Resolução explicada dos exercícios 4, 5 e 6 da folha 2 (tratados nas aulas PL dos dias 10, 11, 12 e 13 de novembro)

exercício 4 Comecemos por observar que o valor exato é $z = 1$ para quaisquer x e $y \neq 0$. No Matlab tem-se

```
>> format long, x=100; for k=0:10, y=10^-k; z=((x+y)^2-x^2-2*x*y)/y^2, end

z =

     1


z =

   0.999999999839929


z =

   1.000000011117663


z =

   1.000001066120415


z =

   1.000124029773564


z =

   1.004518707797830


z =

   0.404543260869306


z =

 -50.524249497682646


z =

     -1.141917891709900e+04
```

```
z =

     1.522898673993811e+06


z =

    -2.115080133084667e+07
```

Ocorre cancelamento substrativo no cálculo do numerador. Por exemplo, para k=5, o valor exato do numerador é 1e-10 mas

```
>> x=100; y=1e-5; (x+y)^2-x^2-2*x*y

ans =

     1.004518707797830e-10
```

O numerador está assim calculado com um elevado erro relativo (perda de algarismos significativos corretos). Dividindo pelo denominador que é 1e-10, produz-se o resultado 1.004518707797830 que já tem um erro absoluto grande.

O cancelamento subtrativo é tanto mais severo quanto mais pequeno for y, isto é, quanto maior for o valor de k. Para k=10, tem-se

```
>> x=100; y=1e-10; (x+y)^2-x^2-2*x*y

ans =

    -2.115080133084667e-13
```

que não tem qualquer algarismo significativo correto. Dividido pelo valor exato 1e-20, obtem-se -2.115080133084667e+7 (recorde-se que o resultado correto é 1).

exercício 5

```
> S=inline('(1+1/n)^n')

S =

     Inline function:
     S(n) = (1+1/n)^n

>> S(2^52)

ans =

   2.718281828459045

>> S(2^53)
```

```
ans =

     1
```

Uma vez que se tem

```
>> exp(1)

ans =

   2.718281828459046
```

o valor de $S(2^{52})$ é uma boa aproximação do número e mas tal não acontece com $S(2^{53})$. Isto parece estar em contradição com o que se sabe da sucessão que é crescente e convergente. A culpa disto é dos erros de arredondamento. Com efeito, o sucessor de 1 em $\mathcal{F}$ é $1 + 2^{-52}$ e o valor de $1 + 2^{-53}$ é representado (arredondado) por 1.

exercício 6 A sucessão tende para zero porque o factorial $n!$ cresce mais rapidamente do que a exponencial a^n por muito grande que seja a base $a > 1$. No entanto, esta afirmação deve ser clarificada: só a partir de um certo valor de n, que depende de a, é que a sucessão começa a decrescer e a tender para zero. A relação

$$\frac{100^{n+1}}{(n+1)!} = \frac{100^n}{n!} \times \frac{100}{n+1} \tag{1}$$

mostra que a os termos da sucessão atingem o valor máximo para

$$\frac{100^{99}}{99!} = \frac{100^{100}}{100!}$$

e só a partir deste valor enorme começam a decrescer, no princípio muito lentamente como se percebe de

$$\frac{100^{101}}{101!} = \frac{100^{100}}{100!} \times \frac{100}{101} = \frac{100^{100}}{100!} \times 0.99...$$

$$\frac{100^{102}}{102!} = \frac{100^{101}}{101!} \times \frac{100}{102} = \frac{100^{101}}{101!} \times 0.98...$$

Será assim necessário calcular termos de ordem mais elevada obter valores que se aproximem de zero (o limite da sucessão). O problema é que antes que tal ocorra, produz-se um erro de overflow:

```
>> n=154; 100^n/factorial(n)

ans =

     3.236487262734163e+36

>> n=155; 100^n/factorial(n)

ans =

   Inf
```

Assim, o último termo que se consegue calcular diretamente a partir do termo geral da sucessão é para n=154, porque $100^{155} > realmax$. A relação (1) permite ultrapassar esta dificuldade. No Matlab, partir do primeiro termo u(1)=100, calculamos recursivamente cada um dos primeiros 300 termos a partir do anterior e listamos os últimos 10 termos calculados que já são muito próximos de zero.

```
>> u(1)=100; for n=2:300, u(n)=u(n-1)*100/n; end, u(291:300)'

ans =

   5.6974e-11
   1.9512e-11
   6.6592e-12
   2.2650e-12
   7.6781e-13
   2.5940e-13
   8.7338e-14
   2.9308e-14
   9.8021e-15
   3.2674e-15
```

Resolução explicada dos exercícios 9, 10 e 12 da folha 2 (ex. 9 e 10 foram resolvidos nas aulas PL dos dias 17, 18, 19 e 20 de novembro)

exercício 9.a) O código seguinte está disponível na área "Matlab" da Blackboard

```
function [soma, n]=expTaylor(x, tol)

% calcula a soma dos termos da série de potências de x para a função
% exponencial até encontrar um termo cujo valor absoluto
% é inferior a uma tolerância tol.
% n é o grau do último termo somado.

termo=1;
soma=0;
n=0;
while abs(termo)> tol
    soma = soma + termo;
    % [n termo soma], pause
    n=n+1;
    termo = termo*x/n;
end
n=n-1;
```

nota 1: sendo $\frac{x^n}{n!}$ o termo geral da série, cada termo é obtido do termo anterior multiplicando-o por $\frac{x}{n}$, evitando o cálculo das potências de x e dos fatoriais.

nota 2: o primeiro termo que falhar a condição $abs(termo) > tol$ já não é adicionado.

exercício 9.b)

```
>> [soma, n]=expTaylor(-1, 1e-5)

soma =

    0.3679


n =

     8

>> abs(soma-exp(-1))

ans =

   2.5033e-06
```

nota: observe-se que o erro de truncatura é inferior ao valor absoluto do primeiro termo desprezado

```
>> 1/factorial(9)

ans =

   2.7557e-06
```

por se tratar de uma série alternada.

exercício 9.c)
```
>>  [soma, n]=expTaylor(-100, 1e-5)

soma =

  -2.9138e+25


n =

   279
```
Com
```
>> exp(-100)

ans =

   3.7201e-44
```
neste caso não se verifica $|soma - \exp(-100)| < tol$. (atente-se que o valor da soma é $O(10^{25})$, muito grande, e o valor de $exp(-100)$ é $O(10^{-44})$, muito próximo de zero.

exercício 9.d)
```
>> [soma, n]=expTaylor(100, 1e-5)

soma =

   2.6881e+43


n =

   279

>> 1/soma

ans =

   3.7201e-44
```
Este valor está correto enquanto que o valor $-2.9138e+25$ da soma calculado em 9.c) para $x = -100$ está errado. O problema é o cancelamento subtrativo que ocorre na soma dos termos da série neste último caso. Com efeito, o valor correto é, como se disse antes, $O(10^{-44})$, muitas ordens de grandeza inferior às dos termos que estão a ser adicionados (por exemplo, $100^{100}/100!$ é $O(10^{42})$.

exercício 10.a) Com $x \neq k\pi$,

$$\frac{1-cos(x)}{sin(x)} = \frac{1-cos(x)}{sin(x)}.\frac{1+cos(x)}{1+cos(x)} = \frac{1-cos^2(x)}{sin(x)(1+cos(x))} = \frac{sin(x)}{1+cos(x)}$$

exercício 10.b)
```
>> F1=inline('(1-cos(x))/sin(x)')

F1 =
```

```
     Inline function:
     F1(x) = (1-cos(x))/sin(x)

>> F2=inline('sin(x)/(1+cos(x))')

F2 =

     Inline function:
     F2(x) = sin(x)/(1+cos(x))

>> x=pi*1e-8; F1(x), F2(x)

ans =

   1.4136e-08


ans =

   1.5708e-08

>> x=pi*1e-9; F1(x), F2(x)

ans =

     0


ans =

   1.5708e-09
```

Os valores corretos são dados pela expressão F2. Para valores próximos de 0, é cos(x) próximo de 1 e ocorre cancelamento subtrativo no cálculo do numerador de F1. O cancelamento subtrativo é tanto mais grave quanto mais próximo x estiver de zero.

exercício 12.a)

$$\sqrt{x+1}-\sqrt{x}=\left(\sqrt{x+1}-\sqrt{x}\right)\frac{\sqrt{x+1}+\sqrt{x}}{\sqrt{x+1}+\sqrt{x}}=\frac{1}{\sqrt{x+1}+\sqrt{x}}$$

exercício 12.b) O número de condição relativo de uma função f num ponto x onde $f(x)\neq 0$ é dado por $x.\frac{f'(x)}{f(x)}$ (uma vez que os erros são usualmente tomados em valor absoluto também se pode definir o número de condição em termos do valor absoluto da expressão anterior). Com

$$f(x)=\sqrt{x+1}-\sqrt{x}$$

tem-se

$$f'(x)=\frac{1}{2\sqrt{x+1}}-\frac{1}{2\sqrt{x}}$$

e o número de condição é, neste caso,

$$\frac{x}{2}\frac{\frac{1}{\sqrt{x+1}}-\frac{1}{\sqrt{x}}}{\sqrt{x+1}-\sqrt{x}}=\frac{x}{2}.\frac{\frac{\sqrt{x}-\sqrt{x+1}}{\sqrt{x(x+1)}}}{\sqrt{x+1}-\sqrt{x}}=\frac{x}{2}.\frac{\sqrt{x}-\sqrt{x+1}}{\sqrt{x(x+1)}}.\frac{1}{\sqrt{x}-\sqrt{x+1}}=\frac{x}{2}.\frac{-1}{\sqrt{x(x+1)}}$$

Uma vez que as expressões de f e g são equivalentes, conclui-se de imediato que os números de condição das funções são iguais.

exercício 12.c) No Matlab, usaremos format long para melhor comparar os valores produzidos pelas expressões que definem f e g.

```
>> f=inline('sqrt(x+1)-sqrt(x)')

f =

     Inline function:
     f(x) = sqrt(x+1)-sqrt(x)

>> g=inline('1/(sqrt(x+1)+sqrt(x))')

g =

     Inline function:
     g(x) = 1/(sqrt(x+1)+sqrt(x))

>> format long, x=10; f(x), g(x)

ans =

   0.154347130187020


ans =

   0.154347130187021

>> x=1e7; f(x), g(x)

ans =

     1.581138790243131e-04


ans =

     1.581138790555721e-04

>> x=1e11; f(x), g(x)

ans =

     1.581152901053429e-06


ans =

     1.581138830080237e-06
```

```
>> x=1e16; f(x), g(x)

ans =

     0


ans =

     5.000000000000000e-09
```

Para $x = 10$ os valores de $f(x)$ e $g(x)$ são praticamente iguais (coincidem em todos os algarismos significativos com excepção do último). Para os restantes valores, tal não é verdade e o problema agrava-se à medida que x aumenta. Isto deve-se à perda de algarismos significativos no cálculo de $g(x)$ para valores de x grandes. Isto acontece porque $\sqrt{x}$ e $\sqrt{x+1}$ se aproximam à medida que x aumenta. No caso extremo de ser $x = 10^6$, tem-se no Matlab,

```
> sqrt(1+1e16)==sqrt(1e16)

ans =

     1
```

e o cancelamento subtrativo é total.

Resolução explicada dos exercícios 6 e 7 da folha 1 (tratados nas aulas PL dos dias 27, 28, 29 e 30 de outubro

exercício 6.a) O seguinte código faz o que é pedido

```
% exercício 6.a da folha 1
k=1;
while 1+2^-k>1
    k=k+1;
end
k=k-1
```

Guardado num ficheiro executável do Matlab, por exemplo **epsilon.m**, tem-se

```
>> epsilon

k =

    52
```

Explicação: no formato duplo da norma IEEE 754, a representação normalizada de um número é a seguinte

$$\pm (1.b_{-1}b_{-2}\cdots b_{-52})_2 \times 2^e$$

onde $b_i = 0$ ou $b_i = 1$, para cada $i = 1, \cdots, 52$, e $-1022 \leq e \leq 1023$. Denotamos por $\mathcal{F}$ o conjunto destes números. Os números 1 e 2^{-52} têm as representações normalizadas (só diferem nos expoentes)

$$+ (1.00\cdots 00)_2 \times 2^0$$

e

$$+ (1.00\cdots 00)_2 \times 2^{-52}$$

Para efeitos da adição, o número de menor expoente, isto é, o número 2^{-52}, terá de ser desnormalizado por forma a ficar com o mesmo expoente, neste caso 0. A representação obtida neste processo é então

$$+ (0.00\cdots 01)_2 \times 2^0$$

resultando que a soma $1 + 2^{-52}$ pertence a $\mathcal{F}$ uma vez que tem a representação

$$+ (1.00\cdots 01)_2 \times 2^0$$

Portanto, no Matlab, a execução de

```
>> 1+2^-52
```

produz o valor lógico 1. Já o mesmo não acontece com k=53, isto é,

```
>> 1+2^-53
```

produz o valor lógico 0. Porquê? A representação normalizada de 2$^-$53 é

$$+(1.00\cdots 00)_2 \times 2^{-53}$$

que, para efeitos da soma com 1, terá de ser desnormalizada para

$$+(0.00\cdots 00|1)_2 \times 2^0$$

O bit 1 está agora na posição 53 à direita do ponto, isto é, não "cabe na caixa"dos 52 bits reservados para a mantissa no formato duplo da norma IEEE 754. Por outras palavras, o número $1+2^{-53}$ não pertence a $\mathcal{F}$ e terá de ser arredondado. Do que se disse até agora, deverá estar claro que $1+2^{-52}$ é o sucessor de 1 em $\mathcal{F}$, portanto $1+2^{-53}$ será arredondado para 1 ou para $1+2^{-52}$. O arredondamento usual no Matlab (isto é, aquele que é implementado pelo sistema se o utilizador não o alterar) é o arredondamento "para o mais próximo". Mas $1+2^{-53}$ está à mesma distância, igual a 2^{-53}, de 1 e de $1+2^{-52}$ e por esta razão terá de ser usada a "regra de desempate"implementada na norma IEEE. Esta regra determina que o arredondamento é feito para o número que tem o bit na última posição igual a 0, que neste caso é o número 1. Confirmando no Matlab

```
>> 1+2^-53==1

ans =

     1
```

exercício 6.b)

```
>> 2^-52==eps

ans =

     1
```

<u>Explicação:</u> no Matlab, **eps** (abreviatura de epsilon) é a constante 2$^-$52 que é valor de um bit igual a 1 na última posição da mantissa, no formato duplo da norma IEEE 754. É também a distância entre os números de $\mathcal{F}$ que têm expoente zero e os respetivos sucessores. Mas a importância desta constante resulta do facto de se ter, qualquer que seja x não inferior a 2^{-1022},

$$\left|\frac{x-fl(x)}{x}\right| < eps$$

isto é, o erro relativo devido ao arredondamento é inferior a eps. No caso do arredondamento para o mais próximo, podemos melhorar o majorante deste erro e escrever

$$\left|\frac{x-fl(x)}{x}\right| \leq \frac{eps}{2}.$$

exercício 7) Para x entre 15 e o respetivo sucessor tem-se

$$|x - fl(x)| \leq 2^{-50}$$

Explicação: uma vez que

$$15 = 2^3 + 2^2 + 2^1 + 2^0$$

tem a representação

$$+(1.1110 \cdots 00)_2 \times 2^3$$

e o seu sucessor tem a representação (adicione-se uma unidade no último bit da mantissa)

$$+(1.1110 \cdots 01)_2 \times 2^3$$

que é o número $15 + 2^{-52} * 2^3$ ou seja, $15 + 2^{-49}$. No caso do arredondamento para o mais próximo, o erro absoluto $|x - fl(x)|$ não é superior a metade da amplitude 2^{-49} do intervalo $[15, 15 + 2^{-49}]$ e será igual a 2^{-50} se x for o ponto médio daquele intervalo.

Resolução explicada dos exercícios 1, 2 e 3 da folha 2 (tratados nas aulas PL dos dias 3, 4, 5 e 6 de novembro

exercício 1 No Matlab, tem-se

```
>> format long, pi

ans =

   3.141592653589793
```

Com 3 algarismos significativos corretos, é $\pi = \mathbf{3.14}$ e com 5 algarismos significatvos corretos é $\pi = \mathbf{3.1416}$ (ver p. 25 das notas das aulas: o o último dos algarismos pedidos deve ser arredondado, acrescentando-lhe uma unidade se o primeiro algarismo que se despreza é igual ou maior do que 5).

```
>> 1/11

ans =

   0.090909090909091
```

O primeiro algarismo significativo é, por definição, maior do que zero. Assim, as aproximações com 3 e 5 algarismos significativos corretos são, neste caso, **0.0909** e **0.090909**.

```
>> log(5)

ans =

   1.609437912434100
```

(nota: no Matlab, log(x) é o logaritmo natural (de base e) de x; log10 e log2 denotam os logaritmos de base 10 e 2, respetivamente). As aproximações neste caso são **1.61** e **1.6094**.

exercício 2a) Numa série alternada convergente, o valor absoluto do erro de truncatura é inferior ao valor absoluto do primeiro termo que se despreza. Por exemplo, denotando por S a soma da série dada (trata-se da série harmónica alternada) e $S_3 = 1 - 1/2 + 1/3$ a soma dos primeiros 3 termos, tem-se

$$|S - S_3| < 1/4.$$

Analogamente, $S_4 = 1 - 1/2 + 1/3 - 1/4$ aproxima o valor de S com erro de truncatura (em valor absoluto) inferior a 1/5, etc.

Portanto, para a soma dos primeiros 999 termos

$$S_{999} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \ldots + \frac{1}{999}$$

tem-se

$$|S - S_{999}| < 0.001$$

uma vez que o primeiro termos que se despreza é $-1/1000$.

Para calcular o valor de S_{999} no Matlab podemos executar

```
>> s999=0; for k=1:999, s999=s999+(-1)^(k+1)/k; end, s999

s999 =

   0.693647430559822
```

Nota final: a soma da série harmónica alternada é conhecida, é igual a log(2); sabendo isto, podemos agora confirmar que o valor calculado de s999 aproxima o vale da soma da série com erro de truncatura inferior a 0.001:

```
>> abs(s999-log(2))

ans =

     5.002499998769672e-04
```

exercício 2b) Para

$$S_{10} = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \ldots - \frac{1}{2^9}$$

tem-se

$$|S - S_{10}| < \frac{1}{2^{10}} < 0.001.$$

Calculamos a seguir o valor da aproximação.

```
>> s10=0; for k=0:9, s10=s10+(-1)^k/(2^k); end, s10


s10 =

   0.666015625000000
```

Nota 1: também neste caso o valor da soma da série é conhecido, $S = 2/3$, por se tratar da série geométrica cujo primeiro termo é 1 e a razão é -1/2. Em geral, a série geométrica

$$a_1 + a_1.r + a_1.r^2 + \ldots$$

(cada termo é obtido do anterior multiplicando pela razão r) é convergente se e só se $|r| < 1$ e, neste caso, a soma é

$$S = \frac{a_1}{1 - r}.$$

Tal como na alínea a), podemos agora confirmar que o erro de truncatura é inferior a 0.001:

```
>> abs(s10-2/3)

ans =

     6.510416666666297e-04
```

Nota 2: embora ambas as séries tratadas sejam convergentes, a série geométrica de razão -1/2 converge muito mais rapidamente do que a série harmónica alternada. Se se pretender garantir um erro de truncatura inferior a 10^{-9}, teremos de somar os primeiros 10^9-1 termos da série harmónica alternada (para a série geométrica de razão -1/2, bastam os primeiros 29 primeiros termos). A soma de um elevado número de termos requer um tempo de computação maior, obviamente. Ponha à prova a performance da sua máquina, executando no Matlab

```
>> n=10^9; tic, soma=0; for k=1:n, soma=soma+(-1)^(k+1)/k; end, soma, toc
```

exercício 3) Vamos usar o resto de ordem 8 do desenvolvimento da função *sin* em série de potências de x (usamos o resto de ordem 8 porque, neste caso, o polinómio de ordem 8 coincide com o polinómio de ordem 7). Tem-se (ver p. 41 das notas das aulas)

$$sin(x) = p_7(x) + R_8(x)$$

onde

$$R_8(x) = \frac{cos(\theta)}{9!}\left(\frac{\pi}{4}\right)^9$$

e θ é um ponto (não determinado) que está entre 0 e $\frac{\pi}{4}$ (a derivada de ordem 9 da função *sin* é a função *cos*). Uma vez que $|cos(\theta)| < 1$ resulta

$$|p_7(\frac{\pi}{4}) - sin(\frac{\pi}{4})| < \frac{(\frac{\pi}{4})^9}{9!}.$$

nota 1: Porque a série é alternada, também neste caso se pode usar o valor do primeiro termo desprezado para majorar o erro, ou seja

$$|p_7(\frac{\pi}{4}) - sin(\frac{\pi}{4})| < \frac{(\frac{\pi}{4})^9}{9!}$$

que é afinal o mesmo majorante a que chegámos usando o resto de ordem 8.

nota 2: comparemos o majorante com o erro efetivamente cometido. O majorante:

```
>> (pi/4)^9/factorial(9)

ans =

     3.133616890378120e-07.
```

O erro de truncatura:

```
>> x=pi/4; p7=x-x^3/factorial(3)+x^5/factorial(5)-x^7/factorial(7); abs(p7-sin(pi/4))

ans =

     3.116113693746314e-07
```