Resolução explicada dos exercícios 1, 2 e 5 da folha 5 (resolvidos nas aulas PL dos dias 5, 6, 7 e 8 de janeiro)

exercício 1.a) A regra do ponto médio consiste em substituir a função integranda f pelo polinómio de grau zero (isto é, constante) que coincide com f no ponto médio $\frac{a+b}{2}$, ou seja, esta regra aproxima

$$\int_a^b f(x)dx$$

por

$$(b-a)f\left(\frac{a+b}{2}\right).$$

No caso presente dá

$$(2-0)(3\times 1+2)=10$$

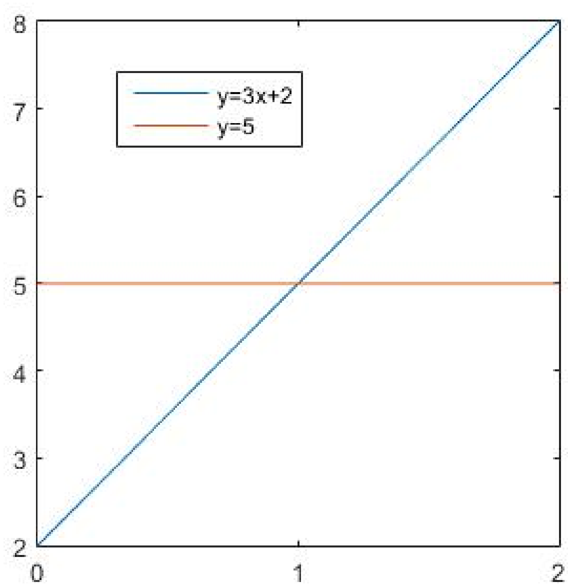
que é o valor exato do integral:

$$\int_0^2 (3x+2)dx = \left[\frac{3}{2}x^2+2x\right]_0^2 = 10.$$

exercício 1.b) No Matlab,

```
fplot('[3*x+2 5]',[0,2])
```

produz os gráficos da função e do polinómio (neste caso, a reta horizontal)



A área da figura limitada superiormente pela reta $y=3x+2$ é igual à área da figura (retângulo) limitada superiormente pela reta $y=5$.

exercício 1.c)

$$\begin{aligned}
\int_a^b (mx+r)dx &= \left[\frac{m}{2}x^2 + rx\right]_a^b \\
&= \frac{m}{2}b^2 + rb - \frac{m}{2}a^2 - ra \\
&= \frac{m}{2}\left(b^2 - a^2\right) + r(b-a) \\
&= (b-a)\left(m\frac{a+b}{2} + r\right)
\end{aligned}$$

exercício 2.a) A regra (simples) dos trapézios neste caso dá

$$\int_0^1 exp(-x)dx \approx \frac{1-0}{2}\left[exp(-0) + exp(-1)\right]$$

e

```
 (1+exp(-1))/2

ans =

    0.6839
```

O resultado exato é

$$\left[-exp(-x)\right]_0^1 = -exp(-1) + exp(0)$$

e

```
>> 1-exp(-1)

ans =

    0.6321
```

exercício 2.b) nota prévia: na alínea anterior usou-se a regra simples dos trapézios para aproximar o valor do integral. Para reduzir o erro de truncatura há que usar a regra composta que consiste em dividir o intervalo de integração em n subintervalos de igual amplitude $h = (b-a)/n$ e usar a mesma regra em cada um destes subintervalos. É o que se ilustra nesta alínea (com $n = 2$) e na próxima, onde $n = 4$. Tem-se

$$\int_0^{1/2} exp(-x)dx \approx \frac{1/2-0}{2}\left[exp(-0) + exp(-1/2)\right],$$

e

$$\int_{1/2}^1 exp(-x)dx \approx \frac{1-1/2}{2}\left[exp(-1/2) + exp(-1)\right],$$

ou seja

$$\int_0^1 exp(-x)dx \approx \frac{h}{2}\left[exp(-0) + 2 \times exp(-1/2) + exp(-1)\right],$$

onde $h = (1-0)/2 = 1/2$. No Matlab, tem-se

```
>> (exp(0)+2*exp(-1/2)+exp(-1))/4

ans =

    0.6452
```

que, como era de esperar, tem menor erro de truncatura do que a aproximação calculada antes.

exercício 2.c) Agora, tem-se, com $n = 4$ e $h = 1/4$ e procedendo tal como antes, obtemos

$$\int_0^1 exp(-x)dx \approx \frac{h}{2}\left[exp(-0) + 2 \times exp(-1/4) + 2 \times exp(-1/2) + 2 \times exp(-3/4) + exp(-1)\right].$$

Para fazer o cáculo da aproximação vamos usar a função trapezios (disponível na BB, no local do costume)

```
>> T=trapezios(@(x)exp(-x),0,1,4)

T =

    0.6354
```

Com era de esperar, esta é a melhor de todas as aproximações calculadas.

Nota adicional: o erro de truncatura da fórmula composta dos trapézios é dado por

$$-\frac{h^2}{12}(b-a)f''(\eta),$$

onde η é um ponto que não se conhece mas sabe-se que está no intervalo de integração. Uma vez que neste caso é $f''(x) = exp(-x)$, o maior valor que $|f''(\eta)|$ pode tomar entre 0 e 1, é $exp(-0) = 1$ e o erro de truncatura, em valor absoluto, é majorado por $h^2/12$. Portanto, quando n aumenta, o erro de truncatura tende para zero.

exercício 5.a) Vamos usar o format short no Matlab para podermos apreciar um maior número de algarismos das aproximações calculadas.

```
>> format long; for k=1:8, T(k)=trapezios(@(x)4./(1+x.^2),0,1,2^k); end, T'

ans =

   3.100000000000000
   3.131176470588235
   3.138988494491089
   3.140941612041389
   3.141429893174975
   3.141551963485654
   3.141582481063752
   3.141590110458282

>> pi

ans =

   3.141592653589793
```

exercício 5.b)

```
>> format long; for k=1:8, S(k)=simpson(@(x)4./(1+x.^2),0,1,2^k); end, S'

ans =

   3.133333333333333
   3.141568627450980
   3.141592502458707
   3.141592651224822
   3.141592653552837
   3.141592653589216
   3.141592653589783
   3.141592653589794
```

Os erros das paroximações calculadas são

```
>> [T'-pi S'-pi]

ans =

  -0.041592653589793  -0.008259320256460
  -0.010416183001558  -0.000024026138813
  -0.002604159098704  -0.000000151131086
  -0.000651041548404  -0.000000002364971
  -0.000162760414818  -0.000000000036957
  -0.000040690104139  -0.000000000000577
  -0.000010172526041  -0.000000000000010
  -0.000002543131512   0.000000000000000
```

o que mostra que as aproximações calculadas com a regra de Simpson são, para o mesmo valor de n em ambas as regras, bastante melhores do que as aproximações calculadas com a regra dos trapézios.

nota importante: os códigos trapezios e simpson requerem que as funções integrandas estejam definidas de modo a aceitar vetores e não apenas um número de cada vez; por esta razão, a função integranda foi definida pela expressão

```
4./(1+x.^2)
```

com um ponto depois do numerador 4 e outro ponto em

```
x.^2
```

Sem isto, o Matlab produzirá um erro. Veja-se, por exemplo,

```
>> format long; for k=1:8, S(k)=simpson(@(x)4./(1+x^2),0,1,2^k); end, S'
Error using  ^
Inputs must be a scalar and a square matrix.
To compute elementwise POWER, use POWER (.^) instead.

Error in @(x)4./(1+x^2)


Error in simpson (line 12)
Q=Q+2*sum(f(x));
```