

Notes generals

- Informació sobre el Taller:
 - Atenea (<http://atenea.upc.edu/moodle/course/view.php?id=33376>): calendari i aules, llibre, exàmens, normativa, instal·lació de Matlab
 - aules amb PC's lliures: <http://www.etseib.upc.edu/disponibilitat/disponibilitat.html>
- En aquestes pràctiques, podrem usar indistintament Octave o Matlab.
 - Matlab és un paquet comercial (requereix llicència: vegeu a Atenea com obtenir-la i instruccions per a instal·lar-lo)
 - Octave és programari lliure (es pot instal·lar i usar lliurement): <https://www.gnu.org/software/octave/>
 - Octave online: <https://octave-online.net> (més lent)
- per entrar al programa,
 - càrrega de Windows 7
 - Ctrl+Alt+Supr
 - *username@est* / *password*
 - Inici → Tots els programes → Progr. Docència → Matemàtiques → Taller Mat. → Matlab o Octave
- utilitzem la consola o línia d'ordres (>>)
- `pwd` (comprovar directori)
`dir` (fitxers del directori)
`cd C:\nou_directori` (canvi de directori, si cal)
- sortir del programa: `quit` (o `exit`)
- jerarquia d'operacions: `^` `* /` `+ -`
- `format short` (per defecte, mostra 4 decimals)
`format long` (mostra 15 decimals)
- `pi` (constant interna que conté el valor numèric de π)
- funcions incorporades:
`sqrt()`, `abs()`, `exp()`, `log()`, `log10()`, `sin()`, `cos()`, `tan()`, `asin()`, `acos()`, `atan()`, ...
- assignació d'una variable (o vector),
`a=expressió` (mostra el resultat)
`a=expressió`; (no mostra el resultat, útil amb vectors llargs o per a càlculs intermedis)
- `clear a` (neteja la variable *a*)
`clear all` (neteja totes les variables)

1. Vectors i gràfiques

`x=[0,1,2,3,4]` vector de coordenades x ,

`y=[2,-1,4,2,1]` vector de coordenades y ,

`plot(x,y)` gràfica que uneix els punts $(0, 2)$, $(1, -1)$, $(2, 4)$, $(3, 1)$, $(4, 1)$,

`plot(x,y,'ro--')` mateixa gràfica, en color vermell (colors `r`, `g`, `b`, `k`), representant els punts amb petits cercles (tipus de punts `+`, `*`, `o`, `x`, `^`, `.`), i units per línies discontinües (tipus de línies `-`, `--`, `:`).

2. Rangs de valors i operacions “terme a terme”

- `x=1:0.1:3` (equivalent a `x=linspace(1,3,21)`) creem el vector `x=[1, 1.1, 1.2, ..., 2.9, 3]` (21 termes)

- Podem sumar o multiplicar un valor constant, o aplicar qualsevol de les funcions incorporades, a tots els termes d'un vector, i d'aquesta manera crear altres vectors:

`y=x+1`, `z=sqrt(x)`, `w=sin(x-2)`

- Operacions “terme a terme”: `.*` `./` `.^`

Exemples: `x.*y`, `x./y`, `1./x`, `x.^2`

3. Funcions definides per una fórmula

- Per definir una funció que es pugui expressar mitjançant una única fórmula, per exemple la funció $f(x) = e^x - 3x^2$, podem escriure

`f = @(x) exp(x)-3*x.^2` (notem que `x` és una variable interna de la funció).

- Ara podem avaluar la funció en un punt, o en tot un vector o rang de valors:

`f(3)`, `f(x)`, `f(2*w+1)`

És important usar les operacions “terme a terme” en la definició de la funció, per poder aplicar-la sobre vectors.

4. Gràfica d'una funció

- Volem representar la funció $y = f(x)$, definida abans, per a $x \in [0, 4]$,

`x=0:0.01:4`; rang de valors en $[0, 4]$, amb un increment $h = 0.01$, i no l'escrivim

`y=f(x)`; apliquem la funció $f(x) = e^x - 3x^2$ a cada terme del vector `x`, obtenint un vector `y`, que no escrivim

`plot(x,y)` gràfica de $y = f(x)$, $x \in [0, 4]$

(també podríem haver escrit directament `plot(x,f(x))`)

- Més directe, especificant un interval però no un rang de valors: `fplot(f,[0,4])`

- Algunes opcions:

`axis([2,3.5,-8,-6])` visualitzem la finestra $[2, 3.5] \times [-8, -6]$,

`axis('tight')` retornem a la finestra original,

`axis('equal')` igualement les unitats dels eixos

`title('Grafica de f')`

`legend('f')`

`xlabel('coordenada x')`

`ylabel('coordenada y')`

`print -dpng 'dibuix.png'` exporta la gràfica a un fitxer `png` (hi ha d'altres possibilitats: `jpg`, ...)

5. Diverses gràfiques en un sol dibuix

$$f(x) = \begin{cases} 1-x & \text{si } x \in [-1, 0), \\ e^x & \text{si } x \in [0, 2), \\ 1+4x-x^2 & \text{si } x \in [2, 3], \end{cases} \quad \text{funció definida a trossos}$$

```
x1=-1:0.01:0; y1=1-x1;
x2=0:0.01:2; y2=exp(x2);
x3=2:0.01:3; y3=1+4*x3-x3.^2;
plot(x1,y1)
hold on    les gràfiques següents s'afegiran al mateix dibuix,
plot(x2,y2)
plot(x3,y3)
hold off   la gràfica següent es farà en un dibuix nou.
(de fet, directament podíem haver escrit plot(x1,y1,x2,y2,x3,y3) )
```

6. Funcions definides en un fitxer (M-file)

edit fmodel.m (editem un fitxer per a una funció que anomenem *fmodel*)

<pre>function y=fmodel(x) y=sin(x)-x.^2/9;</pre>	guardem amb Ctrl-S (i podem tancar la finestra);
--	--

```
x=0:0.001:10;
y=fmodel(x);  avaluem la funció fmodel a tots els punts del vector x i obtenim el vector y,
plot(x,y)
```

Una manera més directa d'obtenir la gràfica, quan hem definit la funció en un fitxer:
`fplot(@fmodel,[0,10])`

* Exercici

- Representeu, a l'interval $[-\pi, \pi]$, la funció $f(x) = \sin x$ i el seu polinomi de Taylor de grau 3 a l'origen, $p_3(x) = x - \frac{x^3}{6}$.

CONTROL DE FLUX

[capítol 9 del llibre]

Veurem com escriure petits programes per dur a terme *iteracions*, és a dir, repetir un cert nombre de vegades un mateix patró (constituït per una o diverses instruccions). També veurem com fer servir *condicionals*, per dur a terme unes instruccions o unes altres depenent de si certa condició es compleix o no.

Exemple 1: Una successió recurrent

Considerem la successió (a_n) definida per la *recurrència* següent:

$$\begin{aligned} a_1 &= 0.3, \\ a_n &= \cos a_{n-1}, \quad n \geq 2. \end{aligned}$$

- Podem obtenir alguns termes de la successió (a_n) escrivint:
`a=0.3, a=cos(a), a=cos(a), a=cos(a), ...`
- Ara bé, si volem obtenir termes a_n amb n gran, haurem d'usar un iterador. Per exemple, vegem com calcular tots els termes $a_1, \dots, a_n$ amb $n = 25$, i assignar-los a un vector:

```
n=25;
a(1)=0.3;
for i=2:n    (rang de valors i = 2, ..., n)
    a(i)=cos(a(i-1));
end    (important: l'iterador no s'executa fins que haguem escrit aquesta ordre)
a(25)    (valor de a25)
```

També, si escrivim simplement **a'**, visualitzarem tots els termes $a_1, \dots, a_{25}$ escrits en un vector columna.

- A la pràctica, és més recomanable editar un fitxer d'instruccions (*script*), que puguem guardar i modificar posteriorment.

Així doncs, editem un M-file,

edit progmodel.m

```
a(1)=0.3;
for i=2:n
    a(i)=cos(a(i-1));
end
```

Un cop escrit el fitxer progmodel.m, si volem calcular el terme a_{25} escriurem:

```
n=25;    (cal donar un valor n si no l'hem inclòs al fitxer)
progmodel    (executem l'iterador)
a(25)
```

De fet, aquesta successió és convergent, i $L = \lim_{n \rightarrow \infty} a_n = 0.739085133215161$ és solució de l'equació $x = \cos x$. Trobarem aproximacions d'aquest valor calculant termes a_n amb n prou gran.

Exemple 2: Successió de Fibonacci

La successió de Fibonacci (1, 1, 2, 3, 5, 8, 13, 21, ...) ve definida per una recurrència de *segon ordre*, en què cada terme es calcula a partir dels dos anteriors,

$$\begin{aligned} f_1 &= f_2 = 1, \\ f_n &= f_{n-1} + f_{n-2}, \quad n \geq 3. \end{aligned}$$

En altres paraules, la successió (f_n) és la solució de l'equació en diferències $f_n - f_{n-1} - f_{n-2} = 0$ amb les condicions inicials $f_1 = 1, f_2 = 1$.

- Escriurem una *funció* que ens calcularà, a partir d'una n donada (que passarem com a argument), un vector de components $f_1, \dots, f_n$ (que retornarà com a resultat):

edit fibonacci.m

```
function fib=fibonacci(n)
fib(1)=1;
fib(2)=1;
for i=3:n
    fib(i)=fib(i-1)+fib(i-2);
end
```

Observació: També podríem escriure un fitxer d'instruccions com a l'exemple anterior. Remarquem que en el cas d'una funció les variables internes (**fib,i**) no es reconeixen fora de la funció i només es retorna el resultat.

Ara podem assignar un vector amb el resultat de la funció per a una n donada,

```
f=fibonacci(12)
```

- El quocient de termes consecutius de la successió de Fibonacci convergeix cap al nombre d'or,

$$\lim_{n \rightarrow \infty} \frac{f_n}{f_{n-1}} = \frac{1 + \sqrt{5}}{2} = 1.61803398874989.$$

Per calcular alguns quocients, podem fer:

```
g=f(2:12);    vector dels  $f_2, \dots, f_{12}$ 
h=f(1:11);    vector dels  $f_1, \dots, f_{11}$ 
(g./h)'       quocient "terme a terme", expressat com a vector columna
```

Exemple 3: Sumes de sèries

La següent sèrie numèrica és convergent, amb suma coneguda: $\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots = \frac{\pi}{4}$.

Vegem com usar les sumes parcials $\sum_{k=0}^n \frac{(-1)^k}{2k+1}$ per obtenir una aproximació del nombre π .

edit calculpi.m

```
suma=0;
for k=0:n
    suma=suma+(-1)^k/(2*k+1);
end
```

```
n=50000;    (considerem els senars 1,3,5,...,100001)
calculpi
piaprox=4*suma
```

Exemple 4: Ús de condicionals

Podem usar condicionals (**if**, **else**, **elseif**) dins d'un fitxer d'instruccions, o en la definició d'una funció. Per exemple,

per definir una funció a trossos:
$$f(x) = \begin{cases} 1-x & \text{si } x \in (-\infty, 0), \\ 3 & \text{si } x = 0, \\ e^x & \text{si } x \in (0, 2), \\ 1+4x-x^2 & \text{si } x \in [2, \infty), \end{cases}$$

podem fer:

edit f.m

```
function y=f(x)
if x<0
    y=1-x;
elseif x==0
    y=3;
elseif x<2
    y=exp(x);
else
    y=1+4*x-x.^2;
end
```

Ara podem comprovar els valors de $f(-2)$, $f(0)$, $f(1)$, $f(3)$.

ZEROS I OPTIMITZACIÓ DE FUNCIONS

[capítol 4 del llibre]

1. Localització gràfica dels zeros

Volem trobar els zeros de la funció $f(x) = \ln(x^2 - x + 2) \cos x$, a l'interval $[-10, 15]$.

- Definim la funció per fórmula: `f = @(x) ...`
- La representem gràficament, a l'interval donat:
`x=-10:0.001:15;`
`plot(x,f(x))`
o bé, més directament, fent: `fplot(f,[-10,15])`
- `grid on` dibuixem una graella que ens permet situar millor els zeros de la funció (podem suprimir la graella fent `grid off`).
Així localitzem intervals on la funció f (contínua) canvia de signe i per tant, d'acord amb el teorema de Bolzano, hi té algun zero: $[-10, -5]$, $[-5, 2.5]$, $[-2.5, 0]$, $[0, 2.5]$, $[2.5, 5]$, $[5, 10]$, $[10, 12.5]$, $[12.5, 15]$
(podem veure-ho millor ampliant alguna part de la gràfica amb `axis(...)`).
- Per assegurar-nos que tenim zeros en aquests intervals, comprovem numèricament que hi ha un canvi de signe entre els seus extrems. Per exemple a l'interval $(0, 2.5)$,
`f(0)*f(2.5)` dóna un valor negatiu, i per tant $f(0)$ i $f(2.5)$ són de diferent signe.

2. Càlcul dels zeros

- En el cas de polinomis, tenim la funció `roots()` que ens dóna directament una aproximació dels zeros (reals o complexos). Per exemple, per al polinomi $p(x) = 3x^3 - 5x + 1$, fem:

`p=[3,0,-5,1]` (vector amb els coeficients)
`r=roots(p)`

- Tornant a la funció $f(x) = \ln(x^2 - x + 2) \cos x$ que hem definit abans, donat un interval amb un canvi de signe entre els seus extrems, usant la funció `fzero()` podem donar una aproximació a algun zero de f en aquest interval. Per exemple, amb l'interval $[0, 2.5]$,
`c=fzero(f,[0,2.5])` assignem l'aproximació del zero a una variable `c`.
Notem que `f(c)` ens dona un valor molt petit, però $\neq 0$.
- Podem assegurar-nos que l'aproximació `c` té un nombre determinat de xifres decimals correctes. Per exemple, si volem veure que l'error és $< 10^{-6}$, fem:
`delta=1e-6`
`f(c-delta/2)*f(c+delta/2)` ha de donar un valor negatiu
- Si haguéssim definit la funció en un fitxer (`edit f.m`), aleshores escriuríem:
`fzero(@f,[0,2.5])`

3. Mínims i màxims

- Per trobar un mínim aproximat de la funció f (definida per fórmula: `f = @(x)...`) a l'interval $[-10, 15]$,
`xmin=fminbnd(f,-10,15)` (assignem l'abscissa del mínim a la variable `xmin`)
El mínim que trobem no és necessàriament el mínim absolut (el qual pot trobar-se en un dels extrems de l'interval).
- Com abans, si haguéssim definit la funció en un fitxer `f.m`, escriuríem:
`xmin=fminbnd(@f,-10,15)`
- Per conèixer el valor de f al mínim trobat, podem fer
`ymin=f(xmin)`
o bé, directament, podíem haver obtingut els dos valors `xmin` i `ymin` escrivint:
`[xmin,ymin]=fminbnd(f,-10,15)`
- Per anar trobant els diferents mínims relatius, amb l'ajut de la gràfica podem localitzar intervals on n'hi hagi només un. Per exemple, gràficament observem que hi ha un mínim relatiu a l'interval $[3, 4]$, i podem fer
`[xmin2,ymin2]=fminbnd(f,3,4)`
- De fet, per defecte no es calculen els mínims amb una gran precisió, però ho podem millorar canviant la tolerància d'error. Per exemple, si volem calcular el mínim amb una tolerància 10^{-8} , escriuríem:
`xmin=fminbnd(f,-10,15,optimset('TolX',1e-8))`
- Per trobar màxims de f , definirem una funció $g(x) = -f(x)$ i en trobarem els mínims. Per definir g (si ja hem definit f), podem escriure
`g = @(x) -f(x)`
Llavors, els mínims de g ens donen els màxims de f :
`[xming,yming]=fminbnd(g,2,9)`
`xmaxf=xming, ymaxf=-yming`

* Exercici

- Trobeu els zeros i els extrems relatius de la funció $h(x) = x - 2 \cos(x^2)$, a l'interval $[-3, 3]$.

MÈTODES D'INTEGRACIÓ NUMÈRICA

[capítol 8 del llibre]

Il·lustrarem els mètodes amb la integral $I = \int_1^3 f(x)dx$ de la funció $f(x) = \frac{\sin x}{x^2 + 1}$.

Començarem definint la funció f per fórmula (`f = @(x)...`) o bé en un fitxer (`edit f.m`). També assignarem els extrems d'integració a variables: `a=1, b=3`.

1. Sumes de Riemann

- Donades abscisses $a = x_0 < x_1 < \dots < x_N = b$, amb $x_i = a + ih$, $i = 0, 1, \dots, N$, si per a cada subinterval escollim un punt $c_i \in [x_{i-1}, x_i]$, $i = 1, \dots, N$, i aproximem $f(x)$ pel seu valor en aquest punt, tenim una *suma de Riemann*:

$$I \simeq h(f(c_1) + \dots + f(c_N)).$$

Per exemple, amb $N = 50$ subintervals i escollint els c_i com el punt mig de cada subinterval, faríem

```
N=50; h=(b-a)/N;
i=1:N;    vector d'índexs
c=a+(i-0.5)*h;    vector dels  $c_1, \dots, c_N$ 
yc=f(c);    vector dels  $f(c_1), \dots, f(c_N)$ 
riem=h*sum(yc)
```

2. Regla dels trapezis

- Donat N , considerem abscisses equidistants $a = x_0 < x_1 < \dots < x_N = b$, és a dir $x_i = a + ih$, $i = 0, 1, \dots, N$, amb pas d'integració $h = (b - a)/N$. La regla dels trapezis ve donada per la fórmula

$$I \simeq T(h) = \frac{h}{2} (f(x_0) + 2f(x_1) + 2f(x_2) + \dots + 2f(x_{N-1}) + f(x_N)).$$

- Per al càlcul de la regla dels trapezis, podem usar la funció `trapz()`, que té com a arguments el vector d'abscisses i el vector d'ordenades. Per exemple, amb 50 subintervals, faríem

```
N=50; h=(b-a)/N;
x=a:h:b;    vector d'abscisses
y=f(x);    vector d'ordenades
tr=trapz(x,y)
```

3. Regla de Simpson

- Ara dividim l'interval $[a, b]$ en un nombre *parell* de subintervals $N = 2M$, considerant abscisses $x_i = a + ih$, $i = 0, 1, \dots, N$, amb pas d'integració $h = (b - a)/N$. La regla de Simpson ve donada per la fórmula

$$I \simeq S(h) = \frac{h}{3} (f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \dots + 2f(x_{N-2}) + 4f(x_{N-1}) + f(x_N)).$$

Una relació entre les regles dels trapezis i de Simpson ve donada per la *fórmula d'extrapolació*:

$$S(h) = \frac{4T(h) - T(2h)}{3}.$$

- Tot i que no hi ha una funció específica per a calcular una integral mitjançant la regla de Simpson, podem combinar la funció `trapz()` amb la fórmula d'extrapolació. Per exemple, per calcular la regla de Simpson amb 20 subintervals (un nombre parell), podem fer

```
N=20; h=(b-a)/N;
x1=a:h:b; y1=f(x1); tr1=trapz(x1,y1)
x2=a:2*h:b; y2=f(x2); tr2=trapz(x2,y2)
sims=(4*tr1-tr2)/3
```

4. Quadratura adaptativa de Gauss–Legendre–Lobatto

- Els mètodes de quadratura adaptativa permeten decidir els passos adequats d'integració, a partir de les característiques de la funció i d'una tolerància d'error donada. Per exemple, per calcular la integral $\int_a^b f(x)dx$ amb una tolerància 10^{-6} , faríem

`lob=quadl(f,a,b,1e-6)` si hem definit la funció per fórmula (`f = @(x)...`),

o bé

`lob=quadl(@f,a,b,1e-6)` si tenim la funció f definida en un fitxer (`edit f.m`).

Pràctiques (CÀLCUL I)

* *Expresseu els resultats en “format long”, amb tots els decimals.
Feu “clear all” abans de començar un nou exercici.*

1 Càlcul d’una suma finita

Indiqueu el valor de la suma $S = \sum_{n=3}^{500} \frac{\sin(n^2 + 1)}{n^3 + \ln n}$.

Sol.: $S = -0.0283393000786980$

2 Zeros i extrems

Trobeu els zeros i els extrems relatius de la funció $f(x) = x^2 + 8 \sin x - 10$ que es trobin continguts a l’interval $[1, 5]$.

*Sol.: zeros 1.43865895243236, 3.068452733699908, 4.01910733939131;
mínim relatiu 3.59530486887476; màxim relatiu 2.13333224860555*

3 Recurrència

Donada la successió (a_n) definida per la recurrència (o equació en diferències)

$$a_1 = 5, \quad a_n = 3 - \frac{a_{n-1}}{2} + \sin(a_{n-1}^2), \quad n \geq 2,$$

calculeu-ne el terme a_{45} .

Sol.: $a_{45} = 0.913841832170866$

4 Recurrència de segon ordre

Donada la successió (b_n) definida per la recurrència

$$b_1 = 0, \quad b_2 = 3, \quad b_n - 2b_{n-1} + 3b_{n-2} = (-1)^n \ln(n^2 + 3), \quad n \geq 3,$$

calculeu-ne el terme b_{21} .

Sol.: $b_{21} = 4.38168530805157 \cdot 10^4$

5 Integral numèrica

Donada la integral $\int_{-1}^5 \frac{dx}{3 + \cos(x^2 + e^x)}$, siguin T i S els resultats d’aplicar les regles dels trapezis i de Simpson, respectivament, amb 60 subinterval·ls, i sigui Q el resultat d’aplicar la quadratura adaptativa de Gauss–Legendre–Lobatto amb una tolerància d’error 10^{-9} . Indiqueu el valor de S i $|T - Q|$.

Sol.: $S = 2.07094379287543$; $|T - Q| = 0.00582854705612723$

6 Càlcul d’un doble factorial

Editeu una funció `dfactorial.m` que calculi el doble factorial d’un nombre natural:

$$\begin{aligned} n!! &= 1 \cdot 3 \cdot 5 \cdots n && \text{si } n \text{ és senar,} \\ n!! &= 2 \cdot 4 \cdot 6 \cdots n && \text{si } n \text{ és parell.} \end{aligned}$$

(Per diferenciar si n és senar o parell, podeu usar que `mod(n,2)` ens dona la resta de dividir n entre 2.)
Calculeu $24!!$, $25!!$ i $26!!$, i verifiqueu per a $n = 25$ i $n = 26$ les fórmules de Wallis:

$$\int_0^{\pi/2} \cos^n x \, dx = \begin{cases} \frac{(n-1)!!}{n!!} & \text{si } n \text{ és senar,} \\ \frac{(n-1)!!}{n!!} \cdot \frac{\pi}{2} & \text{si } n \text{ és parell.} \end{cases}$$

Sol.: $24!! = 1961990553600$, $25!! = 7905853580625$, $26!! = 51011754393600$

7 Dibuix d’un pentàgon

Representeu el pentàgon de vèrtexs $A_k = \left(\cos \frac{2k\pi}{5}, \sin \frac{2k\pi}{5} \right)$, $k = 0, \dots, 4$, i afegiu-hi el segment que uneix A_2 amb A_4 .

8 Polinomis de Legendre

Els polinomis de Legendre venen definits per la recurrència següent:

$$\begin{aligned}P_0(x) &= 1, \\P_1(x) &= x, \\P_k(x) &= \frac{2k-1}{k} x P_{k-1}(x) - \frac{k-1}{k} P_{k-2}(x), \quad k \geq 2.\end{aligned}$$

- (a) Recordeu que representem un polinomi com un vector que conté els seus coeficients, en ordre decreixent de grau: $\mathbf{p}=[1,0,0,0,0,3,0,-1]$ correspon al polinomi $p(x) = x^7 + 3x^2 - 1$.

Per avaluar un polinomi, en un punt x o en els termes d'un vector o rang de valors:

`polyval(p,0.5)` ens dóna el valor de $p(0.5)$,

`polyval(p,0:0.1:2)` avalua el polinomi p en $x = 0, 0.1, 0.2, \dots, 2$.

Arrels d'un polinomi: `r=roots(p)` crea un vector columna $\mathbf{r}$, amb una aproximació numèrica de les arrels (reals i complexes) de $p(x)$.

Per sumar polinomis, podem usar la funció `sumapolinomi.m` següent:

```
function suma=sumapolinomi(p,q)
n=length(p); m=length(q);
if n<m
    p=[zeros(1,m-n),p];
elseif n>m
    q=[zeros(1,n-m),q];
end
suma=p+q;
```

El producte de polinomis s'efectua a través d'una operació amb els seus coeficients anomenada convolució:

`prod=conv(p,q)`

- (b) Editeu una funció `leg.m` que calculi el vector de coeficients del polinomi de Legendre $P_n(x)$ per a un n donat. A cada iteració, calculem el polinomi a partir dels dos anteriors, que guardem en variables `penultim` i `ultim`, i reassignem aquestes variables per a la iteració següent.

```
function p=leg(n)
penultim=1;
ultim=[1,0];
for k=2:n
    p=(càlcul de  $P_k(x)$  a partir de ultim i penultim);
    (reassignació de penultim i ultim)
end
```

- (c) Calculeu els polinomis de Legendre $P_n(x)$, per a $n = 2, 3, 4, 5, 6$, i les seves arrels. Observeu que cada $P_n(x)$ és de grau n , i que totes les arrels són reals i contingudes a l'interval $(-1, 1)$.

Feu simultàniament la gràfica dels polinomis anteriors, a l'interval $[-1, 1]$. Observeu les propietats següents:

- es compleix que $|P_n(x)| \leq 1$ sempre que $|x| \leq 1$ (gràfiques contingudes a la finestra $[-1, 1] \times [-1, 1]$);
- el polinomi $P_n(x)$ és funció parella si n és parell, i és funció senar si n és senar;
- valors als extrems: $P_n(-1) = (-1)^n$, $P_n(1) = 1$;
- arrels entrellaçades: entre cada dues arrels de $P_n(x)$ hi ha una única arrel de $P_{n+1}(x)$.