Computació Numèrica

==
=

Autoevaluació

Table of Contents

Entrega AA3 Marta Granero I Martí

EXERCICI 1 - Resolució de sistemas d'equacions no lineals

El sistema d'equacions no lineals

$$\begin{aligned} z_1^2 - 10z_1 + z_2^2 + 8 &= 0\,, \\ z_1 z_2^2 + z_1 - 10z_2 + 8 &= 0\,, \end{aligned} \qquad \text{(A}$$

té dues arrels, una és $(-1,1)^t$ i l'altre és a prop de $(2,3)^t$.

Objectiu: Determinar un valor aproximat de la solució prop de $(2,3)$ amb una exactitud tal que

$$||\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}|| \le 10^{-6} \quad \text{i} \quad ||F(\mathbf{z}^{(k+1)})|| < 10^{-6} \quad \text{si} \quad \mathbf{z} = (z_1, z_2)^t\,.$$

1a) Aproximeu totes les solucions del sistema (A.1) gràficament.

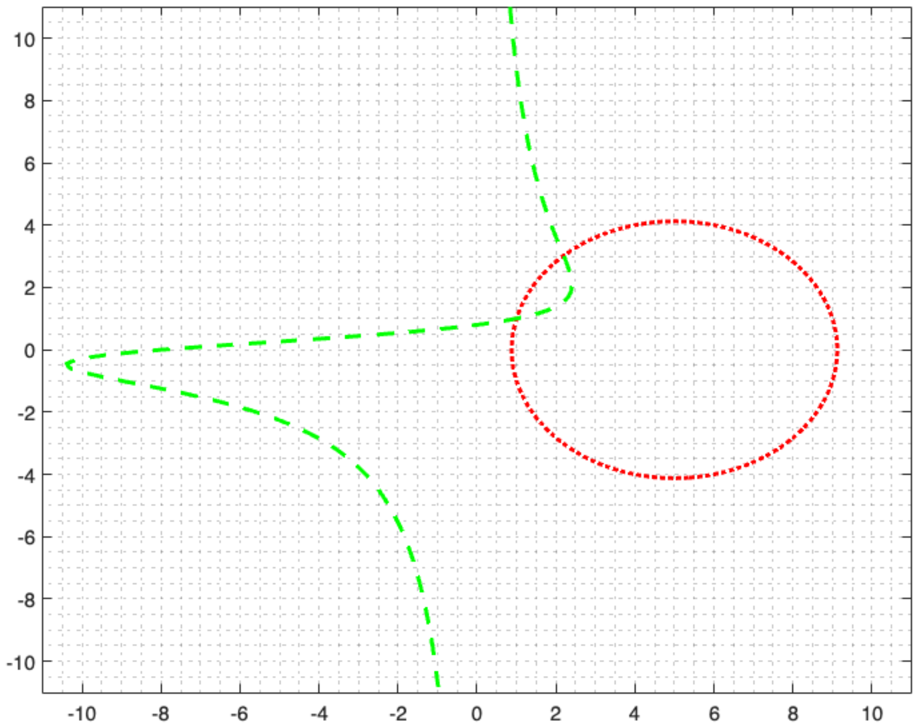
```
%Procedim a la resolució gràfica

F1 = @(x,y) x^2-10*x + y^2 + 8; %z1 = x, z2 = y
F2 = @(x,y) x*y^2 + x - 10*y + 8;

fimplicit(F1,[-11 11 -11 11],':r','LineWidth',2)
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize
your function to return an output with the same size and shape as the input arguments.
```

```
hold on
fimplicit(F2,[-11 11 -11 11],'--g','LineWidth',2)
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize
your function to return an output with the same size and shape as the input arguments.
```

```
grid minor

hold off
```




```
%Solució bona es fa servir en apartats posteriors per calcular el nombre de
%xifres signficatives
x0=[2;3];
[x, ~] = fsolve(@myfun,x0);
```

```
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
```

```
myfun(x);

valorBo = x;
```

1b) Fent ús del mètode de **Newton** i prenent $x_0 = (2,3)$. Presenteu les iteracions en una taula.

```
%Mètode de Newton prenent 6 iteracions
%taula
format longG
x0 =[2;3];
F  = @(x,y)[x^2-10*x + y^2 + 8; x*y^2 + x - 10*y + 8];
F(x0(1),x0(2));
JF = @(x,y)[2*x-10, 2*y; 1+y^2, 2*x*y-10];

for k=1:200
    y  = linsolve(JF(x0(1),x0(2)),(-F(x0(1),x0(2))));
    x0 = x0+y;

    res(k).y = y;
    res(k).x0 = x0;
    res(k).decimalsCorrectes = min(abs(valorBo'-x0'));
end

res %taula on es mostren tambe els decimals correctes
```

res = 1×200 struct

Fields	y	x0	decimalsCor...
1	[0.19444...	[2.19444...	0.001005...
2	[-0.0009...	[2.19344...	1.000158856...
3	[-1.0001588...	[2.19343...	6.838973831...
4	[-1.2185463...	[2.19343...	1.219229162...
5	[-1.4544086...	[2.19343...	1.219229162...
6	[-2.7030303...	[2.19343...	1.219233602...
7	[4.15743901...	[2.19343...	1.219229162...
8	[-1.4544086...	[2.19343...	1.219229162...
9	[-2.7030303...	[2.19343...	1.219233602...
10	[4.15743901...	[2.19343...	1.219229162...
11	[-1.4544086...	[2.19343...	1.219229162...
12	[-2.7030303...	[2.19343...	1.219233602...
13	[4.15743901...	[2.19343...	1.219229162...
14	[-1.4544086...	[2.19343...	1.219229162...

⋮

```
tolF = norm(F(x0(1),x0(2)),'inf');

tolX = norm(y,'inf');
```

```
disp(['Solució = ',num2str(x0')])
```

```
Solució = 2.1934      3.0205
```

```
disp(['tolF = ',num2str(tolF)])
```

```
tolF = 3.5527e-15
```

```
disp(['tolX = ',num2str(tolX)])
```

```
tolX = 4.5297e-16
```

```
disp(['Iteracions = ',num2str(k)])
```

```
Iteracions = 200
```

```
%Obtenim amb 200 iteracions 9 decimals correctes i és un mètode convergent
```

1c) Fent ús del mètode de **Newton modificat** i prenent $x_0 = (2, 3)$. Presenteu les iteracions en una taula.

```
valor = zeros(200,2);

for k = 1:200
    valor(k,:) = newtonModificat(F,JF,x0,k,tolF,tolX,5); %30 = #iteracions fixes
    if k > 1 && valor(k,1) == valor(k-1,1) && valor(k,2) == valor(k-1,2) %condició para
        break;
    end
end
```

```
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 1
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 2
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 3
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 4
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 5
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 6
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
```

```
Iteracions = 7
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 8
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 9
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 10
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 11
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 12
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 13
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 14
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 15
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 16
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 17
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 18
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 19
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 20
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 21
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 22
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
```

```
Iteracions = 23
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 24
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 25
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 26
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 27
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 28
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 29
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 30
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 31
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 32
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 33
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 34
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 35
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 36
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 37
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 38
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
```

```
Iteracions = 39
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 40
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 41
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 42
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 43
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 44
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 45
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 46
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 47
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 48
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 49
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 50
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 51
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 52
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 53
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 54
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
```

```
Iteracions = 55
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 56
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 57
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 58
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 59
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 60
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 61
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 62
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 63
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 64
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 65
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 66
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 67
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 68
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 69
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 70
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
```

```
Iteracions = 71
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 72
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 73
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 74
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 75
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 76
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 77
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 78
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 79
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 80
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 81
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 82
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 83
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 84
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 85
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 86
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
```

```
Iteracions = 87
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 88
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 89
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 90
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 91
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 92
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 93
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 94
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 95
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 96
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 97
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 98
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 99
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 100
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 101
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 102
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
```

```
Iteracions = 103
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 104
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 105
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 106
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 107
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 108
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 109
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 110
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 111
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 112
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 113
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 114
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 115
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 116
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 117
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 118
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
```

```
Iteracions = 119
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 120
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 121
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 122
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 123
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 124
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 125
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 126
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 127
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 128
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 129
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 130
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 131
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 132
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 133
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 134
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
```

```
Iteracions = 135
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 136
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 137
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 138
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 139
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 140
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 141
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 142
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 143
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 144
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 145
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 146
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 147
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 148
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 149
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 150
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
```

```
Iteracions = 151
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 152
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 153
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 154
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 155
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 156
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 157
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 158
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 159
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 160
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 161
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 162
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 163
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 164
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 165
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 166
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
```

```
Iteracions = 167
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 168
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 169
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 170
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 171
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 172
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 173
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 174
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 175
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 176
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 177
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 178
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 179
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 180
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 181
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 182
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
```

```
Iteracions = 183
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 184
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 185
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 186
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 187
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 188
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 189
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 190
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 191
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 192
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 193
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 194
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 195
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
Iteracions = 196
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 197
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.7574e-16
Iteracions = 198
Solució = 2.1934      3.0205
tolF = 5.0243e-15
tolX = 3.6898e-16
```

```
Iteracions = 199
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolX = 4.6213e-16
Iteracions = 200
```

```
array2table(valor(1:k,:), "VariableNames", {'x', 'y'})
```

ans = 200×2 table

	x	y
1	2.193439415...	3.020466468...
2	2.193439415...	3.020466468...
3	2.193439415...	3.020466468...
4	2.193439415...	3.020466468...
5	2.193439415...	3.020466468...
6	2.193439415...	3.020466468...
7	2.193439415...	3.020466468...
8	2.193439415...	3.020466468...
9	2.193439415...	3.020466468...
10	2.193439415...	3.020466468...
11	2.193439415...	3.020466468...
12	2.193439415...	3.020466468...
13	2.193439415...	3.020466468...
14	2.193439415...	3.020466468...

⋮

```
decimalsCorrectes = min(abs(valorBo' - valor(k,:)))
```

```
decimalsCorrectes =
      1.21922916207495e-10
```

```
%Obtenim amb 200 iteracions 9 decimals correctes i és un mètode convergent
```

1d) Fent ús del mètode **de Jacobi** i prenent $x_0 = (2, 3)$. Presenteu les iteracions en una taula.

```
valor = zeros(200,2);
for k = 1:200
    valor(k,:) = newtonJacobi(F,JF,x0,k,tolF,tolX);
    if k > 1 && valor(k,1) == valor(k-1,1) && valor(k,2) == valor(k-1,2) %condició para
        break;
    end
end
```

```
Solució = 2.1934      3.0205
tolF = 5.3291e-15
tolZ = 1.093e-15
Iteracions = 1
Solució = 2.1934      3.0205
tolF = 1.0805e-14
tolZ = 9.494e-16
Iteracions = 2
Solució = 2.1934      3.0205
tolF = 1.986e-14
tolZ = 3.2942e-15
Iteracions = 3
Solució = 2.1934      3.0205
tolF = 3.6231e-14
tolZ = 3.6487e-15
Iteracions = 4
Solució = 2.1934      3.0205
tolF = 6.5509e-14
tolZ = 1.1003e-14
Iteracions = 5
Solució = 2.1934      3.0205
tolF = 1.1638e-13
tolZ = 1.2203e-14
Iteracions = 6
Solució = 2.1934      3.0205
tolF = 2.1913e-13
tolZ = 3.5256e-14
Iteracions = 7
Solució = 2.1934      3.0205
tolF = 3.9577e-13
tolZ = 4.0476e-14
Iteracions = 8
Solució = 2.1934      3.0205
tolF = 7.3402e-13
tolZ = 1.2002e-13
Iteracions = 9
Solució = 2.1934      3.0205
tolF = 1.3277e-12
tolZ = 1.3578e-13
Iteracions = 10
Solució = 2.1934      3.0205
tolF = 2.47e-12
tolZ = 4.0288e-13
Iteracions = 11
Solució = 2.1934      3.0205
tolF = 4.4672e-12
tolZ = 4.5618e-13
Iteracions = 12
Solució = 2.1934      3.0205
tolF = 8.2934e-12
tolZ = 1.356e-12
Iteracions = 13
Solució = 2.1934      3.0205
tolF = 1.4975e-11
tolZ = 1.5316e-12
Iteracions = 14
Solució = 2.1934      3.0205
tolF = 2.7797e-11
tolZ = 4.5454e-12
Iteracions = 15
Solució = 2.1934      3.0205
tolF = 5.0187e-11
tolZ = 5.1336e-12
Iteracions = 16
```

```
Solució = 2.1934      3.0205
tolF = 9.3163e-11
tolZ = 1.5234e-11
Iteracions = 17
Solució = 2.1934      3.0205
tolF = 1.6821e-10
tolZ = 1.7205e-11
Iteracions = 18
Solució = 2.1934      3.0205
tolF = 3.1227e-10
tolZ = 5.1062e-11
Iteracions = 19
Solució = 2.1934      3.0205
tolF = 5.6383e-10
tolZ = 5.7669e-11
Iteracions = 20
Solució = 2.1934      3.0205
tolF = 1.0467e-09
tolZ = 1.7115e-10
Iteracions = 21
Solució = 2.1934      3.0205
tolF = 1.8898e-09
tolZ = 1.9329e-10
Iteracions = 22
Solució = 2.1934      3.0205
tolF = 3.5082e-09
tolZ = 5.7365e-10
Iteracions = 23
Solució = 2.1934      3.0205
tolF = 6.3343e-09
tolZ = 6.4788e-10
Iteracions = 24
Solució = 2.1934      3.0205
tolF = 1.1759e-08
tolZ = 1.9228e-09
Iteracions = 25
Solució = 2.1934      3.0205
tolF = 2.1231e-08
tolZ = 2.1716e-09
Iteracions = 26
Solució = 2.1934      3.0205
tolF = 3.9413e-08
tolZ = 6.4448e-09
Iteracions = 27
Solució = 2.1934      3.0205
tolF = 7.1163e-08
tolZ = 7.2787e-09
Iteracions = 28
Solució = 2.1934      3.0205
tolF = 1.321e-07
tolZ = 2.1602e-08
Iteracions = 29
Solució = 2.1934      3.0205
tolF = 2.3853e-07
tolZ = 2.4397e-08
Iteracions = 30
Solució = 2.1934      3.0205
tolF = 4.4279e-07
tolZ = 7.2404e-08
Iteracions = 31
Solució = 2.1934      3.0205
tolF = 7.9949e-07
tolZ = 8.1773e-08
Iteracions = 32
```

```
Solució = 2.1934      3.0205
tolF = 1.4841e-06
tolZ = 2.4268e-07
Iteracions = 33
Solució = 2.1934      3.0205
tolF = 2.6797e-06
tolZ = 2.7409e-07
Iteracions = 34
Solució = 2.1934      3.0205
tolF = 4.9745e-06
tolZ = 8.1343e-07
Iteracions = 35
Solució = 2.1934      3.0205
tolF = 8.982e-06
tolZ = 9.1869e-07
Iteracions = 36
Solució = 2.1934      3.0205
tolF = 1.6674e-05
tolZ = 2.7265e-06
Iteracions = 37
Solució = 2.1934      3.0205
tolF = 3.0106e-05
tolZ = 3.0793e-06
Iteracions = 38
Solució = 2.1934      3.0205
tolF = 5.5887e-05
tolZ = 9.1386e-06
Iteracions = 39
Solució = 2.1934      3.0205
tolF = 0.00010091
tolZ = 1.0321e-05
Iteracions = 40
Solució = 2.1934      3.0204
tolF = 0.00018732
tolZ = 3.063e-05
Iteracions = 41
Solució = 2.1934      3.0205
tolF = 0.00033822
tolZ = 3.4594e-05
Iteracions = 42
Solució = 2.1934      3.0206
tolF = 0.0006279
tolZ = 0.00010267
Iteracions = 43
Solució = 2.1935      3.0205
tolF = 0.0011338
tolZ = 0.00011596
Iteracions = 44
Solució = 2.1935      3.0202
tolF = 0.002104
tolZ = 0.00034407
Iteracions = 45
Solució = 2.1931      3.0203
tolF = 0.0037987
tolZ = 0.00038857
Iteracions = 46
Solució = 2.1933      3.0214
tolF = 0.0070589
tolZ = 0.001154
Iteracions = 47
Solució = 2.1945       3.021
tolF = 0.012749
tolZ = 0.0013036
Iteracions = 48
```

```
Solució = 2.194      3.0172
tolF = 0.023586
tolZ = 0.0038595
Iteracions = 49
Solució = 2.1899      3.0186
tolF = 0.042553
tolZ = 0.0043566
Iteracions = 50
Solució = 2.1914      3.0315
tolF = 0.079895
tolZ = 0.013033
Iteracions = 51
Solució = 2.2054      3.0267
tolF = 0.14464
tolZ = 0.014746
Iteracions = 52
Solució = 2.2002      2.9844
tolF = 0.25863
tolZ = 0.04262
Iteracions = 53
Solució = 2.1548      2.9995
tolF = 0.46277
tolZ = 0.047849
Iteracions = 54
Solució = 2.171      3.1545
tolF = 0.98098
tolZ = 0.15579
Iteracions = 55
Solució = 2.3396      3.0924
tolF = 1.8247
tolZ = 0.17963
Iteracions = 56
Solució = 2.2721      2.6922
tolF = 2.3178
tolZ = 0.40588
Iteracions = 57
Solució = 1.8486      2.7736
tolF = 3.7194
tolZ = 0.43127
Iteracions = 58
Solució = 1.94766      17.1778
tolF = 503.0574
tolZ = 14.4046
Iteracions = 59
Solució = 49.0255      9.92326
tolF = 5194.1496
tolZ = 47.6335
Iteracions = 60
Solució = 26.0875      4.95394
tolF = 771.2636
tolZ = 23.4701
Iteracions = 61
Solució = 15.3649      2.43947
tolF = 132.1477
tolZ = 11.0134
Iteracions = 62
Solució = 10.7155      1.04783
tolF = 26.0985
tolZ = 4.8533
Iteracions = 63
Solució = 9.2489      -0.558
tolF = 25.7448
tolZ = 2.1748
Iteracions = 64
```

```
Solució = 9.0883     0.70708
tolF = 14.5629
tolZ = 1.2752
Iteracions = 65
Solució = 9.0621     -4.3978
tolF = 237.0615
tolZ = 5.105
Iteracions = 66
Solució = 6.7429     -1.7636
tolF = 54.4439
tolZ = 3.5097
Iteracions = 67
Solució = 9.8561     -0.1844
tolF = 21.0993
tolZ = 3.4909
Iteracions = 68
Solució = 9.1749       1.285
tolF = 19.5859
tolZ = 1.6196
Iteracions = 69
Solució = 8.9257     -0.1491
tolF = 18.6809
tolZ = 1.4556
Iteracions = 70
Solució = 9.1252      1.3211
tolF = 19.9187
tolZ = 1.4837
Iteracions = 71
Solució = 8.9116   -0.084967
tolF = 17.9057
tolZ = 1.4222
Iteracions = 72
Solució = 9.1279      1.4631
tolF = 22.145
tolZ = 1.5632
Iteracions = 73
Solució = 8.8638      0.1444
tolF = 15.7387
tolZ = 1.3449
Iteracions = 74
Solució = 9.1291      2.2418
tolF = 40.9062
tolZ = 2.1141
Iteracions = 75
Solució = 8.5146     0.92948
tolF = 15.0589
tolZ = 1.4491
Iteracions = 76
Solució = 9.0529     -1.5714
tolF = 55.1532
tolZ = 2.5582
Iteracions = 77
Solució = 8.8191    -0.13786
tolF = 18.5209
tolZ = 1.4525
Iteracions = 78
Solució = 9.1327      1.3394
tolF = 20.2103
tolZ = 1.5102
Iteracions = 79
Solució = 8.9061    -0.05169
tolF = 17.5333
tolZ = 1.4095
Iteracions = 80
```

```
Solució = 9.1288      1.5459
tolF = 23.6118
tolZ = 1.613
Iteracions = 81
Solució = 8.8337     0.25719
tolF = 15.0137
tolZ = 1.3221
Iteracions = 82
Solució = 9.1254      2.9782
tolF = 68.8582
tolZ = 2.7366
Iteracions = 83
Solució = 8.0481      1.4387
tolF = 19.168
tolZ = 1.879
Iteracions = 84
Solució = 8.9731    0.046411
tolF = 16.5727
tolZ = 1.6716
Iteracions = 85
Solució = 9.1257      1.8494
tolF = 30.0422
tolZ = 1.8094
Iteracions = 86
Solució = 8.7086     0.59304
tolF = 14.1404
tolZ = 1.3238
Iteracions = 87
Solució = 9.09885     −41.4625
tolF = 16165.5773
tolZ = 42.0574
Iteracions = 88
Solució = −200.5866     −20.43772
tolF = 94012.7923
tolZ = 210.7369
Iteracions = 89
Solució = −96.8188     −10.2078
tolF = 14519.009
tolZ = 104.2709
Iteracions = 90
Solució = −45.4812      −5.0847
tolF = 2809.0427
tolZ = 51.5926
Iteracions = 91
Solució = −20.1529      −2.5157
tolF = 632.4546
tolZ = 25.4582
Iteracions = 92
Solució = −7.7886     −1.2625
tolF = 148.1421
tolZ = 12.4277
Iteracions = 93
Solució = −1.9966     −1.3062
tolF = 37.1229
tolZ = 5.7921
Iteracions = 94
Solució = 0.40874      1.9669
tolF = 12.5242
tolZ = 4.0619
Iteracions = 95
Solució = 1.2743     0.81356
tolF = 3.1573
tolZ = 1.442
Iteracions = 96
```

```
Solució = 0.94452      1.0636
tolF = 0.85014
tolZ = 0.41389
Iteracions = 97
Solució = 1.0158     0.98564
tolF = 0.21295
tolZ = 0.10566
Iteracions = 98
Solució = 0.99639      1.0039
tolF = 0.053348
tolZ = 0.026678
Iteracions = 99
Solució = 1.001      0.9991
tolF = 0.013338
tolZ = 0.0066659
Iteracions = 100
Solució = 0.99977      1.0002
tolF = 0.0033348
tolZ = 0.0016674
Iteracions = 101
Solució = 1.0001     0.99994
tolF = 0.0008337
tolZ = 0.00041684
Iteracions = 102
Solució = 0.99999           1
tolF = 0.00020843
tolZ = 0.00010421
Iteracions = 103
Solució = 1           1
tolF = 5.2107e-05
tolZ = 2.6053e-05
Iteracions = 104
Solució = 1           1
tolF = 1.3027e-05
tolZ = 6.5133e-06
Iteracions = 105
Solució = 1           1
tolF = 3.2567e-06
tolZ = 1.6283e-06
Iteracions = 106
Solució = 1           1
tolF = 8.1417e-07
tolZ = 4.0708e-07
Iteracions = 107
Solució = 1           1
tolF = 2.0354e-07
tolZ = 1.0177e-07
Iteracions = 108
Solució = 1           1
tolF = 5.0885e-08
tolZ = 2.5443e-08
Iteracions = 109
Solució = 1           1
tolF = 1.2721e-08
tolZ = 6.3607e-09
Iteracions = 110
Solució = 1           1
tolF = 3.1803e-09
tolZ = 1.5902e-09
Iteracions = 111
Solució = 1           1
tolF = 7.9508e-10
tolZ = 3.9754e-10
Iteracions = 112
```

```
Solució = 1          1
tolF = 1.9877e-10
tolZ = 9.9385e-11
Iteracions = 113
Solució = 1          1
tolF = 4.9692e-11
tolZ = 2.4846e-11
Iteracions = 114
Solució = 1          1
tolF = 1.2422e-11
tolZ = 6.2114e-12
Iteracions = 115
Solució = 1          1
tolF = 3.1034e-12
tolZ = 1.5527e-12
Iteracions = 116
Solució = 1          1
tolF = 7.7523e-13
tolZ = 3.8793e-13
Iteracions = 117
Solució = 1          1
tolF = 1.935e-13
tolZ = 9.6903e-14
Iteracions = 118
Solució = 1          1
tolF = 4.9003e-14
tolZ = 2.4188e-14
Iteracions = 119
Solució = 1          1
tolF = 1.2561e-14
tolZ = 6.1254e-15
Iteracions = 120
Solució = 1          1
tolF = 2.5121e-15
tolZ = 1.5701e-15
Iteracions = 121
Solució = 1   1
tolF = 0
tolZ = 3.1402e-16
Iteracions = 122
Solució = 1   1
tolF = 0
tolZ = 3.1402e-16
Iteracions = 122
```

```
array2table(valor(1:k,:), "VariableNames", {'x', 'y'})
```

ans = 123×2 table

	x	y
1	2.193439415...	3.020466468...
2	2.193439415...	3.020466468...
3	2.193439415...	3.020466468...
4	2.193439415...	3.020466468...
5	2.193439415...	3.020466468...
6	2.193439415...	3.020466468...

	x	y
7	2.193439415...	3.020466468...
8	2.193439415...	3.020466468...
9	2.193439415...	3.020466468...
10	2.193439415...	3.020466468...
11	2.193439415...	3.020466468...
12	2.193439415...	3.020466468...
13	2.193439415...	3.020466468...
14	2.193439415...	3.020466468...

⋮

```
decimalsCorrectes = min(abs(valorBo' - valor(k,:)))
```

```
decimalsCorrectes =
          1.19343941553723
```

```
%Amb 123 iteracions no obtenim cap decimal correcte i veiem que
%ens troba la segona solució al sistema que es troba al punt [1,1]. Veiem
%que és un mètode convergent cap a la solució [1,1] a partir del valor inicial [2,3]
```

1e) Fent ús del mètode de **Gauss-Seidel** i prenent $x_0 = (2, 3)$. Presenteu les iteracions en una taula.

```
valor = zeros(200,2);
for k = 1:200
    valor(k,:) = newtonGaussSeidel(F,JF,x0,k,tolF,tolX);
    if k > 1 && valor(k,1) == valor(k-1,1) && valor(k,2) == valor(k-1,2)
        break;
    end
end
```

```
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 1
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 2
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 3
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 4
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
```

```
Iteracions = 5
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 6
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 7
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 8
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 9
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 10
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 11
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 12
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 13
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 14
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 15
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 16
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 17
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 18
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 19
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 20
Solució = 2.1934      3.0204
tolF = 3.5527e-15
tolZ = 4.5297e-16
```

```
Iteracions = 21
Solució = 2.1934      3.0205
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 22
Solució = 2.1935      3.0202
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 23
Solució = 2.1932      3.0213
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 24
Solució = 2.1943      3.0177
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 25
Solució = 2.1904      3.0298
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 26
Solució = 2.2035      2.9892
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 27
Solució = 2.1598      3.1259
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 28
Solució = 2.3073       2.671
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 29
Solució = 1.8217      4.3444
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 30
Solució = 3.7057     -2.2097
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 31
Solució = -0.32802     -1.1419
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 32
Solució = 0.86301      1.1722
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 33
Solució = 1.0429       1.016
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 34
Solució = 1.0038      1.0008
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 35
Solució = 1.0002           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 36
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
```

```
Iteracions = 37
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 38
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 39
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 40
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 41
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 42
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 43
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 44
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 45
Solució = 1           1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 46
Solució = 1  1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 47
Solució = 1  1
tolF = 3.5527e-15
tolZ = 4.5297e-16
Iteracions = 47
```

```
array2table(valor(1:k,:), "VariableNames", {'x', 'y'})
```

ans = 48×2 table

	x	y
1	2.193439415...	3.020466468...
2	2.193439415...	3.020466468...
3	2.193439415...	3.020466468...
4	2.193439415...	3.020466468...
5	2.193439415...	3.020466468...
6	2.193439415...	3.020466468...

	x	y
7	2.193439415...	3.020466468...
8	2.193439415...	3.020466468...
9	2.193439415...	3.020466468...
10	2.193439415...	3.020466468...
11	2.193439415...	3.020466467...
12	2.193439415...	3.020466468...
13	2.193439415...	3.020466466...
14	2.193439413...	3.020466472...

⋮

```
decimalsCorrectes = min(abs(valorBo' - valor(k,:)))
```

```
decimalsCorrectes =
          1.19343941553723
```

```
%Amb 48 iteracions no obtenim cap decimal correcte tot i així veiem que
%ens troba la segona solució al sistema que es troba al punt [1,1].
% És un mètode convergent cap a la solució [1,1] a partir del valor inicial [2,3]
```

1f) Fent ús de **Fsolve** prenent $x_0 = (2, 3)$. Presenteu les iteracions en una taula.

```
%Sense opcions
x0=[2;3];
[x, ~] = fsolve(@myfun,x0);
```

```
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
```

```
myfun(x);

valorBo = x;

%Mètode amb iteracions i opcions configurables
clear vars
options = optimoptions('fsolve','Display','iter','StepTolerance',5e-15,'OptimalityTole
```

```
options =
  fsolve options:

   Options used by current Algorithm ('trust-region-dogleg'):
   (Other available algorithms: 'levenberg-marquardt', 'trust-region')
```

```
  Set properties:
                    Display: 'iter'
       OptimalityTolerance: 5e-15
             StepTolerance: 5e-15

  Default properties:
                  Algorithm: 'trust-region-dogleg'
             CheckGradients: 0
   FiniteDifferenceStepSize: 'sqrt(eps)'
       FiniteDifferenceType: 'forward'
          FunctionTolerance: 1e-06
    MaxFunctionEvaluations: '100*numberOfVariables'
              MaxIterations: 400
                  OutputFcn: []
                    PlotFcn: []
   SpecifyObjectiveGradient: 0
                   TypicalX: 'ones(numberOfVariables,1)'
                UseParallel: 0

  Show options not used by current Algorithm ('trust-region-dogleg')
```

```
x0 = [2,3];
[~, fval] = fsolve(@myfun,x0,options)
```

```
                                           Norm of      First-order   Trust-region
 Iteration  Func-count     f(x)            step         optimality    radius
     0          3               5                              26             1
     1          6      0.00265129        0.196419           0.346             1
     2          9     2.87311e-08      0.00736061         0.00132             1
     3         12     3.55929e-18     2.08213e-05        1.62e-08             1
     4         15     1.26218e-29     2.21674e-10         3.6e-14             1

Equation solved, inaccuracy possible.

The vector of function values is near zero, as measured by the value
of the function tolerance. However, the last step was ineffective.

<stopping criteria details>
fval = 1×2
                         0      3.5527136788005e-15
```

1g) Fent ús del **mètode de la iteració simple** prenent una funció d'iteració G adient. Verifiqueu la convergència a priori del vostre mètode abans de calcular les iteracions. Preneu $x_0 = (2, 3)$ si és possible, altrement raoneu la el·lecció de x_0. Presenteu les iteracions en una taula.

```
format longG;
G1 = @(x,y)(x^2)/10 + (y^2)/10 + 0.8; %z1^2/10 + z2^2/10 +8/10
G2 = @(x,y) sqrt((-x+10*y-8)/x); %(x*y^2)/10 + x/10 + 0.8; % converg

x0 = [2;3];

for k=1:40
    resI(k).x0(1) = G1(x0(1),x0(2));
    resI(k).x0(2) = G2(x0(1),x0(2));
    %disp([k,x0']);
    resI(k).decimalsCorrectes = min(abs(valorBo'-x0'));
end
```

```
tolF = 10^(-6);

resI
```

resI = 1×40 struct

Fields	x0	decimalsCor...
1	[2.1,3.1...	0.020466...
2	[2.1,3.1...	0.020466...
3	[2.1,3.1...	0.020466...
4	[2.1,3.1...	0.020466...
5	[2.1,3.1...	0.020466...
6	[2.1,3.1...	0.020466...
7	[2.1,3.1...	0.020466...
8	[2.1,3.1...	0.020466...
9	[2.1,3.1...	0.020466...
10	[2.1,3.1...	0.020466...
11	[2.1,3.1...	0.020466...
12	[2.1,3.1...	0.020466...
13	[2.1,3.1...	0.020466...
14	[2.1,3.1...	0.020466...

⋮

```
disp(['Solució = ',num2str(x0')])
```

Solució = 2   3

```
disp(['tolF = ',num2str(tolF)])
```

tolF = 1e-06

```
disp(['Iteracions = ',num2str(k)])
```

Iteracions = 40

```
%Si aïllem G1 i de G2 de la forma que veiem a continuació:
% G1 = @(x,y)(x^2)/10 + (y^2)/10 + 0.8;
% G2 = @(x,y) (x*y^2)/10 + x/10 + 0.8;
% el mètode és convergent a partir del punt[2,3] a la solució [1,1].
%
% Per trobar la descomposició adient per G i que el mètode sigui convergent a la soluc
% hem de seguir provant possibles combinacions per aïllar G1 i G2 correctament.
%
% Finalment les aïllacions correctes són:
```

```
%G1 = @(x,y)(x^2)/10 + (y^2)/10 + 0.8;
%G2 = @(x,y)sqrt((-x+10*y-8)/x)

%Un cop hem trobat la combinació correcta, obtenim 6 xifres decimals
%correctes i efectivament és un mètode convergent.
```

1h) Conclusions: comenta les diferències trobades. Quants decimals correctes obteniu?

El nombre de decimals correctes obtinguts es troba a cada apartat respecivament.

EXERCICI 2 - Derivació i ajust de corbes

L'any 2009 (a Berlín) Usain Bolt va situar el record dels 100m en 9.58s. Les dades de la carrera són les següents

r	0	10	20	30	40	50	60	70	80	90	100
t(r)	0	1.85	2.89	3.78	4.64	5.49	6.31	7.11	7.92	8.74	9.58

on la primera fila és la distància recorreguda en metres i la segona el temps emprat en segons

(font: NBC, http://www.universalsports.com/news/article/newsid=385633.html).

Calculeu una aproximació de la velocitat i l'acceleració en la carrera a partir de les dades de la taula.

```
%dades
r  = [0:10:100];
tr = [0 1.85 2.89 3.78 4.64 5.49 6.31 7.11 7.92 8.74 9.58];
%disp([r;tr]')
dades = array2table([r;tr]', 'VariableNames', {'Distància (m)', 'Temps (s)'})
```

dades = 11×2 table

	Distància (m)	Temps (s)
1	0	0
2	10	1.85
3	20	2.89
4	30	3.78
5	40	4.64
6	50	5.49
7	60	6.31
8	70	7.11
9	80	7.92
10	90	8.74
11	100	9.58

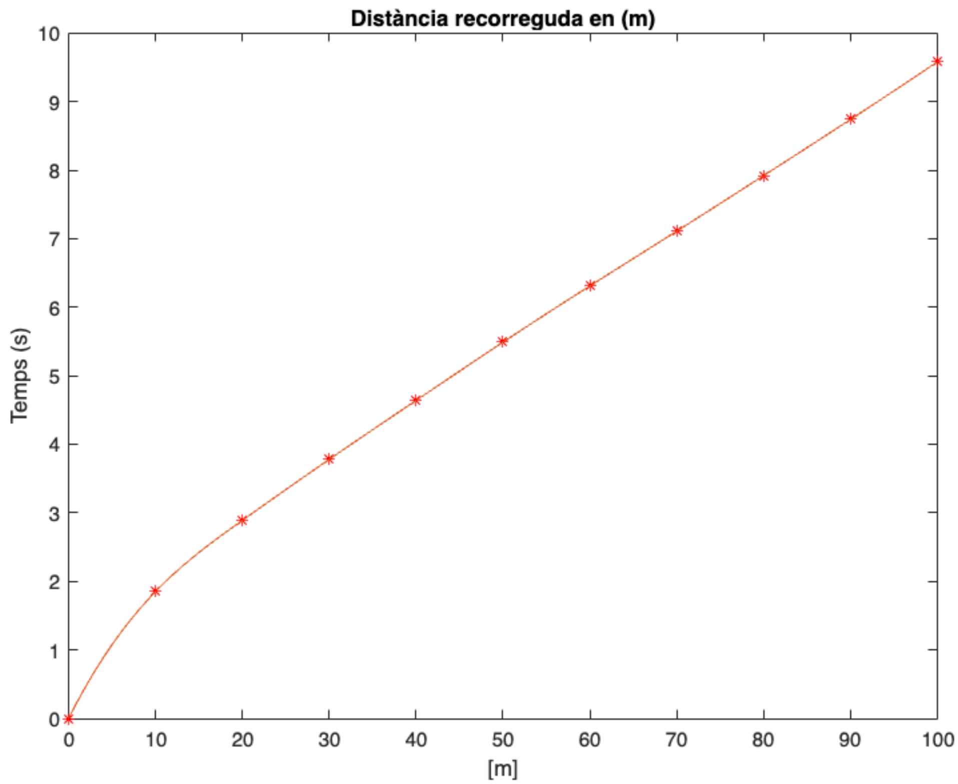
```
%càlcul aproximació [v,a] on fem servir per cadascun la fórmula enrere
v = [];
a = [];
for i=2:11
    v(i)=(r(i)-r(i-1))/(tr(i)-tr(i-1));
    a(i)=(v(i)-v(i-1))/(tr(i)-tr(i-1));
end

resultat = array2table([r;tr;v;a]', 'VariableNames',{'Distància (m)', 'Temps (s)', 'Ve
```

resultat = 11×4 table

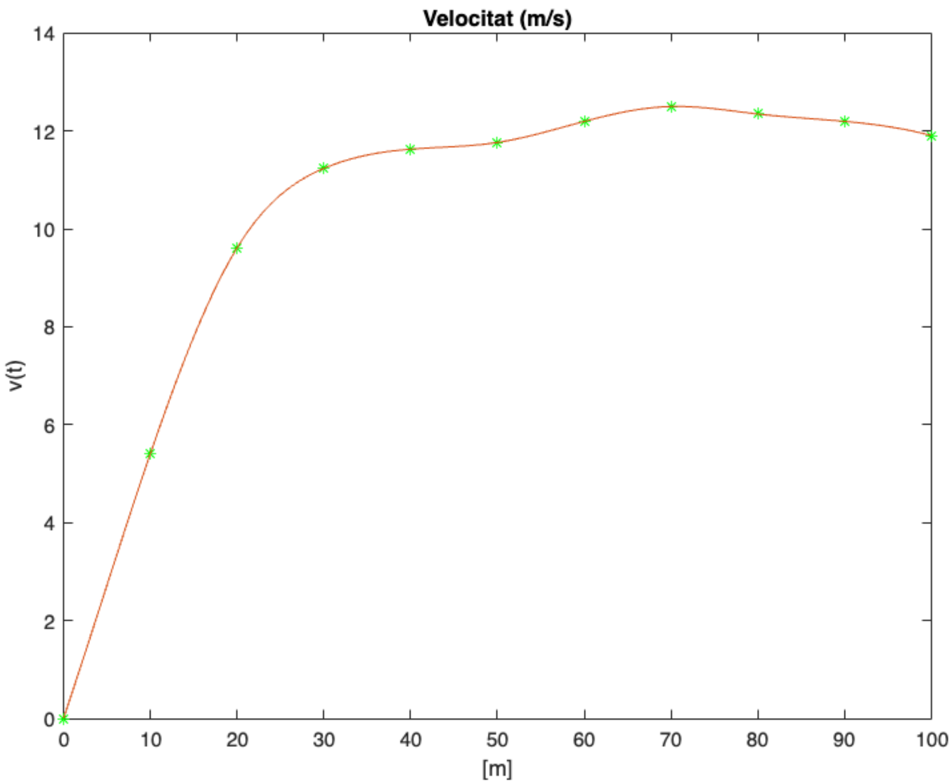
	Distància (m)	Temps (s)	Velocitat (m/s)	Acceleració (m/s^2)
1	0	0	0	0
2	10	1.85	5.40540540540541	2.9218407596786
3	20	2.89	9.61538461538461	4.04805693267232
4	30	3.78	11.2359550561798	1.82086566381479
5	40	4.64	11.6279069767442	0.455758047167916
6	50	5.49	11.7647058823529	0.160939888951466
7	60	6.31	12.1951219512195	0.524897644959254
8	70	7.11	12.5	0.381097560975585
9	80	7.92	12.3456790123457	-0.190519737844821
10	90	8.74	12.1951219512195	-0.183606172105092
11	100	9.58	11.9047619047619	-0.345666721973336

Feu una representació gràfica dels valors obtinguts.

```
%representació de la distància
novaX = 0:1:100;
S = spline(r,tr,novaX);
plot(r,tr,'r*',novaX,S)

xlabel('[m]')
ylabel('Temps (s)');
title('Distància recorreguda en (m)')
hold off
```




```
%representació de la velocitat
Sv = spline(r,v,novaX);
plot(r,v,'g*',novaX,Sv)

xlabel('[m]')
ylabel('v(t)');
title('Velocitat (m/s)')
hold off
```




```
%representació de l'acceleració
Sa = spline(r,a,novaX);
plot(r,a,'b*',novaX,Sa)

xlabel('[m]')
ylabel('a(t)');
title('Acceleració (m/s^2)')
hold off
```

EXERCICI 3 - Interpolació

Obteniu una corba que ajusti les dades de la taula següent per als diferents mètodes explicats en les classes: polinomica, spline lineal, spline cúbic.

x	0.9	1.3	1.9	2.1	2.6	3.0	3.9	4.4	4.7	5.0	6.0	7.0	8.0	9.2	10.5	11.3	11.6	12.0	12.6	13.0	13.3
$f(x)$	1.3	1.5	1.85	2.1	2.6	2.7	2.4	2.15	2.05	2.1	2.25	2.3	2.25	1.95	1.4	0.9	0.7	0.6	0.5	0.4	0.25

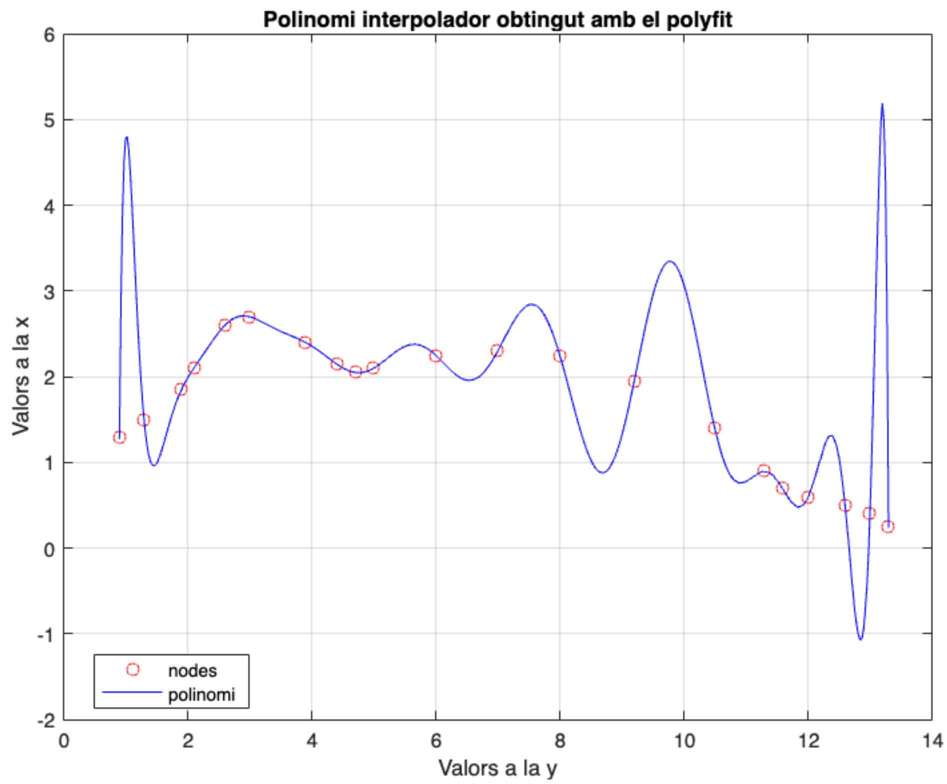
```
x = [0.9,1.3,1.9,2.1,2.6,3.0,3.9,4.4,4.7,5.0,6.0,7.0,8.0,9.2,10.5,11.3,11.6,12.0,12.6,
y = [1.3, 1.5, 1.85, 2.1, 2.6, 2.7, 2.4, 2.15 2.05, 2.1, 2.25, 2.3, 2.25, 1.95, 1.4, 0
%scatter(x, y)
%disp([x,y]);
%plot(x,y);
```

```
%interpolació polinomica fent servir polyinterp
z  = [0:20];
pi = polyinterp(x,y,z);
plot(z,pi,x,y,'*r','LineWidth',2)
grid on
ylabel('Valors a la y')
xlabel('Valors a la x')
title('Polinomi interpolador a partir de polyinterp')
xlim([0 14])
ylim([0 3])
hold off

%spline lineal
z  = [0:20];
pz = interp1(x,y,z);
plot(z,pz,x,y,'*r','LineWidth',2)

grid on
ylabel('Valors a la y')
xlabel('Valors a la x')
title('Spline lineal (interpolació lineal a trossos)')
hold off

%spline cúbica
x = [0.9,1.3,1.9,2.1,2.6,3.0,3.9,4.4,4.7,5.0,6.0,7.0,8.0,9.2,10.5,11.3,11.6,12.0,12.6,
y = [1.3, 1.5, 1.85, 2.1, 2.6, 2.7, 2.4, 2.15 2.05, 2.1, 2.25, 2.3, 2.25, 1.95, 1.4, 0

z  = [0:20];
pc = interp1(x,y,z, 'pchip');
plot(z,pc,x,y,'*r','LineWidth',2)

grid on
ylabel('Valors a la x')
xlabel('Valors a la y')
title('Cúbica a trossos - Spline preserving')
xlim([0 14])
ylim([0 3])
hold off

%fent servir polyfit
%disp([x; y]);
grau = length(x)-1;
p = polyfit(x,y,grau);
```

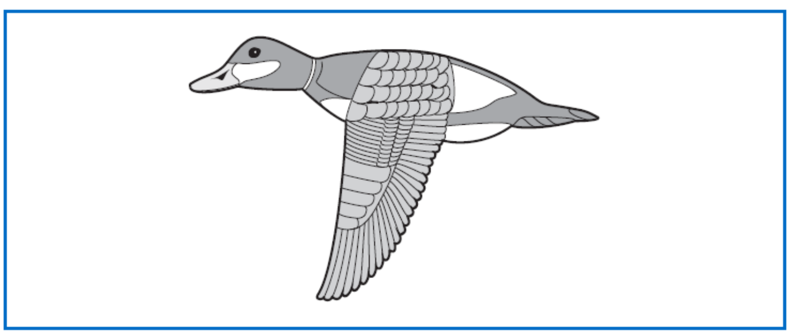
```
Warning: Polynomial is badly conditioned. Add points with distinct X values, reduce the degree of
the polynomial, or try centering and scaling as described in HELP POLYFIT.
```

```
u = linspace(x(1),x(end),500);
v = polyval(p,u);
plot(x,y,'ro',u,v,'b-')
grid on
```

```
ylabel('Valors a la x')
xlabel('Valors a la y')
title('Polinomi interpolador obtingut amb el polyfit')
legend('nodes','polinomi','Location','best')
hold off
```



Les dades són del llibre Numerical Analysis, Burden&Faires, per la imatge



EXERCICI 4 - Integració numèrica

Doneu una aproximació de l'area de la regió delimitada per la vostra ma. Referència: https://es.mathworks.com/moler/chapters.html

Apartat 1.

1. Obteniu una imatge de la vostra ma. Seguiu les indicacions de l'exercici 3.4 de la pàgina 20 del capítol 3, "Interpolation"de Cleve Moler.

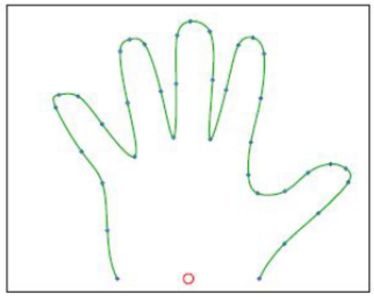


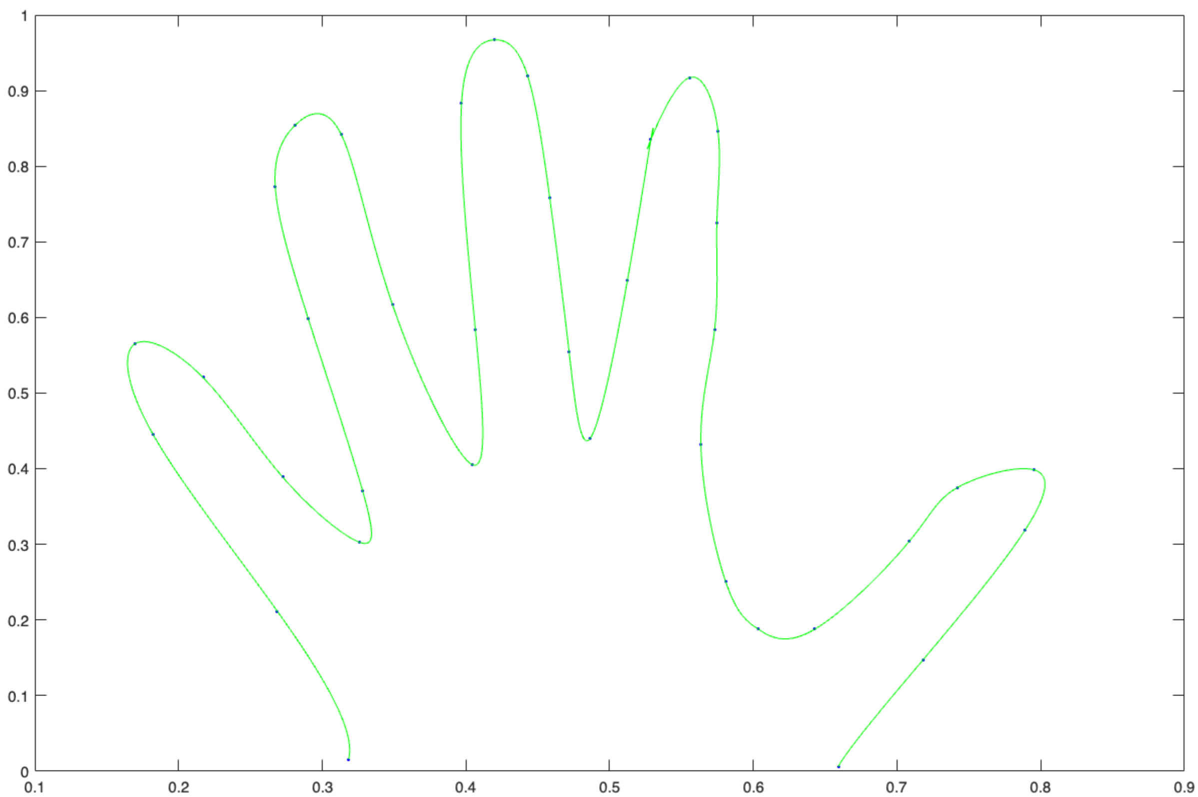
Figure 3.11. *A hand.*

Figura 1:

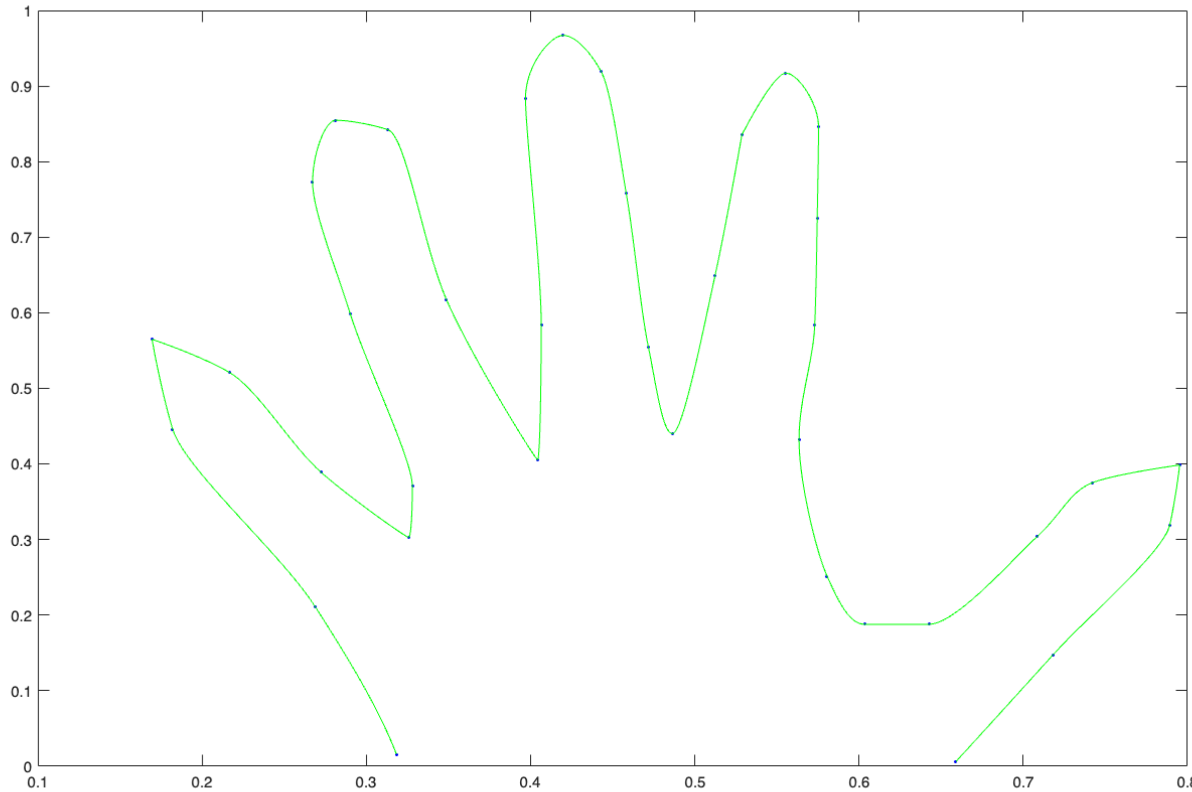
```
figure('position', get(0,'ScreenSize'));
axes('position', [0 0 1 1]);

legend('location','westoutside');
legend('hide');
```


```
%[x,y] = ginput;
```

```
puntsX = x;
puntsY = y;

puntsNousX = unnamed(:,1); %és una variable que es troba al fitxer maPunts.mat
puntsNousY = unnamed(:,2);

n = length(puntsNousX);
s = (1:n)';
t = (1:.05:n)';
u = splinetx(s,puntsNousX,t);
v = splinetx(s,puntsNousY,t);
clf reset
plot(puntsNousX,puntsNousY,'b.',u,v,'g-');
```



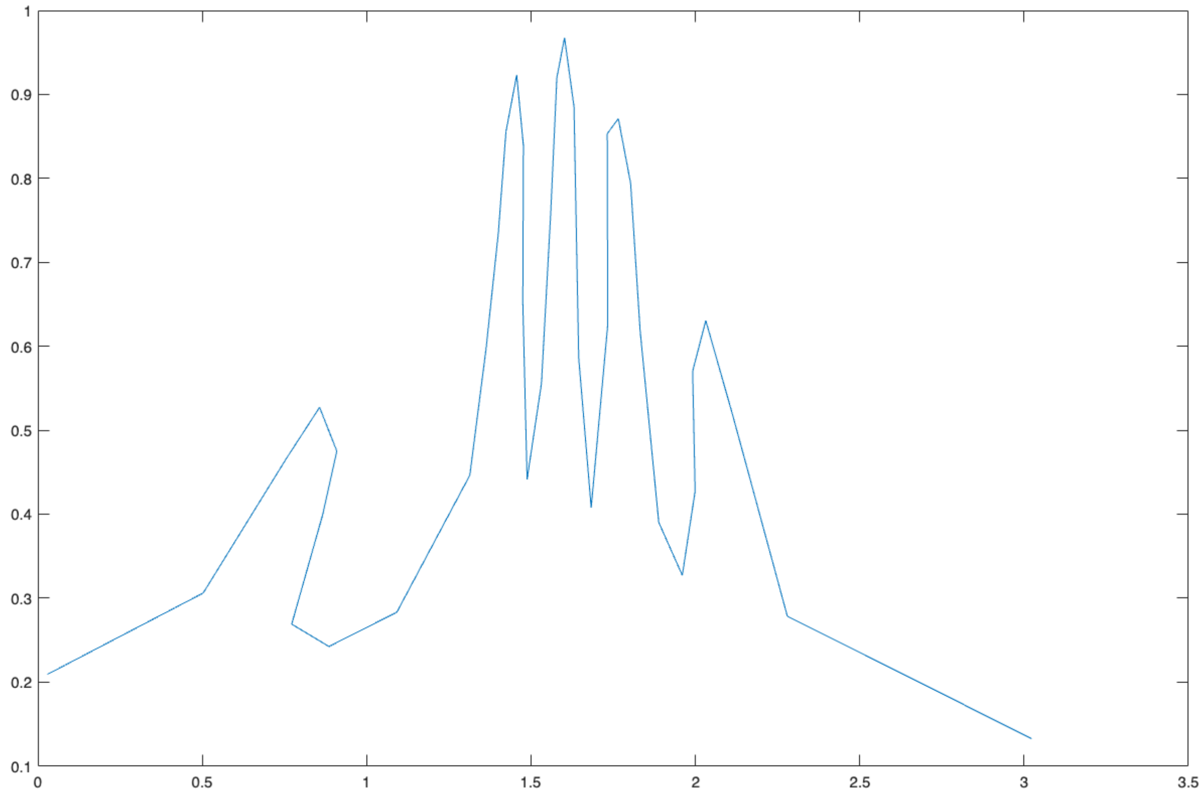

```
u2 = pchiptx(s,puntsNousX,t);
v2 = pchiptx(s,puntsNousY,t);
clf reset
plot(puntsNousX,puntsNousY,'b.',u2,v2,'g-');
```



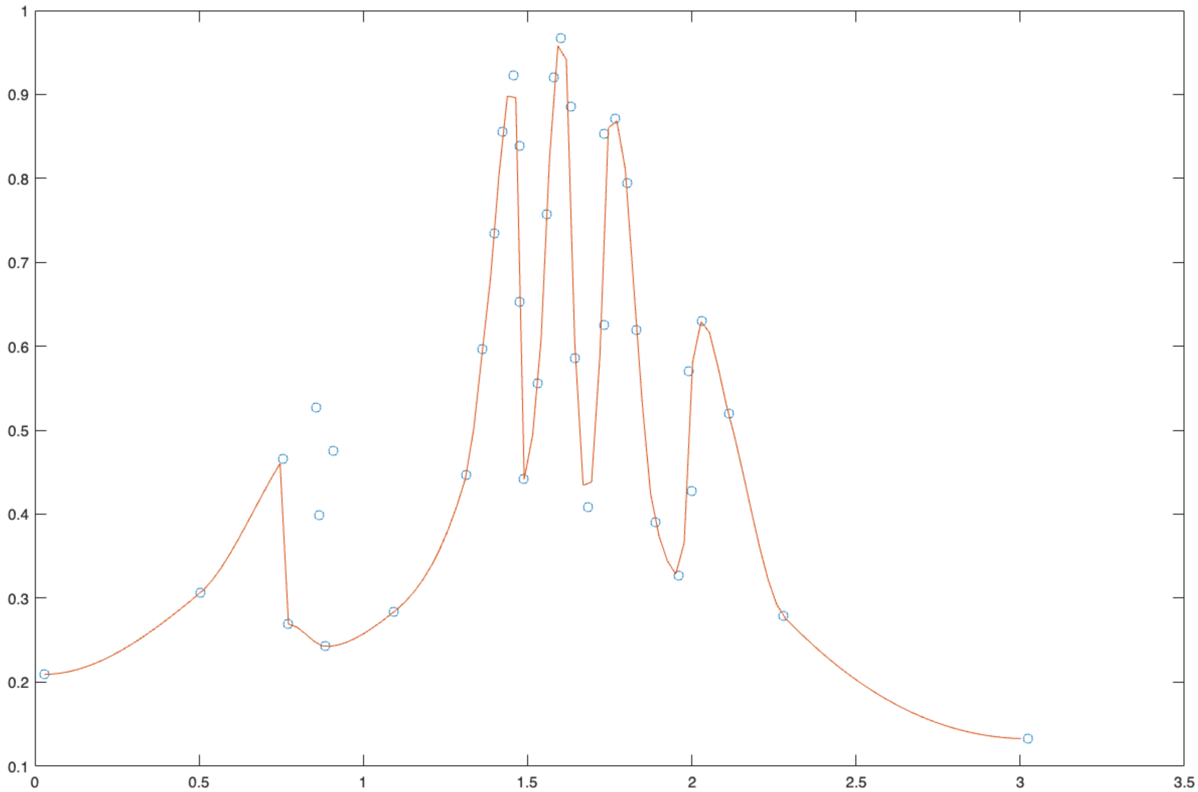
Apartat 2.

2. Obteniu per interpolació $2D$ la corba que delimita la imatge de la vostra ma seguint els indicacions de l'exercici 3.4 i l'exercici 3.5 de les pàgines $20 - 21 - 22$. Responeu les preguntes que us formulen en els dos exercicis.

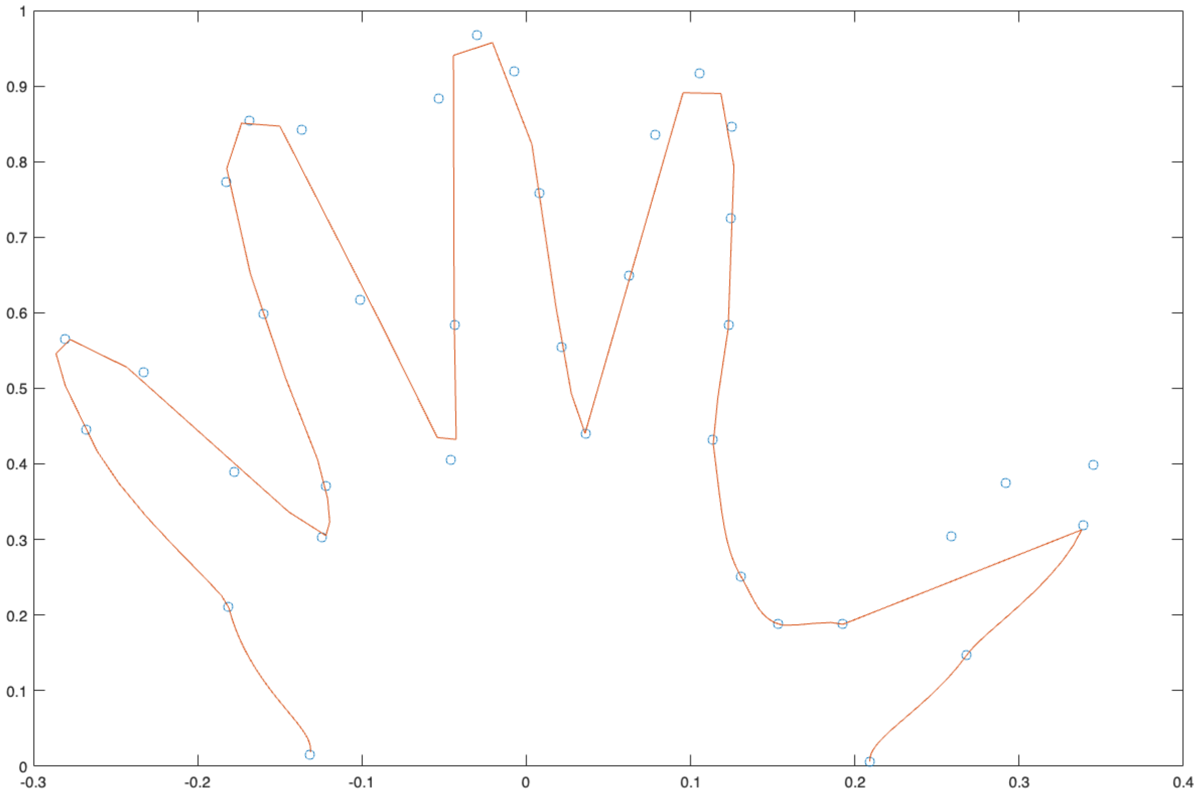
```
%punt vermell al mig del palmell
x0 = 0.45;
y0 = 0.0;

n = length(puntsNousX);
s = (1:n)';
t = (1:.05:n)';
u = splinetx(s,puntsNousX,t);
v = splinetx(s,puntsNousY,t);

puntsNousX = puntsNousX - x0;
puntsNousY = puntsNousY - y0;
theta = atan2(puntsNousY,puntsNousX);
r = sqrt(puntsNousX.^2+puntsNousY.^2);
plot(theta,r)
```




```
t = (theta(1):1/length(theta):theta(end))';
p = pchiptx(theta,r,t);
s = splinetx(theta,r,t);
plot(theta,r,'o',t,[p s],'-')
```




```
plot(puntsNousX,puntsNousY,'o',p.*cos(t),p.*sin(t),'-', s.*cos(t),s.*sin(t),'-')
```



3.4 Pchip vs Spline?

Prefereixo en aquest cas l'spline ja que tal i com es pot veure obtenim una imatge molt més definida i més suau de la mà. És a dir, la derivada segona de l'spline és continua i podem veure que si les dades consisteixen en valors d'una funció contínua l'spline produeix un resultat més precís. Conclusió extreta de: https://es.mathworks.com/content/dam/mathworks/mathworks-dot-com/moler/interp.pdf (apartat 3.7)

La figura que es mostra està feta amb l'spline.

3.5 Quina aproximació és millor? Aquesta o la del darrer exercici?

L'aproximació més bona és que es realiza a l'apartat anterior.

Apartat 3.

3. Què tan gran és la teva mà? Calcula l'àrea que ocupa la teva mà. Segueix les indicacions i respon les preguntes de l'exercici 6.23 de les pàgines 19 – 20 del capítol 6 "Quadrature" de Cleve Moler.

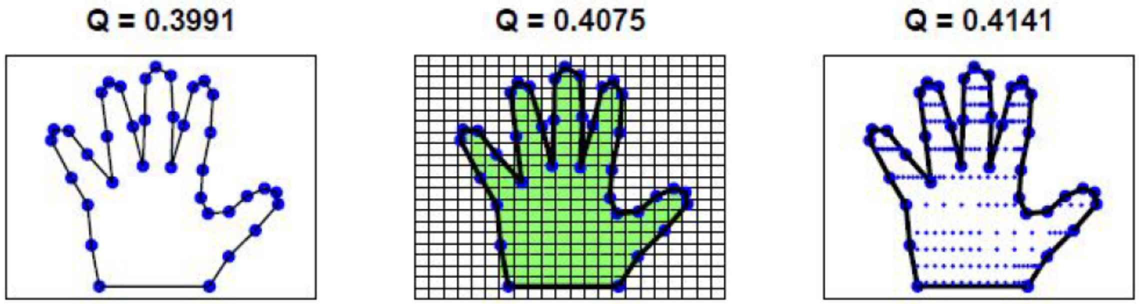


Figure 6.5. *The area of a hand.*

Figura 2: Moler - chpater 6 - exercici 6.23

```
%Podem calcular l'àrea de 3 formes possibles:
%àrea del polígon
area = (puntsNousX'*puntsNousY([2:n 1]) - puntsNousX([2:n 1])'*puntsNousY)/2;

%simple quadratura
[u,v] = meshgrid(min(puntsNousX):0.1:max(puntsNousX),min(puntsNousY):0.1:max(puntsNous
k = inpolygon(u,v,puntsNousX,puntsNousY);
area2 = 0.1^2*nnz(k);

%R^2 quadratura adaptativa
area3 = dblquad(@(u,v)chi(u,v,puntsNousX,puntsNousY),min(puntsNousX),max(puntsNousX),m
%si posem epsilon peta i tarda molt
```

```
%mostrem els valors de l'àrea en una taula
array2table([area,area2,area3], 'VariableName',{'Àrea del polígon', 'Simple Quadratura
```

ans = 1×3 table

	Àrea del polígon	Simple Quadratura	R^2 Quadratura Adaptativa
1	0.25482822	0.2	0.254580717019077

```
function F = myfun(x)
    F(1) = x(1)^2-10*x(1) + x(2)^2 + 8; %x(1) = x, x(2) = y
    F(2) = x(1)*x(2)^2 + x(1) - 10*x(2) + 8;
end
```

```
% Mètode de Newton per a funcions de dues variables
function valor = newtonModificat(F, DF, Z, n, tolF, tolZ, iteracions)
    fz     = F(Z(1),Z(2));
    jacobianafz  = DF(Z(1),Z(2));

    valorf = Z(1);
    valorz = Z(2);

    k = 0;
    while (k < n && (valorf > tolF || valorz > tolZ)) %si convergeix, concloem
        Y  = jacobianafz \ (-fz); %soluciona per y, o també fent la Jacobiana de F(x)

        vz = Z;
        Z  = vz + Y; %actualitzem estimació arrel

        k = k+1;
        if mod(k , iteracions) == 0 jacobianafz = DF(Z(1),Z(2)); end

        fz     = F(Z(1),Z(2));

        valorz = norm(Y);
        valorf = norm(fz);
    end

    valor = Z;

    disp(['Solució = ',num2str(Z')])
    disp(['tolF = ',num2str(valorf)])
    disp(['tolX = ',num2str(valorz)])
    disp(['Iteracions = ',num2str(k)])
end

% Mètode de Jacobi per a funcions de dues variables
function valor = newtonJacobi(F, DF, Z, iteracions, tolF, tolZ)
    fz  = F(Z(1),Z(2));
    jacobianafz = diag(diag(DF(Z(1),Z(2))));

    vf = Z(1);
    vz = Z(2);

    k  = 0;
    while (k < iteracions && (vf > tolF || vz > tolZ))
        Y  = jacobianafz \ (-fz); %soluciona per y, o també fent la Jacobiana de F(x)

        pz = Z;
        Z  = pz + Y;  %actualitzem estimació arrel

        jacobianafz = diag(diag(DF(Z(1),Z(2))));
        fz    = F(Z(1),Z(2));
```

```
        vz = norm(Y);
        vf = norm(fz);

        k  = k + 1;
    end

    valor = Z;

    disp(['Solució = ',num2str(Z')]);
    disp(['tolF = ',num2str(vf)]);
    disp(['tolZ = ',num2str(vz)]);
    disp(['Iteracions = ',num2str(k)]);

end

% Mètode de Gauss Seidel per a funcions de dues variables
function valor = newtonGaussSeidel(F, DF, Z, iteracions, tolF, tolZ)
    fz    = F(Z(1),Z(2));
    jacobianafz = tril(DF(Z(1),Z(2)));

    valorf = Z(1);
    valorz = Z(2);

    k = 0;
    while (k < iteracions && (valorf > tolF || valorz > tolZ))
        y  = jacobianafz \ (-fz);

        pz  = Z;
        Z   = pz + y;

        jacobianafz    = tril(DF(Z(1),Z(2)));
        fz       = F(Z(1),Z(2));

        valorz = norm(y);
        valorf = norm(fz);

        k = k + 1;
    end

    valor = Z;

    disp(['Solució = ',num2str(Z')]);
    disp(['tolF = ',num2str(tolF)]);
    disp(['tolZ = ',num2str(tolZ)]);
    disp(['Iteracions = ',num2str(k)]);
end


function valor = iteracioSimple(G, F, Z, iteracions, tolerancia)
    FZ   = F(Z(1),Z(2));
```

```
    tolF = 10^(-6);
    tolZ = 10^(-6);

    k = 0;
    while (k < iteracions && (tolF > tolerancia || tolZ > tolerancia))

        pz   = Z;
        Z    = G(Z(1),Z(2));

        tolF = norm(F(Z(1),Z(2)));
        tolZ = norm(Z-pz);

        k = k + 1;
    end

    valor = Z;

    disp(['Solució = ',num2str(Z')]);
    disp(['tolF = ',num2str(tolF)]);
    disp(['tolZ = ',num2str(tolZ)]);
    disp(['Iteracions = ',num2str(k)]);
end


function v = polyinterp(x,y,u)
%POLYINTERP  Polynomial interpolation.
%   v = POLYINTERP(x,y,u) computes v(j) = P(u(j)) where P is the
%   polynomial of degree d = length(x)-1 with P(x(i)) = y(i).
%   Copyright 2014 Cleve Moler
%   Copyright 2014 The MathWorks, Inc.
% Use Lagrangian representation.
% Evaluate at all elements of u simultaneously.
n = length(x);
v = zeros(size(u));
for k = 1:n
   w = ones(size(u));
   for j = [1:k-1 k+1:n]
      w = (u-x(j))./(x(k)-x(j)).*w;
   end
   v = v + w*y(k);
end
end


%la mà
function v = splinetx(x,y,u)
%SPLINETX  Textbook spline function.
%  v = splinetx(x,y,u) finds the piecewise cubic interpolatory
%  spline S(x), with S(x(j)) = y(j), and returns v(k) = S(u(k)).
```

```
%
%  See SPLINE, PCHIPTX.

%   Copyright 2014 Cleve Moler
%   Copyright 2014 The MathWorks, Inc.

%  First derivatives

   h = diff(x);
   delta = diff(y)./h;
   d = splineslopes(h,delta);

%  Piecewise polynomial coefficients

   n = length(x);
   c = (3*delta - 2*d(1:n-1) - d(2:n))./h;
   b = (d(1:n-1) - 2*delta + d(2:n))./h.^2;

%  Find subinterval indices k so that x(k) <= u < x(k+1)

   k = ones(size(u));
   for j = 2:n-1
      k(x(j) <= u) = j;
   end

%  Evaluate spline

   s = u - x(k);
   v = y(k) + s.*(d(k) + s.*(c(k) + s.*b(k)));
end
function d = splineslopes(h,delta)
%  SPLINESLOPES  Slopes for cubic spline interpolation.
%  splineslopes(h,delta) computes d(k) = S'(x(k)).
%  Uses not-a-knot end conditions.

%  Diagonals of tridiagonal system

   n = length(h)+1;
   a = zeros(size(h)); b = a; c = a; r = a;
   a(1:n-2) = h(2:n-1);
   a(n-1) = h(n-2)+h(n-1);
   b(1) = h(2);
   b(2:n-1) = 2*(h(2:n-1)+h(1:n-2));
   b(n) = h(n-2);
   c(1) = h(1)+h(2);
   c(2:n-1) = h(1:n-2);

%  Right-hand side

   r(1) = ((h(1)+2*c(1))*h(2)*delta(1)+ ...
```

```
          h(1)^2*delta(2))/c(1);
   r(2:n-1) = 3*(h(2:n-1).*delta(1:n-2)+ ...
              h(1:n-2).*delta(2:n-1));
   r(n) = (h(n-1)^2*delta(n-2)+ ...
          (2*a(n-1)+h(n-1))*h(n-2)*delta(n-1))/a(n-1);

%  Solve tridiagonal linear system

   d = tridisolve(a,b,c,r);
end

function x = tridisolve(a,b,c,d)
%   TRIDISOLVE  Solve tridiagonal system of equations.
%     x = TRIDISOLVE(a,b,c,d) solves the system of linear equations
%     b(1)*x(1) + c(1)*x(2) = d(1),
%     a(j-1)*x(j-1) + b(j)*x(j) + c(j)*x(j+1) = d(j), j = 2:n-1,
%     a(n-1)*x(n-1) + b(n)*x(n) = d(n).
%
%   The algorithm does not use pivoting, so the results might
%   be inaccurate if abs(b) is much smaller than abs(a)+abs(c).
%   More robust, but slower, alternatives with pivoting are:
%     x = T\d where T = diag(a,-1) + diag(b,0) + diag(c,1)
%     x = S\d where S = spdiags([[a; 0] b [0; c]],[-1 0 1],n,n)

%   Copyright 2014 Cleve Moler
%   Copyright 2014 The MathWorks, Inc.

x = d;
n = length(x);

for j = 1:n-1
   mu = a(j)/b(j);
   b(j+1) = b(j+1) - mu*c(j);
   x(j+1) = x(j+1) - mu*x(j);
end

x(n) = x(n)/b(n);
for j = n-1:-1:1
   x(j) = (x(j)-c(j)*x(j+1))/b(j);
end
end

function v = pchiptx(x,y,u)
%PCHIPTX  Textbook piecewise cubic Hermite interpolation.
%  v = pchiptx(x,y,u) finds the shape-preserving piecewise cubic
%  interpolant P(x), with P(x(j)) = y(j), and returns v(k) = P(u(k)).
%
%  See PCHIP, SPLINETX.

%  First derivatives
```

```
   h = diff(x);
   delta = diff(y)./h;
   d = pchipslopes(h,delta);

%  Piecewise polynomial coefficients

   n = length(x);
   c = (3*delta - 2*d(1:n-1) - d(2:n))./h;
   b = (d(1:n-1) - 2*delta + d(2:n))./h.^2;

%  Find subinterval indices k so that x(k) <= u < x(k+1)

   k = ones(size(u));
   for j = 2:n-1
      k(x(j) <= u) = j;
   end

%  Evaluate interpolant

   s = u - x(k);
   v = y(k) + s.*(d(k) + s.*(c(k) + s.*b(k)));

end
% -------------------------------------------------------

function d = pchipslopes(h,delta)
%  PCHIPSLOPES  Slopes for shape-preserving Hermite cubic
%  interpolation.  pchipslopes(h,delta) computes d(k) = P'(x(k)).

%  Slopes at interior points
%  delta = diff(y)./diff(x).
%  d(k) = 0 if delta(k-1) and delta(k) have opposites signs
%         or either is zero.
%  d(k) = weighted harmonic mean of delta(k-1) and delta(k)
%         if they have the same sign.

   n = length(h)+1;
   d = zeros(size(h));
   k = find(sign(delta(1:n-2)).*sign(delta(2:n-1)) > 0) + 1;
   w1 = 2*h(k)+h(k-1);
   w2 = h(k)+2*h(k-1);
   d(k) = (w1+w2)./(w1./delta(k-1) + w2./delta(k));

%  Slopes at endpoints

   d(1) = pchipendpoint(h(1),h(2),delta(1),delta(2));
   d(n) = pchipendpoint(h(n-1),h(n-2),delta(n-1),delta(n-2));
end
% -------------------------------------------------------
```

```
function d = pchipendpoint(h1,h2,del1,del2)
%  Noncentered, shape-preserving, three-point formula.
   d = ((2*h1+h2)*del1 - h1*del2)/(h1+h2);
   if sign(d) ~= sign(del1)
      d = 0;
   elseif (sign(del1) ~= sign(del2)) & (abs(d) > abs(3*del1))
      d = 3*del1;
   end
end


%quadratura
function k = chi(u,v,x,y)
if all(size(u) == 1), u = u(ones(size(v)));
end
if all(size(v) == 1), v = v(ones(size(u)));
end
k = inpolygon(u,v,x,y);
end
```