiladas dentro de uma dll. Págs 436-462
`{R}←X ⎕NAPPEND Y`	Append em arquivo nativo	$Y[1]$ é um inteiro negativo do arquivo. X é o conteúdo e $Y[2]$ indica qual o tipo de conversão de X antes de gravar
`⎕NC Y`	Classificação de nome	Y é classificado: -1:inválido; 0:não usado; 1:label; 2:variável; 3:função; 4:operador; 8:evento; 9:objeto
`{R}←X ⎕NCOPY Y`	Copia de arquivo nativo	Copia o arquivo de nome Y para o local X
`{R}←X ⎕NCREATE Y`	Cria arquivo nativo	Y é o número negativo pelo qual será conhecido. Se for 0, o sistema escolhe e devolve em R. X é o nome

{R}←{X} ⎕NDELETE Y	Elimina arquivo nativo	Elimina o arquivo Y
{R}←X ⎕NERASE Y	Apaga arquivo nativo	Y é o número negativo e X o nome igual a quando ele foi criado/aberto
⎕NEW Y	Cria instância	de classe, diálogo com objeto GUI ou tipo NET especificado por Y
⎕NEXISTS Y	Existe ?	Responde se o arquivo de nome Y existe ou não
{X} ⎕NGET Y	Lê arquivo de texto	X pode ser 'ASCII', 'UTF-8', entre outros e Y tem o nome e o flag. Se 0, R é um vetor de caracteres. Se é 1, R é um nested array. Veja o exemplo `aux ← ⎕NGET (arq) 1` `r←↑⊃aux[1]`
{X} ⎕NINFO Y	Informação de arquivo nativo	Y pode ser número ou nome
{X} ⎕NL Y	Name List	Informa quais os nomes do tipo Y possivelmente na ordem X
{R}←X ⎕NLOCK Y	Lock de arquivo	controla a atualização de um intervalo de bytes no arquivo. X diz o que fazer no arquivo $Y[1]$ a partir do byte $Y[2]$ pelo tamanho $Y[3]$
X ⎕NMOVE Y	Move arquivo	Similar a `⎕NCOPY` mas apaga na origem
⎕NNAMES	Nomes de arquivos nativos	Retorna nomes de arquivos nativos
⎕NNUMS	Números de arquivos nativos	Retorna números de arquivos nativos
{X} ⎕NPARTS Y	Nomes parciais	Y é um ou mais nomes de acordo com o sistema operacional. X indica se os nomes devem ser normalizados. $R[1]$ é o path; $R[2]$ é o nome base e $R[3]$ é a extensão
{R}←X ⎕NPUT Y	Escreva em arquivo de texto	$Y[1]$ é o nome do arquivo. $Y[2]$ é o flag: 0 se o arquivo já existe dá erro. 1 se o arquivo já existe é sobreescrito e 2 se o arquivo já existe ele é apendado. Exemplo `LF←⎕UCS 10` `(,a,LF) ⎕NPUT (arq) 1`
{R}←{X} ⎕NQ Y	Desenfileira evento	
⎕NR Y	Representação aninhada	Mostra a representação nested da função ou operador
⎕NREAD Y	Leitura arquivo nativo	$Y[1]$=número negativo do aqruivo; $Y[2]$=código de conversão; $Y[3]$=contador e $Y[4]$=início. Se omitido ou -1 o dado é lido da posição corrente
X ⎕NRENAME Y	Renomeia	Troca o nome do arquivo número negativo Y pelo nome X
{R}←X ⎕NREPLACE Y	Troca	O dado X é posto em Y, onde [1]=número negativo do arquivo; $Y[2]$=start byte. -1 aqui indica posição atual e $Y[3]$=conversão
{R}←X ⎕NRSIZE Y	Resize	Troca o tamanho do arquivo número Y para o tamanho X
{R}←{X}⎕NS Y	Namespace	
⎕NSI	Namespace indicator	
⎕NSIZE Y	Tamanho	Informa o tamanho do arquivo de número negativo Y
X ⎕NTIE Y	Abertura de arquivo nativo	X é o nome do arquivo. $Y[1]$ é o número negativo que ele deverá ter. $Y[2]$ é opcional e indica 0=leitura, 1=gravação e 2=ambos
⎕NULL	Nulo	Valor nulo
{R}←⎕NUNTIE Y	Fechamento	Fecha o arquivo negativo Y
{R}←{X} ⎕NXLATE Y	Translate de arquivo nativo	
⎕OFFDesliga	a sessão APL	
⎕OR Y	representação do objeto	Y
⎕PATH	Caminho	Mais ou menos análogo à variável PATH do windows
{X}⎕PFKEY Y	Tecla de função	Exemplo:`(')RESET',⊂'ER') ⎕PFKEY 2`
⎕PP	Precisão de impressão	Informa o número de decimais na saída, com 1..34
{R}←{X} ⎕PROFILE Y	Profile aplication	Págs 537-543
⎕PW	Print width	Tamanho da linha de saída
{X}(A ⎕R B) Y	Replace	Y é a entrada. Troca A por B. Págs 548-568
{X}(A ⎕S B) Y	Search	Págs. 549-568
⎕REFS Y	Referência cruzada	Y é uma função ou operador e a resposta é uma lista de nomes
⎕RL Y	random link	Valor inicial da randomização
⎕RSI	Indicador de espaço	
⎕RTL	Limite de timeout	Segundos de `⍞`, `⎕ARBIN` e `⎕SR`
{R}←{X}⎕SAVE Y	save workspace	Y é o nome do ws a salvar. Se X é 0, o)SI não é salvo, e se é 1 sim

⎕SD	Screen dimensions	2 inteiros com as linhas e colunas da tela
⎕SE	Namespace da sessão	
{R}←⎕SHADOW Y	Nome sombra	
⎕SI	State indicator	indicador de estado
{R}←{X} ⎕SIGNAL Y	Sinal de evento	
⎕SIZE Y	Tamanho de objeto	
⎕SM	Screen map	
{X}⎕SR Y	Screen read	
⎕SRC Y	Source	
⎕STACK	Pilha de SI	
⎕STATE Y	Estado de um objeto	
{X} ⎕STOP Y	Parada	X são as linhas de Y onde a execução deve parar. Se monádico estou perguntando em quais linhas ele para
X ⎕SVC Y	Set Access control	
⎕SVC Y	Query access control	
{X} ⎕SVO Y	Shared variable offer	Se monádico, estou perguntando o grau de acoplamento
⎕SVQ Y	Shared variable query	
⎕SVR Y	Shared variable retract offer	
⎕SVS Y	shared variable state	
⎕TC	Terminal control	depreciado. Substituir por $TC[1] = UCS[8]$, $TC[2] = UCS[10]$ e $TC[3] = UCS[13]$
⎕TCNUMS Y	thread child numbers	
{R}←{X}⎕TGET Y	Get tokens	
⎕THIS	This space	
⎕TID	Current thread identity	
{R}←{X} ⎕TKILL Y	kill thread	
⎕TNAME	Current thread name	
⎕TNUMS	Thread numbers	
⎕TPOOL	Token pool	
{R}←{X}⎕TPUT Y	Put tokens	
{R}←{X} ⎕TRACE Y	Estabelecer trace	Mostre as linhas X da função Y. Se monádico estou perguntando
⎕TRAP	Trap evento	
⎕TREQ Y	Token request	
⎕TS	Time stamp	ano, mês, dia, hora, minuto, segundo e milissegundo
⎕TSINC Y	Espera para a thread terminar	
{X}⎕UCS Y	Conversão unicode	Converte caracteres Unicode em inteiros e vioce-versa. Usada monadicamente converte caracteres Unicode em codepoints e vice versa. Se X é UTF-8 a tradução é entre caractres unicode e os bytes do caso
⎕VR Y	Representação em vetor	da função ou operador Y
{X}⎕VFI Y	verificação e fix da entrada	Converte e fixa números
⎕WA	Tamanho do workspace	Padrão aqui é 200MB
{R}←{X}⎕WC Y	Janela de criação de objeto	Para uso em GUI
{X}⎕WG Y	Obtenção de propriedades da janela	
{X}⎕WN Y	Nomes das janelas filho	
{R}←{X}⎕WS Y	estabelecimento de propriedades da janela	
⎕WSID	Nome do workspace	
⎕WX	Exposição de janela	
{X} ⎕XML Y	Converte string XML em array APL	
⎕XSI	Indicador de estado extendido	

`{X}⎕XT Y`	Seta variável externa	Se monádico estou perguntando

SALT Introduzido a partir da versão 11 do Dyalog, o SALT (Simple APL Libray Toolkit) permite que coisas (funções, operadores, variáveis... APL) sejam guardadas na forma de arquivos planos Unicode e possam ser carregados e usados em uma sessão APL. O que tecnicamente é chamado na indústria de *script*. As vantagens deste procedimento são evidentes

- Sistemas APL podem ser desenvolvidos usando qualquer editor.
- Podem ser armazenados em bibliotecas como arquivos de texto e podem ser lidos e mantidos de maneira independente.
- Podem ser enviados e recebidos por e-mail.
- Podem ser publicados e descarregados em páginas web.
- São carregados no APL quando necessários.
- Podem ser centralizados e geridos por um bibliotecário.
- Pode-se assinalar uma versão a cada módulo, seguindo-se uma gerência de versão adequada.

Vamos por partes (Como disse o Jack, aquele estripador)

Criando o arquivo de script Usando um editor, que obviamente deve ter suporte a Unicode, deve-se criar um arquivo cuja extensão deve ser `.dyalog`. É possivel usar outra extensão, mas daí ela precisará ser explicitamente declarada no `load`. Mas, não é uma boa idéia.

Este arquivo precisa iniciar por uma de 3 cláusulas, a saber `:Namespace`, `:Class` ou `:Interface` e deve encerrar por `:EndNamespace`, `:EndClass` ou `:EndInterface` respectivamente.

Logo após esta cláusula vem o nome interno do arquivo que será utilizado mais tarde como qualificador das coisas aqui definidas. Há também o nome externo, que será aquele do sistema operacional (antes da extensão `.dyalog` acima citada), Nada impede que ambos nomes sejam o mesmo. Vamos fazer isto aqui, chamando o arquivo de `trevas.dyalog`.

Assim, este arquivo é

```
:Namespace trevas
∇ r←media v;s
s←+/,v
r←s÷⍴,v
∇
primos←{1↓(~v∊1 1↓v∘.×v)/v←⍳⍵}
hoje ← 22 7 2020
:EndNamespace
```

e ele será salvo como `trevas.dyalog`. Note que neste arquivo definiram-se 1 função direta de nome `primos`, uma função tradicional de nome `media` e uma variável de nome `hoje`. Se quiser inserir os números de linha na função tradicional pode fazê-lo, mas eles serão descartados mais a frente.

Carga Posterior Tudo começa fazendo um

```
      )clear
clear ws
```

Para termos certeza de que nenhum fiapo solto vai ficar... Agora, cria-se uma variável qualquer contendo o caminho até o arquivo SALT, como em

```
Path←'c:\p\dapl2\'
```

e finalmente, usando a função de sistema `⎕SE` que atribui um nome ao namespace da sessão atual.

```
      ⎕←⎕SE.SALT.Load Path,'trevas'
#.trevas
```

As referências futuras ao conteúdo do arquivo SALT são feitas apendando no início o nome `trevas` como em

```
       trevas.primos 100
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

ou calculando alguma média

```
      trevas.media 1 2 3 4 5 6
3.5
```

Ou simplesmente perguntando que dia é hoje

```
      trevas.hoje
22 7 2020
```

É meio óbvio, que havendo um problema (erro) qualquer no código, volta-se ao editor unicode, altera-se o arquivo, salva-se o mesmo, recarrega-se o arquivo SALT e segue o baile. Se você clicar 2 vezes no nome da função que deu erro, o arquivo original será reaberto e se você tiver autoridade (questão do sistema operacional) poderá atualizar e salvar o arquivo.

A qualificação dos nomes no namespace é adequada para impedir conflitos neste workspace. Assim, nada impede (embora não seja uma boa idéia) criar uma função local com o mesmo nome como em

```
      ∇primos
[1]  33,55,66∇
```

E, a seguir

```
      primos
33 55 66
      trevas.primos 10
2 3 5 7
```

Comandos de usuário É uma facilidade (junto com a idéia de SALT = arquivos plenos Unicode) de desenvolver comandos que funcionam como os comandos `)xxxx` mas que são programados diretamente pelo usuário. A diferença entre uns e outros é que o caracter inicial é um "fecha colchetes" e portanto tais comandos têm a cara de `]xxxx`. Não podem ser emitidos dentro de funções do usuário e o Dyalog apresenta uma vigorosa lista de comandos do usuário (veja no manual).

Estruturas de Controle Uma função do usuário (programa) APL executa do primeiro ao último comando. Eventuais saltos podem ser implementados usando o formato `→n` onde n é o número da linha para onde se vai.

Eventuais melhoramentos deste desvio são:

- O desvio relativo (ao contrário do anterior – que é absoluto). Agora o desvio surge como `→label` e em algum ponto do código deve-se iniciar uma linha com a constante `label:`. Agora o desvio vem para cá. A vantagem é que ao renumerar o código – coisa que o APL faz de maneira automática cada vez que o código é alterado, o desvio absoluto corre o enorme risco de se perder, enquanto o desvio relativo segue íntegro.
- O desvio condicional. Agora o desvio só acontecerá se alguma condição for verdadeira.
  O formato é `→(condição)/label:`. Se a condição retornar 1, haverá desvio. Senão, o processamento segue a gravidade.

O APL original tinha apenas isto em termos de desvios. Era pouco, mas o APL se justificava dizendo que características da linguagem minimizavam esta necessidade. Um exemplo: o que em C++ se escreve

```
   if (0==x%2) {
      z='par'; }
   else {
      z='impar'; }
```

Em APL se escreve

```
z ← ((2⍴2|x),1 1 1)/'impar'
```

Acho até que era um cavalo de batalha dos APListas, dizendo mais ou menos assim, *olhe como somos diferentes*.

Mas, com exceção da IBM, toda a comunidade APL se rendeu ao fato de que não dá para ter uma linguagem de programação decente sem as ferramentas da programação estruturada. Por essa razão, quase todos implementaram tais ferramentas. Aqui vai-se ver como o Dyalog faz isso.

If No formato

```
:If <condição>
   comando1
   comando2
   ...
[:Else
   comando11
   ... ]
:EndIf
```

Observações: As palavras `:If`, `:Else` e `:EndIf` são escritas com essa configuração de caixa alta/baixa. Mas, se você escrever diferente, o editor corrige para você. :Else é opcional e se ausente implicará que nada é executado se a condição original for falsa.

Variações Há algumas, a saber:

:ElseIf Para simplificar longos textos de condicionais compostas. Acompanhe o exemplo

```
:If a>5 ⋄ ...a maior do que cinco
:ElseIf a>3 ⋄ ... a maior do que 3 e menor ou igual a 5
:ElseIf a>1 ⋄ ... a maior do que 1 e menor ou igual a 3
:Else ⋄ ... a menor ou igual a 1
:EndIf
```

:AndIf Permite incluir cláusulas AND à condicional original, veja

```
:If a>5 ⋄ ... a maior que 5
:AndIf a<10 ⋄ ... a maior que 5 E menor que 10
...
```

:OrIf Permite incluir cláusulas OR à condicional original

Eu particularmente prefiro deixar de usar :AndIf e :OrIf e usar condições compostas com as cláusulas and ($\wedge$) e or ($\vee$). Ao agir assim mantem-se a compatibilidade com outros ambientes e segue o baile (Teoria do homem comum).

While No formato

```
:While <condição1>
   comando1
   comando2
   ...
:EndWhile [ou :Until <condição2>]
```

A regra é a de sempre. A condição1 é avaliada e se verdadeira, o bloco subordinado é executado. No caso do bloco se encerrar por EndWhile, ocorre um desvio incondicional para p início do bloco, quando então a condição1 volta a ser reavaliada.

No caso do bloco terminar por um Until, ocorre uma segunda condição. Se esta segunda condição for verdadeira, há uma saída do bloco e se ela for falsa, retorna-se ao início do bloco. Veja um exemplo

```
∇  teste2 x
 :While x<100
     x←x+2
     x
 :Until x>50
 'acabou' ∇
```

Neste caso, se x começar com 1 são impressos 3, 5, 7, ... 49, 51 e fim de processamento.

Repeat No formato

```
:Repeat
   comando1
   comando2
   ...
   :Until condição
```

Haverá a repetição do bloco, pelo menos 1 vez e até que a condição seja verdadeira. Acompanhe

```
 teste3 x
 :Repeat
     x←x+2
     x
 :Until x>50
 'acabou'
```

Neste caso, se chamado com $x = 30$ haverá a impressão de 32, 34, 36, ..., 50, 52 e acabou o processamento. Note que :EndRepeat pode ser abreviado apenas com :End.

For O formato agora é

```
:For <variável> :In <valores>
   comando1
   comando2
   ...
:EndFor
```

Quando a cláusula :For é encontrada, a expressão ao lado de :In é avaliada e o resultado armazenado. A variável recebe o primeiro valor do resultado armazenado e o bloco entre :For e :EndFor é executado. Quando :EndFor for achado, o segundo valor armazenado é atribuído à variável e segue o baile. O bloco termina quando não houver mais valores a assinalar. Veja

```
 teste4
 :For a :In ⍳10
     a
 :EndFor
 'acabou'
```

Neste caso, haverá a impressão de 1,2,3,..., 10.

Select Para múltiplos valores, veja

```
:Select <variável>
:Case <valor1>
   comando1
:Case <valor2>
   comando2
:Else
   comando11
:EndSelect
```

With Usado para simplificar referências a um objeto ou a um namespace. Termina por :EndWith.

Hold Usado para sincronizar o acesso a blocos de dados ou recursos compartilhados entre várias threads.

Trap Mecanismo para tratamento de erros a ser usado em conjunto com a função `⎕TRAP`.

GoTo Implementa uma versão alternativa a `→n` ou `→label`.

Return Encerra o processamento de uma função de maneira incondicional. Equivale a `→0`.

Leave Usado para encerrar a execução de um bloco de comandos :For, :Repeat ou :While de maneira incondicional.

Continue Usado para desprezar o ciclo atual e iniciar uma nova interação do bloco onde ele é emitido. (:For, :Repeat e :While).

Section Usado para subdividir funções e scripts (classes, namespaces). Termina com :EndSecton. Este cara não altera a execução de nada.

Disposable Usado para eliminar objetos da interface com .NET.

Graphical User Interface - GUI Aqui estuda-se como construir aplicações com a cara do Windows. É tarefa complexa, pois muda-se a maneira de construir software. Vale um curso separado.

2.5 www.mathway.com

Este é um site capaz de resolver expressões algébricas de modo online, sem precisar instalar nada. Vale uma visita, e sobretudo o estudo das centenas de exemplos de uso.

2.6 Tudo junto

$16^{-\frac{3}{4}}$.

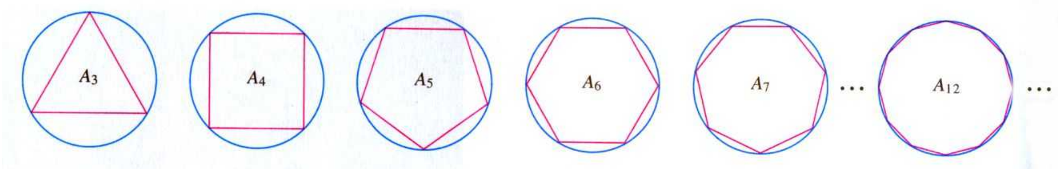
```
Em Python:
>>> import sympy as sp
>>> x,expr=sp.symbols('x,expr')
>>> expr=x**(-3/4)
>>> expr.subs(x,16)
0.125000000000000

Muito mais simples
>>> 16**(-3/4)
0.125

Em Maple:
> evalf(16**(-3/4));
.1250000000

No Mathway (https://www.mathway.com/Algebra) deu a mesma coisa...
```

Capítulo 3

Cálculo

3.1 Problema da área

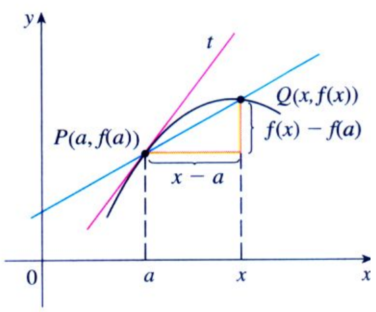
Há 2500 anos, na Grécia antiga, usou-se um método para calcular áreas chamado **Método da Exaustão**. Para um polígono qualquer, só havia que cortá-lo em triângulos, já que a área de um triângulo era conhecida: $S = \frac{1}{2}.b.h$ Já para encontrar a área de um círculo, a coisa era mais complicada. O problema foi resolvido inscrevendo (por dentro) e/ou circunscrevendo (por fora) polígonos regulares ao círculo e depois aumentando o número de lados dos polígonos como se pode ver em



Dado o número n de lados do polígono e chamando A_n a área de cada polígono, é fácil ver que a área do circulo S pode ser escrita como $S = \lim\limits_{n\to\infty} A_n$. Registre-se que os gregos não usaram limites, mas o raciocínio é o mesmo. O cálculo de áreas (e volumes) é central no Cálculo Integral.

3.2 Problema da tangente

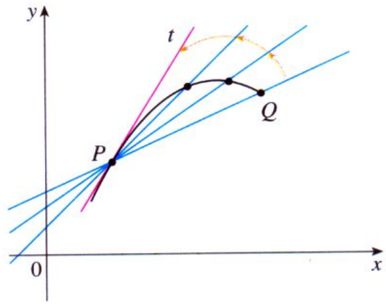
Achar a reta tangente t à função $y = f(x)$ em um ponto P ($P \in f(x) \wedge P \in t$). Como P é ponto de t, pode-se encontrar a equação de t se se conhecer a inclinação m de t. Mas, para conhecer a inclinação de t precisam-se 2 pontos e só se conhece P. Para resolver, vamos aproximar m, tomando um ponto Q próximo a P (na curva $y = f(x)$) e calculando a inclinação m_{PQ}, veja-se em



Desta figura tem-se

$$m_{PQ} = \frac{f(x) - f(a)}{x - a}$$

Agora deve-se imaginar o ponto Q movendo-se na curva em direção a P



A reta secante (2 pontos) vai-se aproximando da reta tangente (1 ponto). Com isso, a inclinação m_{PQ} vai-se tornando a inclinação M no limite. Isto é denotado por

$$m = \lim_{Q \to P} m_{PQ}$$

e pode-se reescrever a equação acima como

$$m = \lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

O problema da tangente deu origem ao cálculo diferencial que só foi inventado mais de 2000 anos após o cálculo integral, entre outros por Fermat, Newton e Leibniz entre outros.

Estes dois problemas (área e tangente) embora pareçam completamente distintos de alguma maneira são inversos, têm estreita conexão.

3.3 Velocidade

Supondo olhar o velocímetro de um carro e anotar o resultado, (tempo em segundos e distância em metros):

t=tempo	0	2	4	6	8	10
d=distância	0	2	10	25	43	78

Para medir a velocidade entre $t = 4$ e $t = 8$ fazemos $velocidade = \frac{distancia}{tempo}$ e temos $v = \frac{43-10}{8-4} = 8.25$ m/seg.

Se quisermos a velocidade entre $t = 4$ e $t = 6$, temos $v = \frac{25-10}{6-4} = 7.5$ m/seg.

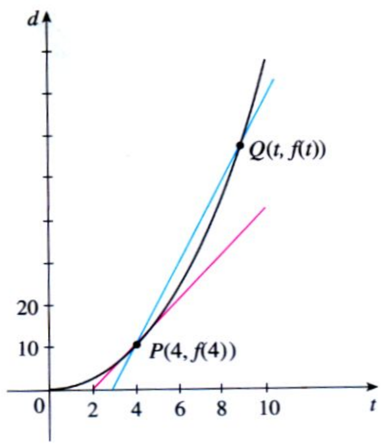
Intuitivamente, a velocidade no instante $t = 4$ não pode ser muito diferente da velocidade média durante um pequeno intervalo que começa em $t = 4$. Vamos agora medir a distância em intervalos de 0.3 segundos, obtendo

t=tempo	4	4.2	4.4	4.6	4.8	5
d=distância	10	11.2	12.16	13.45	14.96	16.80

Agora pode-se calcular qualquer velocidade nos intervalos citados como em

t=tempo	[4,6]	[4,5]	[4,4.8]	[4,4.6]	[4,4.4]	[4,4.2]
v=velocidade (m/s)	7.5	6.8	6.2	5.75	5.4	5.1

As velocidades médias em intervalos cada vez menores parecem ficar cada vez mais próximas a 5, assim espera-se que em $t = 4$ a velocidade seja $v = 5m/s$. Na figura a seguir vê-se a representação da distância em função do tempo



Se escrevermos $d = f(t)$ então $f(t)$ é o número de metros percorridos após t segundos. A velocidade média no intervalo de tempo [4,t] é

$$velocidade\ m\acute{e}dia = \frac{dist\hat{a}ncia\ percorrida}{tempo\ decorrido} = \frac{f(t) - f(4)}{t - 4}$$

que é a mesma coisa que a inclinação da reta secante PQ da figura anterior. A velocidade v quando $t = 4$ é o limite da velocidade média quando t se aproxima de 4, ou seja

$$v = \lim_{t \to 4} \frac{f(t) - f(4)}{t - 4}$$

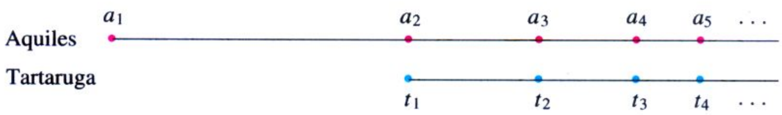
e comparando-se vê-se que isso é igual à inclinação da reta tangente em P.

Capítulo 4

Limites

4.1 Limite de uma sequência

500 anos antes de Cristo, o grego Zenão propôs quatro problemas, conhecidos como os *Paradoxos de Zenão.* O segundo deles fala sobre o campeão Aquiles e uma tartaruga disputando uma corrida. Nela, a tartaruga por razões óbvias ganha uma vantagem inicial. Zenão argumentava dizendo que a tartaruga nunca seria ultrapassada.



Se Aquiles começa-se em a_1 e a tartaruga em t_1, quando Aquiles chegasse a $a_2 = t_1$ a tartaruga estaria em t_2. No momento em que Aquiles atingisse $a_3 = t_2$ a tartaruga estaria em t_3 e se este processo ocorresse indefinidamente à frente, Aquiles nunca ultrapassaria a tartaruga, mas isso desafia o senso comum.

Uma maneira de estudar este paradoxo é usar a idéia de **sequência**. As posições sucessivas de Aquiles e da tartaruga são respectivamente $(a_1, a_2, a_3, ...)$ e $(t_1, t_2, t_3, ...)$ conhecidas como sequências.

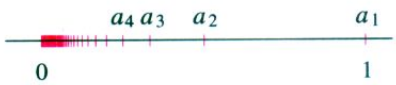
Por exemplo a sequência

$$\{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, ...\}$$

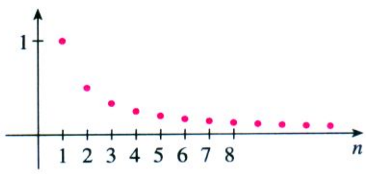
pode ser descrita pela fórmula do termo genérico

$$a_n = \frac{1}{n}$$

Para visualizar a sequência, podemos marcá-la numa reta real como em



ou desenhar seu grafico como em



Observe que em ambas as figuras os termos da sequência $a_n = 1/n$ tornam-se cada vez mais próximos a zero à medida que n cresce. Pode-se obter termos tão pequenos quanto se deseje, bastando para isso tomar n tão grande quanto se queira. Diz-se então que o limite da sequência é 0 e indica-se isso por

$$\lim_{n\to\infty} \frac{1}{n} = 0$$

O conceito de limite de uma sequência ocorre sempre que se usa uma representação decimal de um número real. Por exemplo

$$
\begin{aligned}
a_1 &= 3.1 \\
a_2 &= 3.14 \\
a_3 &= 3.141 \\
a_4 &= 3.1415 \\
. & \\
. & \\
. &
\end{aligned}
$$

$$\lim_{n\to\infty} a_n = \pi$$

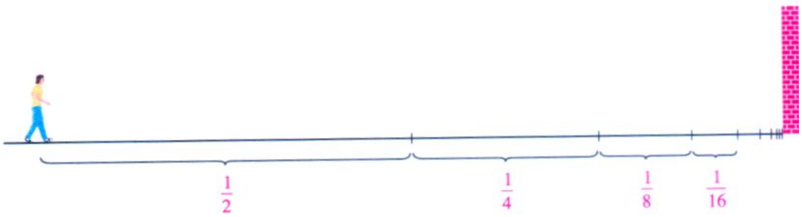
Voltando ao paradoxo de Zenão: as posições sucessivas de Aquiles e da tartaruga formam as sequência $\{a_n\}$ e $\{t_n\}$ onde $a_n < t_n, \forall n$ Pode-se mostrar que ambas as seuências têm o mesmo limite

$$\lim_{n\to\infty} a_n = p = \lim_{n\to\infty} t_n$$

É precisamente no ponto p que Aquiles ultrapassa a tartaruga.

4.2 Soma de uma série

Outro paradoxo de Zenão diz que é impossível uma pessoa no meio de uma sala chegar até uma parede. Para fazer isso deveria percorrer a metade da distância, depois a metade da distância restante, depois novamente a metade e assim por diante num processo interminável.



Como se sabe que a pessoa chega na parede a sugestão é que a distância possa ser expressa como a soma de infinitas distâncias cada vez mais pequenas. Zenão argumentava que não fazia sentido somar um número infinito de números. Porém nós fazemos isso, por exemplo ao escrever $0.\overline{3} = 0.33333...$ este número significa

$$\frac{3}{10} + \frac{3}{100} + \frac{3}{1000} + \frac{3}{10000} + ... = \frac{1}{3}$$

Portanto algumas somas infinitas têm um significado.

4.3 Limites

O cálculo é o ramo da matemática que trata de limites.

Suponha que $f(x)$ seja definido quando está próximo ao número a. Então escrevemos

$$\lim_{x\to a} f(x) = L$$

e dizemos "o limite de $f(x)$ quando x tende a a é igual a L" se pudermos tornar os valores de $f(x)$ arbitrariamente próximos a L tornando x suficientemente próximo de a por ambos os lados de a, mas não igual a a. Note que nunca consideramos $x = a$. $f(x)$ não precisa nem estar definida em $x = a$. Só nos interessa o que acontece *próximo* a a.

Seja o exemplo

$$\lim_{x\to 1} \frac{x-1}{x^2-1}$$

Note que a função não está definida para $x = 1$. A seguir, duas tabelas que mostram o que acontece quando a função se aproxima do ponto $x = 1$ pela direita e pela esquerda

```
      x←0.5 0.9 0.99 0.999 0.9999
      y←1.5 1.1 1.01 1.001 1.0001
      ∇r←lim x
[1]   r←(x-1)÷(x*2)-1∇
      10 7⍕5 2⍴ ⍉2 5⍴x,lim x
  .5000000  .6666667
  .9000000  .5263158
  .9900000  .5025126
```

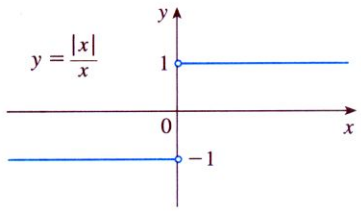
```
 .9990000  .5002501
 .9999000  .5000250
     10 7⍕5 2⍴⍉2 5⍴y,lim y
1.5000000  .4000000
1.1000000  .4761905
1.0100000  .4975124
1.0010000  .4997501
1.0001000  .4999750
```

Daqui, o limite procurado claramente é 0.5

4.4 Limites Laterais

Define-se o limite a esquerda de $f(x)$ e escreve-se $\lim_{x\to a-} f(x) = L$ quando x tende a a pela esquerda. Define-se o limite a direita de $f(x)$ e escreve-se $\lim_{x\to a+} f(x) = L$ quando x tende a a pela direita. Finalmente, considerando estes limites laterais, pode-se afirmar que o $\lim_{x\to a} f(x) = L$ se e somente se $\lim_{x\to a-} f(x) = L$ e $\lim_{x\to a+} f(x) = L$ isto é, ambos os limites à esquerda e à direita convergem para o mesmo valor.

Seja um exemplo: demonstrar que $\lim_{x\to 0} \frac{|x|}{x}$ não existe. Olhando o gráfico da função tem-se



e resolve-se
$\lim_{x\to 0+} \frac{|x|}{x} = \lim_{x\to 0+} \frac{x}{x} = \lim_{x\to 0+} 1 = 1$ e
$\lim_{x\to 0-} \frac{|x|}{x} = \lim_{x\to 0-} \frac{-x}{x} = \lim_{x\to 0+} (-1) = -1$ Como os valores acima são diferentes, da definição, o limite não existe.

4.5 Definição precisa

No século 18, depois da invenção do cálculo (no 17), houve um certo clima de *barata voa*. Os irmãos Bernoulli e Euler sairam explorando o cálculo sem ser muito rigorosos. O século 19, ao contrário assistiu um retorno ao assunto com definições e demonstrações rigorosas. Agora, a definição precisa do limite, devida a Weierstrass, depois do trabalho de Cauchy é

Seja f uma função definida em algum intervalo aberto que contenha o número a exceto possivelmente no próprio a. Dizemos que o limite de $f(x)$ quando x tende a a é L e escrevemos

$$\lim_{x\to a} f(x) = L$$

se para todo número $\varepsilon > 0$ houver um número $\delta > 0$ tal que se $0 < |x - a| < \delta$ então $|f(x) - L| < \varepsilon$.

Já a definição de um limite infinito, após este rigor é

Seja uma função f definida em algum intervalo aberto que contenha a, exceto possivelmente o próprio a. Então

$$\lim_{x\to a} f(x) = \infty$$

significa que que para todo número positivo M há um número positivo δ tal que se $0 < |x - a| < \delta$ então $f(x) > M$, com uma fácil generalização para $-\infty$ (pág. 107 de [Ste14]).

4.6 Continuidade

Uma função f é contínua em um número a se e somente se $\lim_{x \to a} f(x) = f(a)$. Esta definição requer 3 coisas

1. $f(a)$ está definida, ou seja $a \in$ domínio de f.
2. $\lim_{x \to a} f(x)$ existe
3. $\lim_{x \to a} f(x) = f(a)$

Esta definição diz que uma pequena mudança em x produz somente uma pequena modificação em $f(x)$. Geometricamente, uma função contínua pode ser pensada como tendo um gráfico que não se quebra. O desenho pode ser feito sem remover a caneta do papel.

A seguir um teorema: Qualquer polinômio é contínuo em toda a parte ou seja é contínuo em $\mathbb{R} = (-\infty, \infty)$. Segue a demonstração:

Um polinômio é uma função da forma $P(x) = c_n x^n + c_{n-1} x^{n-1} + ... + c_1 x + c_0$ onde $c_0, c_1, ...c_n$ são constantes. Sabe-se que $\lim_{x \to a} c_0 = c_0$ e que $\lim_{x \to a} x^m = a^m$ para todo $m = 1, 2, ..., n$ Desta última equação tem-se que $f(x) = x^m$ é contínua. Daqui, $g(x) = cx^m$é contínua. Uma vez que o polinômio é a soma destas funções e de uma constante, segue que P é contínua.

Mais funções contínuas As seguintes funções são contínuas para todo número de seus domínios:

polinômios visto acima

funções racionais Uma função racional é a razão de dois polinômios $f(x) = \frac{P(x)}{Q(x)}$

funções raízes $f(x) = x^{1/n} = \sqrt[n]{x}$

funções trigonométricas seno, cosseno e tangente

funções trigonométricas inversas secante, cossecante e cotangente

funções exponenciais na forma $f(x) = x^x$ onde a é uma constante positiva

funções logarítmicas na forma $f(x) = \log_a x$ onde a é uma constante positiva. Considerando que $\log_a x = N \Leftrightarrow a^N = x$ as funções exponenciais e logarítmicas são inversas.

Capítulo 5

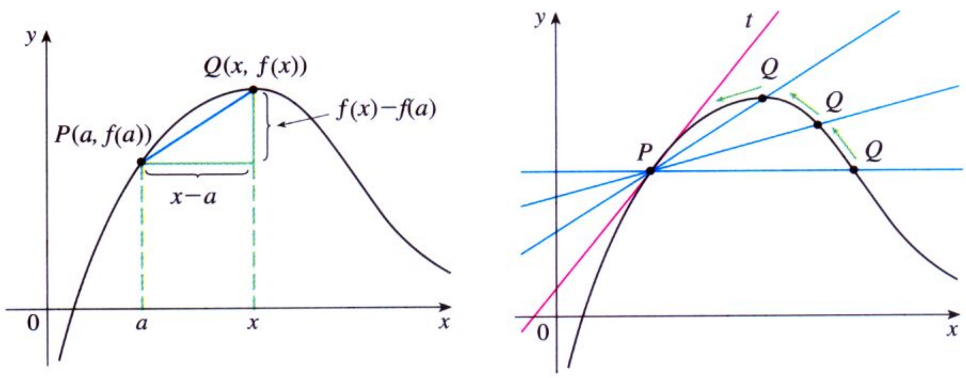
Derivadas

5.1 Derivada e taxa de variação

Se uma curva C tem uma equação $y = f(x)$ e precisa-se encontrar a tangente a C em um ponto $P(a, f(a))$ considera-se um ponto próximo $Q(x, f(x))$ onde $x \neq a$ e calcula-se a inclinação da reta secante PQ

$$m_{PQ} = \frac{f(x) - f(a)}{x - a}$$

ou



Este limite é muito frequente e surge sempre que se calcula a taxa de variação de alguma coisa, tanto que recebe nome e notação especiais:

A **derivada de uma função em um número** a denotada $f'(a)$ é

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$$

se o limite existir. Se escrevermos $x = a + h$, então $h = x - a$ e h tende a 0 se e somente se x tende a a. Consequentemente a fórmula acima pode ser reescrita como

$$f'(a) = \lim_{x \to a} \frac{f(x) - f(a)}{x - a}$$

Calculando uma derivada à mão Seja um exemplo:
Ache a derivada da função $f(x) = x^2 - 8x + 9$ em um número a. Da definição tem-se
$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$
$f'(a) = \lim_{h \to 0} \frac{[(a+h)^2 - 8(a+h) + 9] - [a^2 - 8a + 9]}{h}$
$f'(a) = \lim_{h \to 0} \frac{a^2 + 2ah + h^2 - 8a - 8h + 9 - a^2 + 8a - 9}{h}$
$f'(a) = \lim_{h \to 0} \frac{2ah + h^2 - 8h}{h}$
$f'(a) = \lim_{h \to 0} (2a + h - 8)$
$f'(a) = 2a - 8$

Outra interpretação da derivada é que a derivada $f'(a)$ é a taxa instantânea de variação de $y = f(x)$ em relação a x quando $x = a$ ou

$$f'(a) = \lim_{\Delta \to 0} \frac{\Delta y}{\Delta x} = \lim_{x_2 \to x_1} \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

Uma função descrita em palavras Interpretando isto para uma função descrita em palavras. Um fabricante produz peças de tecido com tamanho fixo. O custo em reais da produção de x metros de um certo tecido é $C = f(x)$.

a) Qual o significado da derivada $f'(x)$? Quais são suas unidades?

b) Em termos práticos, o que significa dizer que $f'(1000) = 9$?

c) O que é maior: $f'(50)$, $f'(500)$ ou $f'(5000)$?

Respostas

a) A derivada $f'(x)$ é a taxa de variação instantânea de C em relação a x, isto é como varia o custo de produção em relação ao número de metros produzidos. Os economistas chamam isto de *custo marginal.* Como

$$f'(x) = \lim_{\Delta \to 0} \frac{\Delta C}{\Delta x}$$

as unidades para $f'(x)$são as do quociente $\Delta C/\Delta x$. Como ΔC é medida em reais e Δx em metros, a unidade de f' é reais/metro.

b) A afirmação de que $f'(1000) = 9$ significa que depois de 1000 metros terem sido produzidos, a taxa do custo de produção está aumentando $9\$/m$ ou quando já se produziram 1000 m C está aumentando 9 vezes mais rápido que x. Uma vez que $\Delta x = 1$ é pequeno comparado com $x = 1000$, pode-se simplificar: $f'(1000) \approx \frac{\Delta C}{\Delta x} = \frac{\Delta C}{1} = \Delta C$ e dizer que o custo de fabricação do milésimo metro está em torno de 9 reais.

c) Provavelmente o custo de produção é menor ao produzir 500 m do que ao produzir 50, devido à economia de escala. Ou $f'(50) > f'(500)$ Mas, à medida em que a produção cresce pode se tornar ineficiente, talvez com custos de horas extras. Então pode acontecer que $f'(5000) > f'(500)$ O segredo da administração de empresas está em descobrir onde a função $f'(x)$ tem um mínimo.

5.1.1 Mais taxas de variação

♦ A velocidade de um objeto é a taxa de variação do deslocamento em relação ao tempo

♦ O custo marginal é a variação do custo de produção em relação ao número de ítens produzidos

♦ A taxa de variação de um débito em relação ao tempo descreve a amortização do mesmo

♦ Na física, a taxa de variação do trabalho em relação ao tempo é chamada *potência.* Na eletricidade, a corrente é a taxa de variação de cargas elétricas em relação ao tempo.

♦ Na química a taxa de variação da concentração de um reagente em relação ao tempo é a taxa de reação

♦ Na biologia, estuda-se a taxa de variação da população de uma colônia de bactérias em relação ao tempo

♦ Na geologia estuda-se a taxa pela qual uma massa de rocha fundida resfria para o meio rochoso que a envolve em realção ao tempo

♦ Na engenharia quer-se saber a taxa segundo a qual a água escoa de um reservatório

♦ Um geógrafo pode querer saber a taxa de variação da densidade da população à medida em que a distância do centro da cidade aumenta.

♦ Um metereologista pode precisar saber a variação da pressão atmosférica em relação a altura

♦ Em psicologia estuda-se a curva de aprendizado que é o gráfico do desempenho $P(t)$ de alguém aprendendo alguma coisa como função do tempo de estudo t. É importante estudar dP/dt isto é a taxa segundo a qual o desempenho melhra à medida em que o tempo passa.

♦ Em sociologia, ao analisar a divulgação de um boato se se tem $p(t)$ a proporção de uma população que conhece o boato no tempo t então a derivada dp/dt é a taxa de divulgação do boato.

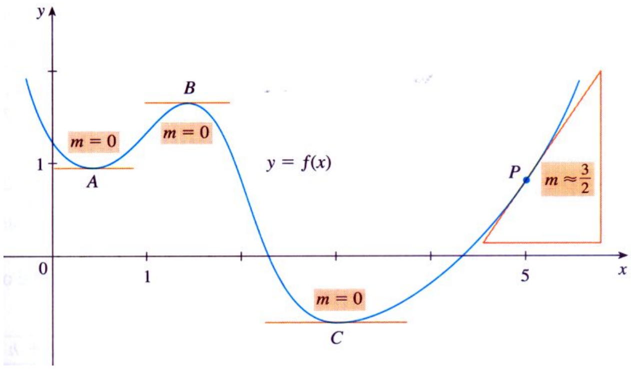
Uma única idéia, muitas interpretações. A velocidade, a densidade, a corrente, a potência, o gradiente da temperatura, a taxa de reação e a compressibilidade na química, o gradiente da velocidade do sangue, todos esses são casos especiais de um único conceito matemático: **a derivada**. É uma ilustração do fato de que parte do poder da matemática está em sua abstração. Um único conceito abstrato (a derivada) pode ter interpretações diferentes em cada ciência. Ao invés de gastar tempo e dinheiro em cada ramo do conhecimento, gasta-se-o na matemática e depois adapta-se a idéia a cada necessidade. Fourier (1768-1830) disse-o muito bem: *A matemática compara os mais diversos fenômenos e descobre as analogias secretas que os unem.*

Capítulo 6

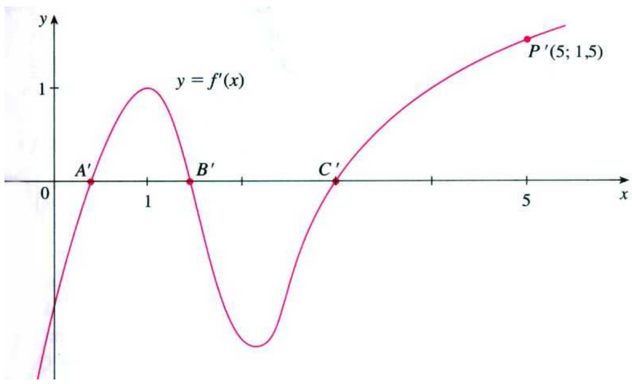
Regras de derivação

6.1 A derivada como função

Anteriormente calculamos a derivada em um ponto fixo. Agora vai-se fazer o número a variar. Agora $f'(x)$ é uma nova função **derivada** de $f(x)$. Geometricamente, é a inclinação da tangente à curva $(f(x)$ no ponto a. Vamos ver isto mais de perto. Suponha-se a função f na figura abaixo. Como será sua derivada ?



Pode-se estimar o valor da derivada em qualquer valor de x traçando a tangente no ponto $(x, f(x))$ e estimando a sua inclinação. Por exemplo, em $x = 5$ estima-se a inclinação como $3/2 = 1.5$, então $f'(5) = 1.5$. Isto permite desenhar o ponto $P'(5, 1.5)$sobre o gráfico de f'. Repetindo para diversos pontos, obtém-se a figura



Em A, B e C as tangentes são horizontais, logo a derivada ali é 0 e o gráfico de f'cruza o eixo x nos pontos A', B' e C' diretamente baixo de A, B e C. Entre A e B as tangentes têm inclinação positiva, logo $f'(x)$ é positiva ali. Entre B e C as tangentes têm inclinação genativa, logo $f'(x)$ lá é negativa.

6.2 Derivada do seno

Para treinar demonstrações, vai-se agora demonstrar a derivada da função $sen(x)$.

1. Seja calcular $\frac{d}{dx}[f(x)] =$
2. Da definição: $\dfrac{d}{dx}[f(x)] = \lim\limits_{h \to 0} \dfrac{f(x+h) - f(x)}{h}$

3. Aplicando $sen(x)$ fica: $\frac{d}{dx}[sen(x)] = \lim_{h\to 0} \frac{sen(x+h) - sen(x)}{h}$

4. Aplicando a fórmula da soma do seno que diz $sen(a+b) = sen(a)cos(b) + cos(a)sen(b)$, fica:
$\frac{d}{dx}[sen(x)] = \lim_{h\to 0} \frac{sen(x)cos(h) + cos(x)sen(h) - sen(x)}{h}$

5. Aplicando a propriedade da soma de limites que diz $\lim(a+b) = \lim(a) + \lim(b)$ e reorganizando a expressão, fica:
$\frac{d}{dx}[sen(x)] = \lim_{h\to 0} \frac{cos(x)sen(h)}{h} + \lim_{h\to 0} \frac{sen(x)cos(h) - sen(x)}{h}$

6. Colocando $sen(x)$ em evidência no segundo termo, fica:
$\frac{d}{dx}[sen(x)] = \lim_{h\to 0} \frac{cos(x)sen(h)}{h} + \lim_{h\to 0} \frac{sen(x)[cos(h) - 1]}{h}$

7. $cos(x)$ no primeiro limite e $sen(x)$ no segundo limite são constantes (não dependem de h) e podem ser postos em evidência:
$\frac{d}{dx}[sen(x)] = cos(x). \lim_{h\to 0} \frac{sen(h)}{h} + sen(x). \lim_{h\to 0} \frac{cos(h) - 1}{h}$

8. sabendo que $a - 1$ pode ser reescrito como $-(1 - a)$ fica
$\frac{d}{dx}[sen(x)] = cos(x). \lim_{h\to 0} \frac{sen(h)}{h} - sen(x). \lim_{h\to 0} \frac{1 - cos(h)}{h}$

9. Lembrando o limite fundamental $\lim_{a\to 0} sen(a)/a = 1$, que pode ser aceito por considerações geométricas e que tem excelente demonstração na academia Khan e em [Ste14].

10. Vamos provar $\lim_{a\to 0}(1 - cos(a))/a = 0$:

    (a) Supondo que $\lim_{t\to 0} \frac{sen(t)}{t} = 1$

    (b) Multiplicando numerador e denominador por $1 + cos(a)$ fica:
    $\lim_{a\to 0} \frac{(1 - cos(a))(1 + cos(a))}{a(1 + cos(a))} =$

    (c) Lembrando que $(a - b)(a + b) = a^2 - b^2$ fica
    $\lim_{a\to 0} \frac{(1 - cos^2(a))}{a(1 + cos(a))} =$

    (d) Da igualdade fundamental (Teorema de Pitágoras) $sen^2(t) + cos^2(t) = 1$ fica:
    $\lim_{a\to 0} \frac{(sen^2(a))}{a(1 + cos(a))} =$

    (e) Reescrevendo $sen^2(t) = sen(t)sen(t)$ e reorganizando a expressão:
    $\lim_{a\to 0} \frac{sen(a)}{a}.\frac{sen(a)}{1 + cos(a)} =$

    (f) Agora, o limite de um produto é o produto dos limites
    $\lim_{a\to 0} \frac{sen(a)}{a}. \lim_{a\to 0} \frac{sen(a)}{1 + cos(a)} =$

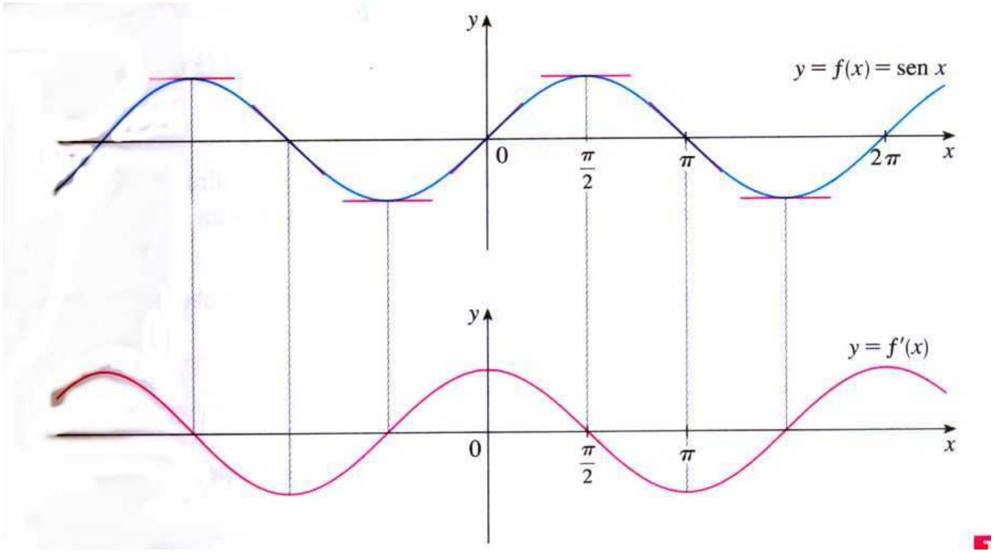
    (g) O primeiro limite, vimos acima, vale 1

    (h) O segundo limite é fácil. Quando a tende a 0, o sen(a) tende a 0 e o cos(a) tende a 1. Substituindo o segundo limite é $\frac{0}{1+1} = \frac{0}{2} = 0$. E $1 \times 0 = 0$

    (i) Finalmente $\lim_{a\to 0}(1 - cos(a))/a = 0$, CQD

11. Ufa, $\frac{d}{dx}[sen(x)] = cos(x)$

Restam as equações fundamentais que são $y = sen(x) \Rightarrow y' = cos(x)$ e $y = cos(x) \Rightarrow y' = -sen(x)$, cujo acerto pode ser observado nos seus gráficos

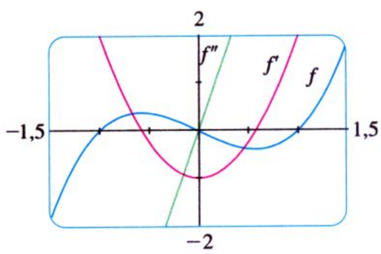


6.3 Derivada de ordem superior

Se $f(x)$ é uma função diferenciável, então sua derivada $f'(x)$ também é uma função e pode portanto ter a sua própria derivada denotada por $(f')'(x)$, ou mais simplificadamente $f''(x)$. Esta é a segunda derivada de $f(x)$. Usando a notação de Leibniz, pode-se escrever a segunda derivada como

$$\frac{d}{dx}\left(\frac{dy}{dx}\right) = \frac{d^2y}{dx^2}$$

A derivada segunda pode ser interpretada como a inclinação da derivada primeira em um ponto. É portanto a taxa de variação da inclinação da curva original. Acompanhe no desenho a seguir a função $f(x) = x^3 - x$ (na cor azul), sua primeira derivada $f'(x) = 3x^2 - 1$ (cor magenta) e a sua derivada segunda $f''(x) = 6x$ (verde).



Observe que $f''(x)$ é negativa quando $f'(x)$ tem inclinação negativa e positiva quando $f'(x)$ tem inclinação positiva.

De maneira idêntica pode-se definir as derivadas de ordem superior, sem parar. Se tem-se uma função de deslocamento $y = f(x)$, a derivada primeira é a velocidade, a derivada segunda é a aceleração e a derivada terceira recebe o nome de *jerk* (jerk = solavanco, sacudida) que é variação da aceleração. Lembre quando um avião desembesta na pista, com uma variação súbita na aceleração quem está sentado sofre um movimento abrupto.

A taxa média de mudança de uma função f em relação à sua variável independente x é o quociente

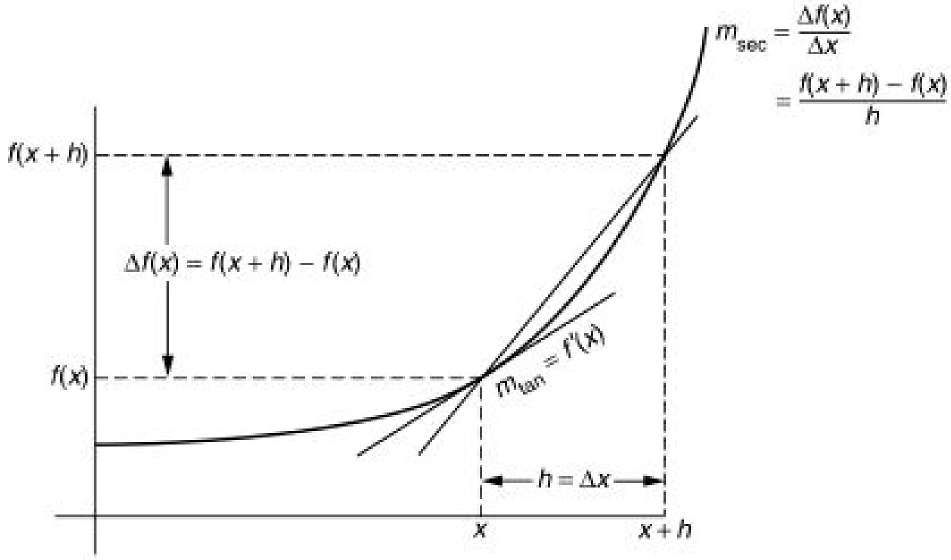
$$\frac{\Delta f(x)}{\Delta x} = \frac{f(x+h) - f(x)}{h}$$

A taxa instantânea de mudança é

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

que por definição é $f'(x)$. Em palavras, a taxa instantânea de variação de uma função em relação à variação de sua variável independente é a derivada dessa função em relação a essa variável.

Graficamente a taxa de mudança de uma função em um ponto equivale à inclinação da tangente à função nesse ponto. A taxa média de mudança em um intervalo é a inclinação da secante conectando os pontos da curva correspondentes aos limites do intervalo.



Exemplo 1

(a) Ache a taxa média de mudança da área de um quadrado em relação a seu lado x quando x varia de 4 a 7.

(b) Ache as taxas de mudança da área de um quadrado com respeito ao lado x quando $x = 4, 5, 6\ e\ 7$

Solução

(a) $A(x) = x^2$ A taxa média de mudança de $A(x)$ em relação a x é $\frac{\Delta A(x)}{\Delta x}$.

$$A(x) = x^2$$

$$A(4) = 16, \quad A(7) = 49$$

$$\Delta A(x) = A(7) - A(4) = 33$$

$$\Delta x = 7 - 4 = 3$$

$$\frac{\Delta A(x)}{\Delta x} = \frac{33}{3} = 11$$

Alternativamente:

$$A(x) = x^2,\ x = 4,\ h = 3$$

$$\frac{A(x+h) - A(x)}{h} = \frac{A(7) - A(4)}{3} = \frac{49 - 16}{3} = \frac{33}{3} = 11$$

(b) Determina-se a taxa instantânea computando a derivada e calculando-a nos pontos $x = 4, 5, 6\ e\ 7$

$$A(x) = x^2$$

$$A'(x) = 2x$$

$$A'(4) = 8$$

$$A'(5) = 10$$

$$A'(6) = 12$$

$$A'(7) = 14$$

Perceba que a área cresce mais rápido à medida em que x cresce.

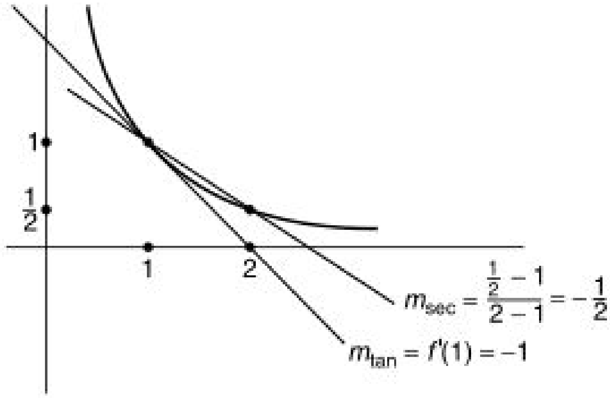
Exemplo 2 Ache a taxa de mudança do volume de uma esfera com relação a seu raio r quando o raio é 5.
O volume da esfera de raio r é $V(r) = \frac{4}{3}\pi r^3$. Daqui $V'(r) = 4\pi r^2$ e $V'(5) = 4.25.\pi = 100\pi$.

Exemplo 3 Seja $f(x) = 1/x$. Determine (a) a taxa de mudança de f em $x = 1$ e (b) a taxa média de mudança entre $x = 1$ e $x = 2$. Interprete os resultados graficamente.

(a) $f(x) = x^{-1}$ e $f'(x) = -x^{-2} = -\frac{1}{x^2}$ e $f'(1) = -1$ logo a taxa de mudança é -1.

(b) $x = 1,\ h = 1$ e

$$\frac{f(x+h) - f(x)}{h} = \frac{\frac{1}{x+h} - \frac{1}{x}}{h} = \frac{\frac{1}{2} - \frac{1}{1}}{1} = -\frac{1}{2}$$



6.4 Movimento em linha reta

Velocidade é a taxa de mudança da posição de um objeto que se move em relação ao tempo. Se estamos estudando a viagem de 100 milhas de um carro ao longo de uma estrada reta a análise de tempos e distâncias pode nos informar sobre como foi a viagem. Se a viagem foi feita em 2 horas pode-se afirmar que a sua velocidade média foi de 50 milhas/hora. A velocidade média do carro é dada por

$$v_{av} = \frac{\Delta s}{\Delta t}$$

onde s representa a posição do carro relativo a um ponto fixo.

A velocidade instantânea v é definida pelo limite da velocidade média quando o intervalo de tempo tende a 0.

$$v = \lim_{\Delta t \to 0} \frac{\Delta s}{\Delta t}$$

Esta quantidade é a derivada de s em relação a t ou

$$v = \frac{ds}{dt}$$

Exemplo 4 Uma bicicleta viaja ao longo de uma estrada reta. À 1h00 ela está a 1 milha do final da estrada e às 4h00 está a 16 milhas do final da estrada. Calcule (a) a velocidade média entre 1h00 e 4h00 e (b) a velocidade instantânea às 03h00.

(a) Seja s a posição da bicicleta relativamente ao final da estrada

$$v_{av} = \frac{\Delta s}{\Delta t} = \frac{16 - 1}{4 - 1} = \frac{15}{3} = 5\ mi/h$$

(b) Não se pode resolver esta parte do problema, já que não se sabe a localização da bicicleta ao longo do tempo. Não há as informações suficentes para calcular a velocidade instantânea. Para calcular isto, precisa-se conhecer a posição da bicicleta como uma função do tempo.

Exemplo 5 Uma bicicleta viaja ao longo de uma estrada reta. No instante t ela está a t^2 milhas do final da estrada. Calcule (a) a velocidade média entre 1h00 e 4h00 e (b) a velocidade instantânea às 03h00.

(a) Se $s = t^2$, quando $t = 1,\ s = 1$ e quando $t = 4,\ s = 16$ e obtemos os mesmos resultados vistos acima ou $v_{av} = 5\ mi/h$

(b) Já que a posição da bicicleta agora é conhecida em qualquer tempo, a velocidade instantânea pode ser calculada. Seja $S = t^2$ então $v(t) = s'(t) = 2t$. Às 3h00 então $t = 3$ e $s'(3) = 6\ mi/h$.

A derivada determina a velocidade de um objeto se movendo. Uma questão sutil: a velocidade pode ser positiva, zero ou negativa. Já a rapidez sempre é positiva e pode ser definida como o módulo da velocidade.

Exemplo 6 A posição de uma partícula (em polegadas) ao longo do eixo x após t segundos é dada pela equação

$$x = 24t^2 - t^3 + 10$$

(a) Qual a velocidade média da partícula durante os 3 primeiros segundos de viagem ?
(b) Onde a particula está e quão rápido ela se move após 3 segundos de viagem ?
(c) Onde a particula está e quão rápido ela se move após 20 segundos de viagem ?
(d) Quando a velocidade da partícula é 0 ? Onde ela está nessa hora ?
(e) Descreva o movimento da partícula durante os primeiros 20 segundos da viagem

solução
(a) A posição em $t = 0$ é $x = 10$. Em $t = 3$, $x = 199$

$$v_{av} = \frac{\Delta x}{\Delta t} = \frac{199 - 10}{3 - 0} = \frac{189}{3} = 63\ in/sec$$

(b) Quando $t = 3$ a posição é $x = 199$ A velocidade instantânea é determinada usando a derivada

$$v = \frac{dx}{dt} = 48t - 3t^2$$

quando $t = 3$, $v = 117in/seg$
(c) Quando $t = 20$ sua posição é $24(20)^2 - 20^3 + 10 = 1610$. Como $v = \frac{dx}{dt} = 48t - 3t^2$, quando $t = 20$ então $v = -240in/seg$. O valor negativo indica que ela se move na direção negativa, com uma rapidez de $240in/seg$.
(d)

$$v = 48t - 3t^2$$

$$0 = 48t - 3t^2 \quad = \quad 3t(16 - t)$$

e as raízes são $t = 0$ e $t = 16$ A velocidade é zero em $t = 0$ e em $t = 16$. Quando $t = 0$, $x = 10$ e quando $t = 16$, $x = 2058$
(e) A partícula começa do repouso em $x = 10$. Pelos primeiros 16 segundos, tem uma velocidade positiva e se move na direção positiva. Quando $t = 16$ a partícula pára momentaneamente ($v = 0$) em $x = 2058$. Agora ela se move na direção negativa e retorna a $x = 1610$ no instante $t = 20$.

A *aceleração* de um objeto se movendo é a taxa de mudança de sua velocidade em relação ao tempo. Então $a(t) = v'(t)$. Se $s(t)$ representa a posição de um objeto no instante t então $a(t) = s''(t)$.

Exemplo 7 Calcule a aceleração da partícula do exemplo 6 nos tempos $t = 3, 5, 10$ e 15

Já que $v(t) = 48t - 3t^2$, $a(t) = v'(t) = 48 - 6t$.
Quando $t = 3$, $a = 30pol/seg^2$; quando $t = 5$, $a = 18pol/seg^2$; $t = 10$, $a = -12pol/seg^2$ e $t = 15$, $a = -42pol/seg^2$. Neste exemplo a unidade de velocidade é polegada por segundo. Daí a unidade de aceleração deve ser polegada por segundo por segundo. Isto é usualmente escrito $pol/seg/seg$ ou pol/seg^2.

Exemplo 8 A altura em pés em qualquer instante t de um projétil lançado verticalmente é

$$h(t) = -16t^2 + 256t$$

(a) Qual a velocidade média do prejétil nos primeiros 5 segundos de viagem ?
(b) Qual a velocidade instantânea no tempo $t = 6$? Quão alto ele está ?
(c) Qual a velocidade instantânea no tempo $t = 10$? Quão alto ele está ?
(d) Quando ocorre a altura máxima ? Qual a altura máxima do projétil ?
(e) Quando o projétil retorna ao solo e com qual velocidade ?
(f) Descreva o movimento do projétil

(a) $h(t) = -16t^2 + 256t$. Quando $t = 0$, $h = 0$ e quando $t = 5$, $h = -16(5)^2 + 256(5) = 880pés$

$$v_{av} = \frac{\Delta h}{\Delta t} = \frac{880 - 0}{5 - 0} = 176p/seg$$

(b) Para obter a velocidade instantânea, deve-se computar a derivada $\frac{dh}{dt}$. A velocidade $v = \frac{dh}{dt} = -32t + 256$. Quando $t = 6$, $v = -32(6) + 256 = 64p/seg$. A altura do projétil com $t = 6$ é $h = -16(6)^2 + 256(6) = 960pés$
(c) $v = \frac{dh}{dt} = -32t + 256$. Quando $t = 10$, $v = -64p/seg$. A velocidade negativa indica que o projétil está se movendo na direção negativa (descendo) com a velocidade de 64 pés por segundo. A altura do projétil com $t = 10$ é $h = -16(10)^2 + 256(10) = 960$ pés.
(d) A altura máxima é alcançada quando $v = 0$. Como $v = -32t + 256$, $t = 8seg$. A altura máxima (neste ponto) é de $h_{max} = -16(8)^2 + 256(8) = 1024$ pés.
(e) O projétil retorna ao chão quando $h = 0$ e como $h(t) = -16t^2 + 256t = 0 = -16t(t - 16)$ tem-se $t = 0$ e $t = 16$. Já que o projétil começou sua viagem em $t = 0$ ele retorna ao chão em $t = 16$. Sua velocidade neste instante é $v(t) = -32t + 256$.

Daqui $v(16) = -32(16) + 256 = -256$ pés/seg. A velocidade negativa indica queda.
(f) Quando $t = 0$, $h = 0$ e o projétil está no nível do solo. Sua velocidade é de 256 p/seg. É a sua velocidade inicial. À medida em que t aumenta, a altura h aumenta e a velocidade v diminui, atingindo o valor $v = 0$ no instante $t = 8$. Neste ponto o projétil alcança sua altura máxima de 1024 pés. A partir de agora a velocidade torna-se negativa e h diminui, com o projétil retornando ao solo, o que ocorre em $t = 16$.

A seguir a construção do exemplo acima em Python


```
import numpy as np
import matplotlib.pyplot as plt
def h(t):
  return -16*t**2+256*t
def v(t):
  return -32*t+256
def p34():
  t=np.arange(0,16,0.1)
  hh=h(t)
  vv=v(t)
  plt.plot(t,hh,'y-')
  plt.plot(t,vv,'r-')
  plt.grid()
  plt.show()
p34()
```

6.5 Aplicações à ciência e engenharia

Exemplo 9 A densidade linear de uma vara é a taxa de mudança de sua massa em relação ao seu comprimento. Uma vara não homogênea tem um comprimento de 9 pés e uma massa de 24 slugs [1]. Se a massa de uma sessão da vara de comprimento x – medido à partir do seu limite mais à esquerda – é proporcional à raiz quadrada de seu comprimento,
(a) calcule a densidade média da vara e
(b) determine a função de densidade e calcule a densidade da vara de 4 pés a partir do seu limite à esquerda.

Representando $m(x)$ como a massa da sessão de comprimento x. A descrição do problema nos diz que $m(x) = k\sqrt{x}$. Já que a massa total da vara é 24 slugs, então $m(9) = 24$

$$m(9) - 24$$

$$k\sqrt{9} = 24$$

$$3k = 24$$

$$k = 8$$

Então $m(x) = 8\sqrt{x}$
(a) A densidade média é $\frac{m(9)-m(0)}{9-0} = \frac{24-0}{9} = \frac{8}{3} slugs/feet$
(b) Representando a função de densidade por $\rho(x)$. Seja $\rho(x) = m'(x)$ e

$$m(x) = 8\sqrt{x} = 8x^{1/2}$$

$$\rho(x) = m'(x) = 4x^{-1/2} = \frac{4}{\sqrt{x}}$$

$$\rho(4) = 2\ slugs/ft$$

[1] O slug é a unidade de massa no sistema inglês e equivale a 14,59 kilogramas

Exemplo 10 O movimento de elétrons através de um fio produz uma corrente elétrica. Se Q é a carga fluindo através do fio medida em coulombs, então a corrente I medida em ampéres, é a taxa de mudança de Q em relação ao tempo medido em segundos.

A carga em coulombs que passa através de um fio após t segundos é dada pela função

$$Q(t) = t^3 - 2t^2 + 5t + 2$$

(a) Determine a corrente média durante os primeiros 2 segundos
(b) Determine a corrente ao final de 2 segundos

(a) $I_{av} = \frac{\Delta Q}{\Delta t} = \frac{Q(2)-Q(0)}{2-0} = \frac{12-2}{2} = 5$ *ampéres*
(b) $I(t) = Q'(t) = 3t^2 - 4t + 5$
$i(2) = 3(2)^2 - 4(2) + 5 = 9$ *ampéres*

Exemplo 11 O número de bactérias em uma placa de petri após t horas é de $n(t) = 2t^3 + 5t + 2$. Quão rápido a população cresce após 3 horas ?

$$\frac{dn}{dt} = 6t^2 + 10t + 1$$

Quando $t = 3$, $dn/dt = 85$. A colonia cresce a 85 bactérias por hora.

Exemplo 12 Um tanque de 1800 litros de água drena (escoa) em 30 minutos. De acordo com a lei de Torricelli[2] é

$$V = 1800\left(1 - \frac{t}{30}\right)^2 \quad 0 \le t \le 30$$

Quão rápido a água drena do tanque após 20 minutos ?

$$\begin{aligned}\frac{dV}{dt} &= 3600\left(1 - \frac{t}{30}\right)\left(-\frac{1}{30}\right)\\ &= -120\left(1 - \frac{t}{30}\right)\end{aligned}$$

Quando $t = 20$

$$\frac{dV}{dt} = -120\left(1 - \frac{20}{30}\right) = -40 \; litros/minuto$$

A água drena a 40 litros/minuto. O sinal negativo indica que o volume de água diminui.

Exemplo 13 Um estudo ambiental em uma pequena cidade indica que em t anos o nível de monóxido de carbono será $q(t) = 0.005t^3 + 0.02t^2 + 0.01t + 2.5$ partes por milhão (ppm). A que taxa o monóxido de carbono crescerá em 1 e 2 anos a partir de agora ?
A taxa de mudança do nível de CO após t anos é $q'(t) = 0.015t^2 + 0.04t + 0.01$
Após 1 ano, $q'(1) = 0.015 + 0.04 + 0.01 = 0.065 \; ppm/ano$
Após 2 anos, $q'(2) = 0.06 + 0.08 + 0.01 = 0.15 \; ppm/ano$

6.6 Administração e economia

O custo marginal de produção de um item é a taxa pela qual os custos variam em relação ao número de itens produzidos. Então, se $C(x)$ é o custo de produção de x itens, o custo marginal é $C'(x)$. O conceito de custo marginal é o custo adicional para produzir um item adicional. Assim, $C'(x)$ é o custo adicional para produzir o item $(x + 1)$. Similarmente, se $R(x)$ e $L(x)$ são a receita e o lucro ao vender uma quantidade de x unidades, então $R'(x)$ e $L'(x)$ representam a receita marginal e o lucro marginal.

Exemplo 14 Suponha que o custo total de produção de x itens é dado pela função

$$C(x) = 0.001x^3 + 0.025x^2 + 3x + 5$$

Calcule o custo marginal para produzir o 51^o item.

Já que $C'(x)$é o custo aproximado para produzir o elemento $(x + 1)$, primeiro precisa-se calcular o custo de $C'(50)$

$$C'(x) = 0.003x^2 + 0.05x + 3$$

[2]esta lei estuda o fluxo de um líquido contido em um recipiente, através de um pequeno orifício, sob a ação da gravidade

$$C'(50) = 0.003(50)^2 + 0.05(50) + 3 = 7.5 + 2.5 + 3 = 13.00$$

O custo marginal é \$13.00. Para comparação o custo exato para produzir o item 51 é

$$C(51) - C(50) = [0.001(51)^3 + 0.025(51)^2 + 3(51) + 5] - [0.001(50)^3 + 0.025(50)^2 + 3(50) + 5] = 355.676 - 342.50 = 13.176$$

Exemplo 15 O valor de uma máquina após t anos de serviço é $V(t) = 100t^2 - 3000t + 20000$ dólares. A qual taxa a maquina deve ser depreciada após 5 anos ?

A taxa de depreciação é a taxa na qual valor é perdido. Se $V(t)$ representa o valor da máquina após t anos, $V'(t)$ representa a taxa pela qual o valor muda

$$V'(t) = 200t - 3000$$

$$V'(5) = -2000$$

A máquina deprecia a 2000/ano após 5 anos.

6.7 Problemas suplementares

- 1. Uma caixa fechada com uma base quadrada tem volume de $100pol^3$. Compute a taxa de mudança de sua superfície com relação a sua dimensão de base x, quando $x = 5$.

- 2. Se $f(x) = \sqrt{x}$ determine
(a) a taxa de mudança em relação a x quando $x = 16$
(b) a taxa média de mudança de f de $x = 15$ até $x = 25$

- 3. A posição de um objeto se movendo após t minutos de viagem é $s(t) = 2t^3 - 27t^2 + 84t + 25$ *pés*
(a) Qual a velocidade média durante os primeiros 2 minutos ?
(b) Onde o objeto está e quão rápido ele se move após 1 minuto ? 5 minutos ? 10 minutos ?
(c) Qual a aceleração do objeto após 1 minuto ? 5 minutos ? 10 minutos ?
(d) Quando o objeto alcança $v = 0$? Onde ele está ?
(e) Descreva o movimento nos primeiros 10 minutos ?

- 4. A altura alcançada por um projétil atirado para cima no topo de um prédio de 96 pés é de $h(t) = 80t - 16t^2 + 96$ (dica: projétil [pl: projéteis] e projetil [pl:projetis] – ambas estão certas)
(a) Qual a velocidade média do projétil nos primeiros 2 segundos ?
(b) Qual a velocidade inicial ?
(c) Qual a velocidade após 5 segundos ? Qual a sua posição ?
(d) Qual a altura máxima alcançada pelo projetil ?
(e) Quando o projetil acerta o chão e qual a sua velocidade nessa hora?

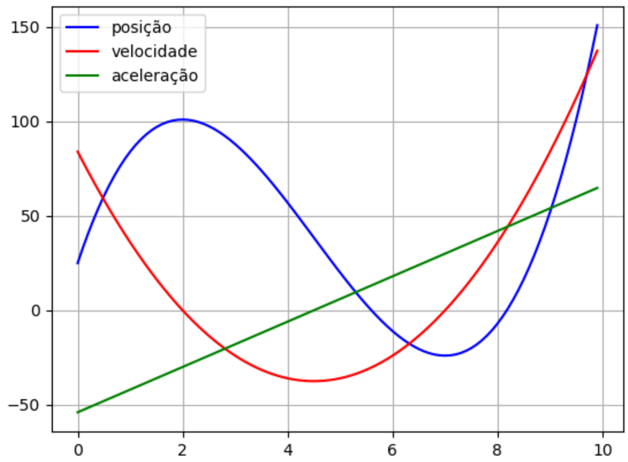
- 5. Um cajado de 5 pés tem uma densidade linear proporcional ao cubo da distância de uma das extremidades. Se a massa total é de 500 slugs, calcule
(a) a densidade média do cajado
(b) a densidade no ponto médio do cajado

- 6. A Lei de Boyle afirma que quando um gás é comprimido a temperatura constante, o produto da pressão pelo volume permanece constante. Se a pressão de um gás é de $80lb/in^2$ quando o volume é de $40in^3$, ache a taxa de variação da pressão em relação ao volume quando o volume é de $20in^3$

- 7. Se x bolinhas são vendidas, o preço por bolinha é de $500 - 2x - 0.1x^2$ reais. Calcule o faturamento marginal derivada da venda da 11^a bolinha.

As respostas feitas estão no material, págs 41 a 45.

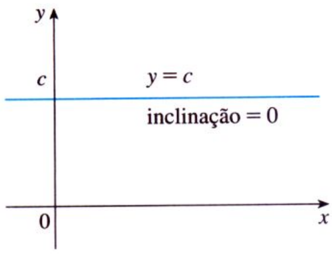
A seguir a construção do exercício 3 acima em Python




```
import numpy as np
import matplotlib.pyplot as plt
def s(t):
  return 2*t**3-27*t**2+84*t+25
def v(t):
  return 6*t**2-54*t+84
def a(t):
  return 12*t-54
def p40():
  t=np.arange(0,10,0.1)
  ss=s(t)
  vv=v(t)
  aa=a(t)
  plt.plot(t,ss,'b-',label='pos')
  plt.plot(t,vv,'r-',label='vel')
  plt.plot(t,aa,'g-',label='ace')
  plt.grid()
  plt.legend()
  plt.show()
p40()
```

6.8 Função constante

Seja $f(x) = c$. Como se pode ver em

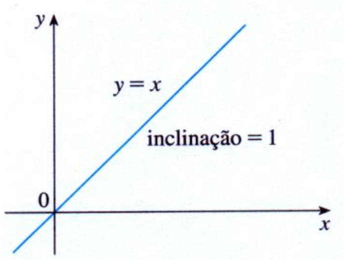


Esta função corresponde à reta horizontal que cruza o eixo y em c. A inclinação desta reta é sempre 0. Da definição de derivada, tem-se

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} = \lim_{h\to 0} \frac{c-c}{h} = \lim_{h\to 0} 0 = 0$$

6.9 Função potência

Vão-se ver as funções $f(x) = x^n$ onde n é um inteiro positivo. Se $n = 1$ o gráfico de $f(x) = x$ é a reta $y = x$ cuja inclinação é 1 como em



então

$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} = \lim_{h\to 0} \frac{x+h-x}{h} = \lim_{h\to 0} 1 = 1$$

Para os casos de $n = 2$ já se viu que $f'(x^2) = 2x$ e $f'(x^3) = 3x^2$. Vai-se ver agora o caso para $n = 4$. Se $y = x^4$ então

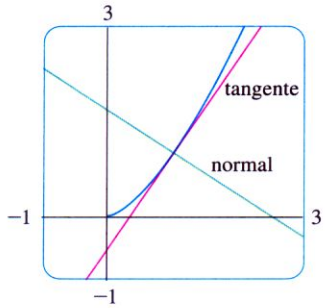
$$f'(x^4) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} = \lim_{h\to 0} \frac{(x+h)^4 - x^4}{h}$$

$$= \lim_{h\to 0} \frac{x^4 + 4x^3h + 6x^2h^2 + 4xh^3 + h^4 - x^4}{h}$$

$$= \lim_{h\to 0} \frac{4x^3h + 6x^2h^2 + 4xh^3 + h^4}{h}$$

$$= \lim_{h\to 0} (4x^3 + 6x^2h + 4xh^2 + h^3) = 4x^3$$

Vai-se resolver um exemplo: Encontre as equações da reta tangente e da reta normal à curva $y = x\sqrt{x}$ no ponto $(1,1)$. Ilustre o gráfico da curva e das retas. Começa-se informando que a reta normal e a reta tangente são perpendiculares, portanto $m_t \times m_n = -1$. Segue o barco: A derivada de $f(x) = x\sqrt{x} = xx^{1/2} = x^{3/2}$ é $f'(x) = \frac{3}{2}x^{(3/2)-1} = \frac{3}{2}x^{1/2} = \frac{3}{2}\sqrt{x}$ Logo a inclinação da reta tangente em (1,1) é $f'(1) = \frac{3}{2}$ Portanto, uma equação da reta tangente é $y - 1 = \frac{3}{2}(x-1)$ ou $y = \frac{3}{2}x - \frac{1}{2}$. A reta normal é perpendicular à reta tangente de modo que sua inclinação é $-\frac{2}{3}$. Assim, a equação de uma normal é $y - 1 = -\frac{2}{3}(x-1)$ ou $y = -\frac{2}{3}x + \frac{5}{3}$. No desenho a seguir a função original é azul, a tangente é magenta e a normal é verde.



6.10 Derivada de função multiplicada por constante

A regra diz que *A derivada de uma constante vezes uma função é a constante vezes a derivada da função.* Seja um exemplo: Se $f(x) = 3x^4$ então a sua derivada $f'(3x^4)$ é $3.f'(x^4) = 3.4x^3 = 12x^3$.

6.11 Derivada de soma de funções

A regra diz que *a derivada de uma soma de funções é igual à soma das derivadas de cada função.* Então se $f(x) = g(x) + h(x)$ então a derivada $f'(x)$ é igual a $g'(x) + h'(x)$. Note que esta regra usada em conjunto com a regra da multiplicação por constante ajuda muito a achar as derivadas de polinômios de maneira quase imediata.

6.12 Derivada de função exponencial

Veja-se o que acontece ao usar a definição Se $f(x) = a^x$ então

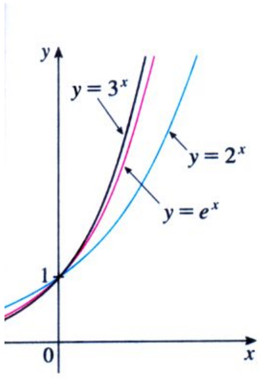
$$f'(x) = \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} = \lim_{h\to 0} \frac{a^{x+h} - a^x}{h} = \lim_{h\to 0} \frac{a^x a^h - a^x}{h} = \lim_{h\to 0} \frac{a^x(a^h - 1)}{h}$$

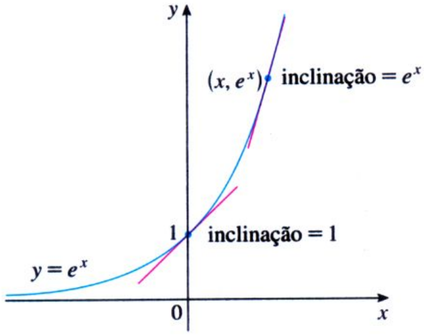
O fator a^x não depende de h, e pode ser tirado do limite. Fica

$$f'(x) = a^x \lim_{h\to 0} \frac{a^h - 1}{h} = f'(0).a^x$$

Esta equação diz que a *taxa de variação de qualquer função exponencial é proporcional à própria função.* A inclinação é proporcional à altura.

Numericamente se $a = 2$, $f'(0) \approx 0.693147$ e se $a = 3$, $f'(0) \approx 1.098612$. De todas as possíveis escolhas para a base a do exemplo acima a derivação mais simples ocorre quando $f'(0) = 1$. Em vista das estimativas para a acima vistas, parece plausível haver um número entre 2 e 3 para o qual $f'(0) = 1$. É tradição denotar este número por e (em homenagem a Euler). Define-se e como $\lim_{h\to 0} \frac{e^h - 1}{h} = 1$. Geometricamente, isto significa que de todas as possíveis exponenciais $y = a^x$ a função $f(x) = e^x$ é aquela cuja tangente em (0,1) tem uma inclinação $f'(0)$ que é exatamente 1. Veja nas figuras





Daqui, se $f(x) = e^x$ então $f'(x) = e^x$ e esta é a única função do universo que é igual à sua derivada.

6.13 Regra do produto

Alguém poderia achar (como Leibniz o fez há 3 séculos), que por analogia com a regra da soma, que a derivada de um produto é o produto das derivadas. Cairia do cavalo, como veremos em um exemplo. Seja $f(x) = x$ e $g(x) = x^2$. A regra da potência fornece $f'(x) = 1$ e $g'(x) = 2x$. Mas $(fg)(x) = x^3$, logo $(fg)'(x) = 3x^2$ e como $2x \neq 3x^2$ temos que, no caso mais geral, $(fg)' \neq f'g'$. Para não deixar o Leibniz mal, deve-se dizer que ele achou a fórmula correta, depois da vacilada acima. É chamada *regra do produto*. A derivada do produto de duas funções é igual a uma soma de 2 parcelas. Cada parcela toma uma função multiplicada pela derivada da outra. No exemplo acima, se $f(x) = x$ e $g(x) = x^2$, a derivada do produto é $f(x).g'(x) + f'(x)g(x) = x.2x + 1.x^2 = 2x^2 + x^2 = 3x^2$. Então, se $f(x)$ e $g(x)$ são deriváveis, a derivada do produto é

$$(f(x).g(x))' = f'(x).g(x) + f(x).g'(x)$$

Vejamos um exemplo, para treinar: Se $f(x) = x.e^x$ ache a derivada primeira e depois a n-ésima derivada. Pela regra do produto, fazemos $f(x) = x$ e $g(x) = e^x$, daqui a derivada primeira é $1.e^x + x.e^x = e^x(1 + x) = (1 + x)e^x$. Aplicações subsequentes da regra do produto nos dão $f^{(n)} = (x + n)e^x$ que é a resposta procurada.

6.14 Regra do quociente

Se $f(x) = \frac{g(x)}{h(x)}$ a derivada $f'(x)$ é igual a

$$f'(x) = \left(\frac{g(x)}{h(x)}\right)' = \frac{hg' - gh'}{h^2}$$

Note que deve cuidar com o numerador e o denominador (ao contrário da regra do produto). As parcelas são denominador vezes a derivada do numerador menos o numerador vezes a derivada do denominador, tudo dividido pelo denominador ao quadrado, ufa !

Note-se que com as regras acima, já é possível achar a derivada de qualquer função racional do tipo $\frac{P(x)}{Q(x)}$ onde $P(x)$ e $Q(x)$ são polinômios.

6.15 Derivadas de funções trigonométricas

Já estudamos lá acima e portanto se $f(x) = sen(x)$, então $f'(x) = cos(x)$ e se $f(x) = cos(x)$ então $f'(x) = -sen(x)$. Também se $f(x) = tan(x)$ então $f'(x) = sec^2(x)$.

6.16 Regra da Cadeia

Suponha precisar derivar

$$F(x) = \sqrt{x^2 + 1}$$

Perceba-se que F é uma função composta. Se assumirmos $y = f(u) = \sqrt{u}$ e $u = g(x) = x^2 + 1$ então pode-se escrever $y = F(x) = f(g(x))$ ou seja $F = f \circ g$. Sabe-se derivar ambas, f e g, então é útil uma regra para achar a derivada de $F = f \circ g$ em termos das derivadas de f e de g. É a regra da cadeia que diz que a derivada da função composta $f \circ g$ é o produto das derivadas de f e de g.

Eis um exemplo: Encontre $F(x) = \sqrt{x^2 + 1}$. Encontre $F'(x)$. Escrevendo $f(u) = \sqrt{u}$ e $g(x) = x^2 + 1$, daqui obtém-se com facilidade as derivadas $f'(u) = \frac{1}{2}u^{-1/2} = \frac{1}{2\sqrt{u}}$ e $g'(x) = 2x$ $F'(x) = f'(g(x)).g'(x) = \frac{1}{2\sqrt{x^2+1}}.2x = \frac{x}{\sqrt{x^2+1}}$

Mais um exemplo: Derive $y = sen(x^2)$ e depois $y = sen^2(x)$.
Solução do primeiro: se $y = sen(x^2)$ a função de fora é a função *sen* e a de dentro é a quadrática. Logo a derivada é
derivada da função de fora *cos*
avaliada na função de dentro (x^2)
derivada da função de dentro $2x$

Solução do segundo:
Veja que agora a função de fora é a quadrática e a de dentro é a função sen. Logo
derivada da função de fora 2
avaliada da função de dentro $senx$
derivada da função de dentro $.cosx$

Respostas: a primeira é $2xcos(x^2)$ e a segunda é 2 $senx\ cosx$ ou então $sen\ 2x$ pela identidade trigonométrica ($sen\ 2x = 2senx\ cosx$).

Mais um exemplo: Derive $y = e^{sen\ x}$. Aqui a função de dentro é $g(x) = senx$ e a função de fora é a função exponencial $f(x) = e^x$. Logo, pela regra da cadeia: $= e^{sen\ x}\frac{d}{dx}(senx) = e^{sen\ x}cosx$.

6.17 Derivação implícita

Suponha uma função definida implicitamente por uma relação entre x e y como $x^3 + y^3 = 6xy$. Quando possível pode-se isolar uma variável e resolver a derivada como já estudado. Mas nem sempre é fácil. (Observação: O Maple achou fácil, mas as respostas não couberam na tela...). Esta equação é a chamada *Fólio de Descartes*. Para resolver pode-se usar o método da derivação implícita: O método consiste em derivar ambos os lados da equação em relação a x e então na resolução isolando y'.

Vamos começar com um exemplo mais simples. Seja $x^2 + y^2 = 25$ a circunferência de raio 5 e centro na origem. Encontrar a tangente à circunferência no ponto (3,4). Solução: Derivando ambos os lados em relação a x, fica

$$\frac{d}{dx}(x^2 + y^2) = \frac{d}{dx}(25)$$
$$\frac{d}{dx}(x^2) + \frac{d}{dx}(y^2) = 0$$

Lembrando que y é uma função de x e usando a regra da cadeia, fica

$$\frac{d}{dx}(y^2) = \frac{d}{dy}(y^2)\frac{dy}{dx} = 2y\frac{dy}{dx}$$

e $x + 2y\frac{dy}{dx} = 0$. Isolando dy/dx nesta equação fica

$$\frac{dy}{dx} = -\frac{x}{y}$$

No ponto $x = 3$ e $y = 4$ temos $\frac{dy}{dx} = -\frac{3}{4}$ Uma equação da reta tangente à circunferência em (3,4) é: $x - 4 = -\frac{3}{4}(x - 3)$ ou $3x + 4y = 25$.

Quanto ao Fólio de Descartes, a solução completa está em [Ste14], pág. 190.

6.18 Derivadas de funções logarítmicas

Usar-se-á a derivada implícita para achar a derivada de $y = \log_a x$ e em particular a derivada da função log natural $y = \ln x$. Vamos à demonstração:
Seja $y = \log_a x = y$. Então, $a^y = x$. Derivando esta, implicitamente em relação a x e lembrando que se $y = a^x$ então $y' = a^x.\ln a$, fica $a^y(\ln a)\frac{dy}{dx} = 1$. Daqui se $y = \log_a x$ então $y' = \frac{1}{x \ln a}$. Em particular, se o logaritmo é o neperiano, a derivada fica mais simples, a saber: se $y = \ln x$ então $y' = \frac{1}{x}$.

6.19 Número e como um limite

Da derivada acima, sabe-se que se $f(x) = \ln x$ então $f'(x) = 1/x$. Daqui $f'(1) = 1$. Como sabemos a derivada é o limite
$f'(1) = \lim_{h\to 0} \dfrac{f(1+h) - f(1)}{h}$ = e trocando h por x fica:
$f'(1) = \lim_{x\to 0} \dfrac{f(1+x) - f(1)}{x}$ = e considerando que $f(x) = \ln x$ fica
$f'(1) = \lim_{x\to 0} \dfrac{\ln(1+x) - \ln(1)}{x}$ = e como $\ln 1 = 0$ fica
$f'(1) = \lim_{x\to 0} \dfrac{1}{x}\ln(1+x)$ e passando o x para cima (propriedade dos logaritmos)

$f'(1) = \lim_{x\to 0} \ln(1+x)^{1/x}$ e como $f'(1) = 1$ e precisamos lembrar um teorema que diz

Se f é contínua e $\lim_{x\to a} g(x) = b$ então $\lim_{x\to a} f(g(x)) = f(b)$ ou dizendo melhor

$$\lim_{x\to a} f(g(x)) = f\left(\lim_{x\to a} g(x)\right)$$

Agora, o duplo mortal carpado:

$$e = e^1 = e^{\lim_{x\to 0} \ln(1+x)^{1/x}} = \lim_{x\to 0} e^{\ln(1+x)^{1/x}} = \lim_{x\to 0} (1+x)^{1/x} = e$$

Veja-se o cálculo disto em APLX:

```
      ∇r←ee x
[1]   r←(1+x)*÷x∇
      aa←0.1 0.01 0.001 0.0001 0.00001 0.000001 0.0000001 0.00000001 0.000000001
      15 9⍕(9 1⍴aa),9 1⍴ee aa
     0.100000000    2.593742460
     0.010000000    2.704813829
     0.001000000    2.716923932
     0.000100000    2.718145927
     0.000010000    2.718268237
     0.000001000    2.718280469
     0.000000100    2.718281694
     0.000000010    2.718281798
     0.000000001    2.718282052
```

Se colocarmos $n = 1/x$ então $n \to \infty$ quando $x \to 0^+$ e uma expressão alternativa para e é

$$e = \lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n$$

6.20 Em resumo

função $f(x)$	derivada $f'(x)$
c	0
x	1
$x^n, (x \in \mathbb{R})$	nx^{n-1}
$c.f(x)$	$c.f'(x)$
$f+g$	$f'+g'$
e^x	e^x
$(fg)(x)$	$fg' + gf'$
$\frac{n}{d}(x)$	$\frac{dn' - nd'}{d^2}$ n=numerador e d=denominador
$sen(x)$	$cos(x)$
$cos(x)$	$-sen(x)$
$tan(x)$	$sec^2(x)$
$f \circ g$	$f'.g'$
$\log_a x$	$\frac{1}{x \ln a}$
$\ln x$	$1/x$

Capítulo 7

Derivadas Relacionadas

Anteriormente vimos a taxa de mudança de uma função. Não é incomum um problema envolver diversas taxas de mudança. Problemas de taxas relacionadas envolvem relações entre diversas variáveis e como a mudança de uma taxa influencia as demais. A ferramenta básica aqui é a chamada **regra da cadeia**. Como muitos problemas lidam com o tempo como variável independente pode-se afirmar esta regra em termos de t

$$\frac{dy}{dt} = \frac{dy}{dx}.\frac{dx}{dt}$$

A seguir, algumas implicações desta regra

Exemplo 1 Calcule a derivada de x^3 em relação a t.

A derivada de x^3 em relação a x é $3x^2$, entretanto a derivada em relação a t é outra coisa. Para entender, façamos $y = x^3$ e daqui

$$\frac{dy}{dt} = \frac{dy}{dx}.\frac{dx}{dt} = 3x^2\frac{dx}{dt}$$

A menos que se conheça a relação entre x e t não se pode simplificar o resultado. Generalizando este resultado

$$\frac{d}{dt}f(x) = f'(x)\frac{dx}{dt}$$

Obviamente a variável nem sempre vai ser x, mas quanto a t é o mais comum.

Exemplos 2

$$\frac{d}{dt}(w^5) = 5w^4\frac{dw}{dt} \qquad \frac{d}{dt}\left(\frac{1}{y}\right) = -\frac{1}{y^2}\frac{dy}{dt} \qquad \frac{d}{dt}(\sqrt{z}) = \frac{1}{2\sqrt{z}}\frac{dz}{dt}$$

Exemplo 3 Se $y = x^3 + 4x^2$ e $\dfrac{dx}{dt} = 3$, calcule $\dfrac{dy}{dt}$ quando $x = 1$

$$\frac{dy}{dt} = \frac{d}{dt}(x^3 + 4x^2) = (3x^2 + 8x)\frac{dx}{dt}$$

Já que $\dfrac{dx}{dt} = 3$,

$$\frac{dy}{dt} = (3x^2 + 8x).3$$

Como $\dfrac{dx}{dt}$ é constante, $\dfrac{dy}{dt}$ varia conforme x varia. Quando $x = 1$

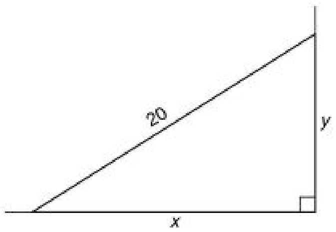
$$\frac{dy}{dt} = 11.3 = 33$$

Na solução de problemas textuais, os seguintes passos devem ser seguidos

1. Desenhe um diagrama (se for o caso). Rotule as variáveis com um símbolo apropriado. Identifique constantes com seus valores numéricos.
2. Determine as variações que são dadas e as que você precisará calcular.
3. Ache a ou as equações relacionadas as variáveis definidas em 1.
4. Diferencie as equações da etapa 3 em relação ao tempo.
5. Substitua todas as informações da etapa 4 e ache a variação desconhecida. Insira as unidades apropriadas

Exemplo 4 Uma escada de 20 m é colocada contra um muro. O pé da escada começa a deslizar se afastando do muro à razão de 1 m/segundo. Quão rápido o topo da escada desliza para baixo no muro quando o pé da escada está a 12 m do muro ?

Passo 1



Passo 2 Dado $\frac{dx}{dt} = 1$, lembrando que como x está aumentando dx/dt é positivo, ainda que o pé da escada esteja se movendo para a esquerda. Quem tiver dificuldade com isso, é só virar o desenho.
Achar $\frac{dy}{dt}$ quando $x = 12$

Etapa 3 Observando o triângulo retângulo, usa-se Pitágoras para obter a relação entre x e y.

$$x^2 + y^2 = 20^2$$

Etapa 4 Diferenciando em relação a t:

$$\frac{d}{dt}x^2 + \frac{d}{dt}y^2 = \frac{d}{dt}400$$

$$2x\frac{dx}{dt} + 2y\frac{dy}{dt} = 0 \qquad \textit{dividindo por } 2$$

$$x\frac{dx}{dt} + y\frac{dy}{dt} = 0$$

Etapa 5 Sabemos da etapa 2 que $dx/dt = 1$ e também sabemos que $x = 12$ no instante em questão. Antes de calcular dy/dt precisamos saber o valor de y nesse instante. Isto é feito usando a equação obtida em 3

$$x^2 + y^2 = 20^2$$

$$12^2 + y^2 = 20^2...$$

$$y = 16$$

Substituindo na equação obtida no passo 4, fica

$$12.1 + 16.\frac{dy}{dt} = 0$$

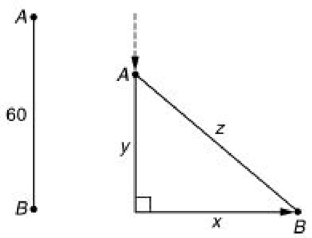
$$16\frac{dy}{dt} = -12$$

$$\frac{dy}{dt} = -\frac{3}{4}$$

A derivada negativa significa que y está diminuindo. O topo da escada está caindo à razão de $\frac{3}{4}$ m por segundo.

Exemplo 5 Em um certo instante o carro A está a 60 milhas ao norte do carro B. A viaja em direção sul a 20 milhas/hora, enquanto B viaja a este a 30 milhas/hora. Quão rápido a distância entre eles variou 1 hora mais tarde ?

Passo 1



Passo 2 Dado $\frac{dx}{dt} = 30$ e $\frac{dy}{dt} = -20$, lembrando que dy/dt é negativo porque y está diminuindo. Se este valor for tomado como 20, a resposta será incorreta.
Deve-se achar $\frac{dz}{dt}$ 1 hora mais tarde.

Etapa 3

$$z^2 = x^2 + y^2$$

Etapa 4 Diferenciando em relação a t:

$$2z\frac{dz}{dt} = 2x\frac{dx}{dt} + 2y\frac{dy}{dt}$$
$$z\frac{dz}{dt} = x\frac{dx}{dt} + y\frac{dy}{dt}$$

Etapa 5 Após 1 hora, $x = 30$ milhas e $y = 60 - 20 = 40$ milhas, lembrando que *distância* = *velocidade* × *tempo*.

$$z = \sqrt{x^2 + y^2} = 50 \ milhas$$

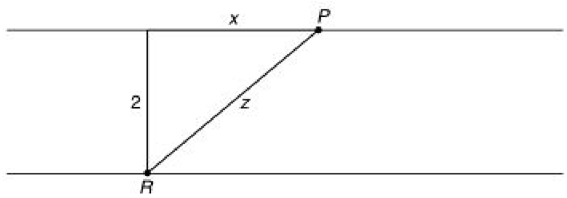
Da etapa 4:

$$50\frac{dz}{dt} = 30.30 + 40(-20)$$
$$50\frac{dz}{dt} = 100$$
$$\frac{dz}{dt} = 2$$

A distância entre os carros está aumentando à razão de 2 milhas/hora.

Exemplo 6 Um avião P viaja horizontalmente em uma altitude de 2 milhas com uma velocidade de 480 $milhas/hora$. Em um certo momento ele passa diretamente sobre uma estação de radar R. Quão rápido a distância entre o avião e a estação aumenta 1 minuto mais tarde ?

Passo 1



Passo 2 Dado $\frac{dx}{dt} = 480$, achar $\frac{dz}{dt}$ 1/60 horas mais tarde. Não se esqueça que as unidades precisam ser consistentes dentro do problema.

Passo 3 Pelo teorema de Pitágoras, $z^2 = x^2 + 2^2$.

Passo 4

$$2z\frac{dz}{dt} = 2x\frac{dx}{dt} + 0$$
$$z\frac{dz}{dt} = x\frac{dx}{dt}$$

Passo 5 Já que o avião viaja a 480 milhas por hora, ele deverá fazer 8 milhas em $\frac{1}{60}$ de hora $\left(480 \times \frac{1}{60} = 8\right)$. Já que $x = 8$, o valor de z é facilmente determinado por Pitágoras

$$z^2 = x^2 + 2^2 = 64 + 4 = 68$$

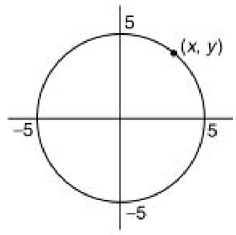
e $z = \sqrt{68} = 2\sqrt{17}$.

Do passo 4, $2\sqrt{17}\frac{dz}{dt} = 8 \times 480$ e

$$\frac{dz}{dt} = \frac{1920}{\sqrt{17}} \ milhas/hora$$

Exemplo 7 um ponto está se movendo ao longo do círculo $x^2+y^2 = 25$ no primeiro quadrante, de tal modo a coordenada x à taxa de 2 cm/seg. Quão rápido a coordenada y muda quando o ponto passa pelo ponto $(3, 4)$?

Passo 1



Passo 2 Dado $\dfrac{dx}{dt} = 2$, ache $\dfrac{dy}{dt}$ quando $x = 3$ e $x = 4$

Passo 3 Dado que o ponto deve estar no círculo, a relação entre x e y deve ser $x^2 + y^2 = 25$

Passo 4

$$2x\frac{dx}{dt} + 2y\frac{dy}{dt} = 0$$

$$x\frac{dx}{dt} + y\frac{dy}{dt} = 0$$

Passo 5

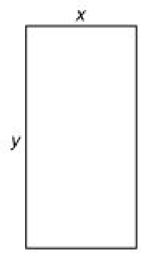
$$3.2 + \frac{dy}{dt} = 0$$

$$\frac{dy}{dt} = -\frac{3}{2}\ cm/seg$$

Já que dx/dt é positivo e o ponto está no primeiro quadrante o ponto está se movendo no circulo no sentido horário. Faz sentido assim que dy/dt seja negativo com o ponto se movendo para baixo.

Exemplo 8 As dimensões de um retângulo estão continuamente se modificando. A largura cresce a 3 polegadas/segundo enquanto o comprimento diminui à taxa de 2 polegadas/segundo. Em um instante o retângulo é um quadrado que tem uma área de 20 polegadas quadradas. Quão rápido está mudando sua área 3 segundos mais tarde ? A área está aumentando ou diminuindo ?

Passo 1



Passo 2 Dados $\dfrac{dx}{dt} = 3$, $\dfrac{dy}{dt} = -2$, achar $\dfrac{dA}{dt}$ 3 segundos mais tarde.

Passo 3

$$A = xy$$

Passo 4 Usar-se-á a regra do produto para calcular $\dfrac{dA}{dt}$. Todas as derivadas serão em relação a t. [1]

$$\frac{dA}{dt} = x\frac{dy}{dt} + y\frac{dx}{dt}$$

[1] A regra do produto (também conhecida como Lei de Leibniz) diz que *a derivada de um produto de duas funções é a primeira vezes a derivada da segunda mais a segunda vezes a derivada da primeira* ou em símbolos $(fg)' = f'g + fg'$

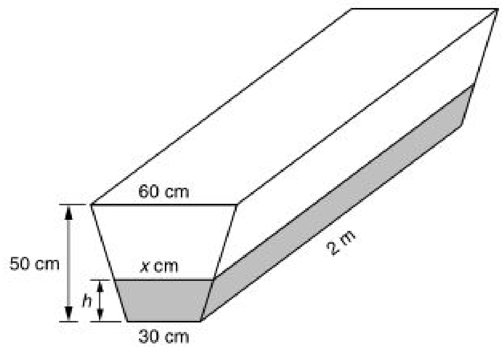
Passo 5 3 segundos depois, $x = 20 + 3.3 = 29$ e $y = 20 + 3(02) = 14$

$$\frac{dA}{dt} = 29.(-2) + 14.3 = -16$$

Já que $\dfrac{dA}{dt}$ é negativa, a área está diminuindo à taxa de 16 pol^2/seg.

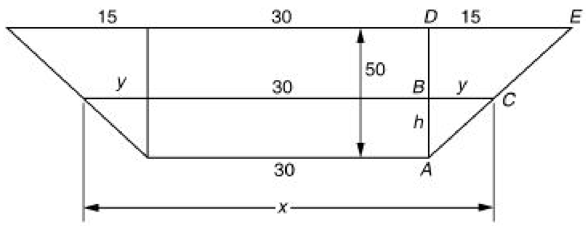
Exemplo 9 Um cocho preenchido com água tem 2 m de comprimento e uma sessão na forma de um trapezóide isósceles com 30 cm de largura no chão e 60 cm no alto, além de uma altura de 50 cm. Se o cocho perde água à taxa de 2000 cm^3/min, quão rápido o nível da água diminui quando a água tem profundidade de 20cm ?

Passo 1



Passo 2 Seja V o volume de água no tanque. Dado $\dfrac{dV}{dt} = -2000$, achar $\dfrac{dh}{dt}$ quando $h = 20$

Passo 3 Lembrando que a área de um trapezóide é $\dfrac{h}{2}(a+b)$, sabe-se que a área da sessão do cocho é $A = \dfrac{h}{2}(x+30)$.O volume de água é $V = l \times A$. l é o comprimento do cocho, então $V = l.\dfrac{h}{2}(x+30)$. Já que as unidades devem ser consistentes, então $l = 200cm$, daqui $V = 100.h(x + 30)$. Como x não é mencionada na pergunta, deve-se eliminá-lo da equação. Para fazer isso, observe os triângulos similares



Na figura, $\triangle ABC$ é similar a $\triangle ADE$. Logo seus lados correspondentes são proporcionais

$$\frac{y}{15} = \frac{h}{50}$$

ou $y = \dfrac{3}{10}h$. Como $x = 2y + 30$, segue-se que $x = \dfrac{3}{5}h + 30$ e

$$V = 100.h(x + 30) = 100h\left(\frac{3}{5}h + 60\right)$$

e $V = 60h^2 + 6000h$

Passo 4

$$\frac{dV}{dt} = (120h + 6000)\frac{dh}{dt}$$

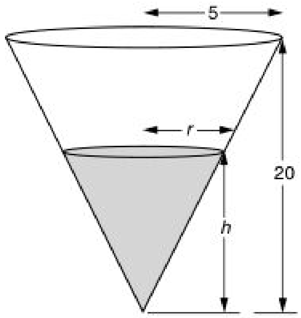
Passo 5 Substituindo $\dfrac{dV}{dt} = -2000$ e $h = 20$ no resultado da etapa 4

$$-2000 = 8400\frac{dh}{dt}$$

e daqui $\dfrac{dh}{dt} = -\dfrac{2000}{8400} = -\dfrac{5}{21}$ cm/min com o sinal negativo informando que a água está saindo.

Exemplo 10 Água é bombeada em um tanque cônico à taxa de 100 $pés^3/minuto$. A altura do tanque é de 20 pés e seu raio é 5 pés. Quão rápida é a mudança do nível de água quando a altura da água é de 10 pés ?

Passo 1

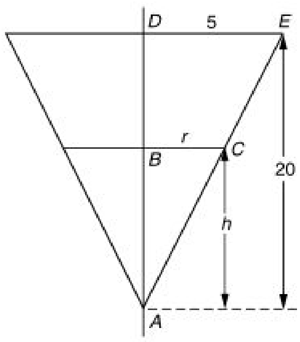


Passo 2 Seja V o volume de água no tanque. O volume está aumentando à razão de 100 pés ao cubo por minuto. Então, dado $\frac{dV}{dt} = 100$, achar $\frac{dh}{dt}$ quando $h = 10$

Passo 3 A relação entre V, r e h é dada pela fórmula do volume de um cone

$$V = \frac{\pi}{3} r^2 h$$

Já que não se tem informação sobre r ou dr/dt, é melhor eliminar r desta equação. Isto pode ser feito observando o cone de uma perspectiva bidimensional



Observando que o $\triangle ABC$ é similar a $\triangle ADE$ pode-se escrever

$$\frac{h}{r} = \frac{20}{5}$$

e $20r = 5h$ e finalmente $r = \frac{h}{4}$. Trocando r na fórmula do volume pode-se obter V como função de h

$$V = \frac{\pi}{3} \left(\frac{h}{4} \right)^2 .h$$

$$V = \frac{\pi}{48} h^3$$

Passo 4

$$\frac{dV}{dt} = \frac{3\pi}{48} h^2 \frac{dh}{dt}$$

$$= \frac{\pi}{16} h^2 \frac{dh}{dt}$$

Passo 5

$$100 = \frac{\pi}{16}(10)^2\frac{dh}{dt}$$
$$100 = \frac{100\pi}{16}.\frac{dh}{dt}$$
$$1600 = 100\pi\frac{dh}{dt}$$
$$\frac{dh}{dt} = \frac{16}{\pi} \approx 5.092$$

e a água está subindo à razão de $16/\pi$ pés por minuto

7.1 Problemas suplementares

1. Uma bola de neve esférica está derretendo de modo que sua superfície diminui à taxa de 1 $polegada^2/minuto$
   (a) Quão rápido seu raio diminui quando ele é de 3 polegadas ?
   (b) Quão rápido seu volume diminui quando o raio é de 3 polegadas ?

2. Dois carros começam uma viagem a partir do mesmo ponto P. Se o carro A viaja para o norte com a taxa de 30 milhas/hora e o carro B viaja para o oeste com a taxa de 40 milhas/hora, quão rápido a distância entre eles varia 2 horas mais tarde ?

3. O barco A está a 70 km a oeste do barco B e está navegando para o sul com a velocidade de 25 km/h. O barco B está navegando norte com a velocidade de 45 km/h. Quão rápido a distância entre os 2 barcos muda após 2 horas ?

4. Um diamante de baseball é um quadrado cujos lados têm 90 pés. Se um batedor acerta a bola e corre para a primeira base à velocidade de 20 pés/seg, quão rápido da segunda base muda quando ele já correu 50 pés ?

5. Os dois lados de um triângulo retângulo têm cada 70cm. Se um lado cresce à taxa de 5cm/min e o outro diminui com a mesma taxa
   (a) quão rápido a hipotenusa varia 2 minutos após ?
   (b) quão rápido a área do triângulo varia 2 minutos após ?

6. Um pescador pegou um peixe que está no final da linha que está sendo enrolada à taxa de 2 pés/seg sobre uma ponte a 30 pés acima da água. A que velocidade o peixe está se movendo sobre a água quando a quantidade de linha é de 50 pés ? Assuma que o peixe está na superfície da água e que não há folga na linha.

7. Areia está sendo descarregada de uma esteira à taxa de 10 pés cúbicos por minuto e forma uma pilha similar a um cone cuja altura é sempre metade do seu raio. Quão rápido sua altura varia quando a pilha tem 5 pés de altura ?

8. Uma estação de radar está a 2000 pés do local de lançamento de um foguete. Se o foguete é lançado verticalmente à taxa de 500 pés por segundo, quão rápido varia a distância entre o foguete e a estação 10 segundos após o lançamento ?

A solução destes problemas está no material, págs 60 a 69.

Capítulo 8

Aplicações

8.1 Problemas de otimização

Trata-se de encontrar a melhor maneira (também chamada de ótima) de fazer alguma coisa, por exemplo

- qual a forma de uma lata que otimiza a logística ? (menos aço e mais conteúdo)
- qual a aceleração máxima de um foguete (interessa aos tripulantes...)
- qual o raio de uma traquéia contraída que expele mais rapidamente o ar durante a tosse ?
- qual o ângulo pelo qual os vasos sanguíneos devem se ramificar para que a energia despendida pelo coração na circulação seja mínimo ?

Estes problemas se reduzem a encontrar valores máximos ou mínimos em uma função de x. Daqui a definição

Seja c um número no domínio D de uma função f. Então $f(c)$ é

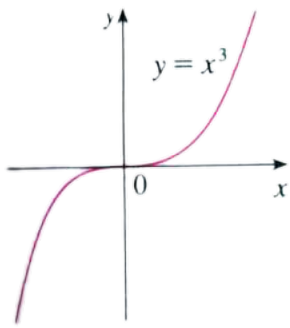
máximo global de f em D sse $f(c) \geq f(x), \forall x \in D$

mínimo global de f em D sse $f(c) \leq f(x), \forall x \in D$

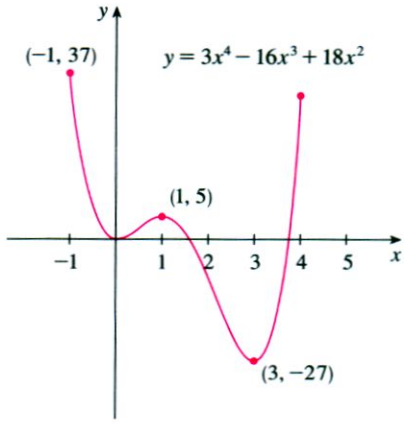
máximo local de f sse $f(c) \geq f(x)$ quando x é próximo a c.

mínimo local de f sse $f(c) \leq f(x)$ quando x é próximo a c.

Quando se diz que algo é verdadeiro se um valor x é próximo a c, significa que ele é verdadeiro em algum intervalo aberto contendo c. Exemplos: a função $y = cos(x)$ assume seu valor máximo local e global em 1, infinitas vezes, uma vez que $cos(2k\pi) = 1$ para todo inteiro k. A função $y = f(x^2)$ informa que $f(x) \geq 0$ pois $x^2 \geq 0$ qualquer que seja o x. Consequentemente 0 é o valor de mínimo absoluto. Esta função não tem máximos. Já a função $y = f(x^3)$ não tem máximo nem mínimo, nem local nem global. Veja seu gráfico



Já o gráfico da função $f(x) = 3x^4 - 16x^3 + 18x^2$ com $-1 \leq x \leq 4$ está mostrado em



Pode-se ver que $f(1) = 5$ é um máximo local, enquanto o máximo absoluto é $f(-1) = 37$ que a propósito não é máximo local pois ocorre em um extremo do intervalo. Alem disso $f(0) = 0$ é mínimo local e $f(3) = -27$ é mínimo tanto local quanto absoluto. Observe que $f(4)$ não é máximo nem local nem absoluto.

É hora de dar a palavra a Fermat

> Teorema de Fermat: Se f tem um máximo ou mínimo local em c e se $f'(c)$ existe, então $f'(c) = 0$. A demostração completa está em [Ste14] e em resumo ela mostra que se existe máximo ou mínimo $f'(c) \leq 0$ e também $f'(c) \geq 0$. Para que as duas condições sejam verdadeiras (e são) a conclusão é que $f'(c) = 0$.

Cuidado que este teorema só vai em um sentido. Ou seja, $f'(x)$ pode ser 0, sem que haja ponto de máximo ou de mínimo. Eis um exemplo simplório: Seja $f(x) = x^3$. Aqui, $f'(0) = 0$ e como se viu no gráfico acima, não há ponto de máximo ou de mínimo no potno $x = 0$.

8.2 O que as derivadas dizem de uma função

Começamos definindo um número crítico de uma função f é um número c no domínio de f tal que $f'(c) = 0$ ou $f'(c)$ não existe. Além disso, se $f'(x) > 0$ em um intervalo, então f é crescente neste intervalo. E, se $f'(x) < 0$ em um intervalo, então f é decrescente nele.

Ainda, se c é um número crítico de função contínua f.

(a) Se o sinal de $f'(c)$ muda de positivo para negativo em c, então $f(c)$ é máximo local.

(b) Se o sinal de $f'(c)$ muda de negativo para positivo em c , então $f(c)$ é mínimo local.

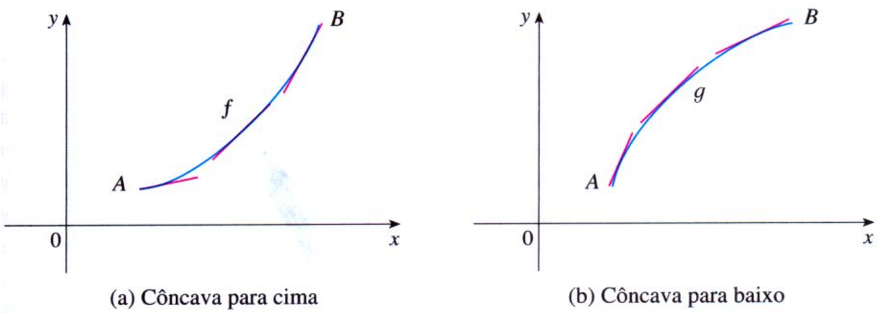
(c) Se $f'(c)$ não muda de sinal em c, não há máximo ou mínimo local em c

A derivada segunda nos fala da concavidade da função.

(a) Se $f''(x) > 0 \forall x \in I$, então o gráfico de f é côncavo para cima em I.

(b) Se $f''(x) < 0 \forall x \in I$, então o gráfico de f é côncavo para baixo em I.

Veja estes conceitos nas figuras a seguir



(a) Côncava para cima (b) Côncava para baixo

Pondo em palavras, se o gráfico de f está acima de todas as suas tangentes no intervalo I então se diz que f é côncava para cima em I. Se o gráfico de f está abaixo de todas as tangentes em I, então f é côncava para baixo em I.

Ainda, um ponto P na curva $y = f(x)$ é chamado ponto de inflexão se f é contínua no ponto e a curva muda de concavidade no ponto.

Ainda temos o teste da segunda derivada: Supondo que f'' é contínua nas proximidades de c

(a) Se $f'(c) = 0$ e $f''(c) > 0$ então $f(c)$ é mínimo local.

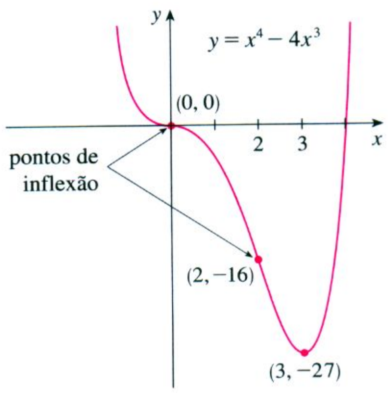
(b) Se $f'(c) = 0$ e $f''(c) < 0$ então $f(c)$ é máximo local.

Veja-se um exemplo: Deve-se examinar a curva $y = x^4 - 4x^3$ em relação a concavidade, pontos de inflexãoe mínimos e máximos locais. Então $f'(x) = 4x^3 - 12x^2 = 4x^2(x-3)$ e $f''(x) = 12x^2 - 24x = 12x(x-2)$. Para encontrar os números críticos faz-se $f'(x) = 0$ e obtém-se $x = 0$ e $x = 3$. Para o teste da segunda derivada nestes pontos acha-se $f''(0) = 0$ e $f''(3) = 36 > 0$. Uma vez que $f'(3) = 0$ e $f''(3) > 0$ então $f(3) = -27$ é um mínimo local.

Como $f''(0) = 0$ o teste da segunda derivada nada fala sobre o número crítico 0. Mas como $f'(x) < 0$ para $x < 0$ e também para $0 < x < 3$ o teste da primeira derivada nos diz que não há máximo ou mínimo locais em 0. Como $f''(x) = 0$ quando $x = 0$ ou $x = 2$ contrói-se a seguinte tabela

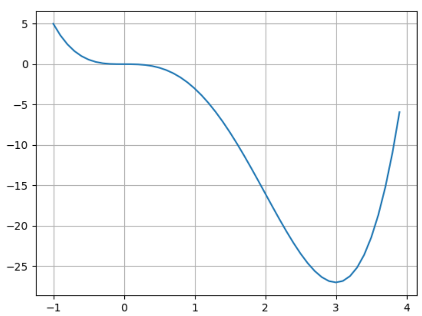
Intervalo	$f''(x) = 12x(x-2)$	concavidade
$(-\infty, 0)$	$+$	para cima
$(0, 2)$	-	para baixo
$(2, \infty)$	$+$	para cima

O ponto $(0, 0)$ e o ponto $(2, -16)$ são pontos de inflexão. Eis como ficou a curva



A propósito, eis o python para gerar este gráfico A mesma imagem capturada acima pode ser facilmente produzida em Python como se verá a seguir. Eventuais melhoramentos podem ser pedidos facilmente. Aqui só está o básico.

Veja



```
import matplotlib.pyplot as plt
import numpy as np
def fu(x):
  return x**4-4*x**3
x=np.arange(-1,4,0.1)
y=fu(x)
il=plt.plot(x,y)
plt.grid()
plt.show()
```

8.3 Formas indeterminadas e a regra de L'Hospital

Suponha precisar analisar o comportamento da função $F(x) = \dfrac{\ln x}{x-1}$ no ponto $x = 1$. Em particular quer-se examinar o valor do limite $\lim\limits_{x\to 1} \dfrac{\ln x}{x-1}$. Não se pode aplicar a regra dos limites (o limite de um quociente é o quociente dos limites) pois o limite do denominador é 0. Embora o limite exista, ele não é óbvio pois tanto o numerador como o denominador tendem a 0. E, certamente $\frac{0}{0}$ não está definido. Da mesma maneira, quando se calcula $\lim\limits_{x\to\infty} \dfrac{\ln x}{x-1}$. Aqui o numerador e o denominador disputam uma corrida. Se o numerador ganhar o resultados será ∞ e se o denominador ganhar o resultado será 0. Ou, pode haver agum equilíbrio e neste caso a resposta será algum positivo finito. Daqui vem a **regra de L'Hospital**:

Se, no cálculo de limites, houver indeterminações do tipo $0/0$ ou ∞/∞ então

$$\lim_{x\to a} \frac{f(x)}{g(x)} = \lim_{x\to a} \frac{f'(x)}{g'(x)}$$

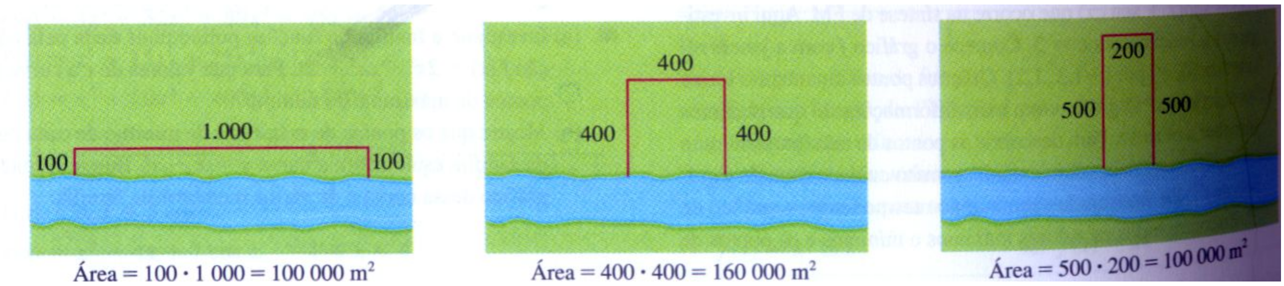
Se, ao contrário houver uma corrida como descrito acima, tem-se um limite indeterminado do tipo $0.\infty$ Lida-se com ela escrevendo o produto fg como um quociente do tipo $fg = \frac{1}{1/g}$ e agora pode-se usar a regra de L'Hospital.

Uma curiosidade aqui é que a regra de L'Hospital não foi descoberta por L'Hospital. Foi por Bernouilli. Só que anos antes da descoberta, o Marques de L'Hospital havia firmado um contrato com Bernouilli, pelo qual, mediante um pagamento em dinheiro, todas as descobertas deste seriam publicadas por aquele. É por isso que a regra se chama assim.

8.4 Problemas de otimização

Seja o caso: Um fazendeiro tem 1.200m de cerca e quer cercar um campo retangular que está na margem de um rio reto. Não há necessidade de cerca ao longo do rio. Quais as dimensões do campo que tem maior área ?

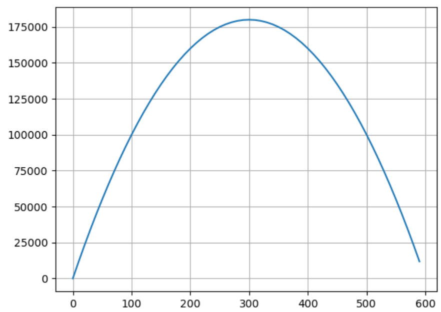
Solução: uma boa estratégia é estudar alguns casos especiais, veja:



Depois de estudar o jeitão do problema vai-se à calculejação. Supondo que a área do terreno é $a = xy$ (x perpendicular ao rio e y paralelo a ele). Seria bom expressar A apenas por uma variável, assim precisa-se eliminar o y. Usa-se para isso a informação de existem 1.200 m de cerca. Daqui: $2x + y = 1200$ e portanto $y = 1200 - 2x$. Agora volta-se à fórmula da área e $A = xy = x(1200 - 2x) = 1200x - 2x^2$. Estudando o domínio desta função, ve-se que $x \geq 0$ e que $x \leq 600$. De outra forma, a área seria negativa, o que não tem sentido. Logo, deseja-se maximizar $A(x) = 1200x - 2x^2$ sujeito a $0 \leq x \leq 600$. A derivada de A em relação a x é $A'(x) = 1200 - 4x$. Logo para encontrar os números críticos resolve-se $1200 - 4x = 0$ e acha-se o valor $x = 300$. O valor máximo deve ocorrer neste ponto ou nos extremos do domínio. Como $A(0) = 0$ e $A(600) = 0$ só sobra o $x = 300$ e $A(300) = 180.000$. Alternativamente poder-se-ia olhar que $A''(x) = -4$ e que portanto $A''(x) < 0/\forall/x$, logo A é sempre côncava para baixo e o máximo local em $x = 300$ é máximo global.

Resolvido o problema.

Veja




```
import matplotlib.pyplot as plt
import numpy as np
def fu(x):
  return 1200*x-2*x**2
x=np.arange(0,600,10)
y=fu(x)
il=plt.plot(x,y)
plt.grid()
plt.show()
```

Mais um exemplo: Uma lata é feita para armazenar 1 litro de óleo. Encontre as dimensões que minimizam o gasto de material.

Da geometria, a área total da lata é base (πr^2), tampa (πr^2) e lateral ($2\pi r \times h$), dando uma área total de $A = 2\pi r^2 + 2\pi r h$. Para eliminar o h, usa-se a informação de que o volume da lata é 1 litro que é igual a 1000 cm^3, já que a nossa unidade no problema será o cm. O volume de uma lata é $V = \pi r^2 h = 1000$, o que nos dá $h = \frac{1000}{\pi r^2}$. Substituindo este h na expressão da área fica:

$$A = 2\pi r^2 + 2\pi r \left(\frac{1000}{\pi r^2}\right) = 2\pi r^2 + \frac{2000}{r}$$

A função que se quer otimizar é $A(r) = 2\pi r^2 + \frac{2000}{r} \quad r > 0$

Para achar os números críticos, deriva-se: $A'(r) = 4\pi r - \frac{2000}{r^2} = \frac{4(\pi r^3 - 500)}{r^2}$ e daqui $A'(r) = 0$ quando $\pi r^3 = 500$ logo o número crítico é $\sqrt[3]{500/\pi}$.

O domínio de A é $(0, \infty)$. Veja-se que $A'(r) < 0$ para $r < \sqrt[3]{500/\pi}$ e $A'(r) > 0$ para $r > \sqrt[3]{500/\pi}$. Portanto A está decrescendo para todo r à esquerda do número crítico e crescendo para todo r à direita. Assim, $r = \sqrt[3]{500/\pi}$ deve originar um mínimo absoluto.

O valor de h correspondente a $r = \sqrt[3]{500/\pi}$ é $h = \frac{1000}{\pi r^2} = \frac{1000}{\pi (500/\pi)^{(2/3)}} = 2\sqrt[3]{\frac{500}{\pi}} = 2r$.

Concluindo, para minimizar o custo da lata o raio deve ser $\sqrt[3]{500/\pi} = 5.419260701cm$ e a altura duas vezes isso ou $h = 2r = 10.8385214\ cm$. Para verificar se está certo, retorna-se à fórmula do volume ($V = \pi r^2 h$) e

```
      2×(○1)×r*3
999.9999998
```

Um método alternativo para este mesmo problema é usar a derivação implícita. Começando com as mesmas equações

$$A = 2\pi r^2 + 2\pi rh \qquad \pi r^2 h = 1000$$

ao invés de eliminar h deriva-se implicitamente ambas as equações em relação a r e fica:

$$A' = 4\pi r = 2\pi h + 2\pi rh' \qquad 2\pi rh + \pi r^2 h' = 0$$

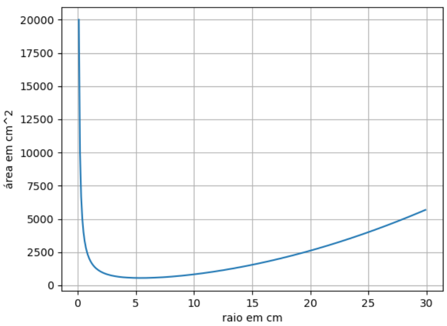
O mínimo ocorre em um número crítico, assim, faz-se $A' = 0$ simplifica-se e chega-se a

$$2r + h + rh' = 0 \qquad 2h + rh' = 0$$

Uma subtração nos fornece $2r - h = 0$ ou $h = 2r$.

O gráfico

Veja




```
import matplotlib.pyplot as plt
import numpy as np
def ar(r):
  return 2*3.14159265*r**2+2000/r
x=np.arange(0,30,0.1)
y=ar(x)
il=plt.plot(x,y)
plt.xlabel('raio em cm')
plt.ylabel('área em cm^2')
plt.grid()
plt.show()
```

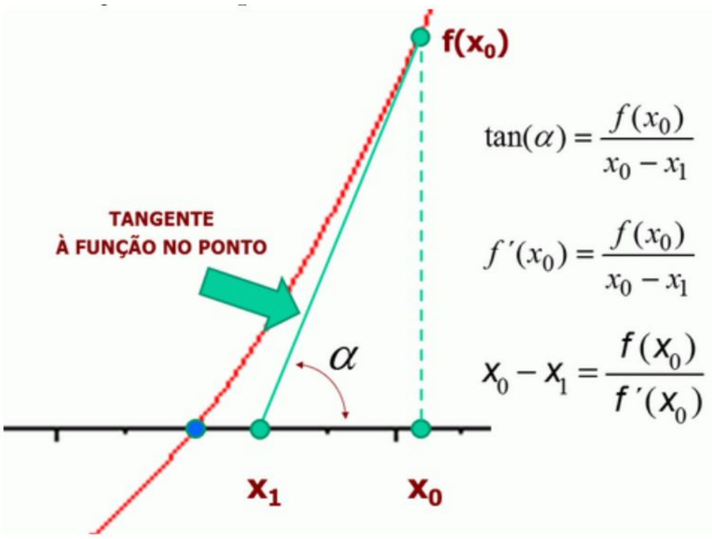
Agora, um desafio: Ache 2 números cuja soma seja 23 e o produto seja máximo. Usando a força bruta:

```
      a←⍳22
      b←23-a
      3 22⍴a,b,a×b
a=  1  2  3  4  5   6   7   8   9  10  11  12  13  14  15  16  17 18 19 20 21 22
b= 22 21 20 19 18  17  16  15  14  13  12  11  10   9   8   7   6  5  4  3  2  1
p= 22 42 60 76 90 102 112 120 126 130 132 132 130 126 120 112 102 90 76 60 42 22
```

Solução elegante pelo cálculo:
$a+b = 23$ e $P = ab$. Da primeira $b = 23-a$ e deve-se maximizar a segunda. De ambas, fica $P = a(23-a)$ ou $P = 23a-a^2$. derivando esta em relação a a, fica $P'(a) = 23 - 2a$ Como se quer os números críticos, iguala-se a derivada a zero e fica $P'(a) = 0 = 23 - 2a$ e daqui $a = 11.5$ com $b = 11.5$. Este resultado precisa ser adaptado ao problema e ele sugere duas possíveis respostas: $a = 11$ e $b = 12$ ou $a = 12$ e $b = 11$ que como se viu na força bruta é a resposta procurada.

8.5 Método de Newton-Raphson

Imagine precisar calcular a $\sqrt[20]{17890}$ que pode ser traduzido para $f(x) = x^{20} = 17890$ ou ainda $f(x) = x^{20} - 17890 = 0$ que é a forma que precisamos. Se pudermos contar com um valor inicial aproximado ao resultado poderemos refiná-lo usando um método devido a Newton e chegar ao valor desejado com a precisão que for requerida. Examinando um possível gráfico da função, e lembrando que a derivada em um ponto é a tangente da secante naquele ponto, tem-se



Neste método x_0 é a aproximação inicial. Com ela, chega-se à raiz x_1 e com esta a x_2 e de um modo geral, pode-se escrever $x_n = f(x_{n-1})$. As etapas do método são

1. Obter a derivada da função a resolver
2. Arbitrar uma raiz inicial (x_0)
3. Arbitrar um erro esperado ϵ
4. Calcular $f(x_0)$ e $f'(x_0)$
5. Calcular o novo x que é $x_1 = -\frac{f(x_0)}{f'(x_0)}$
6. Se $|f(x)| > \epsilon$ retornar ao passo 4 acima

Vamos ver o método para calcular algo bem simples, como $\sqrt[20]{17890}$. Cozinhando um pouco os dados pode-se escrever: $f(x) = x^{20} - 17890 = 0$. agora precisa-se algum conhecimento sobre o comportamento desta função, para ajudar na busca da primeira raiz que será aproximada depois. Note que $f(1) = -17889$ e $f(2) = +1030686$, então claramente a raiz procurada está entre 1 e 2, já que a função $f(x)$ mudou de sinal no intervalo. Isto significa que a curva cruzou o eixo x. Qualquer um dos dois valores pode ser usado como raiz inicial. Vai-se usar o valor $x = 2$.

Calculando a derivada $f'(x) = 20x^{19}$, e escrevendo um programa Python bem simples

```
import numpy as np
def pri(x):
  return x**20-17890
def der(x):
  return 20*x**19
x=2
fx=der(x)
epsilo=0.0000001
while abs(fx)>epsilo:
  fx=pri(x)
  dfx=der(x)
  x=x-fx/dfx
  print("{:15.10f} {:16.8f} {:16.8f} ".format(x,fx,dfx))
print('raiz = ',x)
```

e rodando-o obtém-se o seguinte resultado

```
   1.9017061234 1030686.00000000 10485760.00000000
   1.8110656523  364818.49095265 4024896.23662848
   1.7317540351  126216.60306503 1591401.20497090
   1.6714897579   40956.97798589  679622.81931249
   1.6395105138   11088.39953058  346737.38673190
   1.6320099741    1801.74127178  240214.88250073
   1.6316638928      76.19762818  220172.64494105
   1.6316631937       0.15330909  219287.23664102
   1.6316631937       0.00000062  219285.45143173
   1.6316631937      -0.00000000  219285.45142447
raiz =  1.6316631936854427
```

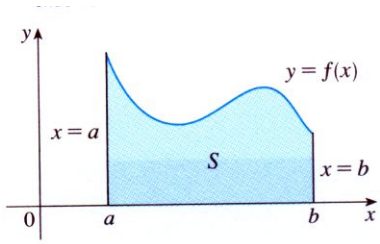
Se o algoritmo tivesse começado com 1, a convergência seria mais lenta, mas funcionaria igual. Na verdade, ele converge para qualquer número real, mas uma má escolha pode levar a *overflow* nos acumuladores do programa. Outra questão nestas soluções puramente numéricas, é que o resultado precisa ser estudado à luz da matemática. Neste caso, a resposta correta é ± 1.6316631936854427. Fica evidente, se a aproximação se der pela esquerda e o algoritmo começar com -1 ao invés de 1, consequência direta de ser um expoente par na equação original. Este método é poderoso, não importa a complexidade da equação a resolver. Basta que ela tenha derivada.

Capítulo 9

Integrais

9.1 Cálculo de áreas e distâncias

Seja uma questão trivial: qual a área da região S que está sob a curva $f(x)$ de a até b. Olhando o gráfico percebe-se que a região em questão está limitada pela função acima, pelas verticais em a e b e pelo eixo x.



Considerando conhecida a área de um retângulo ($base \times altura$). Então, vai-se dividir S em retângulos, somando-se-lhes as áreas e depois aumenta-se o número de retângulos.

9.2 Integral definida

Se f é uma função contínua definida em $a \leq x \leq b$ divide-se o intervalo $[a, b]$ em n subintervalos de comprimentos iguais $\Delta x = (b - a)/n$. Sejam $x_0(= a), x_1, x_2, ..., x_n(= b)$ as extremidades desses subintervalos e sejam $x_1^*, x_2^*, ..., x_n^*$ pontos amostrais arbitrários desses subintervalos de forma que x_i^* esteja no i-ésimo subintervalo $[x_{i-1}, x_i]$. Então a integral definida de f de a até b é

$$\int_a^b f(x)dx = \lim_{n \to \infty} \sum_{i=1}^{n} f(x_i^*)\Delta x$$

Esta definição de integral é devida a Riemann, aluno de Carl Gauss.

Se os pontos amostrais forem sempre os da direita. Então $x_i^* = x_i$ e a definição se simplifica

$$\int_a^b f(x)dx = \lim_{n \to \infty} \sum_{i=1}^{n} f(x_i)\Delta x$$

onde $\Delta x = \frac{b-a}{n}$ e $x_i = a + i\Delta x$

9.2.1 Relembrando somatórias

A seguir uma boa lembrada

$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

$\sum_{i=1}^{n} i^3 = \left[\frac{n(n+1)}{2}\right]^2$

$\sum_{i=1}^{n} c = nc$

$\sum_{i=1}^{n} c.a_i = c\sum_{i=1}^{n} a_i$

$\sum_{i=1}^{n}(a_i + b_i) = \sum_{i=1}^{n} a_i + \sum_{i=1}^{n} b_i$

$\sum_{i=1}^{n}(a_i - b_i) = \sum_{i=1}^{n} a_i - \sum_{i=1}^{n} b_i$

9.2.2 Ponto Médio

Frequentemente se escolhe o ponto amostral como x_i^* como a extremidade direita do i-ésimo intervalo. Porém, se o propósito é encontrar uma aproximação para uma integral, geralmente é melhor escolher x_i^* como o ponto médio do intervalo, denotado $\bar{x_i}$. Qualquer soma é uma aproximação para uma integral, mas se usarmos os pontos médios, obtém-se

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} f(\bar{x_i})\Delta x = \Delta x[f(\bar{x_1}) + ... + f(\bar{x_n})]$$

com $\Delta x = \frac{b-a}{n}$

9.3 Propriedades da integral definida

$\displaystyle\int_a^b f(x)dx = -\int_b^a f(x)dx$

$\displaystyle\int_a^a f(x)dx = 0$

$\displaystyle\int_a^b c.dx = c(b-a)$

$\displaystyle\int_a^b [f(x) + g(x)]dx = \int_b^a f(x)dx + \int_a^b g(x)dx$

$\displaystyle\int_a^b c.f(x)dx = c\int_a^b f(x)dx$

9.4 Teorema Fundamental do Cálculo

Estabelece uma conexão entre o cálculo diferencial e o integral e diz que se f for contínua em $[a, b]$ então a função g definida por $g(x) = \int_a^x f(t)dt \quad a \leq x \leq b$ é contínua em $[a, b]$ e derivável em (a, b) e $g'(x) = f(x)$
Em palavras ele afirma que a derivada de uma integral definida com relação ao seu limite superior é o seu integrando calculado no limite superior.

Uma segunda versão deste teorema: se f for contínua em $[a, b]$ então $\int_a^b f(x)dx = F(b) - F(a)$ onde F é qualquer primitiva de f isto é $F' = f$

9.4.1 Integrais indefinidas

Uma notação conveniente para obter primitivas e nela

$$\int f(x)dx = F(x) \qquad \Longrightarrow \qquad F'(x) = f(x)$$

Por exemplo, pode-se escrever $\int x^2 dx = \frac{x^3}{3} + C$ pois $\frac{d}{dx}\left(\frac{x^3}{3} + C\right) = x^2$ Aqui, uma distinção importante entre integral definida e indefinida: a definida é um número enquanto a indefinida é uma função, ou mais propriamente uma família de funções, por causa do C. Daqui temos uma tabela de integrais indefinidas:

$$
\begin{aligned}
\int cf(x)dx &= c\int f(x)dx \\
\int [f(x)+g(x)]dx &= \int f(x)dx + \int g(x)dx \\
\int kdx &= kx + C \\
\int x^n dx &= \frac{x^{n+1}}{n+1} + C \quad (n \neq -1) \\
\int \frac{1}{x}dx &= \ln(x) + C \\
\int e^x dx &= e^x + C \\
\int a^x dx &= \frac{a^x}{\ln a} + C \\
\int sen\ xdx &= -cos\ x + C \\
\int cos\ xdx &= sen\ x + C \\
\int sec^2\ xdx &= tg\ x + C \\
\int cossec^2\ xdx &= -cotg\ x + C \\
\int sec\ xtg\ x &= sec\ x + C \\
\int cossec\ xcotg\ x &= -cossec\ x + C \\
\int \frac{1}{x^2+1}dx &= tg^{-1}x + C \\
\int \frac{1}{\sqrt{1-x^2}}dx &= sen^{-1}x + C \\
\int senh\ xdx &= cosh\ x + C \\
\int cosh\ xdx &= senh\ xC
\end{aligned}
$$

9.5 Regra da substituição

Se $u = g(x)$ for uma função derivável, então $\int f(g(x))g'(x)dx = \int f(u)du$. Exemplo: Encontre $\int x^3 cos(x^4+2)dx$. Fazemos a substituição $u = x^4 + 2$ porque sua diferencial é $du = 4x^3dx$ que à parte do fator constante 4 ocorre na integral. Assim, usando $x^3dx = \frac{1}{4}du$ e a regra da substituição, temos $\int x^3cos(x^4+2))dx = \int cos\ u.\frac{1}{4}du = \frac{1}{4}\int cos\ udu$ e $= \frac{1}{4}sen\ u + C$ e $= \frac{1}{4}sen(x^4+2) + C$ Observe que no estágio final, precisa-se retornar à variável original x.

9.6 Integração por partes

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

Exemplo, ache $\int x.sen\ xdx$ Escolhendo $f(x) = x$ e $g'(x) = sen\ x$ Então, $f'(x) = 1$ e $g(x) = -cos\ x$ pela fórmula cima tem-se $\int xsen\ xdx = f(x)g(x) - \int g(x)f'(x)dx$
$= x(-cosx) - \int(-cosx)dx$
$= -xcos\ x + \int cos\ xdx$
$= -xcos\ x + sen\ x + C$
Uma verificação sempre prudente é derivar a resposta. Se fizer isso deve obter $x.sen\ x$ como era de se esperar.

Usando uma fórmula mais simples

$$\int udv = uv - \int vdu$$

, fica $u = x$, $du = dx$, $dv = sen\ xdx$ e $v = -cos\ x$ fica $\int x.sen\ xdx = \int xsen\ xdx = x(-cos\ x) - \int(-cos\ x)dx$
$= -xcos\ x + \int cos\ xdx$
$- xcos\ x + sen\ x + C$
Mais um: encontre $\int t^2e^tdt$. Escolhendo
$u = t^2$ e $du = 2tdt$
$dv = e^tdt$ e $v = e^t$, fica
$\int t^2e^tdt = t^2e^t - 2\int te^tdt$ A segunda integral é mais simples, mas ainda não é óbvia. Precisa integrar por partes de novo.

Agora $u = t$ e $dv = e^t dt$ então $du = dt$, $v = e^t$ e
$\int te^t dt = te^t - \int e^t dt$
$= te^t - e^t + C$ e voltando fica
$\int t^2 e^t dt = t^2 e^t - 2\int te^t dt$
$= t^2 e^t - 2(te^t - e^t + C)$
$= t^2 e^t - 2te^t + 2e^t + C_1$ onde $C_1 = -2C$

No Maple, pedir para ele calcular a $\int sen^5 x\ cos^2 x dx =$

```
> int(sin(x)^5*cos(x)^2,x);
                  4      3                2       3                 3
    - 1/7 sin(x)  cos(x)  - 4/35 sin(x)  cos(x)  - 8/105 cos(x)
```

Veja mais do maple

```
> int(x*(x^2+5)^8,x);
        18        16       14            12          10           8
  1/18 x   + 5/2 x   + 50 x   + 1750/3 x   + 4375 x   + 21875 x

                   6            4              2
         + 218750/3 x  + 156250 x  + 390625/2 x

> Int(x*(x^2+5)^8,x);
                  /
                 |      2     8
                 |  x (x  + 5)  dx
                 |
                /
```

Perceba a diferença que o Maple faz entre *int* (calcule) e *Int* (reescreva).

9.7 Integração aproximada

Existem 2 situações nas quais é impossíel encontrar o valor exato de uma integral definida. A primeira é quando é difícil ou mesmo impossíve encontrar uma primitiva. Por exemplo é impossível encontrar $\int_0^1 e^{x^2} dx$ e $\int_{-1}^1 \sqrt{1+x^3} dx$. A segunda situação (mais comum) é quando a função é determinada por experimento por meio de leitura de instrumento ou de dados coletados. Neste caso, não há fórmula para a função. Em ambos casos há que se achar valores aproximados para as integrais.

9.7.1 Regra do Trapézio

Como já se viu uma maneira de calcular uma área é pela definição de integral, usando subintervalos e calculando. $\int_a^b f(x)dx = \sum_{i=1}^n f(x_i^*)\Delta x$ onde x_i^* é um ponto qualquer no i-ésimo subintervalo $[x_{i-1}, x_i]$. Se x_i^* for a extremidade esquerda do subintervalo então $x_i^* = x_{i-1}$ e a integral é $\int_a^b f(x)dx \approx L_n = \sum_{i=1}^n f(x_{i-1})\Delta x$

Se escolhermos a extremidade direita então $x_i^* = x_i$ e a integral é $\int_a^b f(x)dx \approx R_n = \sum_{i=1}^n f(x_i)\Delta x$.

A regra do trapézio calcula a média entre estas duas formulações e fica

$$\int_a^b f(x) \approx T_n = \frac{\Delta x}{2}\left[f(x_0) + 2f(x_1) + 2f(x_2) + ... + 2f(x_{n-1}) + f(x_n)\right]$$

onde $\Delta x = (b-a)/n$ e $x_i = a + i.\Delta x$. Daqui obtém-se a primeira função em Python:

```
def f(x):
  return 1/x  # aqui entra a f. a calcular
def inttrap(a,b,ntrap):
  h=(b-a)/ntrap
  area=f(a)
  i=1
  while (i<ntrap):
    a=a+h
    area=area+2*f(a)
    i=i+1
    a=a+h
  area=area+f(a)
  return area*h/2
print(inttrap(3,3.6,6))
```

9.8 Regra de Simpson

Se ao invés de usar retas, usarmos parábolas para aproximar a curva obtém-se

$$\int_a^b f(x)dx \approx S_n = \frac{\Delta x}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2) + ... + 4f(x_{n-1}) + f(x_n)\right]$$

onde n é par e $\Delta x = (b - a)/n$. Note os coeficientes 1,4,2,4,2,4,2,4,2,...4,2,4,1. Eis a função

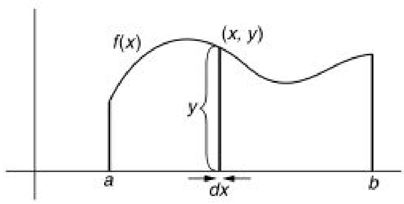
```
def f(x):
  return 1/x # f. a calcular
def intsimpg(a,b,ntrap):
  h=(b-a)/ntrap
  if (0!=ntrap%2):
    print('ntrap não é par, saindo')
    return
  area=f(a)
  i=1
  while i<ntrap:
    a=a+h
    if (0==i%2):
      area=area+2*f(a)
    else:
      area=area+4*f(a)
    i=i+1
    a=a+h
  area=area+f(a)
  area=area*h/3
  return(area)
print(intsimpg(3,3.6,6)) # chamada
```

9.9 Aplicações: problemas de área

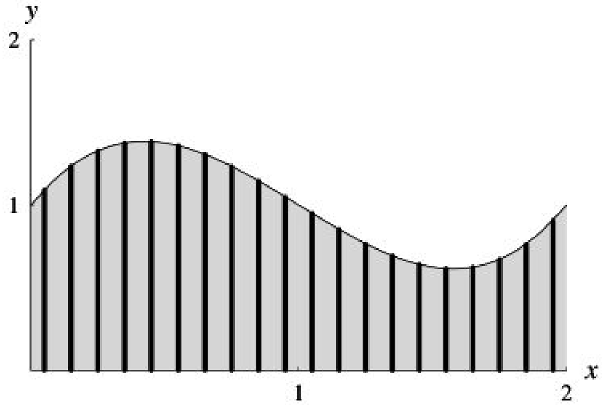
Se $y = f(x)$ é não negativo entre a e b a área da região limitada por seu gráfico e o eixo x entre as linhas verticais $x = a$ e $x = b$ é o valor da integral definida

$$A = \int_a^b f(x)dx$$

Um macete para lembrar desta fórmula é pensar em um retângulo infinitesimalmente estreito cuja altura é $f(x)$ e cuja largura é dx. A área deste retângulo imaginário poderia ser ydx. Se "somarmos" tais áreas retangulares entre a e b pela integração obtem-se $A = \int_a^b f(x)dx$.



Exemplo 1 Ache a área limitada por $y = x^3 - 3x^2 + 2x + 1$, o eixo x e as linhas verticais $x = 0$ e $x = 2$.



O desenho mostra a função y desenhada com 20 retângulos imaginários de altura y e largura dx. Se imaginarmos infinitos de tais retângulos, sua soma dará a área exata da figura.

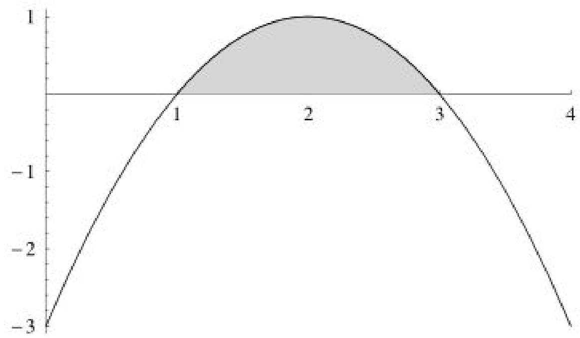
$$\int_0^2 ydx = \int_0^2 (x^3 - 3x^2 + 2x + 1)dx$$

$$= \left[\frac{x^4}{4} - x^3 + x^2 + x\right]_0^2$$

$$= (4 - 8 + 4 + 2) - (0) = 2$$

Aqui os intervalos de integração foram dados. Frequentemente esses intervalos são determinados onde o x intercepta o gráfico. Essas interceptações são obtidas fazendo $f(x) = 0$ e calculando o x correspondente. É sempre salutar fazer um gráfico da função para a estudar.

Exemplo 2 Ache a área da região sobre o eixo x limitada pela função $y = 4x - x^2 - 3$



A intersecção é determinada fazendo $y = 0$ e resolvendo para x

$$y = 4x - x^2 - 3$$

$$0 = 4x - x^2 - 3$$

$$(x - 1)(x - 3) = 0$$

$$x = 1 \qquad x = 3$$

A área pode ser calculada integrando entre 1 e 3

$$A = \int_1^3 ydx$$

$$= \int_1^3 (4x - x^2 - 3)dx$$

$$= \left[2x^2 - \frac{x^3}{3} - 3x\right]_1^3$$

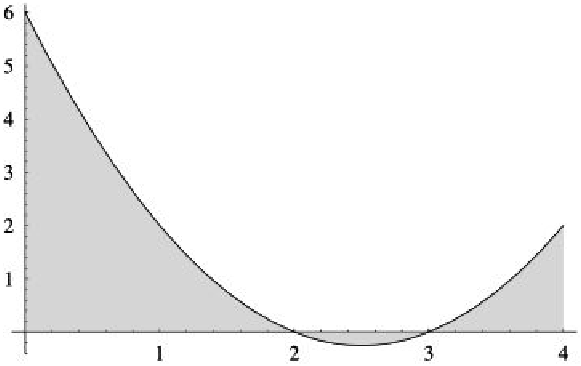
$$= \left(2(3)^2 - \frac{3^3}{3} - 3(3)\right) - \left(2(1)^2 - \frac{1^3}{3} - 3(1)\right)$$

$$= 0 - \left(-\frac{4}{3}\right)$$

$$= \frac{4}{3}$$

Se $f(x)$ é negativa sobre toda ou parte de $[a, b]$ a integral $\int_a^b f(x)dx$ não retornará a área buscada. Ela deve ser obtida fazendo $\int_a^b |f(x)|dx$. Como o Teorema Fundamental do Cálculo não pode ser convenientemente aplicado a uma função envolvendo módulo, primeiro deve ser investigado onde a função cruza o eixo x, quebrando o intervalo $[a, b]$ em sub-intervalos determinados por esses pontos e integrar separadamente cada sub-intervalo e daí somar os valores absolutos destas integrações.

Exemplo 3 Ache a área da região limitada por $y = x^2 - 5x + 6$, o eixo x e as linhas verticais $x = 0$ e $x = 4$.



Vai-se determinar onde a função cruza o eixo x

$$y = x^2 - 5x + 6$$

$$0 = x^2 - 5x + 6$$

$$0 = (x-2)(x-3)$$

$$x = 2 \qquad\quad x = 3$$

Devemos integrar separadamente os 3 intervalor $[0, 2]$, $[2, 3]$ e $[3, 4]$.

$$I_1 = \int_0^2 (x^2 - 5x + 6)dx = \left[\frac{x^3}{3} - \frac{5x^2}{2} + 6x\right]_0^2 = \frac{14}{3}$$

$$I_2 = \int_2^3 (x^2 - 5x + 6)dx = \left[\frac{x^3}{3} - \frac{5x^2}{2} + 6x\right]_2^3 = -\frac{1}{6}$$

$$I_1 = \int_3^4 (x^2 - 5x + 6)dx = \left[\frac{x^3}{3} - \frac{5x^2}{2} + 6x\right]_3^4 = \frac{5}{6}$$

Para obter a área pedida, deve-se somar os absolutos das 3 integrais I_1, I_2 e I_3:

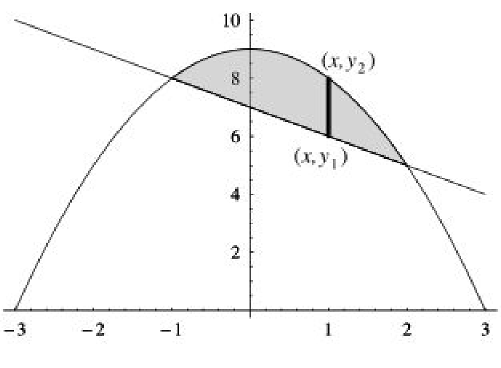
$$A = |I_1| + |I_2| + |I_3| = \frac{14}{3} + \frac{1}{6} + \frac{5}{6} = \frac{17}{3}$$

Para determinar a área limitada por 2 curvas $y = f(x)$ e $y = g(x)$ primeiro deve-se descobrir seus pontos de intersecção. Isto pode ser feito resolvendo $f(x) = g(x)$. Se as curvas se interceptam apenas em 2 locais, digamos $x = a$ e $x = b$ e $f(x)$ está sobre $g(x)$, isto é $f(x) \geqslant g(x)$ para $x \in [a, b]$ a área será

$$A = \int_a^b [f(x) - g(x)]dx$$

Se as curvas se interceptam em mais do que dois locais, a área deve ser calculada pela subdivisão do intervalo, integrando separadamente cada subintervalo e somando os módulos de cada área.

Exemplo 4 Determine a área da região limitada pela parábola $y = 9 - x^2$ e pela reta $x + y = 7$



A parábola é representada pela função $y_2 = 9 - x^2$. Para determinar $g(x)$ precisa-se resolver a equação da reta em y

$$x + y = 7$$

$$y = 7 - x$$

$$y_1 = g(x) = 7 - x$$

Precisa-se agora os pontos de intersecção dessas duas curvas. Isto é obtido fazendo $f(x) = g(x)$

$$9 - x^2 = 7 - x$$

$$0 = x^2 - x - 2$$

$$0 = (x + 1)(x - 2)$$

$$x = -1 \qquad\qquad x = 2$$

Está claro no diagrama que $f(x) \geqslant g(x) \;\; \forall x \;\in\; [-1, 2]$

$$A = \int_a^b (y_2 - y_1)dx$$

$$= \int_a^b [f(x) - g(x)]dx$$

$$= \int_a^b [9 - x^2 - (7 - x)]dx$$

$$= \int_a^b [2 - x^2 + x]dx$$

$$= \left[2x - \frac{x^3}{3} + \frac{x^2}{2}\right]_{-1}^{2}$$

$$= \left(4 - \frac{8}{3} + 2\right) - \left(-2 + \frac{1}{3} + \frac{1}{2}\right)$$

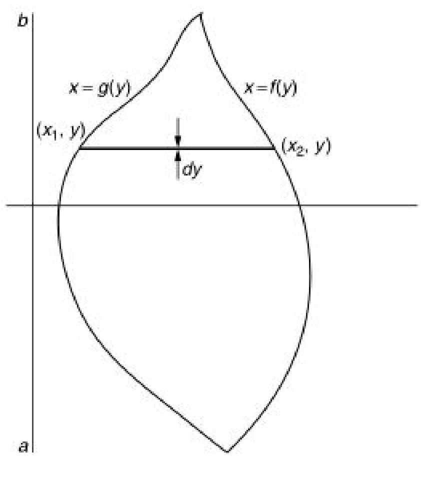
$$= \frac{10}{3} - \left(-\frac{7}{6}\right)$$

$$= \frac{9}{2}$$

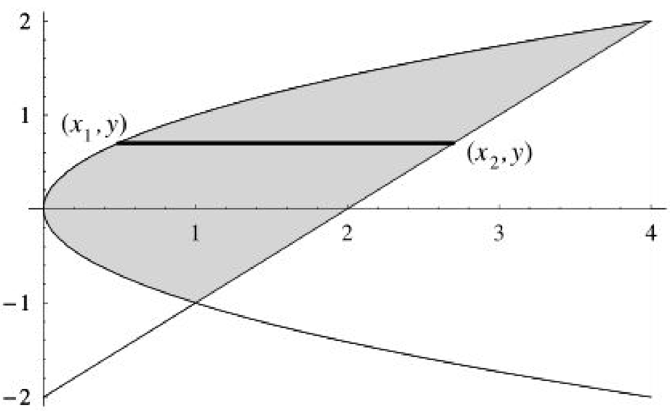
Ocasionalmente é mais conveniente para computar a área avaliar a expressão em relação a y ao invés de x. Se a região é descrita como uma intersecção dos gráficos $x = f(y)$ e $x = g(y)$ a área pode ser representada como uma integral cuja variável de integração é y

$$A = \int_a^b [f(y) - g(y)]dy$$

Assumindo que a e b são as coordenadas de y dos pontos de intersecção dos dois gráficos e que $f(y) \geqslant g(y) \quad \forall y \in [a, b]$. Neste tipo de problema nosso retângulo imaginário anda paralelamente ao eixo x e extende-se da esquerda para a direita de $x_1 = g(y)$ até $x_2 = f(y)$



Exemplo 5 Ache a área da região limitada pela parábola $x = y^2$ e a reta $y = x - 2$



A equação $y = x - 2$ é equivalente à $x = y + 2$. Como a reta fica à direita da parábola na região sob consideração, tem-se $f(y) = y + 2$ e $g(y) = y^2$. Isto garante que $f(y) - g(y)$ é não negativo. As intersecções são obtidas fazendo $g(y) = f(y)$ para y

$$y^2 = y + 2$$

$$y^2 - y - 2 = 0$$

$$(y + 1)(y - 2) = 0$$

$$y = -1 \qquad\qquad y = 2$$

$$A = \int_a^b (x_2 - x_1)dy$$

$$= \int_a^b [f(y) - g(y)]dy$$

$$= \int_{-1}^2 [(y + 2) - y^2]dy$$

$$= \left[\frac{y^2}{2} + 2y - \frac{y^3}{3}\right]_{-1}^2$$

$$= \left(2 + 4 - \frac{8}{3}\right) - \left(\frac{1}{2} - 2 + \frac{1}{3}\right)$$

$$= \frac{10}{3} - \left(-\frac{7}{6}\right)$$

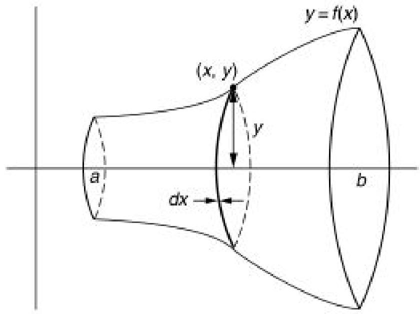
$$= \frac{9}{2}$$

9.10 Volumes de sólidos de revolução

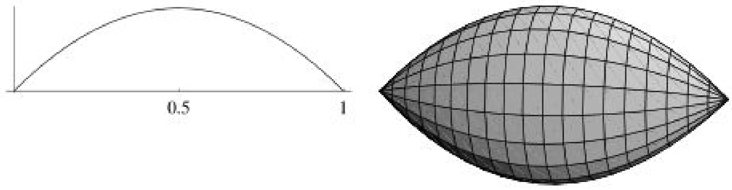
Se a região limitada pela função $y = f(x)$ e o eixo x entre $x = a$ e $x = b$ é revolucionada sobre o eixo x o resultado é uma figura tri-dimensional conhecida como sólido de revolução. A área de uma sessão cruzada é circular e seu volume pode ser computado pelo cálculo da integral

$$V = \pi \int_a^b [f(x)]^2 dx \qquad ou \qquad V = \pi \int_a^b y^2 dx$$

Um jeito mnemônico para lembrar ou entender esta fórmula é pensar no sólido sendo fatiado em discos infinitesimalmente finos de raio y e largura dx. O volume deste disco é $\pi(raio)^2(largura) = \pi y^2 dx$ e a soma destes volumes é $\pi \int_a^b y^2 dx$. Note que como π é uma constante ela passa para fora da integral. Este método é conhecido como método do disco.



Exemplo 6 Ache o volume do sólido de revolução obtido pela revolução da região limitada por $y = x - x^2$ e o eixo x sobre o eixo x.



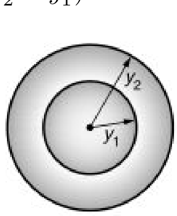
O gráfico intercepta o eixo x quando $y = 0$

$$y = x - x^2$$
$$0 = x - x^2$$
$$0 = x(1 - x)$$
$$x = 0 \qquad x = 1$$

O volume de cada disco imaginário circular é $\pi y^2 dx$ e o volume da região é obtido pela integração em relação a x de 0 a 1.

$$V = \pi \int_0^1 y^2 dx$$
$$= \pi \int_0^1 (x - x^2)^2 dx$$
$$= \pi \int_0^1 (x^2 - 2x^3 + x^4) dx$$
$$= \pi \left[\frac{x^3}{3} - \frac{2x^4}{4} + \frac{x^5}{5} \right]_0^1$$
$$= \pi \left[\frac{1}{3} - \frac{2}{4} + \frac{1}{5} - 0 \right]$$
$$= \frac{\pi}{30}$$

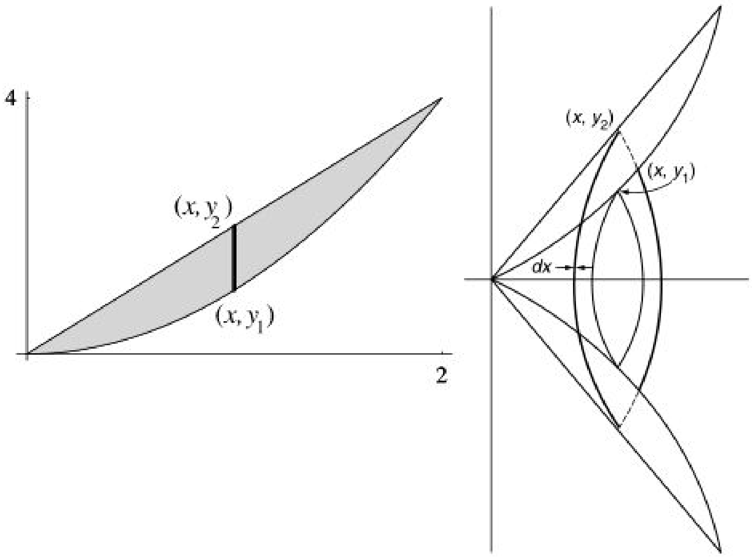
Se a região limitada por duas curvas é revolucionada sobre o eixo x o sólido de revolução será oco. A sessão cruzada terá a forma de uma arruela, isto é a área entre dois circulos concêntricos. O método de calcular este volume é chamado método da arruela. Se o circulo interno é y_1 e o raio externo é y_2 e pensamos na largura da arruela como dx o volume dela será $V_{externo} - V_{interno} = \pi y_2^2 dx - \pi y_1^2 dx = \pi(y_2^2 - y_1^2)dx$



O volume do sólido da revolução expresso como integral é

$$V = \pi \int_a^b (y_2^2 - y_1^2) dx$$

Exemplo 7 Ache o volume obtido pela região limitada por $y = x^2$ e $y = 2x$ quando rotacionada sobre o eixo x.



Primeiro precisa-se achar os pontos de intersecção das duas curvas

$$x^2 = 2x$$

$$x^2 - 2x = 0$$

$$x(x-2) = 0$$

$$x = 0 \qquad x = 2$$

Quando a região é rotacionada o raio externo da arruela y_2 é determinada pela reta e o raio interno y_1 é determinada pela parábola, então

$$V = \pi \int_a^b (y_2^2 - y_1^2) dx$$

$$= \pi \int_0^2 [(2x)^2 - (x^2)^2] dx$$

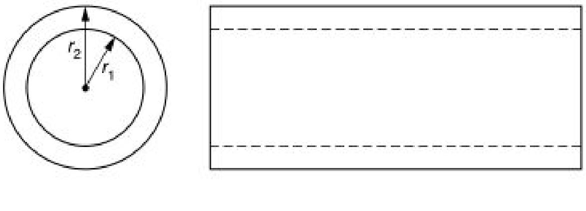
$$= \pi \int_0^2 [4x^2 - x^4] dx$$

$$= \pi \left[\frac{4x^3}{3} - \frac{x^5}{5} \right]_0^2$$

$$= \pi \left[\frac{32}{3} - \frac{32}{5} - 0 \right]$$

$$= \frac{64}{15} \pi$$

Outro método para calcular volumes de sólidos de revolução é o método casca, que usa uma abordagem diferente para construir a integral representando o volume. Considere uma parede fina (casca cilíndrica) com raios internos e externos r_1 e r_2, respectivamente, onde $r_1 \approx r_2$, e altura h. (Imagine uma lata de sopa com a parte superior e inferior cortadas. A espessura da parede da a lata é $r_2 - r_1$.)



O volume da casca é a diferença entre os dois volumes cilíndricos

$$\begin{aligned} V_{casca} &= V_{cilindro\ externo} - V_{cilindro\ interno} \\ &= \pi r_2^2 h - \pi r_1^2 h \\ &= \pi(r_2^2 - r_1^2)h \\ &= \pi(r_2 + r_1)(r_2 - r_1)h \\ &= 2\pi\left(\frac{r2 + r1}{2}\right)h(r2 - r1) \end{aligned}$$

Se imaginarmos $r_{med} = \frac{r_2+r_1}{2}$ como o raio médio e $\Delta r = r_2 - r_1$ representando a largura do muro da casca, pode-se escrever

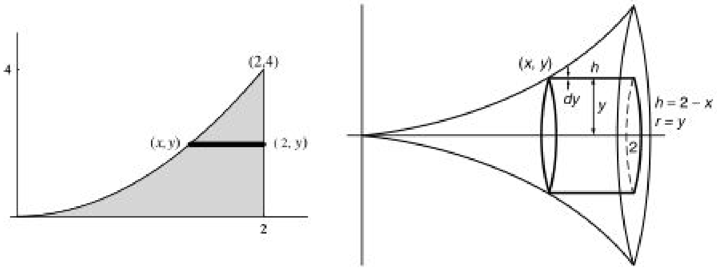
$$V_{casca} = 2\pi r_{med} h \Delta r$$

Como $\Delta r \to 0$ e o número de cascas dentro do sólido $\to \infty$ a soma dos seus volumes tende ao volume do sólido de revolução. Como mnemônico pode-se representar o raio médio por r, o comprimento por h e a largura infinitesimal da casca por dr. O volume de uma típica casca pode ser pensado como $2\pi rhdr$ e o volume total como

$$V = 2\pi \int_a^b rhdr$$

Em um dado problema dr pode ser trocado por dx ou dy dependendo de onde está o eixo de rotação (dx se é rotado no eixo y e dy se rotado no eixo x). Em qualquer caso a altura h e o raio r devem se expressos em termos da variável de integração. Idem para os limites de integração.

Exemplo 8 Ache o volume do sólido de revolução formado pela rotação da região limitada pela parábola $y = x^2$ e as linhas $y = 0$ e $x = 2$ sobre o eixo x.



Se (x, y) está na parábola, o comprimento da casca gerada é $h = x - 2$, o raio $r = y$ e a espessura da casca é dy. O volume de uma típica casca é $2\pi y(2 - x)dy$ e o volume total é obtido pela integração. Como apareceu dy a integração deve ser em y. Como $y = x^2$, $y = 0$ quando $x = 0$ e $y = 4$ quando $x = 2$, daí

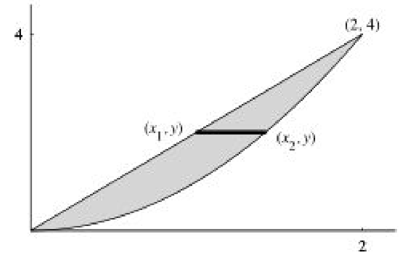
$$V = 2\pi \int_0^4 y(2 - x)dy$$

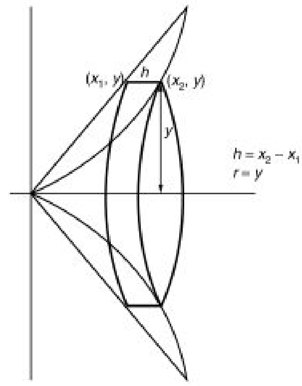
A variável x deve ser expressa em termos de y antes da integração poder ser feita. Já que $y = x^2$, $x = \sqrt{y}$

$$\begin{aligned} V &= 2\pi \int_0^4 y(2 - \sqrt{y})dy \\ &= 2\pi \int_0^4 (2y - y^{3/2})dy \\ &= 2\pi \left[y^2 - \frac{2}{5}y^{5/2}\right]_0^4 \\ &= 2\pi \left(16 - \frac{2}{5}4^{5/2} - 0\right) \\ &= 2\pi \left(16 - \frac{64}{5}\right) \\ &= 2\pi(\frac{16}{5}) \\ &= \frac{32}{5}\pi \end{aligned}$$

A natureza do método da casca sugere seu uso em caso de achar volumes ocos.

Exemplo 9 Ache o volume obtido pela região limitada por $y = x^2$ e $y = 2x$ quando rotacionada sobre o eixo x. (Este problema foi resolvido no exemplo 7 acima através do método da arruela. Compare-os).





O tamanho de cada casca é $h = x_2 - x_1$ e o raio é $r = y$. A espessura de cada casca é dy. Assim, o volume de uma casca típica é $2\pi(x_2 - x_1)ydy$ e o volume total é

$$V = 2\pi \int_a^b (x_2 - x_1)ydy$$

A variável de integração é y, então tanto x_1 como x_2 devem ser expressos em termos de y. Os limites de integração devem ser achados em termos de y também.

Como a equação da parábola é $y = x^2$ então $x_2 = \sqrt{y}$ e a equação da reta $y = 2x$ determina $x_1 = \dfrac{y}{2}$. Os pontos de intersecção são $(0, 0)$ e $(2, 4)$. Assim, o volume é

$$
\begin{aligned}
V &= 2\pi \int_0^4 \left(\sqrt{y} - \frac{y}{2}\right) ydy \\
&= 2\pi \int_0^4 \left(y^{3/2} - \frac{1}{2}y^2\right) dy \\
&= 2\pi \left[\frac{2}{5}y^{5/2} - \frac{1}{6}y^3\right]_0^4 \\
&= 2\pi \left(\frac{64}{5} - \frac{64}{6} - 0\right) \\
&= \frac{64}{15}\pi
\end{aligned}
$$

9.11 Problemas Suplementares

1. Calcula a área da região delimitada pela curva $y = 8 - x^2 - 2x$ e o eixo x.
2. Determine a área da região delimitada pela curva $y = x^3 - 4x^2 + 3x$ e o eixo x, $0 \leq x \leq 3$.
3. Encontre a área da região limitada pela curva $y = x^3$ e a linha $y = 8$ usando
   (a) retângulos verticais e
   (b) retângulos horizontais.
4. Determine a área da região delimitada pelas curvas $y = x^4 - x^2$ e $y = x^2 - 1$.
5. Encontre a área da região delimitada pelas curvas $4x - y^2 = 0$ e $y = 2x - 4$.
6. Encontre a área da região limitada pela parábola $y = x^2$, a linha tangente à parábola no ponto $(2, 4)$ e o eixo x.
7. Derive uma fórmula para o volume de uma esfera de raio r girando o semicírculo $y = \sqrt{r^2 - x^2}$ em torno do eixo x.
8. Calcule o volume do sólido obtido girando a região limitado por $y = x^2$, $y = 8 - x^2$ e o eixo y em torno do eixo x.
9. Um furo de raio 2 é perfurado através do eixo de uma esfera de raio 3. Calcule o volume do sólido remanescente.

As soluções a estes problemas estão nas págs 193 a 204.

9.12 Integrais Impróprias

Vai-se extender conceito integral definida para o caso em que o intervalo é infinito e também para o caso em que f tem uma descontinuidade infinita em $[a, b]$. Em ambos casos a integral é chamada imprópria. Um exemplo real aqui são as distribuições de probabilidade.

No primeiro caso, se $\int_a^b f(x)dx$ existe para casa número $t \geq a$ então

$$\int_a^{\infty} f(x)dx = \lim_{t\to\infty} \int_a^t f(x)dx$$

Há um caso similar quando $t \to -\infty$ e deve ser olhado na pág. 471 de [Ste14]. Finalmente as integrais acima são chamadas convergentes quando o limite existe e divergentes senão.

No segundo caso, a região é infinita na vertical (no caso anterior era na horizontal)

Um exemplo: Os resultados de QI têm distribuição normal com média 100 e desvio padrão 15. Perguntas:

- Qual a percentagem da população com QI entre 85 e 115 ?
- Qual a percentagem da população com QI igual ou maior a 135 ?

Esta distribuição tem a seguinte fórmula: (P=densidade; σ =desvio padrão e μ =média.)

$$P(a \leq x \leq b) = \int_a^b \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\sigma)^2/(2\sigma^2)} dx$$

(Obviamente se $a = -\infty$ e $b = \infty$ então P=1). Direto no Maple

```
> int((1/(15*sqrt((2)*3.14159265)))*exp(-(x-100)^2/(2*15^2)),x=85..115);
      .6826894925
> int((1/(15*sqrt((2)*3.14159265)))*exp(-(x-100)^2/(2*15^2)),x=135..infinity);
      .0098153288
```

Como era de se esperar numa distribuição normal, no primeiro caso são 68% da população. No segundo caso, menos de 1%. Este resultado numérico confere com o [Ste14] pág.519.

A mesma coisa no sympy

```
>>> sp.init_session()
>>> integrate((1/(15*sqrt((2)*3.14159265)))*exp(-(x-100)**2/(2*15**2)),(x,85,115)).n()
0.682689492527129
>>> integrate((1/(15*sqrt((2)*3.14159265)))*exp(-(x-100)**2/(2*15**2)),(x,135,oo)).n()
0.00981532863425316
```

Neste você pode pedir a integral calculada como em

```
>>> latex(Integral((1/(15*sqrt((2)*3.14159265)))*exp(-(x-100)**2/(2*15**2)),(x,85,115)))
\int\limits_{85}^{115} 0.0265961520419574 e^{- \frac{\left(x - 100\right)^{2}}
{450}}\, dx
```

Com o seguinte resultado

$$\int\limits_{85}^{115} 0.0265961520419574 e^{- \frac{\left(x - 100\right)^{2}}{450}}\, dx$$

9.13 Logaritmo natural como integral

Define-se o logaritmo natural como

$$\ln x = \int_1^x \frac{1}{t} dt \qquad x > 0$$

Pelo teorema fundamental do cálculo

$$\frac{d}{dx}(\ln x) = \frac{1}{x}$$

Capítulo 10

Máximos e mínimos

Problemas de otimização são aplicações importantes no cálculo diferencial. Se se está preocupado em ir de A a B na menor quantidade de tempo, ou se se pretende construir uma caixa com o volume máximo para uma determinada quantidade de material empregado está-se procurando o jeito ótimo de resolver um problema.

Este tipo de problema geralmente se reduz a achar os valores máximos ou mínimos da função objetivo sujeitos a um determinado conjunto de condições ou restrições.
Inicia-se por uma revisão de algumas definições e teoremas.

> Def: Uma função f tem um valor máximo num intervalo I se existe um número $c \in I$ tal que $f(x) \leq f(c)$ para todo $x \in I$.

Uma definição similar com a desigualdade revertida aplica-se ao mínimo. Note que em ambos o valor absoluto máximo ou mínimo é $f(c)$ e sua localização $x = c$. A existência do máximo ou mínimo sob certas condições é garantida pelo Teorema do Valor Extremo.

> Teo: Se f é contínua em um intervalo fechado $I = [a, b]$ então f tem um absoluto máximo nesse intervalo e tem também um absoluto mínimo nesse intervalo.

Há ainda a definição de número crítico:

> Def: Um número crítico de f é o número x para o qual ou $f'(x) = 0$ ou $f'(x)$ não existe.

Em um problema textual advindo de um problema físico ou geométrico, é muito difícil que a derivada não exista. Portanto, daqui para a frente só se considerarão pontos críticos onde $f'(x) = 0$. É fácil ver que se uma função é contínua, seus extremos (que existem pelo teorema do valor extremo) ocorrerão em números críticos ou nos limites do intervalo. Os seguintes passos que podem ser chamados de *Método do Intervalo Fechado* podem ser usados para determinar seus valores

Passo 1 Ache todos os pontos críticos de $f(x)$ que estão no intervalo $[a, b]$.

Passo 2 Calcule os valores de $f(x)$ em cada ponto crítico e também nos limites a e b.

Passo 3 O maior valor obtido no passo 2 acima é o máximo. O menor é o mínimo.

Exemplo 1 Ache o máximo e mínimo absolutos de $f(x) = x^3 - 12x + 5$ no intervalo $[-1, 4]$.

Passo 1

$$f(x) = x^3 - 12x + 5$$
$$f'(x) = 3x^2 - 12$$
$$0 = 3x^2 - 12$$
$$12 = 3x^2$$
$$4 = x^2 \qquad \Rightarrow \qquad x = 2, \quad x = -2$$

Passo 2 O valor $x = -2$ está fora do intervalo e pode ser ignorado.

x	$f(x)$
-1	16
2	-11
4	21

Passo 3 O valor máximo absoluto de $f(x)$ em $[-1, 4]$ é 21 e o mínimo absoluto é -11.

Muitas funções que ocorrem em conexão com problemas de máximo e mínimo serão contínuas, mas ocasionalmente o intervalo representando seu domínio será aberto ou terá comprimento infinito. Em tais situações o método do intervalo falha. Tais funções podem até mesmo não ter extremo absoluto.

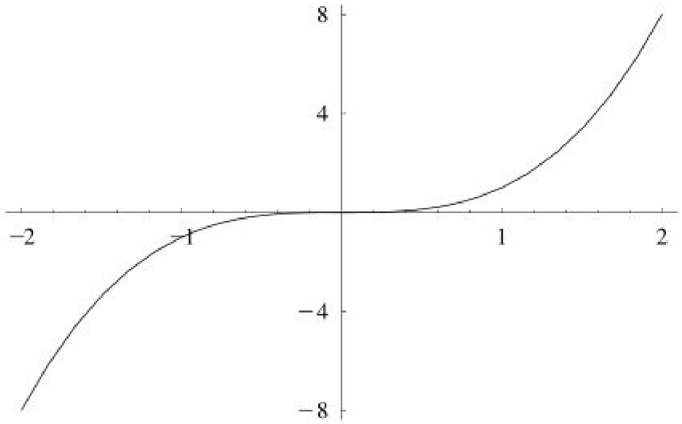
A função $f(x) = x^2$ no intervalo aberto $(-2, 3)$ tem um mínimo absoluto em $x = 0$ mas não tem máximo absoluto.

Def: Uma função f tem máximo local (ou relativo) em c se existe um intervalo aberto I contendo c para o qual $f(x) \leq f(c)$ para qualquer $x \in I$.

Há uma definição similar para mínimo, com a desigualdade revertida.

Pode-se dizer que se $f(x)$ tem um extremo relativo (max ou min) em c, então c é um número crítico da função. Mas, o contrário não é verdadeiro. Uma função pode ter um número crítico que não corresponda nem a mínimo nem a máximo.

Considere, por exemplo, a função $f(x) = x^3$. Já que $f'(x) = 3x^2$ e $f(0) = 0$, 0 é um valor crítico para f. Mas 0 não é nem máximo relativo nem mínimo relativo. Uma olhada em seu gráfico mostra porque



Para decidir se um número crítico é um máximo relativo ou mínimo relativo ou nenhum, vai-se ver 2 testes chamados *Teste da primeira derivada* e *Teste da segunda derivada.*

Teste da primeira derivada: Seja c um valor crítico de f
(a) Se $f'(x)$ muda de positivo para negativo quando x varia da esquerda para a direita de c, então c é máximo de f.
(b) Se $f'(x)$ muda de negativo para positivo quando x varia da esquerda para a direita de c, então c é mínimo de f.

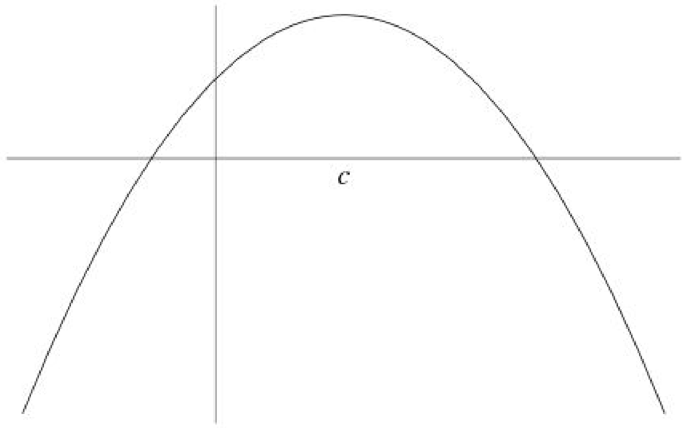
Teste da segunda derivada: Seja c um valor crítico de f
(a) Se $f''(c) < 0$ então c é um c é um mínimo relativo de f

Se $f''(c) = 0$ o teste da segunda derivada falha e deve-se usar o primeiro teste. Mas isto é raro, e portanto o segundo deve ser o preferido.

Quando se resolve um problema maxmin procura-se um maxmin absoluto e a pergunta é porque perder tempo com maxmin relativos. A resposta é dada pelo teorema

Teo: Se uma função contínua f tem apenas um extremo relativo no intervalo I em c então
(a) Se o extremo relativo é um mínimo relativo então f tem mínimo absoluto em c
(b) Se o extremo relativo é um máximo relativo então f tem máximo absoluto em c

Note que se o intervalo é fechado este teorema oferece uma alternativa conveniente ao Método do Intervalo Fechado. O teorema pode ser entendido com a ajuda de um diagrama simples



Muitos problemas textuais miram em apenas um extremo relativo. Para determinar se ele é um maxmin absoluto simplesmente determinamos se ele é um maxmin relativo. Isto é facilmente conseguido com os testes da primeira ou segunda derivada. Para problemas de maxmin, eis um roteiro

Passo 1 Desenhe um diagrama. Rotule as quantidades conhecidas e desconhecidas que aparecem no problema.

Passo 2 Escreva uma equação representando a variável que deve ser maximizada ou minimizada. Esta quantidade será – tipicamente – representada em termos de duas ou mais variáveis.

Passo 3 Use qualquer restrição ou relacionamento entre as variáveis para eliminar todas exceto uma: esta será a variável independente. Isto converte a equação do item 2 em função. Determine o domínio apropriado ao problema, isto é determine os valores para os quais a variável independente o problema faz sentido.

Passo 4 Acha-se todos os números críticos

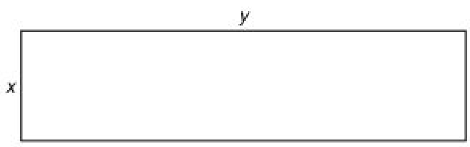
Passo 5 Se a função é contínua em intervalo fechado (extremidades incluídas) use o método dos intervalos fechados para determinar os minmax absolutos

OU

Se há apenas um valor crítico dentro do intervalo sob consideração, use os testes da primeira ou da segunda derivada para ver se é um valor relativo máximo ou mínimo. O valor da função aqui será um maxmin absoluto.

Exemplo 2 Jodi pretende usar 100 pés de cerca para fechar um jardim retangular. Determine a área máxima do jardim.

Passo 1



Passo 2

$$A = xy$$

Passo 3 Já que o perímetro do jardim deve ter 100 pés,

$$2x + 2y = 100$$

$$x + y = 50$$

$$y = 50 - x$$

Do passo 2, tem-se $A(x) = x(50 - x)$ ou $A(x) = 50x - x^2$. Já que x não pode ser negativo, seu valor menor é 0. Como $y = 50 - x$ e y também não pode ser negativo, o maior valor possível para x é 50.

$$0 \le x \le 50$$

Note-se que para obter um intervalo fechado vai-se permitir que x assuma os valores 0 e 50 como dimensões aceitáveis. Estes retângulos são degenerados e têm área 0.

Passo 4 Já que $A(x) = 50x - x^2$ é um polinômio sua derivada sempre existe. O valor crítico ocorre quando $A'(x) = 0$.

$$A(x) = 50x - x^2$$
$$A'(x) = 50 - 2x = 0$$
$$x = 25$$

Passo 5 Usando o método do intervalo fechado

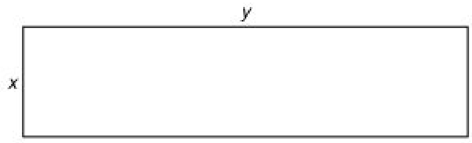
x	$A(x)$
0	0
25	625
50	0

A área máxima é de 625 $pés^2$

Exemplo 3 Qual o perímetro mínimo possível para um retângulo cuja área seja 100 pol^2?

À primeira vista este problema é similar ao anterior. Como veremos, ele é ligeiramente diferente e deve ser resolvido usando outra estratégia.

Passo 1



Passo 2 Deve-se minimizar

$$P = 2x + 2y$$

Passo 3 A restrição dada é que a área deve ser 100 pol^2

$$xy = 100$$
$$y = \frac{100}{x}$$

Agora pode-se representar P como função de x

$$P(x) = 2x + 2\left(\frac{100}{x}\right)$$
$$= 2x + \frac{200}{x}$$
$$= 2x + 200x^{-1}$$

Neste ponto a solução toma outro rumo. Ao contrário do exemplo anterior não se pode permitir $x = 0$. Não apenas porque isto causaria problemas em $P(x)$, mas mais fundamentalmente porque é impossível ter um retângulo de área 100 com um dos seus lados valendo 0. Do outro lado do espectro qual seria o maior valor que x poderia ter ? Pode-se ver que não há limite aqui, pois y sempre poderia ser escolhido com valor suficientemente pequeno para que $xy = 100$. Já que o domínio aqui é $(0, \infty)$ o método dos intervalos fechados não pode ser usado.

Passo 4 Deve-se achar os valores críticos.

$$P'(x) = 2 - 200x^{-2}$$
$$0 = 2 - \frac{200}{x^2}$$
$$\frac{200}{x^2} = 2$$
$$2x^2 = 200$$
$$x^2 = 100$$
$$x = 10$$

O valor $x = -10$ não é considerado porque cai fora do domínio que é $0 < x < \infty$.

Passo 5 Deve-se aplicar a segunda derivada para testar o valor crítico, $x = 10$.

$$P'(x) = 2 - 200x^{-2}$$

$$P''(x) = 400x^{-3} = \frac{400}{x^3}$$

$$P''(10) > 0$$

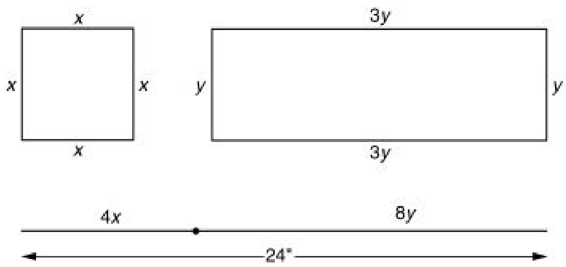
Não se precisa conhecer o valor exato de $P''(10)$. O fato de ser positivo nos informa que tem-se um mínimo relativo em $x = 10$. Como este valor é o único valor crítico, ele é absoluto. Como $y = 100/x$, daqui $y = 10$ quando $x = 10$. O perímetro mínimo é $2x + 2y = 40$.

Testar todos os números críticos pode ser considerado um procedimento opcional. Se você está convencido que o problema tem uma solução e você achou um número crítico então este ponto deve ser a solução do problema. Entretanto, muitas vezes a intuição falha e de uma perspectiva matemática o teste é a única maneira de justificar a solução.

O próximo exemplo ilustra a importância de checar os extremos de um intervalo.

Exemplo 4 Uma peça de arame de 24 polegadas deve ser usada para formar um quadrado e/ou um retângulo cujo comprimento seja 3 vezes a largura. Determine o máximo e mínimo das áreas combinadas.

Passo 1



Passo 2

$$A = x^2 + 3y^2$$

Passo 3 Como o perímetro combinado das duas figuras deve ser 24 polegadas tem-se

$$4x + 8y = 24$$

$$x + 2y = 6$$

$$x = 6 - 2y$$

Note que esta equação poder-se-ia resolver para y e ficaria $y = \dfrac{6-x}{2}$. Entretanto, para evitar o uso de frações prefere-se resolver em x.

Trocando x em termos de y no passo 2 fica

$$A(y) = (6 - 2y)^2 + 3y^2$$

Lembre que $4x + 8y = 24$. Se todo o arame é usado para formar o quadrado, $y = 0$. Se todo o arame é usado para formar o retângulo, $x = 0$ e então $8y = 24$ e $y = 3$. Daqui:

$$A(y) = (6 - 2y)^2 + 3y^2 \qquad 0 \leq y \leq 3$$

Passo 4 O livro sugere usar a regra do produto e tem desenvolvimento ligeiramente diferente (vide pág. 82) mas obviamente o resultado é igual.

$$A(y) = 36 - 24y + 4y^2 + 3y^2$$

$$A(y) = 36 - 24y + 7y^2$$

$$A'(y) = -24 + 14y$$

e igualando a zero fica

$$0 = -24 + 14y$$

$$y = \frac{24}{14} = \frac{12}{7}$$

Passo 5 Calcular $A(y)$ nos números críticos e nos pontos extremos do intervalo.

$$A(y) = (6-2y)^2 + 3y^2$$

$$A(0) = 36$$

Neste caso todo o arame é usado para formar o quadrado

$$A(\frac{12}{7} = \left(6 - \frac{24}{7}\right)^2 + 3\left(\frac{12}{7}\right)^2$$

$$= \left(\frac{18}{7}\right)^2 + 3\left(\frac{12}{7}\right)^2$$

$$= \frac{324}{49} + \frac{432}{49}$$

$$= \frac{756}{49} = \frac{108}{7} \approx 15.43$$

Neste caso o arame forma as 2 figuras

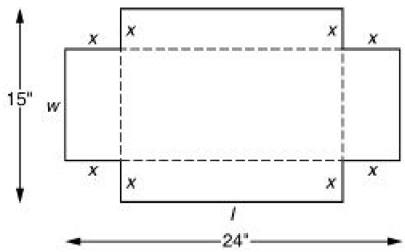
$$A(3) = 27$$

Finalmente neste caso, todo o arame foi usado para formar o retângulo.

A área máxima ocorre no ponto extremo do intervalo de y ($y = 0$). A área mínima ocorre quando $y = \frac{12}{7}$. Para obter esta área mínima o arame deve ser cortado com o y valendo $8y = \frac{96}{7} = 13\frac{5}{7}$ *polegadas* e o remanescente $10\frac{2}{7}$ *polegadas* será usado para construir o quadrado.

Exemplo 5 Uma caixa aberta é formada cortando quatro quadrados de igual tamanho dos cantos de uma chapa retangular de 24 por 15 polegadas e dobrando os lados. Determine as dimensões do corte que maximizam o volume da caixa.

Passo 1 Seja x o comprimento do corte



Passo 2

$$V = l \times w \times h$$

Passo 3

$$l = 24 - 2x$$

$$w = 15 - 2x$$

$$h = x$$

Do passo 2,

$$V(x) = (24-2x)(15-2x)(x)$$

$$= 4x^3 - 78x^2 + 360x \qquad 0 \le x \le \frac{15}{2}$$

Já que se $x > \frac{15}{2}$, w vira negativo e não pode.

Passo 4

$$V'(x) = 12x^2 - 156x + 360$$

igualando a zero, fica

$$0 = 12x^2 - 156x + 360$$

$$0 = 12(x^2 - 13x + 30)$$

$$0 = 12(x - 10)(x - 3)$$

$$x = 10 \qquad x = 3$$

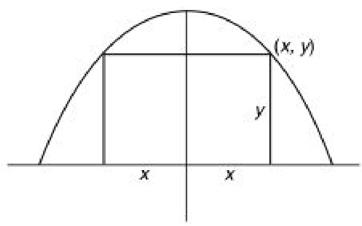
Passo 5 O valor $x = 10$ deve ser ignorado já que ele está fora do intervalo $0 \leq x \leq \dfrac{15}{2}$.

x	$V(x)$
0	0
3	486
15/2	0

O máximo volume ocorre quando $x = 3$ e ele é de 486 pol^3.

Exemplo 6 Ache as dimensões de um retângulo de maior área cuja base está no eixo x e cujos dois vértices superiores estão sobre a parábola $y = 12 - x^2$. Qual é a área máxima ?

Passo 1 Seja (x, y) o ponto na parábola que também pertence ao retângulo



Passo 2 A altura do retângulo é y e sua largura é $2x$. A área do retângulo é $A = 2xy$.

Passo 3 Já que os vértices superiores do retângulo estão sobre a parábola, suas coordenadas devem satisfazer $y = 12 - x^2$ e a área do retângulo vira

$$A(x) = 2x(12 - x^2)$$

$$= 24x - 2x^3 \qquad 0 \leq x \leq \sqrt{12}$$

Passo 4

$$A'(x) = 24 - 6x^2$$

$$0 = 24 - 6x^2$$

$$6x^2 = 24$$

$$x^2 = 4$$

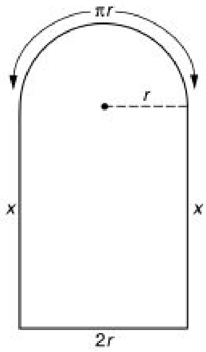
$$x = 2$$

Note que $x = -2$ não serve pois cai fora do intervalo $0 \leq x \leq \sqrt{12}$

Passo 5 $A(0) = 0$, $A(2) = 32$ e $A(\sqrt{12}) = 0$. Se $x = 2$, $y = 12 - x^2 = 8$. Já que a largura do retângulo é $2x$ as dimensões de retângulo são 4×8 e a área máxima é 32.

Exemplo 7 Uma janela de igreja tem a forma de um retângulo embaixo de um semicírculo. Se o perímetro da janela é 20 pés, qual a sua área máxima ?

Passo 1



Passo 2 A área da janela é a soma das áreas do retângulo e do semicírculo de cima. Já que a área de um círculo inteiro é πr^2 o semicírculo tem área $\frac{1}{2}\pi r^2$

$$A = 2rx + \frac{1}{2}\pi r^2$$

Passo 3 A janela tem 4 lados, 3 retos e um semicircular, no alto. A circunferência de um círculo completo é $2\pi r$, o arco semicircular é πr. Como o perímetro é 20 pés,

$$2x + 2r + \pi r = 20$$

$$x = \frac{20 - 2r - \pi r}{2}$$

Substituindo na equação obtida no passo 2, e obtendo a área como função de r

$$A(r) = 2r\left(\frac{20 - 2r - \pi r}{2}\right) + \frac{1}{2}\pi r^2$$

$$= 20r - 2r^2 - \pi r^2 + \frac{1}{2}\pi r^2$$

$$= 20r - 2r^2 - \frac{1}{2}\pi r^2$$

Passo 4

$$A(r) = 20r - 2r^2 - \frac{1}{2}\pi r^2$$

$$A'(r) = 20 - 4r - \pi r$$

$$= 20 - (4 + \pi)r$$

$$0 = 20 - (4 + \pi)r$$

$$(4 + \pi)r = 20$$

$$r = \frac{20}{4 + \pi} = 2.800495768$$

$$A''(r) = -4 - \pi$$

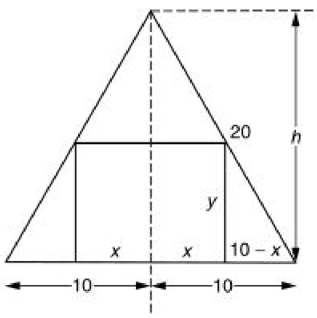
Já que $A''(r) < 0$ para todo r significa que ele é negativo no valor crítico. Então $\frac{20}{4+\pi}$ é um máximo relativo. Como é o único extremo relativo, a área absoluta máxima ocorre neste ponto. A área máxima é

$$A\left(\frac{20}{4+\pi}\right) = 20\left(\frac{20}{4+\pi}\right) - 2\left(\frac{20}{4+\pi}\right)^2 - \frac{1}{2}\pi\left(\frac{20}{4+\pi}\right)^2$$

$$= \frac{200}{4+\pi} \approx 28.005\ pés^2$$

Exemplo 8 Ache o retângulo com a maior área que pode ser inscrito em um triângulo equilátero de lado 20.

Passo 1



É conveniente deixar o lado do retângulo ser $2x$. A altura é y.

Passo 2 A área do retângulo é $2xy$.

Passo 3 Pelo teorema de Pitágoras, a altura h do triângulo equiláterio pode ser facilmente determinada.

$$h^2 + 10^2 = 20^2$$

$$h = \sqrt{300} = 10\sqrt{3}$$

Observando triângulos similares,

$$\frac{10 - x}{10} = \frac{y}{10\sqrt{3}}$$

Multiplicando em cruz,

$$10y = 10\sqrt{3}(10 - x)$$

$$y = \sqrt{3}(10 - x)$$

e agora a área pode ser escrita como

$$A(x) = 2x\sqrt{3}(10 - x)$$

$$= 2\sqrt{3}(10x - x^2)$$

Pelo diagrama, fica claro que $0 \leq x \leq 10$.

Passo 4

$$A'(x) = 2\sqrt{3}(10 - 2x)$$

$$0 = 2\sqrt{3}(10 - 2x)$$

$$0 = 10 - 2x$$

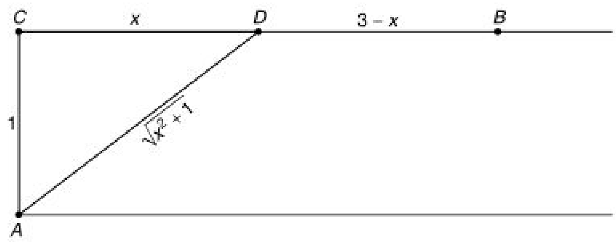
$$x = 5$$

O valor correspondente de y é $y = \sqrt{3}(10 - x) = 5\sqrt{3}$. Como as dimensões do retângulo são $2x$ e y, o maior valor da área possível é $10 \times 5\sqrt{3} = 50\sqrt{3}$.

Passo 5 (opcional) Se $x = 0$ ou $x = 10$ a área do retângulo é 0. Assim, o máximo absoluto ocorre em $x = 5$.

Exemplo 9 Um rio tem largura de 1 km. Frank deseja ir do ponto A ao ponto B do lado oposto do rio, 3 km a jusante. Se Frank pode correr a 5 km/hora e nadar a 3 km/hora, qual a menor quantidade de tempo que ele deve gastar para ir de A a B ?

Passo 1 Seja C o ponto diretamente oposto a A no outro lado do rio. Seja D o ponto onde ele deixa de nadar, para sair da água correndo a B. Seja x a distância de C a D.



Passos 2 e 3 Já que $d = r \times t$ o tempo necessário para correr ou nadar é determinado por $t = d/r$

$$t_{nadar} = \frac{d_{nadar}}{r_{nadar}} = \frac{\sqrt{x^2+1}}{3}$$

$$t_{correr} = \frac{d_{correr}}{r_{correr}} = \frac{3-x}{5}$$

O tempo total para ir de A até B é $t_{nadar} + t_{correr}$ então

$$t(x) = \frac{\sqrt{x^2+1}}{3} + \frac{3-x}{5}$$

está claro que $0 \leq x \leq 3$

Passo 4 Para facilitar a diferenciação, reescrevamos $t(x)$ usando expoentes

$$t(x) = \frac{1}{3}(x^2+1)^{1/2} + \frac{1}{5}(3-x)$$

$$t'(x) = \frac{1}{6}(x^2+1)^{-1/2}(2x) + \frac{1}{5}(-1)$$

$$= \frac{x}{3\sqrt{x^2+1}} - \frac{1}{5}$$

Igualando a zero e resolvendo

$$0 = \frac{x}{3\sqrt{x^2+1}} - \frac{1}{5}$$

$$5x = 3\sqrt{x^2+1}$$

$$25x^2 = 9(x^2+1)$$

$$16x^2 = 0$$

$$x = \frac{3}{4}$$

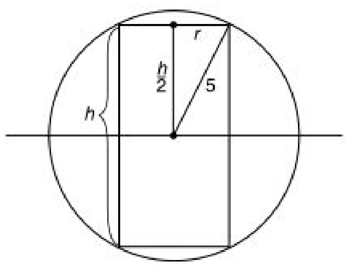
Passo 5 Antes de decidir qualquer coisa sobre o Frank, vamos examinar os limites do intervalo. Relembre que

$$t(x) = \frac{\sqrt{x^2+1}}{3} + \frac{3-x}{5}$$

Para $x = 0$, $t(0) = 56$ *minutos*
Para $x = 3/4$, $t(3/4) = 52$ *minutos*
Para $x = 3$, $t(3) = 63.2$ *minutos*

Exemplo 10 Qual é o maior volume possível de um cilindro inscrito em uma esfera de raio 5?

Passo 1 Examinando a figura a partir de uma perspectiva bidimensional



Passo 2 O volume de um cilindro circular de raio r e altura h é $V = \pi r^2 h$

Passo 3 Se desenharmos um ponto onde a esfera e o cilindro se encontram e usando o teorema de Pitágoras, fica

$$r^2 + \left(\frac{h}{2}\right)^2 = 5^2$$

$$r^2 = 25 - \frac{h^2}{4}$$

Note que como a equação do volume exige r^2 é mais conveniente resolver r em termos de h. O contrário (h em termos de r) – como é evidente – também vai funcionar, mas vai dar mais trabalho. Substituindo na equação do volume do passo 2

$$V(h) = \pi\left(25 - \frac{h^2}{4}\right)h \qquad 0 \leqslant h \leqslant 10$$

$$= \pi\left(25h - \frac{1}{4}h^3\right)$$

Passo 4

$$V'(h) = \pi\left(25 - \frac{3}{4}h^2\right)$$

igualando a zero fica

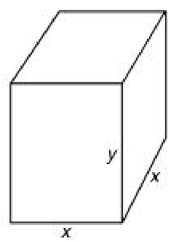
$$0 = 25 - \frac{3}{4}h^2$$

$$h = \frac{10}{\sqrt{3}}$$

Passo 5 (opcional) Já que $V(0) = 0$ e que $V(10) = 0$ o valor máximo do volume deve ocorrer quando $h = \frac{10}{\sqrt{3}}$. O valor correspondente de r deve ser calculado na equação $r^2 = 25 - \frac{h^2}{4}$ e $r = \frac{5}{3}\sqrt{6}$ O volume máximo é $V = \frac{500\pi}{3\sqrt{3}}$.

Exemplo 11 Uma caixa retangular fechada com uma base quadrada deve ter área de 150 polegadas quadradas. Qual o volume máximo que tal caixa pode conter ?

Passo 1



Passo 2 O volume de uma caixa é comprimento x largura x altura. De acordo com o desenho acima $V = x^2y$.

Passo 3 A área de uma caixa é a soma das áreas de seus 6 lados. O topo e a base da caixa têm área de x^2 e cada um dos 4 lados laterais têm área xy. Daqui

$$2x^2 + 4xy = 150$$

$$x^2 + 2xy = 75$$

$$y = \frac{75 - x^2}{2x}$$

$$V(x) = x^2\left(\frac{75 - x^2}{2x}\right)$$

$$= \frac{1}{2}(75x - x^3)$$

Passo 4

$$V'(x) = \frac{1}{2}(75 - 3x^2)$$

$$0 = 75 - 3x^2$$

$$x = 5$$

Calculando y

$$y = \frac{75 - x^2}{2x} = 5$$

O volume correspondente é $V = x^2 y = 125\ pol^3$

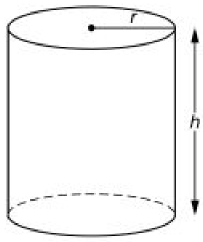
Passo 5 (opcional) Se você está convencido que a tal caixa tem o volume máximo, pare por aqui.

$$V''(x) = -3x \qquad V''(5) < 0$$

V tem máximo relativo em 5. Já que este é único extremo relativo para x positivo, voil-lá.

Exemplo 12 Uma lata cilíndrica deve conter 2000 polegadas cúbicas de líquido. Que dimensões minimizam a quantidade de metal usado para construir a lata ?

Passo 1 Seja r o raio e h a altura da lata



Passo 2

$$S = 2\pi rh + 2\pi r^2$$

Passo 3 Já que o volume deve ser de 2000,

$$\pi r^2 h = 2000$$

$$h = \frac{2000}{\pi r^2}$$

Substituindo na equação da superfície o valor acima fica

$$S(r) = 2\pi r\left(\frac{2000}{\pi r^2}\right) + 2\pi r^2$$

$$S(r) = \frac{4000}{r} + 2\pi r^2 \qquad 0 < r < \infty$$

Passo 4

$$S(r) = 4000r^{-1} + 2\pi r^2$$

$$S'(r) = -4000r^{-2} + 4\pi r$$

igualando a zero

$$0 = -\frac{4000}{r^2} + 4\pi r$$

e daqui

$$r = \frac{10}{\sqrt[3]{\pi}}\ pol$$

Como $h = \dfrac{2000}{\pi r^2}$, a altura correspondente é $h = \dfrac{20}{\sqrt[3]{\pi}}\ pol$

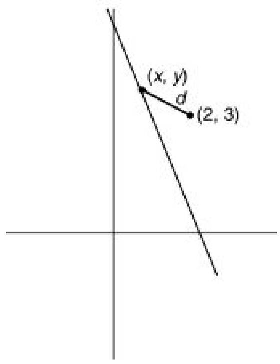
Passo 5

$$S''(r) = 8000r^{-3} + 4\pi$$
$$S''(r) = \frac{8000}{r^3} + 4\pi$$

É evidente, mesmo sem calculo nenhum, que como $r = 10/\sqrt[3]{\pi}$, S'' será positivo. Isto caracteriza a superfície mínima. Como é o único extremo relativo no intervalo $(0, \infty)$ tem-se um mínimo absoluto neste ponto.

Exemplo 13 Ache o ponto da reta $3x + y = 6$ mais próximo de $(2, 3)$.

Passo 1 Seja (x, y) um ponto da linha



Passo 2 A distância entre (x, y) e (2,3) é

$$d = \sqrt{(x-2)^2 + (y-3)^2}$$
$$d = \sqrt{x^2 + y^2 - 4x - 6y + 13}$$

Passo 3 Já que o ponto que se busca está na reta, $y = 6 - 3x$ e

$$d = \sqrt{x^2 + (6-3x)^2 - 4x - 6(6-3x) + 13}$$
$$d = \sqrt{10x^2 - 22x + 13}$$

Passo 4 A diferenciação desta expressão é complicada pela presença do radical. Há um truque simples que pode ser invocado para tornar a calculagem mais simples: Ao invés de minimizar d, vai-se minimizar d^2, já que se o valor de d^2 é mínimo, também o será o valor de d. Assim, minimizar d^2 tem o mesmo efeito prático que minimizar d. Por conveniência, faça-se $D = d^2$ e

$$D(x) = 10x^2 - 22x + 13$$
$$D'(x) = 20x - 22$$

igualando a zero

$$0 = 20x - 22$$
$$x = \frac{11}{10} \qquad y = 6 - 3x = \frac{60}{10} - \frac{33}{10} = \frac{27}{10}$$

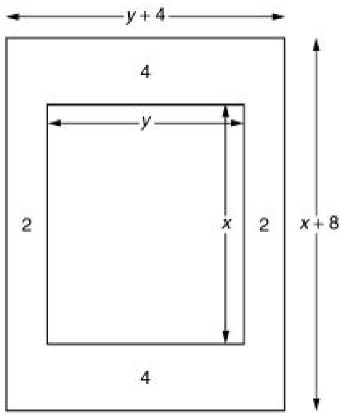
Uma olhada na figura deve nos convencer que a distância mínima certamente existe. Como $x = \dfrac{11}{10}$ é o único número crítico, o ponto $\left(\dfrac{11}{10}, \dfrac{27}{10}\right)$ é o mais próximo a $(2, 3)$.

A propósito, um pequeno trecho Maple

```
> e:=x^2+(6-3*x)^2-4*x-6*(6-3*x)+13;
                          2             2
                   e := x  + (6 - 3 x)  + 14 x - 23
> solve(e);
                       11             11
                       -- + 3/10 I,   -- - 3/10 I
                       10             10
> expand(e);
                                 2
                          10 x  + 13 - 22 x
```

Exemplo 14 Um poster retangular que deve conter 50 polegadas quadradas de impressão deve ter margens de 2 polegadas em cada lado e de 4 polegadas encima e embaixo. Que dimensões minimizam o uso de material ?

Passo 1 Ainda que se possa fazer x e y para representar as dimensões do poster, fica mais simples usar tais variáveis para a área impressa do poster e tem-se



Passo 2 Deve-se minimizar a área total do poster

$$A = (x + 8)(y + 4)$$

$$A = xy + 8y + 4x + 32$$

Passo 3

$$xy = 50$$

$$y = \frac{50}{x}$$

A área a ser minimizada pode ser expressa como uma função de x

$$A(x) = 50 + 8\left(\frac{50}{x}\right) + 4x + 32 \qquad 0 < x < \infty$$

$$= 82 + 400x^{-1} + 4x$$

Passo 4

$$A'(x) = -400x^{-2} + 4$$

$$0 = -\frac{400}{x^2} + 4$$

$$\frac{400}{x^2} = 4$$

$$x = 10$$

Como $y = 50/x$ o valor correspondente de y é 5. As dimensões do poster são $x + 8 = 18$ e $y + 4 = 9$ polegadas.

Passo 5 (opcional)

$$A'(x) = -400x^{-2} + 4$$

$$A''(x) = 800x^{-3}$$

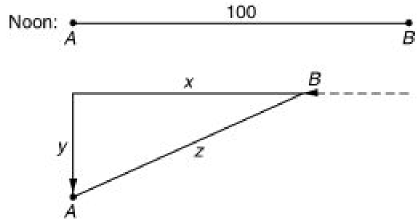
$$= \frac{800}{x^3}$$

$$A''(10) = \frac{800}{1000} > 0$$

Uma derivada segunda positiva no número crítico informa que se tem um mínimo relativo. Como este é o único extremo relativo no intervalo $(0, \infty)$, $A(x)$ tem um mínimo absoluto em 10.

Exemplo 15 à meia noite o barco B está 100 milhas a leste do barco A. Se o barco B navega a oeste a 10 milhas por hora e o barco A navega a sul a 20 milhas por hora, quando os barcos estarão mais próximos um do outro ? Qual a distância entre eles nesse momento ?

Passo 1



Passo 2 A distância entre os barcos é determonada pelo teorema de Pitágoras

$$z = \sqrt{x^2 - y^2}$$

Passo 3 Desde que B está viajando a 10 milhas/hora, após t horas, $x = 100 - 10t$, $y = 20t$. Então

$$z = \sqrt{(100 - 10t)^2 + (20t)^2}$$

$$= \sqrt{500t^2 - 2000t + 10000}$$

Passo 4 Por causa do radical, é mais conveniente minimizar z^2 ao invés de z. O valor que minimiza z^2 é o mesmo que minimiza z.

$$F(t) = 500t^2 - 2000t + 10000$$

$$F'(t) = 1000t - 2000$$

$$0 = 1000t - 2000$$

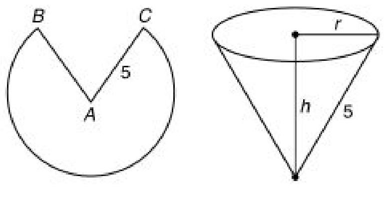
$$t = 2$$

Intuitivamente é óbvio que há algum ponto no tempo em que os barcos estão mais próximos. Já que o único número crítico é $t = 2$ e a distância mínimo entre eles é $z = \sqrt{500(2)^2 - 2000(2) + 10000} = \sqrt{8000} = 40\sqrt{5}\ milhas$. Pode-se usar a segunda derivada para confirmar o resultado matematicamente.

Passo 5 (opcional) Já que $F''(t) = 1000 > 0$, $t = 2$ é um mínimo relativo. Já que é o único relativo mínimo, ele é um mínimo absoluto.

Exemplo 16 Um copo cônico é construído com uma peça circular de papel de raio 5 pela extração de um corte em um setor e juntando o resultado das extremidade. Qual é o volume máximo do copo ?

Passo 1 Juntando AB a AC



Passo 2 O volume do cone resultante é $V = \frac{\pi}{3}r^2h$.

Passo 3 Do diagrama do passo 1, está aparente que $h^2 + r^2 = 25$. Segue-se que $r^2 = 25 - h^2$.

$$V = \frac{\pi}{3}r^2h$$

$$= \frac{\pi}{3}(25 - h^2)h$$

$$V(h) = \frac{\pi}{3}(15h - h^3) \qquad 0 \leqslant h \leqslant 5$$

Passo 4

$$V'(h) = \frac{\pi}{3}(25 - 3h^2)$$
$$0 = \frac{\pi}{3}(25 - 3h^2)$$
$$0 = 25 - 3h^2$$
$$3h^2 = 25$$
$$h = \frac{5}{\sqrt{3}}$$

O valor correspondente de r é $\sqrt{\frac{50}{3}}$ polegadas, já que $r = \sqrt{25 - h^2}$ e o volume máximo é $V = \frac{\pi}{3}r^2h = \frac{\pi}{3}\left(\frac{50}{3}\right)\left(\frac{5}{\sqrt{3}}\right) = \frac{250\pi}{9\sqrt{3}}\ pol^3$

Passo 5 (opcional)

$$V''(h) = \frac{\pi}{3}(-6h) = -2\pi h$$
$$V''\left(\frac{5}{\sqrt{3}}\right) = -\frac{10\pi}{\sqrt{3}} < 0$$

Pelo teste da segunda derivada V tem um máximo relativo em $h = 5/\sqrt{3}$. Já que ele é o único extremo relativo, este ponto é um máximo absoluto.

10.1 Problemas suplementares

1. Um campo aberto é delimitado por um lago com uma linha costeira reta. Uma limitação retangular deve ser construída usando 500 pés de cerca ao longo de três lados e o lago como um limite natural no quarto lado. Quais dimensões irão maximizar a área fechada? Qual é a área máxima?

2. Ryan tem 800 pés de cerca. Ele deseja formar um terreno retangular e dividi-lo em três seções duas cercas paralelas a um lado. Quais devem ser as dimensões da área de forma a maximizar a área encerrada?

3. 20 metros de cerca devem ser dispostos em forma de um triângulo retângulo. Quais devem ser suas dimensões para maximizar a área fechada?

4. Um pedaço de arame de 100 polegadas de comprimento deve ser usado para formar um quadrado e/ou um círculo. Determine sua área (a) máxima e (b) mínima áreas combinada.

5. Encontre a área máxima de um retângulo inscrito em um semicírculo de raio de 5 polegadas se sua base estiver ao longo do diâmetro do semicírculo.

6. Uma caixa aberta deve ser construída a partir de um pedaço de 12 x 12 polegadas de papelão, cortando quadrados de tamanho igual dos quatro cantos e dobrando os lados. Determine o tamanho do recorte que maximiza o volume da caixa.

7. Uma janela deve ser construída na forma de um triângulo isósceles no topo de um retângulo. Se o seu perímetro é de 600 cm, Qual é a área máxima possível da janela?

8. Os regulamentos postais exigem que a soma do comprimento e do barbante de um pacote retangular não pode exceder 108 polegadas (barbante é a maior perímetro das 3 possibilidades de uma caixa). Qual é o volume máximo de um pacote com extremidades quadradas que atenda a este critério ?

9. Um retângulo está inscrito em um triângulo retângulo cujos lados são 5, 12 e 13 polegadas. Dois lados adjacentes do retângulo ficam ao longo das pernas do O triângulo. Quais são as dimensões do retângulo de máximo área? Qual é a área máxima?

10. Encontre as dimensões do - circular direito de máximo volume que pode ser inscrito em um cone circular reto cujo raio é 3 polegadas e cuja altura é 10 polegadas. Qual é o volume máximo?

11. Qual é a quantidade mínima de cerca necessária para construir um recinto retangular contendo 1.800 pés^2 usando um rio como um limite natural de um lado?

12. Uma caixa retangular aberta deve ter uma base com o dobro do comprimento e largura. Se seu volume deve ser de 972 cm^3, quais dimensões irão minimizar o quantidade de material utilizado em sua construção?

13. Encontre os pontos na parábola $y = x^2$ mais próximos do ponto (0, 1).

14. Uma editora deseja imprimir um livro cujas páginas tenham cada uma área de 96 in^2. As margens devem ser de 1 polegada em cada um dos três lados e 2 pol no quarto lado para permitir espaço para a encadernação. Quais dimensões para as folhas vai permitir a área máxima para a região impressa?

15. Uma lata cilíndrica fechada deve ter um volume de 1000 pol^3. Quais dimensões irão minimizar sua área de superfície?

16. Uma lata cilíndrica fechada deve ter um volume de 1000 pol^3. A superfície lateral deve ser construída a partir de uma peça retangular de metal e sua parte superior e inferior devem ser estampadas com peças quadradas de metal e o resto do quadrado descartado. Quais dimensões irão minimizar a quantidade de metal necessária na construção da lata?

17. Um retângulo deve ser inscrito na elipse
$$\frac{x^2}{200} + \frac{y^2}{50} = 1$$
Determine sua área máxima possível.

Os exercícios suplementares estão publicados no livro às págs 100 a 128.

Capítulo 11

Funções Trigonométricas

Anteriormente discutiram-se técnicas para resolver problemas de variação e de máximos e mínimos. Neste aqui, vão se discutir problemas envolvendo funções trigonométricas. O uso da trigonometria frequentemente pode simplificar as soluções. As derivadas de funções trigonométricas têm papel importante nas soluções

$$\frac{d}{dx}sin(x) = cos(x) \qquad \frac{d}{dx}cos(x) = -sen(x)$$
$$\frac{d}{dx}tan(x) = sec^2(x) \qquad \frac{d}{dx}cot(x) = -csc^2(x)$$
$$\frac{d}{dx}sec(x) = sec(x)tan(x) \qquad \frac{d}{dx}csc(x) = -csc(x)cot(x)$$

É importante lembrar que as expressões acima só são verdade se x está expresso em radianos. Dessa maneira qualquer angulos ou variações em graus precisam ser convertidas a radianos. Isto é feito facilmente ao lembrar que 180 graus equivalem a π radianos.

Expressões pitagóricas são usadas para simplificar expressões trigonométricas:

$$sen^2(x) + cos^2(x) = 1$$
$$tan^2(x) + 1 = sec^2(x)$$
$$cot^2(x) + 1 = csc^2(x)$$

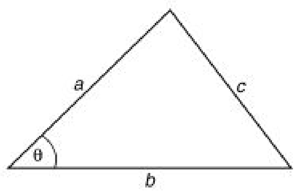
Vale lembrar que

$$\frac{1}{sen(x)} = csc(x)$$
$$\frac{1}{cos(x)} = sec(x)$$
$$\frac{1}{tan(x)} = ctg(x)$$
$$\frac{1}{csc(x)} = sen(x)$$
$$\frac{1}{sec(x)} = cos(x)$$
$$\frac{1}{ctg(x)} = tan(x)$$

Além disso, um velho macete de cursinho ajuda:

$$SEn = \frac{cateto\ SEparado}{hipotenusa}$$
$$COs = \frac{cateto\ COlado}{hipotenusa}$$
$$tan = \frac{seno}{cosseno}$$

A lei dos cossenos é as vezes necessária para lidar com problemas que envolvam triângulos não retângulos

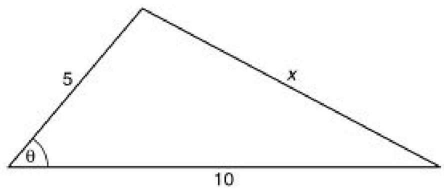


$$c^2 = a^2 + b^2 - 2abcos\theta$$

Observe que se $\theta = \pi/2$ a lei dos cossenos se reduz ao teorema de Pitágoras: $c^2 = a^2 + b^2$

11.1 Problemas de Variação

Exemplo 1 Dois lados de um triângulo são de 5 e 10 polegadas, respectivamente. O ângulo entre eles está sendo incrementado à razão de 5^o por minuto. Quão rápido o terceiro lado está aumentando quando o ângulo é de 60^0 ?



Seja θ o ângulo entre os lados de comprimento 5 e 10 e seja x o comprimento do terceiro lado. Em qualquer cálculo envolvendo derivadas, ângulos e taxas devem estar em radianos. Dado $\dfrac{d\theta}{dt} = \dfrac{\pi}{36}$ radianos por minuto. Achar $\dfrac{dx}{dt}$ quando $\theta = \dfrac{\pi}{3}$. Pela leis dos cossenos

$$x^2 = 5^2 + 10^2 - 2.5.10.cos\theta$$

$$x^2 = 125 - 100cos\theta$$

Diferenciando em relação a t

$$2x\frac{dx}{dt} = 0 - 100(-sen\theta)\frac{d\theta}{dt}$$

$$x\frac{dx}{dt} = 50sen\theta\frac{d\theta}{dt}$$

Necessitamos o valor de x quando $\theta = \pi/3$, lembrando que $sen\frac{\pi}{3} = \frac{\sqrt{3}}{2} \quad e \quad cos\frac{\pi}{3} = \frac{1}{2}$, então

$$x^2 = 125 - 50 = 75$$

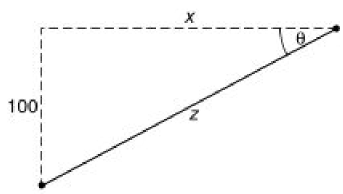
$$x = 5\sqrt{3}$$

Substituindo

$$5\sqrt{3}\frac{dx}{dt} = 50.\frac{\sqrt{3}}{2}.\frac{\pi}{36}$$

$$\frac{dx}{dt} = \frac{5\pi}{36} \ \ pol/min$$

Exemplo 2 Um pássaro está voando 100 pés acima do solo, movendo-se em direção horizontal à taxa de 10 pés/segundo. Quão rápido muda o ângulo entre a visada (observador - pássaro) e a horizontal quando esta visada tem 300 pés de comprimento ?



Dado $\dfrac{dx}{dt} = 10$ achar $\dfrac{d\theta}{dt}$ quando $z = 300$. Poder-se-ia utilizar a relação $tan\theta = \dfrac{100}{x}$, mas é mais fácil trabalhar quando x aparece no numerador da fração. Por isso prefere-se aqui a função *cotangente*

$$cot\theta = \frac{x}{100}$$

$$x = 100cot\theta$$

Segue-se que

$$\frac{dx}{dt} = 100(-csc^2\theta)\frac{d\theta}{dt}$$

$$\frac{dx}{dt} = \frac{-100}{sen^2\theta}.\frac{d\theta}{dt}$$

Resolvendo para $\dfrac{d\theta}{dt}$, fica

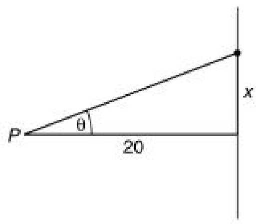
$$\frac{d\theta}{dt} = \frac{-sen^2\theta}{100}.\frac{dx}{dt}$$

Pelo diagrama pode-se ver que quando $z = 300$, o $sen\theta = \frac{1}{3}$

$$\frac{d\theta}{dt} = -\frac{1/9}{100}.10 = -\frac{1}{90}$$

ou seja θ está diminuindo à razão de $1/90$ rad/seg.

Exemplo 3 Um carro de polícia está a 20 pés de um muro. Seu giroflex faz 1 revolução por segundo. Quão rápido o feixe se move quando ele está mais próximo do carro de polícia ?



Dado $\dfrac{d\theta}{dt} = 2\pi$, já que 1 revolução é uma volta completa, o valor de $\dfrac{d\theta}{dt} = 2\pi rd/seg$. Achar $\dfrac{dx}{dt}$ quando $\theta = 0$, já que o feixe está mais próximo do carro quando $\theta = 0$.

$$tan\theta = \frac{x}{20}$$

$$x = 20tan\theta$$

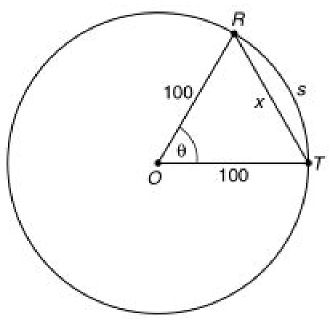
derivando em relação a x

$$\frac{dx}{dt} = 20sec^2\theta.\frac{d\theta}{dt}$$

Substituindo $\theta = 0$ e $\dfrac{d\theta}{dt} = 2\pi$ pode-se calcular $\dfrac{dx}{dt}$

$$\frac{dx}{dt} = 20.1^2.2\pi = 40\pi \ p\acute{e}s/seg.$$

Exemplo 4 Um corredor e seu treinador estão juntos em uma pista circular de raio 100m. Quando o treinador dá o sinal, o corredor inicia a corrida sobre a pista, correndo à velocidade de 10m/seg. Como varia a distância entre treinador e corredor quando o corredor completou 1/4 da volta ?



Seja x a distância reta entre corredor e treinador e seja s a distância corrida sobre a pista. O ângulo formado pelos 2 raios é θ. Dados $\frac{ds}{dt} = 10$, achar $\frac{dx}{dt}$ quando $\theta = \frac{\pi}{2}$ (já que 1 volta completa é $360^o = 2\pi$ e portanto $1/4$ de volta é $\pi/2$). Pela lei dos cossenos,

$$x^2 = 100^2 + 100^2 - 2(100)(100)cos\theta$$
$$x^2 = 20.000 - 20.000cos\theta$$

Diferenciando

$$2x\frac{dx}{dt} = 20.000sen\theta\frac{d\theta}{dt}$$
$$x\frac{dx}{dt} = 10.000sen\theta\frac{d\theta}{dt}$$

Para calcular $\frac{dx}{dt}$ quando $\theta = \frac{\pi}{2}$ se necessitam os valores de x e $\frac{d\theta}{dt}$. Quando o corredor correu $1/4$ da volta, $\theta = \pi/2$. O triângulo $\triangle ORT$ se torna reto.

$$x^2 = 100^2 + 100^2 = 20.000$$
$$x = 100\sqrt{2}$$
$$s = r\theta = 100\theta$$

diferenciando em relação a t

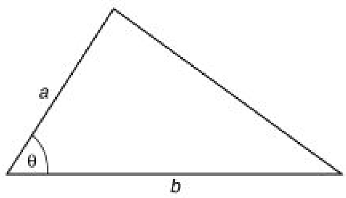
$$\frac{ds}{dt} = 100\frac{d\theta}{dt}$$
$$10 = 100\frac{d\theta}{dt}$$
$$\frac{d\theta}{dt} = \frac{1}{10}$$

Agora pode-se computar $\frac{dx}{dt}$

$$x\frac{dx}{dt} = 10.000sen\theta\frac{d\theta}{dt}$$
$$100\sqrt{2}\frac{dx}{dt} = 10.000.\overbrace{sen\left(\frac{\pi}{2}\right)}^{1}.\left(\frac{1}{10}\right)$$
$$100\sqrt{2}\frac{dx}{dt} = 1000$$
$$\frac{dx}{dt} = \frac{1000}{100\sqrt{2}} = \frac{10}{\sqrt{2}} = 5\sqrt{2}\ m/seg$$

11.2 Problemas de máximos e mínimos

Exemplo 5 Qual a maior área possível de um triângulo de lados a e b ?



Seja θ o ângulo entre os 2 lados. A área do triângulo é

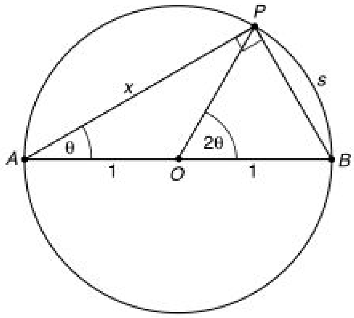
$$A(\theta) = \frac{1}{2}ab.sen(\theta) \qquad 0 \leqslant \theta \leqslant \pi$$

Como a e b são constantes (o que varia é θ), $A'(\theta) = \frac{1}{2}ab.cos\theta$. Igualando a zero e resolvendo, obtém-se

$$0 = \frac{1}{2}ab.cos\theta$$
$$0 = cos\theta$$
$$\theta = \frac{\pi}{2}$$

Como $A(\theta)$ é função contínua e $A(0) = A(\pi) = 0$ a área absoluta máxima ocorre quando $\theta = \frac{\pi}{2}$ (um ângulo reto). Como $sen\frac{\pi}{2} = 1$, a área máxima é $A(\frac{\pi}{2}) = \frac{1}{2}ab$.

Exemplo 6 Um homem no ponto A na margem de um lago circular de raio 1 milha, deseja alcançar o ponto B na borda do lago diametralmente oposto a A. Se ele pode remar um barco a 3 $milhas/hora$ e correr a 6 $milhas/hora$ em que ângulo θ com o maior diâmetro ele deve remar para chegar a B no menor tempo possível ?



Seja P o ponto onde o bote alcança a borda. Como o triângulo $\triangle APB$ está inscrito em um semicírculo, ele é quadrado com hipotenusa 2. Daqui

$$\frac{x}{2} = cos\theta$$

$$x = 2cos\theta$$

Um teorema da geometria afirma que se um ângulo central é medido pelo arco interceptado, um ângulo inscrito é medido pela metade do arco interceptado. Como $\overset{\frown}{PB}$ é o arco interceptado para $\angle POB$ e para $\angle PAB$ segue-se que $\angle POB$ é duas vezes maior do que $\angle PAB$ ou 2θ. O comprimento do arco s é $s = r(2\theta) = 2\theta$.

Se r é uma taxa de movimento (velocidade) constante e d é a distância viajada, o tempo t da viagem é $t = \frac{d}{r}$. Daqui que o tempo para fazer a jornada entre A e B como função de θ é

$$t = t_{remando} + t_{correndo} = \frac{x}{3} + \frac{s}{6}$$

$$t(\theta) = \frac{2cos\theta}{3} + \frac{2\theta}{6}$$

$$= \frac{2}{3}cos\theta + \frac{\theta}{3} \qquad 0 \leqslant \theta \leqslant \frac{\pi}{2}$$

Diferenciando para achar os pontos críticos

$$t'(\theta) = -\frac{2}{3}sen\theta + \frac{1}{3}$$

$$0 = -\frac{2}{3}sen\theta + \frac{1}{3}$$

$$\frac{2}{3}sen\theta = \frac{1}{3}$$

$$sen\theta = \frac{1}{2}$$

$$\theta = \frac{\pi}{6}$$

Como $t(\theta) = \frac{2}{3}cos\theta + \frac{\theta}{3}$ é contínua de θ o valor mínimo deve ocorrer em um número crítico ou nos limites do intervalo $[0, \pi/2]$.

$$t(0) = \frac{2}{3} \approx 0.667$$

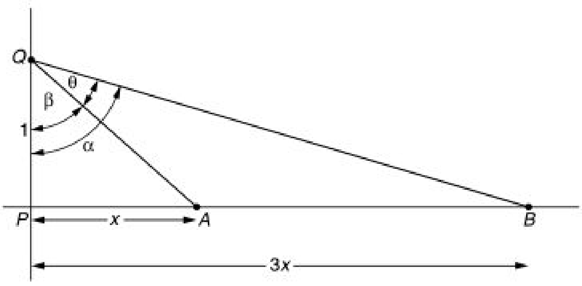
$$t(\frac{\pi}{6}) = \frac{2}{3}.\frac{\sqrt{3}}{2} + \frac{\pi}{18} = \frac{6\sqrt{3}+\pi}{18} \approx 0.752$$

$$t(\frac{\pi}{2}) = \frac{\pi}{6} \approx 0.524$$

Neste caso o menor tempo para ir de A a B e correndo completamente em redor do lago, sem pegar o barco.

Exemplo 7 Dois veículos A e B começam no ponto P e viajam para leste em taxas de 10 km/h e 30 km/h respectivamente. Um observador em Q 1 quilômetro ao norte de P enxergando os 2 veículos. Qual o maior ângulo de visão entre A e B ?

Seja θ o ângulo da visão. É conveniente introduzir os ângulos α e β conforme a figura



Dado que B viaja três vezes mais rápido que A, PB será $3x$ quando $PA = x$. Como $|PQ| = 1$, $tan\alpha = 3x$ e $tan\beta = x$. Como $\theta = \alpha - \beta$ e lembrando

$$tan\ (\alpha + \beta) = \frac{tan\ \alpha + tan\ \beta}{1 - tan\ \alpha tan\ \beta}$$

$$tan\ (\alpha - \beta) = \frac{tan\ \alpha - tan\ \beta}{1 + tan\ \alpha tan\ \beta}$$

segue-se que

$$tan\theta = tan(\alpha - \beta) = \frac{tan\alpha - tan\beta}{1 + tan\alpha tan\beta}$$

$$= \frac{3x - x}{1 + (3x)x} = \frac{2x}{1 + 3x^2} \qquad 0 \leqslant x \leqslant \infty$$

Como se quer maximizar θ, se necessita $\frac{d\theta}{dx}$.

$$sec^2\theta.\frac{d\theta}{dx} = \frac{(1 + 3x^2)(2) - (2x)(6x)}{(1 + 3x^2)^2}$$

$$= \frac{2 - 6x^2}{(1 + 3x^2)^2}$$

Igualando a derivada $\frac{d\theta}{dx} = 0$

$$0 = \frac{2 - 6x^2}{(1 + 3x^2)^2}$$

$$0 = 2 - 6x^2$$

$$6x^2 = 2$$

$$x = \frac{\sqrt{3}}{3}$$

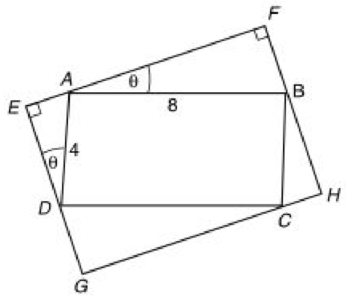
Como $sec^2\theta > 0$, $\frac{d\theta}{dx}$ é positivo se $x < \frac{\sqrt{3}}{3}$ e negativo se $x > \frac{\sqrt{3}}{3}$. Pelo teste da primeira derivada θ tem um máximo relativo em $\sqrt{3}/3$. Já que é o único extremo relativo no intervalo $[0, \infty]$ ele deve ser um máximo absoluto.

Como $tan\ \theta = \frac{2x}{1 + 3x^2}$, e $x = \frac{\sqrt{3}}{3}$ tem-se

$$tan\ \theta = \frac{\frac{2\sqrt{3}}{3}}{1 + 3.\frac{1}{3}} = \frac{\sqrt{3}}{3}$$

O valor correspondente de θ é $\pi/6$. O maior ângulo de vista é 30^o.

Exemplo 8 Ache a área máxima de um retângulo circunscrito sobre um retângulo fixo de comprimento 8 por 4.



Seja $\theta = \angle FAB$ como mostrado na figura. Por geometria $\angle EDA = \theta$. As distâncias $|EA| = 4sen\theta$, $|AF| = 8cos\theta$, $|ED| = 4cos\theta$ e $|DG| = |FB| = 8sen\theta$. A área do retângulo circunscrito é o produto $(|ED|+|DG|)(|EA|+|AF|)$. Daqui

$$A(\theta) = (4cos\theta + 8sen\theta)(4sen\theta + 8cos\theta)$$

$$= 32cos^2\theta + 32sen^2\theta + 80sen\theta cos\theta$$

Lembrando que $sen^2\alpha + cos^2\alpha = 1$ e que $sen2\beta = 2sen\beta cos\beta$ temos

$$= 32 + 40sen2\theta \qquad 0 \leqslant \theta \leqslant \frac{\pi}{2}$$

$$A'(\theta) = 80cos2\theta$$

$$0 = 80cos2\theta$$

$$0 = cos2\theta$$

$$2\theta = \frac{\pi}{2}$$

$$\theta = \frac{\pi}{4}$$

Observe que $A(\theta)$ é contínua. Já que $A(0) = A(\frac{\pi}{2}) = 32$ e $A(\frac{\pi}{4}) = 72$ a área máxima é 72 e ocorre quando $\theta = \frac{\pi}{4}$.

11.3 Problemas suplementares

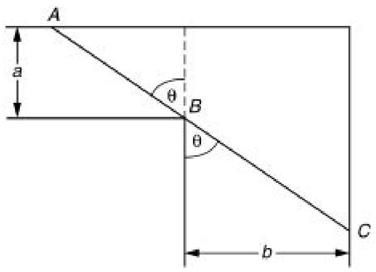
1. Uma escada de 3 metros de comprimento está apoiada na lateral de um prédio. Se o pé da escada desliza para longe da parede a uma taxa de 2 pés / min, quão rápido é o ângulo entre a escada e o prédio está variando quando o pé da escada está a 6 pés de distância do prédio?

2. Os dois lados de um triângulo têm 3 e 4 polegadas de comprimento. Se o ângulo entre eles estão aumentando a uma taxa de 2^o por segundo, quão rápido aumenta a área do triângulo quando o ângulo é de 45 °?

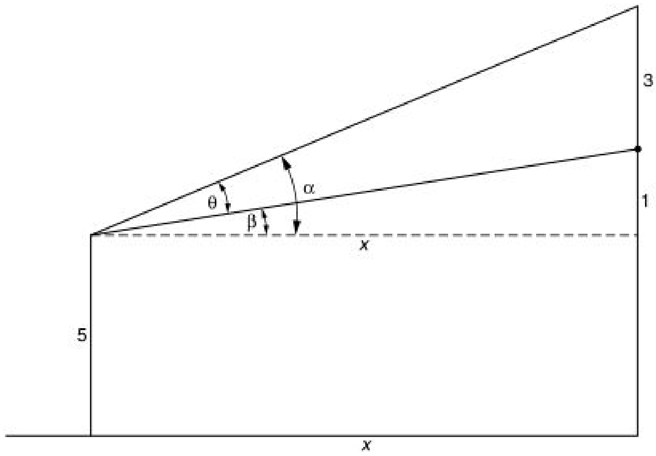
3. Uma câmera de televisão está localizada a 5000 pés da base de um foguete na plataforma de lançamento. A câmera é projetada para seguir o caminho vertical de o foguete. Se a velocidade do foguete for 500 pés / s quando ele aumentar 2.000 pés, quão rápido é o ângulo de elevação da câmera mudando neste instante?

4. Um farol está situado a 2 km de uma praia e seu farol gira a uma taxa de 3 rotações por minuto. Se P é o ponto a praia mais próxima do farol, quão rápido é o feixe de luz mover-se ao longo da praia a 1 km de P?

5. Dois corredores de larguras a e b se cruzam em ângulos retos. Qual é o comprimento do tubo mais longo que pode ser transportado horizontalmente ao virar da esquina?



6. Uma pintura de 3 pés de altura está pendurada em uma parede com a parte inferior do pintura 6 pés acima do chão. A que distância da parede deve Lindsay, cujos olhos estão a 5 pés do chão, fique de pé para obter o melhor vista da pintura? (A melhor visão ocorre quando o ângulo de visão de baixo para cima da pintura é maximizado.)



As soluções para estes problemas estão no original às págs 142 a 149.

Capítulo 12

Funções exponenciais

Se b é positivo e x é real, a função $f(x) = b^x$ é chamada uma função exponencial. Se a base do expoente é $e \approx 2.71828$, a função é conhecida como exponencial natural. Sua inversa é a função logarítmica normalmente representada por $ln\ x$. Algumas de suas propriedades são

$$y = e^x \Leftrightarrow x = ln\ y$$

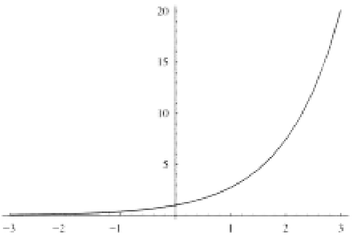
$$\begin{array}{cc} e^0 = 1 & ln\ 1 = 0 \\ e^1 = e & ln\ e = 1 \\ e^{x+y} = e^x e^y & ln\ xy = ln\ x + ln\ y \\ e^{x-y} = \dfrac{e^x}{e^y} & ln\ \dfrac{x}{y} = ln\ x - ln\ y \\ (e^x)^y = e^{xy} & ln\ x^y = y\ ln\ x \end{array}$$

Há ainda algumas consequências das propriedades acima,

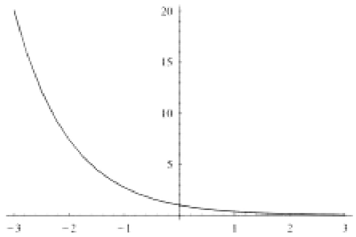
$$ln\ \frac{1}{x} = -ln\ x \qquad -ln\ \frac{x}{y} = ln\ \frac{y}{x}$$

$$e^{ln\ x} = x \qquad ln(e^x) = x$$

Os gráficos das funções são



$f(x) = e^x \quad -3 \leq x \leq 3$



$f(x) = e^{-x} \quad -3 \leq x \leq 3$

Estas funções exercem um papel fundamental em problemas de crescimento ou decaimento exponencial. Observe que

$$\lim_{x \to -\infty} e^x = 0 \quad e \quad \lim_{x \to \infty} e^{-x} = 0$$

12.1 Crescimento e decaimento exponenciais

Se uma substância cresce ou decai em uma taxa proporcional ao seu tamanho, seu crescimento pode ser descrito pela equação

$$\frac{dy}{dt} = ky$$

Onde y é a quantidade da substância e k é uma constante de proporcionalidade. Valores positivos de k indicam crescimento e negativos indicam decaimento.

Para resolver em y, reescreve-se a equação na forma

$$\frac{1}{y}\frac{dy}{dt} = k$$

e segue-se pela regra da cadeia

$$\frac{d}{dt} ln\ y = k$$

Antidiferenciando (integrando) obtem-se

$$ln\ y = kt + C$$

Se $y = y_0$ quando $t = 0$ segue-se que $y_0 = C$ e então pode-se escrever

$$ln\ y = kt + ln\ y_0$$

$$ln\ y - ln\ y_0 = kt$$

$$ln\ \frac{y}{y_0} = kt$$

$$\frac{y}{y_0} = e^{kt}$$

$$y = y_0\ e^{kt}$$

Se o valor de y_0 é conhecido e o valor de k pode ser determinado, pode-se representar a quantidade de uma substância como função do tempo.

Exemplo 1 Uma cultura de bactérias tem uma população inicial de 500. Após 4 horas a população cresceu a 1000. Assumindo que a cultura cresce a uma taxa proporcional a seu tamanho, ache a função que representa o tamanho da população após t horas e determine o tamanho após 6 horas.

Seja $y(t)$ representando o tamanho da população de bactérias após t horas. Pode-se escrever

$$y(t) = y_0\ e^{kt}$$

$$y(t) = 500\ e^{kt}$$

Como $y(4) = 1000$, segue-se que

$$1000 = 500\ e^{4t}$$

$$2 = e^{4t}$$

$$4k = ln\ 2$$

$$k = \frac{1}{4}\ ln\ 2$$

Então

$$y(t) = 500\ e^{(\frac{1}{4}\ ln\ 2)t}$$

$$= 500(2)^{t/4}$$

Para determinar a população após 6 horas ($t = 6$), calcula-se

$$y(6) = 500.2^{1.5} \approx 1414$$

O decaimento ocorre quando uma substância se deteriora quando o tempo passa. Se a taxa é proporcional à quantidade de substância, a massa satisfaz a equação

$$\frac{dy}{dt} = -ky$$

Nesta equação, k é uma constante positiva. Resolvendo a equação de maneira similar à já vista, tem-se

$$y = y_0\ e^{-kt}$$

Exemplo 2 Uma substância radioativa tem massa de 100mg. Após 10 anos ela decaiu para uma de 75mg. Qual será a massa após outros 10 anos ?

Seja $y(t)$ a função de decaimento. $y(0) = 100$ e $y(10) = 75$. Deve-se determinar $y(20)$.

$$y(t) = y_0 \ e^{-kt}$$

$$y(t) = 100 \ e^{-kt}$$

Primeiro, precisa-se obter k. Isto pode ser feito pela substituição em $y(10)$.

$$y(10) = 100 \ e^{-10k} = 75$$

$$k = \frac{1}{10} ln \frac{4}{3}$$

Note que k é positivo. Agora, a função $y(t)$ pode ser determinada

$$y(t) = 100 \ e^{-(\frac{1}{10} ln \frac{4}{3})r}$$

$$= 100 \left(\frac{4}{3}\right)^{-\frac{t}{10}}$$

Para resolver o problema façamos $t = 20$ e

$$y(20) = 100 \left(\frac{4}{3}\right)^{-2} \approx 56.25 \ mg$$

A meia-vida de uma substância radiotiva é o tempo que demora até a substância decair à metade de sua massa original. Se $y(t) = y_0 \ e^{-kt}$, a meia-vida é o valor de t para o qual $y(t) = \frac{1}{2} y_0$

$$\frac{1}{2} y_0 = y_0 \ e^{-kt}$$

$$\frac{1}{2} = e^{-kt}$$

$$-kt = ln \ \frac{1}{2}$$

$$kt = ln \ 2$$

$$t = \frac{ln \ 2}{k}$$

Costuma-se representar a meia vida de uma substância pela letra T, então $T : \dfrac{ln \ 2}{k}$. Se a meia vida T é conhecida, pode-se obter k.

Exemplo 3 A meia vida do ^{14}C (carbono 14) é de 5730 anos. Quanto tempo vai demorar até uma amostra de 100mg de ^{14}C decair até 90mg ?

$$y(t) = y_0 \ e^{-kt}$$

$$k = \frac{ln \ 2}{T} = \frac{ln \ 2}{5730} \approx 0.000121$$

Esta aproximação pode induzir a erro, razão pela qual é melhor adiar a solução.

$$y(t) = 100 \ e^{-\left(\frac{ln \ 2}{5730}\right)t}$$

Para determinar o tempo na redução para 90mg, fazemos $y(t) = 90$

$$90 = 100 e^{\left(\frac{ln2}{5730}\right)t}$$

$$0.9 = e^{\left(\frac{ln2}{5730}\right)t}$$

$$t \approx 870.98 \ anos$$

Veja a interpretação do sinal negativo na sessão APLX abaixo:

```
      ⍟0.9
¯0.1053605157
      ⍟2
0.6931471806
¯0.1053605157×5730
¯603.715755
¯603.715755÷0.6931471806
¯870.9777258
```

12.1.1 Carbono 14

O Carbono 14 é um isótopo radiotivo do carbono. O carbono 12 é um isótopo estável. Ambos são achados na atmosfera. Plantas que absorvem dióxido de carbono do ar, têm a mesma razão entre ^{14}C e ^{12}C que há na atmosfera. O mesmo é verdade para animais que comem plantas. Quando uma planta ou um animal morrem, não há mais absorção de dióxido de carbono. Seu ^{14}C decai, mas o ^{12}C não. Medindo a razão entre ^{14}C e ^{12}C pode-se aproximar a idade de um fóssil comparando este número com a razão que há hoje na atmosfera. Naturalmente, assume-se a hipótese de que não houve variação nesta taxa com o passar dos anos. A taxa diminui exponencialmente e tem a mesma meia-vida do ^{14}C que é de 5730 anos.

Exemplo 14 Qual a idade de um fóssil cuja razão $\frac{^{14}C}{^{12}C}$ é de 10% daquela encontrada na atmosfera hoje ?

Seja $r(t)$ a razão entre ^{14}C e ^{12}C presente após t anos. Como $r(t)$ decai exponencialmente,

$$t(t) = r_0 \ e^{-kt}$$

Para achar k lembremos que $k = \dfrac{ln2}{5730}$

$$r(t) = r_0 \ e^{-(\frac{ln2}{5730})t}$$

$$\frac{r(t)}{r_0} = e^{-(\frac{ln2}{5730})r}$$

Como $\frac{r(t)}{r_0} = 10\% = 0.1$ tem-se

$$0.1 = e^{-(\frac{ln \ 2}{5730})t}$$

$$ln \ 0.1 = -\left(\frac{ln \ 2}{5730}\right) t$$

$$t = \frac{-5730(ln \ 0.1)}{ln \ 2} \approx 19034.6$$

O fóssil tem aproximadamente 19000 anos de idade.

12.1.2 Juros compostos contínuos

Se P dólares são investidos no banco, em uma taxa anual de juros compostos r em períodos de n vezes por ano, o valor retornado após t anos é de

$$A = P\left(1 + \frac{r}{n}\right)^{nt}$$

Juros trimestrais, mensais ou diários usam os valores de $n = 4$, 12 e 365 respectivamente (Alguns bancos usam anos de 360 dias, mas a diferença é irrisória). Quando se faz $n \to \infty$ obtem-se o chamado juro composto contínuo e o juro composto após t anos, se torna

$$A = \lim_{n\to\infty}\left[P\left(1 + \frac{r}{n}\right)^{nt}\right]$$

$$= P\left[\lim_{n\to\infty}\left(1 + \frac{r}{n}\right)^{n}\right]^{t}$$

Pode-se mostrar que $\lim_{n\to\infty}\left(1 + \frac{r}{n}\right)^{n} = e^r$ então a quantidade de dinheiro após t anos é de $A = Pe^{rt}$ com crescimento exponencial.

Exemplo 5 Calcule a quantidade de dinheiro no banco após 10 anos quando 1000 dólares são aplicados trimestralmente, mensalmente, diariamente e continuamente investidos com taxa anual de 6%.

Trimestralmente

$$A = 1000\left(1 + \frac{0.06}{4}\right)^{40} = \$1814.02$$

Mensalmente

$$A = 1000\left(1 + \frac{0.06}{12}\right)^{120} = \$1819.40$$

Diariamente

$$A = 1000\left(1 + \frac{0.06}{365}\right)^{3650} = \$1822.03$$

Continuamente

$$A = 1000\ e^{0.6} = \$1822.12$$

Exemplo 6 Quanto tempo deve ficar investido um dinheiro até ele dobrar com aplicação contínua e taxa anual de 5%?

A quantidade inicial é irrelevante. A coisa importante é que ao final deve-se ter o dobro do início. Ou seja, se se começa com P dólares, ao final deve-se ter $2P$.

$$A = Pe^{rt}$$
$$2P = Pe^{0.05t}$$
$$2 = e^{0.05t}$$
$$ln\ 2 = 0.05t$$
$$t = \frac{ln\ 2}{0.05} \approx 13.86\ anos$$

Exemplo 7 Quanto dinheiro Ariel deve investir em um banco que paga 8% ao ano em juros compostos contínuos se ela quer comprar um carro que custa \$ 20.000 em 4 anos ?

A equação $A = Pe^{rt}$ equivale a $P = \frac{A}{e^{rt}} = Ae^{-rt}$.

$$P = Ae^{-rt}$$
$$= 20000e^{-(0.08)(4)}$$
$$= 20000e^{-0.32}$$
$$= 20000(0.726149)$$
$$= \$14522.98$$

12.1.3 Modelos exponenciais adicionais

Funções exponenciais aparecem em muitos problemas de ciências, administração, economia, medicina e sociologia. Uma função importante, chamada função logística é frequentemente usada para estudar o crescimento da população limitada por fatores ambientais. Ela também é usada para analisar o crescimento de epidemias e a propagação de boatos.

A forma geral da função logística é

$$P(t) = \frac{B}{1 + Ae^{-Bkt}}$$

onde A, B e k são em geral determinadas experimentalmente.

Exemplo 8 Ambientalistas predisseram que em t anos a população de veados em uma floresta deve ser de $P(t) = \frac{20}{1+4e^{-2t}}$ milhares de veados.

(a) Qual a população atual ?

(b) Qual será a população ao final de cada um dos primeiros três anos ?

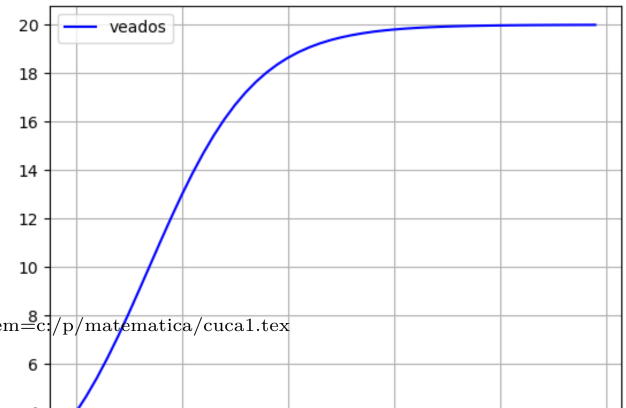
(c) Qual será a taxa de crescimento da população após 3 anos ?

(d) Quando a taxa de crescimento começa a declinar ?

(e) Qual será a população limite de veados na floresta ?

(f) Quando a população atingirá 80% do limite ?

O gráfico da população está em




```
import numpy as np
import matplotlib.pyplot as plt
import math
def P(B,A,k,t):
  return (B)/(1+A*math.exp(-B*k*t))
def p159():
  t=np.arange(0,5,0.1)
  pp=np.zeros(50,float)
  for x in range(50):
    pp[x]=P(20,4,0.1,t[x])
```

```
    plt.plot(t,pp,'b-',label='veados')
    plt.grid()
    plt.legend()
    plt.show()
p159()
```

(a) P(0)=4 mil veados

(b) P(1)=12.98 mil
P(2)=18.63 mil
P(3)=19.80 mil

(c) A taxa do crescimento da população é dada por $r(t) = P'(t)$

$$P(t) = \frac{20}{1+4e^{-2t}} = 20(1+4e^{-2t})^{-1}$$

$$P'(t) = -20(1+4e^{-2t})^{-2}(-8e^{-2t})$$

$$= \frac{160e^{-2t}}{(1+4e^{-2t})^2}$$

Após 3 anos, ($t = 3$) a variação é

$$P'(3) = \frac{160e^{-6}}{(1+4e^{-6})^2} \approx 0.389$$

A taxa de crescimento é 389 veados por ano.

(d) A taxa do crescimento da população aumenta quando $r'(t) > 0$ e diminui quando $r'(t) < 0$. A taxa começa a declinar quanto $r'(t) = P''(t) = 0$

$$P''(t) = \frac{(1+4e^{-2t})^2 \frac{d}{dt}(160e^{-2t}) - (160e^{-2t})\frac{d}{dt}(1+4e^{-2t})^2}{(1+4e^{-2t})^4}$$

$$= \frac{-320e^{-2t}(1-4e^{-2t}}{(1+4e^{-2t})^3}$$

$P''(t) = 0$ se e somente se $1 - 4e^{-2t} = 0$ e $t = \frac{1}{2}ln\ 4 \approx 0.693\ ano$

veja a continuação na pág 161

Exemplo 9 Veja na pág. 162 e seguintes

12.2 Problemas Suplementares

1. A taxa de crescimento de uma cultura de bactérias é proporcional ao número de bactérias presentes. Se uma população cujo tamanho é inicialmente 100 e a bactéria dobra a cada 2 horas,
(a) Quantas bactérias haverá após 3 horas?
(b) Quanto tempo vai demorar para a cultura crescer até 5000 bactérias?

2. Uma hora depois que uma colônia de bactérias começa a crescer, um cientista determina que há 9.000 bactérias presentes. Depois de outro hora, existem 12.000 bactérias. Quantas bactérias estavam presentes inicialmente?

3. Uma substância radioativa cuja massa é de 200 mg decairá para 180 mg após 12 anos. Determine a meia-vida desta substância.

4. 10 mg de uma substância radioativa com meia-vida de 20 horas é injetado na corrente sanguínea de um paciente. O paciente retorna ao instalação médica após 24 horas e um técnico determina que há 2 mg da substância no pâncreas do paciente. Quanto há da substância no restante do corpo do paciente corpo?

5. Um papiro egípcio é descoberto e verifica-se que a proporção de 14C a 12C é 65 por cento da proporção conhecida de 14C a 12C no ar hoje. A meia-vida de 14C é 5730 anos. Quantos anos tem o papiro?

6. Quanto dinheiro Alexis terá no banco após 3 anos se ela investe $ 700 a 8% compostos continuamente?

7. Quanto tempo levará para triplicar o dinheiro se for composto continuamente a uma taxa de 10 por cento?

8. Trevor gostaria de comprar um anel de noivado que é vendido por \$ 8.000. Quanto ele deve colocar em uma conta de poupança hoje que paga 5 por cento compostos continuamente para ter dinheiro suficiente para comprar o anel em 6 meses?

9. Um estudo ecológico mostra que um lago pode suportar um máximo população de 5000 peixes. A função logística que governa os peixes a população é
$$P(t) = \frac{5000}{1 + 4e^{-0.4t}}$$
onde t é medido em meses.
(a) Quantos peixes foram inicialmente colocados no lago?
(b) Quantos peixes existem no lago após 5 meses?
(c) Qual é a taxa de crescimento da população após 5 meses?
(d) Quando é que a taxa de crescimento populacional dos peixes no tanque começa a diminuir?
(e) Quando o número de peixes no tanque será de 70 por cento do capacidade da lagoa?

10. Um relatório de saúde pública afirma que semanas após a eclosão de um nova cepa de gripe, o número de pessoas, em milhares, que contrair a doença é
$$Q(t) = \frac{10}{1 + 100e^{-1.5t}}$$
(a) Quantas pessoas contraíram a doença inicialmente e quantas pessoas contraíram a doença nas próximas duas semanas?
(b) A que taxa as pessoas contraíram a doença após duas semanas?
(c) Quando a taxa de infecção começou a diminuir?
(d) Se não for tratada, quantas pessoas acabariam por contrair a gripe?

11. Em um dia em que a temperatura é $30\circ$ Celsius, uma bebida gelada é tomada de uma geladeira com temperatura de $5\circ$. Se a temperatura de a bebida está a $20\circ$ após 10 minutos, qual será sua temperatura depois 20 minutos?

As soluções estão feitas nas páginas 165 a 174.

Capítulo 13

Equações Diferenciais

Eis um exemplo: um dos modelos para o crescimento populacional informa que uma população cresce a taxa proporcional ao seu tamanho. Dando nome às variáveis t = tempo, variável independente. P = número de indivíduos da população, variável dependente. A taxa de crescimento é a derivada $\frac{dP}{dt}$. Assim, a hipótese é escrita como

$$\frac{dP}{dt} = k.P \quad ou \quad P' = k.P$$

onde k é uma constante de proporcionalidade. A equação acima é diferencial porque contém uma função desconhecida P e a sua derivada P'. A solução nos pede achar uma função cuja derivada seja uma constante multiplicada por ela própria: as funções exponenciais têm esta característica. De fato, fazendo $P(t) = Ce^{kt}$ fica

$$P'(t) = C(ke^{kt}) = k(Ce^{kt}) = kP(t)$$

Qualquer função exponencial na forma Ce^{kt}. Fazendo $t = 0$, tem-se $P(0) = C$ de modo que C acaba sendo a população inicial $P(0)$.

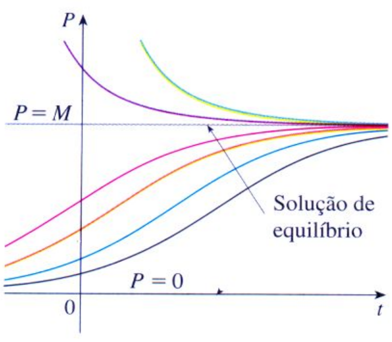
Para ser mais realista nesta análise deve-se estipular uma capacidade de suporte M que sinaliza até onde a população pode crescer (por causa de recursos limitados. Desconsiderar este fato levaria a população a ∞). Agora, a população cresce em direção a M estabilizando-se aí. Se passar de M precisa retroceder a ele. Agora o nosso modelo fica

- $\frac{dP}{dt} \approx kP$ se P pequeno
- $\frac{dP}{dt} < 0 \quad se\ P > M$ ou seja, P diminui quando excede M

Uma expressão simples que incorpora ambas hipóteses é dada pela equação diferencial logística, proposta pelo matemático e biólogo holandês Pierre-François Verhulst na década de 1840, que é

$$\frac{dP}{dt} = kP\left(1 - \frac{P}{M}\right)$$

Note que tanto $P(t) = 0$ quanto $P(t) = M$ são soluções porque o segundo termo da equação vale 0. São as chamadas soluções de equilíbrio. Tanto sendo 0 como sendo M as populações ficam com valor fcd fixo. Assim, as soluções gráficas da equação estão em



Em geral uma equação diferencial contém uma função desconhecida e uma ou mais de suas derivadas. A ordem de uma equação diferencial é a ordem da derivada mais alta que ocorre.

Uma função f é denominada solução de uma equação diferencial se a equação é satisfeita quando a função e suas derivadas são substituídas na equação.

Quando se pede para resolver uma equação diferencial espera-se encontrar todas as soluções possíveis. Em geral não é tarefa fácil. Não existe uma técnica sistemática. Pode-se ir por uma solução gráfica ou numérica se não houver solução simbólica (analítica) disponível.

A boa notícia é que qo resolver uma equação diferencial geralmente não estamos interessados na solução geral e sim em encontrar uma solução que satisfaça condições adicionais. Em muitos problemas tem-se uma condição, muitas vezes $y(t_0) = y_0$ chamada condição inicial e este caso é chamado problema do valor inicial.

13.1 Equação diferencial linear

Uma equação diferencial linear de primeira ordem é aquela que pode ser descrita como

$$y' + P(x)y = Q(x)$$

Um exemplo $xy' + y = 2x$ que pode ser reescrita ($se\ x \neq 0$) como $y' + \frac{1}{x}y = 2$.

Note que ela não é separável, porque é impossível fatorar a expressão para y' como uma função de x vezes uma função de y. Mas, se relembrar a regra do produto que diz $xy' + y = (xy)'$

na verdade a regra do produto diz $(xy)' = x'y + xy'$. Aqui o autor deu um salto acrobático

Será que é porque x é linear ? Daí $x' = 1$??????

Das duas equações, pode-se escrever que $(xy)' = 2x$ e integrando os 2 lados, fica $xy = x^2 + C$ ou $y = x + \frac{C}{x}$.

Generalizando, para resolver a equação diferencial linear $y' + P(x)y = Q(x)$ multiplique ambos os lados pelo fator integrante

$$I(x) = e^{\int P(x)dx}$$

e integre ambos os lados.

Um exemplo completo. Seja resolver $\frac{dy}{dx} + 3x^2y = 6x^2$ A equação é linear porque ela tem a forma de $y' + P(x)y = Q(x)$ com $P(x) = 3x^2$ e $Q(x) = 6x^2$. Um fator integrante é

$$I(x) = e^{\int 3x^2 dx} = e^{x^3}$$

Multiplicando ambos os lados da equação por e^{x^3} fica

$$e^{x^3}\frac{dy}{dx} + 3x^2e^{x^3}y = 6x^2e^{x^3}$$

e a resposta é $y = 2 + Ce^{-x^3}$.

Capítulo 14

Derivadas parciais

Estudemos um exemplo: Em um dia muito quente, a umidade altera a sensação de calor. Igualmente, se o dia está muito seco, tem-se uma sensação de calor menor do que a realmente marcada no termômetro. O serviço meteorológico do Canadá introduziu o *humidex* ou índice temperatura-umidade para descrever os efeitos combinados da temperatura e da umidade. O humidex I é a temperatura aparente do ar quando a temperatura real é T e a umidade é H. Desse modo, I é função de T e H e pode-se escrever $I = f(T, H)$. O serviço compilou uma tabela

Umidade relativa (%)

Temperatura real (°C)	T \ H	40	45	50	55	60	65	70	75	80
	26	28	28	29	31	31	32	33	34	35
	28	31	32	33	34	35	36	37	38	39
	30	34	35	36	37	38	40	41	42	43
	32	37	38	39	41	42	43	45	46	47
	34	41	42	43	45	47	48	49	51	52
	36	43	45	47	48	50	51	53	54	56

Se nos concentrarmos na coluna $H = 60\%$ estaremos considerando o humidex como uma variável de uma única variável T, já que o valor de H está fixado (em 60). Agora pode-se escrever $I = g(T, 60)$ ou mais propriamente $I = g(T)$. A derivada de g quando $T = 30^oC$ é a taxa de variação de I com relação a T quando $T = 30^oC$. Usando a definição de derivada já vista, obtem-se arredondando que quando a umidade é 60% e a temperatura é 30^oC a temperatura aparente (humidex) aumenta cerca de 1.75 ^oC para cada grau que a temperatura real sobe.

Se Olharmos para a linha sombreada que corresponde à T=30^oC. Os números desta linha são os valores da função $G = (H) = f(30, H)$ que descreve como o humidex aumenta à medida em que a umidade relativa H aumenta quando a temperatura real é 30^oC. A derivada desta função quando $H = 60\%$ é a taxa de variação de I com relação a H quando $T = 30^oC$. Calculando pela definição e arredondando, obte-se $G'(60) \approx 0.3$. Isto nos diz que quando a temperatura é 30^oC e a umidade é 60% o humidex aumenta em cerca de 0.3^oC para cada ponto percentual que a umidade relativa aumenta.

Em geral, se g é função de duas variáveis x e y, suponha que deixemos apenas x variar, mantendo y constante, por exemplo, fazendo $y = b$ (b é uma constante). Neste caso a função é de uma única variável, x, a saber $g(x) = f(x, b)$. Se g tem derivada em a ela será chamada derivada parcial de f em relação a x em (a, b) e denotar-se-á $f_x(a, b)$ como $f_x(a, b) = g'(a)$ e pela definição de derivada tem-se

$$g'(a) = \lim_{h \to 0} \frac{g(a+h) - g(a)}{h}$$

A equaão fica

$$f_x(a, b) = \lim_{h \to 0} \frac{f(a+h, b) - f(a, b)}{h}$$

Igualmente, a derivada parcial de f em relação a y em (a, b) é

$$f_y(a, b) = \lim_{h \to 0} \frac{f(a, b+h) - f(a, b)}{h}$$

Usam-se as seguintes notações: Se $z = f(x, y)$ então

$$f_x(x, y) = f_x = \frac{\partial f}{\partial x} = \frac{\partial}{\partial x} f(x, y) = \frac{\partial z}{\partial x} = f_1 = D_1 f = D_x f$$

ou

$$f_x(x, y) = f_y = \frac{\partial f}{\partial y} = \frac{\partial}{\partial y} f(x, y) = \frac{\partial z}{\partial y} = f_2 = D_2 f = D_y f$$

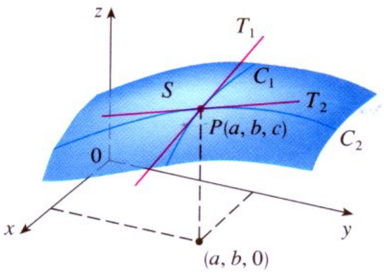
Para calcular as derivadas parciais, basta lembrar que a derivada parcial em relação a x é a derivada ordinária da função g de uma única variável obtida mantendo-se fixo o valor de y, gerando a seguinte regra:

1. Para determinar f_x trate y como constante e derive $f(x, y)$ com relação a x
2. Para determinar f_y trate x como constante e derive $f(x, y)$ com relação a y

Por exemplo: Se $f(x, y) = x^3 + x^2y^3 - 2y^2$ encontre $f_x(2, 1)$ e $f_y(2, 1)$.
Para o primeiro, mantém-se y como constante e derivando em relação a x fica: $f_x(x, y) = 3x^2 + 2xy^3$ e $f_x(2, 1) = 3.2^2 + 2.2.1^3 = 16$
Para o segundo, mantendo x constante e derivando em relação a y fica $f_y(x, y) = 3x^2y^2 - 4y$ e $f_y(2, 1) = 3.2^2.1^2 - 4.1 = 8$

14.1 Interpretação geométrica da derivada parcial

A equação $z = f(x, y)$ representa uma superfície S. Se $f(a, b) = c$, então o ponto $P = (a, b, c)$ está em S. Ao fixar $y = b$ estamos restringindo nossa atenção à curva C_1 na qual o plano vertical $y = b$ intersecciona S. Em outras palavras C_1 é o corte de S no plano $y = b$. O plano vertical $x = a$ secciona S na curva C_2. As curvas C_1 e C_2 passam pelo ponto P.



A curva C_1 é o gráfico da função $g(x) = f(x, b)$ de modo que a inclinação da tangente T_1 em P é $g'(a) = f_x(a, b)$. A curva C_2 é o gráfico da função $G(y) = f(a, y)$ de modo que a inclinação da tangente T_2 em P é $G'(b) = f_y(a, b)$.

As derivadas parciais $f_x(a, b)$ e $f_2(a, b)$ são as inclinações das retas tangentes em $P(a, b, c)$ aos cortes C_1 e C_2 de S nos planos $y = b$ e $x = a$.

14.2 Derivadas de ordem superior

Se $z = f(x, y)$ usa-se a seguinte notação:

$$(f_x)_x = f_{xx} = f_{11} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 z}{\partial x^2}$$

ou

$$(f_x)_y = f_{xy} = f_{12} = \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 z}{\partial y \partial x}...$$

Nesta segunda formulação, primeiro deriva-se em relação a x e depois a y.

14.2.1 Teorema de Clairaut

Devido ao matemático francês Alexis Clairaut (1713-1765). Suponha que f é definido em bola aberta D que contém o ponto $P(a, b)$. Se as funções f_{xy} e f_{yx} forem ambas contínuas em D então

$$f_{xy}(a, b) = f_{yx}(a, b)$$

.

Capítulo 15

Integrais Duplas

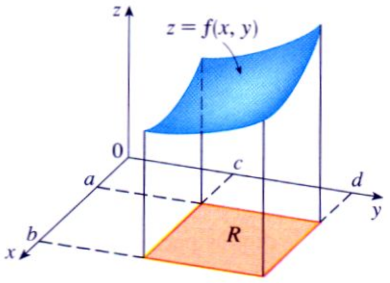
Assim como na integral simples buscava-se a área da curva entre o eixo e a função em dois pontos (a e b), aqui vai-se considerar uma função f de 2 variáveis definida em um retângulo fechado

$$R = [a, b] \times [c, d] = \{(x, y) \in \mathbb{R}^2 \mid a \leqslant x \leqslant b,\ c \leqslant y \leqslant d\}$$

Supondo que $f(x, y) \geqslant 0$. O gráfico de f é a superfície com equação $z = f(x, y)$. Seja S o sólido que está acima da região R e abaixo do gráfico de f, isto é

$$S = \{(x, y, z) \in \mathbb{R}^3 \mid 0 \leqslant z \leqslant f(x, y),\ (x, y) \in R\}$$

como demonstrado em



Como no caso de 1 dimensão, criam-se retângulos em R e a somatória de seus volumens leva ao volume do sólido em questão. Daí A integral dupla de f sobre o retângulo R é

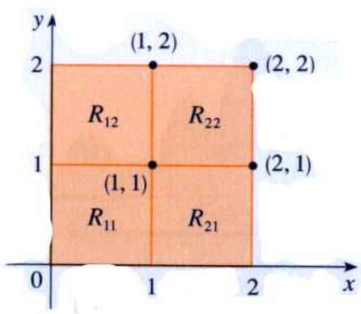
$$\iint_R f(x, y)\ dA = \lim_{m,n\to\infty} f(x_{ij}^*, y_{i,j}^*)\Delta A$$

se o limite existir Daqui, se $f(x, y) \geqslant 0$ então o volume V do sólido acima do retângulo R e abaixo da superfície $z = f(x, y)$ é

$$V = \iint_R f(x, y)\ dA$$

Seja um exemplo gráfico. Suponha o sólido que está acima do quadrado $T = [0, 2] \times [0, 2]$ e abaixo do parabolóide elíptico $z = 16 - x^2 - 2y^2$. Divida R em quatro quadrados iguais e escolha o ponto de amostragem como o canto superior direito de cada quadrado R_{ij}.

Os quadrados estão ilustrados aqui



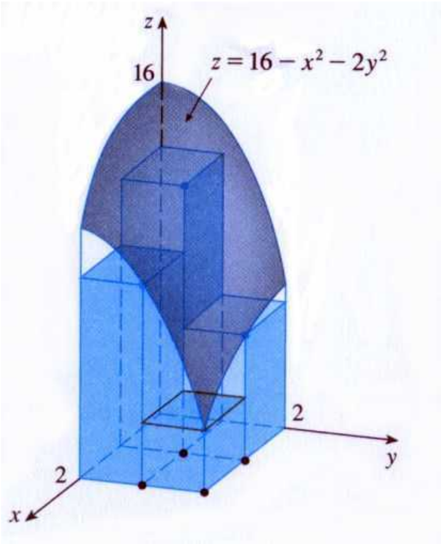
O parabolóide elíptico é o gráfico de $f(x,y) = 16 - x^2 - 2y^2$ e a área de cada quadrado é $\Delta A = 1$. Aproximando o volume com $m = n = 2$ temos

$$V \approx \sum_{i=1}^{2} \sum_{j=1}^{2} f(x_i, y_j) \Delta A$$

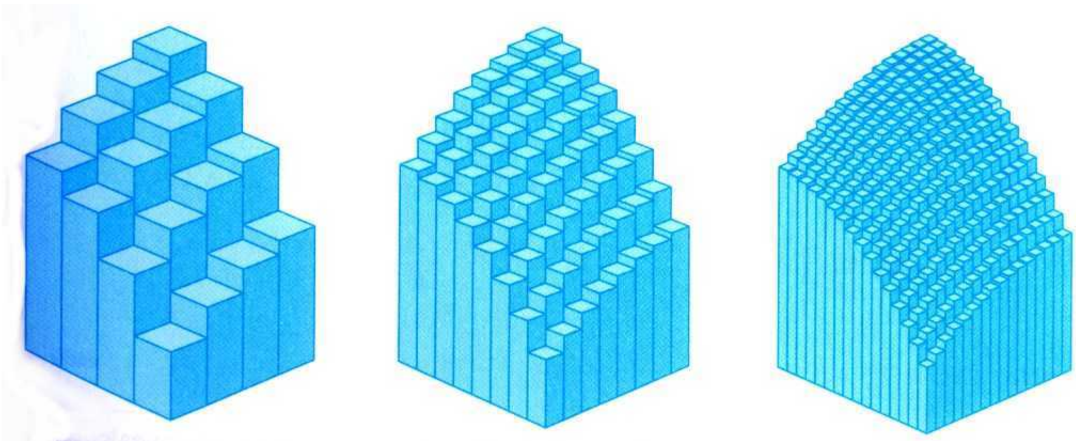
$$= f(1,1)\Delta A + f(1,2)\Delta A + f(2,1)\Delta A + f(2,2\Delta A$$

$$= 13(1) + 7(1) + 10(1) + 4(1) = 34$$

como se pode ver em



Obtém-se melhores aproximações quando se aumenta o número de quadrados. Quando se usa 16, 64 e 256 quadrados. O valor correto deste volume é 48.


(a) $m = n = 4$, $V \approx 41{,}5$ (b) $m = n = 8$, $V \approx 44{,}875$ (c) $m = n = 16$, $V \approx 46{,}46875$

15.1 Coordenadas esféricas

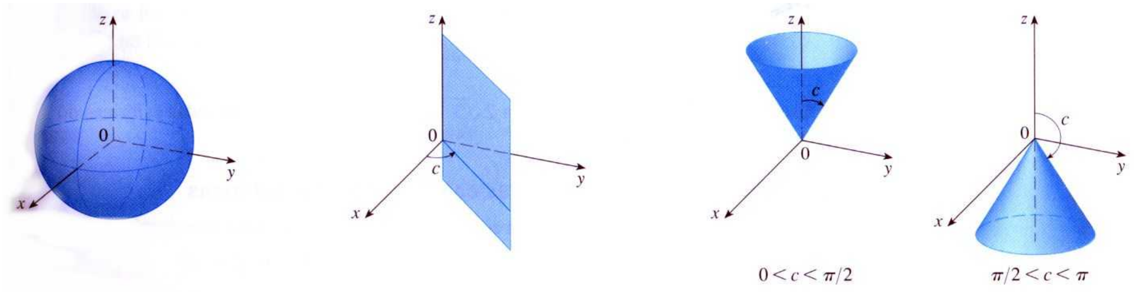
Um ponto P pode ser caracterizado no espaço pelas suas coordenadas esféricas (ρ, θ, ϕ), onde

ρ Distância do ponto à origem do sistema

θ Ângulo de um plano vertical que passa pela origem e contém P com o eixo x

ϕ Ângulo no plano acima entre a horizontal e a reta que passa pela origem e por P

O nome coordenadas esféricas vem do fato de que nele, uma esfera de raio c tem como função $\rho = c$. Para $\theta = c$ temos o semi-plano vertical que passa pela origem com ângulo c em relação a x. Para $\phi = c$ temos um semi-cone com o eixo z como seu eixo. Veja nas figuras



A conversão entre coordenadas cartesianas em 3 dimensões e coordenadas esféricas segue o formulário:
$x = \rho \ sen\phi \ cos\theta$
$y = \rho \ sen\phi \ sen\theta$
$z = \rho \ cos\phi$
$\rho^2 = x^2 + y^2 + z^2$

Capítulo 16

Cálculo vetorial

Seja D um conjunto em $\mathbb{R}^2$ (uma região plana). Um campo vetorial em $\mathbb{R}^2$ é uma função F que associa a cada ponto (x, y) em D um vetor bidimensional $F(x, y)$.

Seja E um sub-conjunto de $\mathbb{R}^3$. Um campo vetorial em $\mathbb{R}^3$ é uma função F que associa a cada ponto (x, y, z) em E um vetor tridimensional $F(x, y, z)$.

16.1 Gradiente

O $gradF$ ou ∇f que se lê ”del-f”(para 2 dimensões) é

$$\nabla f(x, y) =$$

16.2 Rotacional

Se $F = P\mathbf{i} + Q\mathbf{j} + R\mathbf{k}$ é um campo vetorial em $\mathbb{R}^3$ e as derivadas parciais de P, Q e R existem então o **rotacional** de F é um campo vetorial em $\mathbb{R}^3$ definido por

$$rotF = \left(\frac{\partial R}{\partial y} - \frac{\partial Q}{\partial z}\right)\mathbf{i} + \left(\frac{\partial P}{\partial z} - \frac{\partial R}{\partial x}\right)\mathbf{j} + \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right)\mathbf{k}$$

Para auxiliar na memorização, introduz-se o operador diferencial vetorial ∇ como

$$\langle f_x(x, y), f_y(x, y)\rangle = \frac{\partial f}{\partial x}\mathbf{i} + \frac{\partial f}{\partial y}\mathbf{j}$$

Se considerarmos uma função f de 3 variáveis e um ponto $P_0(x + 0, y + 0, z_0)$ de seu domínio, o gradiente $\nabla f(x_0, y_0, z_0)$ dá a direção do aumento mais rápido de f em P.

$$\nabla = \mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z}$$

que quando opera sobre uma função escalar, produz o gradiente de f

$$\nabla f = \mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z}$$

Se pensarmos em ∇ como um vetor de componentes $\partial/\partial x$, $\partial/\partial y$ e $\partial/\partial z$, pode-se considerar o produto vetorial formal de ∇ pelo campo vetorial F como segue

$$\nabla \times F = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ P & Q & R \end{vmatrix} = \left(\frac{\partial R}{\partial y} - \frac{\partial Q}{\partial z}\right)\mathbf{i} + \left(\frac{\partial P}{\partial z} - \frac{\partial R}{\partial x}\right)\mathbf{j} + \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right)\mathbf{k} = rot\ F$$

ou ainda

$$rot\ F = \nabla \times F$$

Capítulo 17

Bibliografia

Don01 Don, Eugene e Don Benay. **How to solve word problems in Calculus**. McGraw Hill, New York, 2001.

Ste14a Steward, James. **Cálculo I**.

Ste14b Steward, James. **Cálculo II**.

New02 Newton, Isaac. **Principia Mathematica**.

Kre83 Kreyszig, Erwin. **Matemática Superior**. Vol 1.

Kan18 Kantek, Pedro. **Python**. UFPR, Curitiba, 2018.

Kan82 Kantek, Pedro. **Uma introdução ao APL**. Celepar, Curitiba, 1982.

Bar60 Baranenkov, G. et alli. **Problemas y ejercicios de analisis matematico**. Mir, Moscou, 1960.

Índice Remissivo

adição, 30
alo mundo, 26
apl, 28, 43
aspas, 28
atribuição, 28

bactéria, 104, 168
bibliografia, 185
break, 35
Báskara, 8

C++, 28
carbono 14, 170
circunferência, 18
concavidade, 120
condicional, 33
continue, 36
continuidade, 94
continuidade de polinômios, 94
convergência, 125
coordenada esférica, 180
custo marginal, 104
cálculo, 87

decaimento, 167
depreciação, 105
derivada de ordem superior, 99
derivada parcial, 177
derivadas relacionadas, 111
derivação implícita, 109
descartes, 17
distância entre 2 pontos, 17
divisão inteira, 30
divisão real, 30
dois pontos, 34
dyalog, 43

else, 34
equação diferencial, 175
equação diferencial linear, 176
expressão aritmética, 30

fermat, 17
float, 29, 31
for, 36
freeware, 24
função exponencial, 167
funções trigonométricas, 108
fólio de descartes, 109

gedit, 27
giroflex, 161
gradiente, 183

if, 33
impares, 34
inclinação, 18
indentação, 33
init_session, sympy, 37
input, 31
int, 31
integral definida, 128
integral indefinida, 128
integração aproximada, 130
Ipython, 27

java, 28
juro composto, 170

l'hospital, 121
lei de boyle, 105
limite, 91
limites laterais, 93
logaritmo natural, 140
loop, 34

mainstream, 24
mais, 30
maiúsculas, 30
maple, 19
mathway, 85
meia vida, 169
menos, 30
minúsculas, 30
multiplataforma, 24
multiplicação, 30
máximo, 119
método da casca, 137
método do disco, 135
mínimo, 119
módulo, 30

newton-raphson, 123
nomes, 30
numpy, 29
número crítico, 120
número da licença aplx, 43

otimização, 119

painel de controle, 25
palavra reservada, 29

paralelismo, 18
parênteses, 30
pascal, 28
pasto cercado, 10
perpendicularidade, 18
PHP, 28
Pitágoras, 7
plano argand-gauss, 17
poluição, 104
ponto médio, 128
precedência, maple, 19
problema da tangente, 87
problema da área, 87
programação, maple, 22
python, 24
python2, 31
python3, 31
pythonanywhere, 27

regra da cadeia, 111
regra da substituição, 129
regra de simpson, 131
regra do produto, 108
regra do quociente, 108
regra do trapézio, 130
resto, 30
rotacional, 183

secante, 88
simpificação, 38
somatória, 127
str, 31
sublinha, 29
subtração, 30
symbols, sympy, 37
sympy, 36
system, 25

tangente, 88
tartaruga, 91
tautologia, 35
taxas de variação, 96
tela azul da morte, 27
teorema de clairaut, 178
teorema de fermat, 120
teorema fundamental do cálculo, 128
tiobe, 24
tipo, 29

unicode, 30
utf8, 30

variável, 28
velocidade, 88
versão liberada, aplx, 43
vezes, 30
volume de sólido de revolução, 135

web, 27
while, 34
winpython, 27

zenão, 91