Instituto Superior Técnico

Matemática Computacional

2023/2024

Relatório:
Projeto II -Método de Romberg

Francisco Augusto 99218 - francisco.augusto@tecnico.ulisboa.pt
Mariana Andrade 106814 -mariana.mendonca.andrade@tecnico.ulisboa.pt
Beatriz Dinis 103941- beatrizdinis@tecnico.ulisboa.pt
Gonçalo Moniz 99469 - goncalomoniz@tecnico.ulisboa.pt

Professora Margarida Baía

10/01/24

Conteúdo

1 Introdução **2**
1.1 Objetivos 2

2 Fundamentos teóricos **2**
2.1 Método de Romberg 2
2.1.1 Regra dos Trapézios 2
2.1.2 Erro de Integração na Regra dos Trapézios 4
2.1.3 Método de Romberg 4

3 Adaptação a Matlab **5**
3.1 Implementação do método de Romberg 5
3.2 Aproximação de P(m) com $\sigma = 1$ e $n_{max} = 5$ 6
3.3 Variação do erro da apromixação de $P(1)$ consoante os valores de n e k 7
3.4 Análise da precisão do Método de Romberg 8
3.5 Análise do efeito da variação de σ 9
3.6 Aproximação de M(1) = $\int_0^\infty tD_\sigma(t)dt$ 10

4 Conclusão **12**

5 Referências **12**

1 Introdução

O segundo projeto, realizado no âmbito da cadeira de matemática computacional, através de pesquisa e análise de conceitos relativos à temática de integração numérica, tem como objetivo a expansão e aprofundamento de conhecimentos nesta área. Foi-nos mais uma vez, à semelhança do primeiro projeto, proporcinada a oportunidade de utilizar a ferramenta $Matlab$, de extrema importância nas áreas de engenharia, de forma a ser possibilitada a resolução dos problemas enumerados no enunciado. Neste caso será através da exploração do método de integração de Romberg, que passaremos a explicar, que podemos tornar esta questão relativa a probabilidades, num problema de simples análise numérica.

1.1 Objetivos

O problema em questão relaciona um intervalo de tempo, em minutos, que separa a chegada de dois indivíduos a uma dada paragem de autocarros, e descreve-o como uma densidade de probablidade, função que refere-se à medida de quão provável é que uma variável aleatória assuma um determinado valor dentro de um intervalo específico. Resumindo, a densidade de probabilidade descreve como a probabilidade está distribuída ao longo do conjunto de possíveis valores de uma variável aleatória contínua. Desta forma temos como função de densidade de probabilidade, para o problema

$$\begin{cases} 0, t \leq 0 \\ \frac{2\sqrt{\sigma}}{\sqrt{\pi}} e^{-\sigma t^2}, t > 0 \end{cases} \tag{1}$$

onde $\sigma > 0$, é a nossa variável aleatória contínua, dependente da localização da paragem e do horário considerado. Já a função:

$$P[t \leq m] = \int_0^m D_\sigma(t) dt \tag{2}$$

descreve a probalidadede de o intervalo de tempo entre dois indivíduos seja menor ou igual a m minutos.

Será assim através do método de integração Romberg, que utiliza o a regra dos trapézios lecionada em aula, para subintervalos de diversos tamanhos, que é a possibilitada a resolução do problema proposto pelo enunciado deste projeto. Há assim o objetivo de, pelo método já referido, aproximar $P(m) = P[t \leq m]$.

2 Fundamentos teóricos

2.1 Método de Romberg

Seja $f \in C\,[a, b]$, consideremos o integral

$$I(f) = \int_a^b f(x) dx \tag{3}$$

Ora, caso $f(x)$ seja uma função demasiado complexa cuja primitiva se desconheça ou a função f seja somente conhecida num número finito de pontos, torna-se necessário recorrer a uma aproximação de $If(x)$. Existem vários métodos de aproximação de um integral, no entanto, será apenas explicitado o Método de Romberg e os conhecimentos teóricos necessários para a sua compreensão.

2.1.1 Regra dos Trapézios

O método de Romberg faz um uso da regra dos trapézios, aplicando-a a subintervalos de tamanhos diferentes. Por sua vez, a Fórmula (ou Regra) dos Trapézios resulta de aproximar $f(x)$ pelo polinómio interpolador em $x_0 = a$ e $x_1 = b$.

Comecemos por concluir que se $f \in C^{(n+1)}\,[a, b]$, então

$$f(x) = p_n(x) + e_n(x) \tag{4}$$

onde e_n é o erro de interpolação. Logo,

$$\int_a^b f(x)dx = \int_a^b p_n(x)dx + \int_a^b e_n(x)dx \tag{5}$$

Desprezando $\int_a^b e_n(x)dx$ obtemos a aproximação

$$\int_a^b f(x)dx \approx \int_a^b p_n(x)dx \tag{6}$$

Para obtermos um expressão para $\int_a^b p_n(x)dx$, utilizamos a fórmula de Lagrange do polinómio interpolador, resultando em

$$\int_a^b p_n(x)dx = \int_a^b \Big(\sum_{k=0}^{n} l_k(x)f(x_k)\Big)dx = \sum_{k=0}^{n}\int_a^b l_k(x)f(x_k)dx = \sum_{k=0}^{n}\Big(\int_a^b l_k(x)dx\Big)f(x_k) \tag{7}$$

Ora, considerando

$$A_k = \int_a^b l_k(x)dx \tag{8}$$

e substituindo na expressão 7, obtemos

$$\int_a^b p_n(x)dx = \sum_{k=0}^{n} A_k f(x_k) \tag{9}$$

Por sua vez, substutindo na equação 6, obtemos a aproximação

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} A_k f(x_k) \tag{10}$$

onde o termo do lado direito é denominado de fórmula de quadratura.

Na Regra dos Trapézios a fórmula da quadratura denomina-se $T(f)$ e é calculada através da aproximação de $f(x)$ pelo polinómio interpolador em x_0 e x_1, como referido. Assim, considerando n=1, obtemos

$$\int_a^b f(x)dx \simeq A_0 f(x_0) + A_1 f(x_1) \tag{11}$$

onde $x_0 = a$, $x_1 = b$ e $h = x_1 - x_0$.

De seguida, procedemos ao cálculo de A_0 e A_1, que será simplificado por razões de praticalidade

$$A_0 = \int_a^b l_0 dx = \int_a^b \frac{(x - x_1)}{(x_0 - x_1)}dx = \frac{h}{2} \tag{12}$$

$$A_1 = \int_a^b l_1 dx = \int_a^b \frac{(x - x_0)}{(x_1 - x_0)}dx = \frac{h}{2} \tag{13}$$

Assim, a fórmula da quadratura é dada por

$$T(f) = \frac{h}{2}f(x_0) + \frac{h}{2}f(x_1) = \frac{h}{2}\big[f(x_0) + f(x_1)\big] \tag{14}$$

2.1.2 Erro de Integração na Regra dos Trapézios

Sabe-se que

$$I(f) = T(f) + E(f) \tag{15}$$

sendo $E(f)$ o erro de integração, logo deduz-se

$$E(f) = I(f) - T(f) \tag{16}$$

Para obtermos a expressãoo do erro cometido, temos que relembrar a equação 4, desenvolvendo-a de modo a que

$$f(x) = p_n(x) + (x - x_0)(x - x_1)...(x - x_n)\frac{f^{(n+1)}(\xi(x))}{(n+1)!} \tag{17}$$

onde $\xi(x) \in [a.b]$.

Ora, adequando a equação 17 às circunstâncias nas quais se deduziu a fórmula de quadraduta, obtemo

$$f(x) = p_1(x) + (x - x_0)(x - x_1)\frac{f''(\xi(x))}{2!} \tag{18}$$

logo,

$$\int_a^b f(x)dx = \int_a^b p_1(x)dx + \int_a^b (x - x_0)(x - x_1)\frac{f''(\xi(x))}{2!}dx \tag{19}$$

$$\Longrightarrow \int_a^b f(x)dx - \int_a^b p_1(x)dx = \int_a^b (x - x_0)(x - x_1)\frac{f''(\xi(x))}{2!}dx$$

$$\Longrightarrow I(f) - T(f) = \int_a^b (x - x_0)(x - x_1)\frac{f''(\xi(x))}{2!}dx$$

Concluimos assim que o erro é dado por:

$$E(f) = \int_a^b (x - x_0)(x - x_1)\frac{f''(\xi(x))}{2!}dx \tag{20}$$

Aplicando o Teorema de Valor Médio de Integrais, obtemos

$$E(f) = -\frac{h^3}{12}f''(z) \tag{21}$$

com $z \in [a, b]$.

Como z é desconhecido não sabemos como avaliar a segunda derivada, a expressão 21 é geralmente reescrita como

$$E(f) = O(h^3 f'')$$

onde o símbolo O se refere à ordem de grandeza de um termo numa equação. No caso dos métodos de integração, quanto mais elevada a sua ordem mais eficiente este se mostra.

2.1.3 Método de Romberg

O Método de Romberg resulta na criação de uma tabela, como veremos no final desta subseccção, cujo último termo é a melhor aproximação de $I(f)$.

Começamos por definir $T(h)$ como o resultado da aplicação da Regra dos Trapézios como espaçamento $h > 0$. Seguidamente, definimos por recorrência uma sucessão de aproximações de $I(f)$ através das seguintes expressões

$$T_0(h) = T(h) \tag{22}$$

$$T_n(h) = \frac{4^n T_{n-1}(h/2) - T_{n-1}(h)}{4^n - 1} \tag{23}$$

com n=1, 2, ...

Consideremos agora que, para cada $k = 0, 1, 2, ..$

$$h_k = \frac{b - a}{2^k} \tag{24}$$

e

$$T_0^{(k)} = T_0(h_k) \tag{25}$$

podemos criar um algoritmo que, dado um valor máximo $n_{max} \in \mathbb{N}$, permite aproximar $I(f)$. Os valores resultantes desse algoritmo será organizados numa tabela, cuja primeira coluna equivale à aplicação direta da Fórmula dos Trapézios, logo

$$T_0^{(k)} = T_0(h_k) \tag{26}$$

com $k = 0, ..., n_{max}$.

Assim, para $n = 1, ..., n_{max}$, resulta

$$T_n^{k-n} = \frac{4^n T_{n-1}^{(k-n+1)} - T_{n-1}^{(k-n)}}{4^n - 1} \tag{27}$$

Dispondo os valores $[T_n^{(k)}]$ numa tabela, obtemos

$T_0^{(0)}$	-	....	-	-
$T_0^{(1)}$	$T_1^{(0)}$	....	-	-
....	....	....	....	....
$T_0^{(n_{max}-1)}$	$T_1^{(n_{max}-2)}$	....	$T_{n_{max}-1}^{(0)}$	-
$T_0^{(n_{max})}$	$T_1^{(n_{max}-1)}$	....	$T_{n_{max}-1}^{(1)}$	$T_{n_{max}}^{(0)}$

Tabela 1: Tabela do Método de Romberg

no qual $T_{n_{max}}^{(0)}$ corresponde à melhor aproximação de $I(f)$.

Ora, aplicando a expressão 16 ao caso atual obtemos

$$E_n(h) = I - T_n(h) \tag{28}$$

Assim como os termos evoluiem a cada coluna, também a grandeza do erro evolui sob a seguinte regra

$$E_n(h) = O(h^{2(n+1)}) \tag{29}$$

tendo em conta que $E_0(h) = I - T_0(h) = O(h^2)$.

3 Adaptação a Matlab

3.1 Implementação do método de Romberg

Na primeira pergunta deste projeto foi-nos pedido para implementar o Metodo de Romberg em $Matlab$, através de uma função que recebia a função f a integrar, os extremos do intervalo de integração e a profundidade máxima do Método, e devolve a aproximação mais precisa da inetgração de f no intervalo dado, utilizando o Método de Romberg.

```
function  [aprox,matrix] = romberg(func, a, b, max)
format long
    matrix = zeros(max+1, max+1);
    for n = 1:max+1
        X = a:((b-a)/2^(n-1)):b;
        Y  = func(X);
        matrix(n,1) = trapz(X, Y);
    end
    for n = 2:max+1
           for z = 2:n
            matrix(n,z) = ((4^z)*matrix(n,z-1 ) - matrix(n,z-1))/(4^z -1);
           end
    end
    aprox = matrix(max+1, max+1);
end
```

Figura 1: Função Romberg

A figura acima apresenta a nossa implementação do Método. Esta implementação começa através da criação da matriz(max+1, max+1), no código denominada de $matrix$ sendo max a profundidade do método pretendida pelo utilizador.

Após a criação da matriz, sucede-se a execução de um $loop$, através do qual calculamos todos os implementos da primeira coluna (aproximações calculadas diretamente através do Método dos Trapézios). Para tal, calcula-se um vetor X que vai conter todos os valores de a até b, com passo $h = \frac{b-a}{2^{n-1}}$, sendo n a linha da matriz. Usando o vetor X, calculam-se as respetivas imagens de cada valor do vetor X, e guardam -se os valores num vetor Y.

Com os vetores X e Y, executamos a função $trapz$, e obtemos o valor da aproximação do integral pelo método dos Trapézios com $n-1$ número de intervalos, sendo que esse valor fica guardado na respetiva posição da matriz, ou seja, coluna 0, linha $n-1$.

Tendo esses valores, prosseguimos para um segundo $loop$, no qual calculamos os valores das colunas seguintes. Visto ser necessário ter todos os valores de uma coluna antes de se calcular a seguinte, usamos um terceiro $loop$, que percorre todas as linhas da coluna que se encontra a ser calculada no momento, e usando os valores da coluna anterior, recorre à fórmula da equação 30, onde z e n correspondem, respetivamente, à coluna e linha que se pretendem calcular, para obter os novos valores:

$$matrix(n,z) = \frac{4^z matrix(n,z-1) - matrix(n-1,z-1)}{4^{z-1}} \tag{30}$$

Por fim, após termos calculado todos os valores da matriz, devolvemos o valor mais preciso, ou seja o da última coluna e última linha, que corresponde à posição na matriz matrix(max+1, max+1).

Dado a função Romberg precisar de receber uma função para funcionar, também codificamos a função de distribuição de probabilidades em Matlab, à qual chamamos tempo.

```
function density = tempo(sig,t)
format long
    if t <= 0
        density = 0;
    end
    density = (((2*sqrt(sig))/sqrt(pi))*exp(-sig.*(t.^2)));
end
```

Figura 2: Função Tempo

A função tempo recebe o valor de $sigma$, e o valor do tempo e retorna:

$$\begin{cases} 0, t \leq 0 \\ \frac{2\sqrt{\sigma}}{\sqrt{\pi}} e^{-\sigma t^2}, t > 0 \end{cases}$$

3.2 Aproximação de P(m) com $\sigma = 1$ e $n_{max} = 5$

Para obter os valores aproximados de $P(m)$, função relativa à probabilidade do intervalo de tempo entre duas pessoas ser menor ou igual a m minutos, vai ser utilizado o código apresentado na figura 3

```
P = 0:0.1:2;
n = 1;
for m = 0:0.1:2
  P(n) = romberg(@(x)tempo(1,x), 0, m, 5);
  n = n +1;
end

X = 0:0.1:2;

plot(X, P, '-o');
grid on

legend('P(m)','Location', 'SE');
```

Figura 3: Cálculo dos valores de $P(m)$

onde é utlilizada a função romberg, implementada no exercicio anterior. A função romberg recebe como input: a função tempo, em que é definida a variável aleatória $\sigma = 1$; os extremos para o valor de t, em que o 3º argumento será um dos valores de m, m = 0, 0.1, incrementando 0.1 até chegarmos a 2.0; e o 4º argumento o valor de $n_{max} = 5$. A função $P(m)$ vai repetir a utilização da função romberg para n = n + 1 iterações, até chegar ao valor de n_{max} Desta forma é obtido o gráfico a que corresponde o valor de $P(m)$ para cada m, como representado na figura 4

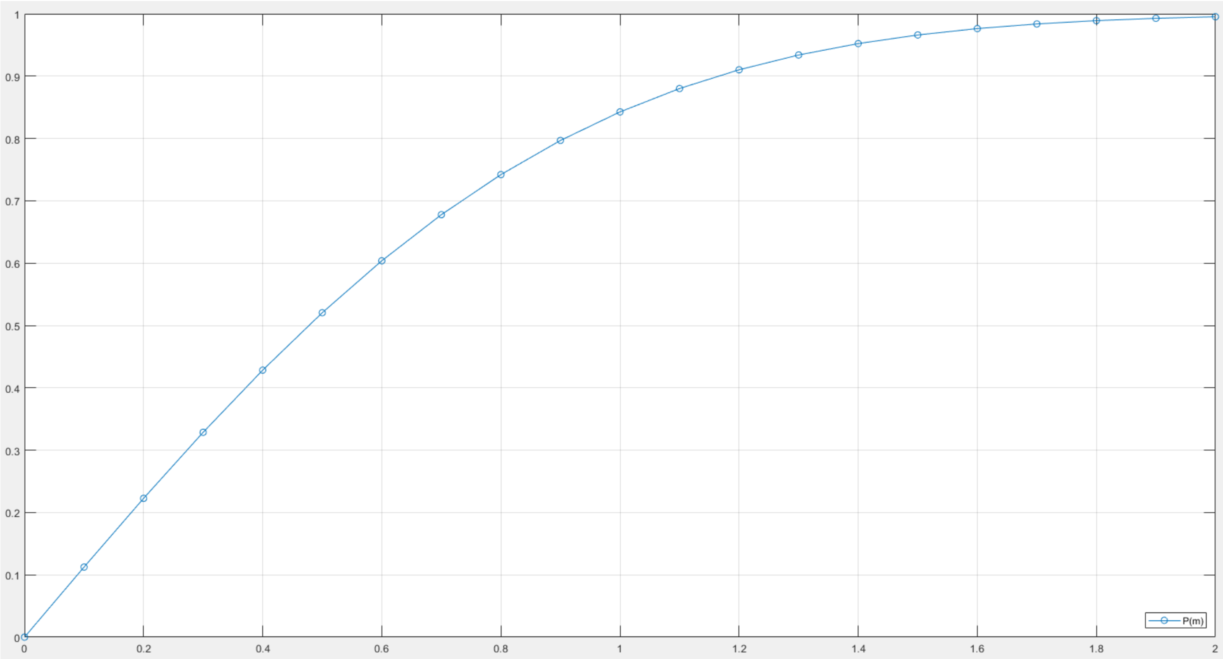

Figura 4: Valores de $P(m)$ para o valor de m correspondente

3.3 Variação do erro da apromixação de $P(1)$ consoante os valores de n e k

Para obter todos os valores dos erros absolutos de E_n^k recorremos ao $Matlab$ para calcular o valor do integral $\int_0^m D_\sigma(t)dt$, com $m = 1$ e assumimos que este corresponde ao valor exato de $P(1)$. Tendo agora o valor exato, I, calculamos o erro absoluto para cada n e k utilizando a seguinte fórmula:

$$E_n^{(k)} = I - T_n^{(k)} \tag{31}$$

```
format long
resultado_integral = quad(@(t) (2 * exp(-t.^2)) / sqrt(pi), 0,1);


fprintf('O valor da integral é: %.15f\n', resultado_integral);

a = 0;
m = 1;
max = 5;

[approx, matrix] = romberg(@(x)tempo(1,x), a, m, 5);;

fprintf('\nCalculando I - (T_n)^k para n e k até 5:\n');


matrixErro = zeros(max+1, max+1);

for n = 1:max+1
    for z = 1:max+1
        result = resultado_integral - matrix(n,z);
        matrixErro(n, z) = result;
    end
end


disp('Matriz dos resultados:');
disp(matrixErro);
```

Figura 5: Script utilizado para os cálculos do Exerc3

n=0	n=1	n=2	n=3	n=4	n=5
k=0 0.070957514756818	-	-	-	-	-
k=1 0.017437891418123	k=0 0.013869916528876	-	-	-	-
k=2 0.004333069573667	k=1 0.003459414784036	k=0 0.003294168724594	-	-	-
k=3 0.001081625770104	k=2 0.000864862849866	k=1 0.000823679485832	k=0 0.000813991292739	-	-
k=4 0.000270341524639	k=3 0.000216255908275	k=2 0.000205960559996	k=1 0.000203538132836	k=0 0.000202941404429	-
k=5 0.000067619333615	k=4 0.000054104520880	k=3 0.000051530689334	k=2 0.000050925081998	k=1 0.000050775900130	k=0 0.000050738741277

Tabela 2: Tabela dos erros absolutos E_n^k

Conseguimos observar que com o aumento de k, ou seja, com cada nova iteração o erro absoluto diminui, como esperado, o mesmo acontece com o aumento dos valores de n.

3.4 Análise da precisão do Método de Romberg

Utilizando os valores obtidos anteriormente de E_n^k, e aplicando a seguinte equação:

$$Q_n^{(k)} = \frac{E_n^{(k)}}{E_n^{(k-1)}} \tag{32}$$

Obtivemos a tabela 3:

n=0	n=1	n=2	n=3	n=4	n=5
k=1 0.245751157969459	-	-	-	-	-
k=2 0.248485867343094	k=1 0.249418572695357	-	-	-	-
k=3 0.249621140790647	k=2 0.250002646070960	k=1 0.250041681132625	-	-	-
k=4 0.249939981194240	k=3 0.250046476511861	k=2 0.250049398508400	k=1 0.25004952098580	-	-
k=5 0.250125590973474	k=4 0.250187480710115	k=3 0.250196879125793	k=2 0.25019921961764	k=1 0.25019980655433	-

Tabela 3: Tabela dos quocientes Q_n^k

Observa-se que os quocientes, Q_n^k, estão a convergir para 0.25 à medida que n e k aumentam, podemos assim confirmar que o método de Romberg está a funcionar como o esperado, visto que a convergência dos quocientes, para um valor constante, implica que este método tem de facto uma convergência consistente.

$$E_n(h) \leq C_n h^{\alpha n}$$

$$Q_n = \frac{C_n(\frac{h}{2})^{\alpha n}}{C_n h^{\alpha n}}$$

$$Q_n = \frac{\frac{h^{\alpha n}}{2^{\alpha n}}}{h^{\alpha n}}$$

$$Q_n = \frac{1}{2^{\alpha n}}$$

$$2^{\alpha n} * Q_n = 1$$

$$ln(2^{\alpha n} * Q_n) = ln(1)$$

$$\alpha n * ln(2) + ln(Q_n) = 0$$

$$\alpha = \frac{-ln(Q_n)}{n * ln(2)}$$

Calculando agora α para cada Q_n^k com exceção dos valores com $n = 0$.

n=0	n=1	n=2	n=3	n=4	n=5
-	-	-	-	-	-
-	k=1 2.00335919692972	-	-	-	-
-	k=2 1.99998473018700	k=1 0.999879743697886	-	-	-
-	k=3 1.99973181919473	k=2 0.999857480113884	k=1 0.666571417859140	-	-
-	k=4 1.99891849550881	k=3 0.999432150489222	k=2 0.666283601735280	k=1 0.49971185520604	-

Tabela 4: Tabela dos α

Observamos através da tabela que para $n = 1$, o α é aproximadamente 2 para todos os valores de k, desta forma concluimos que a convergência é quadrática neste caso. Para $n = 2$, α está muito próximo de 1, logo deduzimos que a convergência é linear, e por fim para $n = 3$ e $n = 4$ a convergência é sublinear, dado que α assume valores que indicam uma taxa de convergência mais moderada. É de notar que a convergência observada para $n = 1, 2, 3, 4$, não coincide com a teoria, visto que a taxa de convergência deveria ser respetivamente $O(h^4)$,$O(h^6)$,$O(h^8)$, $O(h^8)$.

3.5 Análise do efeito da variação de σ

Também se torna relevante analisar o efeito que a variação de σ terá no comportamento da função. Para tal desenvolvemos um script semelhante ao script $Exerc2$, denominado $Exerc5$:

```
%Script Exerc5 - usa um loop para registar os valores de P(m) consoante os
%diferentes valores de m 0<m<2, para sigma = 1, 2 e 3, de 0.1 em 0.1. No fim cria um gráfico que
%relaciona os valores de m com P(m) para cada conjunto de dados, permitindo ver a influência de sigma no valor de P
P1 = 0:0.1:2;
P2 = 0:0.1:2;
P3 = 0:0.1:2;
n = 1;
for m = 0:0.1:2
  P1(n) = romberg(@(x)tempo(1,x), 0, m, 5);
  P2(n) = romberg(@(x)tempo(2,x), 0, m, 5);
  P3(n) = romberg(@(x)tempo(3,x), 0, m, 5);
  n = n +1;
end

X = 0:0.1:2;

plot(X, P1, '-o');
hold on
plot(X, P2, '-*');
plot(X, P3, '-s');
hold off
grid on

legend('Sigma = 1', 'Sigma = 2', 'Sigma = 3', 'Location', 'SE')
```

Figura 6: Script $Exerc5$

Este script tem um funcionamento semelhante ao script $Exerc2$, calculando o valores da aproximação de Romberg nos quais o extremos superior varia de 0.1 a 2, com espaçamento 2. No entanto, este difere do anterior, no sentido em que também calcula os mesmos valores quando σ é 2 e 3. Com este $script$ cria-se também um gráfico que coloca as três retas lado a lado permitindo a sua comparação:

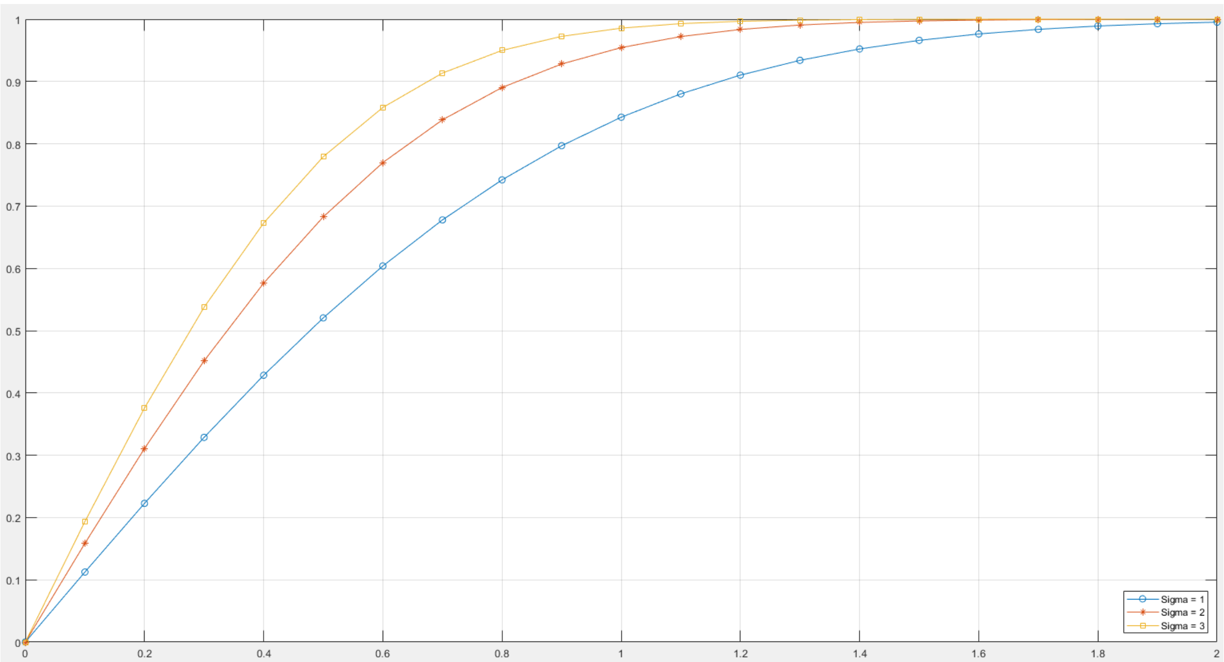

Figura 7: Efeito da variação de $sigma$

Por observação dos resultados, torna-se evidente que o aumento do valor de σ resulta numa convergência mais rápida a 1.

3.6 Aproximação de M(1) = $\int_0^\infty tD_\sigma(t)dt$

Na pergunta 6, é nos pedido para calcular o valor do seguinte integral:

$$M(\sigma) = \int_0^\infty tD_\sigma(t)dt \tag{33}$$

Primeiramente, começamos por criar um novo script para representar a variação na função a integrar.

```
function density = tempo_alt(sig,t)
format long
    if t <= 0
        density = 0;
    end
    density = (((2*sqrt(sig))/sqrt(pi))*exp(-sig.*(t.^2))).*t;
end
```

Figura 8: Função tempo_alt

Dado ser impossível computar o limite superior ∞, usamos um valor de limite elevado o suficiente para que qualquer aumento do mesmo não leve a variações significativas (designado t_{max}). Para encontrar tais valores, foi usado o seguinte script.

```
function tmax = t_max(func,max)
r_ant = romberg(func,0,0.01,max);
for T = 0.02:0.01:20
  r = romberg(func,0,T,max);
  if abs(r-r_ant)<= 10^-12
      t = T;
      break
  end
  r_ant = r;
end
tmax = t;
end
```

Figura 9: Função t_{max}

Desta forma é possível calcular os integrais sucessivos com o limite superior a variar de 0.01 a 20, com incrementos de 0.01. Após calcularmos um dado intervalo, verificamos se a diferença do integral para o integral anterior for inferior a 10^{-12}, paramos e devolvemos o valor do extremo superior encontrado.

Usando este script, podemos encontrar o t_{max} quando $\sigma = 1$, o que nos permite calcular o integral para esse mesmo valor.

$$M(1) = 0,564189583533131$$

De seguida criou-se o seguinte script para calcular todos os valores dos t_{max}, consoante os valores de σ.

```
%Script para o exercicio 6, na qual se procura obter todos os valores de
%tmax
function tmaxs = t_max_calculator()
n = 1;
tmaxs = 0.1:0.1:2;
inte = 0.1:0.1:2;
X = 0.1:0.1:2;
for sigs = 0.1:0.1:2
    tmaxs(n) = t_max(@(x)tempo_alt(sigs, x), 15);
    inte(n) = romberg(@(x)tempo_alt(sigs, x), 0, tmaxs(n), 15);
    n = n +1;
end
plot(X, inte, '-o');
grid on
end
```

Figura 10: Função t_max_calculator

Tendo os valores de t_{max}, conseguimos criar o gráfico que descreve a variação da integral consoante o valor de σ.

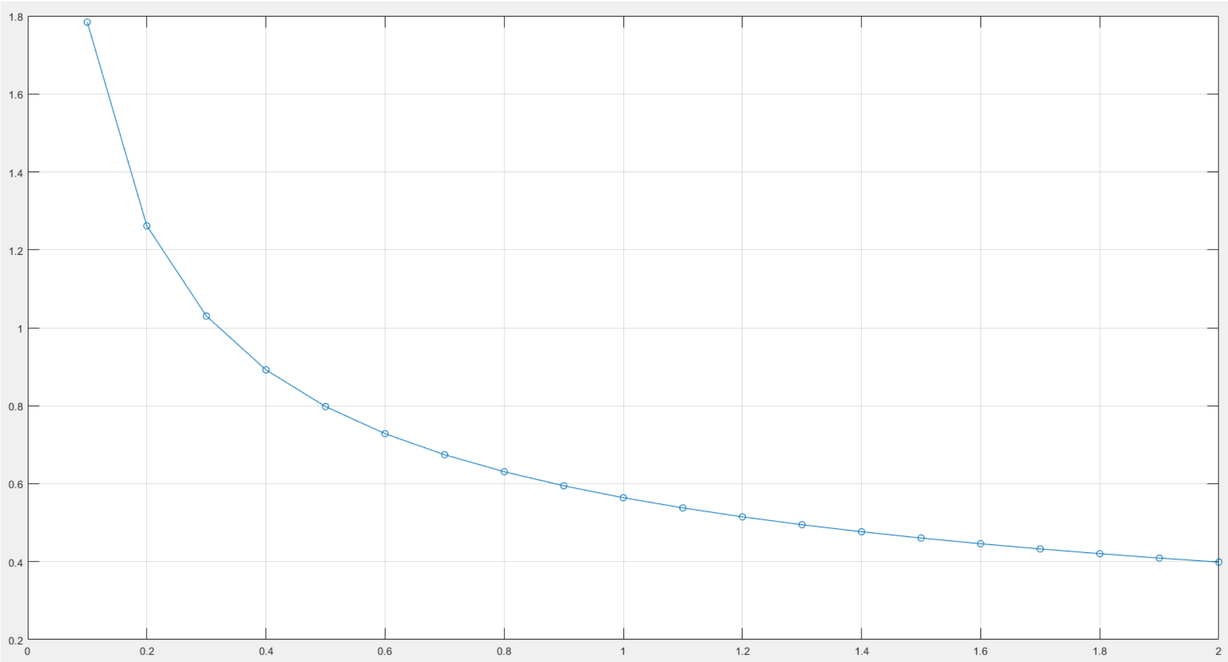

Figura 11: Variação integral segundo σ

Ao analisar a variação da integral segundo σ verifica-se que há uma convergência para valores próximos de 0.4, a medida que σ aumenta.

É de notar que devido ao rápido crescimento da função e aos arredondamentos que o $Matlab$ faz quando guarda as variáveis, o que leva a perdas de precisão, torna-se difícil obter resultados com grandes precisões num tempo de computação razoável, dessa forma as scripts que mostramos acima, e que se encontram na pasta zipada, consideram o valor de t_{max} aceitável quando a aproximação da integral apenas tem variações na ordem de 10^{-12}, obtidos em cerca de 80s, no entanto em anexo encontram-se versões da script que só consideram aceitáveis variações na integral na ordem dos 10^{-15}, apesar de isso levar a que tenha tempos de computação excessivamente elevados(895s).

Por fim, podemos fazer a verificação analítica do integral de forma a verificar o erro. Sabendo que:

$$\int \frac{2x\sqrt{\sigma}}{\sqrt{\pi}} e^{-\sigma x^2} = -\frac{e^{-x\sigma^2}}{\sqrt{x}\sqrt{\sigma}} \tag{34}$$

Podemos desenvolver uma script para calcular os erros da aproximação que calculámos.

```
%Function calculate errors - calcula o erro da aproximação de romberg para
% a integral da função tempo_alt quando o extremo superior é igual ao valor
% obtido pela função t_max
function erros = calculate_errors()
    t_maxs = t_max_calculator;

    real_values = 0.1:0.1:2;
    approximations = 0.1:0.1:2;
    n = 1;
    for sig = 0.1:0.1:2
        value = -((exp((-(t_maxs(n))^2)*sig))/(sqrt(pi)*sqrt(sig))) + (exp(0)/(sqrt(pi)*sqrt(sig)));
        real_values(n) = value;
        approximations(n) = romberg(@(x) tempo_alt(sig, x), 0, t_maxs(n), 15);
        n = n +1;
    end

    difs = approximations - real_values;
    erros = abs(difs);
end
```

Figura 12: Script calculate_errors

A script em cima começa por obter os valores dos extremos superiores do intervalo, de seguida executa um loop no qual calcula o valor analítico e o valor aproximado pelo método de Romberg e guarda esses valores em vetores diferentes. Por fim, acaba por fazer a diferença entre esses dois valores, e devolve o valor absoluto dessa diferença.

σ	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
Erro ($\times 10^{-8}$)	0,47	0,33	0,27	0,24	0,21	0,19	0,18	0,17	0,16	0,15
σ	1,1	1,2	1,3	0,4	0,5	0,6	0,7	0,8	0,9	1
Erro ($\times 10^{-8}$)	0,14	0,14	0,13	0,13	0,12	0,12	0,12	0,11	0,11	0,11

Tabela 5: Erros de aproximação

Ao verificar os erros, podemos concluir que a nossa aproximação apresenta resultados bastante satisfatórios dado todos os erros se encontrarem na ordem dos 10^{-8}. Tal como foi dito anteriormente, seria possível encontrar valores de t_{max} para os quais a aproximação seria mais precisa, no entanto é importante considerar que o aumento de precisão levaria a um aumento considerável do tempo computacional.

4 Conclusão

Através da realização deste trabalho foi-nos possibilitada a resolução de um problema da vida real, permitido-nos explorar variados conceitos da integração numérica ao longo da resolução das questões enumeradas no enunciado deste projeto.

Simultaneamente, também foi possível o estudo, compreensão e aplicação de um método de integração numérica mais avançado do que os estudados em aula, o Método de Romberg.

Desta forma, conseguimos compreender algumas aplicações deste método, como de algoritmos envolvendo questões de convergência e de precisão, e incluir ainda as respetivas representações gráficas, através da ferramenta $Matlab$. Com isto, concluímos o quão importante é a obtenção de valores que sejam o mais perto dos valores reais, tanto quanto possível, de forma a poder prever o comportamento de tudo o que possa ter uma descrição matemática.

5 Referências

1. `https://fenix.tecnico.ulisboa.pt/downloadFile/1970943312408383/NOTASdeAULAS_MC.pdf`
2. `https://www.youtube.com/watch?si=EFilWWPRgXSE7ukf&v=-tpBLjg-VCw&feature=youtu.be`
3. `https://www.youtube.com/watch?si=9NFyfTs_8TtRawXU&v=HtkylnbJklE&feature=youtu.be`
4. `https://www.youtube.com/watch?si=JOyPWoDtqZwbAntR&v=vzo47UoDwDA&feature=youtu.be`
5. `https://www.youtube.com/watch?v=oAhOqTJtSuk`