

Notes generals

- En aquestes pràctiques, podrem usar indistintament Octave o Matlab.
 - Octave és programari lliure (es pot instal·lar i usar lliurement)
 - Matlab és un paquet comercial (requereix llicència)
- Informació sobre el Taller:
 - <http://atenea.upc.edu/moodle/course/view.php?id=26227> (Atenea)
 - <http://www.ma1.upc.edu/~taller/> (exemples d'examen, instal·lació d'Octave)
 - Llibre de Joaquim Puig, *"Taller de Matemàtiques: pràctiques en Matlab/Octave"*
 - aules amb PC's lliures:
<http://www.etseib.upc.edu/disponibilitat/disponibilitat.html>
- per entrar al programa,
 - càrrega de Windows 7
 - Ctrl+Alt+Supr
 - *username@est* / *password*
 - Inici → Tots els programes → Progr. Docència → Mat. → Taller Mat.
→ [Octave (CLI) / Octave (GUI) / Matlab]
- 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`)

ÚS BÀSIC D'Octave/Matlab

1. Càlcul d'expressions numèriques

- Exemples: $\frac{3}{\left(1 + \frac{5}{13}\right)^2}, \frac{3}{1 + \frac{5}{13^2}}.$
- Podem recuperar instruccions anteriors amb $\boxed{\uparrow}$ (si primer escrivim una mica, recuperem instruccions anteriors que comencen amb allò que hem escrit).

- És important tenir en compte la jerarquia d'operacions: $\boxed{\wedge}$ $\boxed{* /}$ $\boxed{+ -}$
 Per trencar aquesta jerarquia, usarem parèntesis.
 Si concatensem dues operacions amb la mateixa jerarquia, s'executen en l'ordre en què apareixen escrites:

Exemple: $3/7*8$ ens dóna $\frac{3}{7} \cdot 8$, en canvi $3/(7*8)$ (o bé $3/7/8$) ens dóna $\frac{3}{7 \cdot 8}$.

Exemple: 5^3^2 ens dóna $(5^3)^2$, en canvi $5^(3^2)$ ens dóna 5^{3^2} .

- els decimals s'expressen amb un punt (.)
- Notació exponencial per a nombres molt grans: $1.5\text{e}6$ ens dóna $1.5 \cdot 10^6 = 1\,500\,000$; o molt petits: $3\text{e}-6$ ens dóna $3 \cdot 10^{-6} = 0.000\,003$.
- **format short** (per defecte, mostra 4 decimals)
format long (mostra 15 decimals)
- El valor numèric de π ens ve donat per la constant interna **pi**.
 Per obtenir el nombre e , podem escriure **exp(1)**.
- Funcions incorporades: **sqrt()**, **abs()**, **exp()**, **log()**, **log10()**, **sin()**, **cos()**, **tan()**, **asin()**, **acos()**, **atan()**, **sinh()**, **cosh()**, ...
 Exemple: calculem $\cos(\sqrt{7\pi} + \ln(3 \arcsin(e^{-2})))$
- Obtenir informació d'alguna funció: **help asin**; o de qualsevol altra instrucció: **help format**.

2. Assignació de variables

- Podem assignar variables, per guardar valors que després usarem en altres càlculs:
 $\mathbf{a=expressió}$ (mostra el resultat)
 $\mathbf{a=expressió;}$ (no mostra el resultat, útil amb resultats vectorials o per a càlculs intermedis)
- Exemple: Donats els valors $a = -7$ i $b = 3$, calculem $z = a + e^{c-a} \cos(b + \ln c)$ essent $c = \frac{a^2}{b^4 + 1}$.
 Després, ho calculem amb els valors $a = 5$ i $b = -2$.
- **clear a** (neteja la variable **a**)
clear all (neteja totes les variables)

3. Precisió de càlcul, missatges d'error

- Internament, els càlculs es realitzen amb precisió alta, però limitada, és a dir, considerant un cert nombre de xifres significatives. Ho podem conèixer amb la constant interna **eps**, que ens dóna la diferència més petita possible (en termes relatius) entre dos nombres. Per exemple, el resultat de $1+1\text{e}-16$ (és a dir, $1 + 10^{-16}$) és 1. Observem també el resultat de la diferència: $10\,000\,000\,000\,000\,001 - 10\,000\,000\,000\,000\,000$.
- Les constants internes **realmax** i **realmin** ens donen el nombre més gran i el més petit que podem considerar (en termes absoluts).
- Si intentem càlculs com $1/0$, $0/0$, $\log(0)$ obtenim missatges d'error: **Inf**, **NaN**.

4. Nombres complexos

- Certs càlculs, com `sqrt(-2)`, ens donen com a resultat nombres complexos, que s'expressen en la forma `a+b*i`.
- Per operar amb nombres complexos, les funcions `abs()`, `angle()`, `real()`, `imag()`, `conj()` ens donen, respectivament, el mòdul, l'argument, les parts real i imaginària, i el conjugat d'un nombre complex.
Com a exemple, apliquem aquestes funcions al nombre $3 + 2i$.

VECTORS I POLINOMIS

1. Vectors

- `v=[4,1,6,7,1]` un vector fila (també: `v=[4 1 6 7 1]`)
`w=[4;3;3;-2;-1]` un vector columna
De fet, podem considerar els vectors com a matrius, d'una sola fila o columna respectivament.
- Podem accedir a un terme concret, posant el seu índex entre parèntesis:
per exemple `v(3)`, o també `w(4)`.
Obtindrem un missatge d'error si demanem un terme inexistent: `v(0)`, `w(7)`.
- Operacions com a espai vectorial: suma i producte per escalar.
Per exemple, definint també `u=[3,2,-5,1,-3]`, podem calcular `v+u` i també `5*v`.
Obtindrem missatges d'error si intentem sumar vectors de diferents dimensions, o un vector fila amb un vector columna; per exemple, si definim `z=[-1,3,0]` i intentem fer `v+z` o bé `v+w`.
- Podem convertir un vector fila en un vector columna, o al revés, fent `v'`
- Producte escalar: `dot(v,u)`
Producte vectorial: `cross(v,u)`, si els vectors tenen dimensió 3
- Operacions “terme a terme”: `.*` `./` `.^`
Exemples: `v.*u`, `v./u`, `3./v`, `v.^3`
No cal en els casos: `3*v`, `v/3`, `v+3` (en aquest darrer, sumem una constant a tots els termes del vector `v`).
Podem aplicar qualsevol funció a tots els termes d'un vector, per exemple `sqrt(v)`, `cos(v)`.
- Altres funcions sobre vectors:
 - `length()` dimensió d'un vector
 - `sort()` ordena un vector en ordre creixent
 - `sum()` suma de tots els termes
 - `prod()` producte de tots els termes
 - `cumsum()` sumes acumulades
 - `cumprod()` productes acumulats
- Rangs de valors: ens seran molt útils per a la representació gràfica de funcions.
La forma general és `v=inici:increment:final` i ens dóna el vector format pels termes *inici*, *inici+increment*, *inici+2*increment*, ... mentre siguin $\leq$ *final*.
Per exemple, `v=1:0.1:3`, que equival al vector de 21 termes `v=[1, 1.1, 1.2, ..., 2.9, 3]`
(de manera equivalent, podríem escriure `v=linspace(1,3,21)`)

L'increment pot ser negatiu: `v=2:-0.1:0`

L'increment és 1 si no l'especifiquem: `v=3:50` (equival a `v=3:1:50`).

- Podem fer operacions, o aplicar qualsevol de les funcions incorporades, a tots els termes d'un vector, i així crear altres vectors (tenint en compte les operacions "terme a terme"):
`x=4:0.1:5`
`y=sin(x.^2.*exp(x)-1)` (avaluem $f(x) = \sin(x^2 e^x - 1)$ en diferents punts de l'interval $[4, 5]$)

2. Polinomis com a vectors, i les seves arrels

- Representem un polinomi com un vector que conté els seus coeficients, en ordre decreixent de grau:

`p=[1,0,0,0,0,3,0,-1]` correspon al polinomi $p(x) = x^7 + 3x^2 - 1$.

- Avaluació d'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 **r**, amb una aproximació numèrica de les arrels reals i complexes de $p(x)$.

En alguns casos l'aproximació numèrica que s'obté és poc precisa, com veiem per exemple amb el polinomi $q(x) = x^7 + 7x^6 + 21x^5 + 35x^4 + 35x^3 + 21x^2 + 7x + 1 = (x + 1)^7$, que hauria de tenir $x = -1$ com a arrel de multiplicitat 7.

3. Operacions amb polinomis

- Per sumar dos polinomis, si són del mateix grau sumarem directament els vectors que els representen. Però si són de diferents graus, haurem d'afegir zeros a les primeres posicions d'un dels vectors perquè siguin de la mateixa dimensió:

`p=[1,0,0,0,0,3,0,-1]` (grau 7)

`q=[1,2,1,0]` (grau 3)

`suma=p+[zeros(1,4),q]`

- El producte de polinomis s'efectua a través d'una operació amb els seus coeficients anomenada convolució: `prod=conv(p,q)`

- Divisió entera de dos polinomis: $p(x) = u(x)q(x) + r(x)$, amb $\text{gr } r(x) < \text{gr } q(x)$.

`u=deconv(p,q)` (obtenim només el quocient $u(x)$)

`[u,r]=deconv(p,q)` (obtenim el quocient $u(x)$ i la resta $r(x)$)

4. Derivació i integració de polinomis

- `polyder(p)` coeficients de la derivada $p'(x)$;
`polyint(p)` coeficients de la primitiva $F(x) = \int p(x)dx$ que compleix $F(0) = 0$.

- Descomposició en fraccions simples: donats $p(x)$ i $q(x)$ amb $\text{gr } p(x) < \text{gr } q(x)$, escrivint `[c,z]=residue(p,q)`

obtenim un vector **c** amb els numeradors de la descomposició, i un altre vector **z** amb les arrels del denominador (repetides si són múltiples). En el cas d'arrels complexes conjugades, obtenim una fracció (complexa) per a cada arrel.

* Exercicis

- 2.1 i 2.4 (pàg. 38–39 del llibre).
- Calculeu la integral definida $\int_{-2}^3 p(x)^2 dx$ per al polinomi $p(x) = 4x^4 - 5x^3 - 2x^2 - 1$.

REPRESENTACIÓ GRÀFICA

1. Vectors i gràfiques

- Una gràfica simple:
`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. 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).
Llavors es guarda `f` com una variable de tipus *function handle*.
- Ara podem avaluar la funció en un punt: `f(3)`,
o en tot un vector o en un rang de valors:
`x=2:0.1:3`, `f(x)`, `f(2*x.^2-1)`
És molt important usar les operacions “terme a terme” `.*` `./` `.^` en la definició de la funció, per poder aplicar-la sobre vectors.

3. 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 podíem 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')         igualem 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 (altres opcions: jpg, ...)
```

4. 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 podríem haver escrit plot(x1,y1,x2,y2,x3,y3) )
```

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

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

```
function y=fmodel(x)
y=sin(x)-x.^2/9;
```

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 a l'origen de grau 3 ($p_3(x) = x - \frac{x^3}{6}$).

ZEROS I OPTIMITZACIÓ DE FUNCIONS

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 una 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 dóna 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, podríem 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 $[-3, 3]$.

CREACIÓ I MANIPULACIÓ DE MATRIUS

1. Matrius

- Una matriu 4×5 :
`A=[-1,-2,-6,-9,4; 9,1,4,4,-5; 7,-7,-5,4,1; -3,9,-7,3,-6]`
- Podem accedir a un terme concret, una fila o columna, o una “submatriu”:
`c=A(2,3)` (un terme)
`v=A(3,:)` (la 3^a fila)
`w=A(:,4)` (la 4^a columna)
`B=A(2:3,1:4)` (submatriu)
`D=A(:,3:4)` (submatriu formada per algunes columnes)

També podem modificar una matriu, redefinint-ne algun terme, fila, columna o submatriu (les dimensions han de ser coherents):

```
A(3,1)=3
A(2,:)=[3,7,0,7,-2]
A(3:4,4:5)=[1,1; 1,1]
```

- Podem ampliar una matriu afegint-hi files o columnes:

`u=[-4;1;7;5]` (columna que volem afegir)

`Aamp=[A,u]` (matriu ampliada)

2. Operacions amb matrius

- Dimensions d'una matriu: `size(A)`
- Suma i producte per escalar: igual que en el cas de vectors.
- Transposada: `A'` (o també `transpose(A)`), s'intercanvien files i columnes
- Producte de matriu per vector:

`v=[-3;5;-8;6;5]` vector columna

`A*v` (el nombre de columnes de `A` ha de coincidir amb la dimensió de `v`)

- Producte matricial i potències:

`A*B` (el nombre de columnes de `A` ha de coincidir amb el nombre de files de `B`)

`A^3` (la matriu `A` ha de ser quadrada)

- Determinant i inversa d'una matriu quadrada: `det(A)`, `inv(A)`.

- Divisió matricial:

`A/B` divisió per la dreta ($A B^{-1}$),

`B\A` divisió per l'esquerra ($B^{-1} A$).

- Operacions “terme a terme”: igual que en el cas de vectors.

3. Matrius especials

`eye(n)` matriu identitat

`diag(v)` matriu diagonal, amb la diagonal principal formada pel vector `v`

`diag(v,±k)` matriu amb una diagonal secundària formada pel vector `v`

`blkdiag(A,B,...)` matriu diagonal per blocs, a partir de les matrius `A`, `B`, ...

`zeros(n,m)`

`ones(n,m)`

`rand(n,m)` matriu de termes aleatoris, escollits a l'interval $(0,1)$

* **Exercicis:** 5.1, 5.3 i 5.4 (pàg. 83–84 del llibre).

SISTEMES D'EQUACIONS LINEALS

1. Resolució de sistemes quadrats invertibles

- Volem resoldre el sistema $Ax = b$, on A és una matriu $n \times n$ invertible, i b és un vector columna de n components. Per exemple, considerem

`A=vander(1:4)` (matriu de Vandermonde associada als nombres 1,2,3,4),
`b=rand(4,1)` (un vector columna qualsevol).

Per resoldre el sistema, escriurem:

`x=A\b`

També podríem calcular la solució escrivint `x=inv(A)*b`, però això comportarà molt més temps de càlcul si la dimensió de la matriu A és gran, ja que per a resoldre el sistema no és necessari calcular tota la matriu inversa A^{-1} .

- Si la matriu A és no invertible ($\det A = 0$), en intentar resoldre el sistema $Ax = b$ apareixerà un missatge d'error, però en Octave obtindrem igualment un resultat. Si el sistema es compatible ($b \in \text{Im } A$) obtindrem una de les infinites solucions, i si és incompatible ($b \notin \text{Im } A$) la solució obtinguda no serà vàlida (de fet, s'intenta minimitzar $\|Ax - b\|$).
Exemple: `A=[1,2; 2,4]`, provem de resoldre el sistema amb `b=[3;6]` i amb `b=[3;7]`.

2. Imatge i nucli d'una matriu

- Considerem per exemple la matriu 3×5 ,
`A=[1,0,1,0,3; 1,0,3,6,7; 2,0,4,6,10]`
Escrivint `rank(A)` podem deduir la dimensió de la imatge i el nucli de A :
 $\dim \text{Im } A = \text{rang } A = 2$, $\dim \text{Nuc } A = 5 - \text{rang } A = 3$.
- Per obtenir una base del nucli de A , escrivim
`nucliA=null(A)`
que ens dóna una matriu 5×3 , les columnes de la qual formen una base del nucli de A . Podem comprovar que el producte `A*nucliA` és 0, o molt petit.
- Per obtenir una base de la imatge de A , podem usar una funció que ens dóna la forma esglaonada reduïda de la matriu:
`[R,col]=rref(A)`
i aprofitar que `col` és un vector que conté els índexs de columnes independents. Així, escrivint `imatgeA=A(:,col)`
obtenim com a columnes una base de la imatge de A .

3. Discussió de sistemes lineals

- Donat un sistema $Ax = b$, on la matriu A és $m \times n$, considerem la matriu ampliada $\overline{A}$, que té el vector b com a columna addicional. Aleshores:
 - si $\text{rang } A = \text{rang } \overline{A} = n$, el sistema és compatible determinat (una única solució);
 - si $\text{rang } A = \text{rang } \overline{A} < n$, el sistema és compatible indeterminat (infinites solucions);
 - si $\text{rang } A < \text{rang } \overline{A}$, el sistema és incompatible (no existeix solució).

- Ho podem discutir definint la matriu ampliada $A_{amp}=[A,b]$ i calculant $\text{rank}(A)$ i $\text{rank}(A_{amp})$.
Per trobar la solució, quan $\text{rang } A = \text{rang } \bar{A} = n = m$ podem aplicar directament $x=A \backslash b$.
En altres casos, podem usar la forma esglaonada reduïda de la matriu ampliada:
 $R_{amp}=rref(A_{amp})$
la qual és la matriu ampliada d'un sistema $Rx = c$, que té les mateixes solucions que l'original, les quals ara es poden trobar de manera immediata.

* **Exercicis:** 6.1, 6.2, 6.3 (pàg. 101 del llibre).

INTEGRACIÓ NUMÈRICA

Il·lustrarem els mètodes d'integració numèrica amb la integral $I = \int_1^3 f(x)dx$,

per a la funció $f(x) = \frac{\sin x}{x^2 + 1}$.

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

1. 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 subintervalls, faríem

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

2. Regla de Simpson

- Ara dividim l'interval $[a, b]$ en un nombre *parell* de subintervalls $N = 2M$, 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 subintervalls (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)
simps=(4*tr1-tr2)/3
```

3. 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`).

CONTROL DE FLUX

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 es compleix certa condició.

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  $i = 2, \dots, n$ )
    a(i)=cos(a(i-1));
end (important: l'iterador no s'executa fins que haguem escrit aquesta ordre)
a(25) (valor de  $a_{25}$ )
```

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. Remarqueu 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 $f(-2)$, $f(0)$, $f(1)$, $f(3)$.

VALORS I VECTORS PROPIS

- Donada una matriu A , de dimensions $n \times n$ (quadrada), un vector v és vector propi de valor propi λ si $Av = \lambda v$. Els valors propis són les arrels del polinomi característic $p(x) = \det(xI - A)$ (de grau n).
- Per calcular els valors propis (*eigenvalues*) de la matriu A , podem fer
`vaps=eig(A)`
i obtenim un vector columna format pels valors propis (reals o complexos, i repetits si són múltiples).
- L'espai propi (els vectors propis) d'un valor propi λ concret, és el nucli de la matriu $\lambda I - A$,
`lambda=vaps(j)` (triem un valor propi)
`esppropi=null(lambda*eye(n)-A)`
- De fet, amb la mateixa funció `eig()` podem trobar vectors propis per a tots els valors propis,
`[veps,diagvaps]=eig(A)`
que ens dona els valors propis en una matriu diagonal, i una altra matriu amb els vectors propis com a columnes. Es compleix que les matrius $A*veps$ i $veps*diagvaps$ són (quasi) iguals.

- La matriu A és diagonalitzable si té una base de vectors propis. En cas contrari, alguns dels vectors propis trobats a la matriu **veps** apareixeran repetits. Per esbrinar-ho, podem calcular **rank(veps)** i si és màxim ($= n$) llavors A és diagonalitzable, i en cas contrari ($< n$) no ho és.
- Altres funcions:

poly(A) (dóna el polinomi característic de A , que té com a arrels els valors propis)

compan(p) (per a un polinomi $p(x)$, donat pel vector dels seus coeficients, dona una matriu de forma especial que té com a valors propis les arrels del polinomi)

* **Exercicis:** 10.1 i 10.2 (pàg. 140 del llibre).

PRÀCTIQUES

* *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 Polinomis

Donat el polinomi $p(x) = x^5 - 2x^2 + 3x + 1$, considereu $q(x) = ((x^3 - 2)p(x) + p'(x))^2$ i calculeu $q(\alpha)$, on α és l'única arrel real de $p(x)$.

Sol: $q(\alpha) = 17.2417156112809$

3 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

4 Càlculs amb una matriu

Donada la matriu 9×9 ,

$$A = \begin{pmatrix} \ln 10 & 0 & 0 & \cdots & 0 & e^{-4} \\ 1 & \ln 9 & 0 & \cdots & 0 & 0 \\ 0 & 2 & \ln 8 & \cdots & 0 & 0 \\ & & \ddots & \ddots & & \\ 0 & 0 & \cdots & 7 & \ln 3 & 0 \\ 0 & 0 & \cdots & \cdots & 8 & \ln 2 \end{pmatrix},$$

calculeu $\det A$ i el terme $b_{2,3}$ de la matriu $B = (I + A^2)^{-1}$.

Sol: $\det A = 800.808209853463$, $b_{2,3} = -0.134618419548104$

5 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: $a_{45} = 0.913841832170866$

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

7 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.: $S = 2.07094379287543$, $|T - Q| = 0.00582854705612723$

8 Resolució d'un sistema quadrat, i un valor propi

Considereu el sistema $Ax = b$, on $A = (a_{jk})$ és la matriu 9×9 definida per $a_{jj} = \cos \sqrt{j}$, i $a_{jk} = 1/2$ si $j \neq k$, i $b = (b_j)$ ve donat per $b_j = (-1)^j$. Doneu el valor de $x_3 - x_4$, i el valor propi de valor absolut més gran de $C = A^\top(I - 3A)$.

Sol.: $x_3 - x_4 = 1.96748744204822$, $\lambda = -35.4255131277798$

9 Càlcul amb matriu i vector

Donada la matriu $A = (a_{jk})$, de dimensions 5×8 definida per $a_{jk} = \sqrt{j + k^2}$, i el vector $v = (v_j)$ definit per $v_j = \sin(j^3)$, calculeu la suma de tots els termes del vector $w = Av$.

Sol.: 47.2122695098379

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

11 Resolució d'un sistema no quadrat

Per al sistema

$$\begin{pmatrix} -2 & 0 & 5 & 3 & 2 \\ 5 & -5 & -3 & -2 & 4 \\ 1 & 2 & 4 & 5 & -1 \\ 4 & 2 & 4 & -4 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ -5 \\ 0 \\ -3 \end{pmatrix},$$

doneu un vector director v de la recta de solucions, tal que $v_5 = 1$.

Sol.: $v = (-0.120967741935484, 0.931899641577061, -0.690860215053763, 0.404121863799283, 1)$