



Universidade Do Minho

Mestrado Integrado Em Engenharia Biomédica

Bibliotecas Digitais

MATLAB

Exercício 1

Braga, Fevereiro 2013



Universidade Do Minho

Mestrado Integrado Em Engenharia Biomédica

Bibliotecas Digitais

MATLAB

Exercício 1

Docente: Joaquim Macedo

Discentes: Hugo Gomes (58536)
Marta Moreira (54459)
Telma Veloso (58521)

Questão 1

Alínea a) A matriz identidade 4×4 foi obtida seguindo as directrizes patentes no enunciado e recorrendo, assim, ao símbolo ';' como indicador de quebra de linha na matriz. A mesma matriz poderia ser obtida recorrendo ao comando `eye(N)`, que retorna a matriz identidade de dimensão N por N.

```
>> MyMatrix=[1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]
```

```
MyMatrix =
```

```
1    0    0    0
0    1    0    0
0    0    1    0
0    0    0    1
```

```
>> MyMatrix=eye(4)
```

```
MyMatrix =
```

```
1    0    0    0
0    1    0    0
0    0    1    0
0    0    0    1
```

Alínea b) A utilização do símbolo ':' permite endereçar simultaneamente todas as posições nas colunas e linhas de `r1` e `c2`, respectivamente. Isto é, executando o comando `r1=MyMatrix(2,:)` obtêm-se todos os elementos da linha 2, obtendo-se, no caso do comando `c2=MyMatrix(:,3)`, todos os elementos da coluna 3 da matriz identidade anteriormente definida.

```
>> r1=MyMatrix(2,:)
```

```
r1 =
```

```
0    1    0    0
```

```
>> c2=MyMatrix(:,3)
```

```
c2 =
```

```
0
0
1
0
```

Alínea c) Os vectores `v1` e `v2` foram criados de forma semelhante àquela indicada no início da questão, através dos comandos `v1=[4 5 6]` e `v2=[3 2 1]`. Atendendo aos resultados da execução dos restantes comandos — representada abaixo —, observou-se que a única tentativa que falha é aquela em que se procura concatenar o vector `v1` e a matriz `c`. Neste caso específico, é emitida uma mensagem de erro relacionada com a função `horzcat`, responsável pela concatenação horizontal, e que deriva do facto de os elementos envolvidos na operação — `v1` e `c` — não apresentarem o mesmo número de linhas.

```
>> v3=[v1 v2]
```

```
v3 =
```

```
     4     5     6     3     2     1
```

```
>> v4=[v1; v2]
```

```
v4 =
```

```
     4     5     6
     3     2     1
```

```
>> v5=[v1 v2; v2 v1]
```

```
v5 =
```

```
     4     5     6     3     2     1
     3     2     1     4     5     6
```

```
>> v6=[v1 c]
```

```
Error using horzcat
```

```
CAT arguments dimensions are not consistent.
```

Questão 2

Alínea a) O *array* pretendido foi gerado utilizando, numa primeira fase, a notação que recorre ao símbolo `' : '` para definir um incremento de valor conhecido (0.1) entre os elementos, sendo que, numa fase posterior, se pressupôs uma amostragem igual a 200, utilizando o comando `linspace()`. O primeiro *array* obtido é constituído por 126 colunas, cada qual contendo um elemento que corresponde ao incremento de 0.1 de 0 até 4π (*vide* Anexos). Por sua vez, o *array* gerado por intermédio do comando `t=linspace(0,4*pi,200)` apresenta valores distanciados de $\frac{4\pi}{200} \approx 0.0628$, que perfazem, naturalmente, 200 colunas (*vide* Anexos).

Alínea b) O *array* resultante da aplicação do comando `y=sin(t)` à matriz 1×200 gerada na alínea anterior resulta numa matriz (*vide* Anexos) de dimensão semelhante em que cada uma das colunas corresponde ao resultado do cálculo do seno de cada um dos valores que constituem as colunas da primeira matriz.

Alínea c) Através do comando `plot(y)`, obteve-se o gráfico representado na Figura 1. O eixo **xx** diz, neste caso específico, respeito ao índice das colunas da matriz **y**, sendo que, de forma correspondente, os pontos nulos do gráfico se referem às colunas 0 e 200, ambas contendo, efectivamente, um valor igual a zero, conforme visível na matriz da alínea anterior (*vide* Anexos).

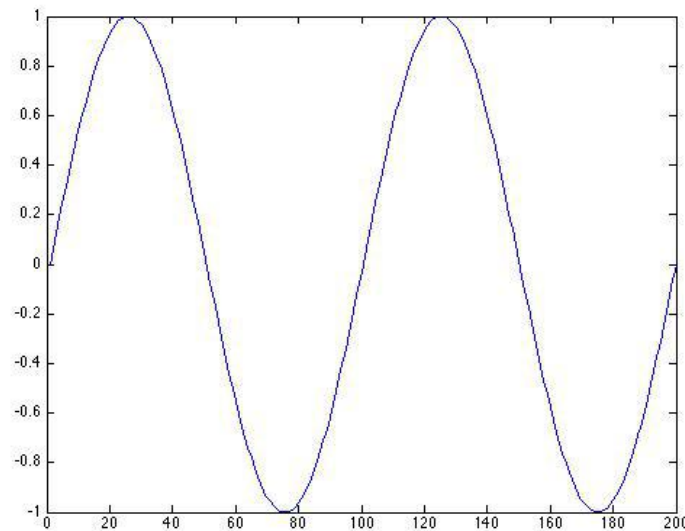


Figura 1: Gráfico resultante da execução do comando `plot(y)`.

Alínea d) Relativamente à função `plot(y)`, a função `plot(t,y)` — cujo *output* gráfico se encontra representado na Figura 2 — difere na medida em que permite produzir um gráfico bidimensional (ao invés de linear) em que, devido ao facto das matrizes em questão apresentarem a mesma dimensão (1×200), se plotam sucessivamente as colunas de **t** contra as colunas de **y**. Assim, o eixo dos **xx** corresponde, neste caso, aos valores de **t**, contrariamente ao caso da alínea anterior, em que correspondia aos índices das colunas da matriz **y**.

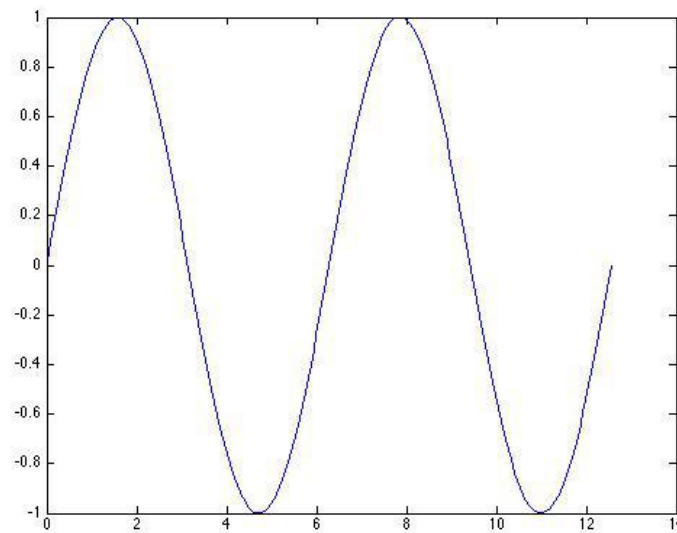
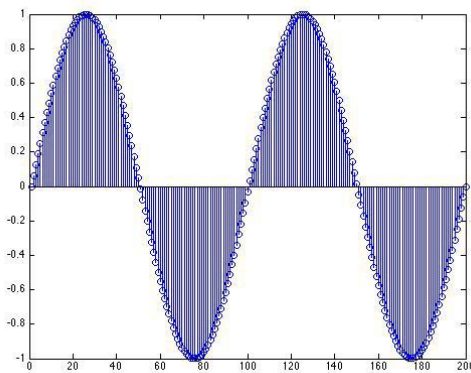
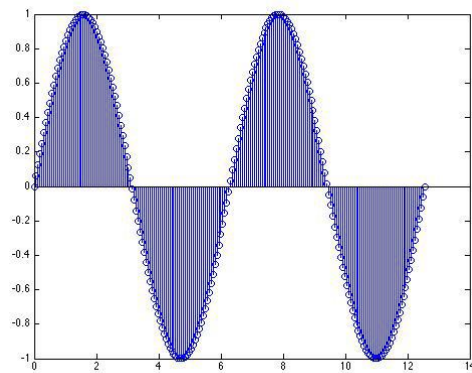


Figura 2: Gráfico resultante da execução do comando `plot(t,y)`.

Alínea e) A função `stem(.)` representa os dados sob a forma de *stems* — linhas que partem do eixo dos `xx` até uma dada posição relativamente ao eixo dos `yy`, assinalada com um marcador (um círculo, por defeito) e correspondente ao valor dos dados em questão. A partir da representação gráfica gerada (Figura 3), retira-se que a informação obtida é semelhante àquela ilustrada nas Figuras 1 e 2, distanciando-se apenas no que concerne a representação.



A)



B)

Figura 3: Gráficos resultantes da aplicação dos comandos A) `stem(y)` e B) `stem(t,y)`.

Alínea f. Os comandos indicados — `title('')`, `xlabel('')` e `ylabel('')` — permitem a legendagem dos gráficos obtidos, nomeadamente ao nível da atribuição de um título e da descrição dos valores do eixo dos `xx` e `yy`, respectivamente. A título demonstrativo, aplicaram-se estes comandos ao gráfico da Figura 1.

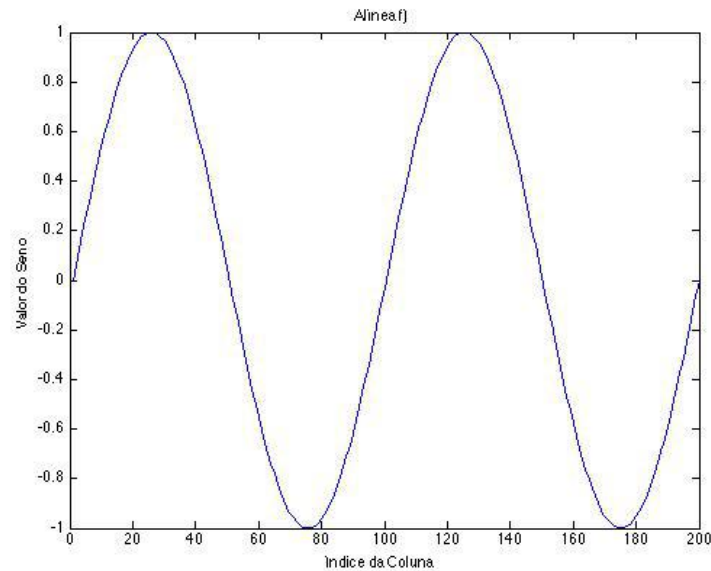


Figura 4: Gráfico resultante da execução do comando `plot(y)`, devidamente legendado.

Questão 3

A indexação permite seleccionar subconjuntos de elementos a partir de um dado *array*. Os comandos especificados traduzem três técnicas distintas de indexação, sendo que a primeira, utilizada em `a`, permite extrair elementos através da aplicação de um intervalo — neste caso, de 2 a 4 —, ou seja, o resultado corresponde aos elementos do vector `v3` nas posições 2, 3 e 4.

```
>> a=v3(2:4)
```

`a =`

```
5    6    3
```

Por sua vez, o segundo comando vai proceder à indexação seleccionando todas as linhas (operação veiculada pelo marcador `:`), extraíndo depois os elementos referentes, mais uma vez, a um intervalo, nomeadamente da coluna 3 à coluna 5.

```
>> b=v5(:,3:5)
```

`b =`

```
6    3    2
1    4    5
```

Um último exemplo realiza a indexação com base na extracção dos elementos das colunas com índice ímpar de cada uma das linhas da matriz `v5`. Isto é conseguido através do já referido marcador de selecção de todas as linhas, seguido da sequência `1:2:6`, que indica a extracção de duas em duas posições, começando na posição 1 e terminando na posição 6.

```
>> c=v5(:,1:2:6)
```

```
c =
```

```
    4    6    2  
    3    1    5
```

Torna-se importante relevar que, contrariamente a outros ambientes e linguagens de programação, a numeração dos elementos tem início no índice 1 e não no índice 0, no MATLAB.

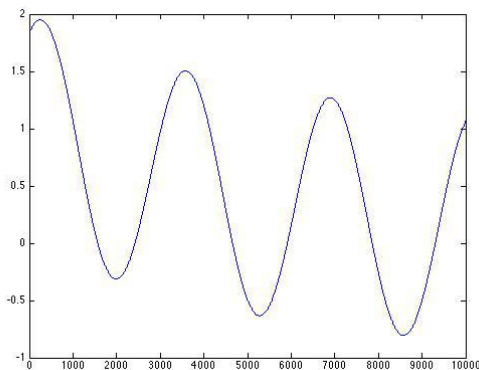
Questão 4

Alínea a. A sequência temporal pedida foi gerada, consoante explorado na Questão 2, utilizando a função `t=linspace(1,20,10000)`.

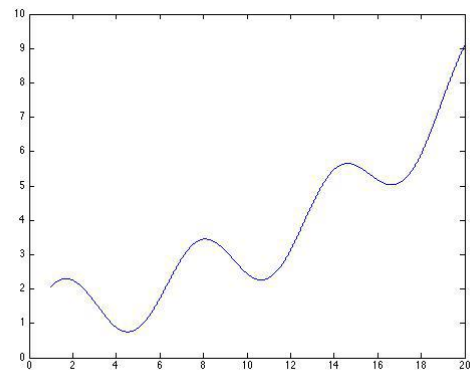
Alínea b. Dado tratar-se de um elemento de alguma complexidade, a função veiculada foi gerada recorrendo à sua divisão em duas funções mais simples, `y1` e `y2`, que foram posteriormente somadas, dando origem à função completa `y`. De seguida, obteve-se o gráfico dessa mesma função aplicando o comando `plot` (Figura 5).

```
>> y1=exp(-(t-1)/10)
>> y2=sin(t)
>> y=y2+y1
>> plot(y)
```

```
>> y1=exp(-(t-1)/10)
>> y2=sin(t)
>> y=y2+y1
>> plot(t,y)
```



A)



B)

Figura 5: Gráficos resultantes da aplicação dos comandos A) `plot(y)` e B) `plot(t,y)`, para `y=y1+y2`.

Alínea c. A obtenção da função `y` foi levada a cabo de forma semelhante àquela descrita na alínea anterior, pese embora, neste caso específico, tenha sido necessário não somar `y1` e `y2`, mas sim multiplicar, o que envolveu um cuidado especial devido ao facto de os *arrays* obtidos terem dimensões distintas. Efetivamente, a tentativa de multiplicar ambos os elementos utilizando o operador convencional `*` retornou uma mensagem de erro.

```
>> y=y1*y2
Error using *
Inner matrix dimensions must agree.
```

Neste sentido, tornou-se necessário recorrer ao operador `.*`, que efetua a multiplicação elemento a elemento, de forma a obter o gráfico desejado (Figura 6).

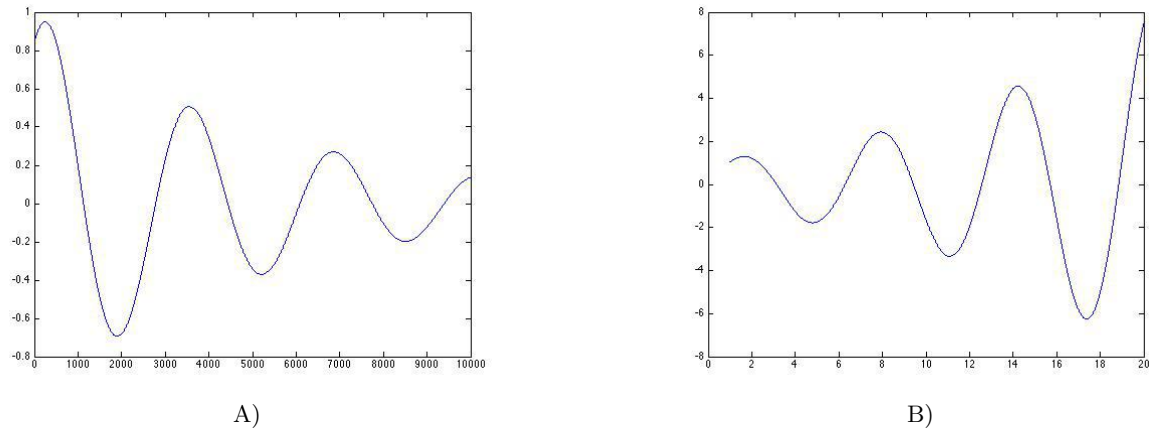


Figura 6: Gráficos resultantes da aplicação dos comandos A) `plot(y)` e B) `plot(t,y)`, para $y=y1.*y2$.

Alínea d. Neste último ponto, utilizou-se o conhecimento recolhido ao longo da realização do restante exercício para gerar e plotar a função fornecida, sendo que a sequência de comando emitida com a referida finalidade foi a seguinte:

```
>> y=sin(500./t)
>> plot(t,y)
>> title('Questão 4 | Alínea d')
>> xlabel('t')
>> ylabel('y(t)')
```

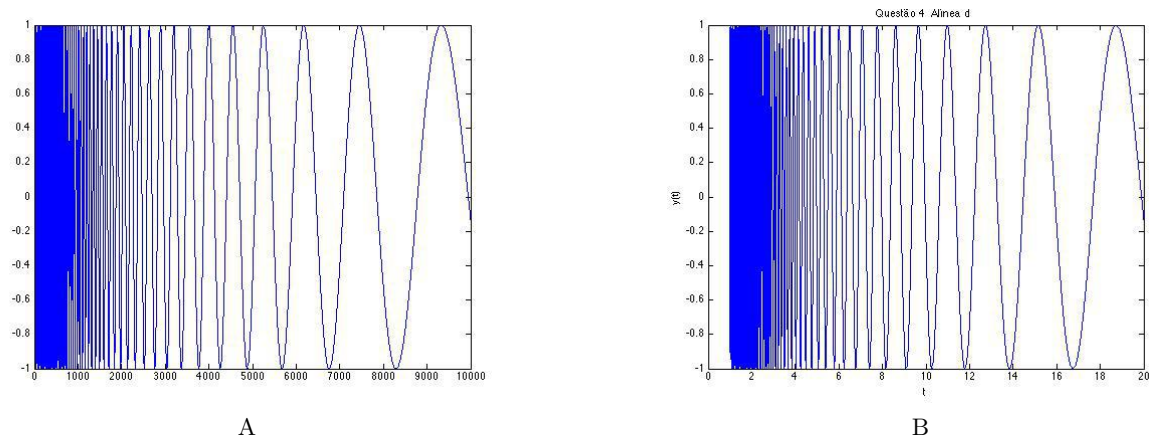


Figura 7: Gráficos resultantes da aplicação dos comandos A) `plot(y)` e B) `plot(t,y)`, para $y = \sin(\frac{500}{t})$.

Anexos

Questão 2

Alínea a.

```
>> t=[0:0.1:4*pi]
```

```
t =
```

```
Columns 1 through 8
```

```
0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000    0.7000
```

```
Columns 9 through 16
```

```
0.8000    0.9000    1.0000    1.1000    1.2000    1.3000    1.4000    1.5000
```

```
Columns 17 through 24
```

```
1.6000    1.7000    1.8000    1.9000    2.0000    2.1000    2.2000    2.3000
```

```
Columns 25 through 32
```

```
2.4000    2.5000    2.6000    2.7000    2.8000    2.9000    3.0000    3.1000
```

```
Columns 33 through 40
```

```
3.2000    3.3000    3.4000    3.5000    3.6000    3.7000    3.8000    3.9000
```

```
Columns 41 through 48
```

```
4.0000    4.1000    4.2000    4.3000    4.4000    4.5000    4.6000    4.7000
```

```
Columns 49 through 56
```

```
4.8000    4.9000    5.0000    5.1000    5.2000    5.3000    5.4000    5.5000
```

```
Columns 57 through 64
```

```
5.6000    5.7000    5.8000    5.9000    6.0000    6.1000    6.2000    6.3000
```

```
Columns 65 through 72
```

```
6.4000    6.5000    6.6000    6.7000    6.8000    6.9000    7.0000    7.1000
```

```
Columns 73 through 80
```

```
7.2000    7.3000    7.4000    7.5000    7.6000    7.7000    7.8000    7.9000
```

```
Columns 81 through 88
```

```
8.0000    8.1000    8.2000    8.3000    8.4000    8.5000    8.6000    8.7000
```

Columns 89 through 96

8.8000	8.9000	9.0000	9.1000	9.2000	9.3000	9.4000	9.5000
--------	--------	--------	--------	--------	--------	--------	--------

Columns 97 through 104

9.6000	9.7000	9.8000	9.9000	10.0000	10.1000	10.2000	10.3000
--------	--------	--------	--------	---------	---------	---------	---------

Columns 105 through 112

10.4000	10.5000	10.6000	10.7000	10.8000	10.9000	11.0000	11.1000
---------	---------	---------	---------	---------	---------	---------	---------

Columns 113 through 120

11.2000	11.3000	11.4000	11.5000	11.6000	11.7000	11.8000	11.9000
---------	---------	---------	---------	---------	---------	---------	---------

Columns 121 through 126

12.0000	12.1000	12.2000	12.3000	12.4000	12.5000		
---------	---------	---------	---------	---------	---------	--	--

```
>> t=linspace(0,4*pi,200)
```

t =

Columns 1 through 8

0	0.0631	0.1263	0.1894	0.2526	0.3157	0.3789	0.4420
---	--------	--------	--------	--------	--------	--------	--------

Columns 9 through 16

0.5052	0.5683	0.6315	0.6946	0.7578	0.8209	0.8841	0.9472
--------	--------	--------	--------	--------	--------	--------	--------

Columns 17 through 24

1.0104	1.0735	1.1367	1.1998	1.2630	1.3261	1.3892	1.4524
--------	--------	--------	--------	--------	--------	--------	--------

Columns 25 through 32

1.5155	1.5787	1.6418	1.7050	1.7681	1.8313	1.8944	1.9576
--------	--------	--------	--------	--------	--------	--------	--------

Columns 33 through 40

2.0207	2.0839	2.1470	2.2102	2.2733	2.3365	2.3996	2.4628
--------	--------	--------	--------	--------	--------	--------	--------

Columns 41 through 48

2.5259	2.5891	2.6522	2.7153	2.7785	2.8416	2.9048	2.9679
--------	--------	--------	--------	--------	--------	--------	--------

Columns 49 through 56

3.0311	3.0942	3.1574	3.2205	3.2837	3.3468	3.4100	3.4731
--------	--------	--------	--------	--------	--------	--------	--------

Columns 57 through 64

3.5363	3.5994	3.6626	3.7257	3.7889	3.8520	3.9152	3.9783
Columns 65 through 72							
4.0414	4.1046	4.1677	4.2309	4.2940	4.3572	4.4203	4.4835
Columns 73 through 80							
4.5466	4.6098	4.6729	4.7361	4.7992	4.8624	4.9255	4.9887
Columns 81 through 88							
5.0518	5.1150	5.1781	5.2413	5.3044	5.3675	5.4307	5.4938
Columns 89 through 96							
5.5570	5.6201	5.6833	5.7464	5.8096	5.8727	5.9359	5.9990
Columns 97 through 104							
6.0622	6.1253	6.1885	6.2516	6.3148	6.3779	6.4411	6.5042
Columns 105 through 112							
6.5673	6.6305	6.6936	6.7568	6.8199	6.8831	6.9462	7.0094
Columns 113 through 120							
7.0725	7.1357	7.1988	7.2620	7.3251	7.3883	7.4514	7.5146
Columns 121 through 128							
7.5777	7.6409	7.7040	7.7672	7.8303	7.8934	7.9566	8.0197
Columns 129 through 136							
8.0829	8.1460	8.2092	8.2723	8.3355	8.3986	8.4618	8.5249
Columns 137 through 144							
8.5881	8.6512	8.7144	8.7775	8.8407	8.9038	8.9670	9.0301
Columns 145 through 152							
9.0933	9.1564	9.2195	9.2827	9.3458	9.4090	9.4721	9.5353
Columns 153 through 160							
9.5984	9.6616	9.7247	9.7879	9.8510	9.9142	9.9773	10.0405
Columns 161 through 168							
10.1036	10.1668	10.2299	10.2931	10.3562	10.4194	10.4825	10.5456

Columns 169 through 176

10.6088	10.6719	10.7351	10.7982	10.8614	10.9245	10.9877	11.0508
---------	---------	---------	---------	---------	---------	---------	---------

Columns 177 through 184

11.1140	11.1771	11.2403	11.3034	11.3666	11.4297	11.4929	11.5560
---------	---------	---------	---------	---------	---------	---------	---------

Columns 185 through 192

11.6192	11.6823	11.7455	11.8086	11.8717	11.9349	11.9980	12.0612
---------	---------	---------	---------	---------	---------	---------	---------

Columns 193 through 200

12.1243	12.1875	12.2506	12.3138	12.3769	12.4401	12.5032	12.5664
---------	---------	---------	---------	---------	---------	---------	---------

Alínea b.

```
>> y=sin(t)
```

y =

Columns 1 through 8

0	0.0631	0.1260	0.1883	0.2499	0.3105	0.3699	0.4278
---	--------	--------	--------	--------	--------	--------	--------

Columns 9 through 16

0.4840	0.5382	0.5903	0.6401	0.6873	0.7318	0.7733	0.8118
--------	--------	--------	--------	--------	--------	--------	--------

Columns 17 through 24

0.8470	0.8789	0.9072	0.9320	0.9530	0.9702	0.9836	0.9930
--------	--------	--------	--------	--------	--------	--------	--------

Columns 25 through 32

0.9985	1.0000	0.9975	0.9910	0.9806	0.9663	0.9481	0.9261
--------	--------	--------	--------	--------	--------	--------	--------

Columns 33 through 40

0.9005	0.8712	0.8385	0.8025	0.7632	0.7209	0.6758	0.6279
--------	--------	--------	--------	--------	--------	--------	--------

Columns 41 through 48

0.5775	0.5249	0.4701	0.4135	0.3552	0.2955	0.2346	0.1728
--------	--------	--------	--------	--------	--------	--------	--------

Columns 49 through 56

0.1103	0.0473	-0.0158	-0.0789	-0.1416	-0.2038	-0.2652	-0.3255
--------	--------	---------	---------	---------	---------	---------	---------

Columns 57 through 64

-0.3845	-0.4420	-0.4977	-0.5515	-0.6030	-0.6521	-0.6987	-0.7424
---------	---------	---------	---------	---------	---------	---------	---------

Columns 65 through 72

-0.7832 -0.8209 -0.8553 -0.8863 -0.9138 -0.9376 -0.9577 -0.9739

Columns 73 through 80

-0.9863 -0.9947 -0.9992 -0.9997 -0.9962 -0.9888 -0.9774 -0.9621

Columns 81 through 88

-0.9429 -0.9201 -0.8935 -0.8634 -0.8298 -0.7930 -0.7529 -0.7099

Columns 89 through 96

-0.6640 -0.6155 -0.5646 -0.5113 -0.4561 -0.3990 -0.3404 -0.2804

Columns 97 through 104

-0.2192 -0.1572 -0.0946 -0.0316 0.0316 0.0946 0.1572 0.2192

Columns 105 through 112

0.2804 0.3404 0.3990 0.4561 0.5113 0.5646 0.6155 0.6640

Columns 113 through 120

0.7099 0.7529 0.7930 0.8298 0.8634 0.8935 0.9201 0.9429

Columns 121 through 128

0.9621 0.9774 0.9888 0.9962 0.9997 0.9992 0.9947 0.9863

Columns 129 through 136

0.9739 0.9577 0.9376 0.9138 0.8863 0.8553 0.8209 0.7832

Columns 137 through 144

0.7424 0.6987 0.6521 0.6030 0.5515 0.4977 0.4420 0.3845

Columns 145 through 152

0.3255 0.2652 0.2038 0.1416 0.0789 0.0158 -0.0473 -0.1103

Columns 153 through 160

-0.1728 -0.2346 -0.2955 -0.3552 -0.4135 -0.4701 -0.5249 -0.5775

Columns 161 through 168

-0.6279 -0.6758 -0.7209 -0.7632 -0.8025 -0.8385 -0.8712 -0.9005

Columns 169 through 176

-0.9261 -0.9481 -0.9663 -0.9806 -0.9910 -0.9975 -1.0000 -0.9985

Columns 177 through 184

-0.9930 -0.9836 -0.9702 -0.9530 -0.9320 -0.9072 -0.8789 -0.8470

Columns 185 through 192

-0.8118 -0.7733 -0.7318 -0.6873 -0.6401 -0.5903 -0.5382 -0.4840

Columns 193 through 200

-0.4278 -0.3699 -0.3105 -0.2499 -0.1883 -0.1260 -0.0631 -0.0000