matemática computacional ii

com o Mathematica

Aulas Laboratoriais

2020

Universidade do Minho
Escola de Ciências

Departamento de Matemática

maria irene falcão :: fernando miranda

Introdução

Esta coletânea de exercícios tem como principal objetivo servir de apoio às aulas laboratoriais da unidade curricular (UC) Matemática Computacional II do 1º ano da Licenciatura em Matemática da Universidade do Minho. Nos primeiros capítulos são apresentados exercícios agrupados por temas, de acordo com o programa e objetivos de aprendizagem da UC, seguindo-se um capítulo com exercícios suplementares selecionados de provas de avaliação. Para os exercícios assinalados com o símbolo ⊛, é apresentada uma sugestão de resolução no capítulo final dos textos.

Na exploração de exemplos e resolução de problemas, é de esperar que o aluno desenvolva as suas faculdades de assimilar informação e de a comunicar, de selecionar ferramentas matemáticas e computacionais aplicadas a várias áreas da matemática, explorando as suas capacidades de realizar cálculo simbólico ou numérico e as potencialidades gráficas.

A exploração do sistema computacional Mathematica em ambiente laboratorial permitirá ao aluno desenvolver a capacidade de:

- descrever as regras básicas do sistema, pesquisar e utilizar funcionalidades recorrendo aos sistemas de ajuda;
- construir funções/programas, usando programação funcional e recursiva, que desempenhem tarefas especificadas e utilizar regras de reescrita como elemento de atribuição e de programação;
- resolver problemas de Análise Real, Álgebra Linear e Teoria de Números com recurso a funcionalidades predefinidas.

Estando esta UC no primeiro ano do plano de estudos, não é ainda possível tirar partido em pleno das capacidades da ferramenta, mas as competências adquiridas serão certamente úteis ao longo de todo a formação de um estudante de Matemática.

Conteúdo

1. Números e Expressões

⇛ Exercícios

Exercício 1.1 Use a função `FullForm` e `TreeForm` para obter a representação interna de cada uma das seguintes expressões e a função `Head` para obter o operador principal das expressão compostas. Indique ainda quais das expressões são atómicas.

a) 2 *b)* $\frac{1}{2}$ *c)* $\sqrt{2}$ *d)* $2+x$ *e)* π *f)* $\frac{2+3x^2}{4}$

Exercício 1.2 Avalie as seguintes expressões e comente os valores obtidos, recorrendo à função `Precision`.

a) 2^{1000} 2.0^{1000} $2.000000000000000000^{1000}$

b) $\sqrt{2}$ $\sqrt{2.000}$ $\sqrt{2.0000000000000000000000000}$

Exercício 1.3 Justifique as respostas dadas pelo Mathematica quando avalia as seguintes expressões:

a) $\frac{1}{3}+\frac{22}{33}$ *b)* $1+\pi$ *c)* $1.5+\frac{\pi}{2}$ *d)* $\frac{1}{2}+2e$ *e)* $\sqrt{4}+\pi-2.0$ *f)* $\operatorname{sen}\frac{\pi}{4}$ *g)* $\operatorname{sen}\frac{\pi}{8}$

h) $\mathrm{e}^{20^{12}}$ *i)* $\mathrm{e}^{20.^{12}}$ *j)* $\mathrm{e}^{-20.^{12}}$ *k)* $\operatorname{tg}\frac{\pi}{2}$ *l)* $\operatorname{tg}\frac{\pi}{2.0}$ *m)* $\operatorname{arctg}(+\infty)$

Exercício 1.4 Determine aproximações com 3 e 100 casas decimais de precisão e ainda com precisão de máquina para:

a) $\frac{15}{31}$ *b)* e^2 *c)* $3\sqrt{7}$ *d)* $\cos 1$ *e)* $\sqrt[5]{71}$ *f)* π *g)* π^3

Exercício 1.5 Determine aproximações racionais para e, π e $\sqrt{11}$ (escolha $dx = 0.01$).

Output esperado: $\frac{19}{7}$, $\frac{22}{7}$, $\frac{43}{13}$

Exercício 1.6⊛ Obtenha a parte decimal de π e de $\sqrt{2}$ com 50 dígitos, usando a função:

a) `FractionalPart` *b)* `Round` *c)* `Floor`

Exercício 1.7 Execute as seguintes instruções e comente os resultados obtidos.

```
Plot[{x,FractionalPart[x]},{x,-3,3}]
Plot[{x,IntegerPart[x]},{x,-3,3}]
Plot[{x,Round[x]},{x,-3,3}]
Plot[{x,Ceiling[x]},{x,-3,3}]
Plot[{x,Floor[x]},{x,-3,3}]
```

Exercício 1.8⊛ Construa uma tabela das primeiras dez potências de π, arredondadas para inteiros.

Output esperado: $\{3, 10, 31, 97, 306, 961, 3020, 9489, 29809, 93648\}$

Exercício 1.9⊛ Construa uma tabela de valores de $(1 + \frac{1}{x})^x$, usando precisão de máquina, e considerando para valores de x as seis primeiras potências de 10.

Output esperado: $\{2.59374, 2.70481, 2.71692, 2.71815, 2.71827, 2.71828\}$

⇛ Alguns comandos

`$MachineEpsilon` `$MachinePrecision` `$MaxMachineNumber` `$MaxNumber` `$MinMachineNumber` `$MinNumber` `Accuracy` `AtomQ` `EvenQ` `Floor` `FractionalPart` `FullForm` `Head` `InputForm` `IntegerPart` `IntegerQ` `MachineNumberQ` `MachinePrecision` `N` `NumberQ` `NumericQ` `OddQ` `Precision` `PrimeQ` `Rationalize` `Round` `TreeForm`

⇛ Links

http://reference.wolfram.com/language/tutorial/NumbersOverview
http://reference.wolfram.com/language/tutorial/ExpressionsOverview

2. Atribuições e Funções

⇛ Exercícios

Exercício 2.1 Atribua o valor $\frac{\pi}{8}$ à variável x.

a) Calcule $\operatorname{sen}^2 x + \cos^2 x$ e simplifique o resultado obtido.

b) Verifique que $\operatorname{tg} x = -1 + \sqrt{2}$.

Exercício 2.2 Chama-se **número de ouro** ao número irracional $\frac{1+\sqrt{5}}{2}$. Atribua à variável `ouro` o valor do número de ouro.

a) Calcule $\displaystyle\sum_{k=0}^{5} \mathtt{ouro}^k$.

b) Calcule $\displaystyle\prod_{k=0}^{5} (\mathtt{ouro} - k)$.

c) Use a ajuda do Mathematica para obter informação sobre a constante `GoldenRatio` e repita as alineas anteriores, usando agora esta constante.

Output esperado: $14 + 6\sqrt{5}$; $125 - 54\sqrt{5}$

Exercício 2.3 Execute as instruções seguintes e faça um comentário aos resultados.

```
a = RandomInteger[20];
b := RandomInteger[20];
Table[a,10]
Table[b,10]
```

Exercício 2.4 Atribua o valor $1 + y$ à variável x. Calcule o valor de $\sum_{k=0}^{12} kx$, quando $y = 0$ e quando $y = 2$.

Output esperado: 78; 234

Exercício 2.5⊛ Considere a expressão $x^2y - 4e^xy^3$. Calcule, de duas formas diferentes, o valor da expressão para $x = y = 0$ e $x = -y = -1$.

Exercício 2.6⊛ Construa uma tabela de valores da função $\operatorname{sen} x$, para valores do argumento iguais a 0, $\frac{\pi}{6}$, $\frac{\pi}{4}$, $\frac{\pi}{3}$ e $\frac{\pi}{2}$. Use dois processos diferentes.

Exercício 2.7 Execute as instruções seguintes e faça um comentário aos resultados.

```
Table[a,10] /. a -> RandomInteger[6]
Table[a,10] /. a :> RandomInteger[6]
```

Exercício 2.8 Defina a função $f(x) = x^4 + 3x^3 - 3x^2 + 10$.

a) Calcule $f(\frac{-5-\sqrt{5}}{2})$, $f(\frac{5-\sqrt{5}}{2})$ e $f(\frac{-5+\sqrt{5}}{2})$.

b) Designe por a o valor de $f(\sqrt{2} - i)$ e obtenha a parte real e a parte imaginária de a, a^2 e a^3.

c) Represente graficamente as funções $f(x), f(-x), -f(x)$, no intervalo $[-4, 2]$.

Output esperado: *a)* 0; $150 - 60\sqrt{5}$;d 0
b) $a = -3\sqrt{2} + i(2\sqrt{2} - 15)$; $a^2 = -215 + 60\sqrt{2} + i(90\sqrt{2} - 24)$;
$a^3 = -1080 + 2043\sqrt{2} + i(2925 - 1258\sqrt{2})$

Exercício 2.9 Defina a função

$$f(x) = \begin{cases} -x+3, & x < 0 \\ 2, & x = 0 \\ x+1, & x > 0 \end{cases}$$

de duas formas diferentes e obtenha, para cada caso, o gráfico de f, no intervalo $[-2, 2]$.

Exercício 2.10 Represente graficamente a seguinte função, no intervalo $[-2, 3]$.

$$f(x) = \begin{cases} -x^3, & x < -1 \\ x, & -1 \leq x \leq 2 \\ x^2 + 2x - 1, & x \geq 2 \end{cases}$$

Exercício 2.11⊛ Defina as funções $g(x) = \begin{cases} x-1, & x < 2 \\ 5, & x = 2 \\ x+1, & x > 2 \end{cases}$ e $f(x) = \begin{cases} g(x), & x \leq 5 \\ g(x)^2, & x > 5 \end{cases}$

⇛ Alguns comandos

`Condition` `FullSimplify` `Piecewise` `Product` `RandomInteger` `Replace` `ReplaceAll` `PiecewiseExpand` `ReplaceRepeated` `Rule` `RuleDelayed` `Simplify` `Sum` `Table`

⇛ Links

http://reference.wolfram.com/language/tutorial/MakingDefinitionsForFunctions
http://reference.wolfram.com/language/tutorial/ImmediateAndDelayedDefinitions
http://reference.wolfram.com/language/tutorial/ApplyingTransformationRules

3. Listas

⇛ Exercícios

Exercício 3.1 Construa a lista $\{\{4, 2, -3\}, 5, \{0, 5\}, 8, \{0, 1, 5, 3, 55\}\}$.

a) Obtenha o elemento na posição $(3, 1)$ da lista.

b) Obtenha o seu último elemento.

c) Indique onde se encontram situados os elementos 5. Confirme a sua resposta utilizando o comando `Position`.

Exercício 3.2 Considere os comandos `Range` e `Table`.

a) Avalie o seguinte código e justifique as respostas obtidas:

```
Range[5]
Range[6.93]
Range[2.1,14]
Range[2.1,14,3.2]
```

b) Avalie o seguinte código e justifique as respostas obtidas:

```
Table[i+33,7]
Table[i+33,{7}]
Table[i+33,{i,7}]
Table[(i-149)/30,{i,-5,38,6}]
Table[{i,Sin[i/23]},{i,12}]
```

c) Use estes comandos para criar, de duas maneiras distintas, se possível, a lista dos quadrados dos primeiros 25 números naturais e a lista dos primeiros 30 números primos.

d) Use os comandos `Take` e `Drop` para obter os 4 primeiros e os 4 últimos elementos das listas obtidas na alínea anterior, se possível de duas maneiras distintas.

e) Use o comando `Partition` para criar uma lista de pares de elementos consecutivos a partir de uma qualquer das listas anteriores.

Exercício 3.3 Aplique os comandos `Sort`, `Reverse`, `Union`, `RotateLeft` e `RotateRight` à lista gerada pelo seguinte código:

```
RandomInteger[{1, 15}, 50]
```

Exercício 3.4 Construa a lista $\{\{2,1,10\},3,9,\{\{9\},5,7\},\{\{2,10,4\},\{10,1,9\}\},-2,\{0,-7\},\{6,1,\{6,9\}\}\}$.

a) Qual é o primeiro elemento da lista?

b) Onde se encontram situados os noves da lista apresentada (tente antes encontrar a resposta)?

c) Aplique o comando `Flatten` à lista dada e interprete o resultado obtido. Designe a lista obtida por `lista`.

d) Retire de `lista` os números não positivos.

e) Substitua em `lista` os números nas posições pares por 0.

Exercício 3.5⊛ Construa uma lista com 50 números inteiros aleatórios entre - 10 e 10 e designe-a por `lista1`.

a) Quantas vezes aparece o número 0 em `lista1`?

b) Defina uma nova lista, `lista2`, eliminando da lista anterior as repetições.

c) Construa uma lista que contenha os números pares de `lista2`.

d) Obtenha uma lista com os números que se encontram nas posição pares de `lista2`.

e) Acrescente à `lista1` os números -12 no início e 12 no final.

Exercício 3.6⊛ Considere a lista com os restos da divisão inteira por 5 dos primeiros 20 números primos.

a) Construa a lista.

Output esperado: $\{2,3,0,2,1,3,2,4,3,4,1,2,1,3,2,3,4,1,2,1\}$

b) Ordene a lista por ordem decrescente.

c) Quantas vezes aparecem cada um dos dígitos de 0 a 4 nessa lista?

Exercício 3.7⊛ Execute o seguinte código

```
listaprimos=Prime[Range[PrimePi[100]]]
```

a) Qual o tipo de números que pertencem a `listaprimos`?

b) Construa a lista formada pelos dígitos que constituem cada elemento de `listaprimos`.

c) Represente graficamente o número de ocorrências de cada dígito que aparece na lista construída em b).

Exercício 3.8⊛ Use a função `RandomPrime` para obter uma lista com 30 primos aleatórios com dois algarismos.

a) Quantos primos distintos existem na lista?

b) Calcule a média dos primos presentes na lista.

c) Indique qual é o primo que aparece mais vezes na lista.

d) Identifique o(s) primo(s) de dois algarismos que não aparece(m) na lista.

Exercício 3.9 Considere a lista `{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z}`

a) Use a função `Alphabet` para construir esta lista.

b) Construa a lista do alfabeto português, usando agora a função `CharacterRange`.

c) Construa uma lista com as consoantes do alfabeto em maiúsculas.

d) Construa as listas com os elementos nas posições pares e ímpares da lista dada.

Exercício 3.10 Considere a lista formada pelos primeiros 10 múltiplos de 2 e 10 múltiplos de 3, ordenados por ordem crescente.

a) Construa a lista e designe-a por `lista1`.
Output esperado: $\{2, 3, 4, 6, 6, 8, 9, 10, 12, 12, 14, 15, 16, 18, 18, 20, 21, 24, 27, 30\}$

b) Construa a lista `lista2` que resulta da substituição do terceiro elemento da `lista1`, por 18.

c) Construa a lista que se obtém quando se substituem todas as ocorrências do elemento 18 na `lista2` por 0.

d) Quantos múltiplos comuns de 2 e 3 estão presentes em `lista1`?

e) Defina a `lista3` formada pelos elementos da `lista2` agrupados cinco a cinco. Apresente a lista em forma de tabela.

f) Acrescente a `lista3`, os primeiros 5 múltiplos de 4.

Exercício 3.11 Considere as listas seguintes:

$\texttt{listaA} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$; $\texttt{listaB} = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29\}$

$\texttt{listaC} = \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}\}$; $\texttt{listaD} = \{\{2, 3\}, \{5, 7\}, \{11, 13\}, \{17, 19\}, \{23, 29\}\}$.

Construa estas listas de forma eficiente, efetue as operações indicadas e interprete os resultados obtidos:

a) 1+listaA;

b) 5+listaC;

c) 2*listaB;

d) π listaD;

e) listaA /2;

f) 2/listaA;

g) listaC /5;

h) listaA + listaB;

i) listaA + listaC;

j) listaC + listaD;

k) listaA * listaB;

l) listaA listaC;

m) listaC listaD;

n) listaA /listaB;

o) listaA /listaC;

p) listaC /listaD;

q) 2^listaA;

r) 2^listaC;

s) listaA^3;

t) listaA^listaB;

u) listaC^listaD.

Exercício 3.12 Use a função `Attributes` para obter os atributos das funções `Plus`, `Times` e `Sin` e verificar que estas funções são listáveis. Ilustre esta característica através de um exemplo.

Exercício 3.13 Execute e comente o seguinte código:

a)

```
Remove[f]
Attributes[f]
A = f[{{a,b},c}]
f[A]
```

b)

```
SetAttributes[f,Listable]
A
f[A]
```

c)

```
Remove[g]
Attributes[g]
B = g[g[a],g[b,c]]
```

d)

```
SetAttributes[g,Flat]
B
g[B]
```

Exercício 3.14 Construa a lista dos valores de sen(x_k), para $x_k \in [0,1]$, $x_0 = 0$ e $|x_k - x_{k+1}| = 0.01$.

Exercício 3.15⊛ Defina uma função `somadigitos` para, dado um número natural n, obter a soma dos dígitos de n. Teste a sua função para vários valores de n escolhidos aleatoriamente no intervalo [20,200].

Output esperado: `somadigitos`[123]→6

Exercício 3.16⊛ Defina uma função `somadivisores` para, dado um número natural n, obter a soma dos seus divisores próprios. Ilustre a sua função através de exemplos.

Output esperado: `somadivisores`[150]→222

Exercício 3.17⊛ Defina uma função `divisorescomuns` para, dados dois números naturais a e b, obter a lista dos divisores comuns. Ilustre a sua função através de exemplos.

Output esperado: `divisorescomuns`[24,32]→{1,2,4,8}

Modifique a função para obter o máximo divisor comum de a e b.

⇛ Alguns comandos

`Append` `AppendTo` `Complement` `Complement` `Delete` `Dimensions` `Drop` `Drop` `Drop` `First` `Flatten` `Insert` `IntegerDigits` `Intersection` `Join` `Last` `Length` `Part` `Partition` `Position` `Prepend` `PrependTo` `Range` `RealDigits` `ReplacePart` `Rest` `Reverse` `RotateLeft` `RotateLeft` `RotateRight` `RotateRight` `Select` `Split` `Table` `TableForm` `Take` `Tally` `Tuple` `Union`

⇛ Links

http://reference.wolfram.com/language/guide/ElementsOfLists
http://reference.wolfram.com/language/guide/ListManipulation
http://reference.wolfram.com/language/tutorial/ListsOverview

4. Matrizes

Álgebra Linear com o Mathematica

⇛ Exercícios

Exercício 4.1 Considere as matrizes

$$A = \begin{pmatrix} 10 & -6 & -9 \\ 6 & -5 & -7 \\ -10 & 9 & 12 \end{pmatrix} \quad \text{e} \quad B = \begin{pmatrix} 3 & -4 & 5 \\ 8 & 0 & -3 \\ 5 & 2 & 1 \end{pmatrix}.$$

a) Obtenha:

(i) o elemento $A_{2\,1}$;

(ii) a terceira linha de A;

(iii) a matriz transposta de A, sob a forma de matriz;

(iv) a matriz A^4.

b) Determine:

(i) $A + B$; (ii) $B - 4\,A$; (iii) $(AB)^{-1}$;

(iv) $B^{-1}A^{-1}$ (v) $((A - 2B)\,B)^{\mathsf{T}}$; (vi) $\det(B)$.

Exercício 4.2$^{\circledast}$ Defina a matriz

$$A = \begin{pmatrix} 1 & -1 & 2 \\ 3 & 2 & 1 \\ 1 & -5 & 4 \end{pmatrix}$$

e obtenha, a partir de A, as seguintes matrizes.

$$A_1 = \begin{pmatrix} 1 & -1 & 2 \\ 3 & 2 & 1 \\ 1 & 5 & 4 \end{pmatrix} \quad A_2 = \begin{pmatrix} 1 & -1 & 2 \\ 0 & 2 & 1 \\ 0 & 0 & 4 \end{pmatrix} \quad A_3 = \begin{pmatrix} 1 & -1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & -5 & 4 & 3 \end{pmatrix}$$

$$A_4 = \begin{pmatrix} 1 & -1 & 2 \\ 3 & 2 & 1 \\ 1 & -5 & 4 \\ 1 & 2 & 3 \end{pmatrix} \quad A_5 = \begin{pmatrix} 2 & 1 & -1 & 2 \\ 0 & 1 & -1 & 2 \\ 0 & 3 & 2 & 1 \\ 0 & 1 & -5 & 4 \end{pmatrix} \quad A_6 = \begin{pmatrix} 1 & -1 \\ 3 & 2 \end{pmatrix}$$

Exercício 4.3 Defina (de uma forma simples) uma matriz:

a) de ordem 5 com todos os elementos iguais a 3;

b) diagonal, de ordem 5, com todos os elementos da diagonal iguais a 8;

c) de ordem 4 em que cada coluna seja o vetor $v = (1, 2, 3, 4)^{\mathsf{T}}$;

d) com a seguinte estrutura

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 2 & 0 \\ 1 & 1 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Exercício 4.4 Execute o comando

```
(A=ToeplitzMatrix[4])//MatrixForm
```

e indique uma forma de:

a) construir a matriz cujas colunas são as colunas *pares* da matriz `A`;

b) construir a matriz cujas linhas são as linhas *ímpares* da matriz `A`;

c) converter `A` numa matriz 2×8;

d) calcular o inverso de cada elemento da matriz `A`;

e) calcular a inversa da matriz `A`;

f) calcular o quadrado de cada elemento da matriz `A`.

Exercício 4.5 Considere o comando `FlattenAt`. Utilize-o para transformar a lista

$$\{\{a, 3, \{6, k\}\}, \{x, 5, -8, 1\}, \{\{0\}, \{a, h, j\}\}\}$$

numa matriz 3×3. Utilize o comando `MatrixQ` para testar se o seu resultado é realmente uma matriz.

Exercício 4.6 Determine a família das matrizes permutáveis com A, quando:

a)[⊛] $A = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$;

b) $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 1 & 2 \end{pmatrix}$.

Exercício 4.7 Identifique quais das seguintes matrizes são simétricas, antissimétricas, hermíticas ou anti-hermíticas.

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 2 & 1+i \\ 1-i & 3 \end{pmatrix} \quad C = \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix} \quad D = \begin{pmatrix} i & 1 \\ -1 & -i \end{pmatrix}$$

$$E = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \quad F = \begin{pmatrix} 2 & 1 & 0 \\ -1 & 1 & 2 \\ 0 & -2 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 0 & 1 & i \\ 1 & 0 & -2 \\ -i & -2 & 1 \end{pmatrix}.$$

Exercício 4.8 Diga quais dos seguintes conjuntos de vetores de $\mathbb{R}^3$ são formados por vetores linearmente independentes e, em caso de dependência, escreva um deles como combinação linear dos restantes:

$a)$ $\{(1,-2,3),(3,-6,9)\}$; $b)$ $\{(1,-2,-3),(3,2,1)\}$;

$c)$ $\{(0,1,-2),(1,-1,1),(1,2,1)\}$; $d)$ $\{(0,2,-4),(1,-2,-1),(1,-4,3)\}$.

Output esperado: Não; Sim; Sim; Não

Exercício 4.9 Verifique se as seguintes matrizes são equivalentes por linhas:

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \quad \text{e} \quad B = \begin{pmatrix} 2 & 2 & 1 \\ -2 & -2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

Exercício 4.10 Considere o sistema de equações lineares $A\mathbf{x} = \mathbf{b}$, onde

$$A = \begin{pmatrix} 1 & -1 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad \text{e} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix}.$$

Sem resolver o sistema, mostre que $(1,1,1,0)$ é solução do sistema, mas $(1,-1,1,1)$ não é.

Exercício 4.11 Resolva, quando possíveis, os seguintes sistemas.

a) $\begin{cases} 2x_1 - 3x_2 = 5 \\ -4x_1 + 6x_2 = 8 \\ 3x_1 - 2x_2 = 0 \end{cases}$ b) $\begin{cases} x_1 + x_2 = 0 \\ 2x_1 + 3x_2 = 0 \\ 3x_1 - 2x_2 = 0 \end{cases}$

c) $\begin{cases} 2x_1 + 3x_2 + x_3 = 1 \\ x_1 + x_2 + x_3 = 3 \\ 3x_1 + 4x_2 + 2x_3 = 4 \end{cases}$ d) $\begin{cases} -x_1 + 2x_2 - x_3 = 2 \\ -2x_1 + 2x_2 + x_3 = 4 \\ 3x_1 + 2x_2 + 2x_3 = 5 \\ -3x_1 + 8x_2 + 5x_3 = 17 \end{cases}$

Output esperado: Impossível; $(0,0)$; $\{(8-2\alpha, -5+\alpha, \alpha),\ \alpha \in \mathbb{R}\}$; $(0, \frac{3}{2}, 1)$

Exercício 4.12 Calcule o valor dos seguintes determinantes.

a) $\begin{vmatrix} 1 & 2 & 1 \\ 2 & 1 & 3 \\ 2 & 1 & 0 \end{vmatrix}$ b) $\begin{vmatrix} 2 & 1 & 3 \\ 1 & 2 & 1 \\ 2 & 1 & 0 \end{vmatrix}$ c) $\begin{vmatrix} -1 & -2 & -1 \\ -2 & -1 & -3 \\ -2 & -1 & 0 \end{vmatrix}$

Exercício 4.13 Deduza as seguintes expressões.

a) $\begin{vmatrix} 1 & x_1 & x_2 \\ 1 & y_1 & x_2 \\ 1 & y_1 & y_2 \end{vmatrix} = (y_1 - x_1)(y_2 - x_2)$ b) $\begin{vmatrix} 1 & x_1 & x_2 & x_3 \\ 1 & y_1 & x_2 & x_3 \\ 1 & y_1 & y_2 & x_3 \\ 1 & y_1 & y_2 & y_3 \end{vmatrix} = (y_1 - x_1)(y_2 - x_2)(y_3 - x_3)$

Exercício 4.14 Designa-se por **matriz de Redheffer** uma matriz quadrada cujas entradas $a_{i,j}$ são 1, se i divide j ou se $j = 1$ e são 0, nos outros casos.

a) Obtenha a matriz de Redheffer de ordem 12 e designe-a por `R12`.

b) Verifique que `R12` é uma matriz invertível.

c) Obtenha a soma dos elementos de cada linha da matriz $\texttt{R12}^{-1}$.

d) Resolva o sistema $\texttt{R12}x = b$, onde b é a quarta coluna da matriz identidade de ordem 12.

Exercício 4.15⊛ Chama-se **matriz de Vandermonde** a uma matriz $V = (v_{i,j})$ cujos elementos de cada coluna estão em progressão geométrica, i.e. $v_{i,j} = a_i^{j-1}$. Por exemplo, a seguinte matriz é uma matriz de Vandermonde de ordem 4:

$$V(a_1, a_2, a_3, a_4) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ a_1 & a_2 & a_3 & a_4 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 \end{pmatrix}$$

a) Defina uma função V que, dada uma lista com os valores de $a_1, a_2, \ldots, a_n$, permita obter a matriz de Vandermonde de ordem n. Teste a sua função para obter as matrizes:

i. $V_1 = V(a_1, a_2, a_3, a_4)$;

ii. $V_2 = V(1, -2, 3, -1, 1)$.

b) Mostre que $\det V_1 = (a_1 - a_2)(a_1 - a_3)(a_2 - a_3)(a_1 - a_4)(a_2 - a_4)(a_3 - a_4)$.

c) Obtenha, de duas formas distintas, o número de elementos de V_2 iguais a 1.

d) Defina a matriz V_3, retirando as linhas 4 e 5 e a primeira coluna de V_2.

Exercício 4.16⊛ Considere as matrizes:

$$A = \begin{pmatrix} 1 & -1 & 2 \\ 1 & -1 & -1 \\ -3 & 4 & 1 \end{pmatrix} \quad \text{e} \quad B = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ -\sin\theta & -\cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

a) Calcule o determinante de A e de B.

b) Calcule o determinante da matriz que se obtém eliminando a primeira linha e a primeira coluna da matriz A.

c) Determine as matrizes $\operatorname{adj} A$ e $\operatorname{adj} B$.

d) Determine $\theta \in [0\ 2\pi]$ tal que $B\,(1\ 1\ 1)^T = (-1\ \ -1\ 1)^T$.

Exercício 4.17⊛ Para cada uma das aplicações lineares seguintes, determine o núcleo e a sua dimensão e diga se a aplicação é injetiva e/ou sobrejetiva.

a) $f : \mathbb{R}^3 \to \mathbb{R}^3$, definida por $f(x,y,z) = (x, x-y, x+z)$.

b) $h : \mathbb{R}^4 \to \mathbb{R}^3$, definida por
$$\begin{aligned} h(1,0,0,0) &= (1,-1,2) \\ h(0,1,0,0) &= (-2,5,3) \\ h(0,0,1,0) &= (-7,16,7) \\ h(0,0,0,1) &= (-3,6,-1). \end{aligned}$$

Exercício 4.18 Verifique quais dos seguintes vetores são vetores próprios da matriz $\begin{pmatrix} -5 & 2 & 0 \\ -12 & 5 & 0 \\ 0 & 0 & -1 \end{pmatrix}$.

a) $(0,0,2)$ *b)* $(1,3,0)$ *c)* $(1,1,3)$

Output esperado: Sim; Sim; Não

Exercício 4.19 Verifique quais dos seguintes valores são valores próprios da matriz $\begin{pmatrix} 4 & 0 & 2 \\ 1 & 1 & 3 \\ 0 & 0 & -2 \end{pmatrix}$.

a) 2 *b)* -2 *c)* 4 *d)* 1 *e)* 0

Output esperado: Não; Sim; Sim; Sim; Não

Exercício 4.20⊛ Considere as matrizes:

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \qquad C = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix} \qquad D = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 2 \end{pmatrix}.$$

a) Escreva a equação característica e calcule os valores e vetores próprios das matrizes.

b) Verifique se as matrizes são diagonalizáveis.

⇛ Alguns comandos

```
AntihermitianMatrixQ AntisymmetricMatrixQ ArrayReshape CharacteristicPolynomial
ConjugateTranspose Det Diagonal DiagonalMatrix DiagonalizableMatrixQ Dot
Eigensystem Eigenvalues Eigenvectors HermitianMatrixQ IdentityMatrix
Insert Inverse LinearSolve LowerTriangularize MatrixForm MatrixPower
MatrixQ MatrixRank NullSpace Reduce RowReduce Solve SquareMatrixQ
SymmetricMatrixQ Take Tr Transpose UpperTriangularize
```

⇛ Links

http://reference.wolfram.com/language/guide/ConstructingMatrices
http://reference.wolfram.com/language/guide/MatrixOperations
http://reference.wolfram.com/language/guide/LinearSystems
http://reference.wolfram.com/language/guide/MatricesAndLinearAlgebra

5. Operadores e funções booleanas

⇛ Exercícios

Exercício 5.1 Escreva uma função-predicado ou função booleana, `naturalQ`, que indique se o seu argumento é um número natural. Mostre alguns exemplos da sua utilização.

Exercício 5.2 Diga, mostrando alguns exemplos, qual a diferença entre as duas funções booleanas `ListQ` e `VectorQ`.

Exercício 5.3 Construa uma função-predicado que indique se o seu argumento é superior a 2. Utilize-a como critério com o comando `Select` sobre uma lista por si escolhida.

Exercício 5.4 Construa uma função-predicado que indique se o seu argumento é um número complexo cuja parte real é igual à parte imaginária. Mostre alguns exemplos da sua utilização.

Exercício 5.5$^{\circledast}$ Construa uma função-predicado, com dois argumentos, que indique se os argumentos são inteiros primos entre si.

Exercício 5.6 Construa uma função-predicado, `vogalQ`, que indique se o seu argumento é uma vogal.

Exercício 5.7 Escreva uma função-predicado, `capicuaQ`, que indique se o seu argumento é uma capicua. Mostre alguns exemplos da sua utilização.

Exercício 5.8$^{\circledast}$ **Palíndromo** é uma palavra ou grupo de palavras em que o sentido é o mesmo, quer se leia da esquerda para a direita quer da direita para a esquerda. Por exemplo, "somos"ou "amor a roma" são palíndromos. Escreva uma função-predicado, `palindromoQ`, que indique se o seu argumento é um palíndromo. Mostre alguns exemplos da sua utilização.

Exercício 5.9 No calendário gregoriano, um ano A é bissexto se A é divisível por 400 ou se A é divisível por 4 mas não por 100. Construa uma função-predicado, `bissextoQ`, que indique se um dado ano é bissexto.

Exercício 5.10 Um **número de Mersenne** é um número da forma $2^n - 1$, onde n é um número natural. Construa uma função-predicado, `MersenneQ`, que indique se o seu argumento é um número de Mersenne. Use esta função para obter alguns primos de Mersenne.

Lógica com o Mathematica

Exercício 5.11 Se q e r são proposições verdadeiras e p é falsa, qual é o valor lógico de cada uma das seguintes proposições?

a) $(p \Rightarrow \neg q) \ \vee \neg(r \wedge q)$

b) $(\neg p \ \vee \neg q) \Rightarrow (p \ \vee \neg r)$

c) $\neg(\neg p \Rightarrow \neg q) \ \vee r$

Output esperado: True; False; True

Exercício 5.12 Construa as tabelas de verdade para cada uma das seguintes fórmulas proposicionais.

a)⊛ $(p \ \vee q) \wedge \neg(p \wedge q)$

b) $(p \Rightarrow \ q) \ \vee (p \Rightarrow \neg q)$

c) $(p \wedge \neg q) \ \vee \neg(p \Leftrightarrow q)$

Exercício 5.13 Verifique se as seguintes fórmulas proposicionais são equivalentes.

a) $p \Rightarrow (q \wedge \neg q)$ e $\neg p$

b) $((p \Rightarrow q) \Rightarrow q) \Rightarrow q$ e $p \Rightarrow q$

c) $p \vee (\neg q \wedge r)$ e $q \ \vee \neg r \Leftrightarrow p$

Output esperado: True; False; False

Exercício 5.14 Verifique quais das seguintes fórmulas proposicionais são tautologias e quais são contradições.

a)⊛ $(p \Rightarrow q) \wedge (\neg q \Rightarrow \neg p)$

b) $(p \wedge q) \wedge \neg(p \vee \neg q)$

c) $(p \vee q) \Leftrightarrow (\neg p \Rightarrow q)$

⇛ Alguns comandos

`AllTrue` `And` `AnyTrue` `BooleanConvert` `BooleanTable` `Conjunction` `Disjunction` `Equal` `Equivalent` `False` `Implies` `Not` `Or` `TautologyQ` `True` `Unequal`

⇛ Links

http://reference.wolfram.com/language/guide/LogicAndBooleanAlgebra
http://reference.wolfram.com/language/guide/BooleanComputation

6. Gráficos

⇛ Exercícios

Exercício 6.1 Desenhe o gráfico do conjunto de pontos $\{(1,3),(2,5),(2,1),(3,0),(4,4),(4,3)\}$; una os pontos dados por segmentos de reta.

Exercício 6.2 Desenhe o triângulo cujos vértices são (1,0), (4,0) e (2,3); desenhe o mesmo triângulo a tracejado.

Exercício 6.3 Desenhe a circunferência centrada em $(0,0)$ e de raio 1:

a) recorrendo ao comando `Plot`;

b) usando o comando `ParametricPlot`;

c) utilizando o comando `RegionPlot`.

Exercício 6.4⊛ Desenhe em simultâneo a circunferência de equação $x^2 + y^2 = 1$ e a elipse de equação $\frac{x^2}{4} + y^2 = 1$. Use traço azul contínuo para a circunferência e tracejado laranja para a elipse.

Exercício 6.5 Desenhe o gráfico das funções:

a) $\cos x,\ x \in [-2\pi,\ 2\pi]$;

b) $x^2 + 7x + 1,\ x \in [-10,\ 6]$;

c) $e^{-x^2},\ x \in [-3,\ 3]$;

d) $\operatorname{tg} x,\ x \in [-2\pi,\ 2\pi]$.

Use os comandos `GraphicsGrid` e `GraphicsRow` para representar conjuntamente os gráficos anteriores.

Exercício 6.6⊛ Reproduza o seguinte gráfico

Exercício 6.7 Desenhe o gráfico do seno-hiperbólico no intervalo $[-3, 3]$, assinalando os eixos dos xx e dos yy. Coloque também o gráfico dentro de uma moldura e escreva "*função seno-hiperbólico*" como legenda da figura.

Exercício 6.8 Desenhe o gráfico da função $f(x) = \mathrm{e}^x \cos x$, para valores de x no intervalo $[0, 5\pi]$. Faça alguns ajustes nas opções do `Plot` de forma a ter a noção do comportamento da função no intervalo dado.

Exercício 6.9 *Estude* o comportamento da função $\mathrm{e}^{-(x-3)^2 \cos(4x-12)}$, no intervalo $[1.5, 4.2]$.

Exercício 6.10⊛ Desenhe em simultâneo os gráficos das funções

$$f(x) = \operatorname{sen}^2 x, \quad g(x) = \cos x, \quad h(x) = \operatorname{sen}^2 x \cos x,$$

quando x percorre o intervalo $[-4\pi, 4\pi]$. Use cores diferentes para cada gráfico.

Exercício 6.11 Desenhe o gráfico da curva definida implicitamente por

a) $x^2 + 2y^2 = 3$

b) $2x + x^3y - y^3 = 1$

⇛ Alguns comandos

`Arrow` `AspectRatio` `Axes` `Circle` `Disk` `Epilog` `Filling` `FrameLabel` `GraphicsGrid` `GraphicsRow` `Joined` `Line` `ListLinePlot` `ListPlot` `Manipulate` `Plot` `PlotLabel` `PlotLegends` `PlotRange` `PlotStyle` `PlotStyle` `Point` `Polygon` `Prolog` `Rectangle` `Show` `Text` `Ticks`

⇛ Links

http://reference.wolfram.com/language/guide/FunctionVisualization

http://reference.wolfram.com/language/guide/GraphicsObjects

http://reference.wolfram.com/language/guide/GraphicsOptionsAndStyling

http://reference.wolfram.com/language/guide/GraphicsDirectives

7. Funções reais

Cálculo com o Mathematica

⇛ Exercícios

Exercício 7.1 Encontre fatorizações dos seguintes polinómios.

a) $x^2 - 2x - 3$ *b)* $x^4 - 6x^3 + 9x^2 - 6x + 8$ *c)* $x^3 + x^2 - 2x - 2$ *d)* $x^2 + 2x - 6$ *e)* $x^2 + x + 3$

Exercício 7.2 Decomponha as seguintes funções racionais em funções racionais elementares:

a) $\dfrac{x+4}{x^2+5x+6}$ *b)* $\dfrac{(x-3)^2}{x^2-8x+12}$ *c)* $\dfrac{2x+7}{x^3+3x^2+3x+1}$

Exercício 7.3 Encontre uma expressão simplificada para a função racional

$$f(x) = \frac{1}{4(x-1)} - \frac{1}{4(x+1)} - \frac{1}{2(1+x^2)} .$$

Exercício 7.4 Calcule os seguintes limites e, para cada alínea, faça também a representação gráfica da função envolvida.

a) $\lim_{x\to 0} \frac{1}{x}$ *b)* $\lim_{x\to \frac{\pi}{2}^+} \operatorname{tg} x$ *c)* $\lim_{x\to \frac{\pi}{2}^-} \operatorname{tg} x$ *d)*⊛ $\lim_{x\to\infty} \frac{x^2-1}{2+\cos x}$ *e)* $\lim_{x\to 0} \operatorname{sen} \frac{1}{x}$

Exercício 7.5 Defina as funções $f_1(x) = \sqrt{x}$ e $f_2(x) = x + \operatorname{sen} x$.

a) Para $i = 1, 2$, calcule o quociente $(f_i(x+h) - f_i(x))/h$, onde h é um parâmetro.

b) Calcule o limite, quando h tende para zero, das expressões encontradas na alínea *a)*.

Exercício 7.6⊛ Determine as seguintes derivadas da função $f(x) = x \cos x^2 \, \mathrm{e}^{-x}$.

a) $f'(x)$ *b)* $f''(x)$ *c)* $f^{(8)}(x)$

Exercício 7.7 Determine a quarta derivada da função $f(x) = x\, e^{-x^2/2} + 5\, x^7 \log x$.

Exercício 7.8 Seja $f(x) = \frac{\operatorname{sen} x}{x}$. Desenhe, simultaneamente os gráficos das funções f, f' e f''.

Exercício 7.9⊛ Encontre todos os zeros dos polinómios $p(x) = -28 + 102x - 126x^2 + 93x^3 - 46x^4 - 30x^5 + 54x^6 - 21x^7 + 2x^8$ e $q(x) = -7 - 3x + 2x^2 - 4x^3 + x^4 + x^5$.

Exercício 7.10 Resolva as seguintes equações não lineares:

a) $\cos x - x = 0$ *b)* $\operatorname{sen} x - x^2 = 0$ *c)* $x \log x - 1 = 0$ *d)* $x - \mathrm{e}^x = 1/4$

Exercício 7.11 Determine os pontos do plano onde os seguintes pares de funções se intersetam:

a) $f(x) = x^3 - 8x^2 + 19x - 12$ e $g(x) = 4x^2 - 8x - 1$;

b) $f(x) = \mathrm{e}^{-(x/4)^2}\cos(x/\pi)$ e $g(x) = 5/4 + \operatorname{sen} x^{3/2}$.

Exercício 7.12 Determine o valor mínimo da função $g(x) = x^2 + 1 - \operatorname{sen} x$.

Exercício 7.13 Determine os pontos de máximo local, pontos de mínimo local e os pontos de inflexão da função $f(x) = x^3 - \mathrm{e}^x$.

Exercício 7.14 Calcule uma primitiva de cada uma das seguintes funções:

a) $f(x) = x\log^2 x$ *b)* $g(x) = x\cos x\, e^{-x}$ *c)* $h(x) = \operatorname{sen}^2 x\cos^3 x$

Exercício 7.15 Considere a função

$$f(x) = \begin{cases} \operatorname{sen} x^2, & \text{se } x > 0 \\ x - x^3, & \text{se } x \leq 0 \end{cases}$$

a) Desenhe o gráfico de $f(x)$, para $x \in [-3, 3]$, e conclua (e confirme) que f' não é contínua em $x = 0$.

b) Calcule os limites $\lim\limits_{x \to 0^{\pm}} f'(x)$.

Exercício 7.16⊛ Considere a função $f : [\pi, 2\pi] \to \mathbb{R}$ tal que $f(x) = x + \operatorname{sen} x + \operatorname{sen}\frac{x\pi}{2}$.

a) Defina a função f e represente o seu gráfico; designe o objeto gráfico criado de `graff`.

b) Usando o objeto gráfico `graff`, acrescente-lhe a reta tangente ao gráfico de f no ponto de abcissa 5. Assinale o ponto de tangência.

c) Considere o objeto gráfico `graff`. Extraia a lista das coordenadas dos pontos utilizados na construção da curva correspondente ao gráfico de f. Designe essa lista por `listaf`.

d) Represente em simultâneo os gráficos de f e da sua inversa, f^{-1}.

Exercício 7.17⊛ Considere a função $f : \mathbb{R} \to \mathbb{R}$ tal que $f(x) = \frac{\operatorname{sen}(2x) - x + 3}{\sqrt{x^2+1}}$.

a) Defina a função f.

b) Represente, em simultâneo, os gráficos de f e da sua derivada, no intervalo [-10,10], especificando a cor e a espessura dos traços usados; designe o objeto gráfico criado de `graff`.

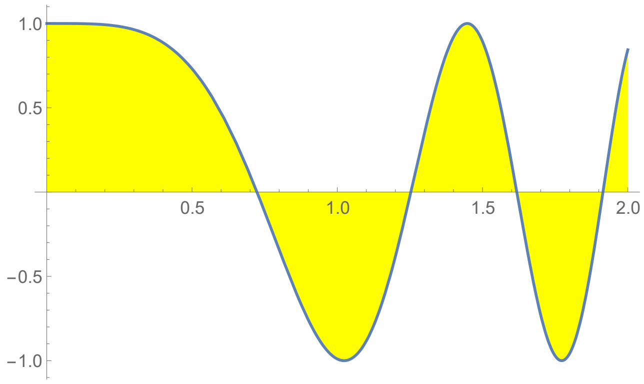
c) Determine, com precisão de máquina, o menor zero de f' no intervalo [0, 10].

d) Estude a existência de assintotas horizontais da função f.

e) Assinale no gráfico obtido na alínea *b)* o máximo da função f no intervalo [-10,10] e as assíntotas obtidas na alínea anterior.

Exercício 7.18 Considere a função $f(x) = \cos(3x^2)$.

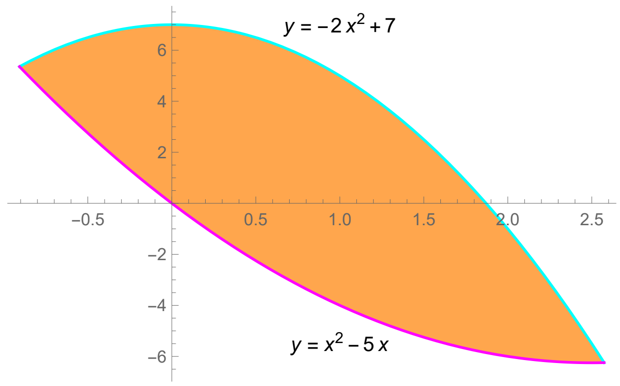
a) Utilize a opção `Filling` para desenhar o gráfico de $f(x)$ e obter a seguinte figura:



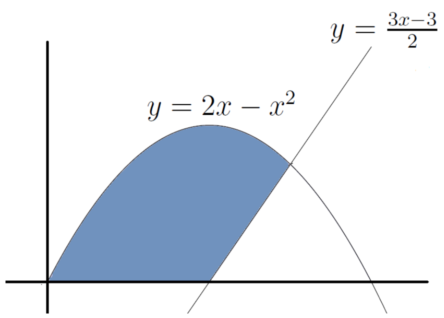
b) Calcule a área da primeira das regiões salientadas.

Output esperado: 0.564332

Exercício 7.19 Mostre que a área da região do plano indicada abaixo é igual a $\frac{109}{54}\sqrt{109}$.



Exercício 7.20$^{\circledast}$ Determine a área da região sombreada na figura.



Output esperado: $\frac{15}{16}$

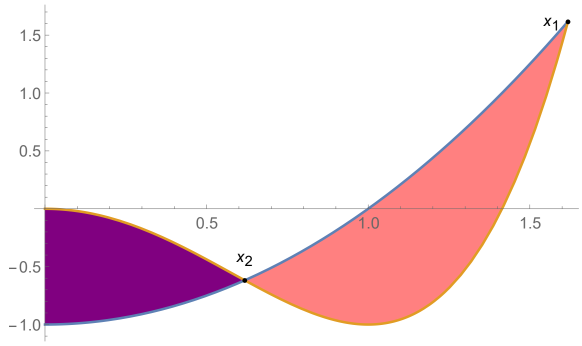
Exercício 7.21⊛ Defina a função $f(x) = x^2 - 1$.

a) Calcule $f(5)$, $f(1.75)$, $f(\sqrt{19})$ e $f(f(x))$.

b) Desenhe o gráfico de $f(x)$ e $f(f(x))$, com $x \in [-2, 2]$.

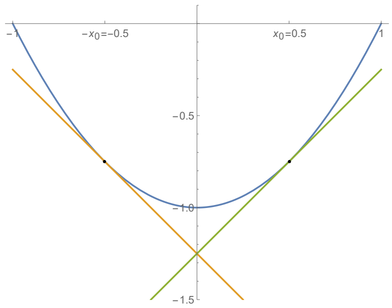
c) Determine as soluções da equação $f(x) = (f \circ f)(x)$.

d) Designando por x_1 a maior das soluções encontradas na alínea anterior, represente os gráficos de f e $f \circ f$, sombreando a região delimitada pelas duas curvas no intervalo $[0, x_1]$. Deverá obter um gráfico análogo ao seguinte:



e) Determine o valor da área sombreada na alínea anterior.

f) Determine o ponto $x_0 \in \mathbb{R}^+$ tal que as retas tangentes a f em $-x_0$ e em x_0 se intersetam ortogonalmente. Represente graficamente a função f, as retas tangentes e os respetivos pontos de tangência. Deverá obter um gráfico análogo ao seguinte:



⇛ Alguns comandos

`Apart` `D` `Direction` `Expand` `ExpandAll` `Extension` `Factor` `FindRoot` `Integrate` `Limit` `NSolve` `Solve` `Together`

⇛ Links

http://reference.wolfram.com/language/guide/Calculus

http://reference.wolfram.com/language/guide/PolynomialFactoring

http://reference.wolfram.com/language/guide/PolynomialAlgebra

8. Funções de ordem superior

⇛ Exercícios

Exercício 8.1[⊛] Construa uma lista ℓ de 20 números ímpares positivos aleatórios, menores que 50.

a) Calcule a média dos valores de ℓ, usando

:: as funções `Apply, Plus` e `Length`;

:: as funções `Total` e `Length`;

:: a função `Mean`.

b) Calcule a mediana dos valores de ℓ, usando

:: as funções `Plus, Take` e `Sort`;

:: a função `Median`.

Exercício 8.2 A partir da expressão $a + \dfrac{b}{c} + \dfrac{d+e}{f+g}$, obtenha a expressão $a + b + \dfrac{1}{c} + d + e + \dfrac{1}{f+g}$.
(Sugestão: Usar a função `FullForm`)

Exercício 8.3 Defina a função $g(x) = a + b\cos x$. Aplique a função g aos elementos 3 e 27 da lista $\{1, 2, 3, \{3, 4, 5, \{27, 9, 3\}\}\}$.

Exercício 8.4 Usando as funções `Apply` e `Expand`, transforme a soma $5a+3b+\dfrac{2c+5a}{7}$ na lista $\left\{\dfrac{40a}{7}, 3b, \dfrac{2c}{7}\right\}$.

Exercício 8.5 Considere as funções $f(x) = x + x^2$, $g(x) = 1 + x^3$ e $h(x) = \operatorname{sen} x + \cos x$. Calcule, usando os comandos `Composition` e `Nest`:

a) $(f \circ g)(x)$ *b)* $(g \circ f)(1 - x)$ *c)* $(f \circ h)(\pi/3)$ *d)* $(g \circ g \circ g \circ g)(x)$

Exercício 8.6 Sejam $f(x)$ e $g(x)$ duas funções reais, diferenciáveis. Calcule a derivada das funções compostas $f(f(f(x)))$ e $f(g(x))$.

Exercício 8.7 Considere as funções $f(x) = 6\,x(1-x)$ e $f^{12}(x)$ (composição de f consigo mesma 12 vezes). Se $x_0 = \frac{7+\sqrt{21}}{12}$, obtenha $f^{12}(x_0)$, utilizando:

a) uma aproximação numérica com 10 algarismos significativos para x_0.

b) uma aproximação numérica com 12 algarismos significativos para x_0.

c) as capacidades simbólicas do *Mathematica* (e o comando `Simplify`).

⇛ Alguns comandos

`Apply` `ApplyTo` `Composition` `FoldList` `Fold` `Inner` `MapAll` `MapAt` `MapThread` `Map` `NestList` `Nest`

⇛ Links

http://reference.wolfram.com/language/tutorial/ApplyingFunctionsToPartsOfExpressions
http://reference.wolfram.com/language/tutorial/ApplyingFunctionsToListsAndOtherExpressions
http://reference.wolfram.com/language/tutorial/ApplyingFunctionsRepeatedly

9. Funções anónimas

⇛ Exercícios

Exercício 9.1 Defina uma função de *forma clássica* para realizar as seguintes operações especificadas por funções anónimas:

a)
```
(1+#^3)&
```

b)
```
{#,#^2}&
```

c)
```
{#1,#2^{#3}}&
```

Exercício 9.2 Construa uma função anónima correspondente a cada uma das seguintes funções:

a)
```
f[x_]:= 1-4x+x^2
```

b)
```
g[x_,y_]:= x-Cos[y]
```

c)
```
h[expr_]:= Length[expr]>=4
```

d)
```
p[x_]:= {Re[x],Im[x]}
```

Exercício 9.3 Construa uma função anónima que dada uma lista de números complexos, represente esses números no plano d'Argand.

Exercício 9.4 Construa, como função anónima, uma função booleana que indique se o seu argumento, uma lista de dois números, é um ponto do terceiro quadrante do plano.

Output esperado: {-1, -2} → `True`

Exercício 9.5 Construa uma função anónima booleana que indique se o seu argumento é uma lista com exatamente 3 elementos. Utilize-a como critério da função `Select` sobre uma lista por si escolhida.

Exercício 9.6⊛ Construa uma função anónima booleana que indique se o seu argumento é uma lista que contém só números pares. Utilize-a como critério da função `Select` sobre uma lista por si escolhida.

Exercício 9.7⊛ Considere uma lista de 10 listas de números inteiros entre 0 e 12, geradas aleatoriamente, com a dimensão de cada uma das listas igual a um número natural gerado aleatoriamente entre 2 e 8. Selecione aquelas listas que tenham o número 7 como seu elemento.

Exercício 9.8⊛ Dois números primos dizem-se **primos gémeos** se diferem 2 unidades, i.e., $(p, p+2)$ é um par de primos gémeos se p e $p+2$ são primos. Os primeiros pares de primos gémeos são $(3,5)$, $(5,7)$, $(11,13)$, $(17,19)$, ... Construa uma lista com todos os pares de números primos gémeos menores que 100.

Exercício 9.9 Construa uma função, `rep`, que dada uma lista devolva os elementos que aparecem repetidos. Ilustre o uso desta função escolhendo várias listas. Use esta função para obter os divisores comuns de dois números.

Output esperado: `rep`[1,2,4,2,2,5,3,5] → {2,5}

⇛ Alguns comandos

`Function` `Slot` `SlotSequence`

⇛ Links

http://reference.wolfram.com/language/tutorial/PureFunctions

10. Moldes

⇛ Exercícios

Exercício 10.1 Defina uma função que aceite como argumento um número inteiro n e devolva a lista dos inteiros de 0 a $|n|$.

Exercício 10.2 Defina uma função que aceite como argumento as coordenadas de dois pontos do plano e tenha como resultado a distância entre eles.

Exercício 10.3 Defina uma função que aceite como argumento uma matriz quadrada e tenha como resultado o módulo do seu maior valor próprio em módulo.

Exercício 10.4 Defina uma função que aceite como argumento um número inteiro positivo n e tenha como resultado um polinómio de grau n de coeficientes aleatórios entre -1 e 1.

Exercício 10.5

a) Considere um número primo p aleatoriamente escolhido entre os 20 primeiros números primos.

b) Sem recorrer à função `Divisible` ou outra análoga, crie uma função-predicado, `divisor`, que, aceitando como argumento dois números inteiros devolva `True` no caso do primeiro elemento do argumento ser divisor do segundo.

c) Use a função que criou para mostrar que todos os elementos do conjunto $\{\binom{p}{k}, k = 1, \ldots, p-1\}$ são divisíveis por p.

Exercício 10.6[⊛] Defina uma função que aceite como argumento um qualquer polinómio na variável x e apresente como resultado uma lista dos seus zeros reais.

Exercício 10.7 Recorde que **primos de Mersenne** são primos da forma $2^n - 1$.

a) Construa a lista de pares da forma $\{n, 2^n - 1\}$, quando n percorre os naturais menores ou iguais a 20.

b) Use o comando `Select` para selecionar da lista construída na alínea a) os pares cujo segundo elemento é primo de Mersenne. Apresente esta lista na forma de tabela.

c) Designando por `coluna1` a lista contendo os elementos da primeira coluna da tabela que obteve na alínea anterior, contrua a lista $\{2^{s-1}(2^s - 1) : s \in \texttt{coluna1}\}$.

d) Crie uma função-predicado, `perfeitoQ`, que aceitando como argumento um número inteiro positivo, devolva `True`, caso o argumento seja um **número perfeito**, i.e., um número igual à soma dos seus divisores próprios (não deve recorrer à função `PerfectQ`).

e) Mostre que a lista construída na alínea c) é formada por números perfeitos.

Exercício 10.8

a) Construa uma tabela 6×6 de zeros e uns, gerada aleatoriamente. Designe-a por `tabela`.

b) Defina um molde que identifique listas cujos elementos são todos iguais a 1.

c) Identifique as linhas da `tabela` que *encaixam* no molde anterior.

Exercício 10.9 Defina uma função que aceite como argumento as coordenadas de três pontos não colineares do plano e tenha como resultado o gráfico do triângulo por eles formado e o valor do seu perímetro.

Exercício 10.10[⊛]

a) Defina uma função, `rep`, que aceite como argumento uma lista formada pela repetição de um único elemento e devolva o elemento repetido da lista.

Output esperado: `rep`[{x,x,x}] → x

b) Usando o comando `Split` e a função `rep` defina a função `eliminarep` que aceita como argumento uma lista e devolve a lista que resulta da que é dada como argumento depois da eliminação dos elementos repetidos que se encontram em posições adjacentes.

Output esperado: `eliminarep` [{x,y,y,y,x,x,z,y,y,z,z,z,z}] → {x,y,x,z,y,z}

c) Considere o molde seguinte: `{a___,b_,b_,c___}`

Caracterize os objetos que *encaixam* neste molde.

d) Avalie a código seguinte e descreva a função `eliminaRep`.

```
eliminaRep[lista_List] := lista //. {a___, b_, b_, c___} -> {a, b, c}
```

e) Teste a função `eliminaRep` definida na alínea (d), tomando como argumento uma lista aleatória de 100 elementos formada por zeros e uns.

Exercício 10.11

a) Construa a lista de todos os números inteiros de quatro algarismos. Designe essa lista por `lista1`.

b) Usando o comando `IntegerDigits`, obtenha a lista, que designará por `lista2`, cujos elementos são listas com os quatro algarismos dos elementos da `lista1`.

c) Defina um molde que identifique listas da forma `{p,p,q,q}`.

d) Selecione da `lista2` os elementos que se encaixam no molde anterior.

Exercício 10.12 Em teoria de matrizes são importantes as matrizes quadradas cujos elementos são números inteiros e cuja matriz inversa é ainda constituída unicamente por números inteiros. Construa uma função booleana, `inteirosQ`, que identique esse tipo de matriz.

Output esperado: `inteirosQ`[{{3,1},{2,1}}] → `True`

Exercício 10.13 Defina uma função que aceite como argumento uma matriz de números inteiros e tenha como resultado o máximo das somas dos módulos dos elementos de cada linha.

Exercício 10.14⊛ Defina uma função, `polinomio`, que aceite como argumento um qualquer número de número inteiros e tenha como resposta o polinómio, na variável x, com esses coeficientes.

Output esperado: `polinomio`[1,2,3,4] $\to 1 + x + x^2 + x^3$

Exercício 10.15 Defina uma função que aceite como argumento um qualquer número de listas de números inteiros e tenha como resultado a lista dos elementos comuns a todas essas listas.

⇛ Alguns comandos

`Alternatives` `Blank` `BlankNullSequence` `BlankSequence` `Cases` `Condition` `Count` `DeleteCases` `Longest` `MatchQ` `MemberQ` `Optional` `Pattern` `PatternTest` `Position` `Repeated` `RepeatedNull` `ReplaceAll` `Shortest`

⇛ Links

http://reference.wolfram.com/language/guide/guide/Patterns
http://reference.wolfram.com/language/guide/RulesAndPatterns

11. Recursividade

⇛ Exercícios

Exercício 11.1 O **fatorial duplo** de um inteiro não negativo n é definido recursivamente do seguinte modo:

$$n!! = \begin{cases} 1, & \text{se } n = 0 \text{ ou } n = 1, \\ n(n-2)!!, & \text{se } n \geq 2. \end{cases}$$

Escreva uma função que, dado o valor de n, calcule $n!!$.

Output esperado: $6!! = 6 \times 4 \times 2 = 48$ e $7!! = 7 \times 5 \times 3 \times 1 = 105$

Exercício 11.2 Escreva uma função recursiva que, dada uma lista de números, calcule a soma dos elementos da lista.

Output esperado: {x,y,z,w}→ x+y+z+w

Exercício 11.3 Escreva uma função recursiva que, dada uma lista com um número par de elementos, retorne uma lista com o produto de cada par consecutivo de elementos presentes na lista.

Output esperado: {x,y,z,w}→ {xy,zw}

Exercício 11.4 Escreva uma função recursiva que, dada uma lista de números, calcule o seu máximo.

Exercício 11.5 Considere a sucessão a_n definida por:

$$a_n = a_{n-1} + 2a_{n-2}; \quad n \geq 2$$
$$a_0 = 2; \qquad a_1 = 7.$$

a) Escreva uma função recursiva para, dado um valor de n, obter o termo a_n. Obtenha os 10 primeiros termos desta sucessão.

b) Apresente uma solução não recursiva para o problema anterior.

c) Sabendo que o termo geral da sucessão é $a_n = 3 \times 2^n + (-1)^{n+1}$, obtenha os primeiros 10 termos da sucessão, usando uma função anónima.

Exercício 11.6 Defina uma função recursiva que, dada uma lista, elimine os elementos repetidos colocados em posições adjacentes. (Reveja o Exercício 10.10).

Output esperado: {x,y,y,y,x,x,z,y,y,z,z,z,z} → {x,y,x,z,y,z}

Exercício 11.7 Defina uma função recursiva que aceite como argumento uma lista e permute os seus elementos dois a dois.

Output esperado:{1,2,3}→ {2,1,3}; {1,2,3,4} → {2,1,4,3}

Exercício 11.8 Os **polinómios de Legendre**[1] $\mathcal{P}_n(x)$ são definidos pela seguinte relação de recorrência

$$n\mathcal{P}_n(x) - (2n-1)x\mathcal{P}_{n-1}(x) + (n-1)\mathcal{P}_{n-2}(x) = 0,$$

com $\mathcal{P}_0(x) = 1$ e $\mathcal{P}_1(x) = x$.

a) Calcule os seis primeiros polinómios de Legendre, usando a relação de recorrência apresentada.

b) Represente graficamente os 6 primeiros polinómios de Legendre, no intervalo $[-1, 1]$.

c) Repita o exercício, recorrendo agora à função `LegendreP` do Mathematica.

Exercício 11.9⊛ A **raiz digital** de um número inteiro positivo n é definida como sendo o número de um só dígito que se obtém somando os dígitos de n. Quando a soma possui mais do que um dígito, o processo é repetido até que a soma final se exprima apenas com um dígito.

Por exemplo, a raiz digital de 2094 é 6, porque 2+0+9+4=15 e 1+5=6.

a) Construa uma função predicado `umdigitoQ` cujo resultado seja, para cada n inteiro positivo, `True` se n tem apenas um dígito e `False`, caso contrário.

b) Construa uma função recursiva `raizdigital` cujo resultado seja, para cada n inteiro positivo, a raiz digital de n.

c) Use a função `FixedPoint` para obter uma outra solução do problema.

d) Obtenha uma lista com as raizes digitais dos primeiros 100 números naturais.

e) Verifique, para a lista usada em d), que a raiz digital de n pode ser obtida como

$$1 + \text{“resto da divisão de } n-1 \text{ por 9”}$$

Exercício 11.10⊛ Considere o seguinte algoritmo[2] que gera uma sequência de inteiros:

- Comece com um inteiro positivo n.
- O próximo número da sequência será $n/2$, se n for par ou $3n+1$ se n for ímpar.
- Repita este processo com o novo valor obtido, terminando quando atingir 1.

Por exemplo, a seguinte sequência de números será gerada para o valor inicial $n = 22$:

`22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1.`

a) Implemente este algoritmo e teste-o para $n = 22, 35, 70$.

b) Para cada n, chama-se **vida** de n ao número de elementos da sequência obtida. Por exemplo, a vida de $n = 22$ é 16. Determine a vida de 35 e 70.

c) Represente graficamente os valores da vida de n, para valores de n entre 2 e 30.

[1] https://mathworld.wolfram.com/LegendrePolynomial.html

[2] Este problema é conhecido como **Problema de Collatz** ou **sequência** $3n+1$. A *Conjetura de Collatz* afirma que este algoritmo termina para todo n positivo.Veja, por exemplo, http://mathworld.wolfram.com/CollatzProblem.html e http://oeis.org/A006577.

Exercício 11.11⊛ (Formulação alternativa do problema de Collatz) Dado um número natural k, a sucessão de Collatz de valor inicial k, $(c_n^k)_n$, é definida através da seguinte fórmula de recorrência:

$$c_0^k = k; \quad c_n^k = \begin{cases} \frac{c_{n-1}^k}{2}, & \text{se } c_{n-1}^k \text{ par} \\ 3c_{n-1}^k + 1, & \text{se } c_{n-1}^k \text{ ímpar} \end{cases}$$

a) Construa uma função f que aceite como argumento um número natural n e que tenha como resultado $n/2$, se n é par ou $3n+1$, se n é ímpar.

b) Defina uma função `collatz[k,n]` , tal que, dados $k \in \mathbb{N}$ e $n \in \mathbb{N}_0$, `collatz[k,n]`$= c_n^k$.

c) Construa a lista `listaCollatz`$= \{\{c_n^k\}_{n=0,\ldots,20}\}_{k=1,\ldots,10}$.

d) Defina um molde onde encaixem listas contendo pelo menos um elemento igual a 1.

e) Construa uma função, `eliminar`, que aceite como argumento listas que encaixem no molde definido na alínea anterior e devolva a sublista contendo os elementos da lista dada como argumento, à qual foram retirados os elementos que surgem a seguir à primeira ocorrência do elemento 1.

f) Explique como pode testar a Conjetura de Collatz a partir das funções construídas e confirme a sua resposta, aplicando a função `eliminar` a `listaCollatz`.

⇛ Links

http://reference.wolfram.com/language/guide/RecurrenceAndSumFunctions

12. Leitura e escrita de ficheiros

⇛ Exercícios

Exercício 12.1⊛ Considere o ficheiro `plano.xlsx` que contém a estrutura curricular da Licenciatura em Matemática da Universidade do Minho.

a) Comece por fixar como diretoria de trabalho, a diretoria onde gravou o ficheiro `plano.xlsx`.

b) Importe o ficheiro e apresente o seu conteúdo na forma de tabela.

c) Transforme os valores correspondentes ao ano letivo em números inteiros.

d) Quantas UCs existem da área científica da Matemática?

e) Quais são as UCs do 2º semestre do 1º ano?

f) Quantas Opções UMinho estão listadas?

g) Acrescente à lista de Opções UMinho a UC com 6 ECTS "Ferramentas para aulas tecnologicamente assistidas".

h) Grave o novo plano com o nome `plano2.xlsx`.

i) Quais são as UCs da Opção I?

j) Confirme que a Licenciatura em Matemática tem 180 ECTS.

Exercício 12.2⊛ Considere o ficheiro `proverbios.txt`.

a) Comece por fixar como diretoria de trabalho, a diretoria onde gravou o ficheiro `proverbios.txt`.

b) Este ficheiro, contém vários provérbios. Use uma função do Mathematica que permita ver (sem importar o ficheiro) o seu conteúdo.

c) Execute o comando seguinte e descreva o seu objetivo.

```
listaproverbios=ReadList["proverbios.txt",Word,WordSeparators->{"\n"}]
```

d) Quantos provérbios contém o ficheiro `proverbios.txt`?

e) Qual é o 5º provérbio do ficheiro `proverbios.txt`?

f) Retire as vírgulas do texto de cada provérbio.

g) Quantas vezes aparece a palavra amigo no texto dos provérbios?

h) Qual é o provérbio que tem mais palavras?

i) Acrescente a `listaproverbios`, o provérbio: A conselho amigo, não feches o postigo.

Na resolução desta pergunta, poderão ser úteis as funções `StringReplace`, `StringCases` ou `StringCount`.

Exercício 12.3 O ficheiro `classificacoes.xlsx` contém as classificações de Matemática Computacional II nos diversos momentos de avaliação.

a) Fixe como diretoria de trabalho, a diretoria onde está gravado o ficheiro, importe-o e apresente os dados importados sob a forma de tabela.

b) Quantos alunos estão inscritos a Matemática Computacional II?

c) Substitua na tabela os valores ausentes por 0.

d) Defina um molde que permita identificar os alunos que faltaram a todas os momentos de avaliação.

e) Use o molde definido anteriormente para obter o número de alunos que faltaram a todas os momentos de avaliação.

f) Construa uma lista, `trabalhos`, que contenha apenas as classificações dos trabalhos (sem cabeçalho).

g) A classificação final dos trabalhos é a média das três melhores classificações.

i. Defina uma função que, dada uma lista de 4 números, devolva a média dos 3 maiores.

ii. Obtenha uma lista com as classificações finais dos trabalhos.

iii. Acrescente à lista `trabalhos` uma coluna com a classificação final dos trabalhos.

⇛ Alguns comandos

`Directory` `Export` `FileNames` `FilePrint` `FilePrint` `Get` `Import` `Put` `PutAppend` `ReadList` `Save` `SetDirectory`

⇛ Links

http://reference.wolfram.com/language/tutorial/ImportingAndExportingData
http://reference.wolfram.com/language/tutorial/ImportingAndExportingFiles
http://reference.wolfram.com/language/guide/StringPatterns

13. Exercícios suplementares

Exercício 13.1® Pretende-se encontrar o menor número natural tal que a soma dos seus seis menores múltiplos distintos, incluindo o próprio número, seja um número formado por algarismos todos iguais.

a) Designe por `lista1` a lista contendo como sublistas os seis menores múltiplos distintos dos números naturais entre 20 e 50.

b) Construa a `lista2` contendo a soma dos elementos que constituem cada sublista da `lista1`.

c) Construa a `lista3` contendo como sublistas os algarismos que constituem cada elemento da `lista2`.

d) Defina um molde onde encaixem apenas listas formadas por um ou mais elementos iguais.

e) Verifique que existe(m) elemento(s) na `lista3` que encaixa(m) no molde definido na alínea anterior e determine a posição em que esse(s) elemento(s) ocorre(m).

f) Indique o inteiro a que faz referência a introdução desta pergunta.

Output esperado: 37

Exercício 13.2® Designa-se por **número sublime** um número natural cuja soma dos divisores é um número perfeito e tal que o número dos seus divisores é também um número perfeito. O número natural 12 é um número sublime.

a) Construa uma função-predicado `sublimeQ` que permita identificar números sublimes. Otimize a função de modo a que, para cada valor do argumento, seja calculada apenas uma vez a lista dos divisores desse número.

b) Verifique que 12 é o único número sublime inferior a 10000.

c) O ficheiro `sublime.txt` contém um número astronómico. Introduza esse número na sua sessão do Mathematica com a designação de astrosublime. Use o predicado da alínea a) para confirmar que se trata de um número sublime.

Exercício 13.3® Considere o ficheiro de imagem `circunferencias.png`.

a) Importe a imagem contida nesse ficheiro para a sessão do Mathematica.

b) A figura é simétrica relativamente ao eixo das ordenadas, as duas circunferências têm raios 1 e 1/3 e o segmento de reta [A B] corresponde a um diâmetro da circunferência de menor raio. Determine as coordenadas dos pontos A e B.

c) Construa uma figura contendo as duas circunferências representadas na figura.

d) Calcule a área da região sombreada na figura.

Output esperado: $\frac{2\sqrt{2}}{9} - \frac{\pi}{18} + \sec^{-1}(3)$

Exercício 13.4⊛ Os **números de Lucas** L_n são definidos pela seguinte relação:

$$L_0 = 2$$
$$L_1 = 1$$
$$L_n = L_{n-1} + L_{n-2};\ n \geq 2$$

a) Defina uma função recursiva, `lucas`, que permita obter o termo n da sucessão L_n (deve usar programação dinâmica).

b) Obtenha a lista, L50, dos primeiros 50 números de Lucas. Confirme o resultado obtido, recorrendo à função `LucasL` do Mathematica.

c) Selecione da lista anterior, os números ímpares maiores que 100.

d) Defina a função $f(x) = \Phi^x + (1 - \Phi)^x$, onde Φ é o número de ouro (`GoldenRatio`).

e) Ilustre, usando a lista L50, as seguintes propriedades conhecidas dos números de Lucas:

i. L_n-L_{n-4} é múltiplo de 5 para $n \geq 4$;

ii. $L_n = f(n)$;

iii. se L_n é um número primo, então n ou é 0, ou é um primo ou é uma potência de 2;

f) Mostre que $\left(\frac{L_n}{Ln-1}\right)_n$ é uma sucessão convergente para o número de ouro (sugestão: use a propriedade e)ii.).

Exercício 13.5⊛

a) Descreva a seguinte função e ilustre a sua aplicação.

```
ordena[x_Integer?NonNegative] := FromDigits[Sort[IntegerDigits[x]]]
```

b) Escreva uma função, `reverso`, que, dado um número natural, devolva o número natural que se obtém deste, invertendo a ordem dos seus dígitos.

Output esperado: `reverso`[1543] → 3451

c) Dado um número natural k, considere a sucessão $(a_n^k)_n$ definida do seguinte modo:

- o primeiro termo é k, isto é, $a_1^k = k$;
- o termo de ordem $n > 1$, a_n^k, obtém-se ordenando os dígitos da soma do termo anterior com o seu reverso, isto é, ordenando os dígitos de $a_{n-1}^k + \texttt{reverso}[a_{n-1}^k]$.

i. Escreva uma função que permita obter o termo de ordem n da sucessão a_n^k.

Output esperado: **241** → 241+142 = **338**→ 338+833=1171→**1117**

primeiros termos da sucessão a_n^{241}: 241, 338, 1117

ii. Construa as listas dos primeiros 20 termos das sucessões a_n^1 e a_n^3.

Exercício 13.6⊛

a) Defina uma função, `rep`, que aceite como argumento uma lista formada pela repetição de um único elemento e devolva uma lista com dois elementos, contendo na primeira posição o elemento repetido da lista que é dada como argumento e na segunda posição o número de repetições desse elemento.

Output esperado: `rep`[x,x,x] → {x,3}

b) Usando o comando `Split` e a função `rep` defina a função `compactar` que aceita como argumento uma lista e devolve uma nova lista formada por sublistas contendo os elementos da lista original na primeira posição e o número de vezes que esse elemento aparece repetido de forma consecutiva.

Output esperado: `compactar`[x,y,y,y,x,x,z,y,y,z,z,z,z] → {{x,1},{y,3},{x,2},{z,1},{y,2},{z,4}}

c) Use a função `compactar` para obter um resultado análogo ao da função `Tally`.

d) Descreva e ilustre a seguinte função `nome`; sugira um nome alternativo para esta função.

```
regra={x___,{y_,i_}, {y_,j_},z___}->{x, {y,i+j},z};
nome[lista_List] :=  Map[{#,1}&,lista]//.regra
```

e) Designe por `lbinaria` a lista contendo os elementos do ficheiro `binario.txt`. Aplique a função `compactar` a esta lista.

f) Defina a função `descompactar` que é a inversa esquerda da função `compactar`, isto é, tal que `descompactar[compactar[lista]]`→`lista`.

Resolução de alguns exercícios

Apresenta-se, de seguida, a versão pdf de um notebook,

criado com a versão 12 do Mathematica,

com propostas de resolução dos exercícios assinalados com ⊛

O notebook pode ser obtido aqui

matemática computacional ii

mif@math.uminho.pt :: fmiranda@math.uminho.pt

2020

1. Números e Expressões

Exercício 1.6

In[•]:=
```
pid50 = N[π, 51];
```

In[•]:=
```
FractionalPart[pid50]
```

Out[•]=
```
0.141592653589793238462643383279502884197169399375 11
```

In[•]:=
```
pid50 - Round[pid50]
```

Out[•]=
```
0.141592653589793238462643383279502884197169399375 11
```

In[•]:=
```
pid50 - Floor[pid50]
```

Out[•]=
```
0.141592653589793238462643383279502884197169399375 11
```

Exercício 1.8

In[•]:=
```
Table[Round[Pi^n], {n, 1, 10}]
```

Out[•]=
```
{3, 10, 31, 97, 306, 961, 3020, 9489, 29 809, 93 648}
```

Exercício 1.9

In[•]:= Table$\left[\left(1+\frac{1}{10^{i}}\right)^{10^{i}}, \{i, 1, 6\}\right]$ // N

Out[•]=
```
{2.59374, 2.70481, 2.71692, 2.71815, 2.71827, 2.71828}
```

---

2. Atribuições e Funções

Exercício 2.5

1º Processo

In[•]:=
```
A = x^2 y - 4 ⅇ^x y^3;
A /. {x → 0, y → 0}
```

Out[•]=
```
0
```

In[∘]:= `A /. {x → -1, y → 1}`

Out[∘]= $1-\frac{4}{e}$

2º processo

In[∘]:=
```
f[x_, y_] = x^2 y - 4 e^x y^3;
f[0, 0]
```

Out[∘]= 0

In[∘]:= `f[-1, 1]`

Out[∘]= $1-\frac{4}{e}$

Exercício 2.6

1º Processo

In[∘]:= $\text{Table}\left[\text{Sin}[x],\left\{x,\left\{0,\frac{\pi}{6},\frac{\pi}{4},\frac{\pi}{3},\frac{\pi}{2}\right\}\right\}\right]$

Out[∘]= $\left\{0,\frac{1}{2},\frac{1}{\sqrt{2}},\frac{\sqrt{3}}{2},1\right\}$

2º Processo

In[∘]:= $\text{Sin}[x]\ /.\ x\ \text{->}\left\{0,\frac{\pi}{6},\frac{\pi}{4},\frac{\pi}{3},\frac{\pi}{2}\right\}$

Out[∘]= $\left\{0,\frac{1}{2},\frac{1}{\sqrt{2}},\frac{\sqrt{3}}{2},1\right\}$

Exercício 2.11

In[∘]:= `g[x_] = Piecewise[{{x + 1, x > 2}, {x - 1, x < 2}}, 5]`

Out[∘]= $\begin{cases} 1+x & x>2 \\ -1+x & x<2 \\ 5 & \text{True} \end{cases}$

In[∘]:= `f[x_] = Piecewise[{{g[x], x ≤ 5}}, g[x]^2]`

Out[∘]= $\begin{cases} \begin{cases} 1+x & x>2 \\ -1+x & x<2 \\ 5 & \text{True} \end{cases} & x\le 5 \\ \left(\begin{cases} 1+x & x>2 \\ -1+x & x<2 \\ 5 & \text{True} \end{cases}\right)^2 & \text{True} \end{cases}$

In[∘]:=
```
PiecewiseExpand[%]
```
Out[∘]=
$$\begin{cases} 5 & x == 2 \\ -1 + x & x < 2 \\ 1 + x & 2 < x \le 5 \\ (1 + x)^2 & \text{True} \end{cases}$$

3. Listas

Exercício 3.5

In[∘]:=
```
lista1 = RandomInteger[{-10, 10}, 50]
```
Out[∘]=
```
{-6, 0, 10, 2, 3, 8, 2, -6, -3, 1, -1, 2, 4, 1, -9, 9, -2, 0, -2, 0, 6, -10, 8, 5, 4,
 9, 0, 10, 2, 1, -8, 2, -1, 8, 2, -6, 0, -6, 7, 2, 7, 4, 9, 5, 6, 3, 1, 9, -2, 10}
```

a)

In[∘]:=
```
Position[lista1, 0] // Length
```
Out[∘]= 5

ou

In[∘]:=
```
Count[lista1, 0]
```
Out[∘]= 5

b)

In[∘]:=
```
lista2 = Union[lista1]
```
Out[∘]=
```
{-10, -9, -8, -6, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Note-se que o Union altera a ordem dos elementos; alternativa:

In[∘]:=
```
DeleteDuplicates[lista1]
```
Out[∘]=
```
{-6, 0, 10, 2, 3, 8, -3, 1, -1, 4, -9, 9, -2, 6, -10, 5, -8, 7}
```

c)

In[∘]:=
```
Select[lista2, EvenQ]
```
Out[∘]=
```
{-10, -8, -6, -2, 0, 2, 4, 6, 8, 10}
```

d)

In[∘]:=
```
lista2[[2 ;; ;; 2]]
```
Out[∘]=
```
{-9, -6, -2, 0, 2, 4, 6, 8, 10}
```

e)

```
In[∘]:= lista1 = Prepend[Append[lista1, 12], -12]
Out[∘]= {-12, -6, 0, 10, 2, 3, 8, 2, -6, -3, 1, -1, 2, 4, 1, -9, 9, -2, 0, -2, 0, 6, -10, 8, 5,
 4, 9, 0, 10, 2, 1, -8, 2, -1, 8, 2, -6, 0, -6, 7, 2, 7, 4, 9, 5, 6, 3, 1, 9, -2, 10, 12}
```

Exercício 3.6

```
In[∘]:= l = Mod[Prime[Range[20]], 5]
Out[∘]= {2, 3, 0, 2, 1, 3, 2, 4, 3, 4, 1, 2, 1, 3, 2, 3, 4, 1, 2, 1}

In[∘]:= Sort[l]
Out[∘]= {0, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4}

In[∘]:= Tally[l] // Sort
Out[∘]= {{0, 1}, {1, 5}, {2, 6}, {3, 5}, {4, 3}}
```

Exercício 3.7

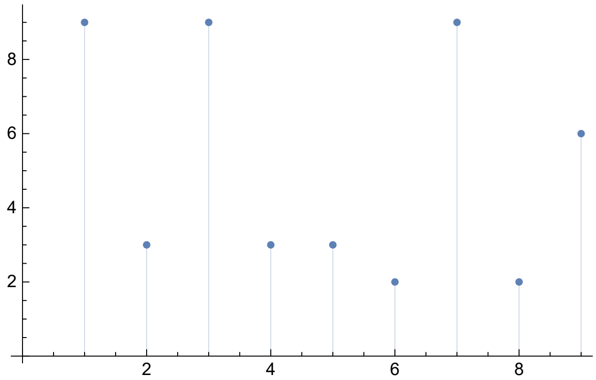
```
In[∘]:= listaprimos = Prime[Range[PrimePi[100]]]
Out[∘]= {2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31,
 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}
```

Esta lista é constituída por todos os primos menores que 100.

```
In[∘]:= IntegerDigits[listaprimos]
Out[∘]= {{2}, {3}, {5}, {7}, {1, 1}, {1, 3}, {1, 7}, {1, 9}, {2, 3},
 {2, 9}, {3, 1}, {3, 7}, {4, 1}, {4, 3}, {4, 7}, {5, 3}, {5, 9},
 {6, 1}, {6, 7}, {7, 1}, {7, 3}, {7, 9}, {8, 3}, {8, 9}, {9, 7}}

In[∘]:= l = Flatten[%]
Out[∘]= {2, 3, 5, 7, 1, 1, 1, 3, 1, 7, 1, 9, 2, 3, 2, 9, 3, 1, 3, 7, 4, 1,
 4, 3, 4, 7, 5, 3, 5, 9, 6, 1, 6, 7, 7, 1, 7, 3, 7, 9, 8, 3, 8, 9, 9, 7}

In[∘]:= ListPlot[Tally[l], Filling → Axis]
Out[∘]=
```



Exercício 3.8

In[∘]:=
```
l = RandomPrime[{10, 99}, 30]
```
Out[∘]=
```
{41, 17, 59, 43, 61, 43, 19, 19, 37, 89, 29, 11, 67, 43,
 67, 17, 59, 53, 79, 71, 17, 79, 53, 79, 59, 67, 41, 19, 17, 13}
```

a)

In[∘]:=
```
Length[Union[l]]
```
Out[∘]=
```
15
```

b)

In[∘]:=
```
Mean[l]
```
Out[∘]= $\frac{228}{5}$

c)

In[∘]:=
```
Commonest[l]
```
Out[∘]=
```
{17}
```

d)

In[10]:=
```
numerosprimos2algarismos = PrimePi[99] - PrimePi[9]
```
Out[10]=
```
21
```

In[9]:=
```
primos2algarismos =
 Complement[Prime[Range[PrimePi[99]]], Prime[Range[PrimePi[9]]]]
```
Out[9]=
```
{11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}
```

e) Os números primos que não estão na lista são:

In[∘]:=
```
Complement[primos2algarismos, l]
```
Out[∘]=
```
{23, 31, 47, 73, 83, 97}
```

Exercício 3.15

In[∘]:=
```
somadigitos[n_] = IntegerDigits[n] // Total;
```

In[∘]:=
```
somadigitos[123]
```
Out[∘]=
```
6
```

Exercício 3.16

In[∘]:=
```
Clear@somadivisores
```

In[∘]:=
```
somadivisores[n_] := Most[Divisors[n]] // Total
```

In[∘]:= `somadivisores[150]`

Out[∘]= 222

In[∘]:= `Divisors[150]`

Out[∘]= {1, 2, 3, 5, 6, 10, 15, 25, 30, 50, 75, 150}

Exercício 3.17

In[∘]:= `divisorescomuns[a_, b_] := Intersection[Divisors[a], Divisors[b]]`

In[∘]:= `divisorescomuns[24, 32]`

Out[∘]= {1, 2, 4, 8}

In[∘]:= `maxdivisorescomuns[a_, b_] := Max[Intersection[Divisors[a], Divisors[b]]]`

In[∘]:= `maxdivisorescomuns[24, 32]`

Out[∘]= 8

In[∘]:= `GCD[24, 32]`

Out[∘]= 8

---

4. Matrizes

Exercício 4.2

In[∘]:= `A = {{1, -1, 2}, {3, 2, 1}, {1, -5, 4}};`

In[∘]:= `A1 = A; A1[[3, 2]] = 5; MatrixForm[A1]`

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & -1 & 2 \\ 3 & 2 & 1 \\ 1 & 5 & 4 \end{pmatrix}$$

In[∘]:= `(A2 = UpperTriangularize[A]) // MatrixForm`

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & -1 & 2 \\ 0 & 2 & 1 \\ 0 & 0 & 4 \end{pmatrix}$$

In[∘]:= `(A3 = Insert[Transpose[A], {1, 2, 3}, 4] // Transpose) // MatrixForm`

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & -1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & -5 & 4 & 3 \end{pmatrix}$$

ou

In[∘]:=
```
(A3 = Append[Transpose[A], {1, 2, 3}] // Transpose) // MatrixForm
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & -1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & -5 & 4 & 3 \end{pmatrix}$$

In[∘]:=
```
(A4 = Append[A, {1, 2, 3}]) // MatrixForm
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & -1 & 2 \\ 3 & 2 & 1 \\ 1 & -5 & 4 \\ 1 & 2 & 3 \end{pmatrix}$$

In[∘]:=
```
(A5 = Prepend[Insert[Transpose[A], {0, 0, 0}, 1] // Transpose, {2, 1, -1, 2}]) //
 MatrixForm
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} 2 & 1 & -1 & 2 \\ 0 & 1 & -1 & 2 \\ 0 & 3 & 2 & 1 \\ 0 & 1 & -5 & 4 \end{pmatrix}$$

In[∘]:=
```
(A6 = A[[1 ;; 2, 1 ;; 2]]) // MatrixForm
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & -1 \\ 3 & 2 \end{pmatrix}$$

Exercício 4.6

a)

In[∘]:=
```
mA = {{2, 0}, {0, 3}};
```

In[∘]:=
```
(permutamA = Array[a, {2, 2}]) // MatrixForm
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} a[1,1] & a[1,2] \\ a[2,1] & a[2,2] \end{pmatrix}$$

In[∘]:=
```
solucao = Solve[mA.permutamA == permutamA.mA, Flatten[permutamA]]
```
Out[∘]= {{a[1, 2] → 0, a[2, 1] → 0}}

As matrizes que permutam com a matriz A são as matrizes da seguinte forma:

In[∘]:=
```
(permutamA = permutamA /. solucao[[1]]) // MatrixForm
```
Out[∘]//MatrixForm=

$$\begin{pmatrix} a[1,1] & 0 \\ 0 & a[2,2] \end{pmatrix}$$

Verificação:

In[∘]:=
```
mA.permutamA == permutamA.mA
```
Out[∘]= True

Exercício 4.15

a)

In[∘]:=
```
Vandermonde[l_] := Table[l^i, {i, 0, Length[l] - 1}];
```

In[∘]:=
```
(V1 = Vandermonde[Table[a_i, {i, 1, 4}]]) // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ a_1 & a_2 & a_3 & a_4 \\ a_1^2 & a_2^2 & a_3^2 & a_4^2 \\ a_1^3 & a_2^3 & a_3^3 & a_4^3 \end{pmatrix}$$

In[∘]:=
```
(V2 = Vandermonde[{1, -2, 3, -1, 1}]) // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & -2 & 3 & -1 & 1 \\ 1 & 4 & 9 & 1 & 1 \\ 1 & -8 & 27 & -1 & 1 \\ 1 & 16 & 81 & 1 & 1 \end{pmatrix}$$

b)

In[∘]:=
```
Det[V1] // FullSimplify
```

Out[∘]= $(a_1 - a_2)\ (a_1 - a_3)\ (a_2 - a_3)\ (a_1 - a_4)\ (a_2 - a_4)\ (a_3 - a_4)$

c)

In[∘]:=
```
Position[V2, 1] // Length
```

Out[∘]= 15

ou

In[∘]:=
```
Count[Flatten@V2, 1]
```

Out[∘]= 15

d)

In[∘]:=
```
(V3 = Delete[V2, {{4}, {5}}]) // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & -2 & 3 & -1 & 1 \\ 1 & 4 & 9 & 1 & 1 \end{pmatrix}$$

In[∘]:=
```
(V3 = Transpose[V3] // Rest // Transpose) // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ -2 & 3 & -1 & 1 \\ 4 & 9 & 1 & 1 \end{pmatrix}$$

Exercício 4.16

In[∘]:=
```
(A = {{1, -1, 2}, {1, -1, -1}, {-3, 4, 1}}) // MatrixForm
(B = {{Cos[θ], -Sin[θ], 0}, {-Sin[θ], -Cos[θ], 0}, {0, 0, 1}}) // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & -1 & 2 \\ 1 & -1 & -1 \\ -3 & 4 & 1 \end{pmatrix}$$

Out[∘]//MatrixForm=

$$\begin{pmatrix} \mathrm{Cos}[\theta] & -\mathrm{Sin}[\theta] & 0 \\ -\mathrm{Sin}[\theta] & -\mathrm{Cos}[\theta] & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

a)

In[∘]:=
```
{Det@A, Det@B}
```

Out[∘]= $\{3, -\mathrm{Cos}[\theta]^2 - \mathrm{Sin}[\theta]^2\}$

b)

In[∘]:=
```
Rest[Rest[Transpose[A]] // Transpose] // Det
```

Out[∘]= 3

c)

Não existe uma função para calcular a adjunta de uma matriz. A função Minors pode ser usada, mas com alguma cautela e mais à frente.

Como $A^{-1} = \frac{1}{\det A} \mathrm{Adj}\, A$, facilmente se obtém a adjunta de uma matriz.

In[∘]:=
```
(AdjA = detA Inverse[A]) // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} \mathrm{detA} & 3\,\mathrm{detA} & \mathrm{detA} \\ \frac{2\,\mathrm{detA}}{3} & \frac{7\,\mathrm{detA}}{3} & \mathrm{detA} \\ \frac{\mathrm{detA}}{3} & -\frac{\mathrm{detA}}{3} & 0 \end{pmatrix}$$

In[∘]:=
```
(AdjB = detB Inverse[B] // Simplify) // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} \mathrm{detB}\,\mathrm{Cos}[\theta] & -\mathrm{detB}\,\mathrm{Sin}[\theta] & 0 \\ -\mathrm{detB}\,\mathrm{Sin}[\theta] & -\mathrm{detB}\,\mathrm{Cos}[\theta] & 0 \\ 0 & 0 & \mathrm{detB} \end{pmatrix}$$

d)

In[∘]:=
```
Solve[B.{1, 1, 1} == {-1, -1, 1} && 0 ≤ θ ≤ 2 π, θ]
```

Out[∘]= $\{\{\theta \to \frac{\pi}{2}\}\}$

Exercício 4.17

a)

In[∘]:=
```
f[x_, y_, z_] = {x, x - y, x - z};
```

In[∘]:=
```
(Mf = {f[1, 0, 0], f[0, 0, 1], f[0, 0, 1]} // Transpose) // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & -1 & -1 \end{pmatrix}$$

In[∘]:=
```
nucleo = NullSpace[Mf]
```

Out[∘]= {{0, -1, 1}}

In[∘]:=
```
dimensao = Length[%]
```

Out[∘]= 1

In[∘]:=
```
dimensao == 0
```

Out[∘]= False

In[∘]:=
```
caracteristica = MatrixRank[Mf]
```

Out[∘]= 2

In[∘]:=
```
caracteristica == First[Dimensions[Mf]]
```

Out[∘]= False

f não é injetiva nem sobrejetiva

b)

In[∘]:=
```
(Mh = {{1, -1, 2}, {-2, 5, 3}, {-7, 16, 7}, {-3, 6, -1}} // Transpose) // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & -2 & -7 & -3 \\ -1 & 5 & 16 & 6 \\ 2 & 3 & 7 & -1 \end{pmatrix}$$

In[∘]:=
```
nucleo = NullSpace[Mh]
```

Out[∘]= {{1, -3, 1, 0}}

In[∘]:=
```
dimensao = Length[%]
```

Out[∘]= 1

In[∘]:=
```
dimensao == 0
```

Out[∘]= False

In[∘]:=
```
caracteristica = MatrixRank[Mh]
```

Out[∘]= 3

In[∘]:=
```
caracteristica == First[Dimensions[Mh]]
```

Out[∘]= True

h não é injetiva, mas é sobrejetiva

Exercício 4.20

```
In[]:= Clear[x]

In[]:= A = {{0, 1}, {1, 0}};

In[]:= PolinomioC = CharacteristicPolynomial[A, x]
Out[]= -1 + x^2

In[]:= ValoresProprios = x /. Solve[PolinomioC == 0, x]
Out[]= {-1, 1}
```

ou

```
In[]:= Eigenvalues[A]
Out[]= {-1, 1}

In[]:= VetoresProprios = Eigenvectors[A]
Out[]= {{-1, 1}, {1, 1}}

In[]:= DiagonalizableMatrixQ[A]
Out[]= True
```

ou

```
In[]:= MatrixRank[VetoresProprios] == Length[A]
Out[]= True

In[]:= B = {{0, 1}, {-1, 0}};
       PolinomioC = CharacteristicPolynomial[B, x]
       ValoresProprios = Eigenvalues[B]
       VetoresProprios = Eigenvectors[B]
       DiagonalizableMatrixQ[B]
Out[]= 1 + x^2
Out[]= {i, -i}
Out[]= {{-i, 1}, {i, 1}}
Out[]= True

In[]:= CC = {{2, 1, 0}, {0, 2, 1}, {0, 0, 2}};
       PolinomioC = CharacteristicPolynomial[CC, x]
       ValoresProprios = Eigenvalues[CC]
       VetoresProprios = Eigenvectors[CC]
       DiagonalizableMatrixQ[CC]
Out[]= 8 - 12 x + 6 x^2 - x^3
Out[]= {2, 2, 2}
Out[]= {{1, 0, 0}, {0, 0, 0}, {0, 0, 0}}
Out[]= False
```

```
In[•]:= DD = {{2, 0, 1}, {0, 3, 0}, {1, 0, 2}};;
        PolinomioC = CharacteristicPolynomial[DD, x]
        ValoresProprios = Eigenvalues[DD]
        VetoresProprios = Eigenvectors[DD]
        DiagonalizableMatrixQ[DD]
```

Out[•]= $9 - 15\,x + 7\,x^2 - x^3$

```
Out[•]= {3, 3, 1}

Out[•]= {{1, 0, 1}, {0, 1, 0}, {-1, 0, 1}}

Out[•]= True
```

5. Operadores e funções booleanas

Exercício 5.5

```
In[•]:= primosEntreSiQ[x_, y_] := GCD[x, y] == 1

In[•]:= primosEntreSiQ[12, 3]

Out[•]= False

In[•]:= primosEntreSiQ[12, 5]

Out[•]= True
```

ou

```
In[•]:= primosEntreSi2Q[x_, y_] := Intersection[Divisors[x], Divisors[y]] == {1}

In[•]:= primosEntreSi2Q[12, 3]

Out[•]= False

In[•]:= primosEntreSi2Q[12, 5]

Out[•]= True
```

Exercício 5.8

```
In[•]:= palindromoQ[x_] := Characters[x] == Reverse[Characters[x]]

In[•]:= palindromoQ["osso"]

Out[•]= True

In[•]:= palindromoQ["amor a roma"]

Out[•]= True

In[•]:= palindromoQ["A base do teto desaba"]
```

Out[∘]= False

Se não tivermos em linha de conta os espaços:

```
In[∘]:= semespaçosQ[x_] := x ≠ " ";
```

```
In[∘]:= palindromo2Q[x_] :=
         (frase = Select[Characters[ToUpperCase@x], semespaçosQ]) == Reverse[frase]
```

```
In[∘]:= palindromo2Q["A base do teto desaba"]
```

Out[∘]= True

Outra solução:

```
In[∘]:= palindromo3Q[x_] := (frase = Characters[StringSplit[ToLowerCase@x]] // Flatten;
          frase == Reverse[frase])
```

```
In[∘]:= palindromo3Q["A base do teto desaba"]
```

Out[∘]= True

Exercício 5.12

```
In[∘]:= Clear[p, q]
```

```
In[∘]:= TableForm[BooleanTable[{p, q, (p ∨ q) ∧ ¬ (p ∧ q)}, {p, q}],
         TableHeadings → {None, {"p", "q", "(p∨q)∧¬(p∧q)"}}, TableAlignments → Center]
```

Out[∘]//TableForm=

p	q	(p∨q)∧¬(p∧q)
True	True	False
True	False	True
False	True	True
False	False	False

Exercício 5.14

a)

```
In[∘]:= TautologyQ[(p ⇒ q) ∧ (¬ q ⇒ ¬ p)]
```

Out[∘]= False

```
In[∘]:= TautologyQ[¬ ((p ⇒ q) ∧ (¬ q ⇒ ¬ p))]
```

Out[∘]= False

Não é uma tautologia nem uma contradição.
Verificação:

```
In[∘]:= TableForm[BooleanTable[{p, q, (p ⇒ q) ∧ (¬ q ⇒ ¬ p)}],
         TableHeadings → {None, {"p", "q", "(p⇒q)∧(¬q⇒¬p)"}}, TableAlignments → Center]
```

Out[∘]//TableForm=

p	q	(p⇒q)∧(¬q⇒¬p)
True	True	True
True	False	False
False	True	True
False	False	True

6. Gráficos

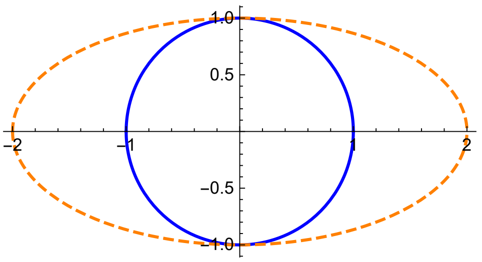
Exercício 6.4

Usando as três formas do Exercício 3:

In[∘]:=
```
f[x_] = Sqrt[1 - x^2]; g[x_] = Sqrt[1 - x^2 / 4];
```

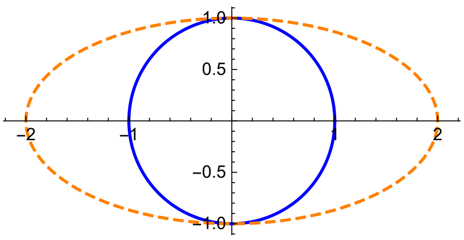
In[∘]:=
```
g1 = Plot[{-f[x], f[x]}, {x, -1, 1}, PlotStyle → Blue, AspectRatio → Automatic];
g2 = Plot[{-g[x], g[x]}, {x, -2, 2},
   PlotStyle → {{Orange, Dashed}, {Orange, Dashed}}, AspectRatio → Automatic];
Show[g1, g2, PlotRange → All]
```

Out[∘]=



In[∘]:=
```
ParametricPlot[{{Cos[t], Sin[t]}, {2 Cos[t], Sin[t]}},
 {t, 0, 2 π}, PlotStyle → {{Blue}, {Orange, Dashed}}]
```

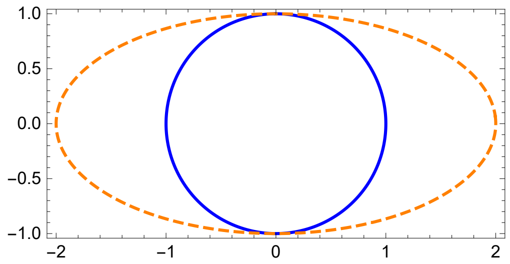
Out[∘]=



In[∘]:=
```
fig1 = RegionPlot[ImplicitRegion[x^2 + y^2 == 1, {x, y}],
   AspectRatio → Automatic, BoundaryStyle → Blue];
fig2 = RegionPlot[ImplicitRegion[x^2 / 4 + y^2 == 1, {x, y}],
   AspectRatio → Automatic, BoundaryStyle → {Orange, Dashed}];
```

In[∘]:=
```
Show[fig1, fig2, PlotRange → All]
```

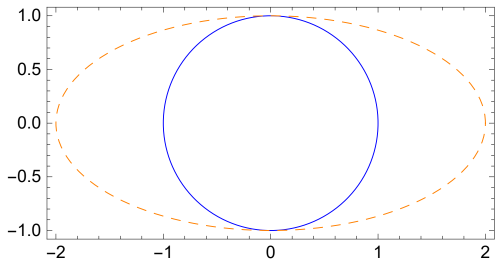
Out[∘]=



Naturalmente que o exercício também poderia ser resolvido, usando a primitiva Circle.

In[∘]:=
```
Graphics[{{Blue, Circle[{0, 0}, 1]},
  {Orange, Dashing[0.02], Circle[{0, 0}, {2, 1}]}}, Frame → True]
```

Out[∘]=



Exercício 6.6

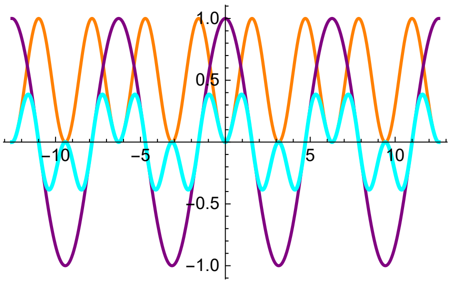
In[∘]:=
```
Graphics[{{Pink, Rectangle[{0, 0}, {1, 1}]},
  {Green, Opacity[0.8], EdgeForm[Dashed], Rectangle[{0.5, 0.5}, {1.5, 1}]},
  {Thick, Yellow, Dashing[0.1], Circle[{0.5, 0.5}, 0.5]},
  {Purple, Polygon[{{1, 0}, {1.5, 0}, {1.5, 0.5}}]},
  {Dashing[Small], Cyan, Arrow[{{1, 0.5}, {1.25, 0.25}}]},
  {Cyan, PointSize[0.03], Point[{1, 0.5}]}},
 PlotLabel → Style["Primitivas e Diretivas gráficas", Bold, Red, 14]]
```

Out[∘]=

Exercício 6.10

In[∘]:=
```
f[x_] = Sin[x]^2; g[x_] = Cos[x]; h[x_] = Sin[x]^2 Cos[x];
Plot[{f[x], g[x], h[x]}, {x, -4 Pi, 4 Pi},
 PlotStyle → {{Orange}, {Purple}, {Cyan, AbsoluteThickness[2]}}]
```

Out[∘]=



7. Funções reais

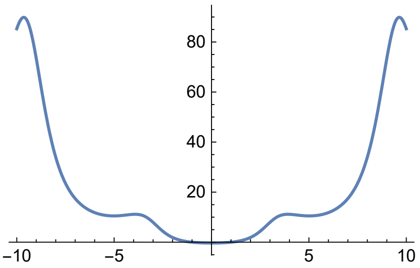
Exercício 7.4

In[∘]:=
```
Limit[(x^2 - 1) / (2 + Cos[x]), x → Infinity, Direction → -1]
```

Out[∘]= ∞

In[∘]:=
```
Plot[(x^2 - 1) / (2 + Cos[x]), {x, -10, 10}]
```

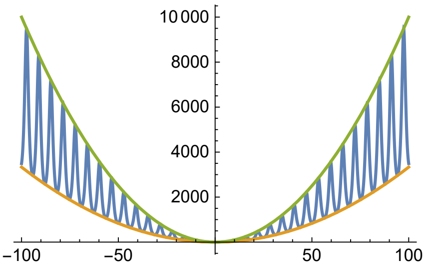
Out[∘]=



Note-se que $\frac{x^2-1}{2+1} \leq \frac{x^2-1}{2+\text{Cos}[x]} \leq \frac{x^2-1}{2-1}$

In[∘]:=
```
Plot[{(x^2 - 1) / (2 + Cos[x]), (x^2 - 1) / 3, x^2 - 1}, {x, -100, 100}]
```

Out[∘]=



Exercício 7.6

Algumas observações sobre a sintaxe.

```
In[•]:= Remove[f]

In[•]:= f'[x] // FullForm

Out[•]//FullForm= Derivative[1][f][x]

In[•]:= D[f[x], x] // FullForm

Out[•]//FullForm= Derivative[1][f][x]

In[•]:= f''[x] // FullForm

Out[•]//FullForm= Derivative[2][f][x]

In[•]:= D[f[x], {x, 2}] // FullForm

Out[•]//FullForm= Derivative[2][f][x]

In[•]:= D[f[x^2], x]
```

Out[•]= $2\,x\,f'\left[x^2\right]$

```
In[•]:= f[x_] = x Cos[x^2] Exp[-x];
```

Diversas sintaxes...

```
In[•]:= f'[x]
```

Out[•]= $e^{-x}\,\mathrm{Cos}\left[x^2\right] - e^{-x}\,x\,\mathrm{Cos}\left[x^2\right] - 2\,e^{-x}\,x^2\,\mathrm{Sin}\left[x^2\right]$

```
In[•]:= D[f[x], x]
```

Out[•]= $e^{-x}\,\mathrm{Cos}\left[x^2\right] - e^{-x}\,x\,\mathrm{Cos}\left[x^2\right] - 2\,e^{-x}\,x^2\,\mathrm{Sin}\left[x^2\right]$

```
In[•]:= Derivative[1][f][x]
```

Out[•]= $e^{-x}\,\mathrm{Cos}\left[x^2\right] - e^{-x}\,x\,\mathrm{Cos}\left[x^2\right] - 2\,e^{-x}\,x^2\,\mathrm{Sin}\left[x^2\right]$

```
In[•]:= % // Simplify
```

Out[•]= $e^{-x}\left(-\left((-1+x)\,\mathrm{Cos}\left[x^2\right]\right) - 2\,x^2\,\mathrm{Sin}\left[x^2\right]\right)$

```
In[•]:= f''[x] // Simplify
```

Out[•]= $e^{-x}\left(\left(-2+x-4\,x^3\right)\mathrm{Cos}\left[x^2\right] + 2\,x\,(-3+2\,x)\,\mathrm{Sin}\left[x^2\right]\right)$

In[∘]:= `D[f[x], {x, 8}]`

Out[∘]= $\left(-8\,e^{-x}+e^{-x}\,x\right)\,\mathrm{Cos}\left[x^2\right]+28\left(-6\,e^{-x}+e^{-x}\,x\right)\left(-4\,x^2\,\mathrm{Cos}\left[x^2\right]-2\,\mathrm{Sin}\left[x^2\right]\right)-16\,x\left(7\,e^{-x}-e^{-x}\,x\right)\,\mathrm{Sin}\left[x^2\right]+70\left(-4\,e^{-x}+e^{-x}\,x\right)\left(-12\,\mathrm{Cos}\left[x^2\right]+16\,x^4\,\mathrm{Cos}\left[x^2\right]+48\,x^2\,\mathrm{Sin}\left[x^2\right]\right)+56\left(5\,e^{-x}-e^{-x}\,x\right)\left(-12\,x\,\mathrm{Cos}\left[x^2\right]+8\,x^3\,\mathrm{Sin}\left[x^2\right]\right)+28\left(-2\,e^{-x}+e^{-x}\,x\right)\left(720\,x^2\,\mathrm{Cos}\left[x^2\right]-64\,x^6\,\mathrm{Cos}\left[x^2\right]+120\,\mathrm{Sin}\left[x^2\right]-480\,x^4\,\mathrm{Sin}\left[x^2\right]\right)+56\left(3\,e^{-x}-e^{-x}\,x\right)\left(160\,x^3\,\mathrm{Cos}\left[x^2\right]+120\,x\,\mathrm{Sin}\left[x^2\right]-32\,x^5\,\mathrm{Sin}\left[x^2\right]\right)+e^{-x}\,x\left(1680\,\mathrm{Cos}\left[x^2\right]-13\,440\,x^4\,\mathrm{Cos}\left[x^2\right]+256\,x^8\,\mathrm{Cos}\left[x^2\right]-13\,440\,x^2\,\mathrm{Sin}\left[x^2\right]+3584\,x^6\,\mathrm{Sin}\left[x^2\right]\right)+8\left(e^{-x}-e^{-x}\,x\right)\left(1680\,x\,\mathrm{Cos}\left[x^2\right]-1344\,x^5\,\mathrm{Cos}\left[x^2\right]-3360\,x^3\,\mathrm{Sin}\left[x^2\right]+128\,x^7\,\mathrm{Sin}\left[x^2\right]\right)$

In[∘]:= `Simplify[%]`

Out[∘]= $e^{-x}\left(\left(3352+10\,921\,x-52\,416\,x^2+46\,928\,x^3-13\,440\,x^4-23\,072\,x^5+14\,336\,x^6-1792\,x^7+256\,x^9\right)\mathrm{Cos}\left[x^2\right]-8\left(798-2919\,x+2518\,x^2+4340\,x^3-6664\,x^4+2352\,x^5-224\,x^6-576\,x^7+128\,x^8\right)\mathrm{Sin}\left[x^2\right]\right)$

Exercício 7.9

Na resolução de equações não lineares podem ser usadas as funções Solve ou Reduce. Estas funções tentam encontrar a solução exata das equações. Caso não seja possível, pode recorrer-se às funções NSolve e FindRoot para obter aproximações para a solução.

In[∘]:= `p[x_] = -28 + 102 x - 126 x^2 + 93 x^3 - 46 x^4 - 30 x^5 + 54 x^6 - 21 x^7 + 2 x^8;`

Função Solve

In[∘]:= `zeros = Solve[p[x] == 0, x]`

Out[∘]= $\left\{\{x\to -i\},\{x\to i\},\left\{x\to \frac{1}{2}\right\},\{x\to 1\},\{x\to 2\},\{x\to 7\},\left\{x\to -\sqrt{2}\right\},\left\{x\to \sqrt{2}\right\}\right\}$

Pode ser usado o comando ReplaceAll (/.) para substituir *x* pelas soluções,
obtendo – se uma lista com as soluções :

In[∘]:= `solucao = x /. zeros`

Out[∘]= $\left\{-i,\ i,\ \frac{1}{2},\ 1,\ 2,\ 7,\ -\sqrt{2},\ \sqrt{2}\right\}$

É ainda possível resolver a equação num determinado domínio .

In[∘]:= `Solve[p[x] == 0, x, Integers]`

Out[∘]= $\{\{x\to 1\},\{x\to 2\},\{x\to 7\}\}$

In[∘]:= `Solve[p[x] == 0, x, Reals]`

Out[∘]= $\left\{\left\{x\to \frac{1}{2}\right\},\{x\to 1\},\{x\to 2\},\{x\to 7\},\left\{x\to -\sqrt{2}\right\},\left\{x\to \sqrt{2}\right\}\right\}$

Função Reduce

In[∘]:= `zeros = Reduce[p[x] == 0, x]`

Out[∘]= $x == -i \,||\, x == i \,||\, x == \frac{1}{2} \,||\, x == 1 \,||\, x == 2 \,||\, x == 7 \,||\, x == -\sqrt{2} \,||\, x == \sqrt{2}$

As funções ToRules e ReplaceRepeated (//.) permitem listar as soluções :

In[∘]:=
```
solucao = x //. {ToRules[zeros]}
```

Out[∘]= $\left\{-i,\ i,\ \frac{1}{2},\ 1,\ 2,\ 7,\ -\sqrt{2},\ \sqrt{2}\right\}$

In[∘]:=
```
Reduce[p[x] == 0, x, Primes]
```

Out[∘]= $x == 2 \,||\, x == 7$

In[∘]:=
```
Reduce[p[x] == 0, x, Rationals]
```

Out[∘]= $x == \frac{1}{2} \,||\, x == 1 \,||\, x == 2 \,||\, x == 7$

Função FindRoot

A função FindRoot permite encontrar apenas uma aproximação para um zero próximo de um dado valor inicial.

In[∘]:=
```
FindRoot[p[x] == 0, {x, 0}]
```

Out[∘]= {x → 0.5}

In[∘]:=
```
FindRoot[p[x] == 0, {x, 1.5}]
```

Out[∘]= {x → 1.41421}

In[∘]:=
```
q[x_] = x^5 + x^4 - 4 x^3 + 2 x^2 - 3 x - 7
```

Out[∘]= $-7 - 3x + 2x^2 - 4x^3 + x^4 + x^5$

In[∘]:=
```
zeros = Solve[q[x] == 0, x]
```

Out[∘]= {{x → -2.74...}, {x → -0.881...}, {x → 1.80...},
{x → 0.415... - 1.20... i}, {x → 0.415... + 1.20... i}}

Neste caso, não é possível obter a solução exata; a função Solve apresenta uma aproximação.

In[∘]:=
```
% // N
```

Out[∘]= {{x → -2.74463}, {x → -0.880858}, {x → 1.79645},
{x → 0.41452 - 1.19996 i}, {x → 0.41452 + 1.19996 i}}

Pode também ser usada a função NSolve para obter aproximações para a solução.

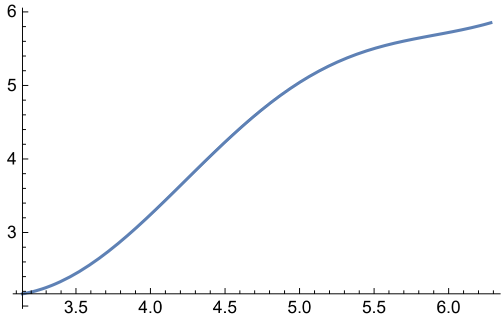
In[∘]:=
```
zeros = NSolve[q[x] == 0, x]
```

Out[∘]= {{x → -2.74463}, {x → -0.880858},
{x → 0.41452 - 1.19996 i}, {x → 0.41452 + 1.19996 i}, {x → 1.79645}}

Exercício 7.16

In[∘]:=
```
f[x_] = x + Sin[x] + Sin[x Pi / 2];
graff = Plot[f[x], {x, Pi, 2 Pi}]
```

Out[∘]=



Equação da reta tangente ao gráfico da função f no ponto x_0 : $y = f'(x_0)(x - x_0) + f(x_0)$

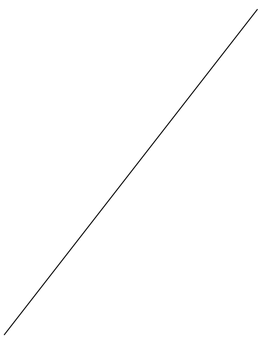
In[∘]:=
```
x0 = 5;
rtan[x_] = f'[x0] (x - x0) + f[x0]
```

Out[∘]= 6 + (-5 + x) (1 + Cos[5]) + Sin[5]

Para desenhar esta reta, basta escolher dois pontos da reta.

In[∘]:=
```
(reta = Line[{{0, rtan[0]}, {6, rtan[6]}}]) // Graphics
```

Out[∘]=

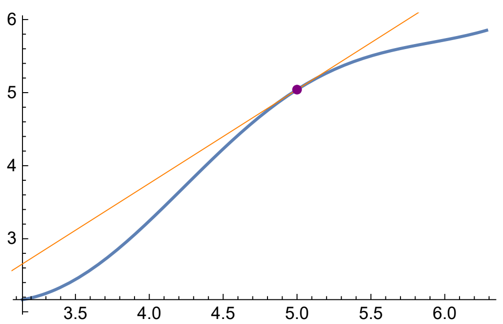


In[∘]:=
```
pontotangencia = {x0, rtan[x0]}
```

Out[∘]= {5, 6 + Sin[5]}

In[∘]:=
```
Show[graff,
 Epilog -> {{Orange, reta}, {Purple, PointSize[.02], Point[pontotangencia]}}]
```

Out[∘]=



Esta parte do exercício é apenas uma curiosidade.

In[∘]:=
```
InputForm[graff] // Short
```

Out[∘]//Short=
```
Graphics[{{{{}, {}, Annotation[{Directive[Opacity[1.],
  <<2>>], Line[<<1>>]}, <<1>>]}}, {}}, {<<25>>}]
```

In[∘]:=
```
Position[%, Line]
```

Out[∘]=
```
{{1, 1, 1, 3, 1, 2, 0}}
```

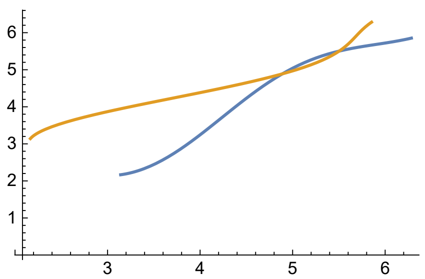
In[∘]:=
```
listacoordenadas = graff[[1, 1, 1, 3, 1, 2, 1]]
```

Out[∘]=
```
{{3.14159, 2.16622}, {3.14256, 2.16656}, {3.14352, 2.1669}, {3.14545, 2.16758},
 {3.1493, 2.16897}, {3.15701, 2.17185}, {3.17243, 2.17805}, {3.17339, 2.17846},
 {3.17435, 2.17887}, {3.17628, 2.17969}, {3.18013, 2.18137}, {3.18784, 2.18482},
 {3.20326, 2.19217}, {3.2043, 2.19269}, {3.20535, 2.19321}, {3.20744, 2.19426},
 {3.21161, 2.19639}, {3.21997, 2.20078}, {3.23668, 2.21005}, {3.27011, 2.23061},
 {3.27109, 2.23125}, {3.27206, 2.2319}, {3.27401, 2.23319}, {3.27791, 2.2358},
 {3.28572, 2.24112}, {3.30132, 2.25221}, {3.33253, 2.2761}, {3.33349, 2.27686},
 {3.33444, 2.27763}, {3.33636, 2.27918}, {3.34018, 2.2823}, {3.34783, 2.28864},
 {3.36313, 2.30172}, {3.39373, 2.32949}, {3.46011, 2.39696}, {3.52207, 2.46853},
 {3.58921, 2.55498}, {3.65513, 2.64825}, {3.71661, 2.74217}, {3.78329, 2.85086},
 {3.84553, 2.95804}, {3.90655, 3.06778}, {3.97275, 3.19125}, {4.03453, 3.30983},
 {4.10149, 3.44111}, {4.16723, 3.57186}, {4.22854, 3.69465}, {4.29503, 3.82789},
 {4.3571, 3.95152}, {4.42435, 4.08384}, {4.49038, 4.21126}, {4.55197, 4.32723},
 {4.61876, 4.44912}, {4.68111, 4.55875}, {4.74224, 4.66183}, {4.80856, 4.7683},
 {4.87044, 4.86227}, {4.93752, 4.95794}, {5.00336, 5.04539}, {5.06478, 5.12106},
 {5.13139, 5.19667}, {5.19356, 5.26124}, {5.25451, 5.31904}, {5.32065, 5.37582},
 {5.38236, 5.42351}, {5.44925, 5.46984}, {5.51172, 5.5085}, {5.57296, 5.54256},
 {5.63938, 5.57579}, {5.70138, 5.6039}, {5.76856, 5.63194}, {5.83452, 5.65778},
 {5.89605, 5.68107}, {5.96277, 5.70626}, {6.02505, 5.73043}, {6.08611, 5.75546},
 {6.15236, 5.78486}, {6.21418, 5.81511}, {6.21526, 5.81566}, {6.21633, 5.81622},
 {6.21849, 5.81734}, {6.2228, 5.81958}, {6.23143, 5.82412}, {6.24868, 5.83342},
 {6.24976, 5.83401}, {6.25084, 5.8346}, {6.25299, 5.83578}, {6.25731, 5.83817},
 {6.26593, 5.843}, {6.26701, 5.84361}, {6.26809, 5.84422}, {6.27025, 5.84544},
 {6.27456, 5.8479}, {6.27564, 5.84852}, {6.27672, 5.84914}, {6.27887, 5.85038},
 {6.27995, 5.85101}, {6.28103, 5.85163}, {6.28211, 5.85226}, {6.28319, 5.85288}}
```

In[∘]:=
```
listafinv = Reverse[listacoordenadas, 2];
```

In[∘]:=
```
ListPlot[{listacoordenadas, listafinv}, Joined → True]
```

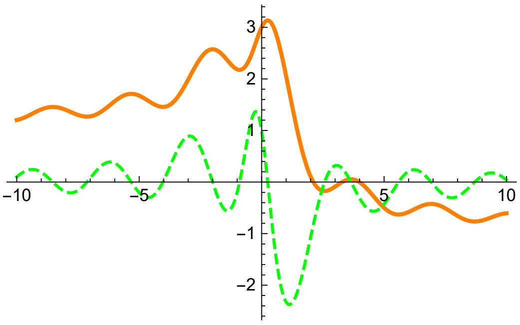
Out[∘]=



Exercício 7.17

In[∘]:=
```
Remove[f]
f[x_] = (Sin[2 x] - x + 3)/Sqrt[x^2 + 1];
```

In[∘]:=
```
graff = Plot[{f[x], f'[x]}, {x, -10, 10},
  PlotStyle -> {{Orange, Thickness[0.008]}, {Green, Dashed}}, PlotRange → All]
```

Out[∘]=


Menor zero

In[∘]:=
```
sol = x /. FindRoot[f'[x], {x, 0}]
```

Out[∘]= 0.249016

In[∘]:=
```
pontos = {{sol, f[sol]}};
```

Assíntotas

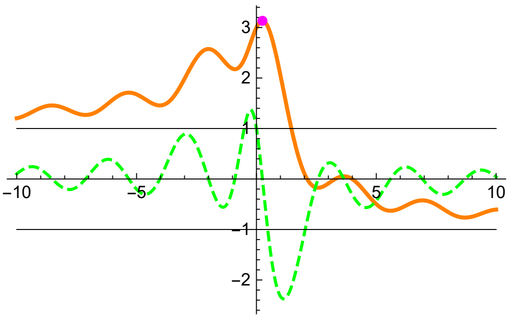
In[∘]:=
```
y1 = Limit[f[x], x → Infinity]
```

Out[∘]= −1

In[∘]:=
```
y2 = Limit[f[x], x → -Infinity]
```

Out[∘]= 1

In[∘]:=
```
Show[graff, Epilog → {{Line[{{-10, y1}, {10, y1}}], Line[{{-10, y2}, {10, y2}}]},
   {PointSize[.02], Magenta, Point[pontos]}}, PlotRange → All]
```

Out[∘]=


Exercício 7.20

In[∘]:=
```
f[x_] = 2 x - x^2;
g[x_] = (3 x - 3)/2;
```

In[∘]:=
```
Solve[f[x] == g[x], x]
```
Out[∘]= $\left\{\{x \to -1\}, \left\{x \to \frac{3}{2}\right\}\right\}$

In[∘]:=
```
x1 = x /. Last[%]
```
Out[∘]= $\frac{3}{2}$

In[∘]:=
```
Solve[g[x] == 0, x] // Flatten
```
Out[∘]= $\{x \to 1\}$

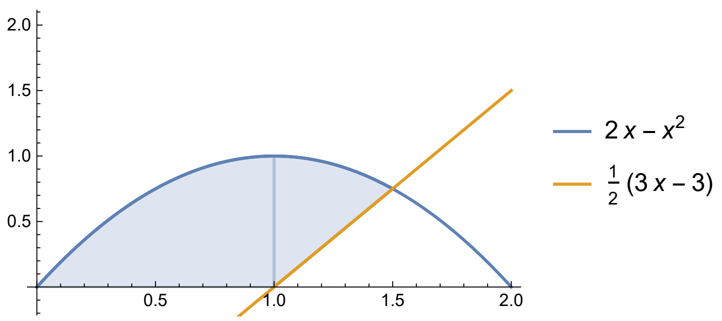
In[∘]:=
```
x2 = x /. %
```
Out[∘]= 1

In[∘]:=
```
Integrate[f[x], {x, 0, x2}] + Integrate[f[x] - g[x], {x, x2, x1}]
```
Out[∘]= $\frac{15}{16}$

Alternativa

In[∘]:=
```
h[x_] := Piecewise[{{g[x], x2 ≤ x ≤ x1}}]
```
In[∘]:=
```
h[x]
```
Out[∘]= $\begin{cases} \frac{1}{2}(-3+3x) & 1 \le x \le \frac{3}{2} \\ 0 & \text{True} \end{cases}$

In[∘]:=
```
Integrate[f[x] - h[x], {x, 0, x1}]
```
Out[∘]= $\frac{15}{16}$

In[∘]:=
```
Show[Plot[{f[x], g[x]}, {x, 0, 2}, PlotLegends → {f[x], g[x]}],
 Plot[f[x], {x, 0, x2}, Filling → Axis],
 Plot[{f[x], g[x]}, {x, x2, x1}, Filling → {1 → {2}}],
 PlotRange → {Automatic, {-0.1, 2}}]
```
Out[∘]=


Exercício 7.21

a)

In[∘]:=
```
Remove[f]
f[x_] = x^2 - 1;
ff[x_] = f[f[x]];
```

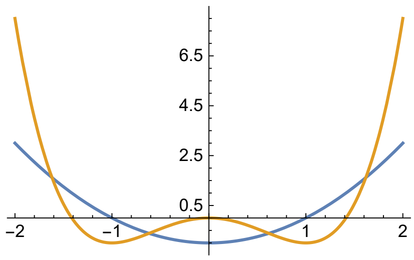
In[∘]:=
```
{f[5], f[1.75], f[Sqrt[19]], ff[x]}
```

Out[∘]= $\left\{24,\ 2.0625,\ 18,\ -1+\left(-1+x^2\right)^2\right\}$

b)

In[∘]:=
```
Plot[{f[x], ff[x]}, {x, -2, 2}]
```

Out[∘]=



c)

In[∘]:=
```
sol = Solve[f[x] == ff[x], x]
```

Out[∘]= $\left\{\left\{x\to\frac{1}{2}\left(-1-\sqrt{5}\right)\right\},\ \left\{x\to\frac{1}{2}\left(1-\sqrt{5}\right)\right\},\ \left\{x\to\frac{1}{2}\left(-1+\sqrt{5}\right)\right\},\ \left\{x\to\frac{1}{2}\left(1+\sqrt{5}\right)\right\}\right\}$

In[∘]:=
```
X = x /. sol // Sort
```

Out[∘]= $\left\{\frac{1}{2}\left(-1-\sqrt{5}\right),\ \frac{1}{2}\left(1-\sqrt{5}\right),\ \frac{1}{2}\left(-1+\sqrt{5}\right),\ \frac{1}{2}\left(1+\sqrt{5}\right)\right\}$

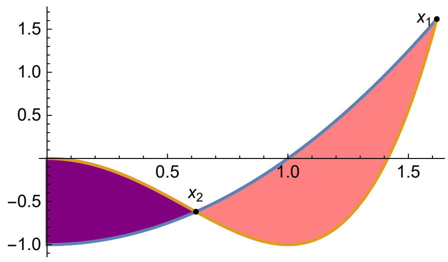
d)

In[∘]:=
```
{x2, x1} = X[[3 ;; 4]]
```

Out[∘]= $\left\{\frac{1}{2}\left(-1+\sqrt{5}\right),\ \frac{1}{2}\left(1+\sqrt{5}\right)\right\}$

In[∘]:=
```
Plot[{f[x], ff[x]}, {x, 0, x1}, Filling → {1 → {{2}, {Purple, Pink}}},
 Epilog → {Point[{{x2, f[x2]}, {x1, f[x1]}}],
   Text["x₁", {x1 - 0.05, f[x1]}], Text["x₂", {x2, f[x2] + 0.2}]}]
```

Out[∘]=



e)

A área é igual à soma das áreas a roxo e a cor de rosa.

In[∘]:=
```
Integrate[ff[x] - f[x], {x, 0, x2}] + Integrate[f[x] - ff[x], {x, x2, x1}]
```

Out[∘]= $\frac{6}{5}$

f) Duas retas são ortogonais ou perpendiculares, neste caso, se o declive de uma for igual igual ao simétrico do inverso da outra.

In[∘]:=
```
Clear[x1, x2]
```

In[∘]:=
```
Solve[f'[-x] == -1 / f'[x], x]
```

Out[∘]= $\left\{\left\{x \to -\frac{1}{2}\right\}, \left\{x \to \frac{1}{2}\right\}\right\}$

Abcissas dos pontos de tangência

In[∘]:=
```
{x1, x2} = x /. %
```

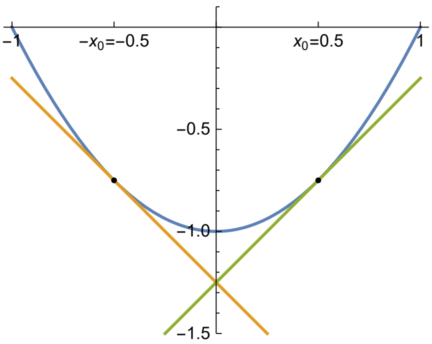
Out[∘]= $\left\{-\frac{1}{2}, \frac{1}{2}\right\}$

Retas

In[∘]:=
```
r1[x_] = f'[x1] (x - x1) + f[x1];
r2[x_] = f'[x2] (x - x2) + f[x2];
```

In[∘]:=
```
Plot[{f[x], r1[x], r2[x]}, {x, -1, 1}, AspectRatio → Automatic,
 Epilog -> {Point[{{x2, f[x2]}, {x1, f[x1]}}]},
 PlotRange → {Automatic, {-1.5, 0.1}},
 Ticks → {{-1, {-0.5, "-x0=-0.5"}, 0, {0.5, "x0=0.5"}, 1}, Automatic}]
```

Out[∘]=



8. Funções de ordem superior

Exercício 8.1

a)

In[∘]:=
```
lista = 2 RandomInteger[{1, 25}, 20] - 1
```

Out[∘]= {11, 45, 37, 19, 9, 29, 17, 19, 49, 19, 5, 17, 15, 23, 41, 27, 21, 7, 9, 33}

Outra solução:

```
In[ ]:= listaImpares = Table[i, {i, 1, 50, 2}]

Out[ ]= {1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23,
         25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49}

In[ ]:= RandomChoice[listaImpares, 20]

Out[ ]= {47, 23, 33, 39, 39, 11, 15, 25, 47, 21, 29, 41, 23, 11, 21, 41, 43, 13, 5, 21}

In[ ]:= n = Length[lista];

In[ ]:= media1 = Apply[Plus, lista] / n
```

Out[ ]= $\frac{113}{5}$

```
In[ ]:= media2 = Total[lista] / n
```

Out[ ]= $\frac{113}{5}$

```
In[ ]:= media3 = Mean[lista]
```

Out[ ]= $\frac{113}{5}$

b) Recordemos que a mediana de um conjunto de n observações é a média dos seus valores centrais (considerando as observações ordenadas), se n é par e é igual ao seu valor central, se n é ímpar.

```
In[ ]:= lista // Sort

Out[ ]= {5, 7, 9, 9, 11, 15, 17, 17, 19, 19, 19, 21, 23, 27, 29, 33, 37, 41, 45, 49}

In[ ]:= valorescentrais = Take[%, {n / 2, n / 2 + 1}]

Out[ ]= {19, 19}

In[ ]:= mediana1 = Mean[valorescentrais]

Out[ ]= 19

In[ ]:= mediana2 = Median[lista]

Out[ ]= 19
```

A solução anterior encontrada para obter o valor de mediana1, usa já o facto de n ser par. Podemos escrever uma função para obter a mediana, tendo em conta a paridade de n.

```
In[ ]:= MyMedian[l_] := Apply[Plus, Take[Sort[l], {Length[l] / 2, Length[l] / 2 + 1}]] / 2 /;
          EvenQ[Length[l]]

In[ ]:= MyMedian[l_] := Part[Sort[l], (Length[l] + 1) / 2] /; OddQ[Length[l]]

In[ ]:= MyMedian[lista]

Out[ ]= 19

In[ ]:= Join[lista, {20}]

Out[ ]= {11, 45, 37, 19, 9, 29, 17, 19, 49, 19, 5, 17, 15, 23, 41, 27, 21, 7, 9, 33, 20}
```

In[ ]:=
```
MyMedian[%]
```
Out[ ]= 19

In[ ]:=
```
Median[%%]
```
Out[ ]= 19

9. Funções anónimas

Exercício 9.6

In[ ]:=
```
Apply[And, EvenQ[#]] &
```
Out[ ]= And @@ EvenQ[#1] &

In[ ]:=
```
Select[{{1, 2, 4}, {3, 7, 5, 9}, {8, 2}, {11, 13, 15}, {16}, {77}},
 Apply[And, EvenQ[#]] &]
```
Out[ ]= {{8, 2}, {16}}

ou

In[ ]:=
```
AllTrue[#, EvenQ] &
```
Out[ ]= AllTrue[#1, EvenQ] &

In[ ]:=
```
Select[{{1, 2, 4}, {3, 7, 5, 9}, {8, 2}, {11, 13, 15}, {16}, {77}},
 AllTrue[#, EvenQ] &]
```
Out[ ]= {{8, 2}, {16}}

Exercício 9.7

In[ ]:=
```
lista = Table[RandomInteger[12], 10, RandomInteger[{2, 8}]]
```
Out[ ]= {{10, 8, 6, 3}, {4, 12, 0, 4, 11}, {5, 9, 10, 0, 11, 2}, {11, 12},
 {9, 2, 10}, {1, 11, 12}, {7, 11}, {3, 11, 1, 3, 10, 11, 1}, {9, 7, 2}, {2, 0}}

In[ ]:=
```
Select[lista, MemberQ[#, 7] &]
```
Out[ ]= {{7, 11}, {9, 7, 2}}

Exercício 9.8

In[11]:=
```
{l = Select[Range[97], (PrimeQ[#] && PrimeQ[# + 2]) &], l + 2} // Transpose
```
Out[11]= {{3, 5}, {5, 7}, {11, 13}, {17, 19}, {29, 31}, {41, 43}, {59, 61}, {71, 73}}

10. Moldes

Exercício 10.6

In[∘]:=
```
Clear@zeros;
zeros[p_?(PolynomialQ[#, x] &)] := x /. Solve[p == 0, x, Reals]
```

In[∘]:=
```
zeros /@ {x^2 - 1, x^2 + 1}
```

Out[∘]=
```
{{-1, 1}, x}
```

Se o polinómio não tiver raízes, não “funciona”!

Alternativas

In[∘]:=
```
Clear@zeros;
zeros[p_?(PolynomialQ[#, x] &)] :=
 Piecewise[{{x /. l, (l = Solve[p == 0, x, Reals]) ≠ {}}}, l]
```

In[∘]:=
```
zeros /@ {x^2 - 1, x^2 + 1}
```

Out[∘]=
```
{{-1, 1}, {}}
```

ou

In[∘]:=
```
Clear@zeros;
zeros[p_?(PolynomialQ[#, x] &)] := If[(l = Solve[p == 0, x, Reals]) ≠ {}, x /. l, l]
```

In[∘]:=
```
zeros /@ {x^2 - 1, x^2 + 1}
```

Out[∘]=
```
{{-1, 1}, {}}
```

ou

In[∘]:=
```
Clear@zeros;
zeros[p_?(PolynomialQ[#, x] && ((l = Solve[# == 0, x, Reals]) ≠ {}) &)] := x /. l
zeros[p_?(PolynomialQ[#, x] &)] := {}
```

In[∘]:=
```
zeros /@ {x^2 - 1, x^2 + 1}
```

Out[∘]=
```
{{-1, 1}, {}}
```

Exercício 10.10

a)

In[∘]:=
```
rep[{(x_) ..}] := x;
```

b)

In[∘]:=
```
eliminarep[lista_List] := Map[rep, Split[lista]]
```

Verificação

In[∘]:=
```
eliminarep[{x, y, y, y, x, x, z, y, y, z, z, z, z}]
```

Out[•]= {x, y, x, z, y, z}

c) Neste molde encaixam listas contendo dois elementos consecutivos iguais.

d) Esta função aceita como argumento uma lista. Usando o comando ReplaceRepeated (//.) aplica sucessivamente uma regra de substituição baseada no molde referido na alínea (c) onde os elementos repetidos colocados em posições adjacentes são eliminados. O objetivo desta função é exatamente o mesmo da função definida na alínea b).

```
In[•]:= eliminaRep[lista_List] := lista //. {a___, b_, b_, c___} → {a, b, c};

In[•]:= eliminaRep[{x, y, y, y, x, x, z, y, y, z, z, z, z}]

Out[•]= {x, y, x, z, y, z}
```

Observação: Qual é a diferença entre a função eliminaRep e a seguinte?

```
In[•]:= eliminaRep2[lista_List] := lista /. {a___, b_, b_, c___} → {a, b, c}
```

e)

```
In[•]:= lista = RandomInteger[1, 100]

Out[•]= {1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0,
         0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0,
         1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0,
         1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1}

In[•]:= eliminaRep[lista]

Out[•]= {1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
         0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1}
```

Exercício 10.14

```
In[•]:= polinomio[l : _Integer ..] := {l}.x^Range[0, Length[{l}] - 1]

In[•]:= polinomio[1, 2, 3, 4]
```

Out[•]= $1 + 2x + 3x^2 + 4x^3$

Alternativa

In[•]:= `polinomio2[l : _Integer ..] := Sum[{l}[[i]]` x^{i-1}`, {i, 1, Length[{l}]}]`

```
In[•]:= polinomio2[1, 2, 3, 4]
```

Out[•]= $1 + 2x + 3x^2 + 4x^3$

11. Recursividade

Exercício 11.9

a)

In[∘]:=
```
umDigitoQ[n_Integer?Positive] := Length[IntegerDigits[n]] == 1
```

b)

In[∘]:=
```
raizdigital[n_Integer?umDigitoQ] := n
raizdigital[n_Integer] := raizdigital[Total@IntegerDigits[n]]
```

c)

In[∘]:=
```
DigitalRoot[n_Integer?Positive] := FixedPoint[Total@IntegerDigits[#] &, n]
```

d)

In[∘]:=
```
Map[raizdigital, Range[100]]
```

Out[∘]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6,
 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4,
 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3,
 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1}
```

In[∘]:=
```
Map[DigitalRoot, Range[100]]
```

Out[∘]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6,
 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4,
 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3,
 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1}
```

e)

In[∘]:=
```
1 + Mod[Range[100] - 1, 9]
```

Out[∘]=
```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6,
 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4,
 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3,
 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1}
```

Exercício 11.10

a)

In[∘]:=
```
col[n_?(Positive[#] && OddQ[#] &)] := Sequence[n, col[3 n + 1]];
col[n_?(Positive[#] && EvenQ[#] &)] := Sequence[n, col[n / 2]];
col[1] = 1;
```

In[∘]:=
```
col[22]
```

Out[∘]=
```
Sequence[22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1]
```

In[∘]:=
```
col[35]
```

Out[∘]=
```
Sequence[35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1]
```

In[∘]:=
```
col[70]
```

Out[∘]=
```
Sequence[70, 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1]
```

b)

In[∘]:=
```
vida[n_Integer?Positive] := Length[{col[n]}]
```

In[∘]:=
```
vida[22]
```

Out[ ]= 16

```
In[ ]:= vida[35]
```

Out[ ]= 14

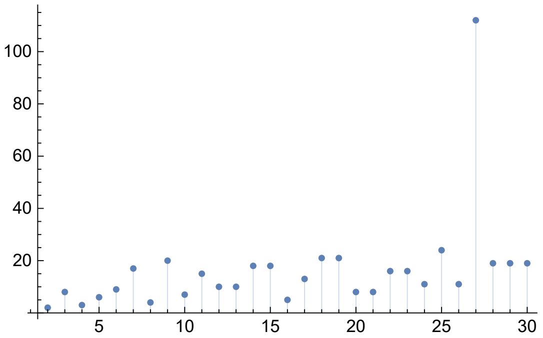
```
In[ ]:= vida[70]
```

Out[ ]= 15

c)

```
In[ ]:= ListPlot[Table[{n, vida[n]}, {n, 2, 30}], Filling → Axis, PlotRange → All]
```

Out[ ]=



Exercício 11.11

a)

```
In[ ]:= Clear[f, n]
```

```
In[ ]:= f[n_Integer ? (Positive[#] && EvenQ[#] &)] = n / 2;
f[n_Integer ? (Positive[#] && OddQ[#] &)] = 3 n + 1;
```

b)

```
In[ ]:= collatz[k_Integer ? Positive, 0] = k;
collatz[k_Integer ? Positive, n_Integer ? Positive] = f[collatz[k, n - 1]];
```

```
In[ ]:= collatz[10, 5]
```

Out[ ]= 2

c)

```
In[ ]:= TableForm[(listaCollatz = Table[collatz[k, n], {k, 10}, {n, 0, 20}]),
  TableSpacing → {1, 1}]
```

Out[ ]//TableForm=

1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2
2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4
3	10	5	16	8	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4
4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1
5	16	8	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1
6	3	10	5	16	8	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1
7	22	11	34	17	52	26	13	40	20	10	5	16	8	4	2	1	4	2	1	4
8	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2
9	28	14	7	22	11	34	17	52	26	13	40	20	10	5	16	8	4	2	1	4
10	5	16	8	4	2	1	4	2	1	4	2	1	4	2	1	4	2	1	4	2

d)

```
In[◦]:= {___, 1, ___};
```

e)

```
In[◦]:= eliminar[{a___, 1, ___}] = {a, 1};
```

A conjetura de Collatz afirma que, qualquer que seja o número k, existe um número natural n tal que $c_n^k = 1$.

```
In[◦]:= Map[eliminar, listaCollatz]
Out[◦]= {{1}, {2, 1}, {3, 10, 5, 16, 8, 4, 2, 1},
  {4, 2, 1}, {5, 16, 8, 4, 2, 1}, {6, 3, 10, 5, 16, 8, 4, 2, 1},
  {7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1}, {8, 4, 2, 1},
  {9, 28, 14, 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1},
  {10, 5, 16, 8, 4, 2, 1}}

In[◦]:= Length[Cases[Map[eliminar, listaCollatz], _eliminar]] == 0
Out[◦]= True

In[◦]:= Length[
   Cases[Map[eliminar, Table[collatz[k, n], {k, 50}, {n, 0, 20}]], _eliminar]] == 0
Out[◦]= False

In[◦]:= Length[
   Cases[Map[eliminar, Table[collatz[k, n], {k, 50}, {n, 0, 130}]], _eliminar]] == 0
Out[◦]= True
```

12. Leitura e escrita de ficheiros

Exercício 12.1

a)

```
In[◦]:= SetDirectory[NotebookDirectory[]];
```

b)

```
In[◦]:= l = Import["plano.xlsx"] // First;

In[◦]:= l // TableForm
```

Out[∘]//TableForm=

Ano	Regime	Unidade Curricular	Área Cient.	ECTS
1.	S1	Álgebra Linear I+C2C2:C41	M	7.5
1.	S1	Cálculo I	M	7.5
1.	S1	Matemática Computacional I	M	7.5
1.	S1	Tópicos de Matemática	M	7.5
1.	S2	Álgebra Linear II	M	7.5
1.	S2	Cálculo II	M	7.5
1.	S2	Geometria	M	5.
1.	S2	Matemática Computacional II	M	5.
1.	S2	Matemática Discreta	M	5.
2.	S1	Álgebra	M	7.5
2.	S1	Análise I	M	7.5
2.	S1	Análise Numérica I	M	7.5
2.	S1	História do Pensamento Matemático	M	7.5
2.	S2	Análise II	M	7.5
2.	S2	Análise Numérica II	M	7.5
2.	S2	Complementos de Álgebra	M	7.5
2.	S2	Geometria Diferencial	M	7.5
3.	S1	Análise Complexa	M	7.5
3.	S1	Lógica e Fundamentos da Matemática	M	7.5
3.	S1	Métodos Matemáticos da Física	M	7.5
3.	S1	Probabilidades e Aplicações	M	7.5
3.	S2	Estatística	M	7.5
3.	S2	Temas da Matemática	M	7.5
3.	S2	Opção I		5.
3.		Álgebra Universal e Categorias	M	5.
3.		Autómatos e Linguagens Formais	CComp	5.
3.		Teoria de Números Computacional	M	5.
3.		Tópicos de Sistemas Dinâmicos	M	5.
3.	S2	Opção II		5.
3.		Álgebra Universal e Categorias	M	5.
3.		Autómatos e Linguagens Formais	CComp	5.
3.		Teoria de Números Computacional	M	5.
3.		Tópicos de Sistemas Dinâmicos	M	5.
3.	S2	Opção UMinho		5.
3.		Ambiente e Energia	ENG	6.
3.		Bioética	ENF	6.
3.		Comunicação e Arte	CCom	5.
3.		Corpo, Género e Sexualidade	SOC	5.
3.		Desporto e Saúde	EF	6.
3.		Ética: As Grandes Questões do Nosso Tempo	F	6.
3.		Fundamentos e Práticas de Primeiros Socorros	ENF	6.
3.		Inglês Académico	LLC	6.
3.		Liderança e Empreendedorismo	ENG	6.
3.		Língua Estrangeira Nível 1 - Francês	LLC	6.
3.		Ótica Aplicada para Fotografia Digital	Fis	6.
3.		Princípios de Economia	E	6.
3.		Princípios de Empreendedorismo	G	6.
3.		Segurança e Saúde do Trabalho	ENG	6.
3.		Tribunais Internacionais	D	5.

c)

In[∘]:=
```
plano = l /. ({x_Real, y__} → {Round[x], y})
```

Out[∘]=
```
{{Ano, Regime, Unidade Curricular, Área Cient., ECTS},
 {1, S1, Álgebra Linear I+C2C2:C41, M, 7.5}, {1, S1, Cálculo I, M, 7.5},
 {1, S1, Matemática Computacional I, M, 7.5}, {1, S1, Tópicos de Matemática, M, 7.5},
 {1, S2, Álgebra Linear II, M, 7.5}, {1, S2, Cálculo II, M, 7.5},
 {1, S2, Geometria, M, 5.}, {1, S2, Matemática Computacional II, M, 5.},
 {1, S2, Matemática Discreta, M, 5.}, {2, S1, Álgebra, M, 7.5},
 {2, S1, Análise I, M, 7.5}, {2, S1, Análise Numérica I, M, 7.5},
 {2, S1, História do Pensamento Matemático, M, 7.5}, {2, S2, Análise II, M, 7.5},
 {2, S2, Análise Numérica II, M, 7.5}, {2, S2, Complementos de Álgebra, M, 7.5},
 {2, S2, Geometria Diferencial, M, 7.5}, {3, S1, Análise Complexa, M, 7.5},
 {3, S1, Lógica e Fundamentos da Matemática, M, 7.5},
 {3, S1, Métodos Matemáticos da Física, M, 7.5},
 {3, S1, Probabilidades e Aplicações, M, 7.5},
 {3, S2, Estatística, M, 7.5}, {3, S2, Temas da Matemática, M, 7.5},
 {3, S2, Opção I, , 5.}, {3, , Álgebra Universal e Categorias, M, 5.},
 {3, , Autómatos e Linguagens Formais, CComp, 5.},
 {3, , Teoria de Números Computacional, M, 5.},
 {3, , Tópicos de Sistemas Dinâmicos, M, 5.}, {3, S2, Opção II, , 5.},
 {3, , Álgebra Universal e Categorias, M, 5.},
 {3, , Autómatos e Linguagens Formais, CComp, 5.},
 {3, , Teoria de Números Computacional, M, 5.},
 {3, , Tópicos de Sistemas Dinâmicos, M, 5.},
 {3, S2, Opção UMinho, , 5.}, {3, , Ambiente e Energia, ENG, 6.},
 {3, , Bioética, ENF, 6.}, {3, , Comunicação e Arte, CCom, 5.},
 {3, , Corpo, Género e Sexualidade, SOC, 5.}, {3, , Desporto e Saúde, EF, 6.},
 {3, , Ética: As Grandes Questões do Nosso Tempo, F, 6.},
 {3, , Fundamentos e Práticas de Primeiros Socorros, ENF, 6.},
 {3, , Inglês Académico, LLC, 6.}, {3, , Liderança e Empreendedorismo, ENG, 6.},
 {3, , Língua Estrangeira Nível 1 - Francês, LLC, 6.},
 {3, , Ótica Aplicada para Fotografia Digital, Fis, 6.},
 {3, , Princípios de Economia, E, 6.}, {3, , Princípios de Empreendedorismo, G, 6.},
 {3, , Segurança e Saúde do Trabalho, ENG, 6.},
 {3, , Tribunais Internacionais, D, 5.}}
```

d)

In[∘]:=
```
areaM = Cases[plano, {_, _, _, "M", _}] // Length
```

Out[∘]=
```
29
```

e)

```
In[•]:= semestre12 = Cases[plano, {1, "S2", __}]
Out[•]= {{1, S2, Álgebra Linear II, M, 7.5},
  {1, S2, Cálculo II, M, 7.5}, {1, S2, Geometria, M, 5.},
  {1, S2, Matemática Computacional II, M, 5.}, {1, S2, Matemática Discreta, M, 5.}}

In[•]:= semestre12[[All, 3]]
Out[•]= {Álgebra Linear II, Cálculo II, Geometria,
  Matemática Computacional II, Matemática Discreta}
```

Alternativa:

```
In[•]:= f[{1, "S2", a_, __}] := a

In[•]:= f[_] = {};

In[•]:= Map[f, plano] // Flatten
Out[•]= {Álgebra Linear II, Cálculo II, Geometria,
  Matemática Computacional II, Matemática Discreta}
```

f)

```
In[•]:= posicaoOpcaoUminho = Position[plano, "Opção UMinho"] // Flatten
Out[•]= {35, 3}

In[•]:= Length[plano] - (First[posicao1opcaoUminho] + 1) + 1
  ··· First: Nonatomic expression expected at position 1 in First[posicao1opcaoUminho].
Out[•]= 50 - First[posicao1opcaoUminho]
```

g)

```
In[•]:= nova = {3, "", "Ferramentas para aulas tecnologicamente assistidas ", "ENG.", 5.`};

In[•]:= AppendTo[plano, nova]

Out[•]= {{Ano, Regime, Unidade Curricular, Área Cient., ECTS},
  {1, S1, Álgebra Linear I+C2C2:C41, M, 7.5}, {1, S1, Cálculo I, M, 7.5},
  {1, S1, Matemática Computacional I, M, 7.5}, {1, S1, Tópicos de Matemática, M, 7.5},
  {1, S2, Álgebra Linear II, M, 7.5}, {1, S2, Cálculo II, M, 7.5},
  {1, S2, Geometria, M, 5.}, {1, S2, Matemática Computacional II, M, 5.},
  {1, S2, Matemática Discreta, M, 5.}, {2, S1, Álgebra, M, 7.5},
  {2, S1, Análise I, M, 7.5}, {2, S1, Análise Numérica I, M, 7.5},
  {2, S1, História do Pensamento Matemático, M, 7.5}, {2, S2, Análise II, M, 7.5},
  {2, S2, Análise Numérica II, M, 7.5}, {2, S2, Complementos de Álgebra, M, 7.5},
  {2, S2, Geometria Diferencial, M, 7.5}, {3, S1, Análise Complexa, M, 7.5},
  {3, S1, Lógica e Fundamentos da Matemática, M, 7.5},
  {3, S1, Métodos Matemáticos da Física, M, 7.5},
  {3, S1, Probabilidades e Aplicações, M, 7.5},
  {3, S2, Estatística, M, 7.5}, {3, S2, Temas da Matemática, M, 7.5},
  {3, S2, Opção I, , 5.}, {3, , Álgebra Universal e Categorias, M, 5.},
  {3, , Autómatos e Linguagens Formais, CComp, 5.},
  {3, , Teoria de Números Computacional, M, 5.},
  {3, , Tópicos de Sistemas Dinâmicos, M, 5.}, {3, S2, Opção II, , 5.},
  {3, , Álgebra Universal e Categorias, M, 5.},
  {3, , Autómatos e Linguagens Formais, CComp, 5.},
  {3, , Teoria de Números Computacional, M, 5.},
  {3, , Tópicos de Sistemas Dinâmicos, M, 5.},
  {3, S2, Opção UMinho, , 5.}, {3, , Ambiente e Energia, ENG, 6.},
  {3, , Bioética, ENF, 6.}, {3, , Comunicação e Arte, CCom, 5.},
  {3, , Corpo, Género e Sexualidade, SOC, 5.}, {3, , Desporto e Saúde, EF, 6.},
  {3, , Ética: As Grandes Questões do Nosso Tempo, F, 6.},
  {3, , Fundamentos e Práticas de Primeiros Socorros, ENF, 6.},
  {3, , Inglês Académico, LLC, 6.}, {3, , Liderança e Empreendedorismo, ENG, 6.},
  {3, , Língua Estrangeira Nível 1 - Francês, LLC, 6.},
  {3, , Ótica Aplicada para Fotografia Digital, Fis, 6.},
  {3, , Princípios de Economia, E, 6.}, {3, , Princípios de Empreendedorismo, G, 6.},
  {3, , Segurança e Saúde do Trabalho, ENG, 6.},
  {3, , Tribunais Internacionais, D, 5.},
  {3, , Ferramentas para aulas tecnologicamente assistidas , ENG., 5.}}
```

h)

```
In[•]:= Export["plano2.xlsx", plano]

Out[•]= plano2.xlsx
```

i)

```
In[•]:= pos1 = Position[plano, "Opção I"] // Flatten
Out[•]= {25, 3}

In[•]:= pos2 = Position[plano, "Opção II"] // First
Out[•]= {30, 3}

In[•]:= plano[[First@pos1 + 1 ;; First@pos2 - 1, All]][[All, 3]]
Out[•]= {Álgebra Universal e Categorias, Autómatos e Linguagens Formais,
  Teoria de Números Computacional, Tópicos de Sistemas Dinâmicos}
```

j)

```
In[•]:= Clear@f

In[•]:= f[{_, "S1" | "S2", _, _, ects_}] := ects

In[•]:= f[_] := 0

In[•]:= Map[f, plano] // Total
Out[•]= 180.
```

Exercício 12.2

a)

```
In[•]:= SetDirectory[NotebookDirectory[]];
```

b)

```
In[•]:= FilePrint["proverbios.txt"]

Amigos, amigos, negócios à parte
Amigo de todos, amigo de ninguém
Amigo não empata amigo
Mais depressa se cansa um amigo de louvar, que um inimigo de injuriar
Os amigos são para as ocasiões
Quem te avisa, teu amigo é
Amigo do meu amigo, meu amigo é
Quem tem amigos, não morre na cadeia
As boas contas fazem os bons amigos
```

c)

```
In[•]:=     listaproverbios = ReadList["proverbios.txt", Word, WordSeparators → {"\n"}]
Out[•]= {Amigos, amigos, negócios à parte,
  Amigo de todos, amigo de ninguém, Amigo não empata amigo,
  Mais depressa se cansa um amigo de louvar, que um inimigo de injuriar,
  Os amigos são para as ocasiões, Quem te avisa, teu amigo é,
  Amigo do meu amigo, meu amigo é, Quem tem amigos, não morre na cadeia,
  As boas contas fazem os bons amigos}
```

Este comando cria uma lista, **`listaproverbios`**, cujos elementos são strings correspondentes ao conteúdo de cada linha do ficheiro **`proverbios.txt`**.

d)

In[∘]:= `Length[listaproverbios]`

Out[∘]= 9

e)

In[∘]:= `listaproverbios[[5]]`

Out[∘]= Os amigos são para as ocasiões

f)

In[∘]:= `StringReplace[listaproverbios, "," → ""]`

Out[∘]=
```
{Amigos amigos negócios à parte,
 Amigo de todos amigo de ninguém, Amigo não empata amigo,
 Mais depressa se cansa um amigo de louvar que um inimigo de injuriar,
 Os amigos são para as ocasiões, Quem te avisa teu amigo é,
 Amigo do meu amigo meu amigo é, Quem tem amigos não morre na cadeia,
 As boas contas fazem os bons amigos}
```

g)

In[∘]:=
```
(StringCases[listaproverbios, "amigo", IgnoreCase → True] // Flatten // Length) -
 (StringCases[listaproverbios, "amigos", IgnoreCase → True] // Flatten // Length)
```

Out[∘]= 9

Alternativa 1:

In[∘]:=
```
(StringCases[listaproverbios, "amigo" | "Amigo"] // Flatten // Length) -
 (StringCases[listaproverbios, "amigos" | "Amigos"] // Flatten // Length)
```

Out[∘]= 9

Alternativa 2:

In[∘]:=
```
(StringCount[listaproverbios, "amigo" | "Amigo"] // Total) -
 (StringCount[listaproverbios, "amigos" | "Amigos"] // Total)
```

Out[∘]= 9

h)

In[∘]:= `StringCount[listaproverbios, " "]`

Out[∘]= {4, 5, 3, 12, 5, 5, 6, 6, 6}

In[∘]:=
```
Select[listaproverbios,
 StringCount[#, " "] == Max[StringCount[listaproverbios, " "]] &]
```

Out[∘]= {Mais depressa se cansa um amigo de louvar, que um inimigo de injuriar}

Alternativa 1:

In[∘]:= `SortBy[listaproverbios, StringCount[#, " "] &] // Last`

Out[∘]= Mais depressa se cansa um amigo de louvar, que um inimigo de injuriar

Alternativa 2:

```
In[•]:= MaximalBy[listaproverbios, StringCount[#, " "] &]
Out[•]= {Mais depressa se cansa um amigo de louvar, que um inimigo de injuriar}
```

i)

```
In[•]:= AppendTo[listaproverbios,
          "A conselho amigo, não feches o postigo"] // TableForm
Out[•]//TableForm=
Amigos, amigos, negócios à parte
Amigo de todos, amigo de ninguém
Amigo não empata amigo
Mais depressa se cansa um amigo de louvar, que um inimigo de injuriar
Os amigos são para as ocasiões
Quem te avisa, teu amigo é
Amigo do meu amigo, meu amigo é
Quem tem amigos, não morre na cadeia
As boas contas fazem os bons amigos
A conselho amigo, não feches o postigo
```

13. Exercícios Suplementares

Exercício 13.1

a)

```
In[•]:= (lista1 = Table[n m, {n, 20, 50}, {m, 1, 6}]) // Short
Out[•]//Short= {{20, 40, 60, 80, 100, 120}, {21, 42, 63, 84, 105, 126},
   <<27>>, {49, 98, 147, 196, 245, 294}, {50, 100, 150, 200, 250, 300}}
```

b)

```
In[•]:= (lista2 = Apply[Plus, lista1, 2]) // Short
Out[•]//Short= {420, 441, 462, 483, 504, 525, 546, 567, 588, 609, 630, 651, 672, <<5>>,
   798, 819, 840, 861, 882, 903, 924, 945, 966, 987, 1008, 1029, 1050}
```

ou

```
In[•]:= Map[Total, lista1] // Short
Out[•]//Short= {420, 441, 462, 483, 504, 525, 546, 567, 588, 609, 630, 651, 672, <<5>>,
   798, 819, 840, 861, 882, 903, 924, 945, 966, 987, 1008, 1029, 1050}
```

c)

```
In[•]:= (lista3 = IntegerDigits[lista2]) // Short
Out[•]//Short= {{4, 2, 0}, {4, 4, 1}, {4, 6, 2}, {4, 8, 3}, {5, 0, 4}, {5, 2, 5}, {5, 4, 6}, <<18>>,
   {9, 4, 5}, {9, 6, 6}, {9, 8, 7}, {1, 0, 0, 8}, {1, 0, 2, 9}, {1, 0, 5, 0}}
```

d)

```
In[•]:= molde = {(x_) ..};
```

e)

In[◦]:=
```
Cases[lista3, molde]
```
Out[◦]=
```
{{7, 7, 7}}
```

In[◦]:=
```
posicao = Position[lista3, molde]
```
Out[◦]=
```
{{18}}
```

f)

In[◦]:=
```
Apply[First, Extract[lista1, posicao]]
```
Out[◦]=
```
37
```

Exercício 13.2

In[◦]:=
```
sublimeQ[n_Integer?Positive] :=
 PerfectNumberQ[Total[Divisors[n]]] && PerfectNumberQ[Length[Divisors[n]]]
```

In[◦]:=
```
Select[Range[10000], sublimeQ]
```
Out[◦]=
```
{12}
```

Função otimizada:

In[◦]:=
```
sublimeQ[n_Integer?Positive] :=
 Module[{d = Divisors[n]}, PerfectNumberQ[Total[d]] && PerfectNumberQ[Length[d]]]
```

In[◦]:=
```
Select[Range[10000], sublimeQ]
```
Out[◦]=
```
{12}
```

In[◦]:=
```
astrosublime = << "http://w3.math.uminho.pt/~mif/MCII/sublime.txt"
```
Out[◦]=
```
6086555670238378989670371734243169622657830773351885970528324860512791691 \
 264
```

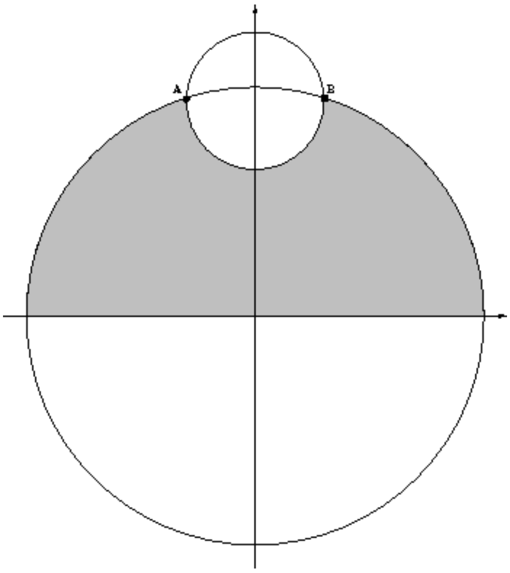
In[◦]:=
```
sublimeQ[astrosublime]
```
Out[◦]=
```
True
```

Exercício 13.3

In[◦]:=
```
SetDirectory[NotebookDirectory[]]
```
Out[◦]=
```
C:\Users\mif\Documents\0-EU\0-Aulas\2-MCompII\AulasLaboratoriais\Resolucoes
```

In[◦]:=
```
Import["circunferencias.png"]
```

Out[ ]=



In[ ]:= coordB = {1 / 3, Sin[ArcCos[1 / 3]]}

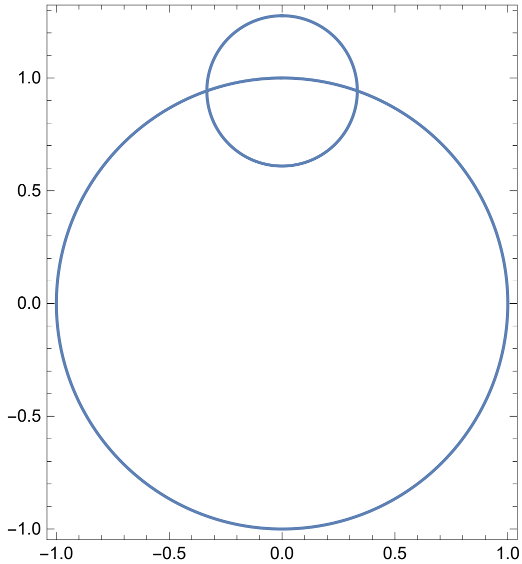
Out[ ]= $\left\{\frac{1}{3}, \frac{2\sqrt{2}}{3}\right\}$

In[ ]:= coordA = {-1 / 3, Sin[ArcCos[-1 / 3]]}

Out[ ]= $\left\{-\frac{1}{3}, \frac{2\sqrt{2}}{3}\right\}$

```
circG = ImplicitRegion[x^2 + y^2 == 1 || x^2 + (y - coordA[[2]]) ^2 == 1 / 9, {x, y}];
RegionPlot[circG, AspectRatio → Automatic]
```

Out[ ]=




```
2 (Integrate[-Sqrt[1 / 9 - x^2] + coordB[[2]], {x, 0, 1 / 3}] +
    Integrate[Sqrt[1 - x^2], {x, 1 / 3, 1}]) // Simplify
```

Out[ ]= $\frac{2\sqrt{2}}{9} - \frac{\pi}{18} + \text{ArcSec}[3]$

Exercício 13.4

a)

```
In[◦]:= lucas[0] = 2;
lucas[1] = 1;
lucas[n_Integer?NonNegative] := lucas[n] = lucas[n - 1] + lucas[n - 2]
```

b)

```
In[◦]:= L50 = Table[lucas[n], {n, 0, 49}]

Out[◦]= {2, 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207, 3571, 5778,
 9349, 15 127, 24 476, 39 603, 64 079, 103 682, 167 761, 271 443, 439 204, 710 647,
 1 149 851, 1 860 498, 3 010 349, 4 870 847, 7 881 196, 12 752 043, 20 633 239, 33 385 282,
 54 018 521, 87 403 803, 141 422 324, 228 826 127, 370 248 451, 599 074 578, 969 323 029,
 1 568 397 607, 2 537 720 636, 4 106 118 243, 6 643 838 879, 10 749 957 122, 17 393 796 001}

In[◦]:= L50 == Table[LucasL[n], {n, 0, 49}]

Out[◦]= True
```

c)

```
In[◦]:= Select[L50, (OddQ[#] && # > 100) &]

Out[◦]= {123, 199, 521, 843, 2207, 3571, 9349, 15 127, 39 603, 64 079, 167 761, 271 443, 710 647,
 1 149 851, 3 010 349, 4 870 847, 12 752 043, 20 633 239, 54 018 521, 87 403 803, 228 826 127,
 370 248 451, 969 323 029, 1 568 397 607, 4 106 118 243, 6 643 838 879, 17 393 796 001}
```

d)

```
In[◦]:= Clear@f

In[◦]:= f[x_] := GoldenRatio^x + (1 - GoldenRatio)^x
```

e)i.

```
In[◦]:= Divisible[Table[L50[[n]] - L50[[n - 4]], {n, 5, 49}], 5] // Intersection

Out[◦]= {True}

In[◦]:= Divisible[L50[[5 ;;]] - L50[[ ;; -5]], 5] // Intersection

Out[◦]= {True}
```

e)ii.

```
In[◦]:= L50 == Table[f[n], {n, 0, 49}] // FullSimplify

Out[◦]= True
```

e)iii

```
In[◦]:= pos = Position[L50, _?PrimeQ] - 1 // Flatten

Out[◦]= {0, 2, 4, 5, 7, 8, 11, 13, 16, 17, 19, 31, 37, 41, 47}

Select[pos, (# == 0 || PrimeQ[#] ||
    (FactorInteger[#][[1, 1]] == 2 && Length@FactorInteger[#] == 1)) &] == pos

Out[◦]= True
```

In[∘]:=
```
Cases[FactorInteger[pos], {{0, 1}} | {{2, _}} | {{_, 1}}] == FactorInteger[pos]
```
Out[∘]=
```
True
```

f)

In[∘]:=
$$\text{Limit}\left[\frac{f[n]}{f[n-1]}, n \to \text{Infinity}\right] == \text{GoldenRatio}$$
Out[∘]=
```
True
```

Exercício 13.5

a) Esta função aceita como argumento um número inteiro não negativo e devolve o número inteiro que se obtém ordenando, por ordem crescente, os algarismos do argumento.

In[∘]:=
```
ordena[x_Integer?NonNegative] := FromDigits[Sort[IntegerDigits[x]]]
```
In[∘]:=
```
ordena[1432]
```
Out[∘]=
```
1234
```

b)

In[∘]:=
```
reverso[x_Integer?Positive] := FromDigits[Reverse[IntegerDigits[x]]]
```
In[∘]:=
```
reverso[1543]
```
Out[∘]=
```
3451
```

```
reverso[-1543]
```
Out[∘]=
```
reverso[-1543]
```

c)

In[∘]:=
```
Clear@a
```
In[∘]:=
```
a[k_Integer?Positive, 1] = k;
```
In[∘]:=
```
a[k_Integer?Positive, n_Integer?Positive] :=
 a[k, n] = ordena[a[k, n - 1] + reverso[a[k, n - 1]]]
```
In[∘]:=
```
Table[a[241, n], {n, 1, 3}]
```
Out[∘]=
```
{241, 338, 1117}
```
In[∘]:=
```
Table[a[1, n], {n, 20}]
```
Out[∘]=
```
{1, 2, 4, 8, 16, 77, 145, 668, 1345, 6677, 13 444, 55 778, 133 345,
 666 677, 1 333 444, 5 567 777, 12 333 445, 66 666 677, 133 333 444, 556 667 777}
```
In[∘]:=
```
Table[a[3, n], {n, 20}]
```
Out[∘]=
```
{3, 6, 12, 33, 66, 123, 444, 888, 1677, 3489,
 12 333, 44 556, 111, 222, 444, 888, 1677, 3489, 12 333, 44 556}
```

Exercício 13.6

a)

```
In[•]:= rep[l : {(x_) ..}] := {x, Length[l]}
In[•]:= Map[rep, {{a, a, a, a}, {1, 2, 3}}]
Out[•]= {{a, 4}, rep[{1, 2, 3}]}
```

b)

```
In[•]:= compactar[l_List] := Map[rep, Split[l]]
In[•]:= compactar[{x, y, y, y, x, x, z, y, y, z, z, z, z}]
Out[•]= {{x, 1}, {y, 3}, {x, 2}, {z, 1}, {y, 2}, {z, 4}}
```

c)

```
In[•]:= Tally[{x, y, y, y, x, x, z, y, y, z, z, z, z}]
Out[•]= {{x, 3}, {y, 5}, {z, 5}}
In[•]:= mytally[l_List] := Map[rep, Split[Sort@l]]
In[•]:= mytally[{x, y, y, y, x, x, z, y, y, z, z, z, z}]
Out[•]= {{x, 3}, {y, 5}, {z, 5}}
```

Note - se que a função mytally não é exatamente igual à Tally, uma vez que os elementos aparecem ordenados.

```
In[•]:= Tally[{x, z, y, y, y, x, x, z, y, y, z, z, z, z}]
Out[•]= {{x, 3}, {z, 6}, {y, 5}}
In[•]:= mytally[{x, z, y, y, y, x, x, z, y, y, z, z, z, z}]
Out[•]= {{x, 3}, {y, 5}, {z, 6}}
```

d)

```
In[•]:= regra = {x___, {y_, i_}, {y_, j_}, z___} → {x, {y, i + j}, z};
        outracompactar[lista_List] := Map[{#, 1} &, lista] //. regra
```

A função nome devolve exatamente o mesmo objeto que a função compactar criada na alínea b), mas usando um processo diferente que é descrito a seguir:

1) dada uma lista como argumento, por exemplo $\{a_1,a_2,\ldots,a_n\}$, cria a lista $\{\{a_1,1\}, \{a_2,1\},\ldots, \{a_n,1\}\}$ aplicando, usando o Map, a função anónima {#, 1} & a cada elemento da lista dada como argumento;

2. usando repetidamente uma regra de substituição (ReplaceRepeated) baseada na utilização de um molde onde encaixam listas contendo elementos da forma $\{a,\mathrm{i}\}$, $\{a,\mathrm{j}\}$ em posições consecutivas e substituindo-os pelo elemento $\{a, \mathrm{i+j}\}$, obtém-se uma função que faz exatamente o mesmo que a função compactar.

```
In[•]:= outracompactar[{x, y, y, y, x, x, z, y, y, z, z, z, z}]
Out[•]= {{x, 1}, {y, 3}, {x, 2}, {z, 1}, {y, 2}, {z, 4}}
```

e)

In[ ]:=
```
(lbinario = ReadList["binario.txt", "Number"]) // Short
```

Out[ ]//Short=
```
{0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1,
 <<945>>, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1}
```

In[ ]:=
```
compactar[lbinario] // Short
```

Out[ ]//Short=
```
{{0, 3}, {1, 1}, {0, 1}, {1, 1}, {0, 1}, {1, 3}, {0, 1}, {1, 1}, {0, 2}, <<494>>,
 {1, 2}, {0, 1}, {1, 3}, {0, 1}, {1, 2}, {0, 1}, {1, 2}, {0, 1}, {1, 1}}
```

f)

In[ ]:=
```
descompactar[lista : {{_, _Integer?Positive} ..}] :=
 Flatten[Map[Table[#[[1]], {#[[2]]}] &, lista]]
```

In[ ]:=
```
(* verificação *)
descompactar[compactar[lbinario]] == lbinario
```

Out[ ]=
```
True
```