

Departament de Matemàtica Aplicada 4 — EPSEVG
Universitat Politècnica de Catalunya

Introducció a Sage



Copyright 2011 Carles Batlle, Imma Massana
(carles.batlle@upc.edu, imma@ma4.upc.edu)

This work is licensed under a Creative Commons Attribution-Share
Alike 3.0 License. A copy of the license can be found at
<http://creativecommons.org/licenses/by-sa/3.0/>

Índex

I	Què és Sage i com utilitzar-lo	5
1	Què és Sage	5
2	Com accedir a Sage	6
3	La interfície web de Sage	6
II	Matemàtiques amb Sage	11
4	Operacions elementals	11
5	Manipulacions algebraiques	14
6	Nombres complexos, polinomis i funcions racionals	14
6.1	Nombres complexos	14
6.2	Polinomis	16
6.3	Funcions racionals	19
7	Àlgebra lineal	20
7.1	Matrius, determinants i sistemes d'equacions	20
7.2	Vectors, combinacions lineals i bases	24
7.3	Aplicacions lineals, imatges i nuclis	28
7.4	Vectors i valors propis	31
7.5	Rotacions	32
8	Càlcul	35
8.1	Funcions, dominis, gràfiques, i límits	35
8.2	Derivades i polinomis de Taylor	41
8.3	Integrals	44

Part I

Què és Sage i com utilitzar-lo

1 Què és Sage

Sage (<http://www.sagemath.org>) és un programari de codi font lliure que permet realitzar manipulacions i càlculs tant simbòlics com numèrics en moltes branques de les matemàtiques. Una de les seves característiques és que es pot instal·lar en un servidor i accedir-hi des de qualsevol dispositiu connectat a la xarxa. Això el fa especialment útil si hom vol emprar-lo per realitzar càlculs de suport al resoldre exercicis a casa, l'aula o la sala d'estudi. Val a dir, però, que Sage no és el programari d'aquest tipus més utilitzat, ni el més desenvolupat, i que Maple, Mathematica i especialment Matlab ofereixen alternatives més populars en entorns científics i industrials; no són, però, de codi font lliure (de fet són programaris comercials de pagament) i no admeten un accés web tant senzill, i això els fa menys útils per a la funció que els volem donar.

Sage està basat en Python, i per tant hereta moltes de les seves construccions. Tenir una certa familiaritat amb Python pot ser útil, sobretot si es vol efectuar càlculs que requereixin programació, però no és necessari per al nivell d'ús de Sage que es mostra en aquest manual.

En aquesta guia descriurem com crear un compte al servidor Sage de l'EPSEVG, com és la interfície, com realitzar operacions bàsiques i com efectuar alguns càlculs elementals, especialment els relatius a l'assignatura de *Fonaments Matemàtics* (FOMA) de l'EPSEVG.

Podeu trobar molta informació sobre Sage a la xarxa, i en particular a

<http://www.sagemath.org>

<http://wiki.sagemath.org>

Els resums de comandes, agrupades per temes, de

<http://wiki.sagemath.org/quickref>

poden ser molt útils.

Cal aclarir que aquesta no és una guia sistemàtica de Sage, sinó sols un recull d'allò que és més útil per a l'assignatura de FOMA. Encara que en aquests moments la major part de la informació sobre Sage està en format web, existeixen diversos textos accessibles, d'abast molt més ampli que aquest. Alguns d'ells són

- Finch, C., *Sage Beginner's Guide*, Packt Publishing, Birmingham-Mumbai (2011).
- Tábara, J.L., *Matemáticas Elementales con Sage* (2009). Es pot obtenir a

www.sagemath.org/es/Introduccion_a_SAGE.pdf

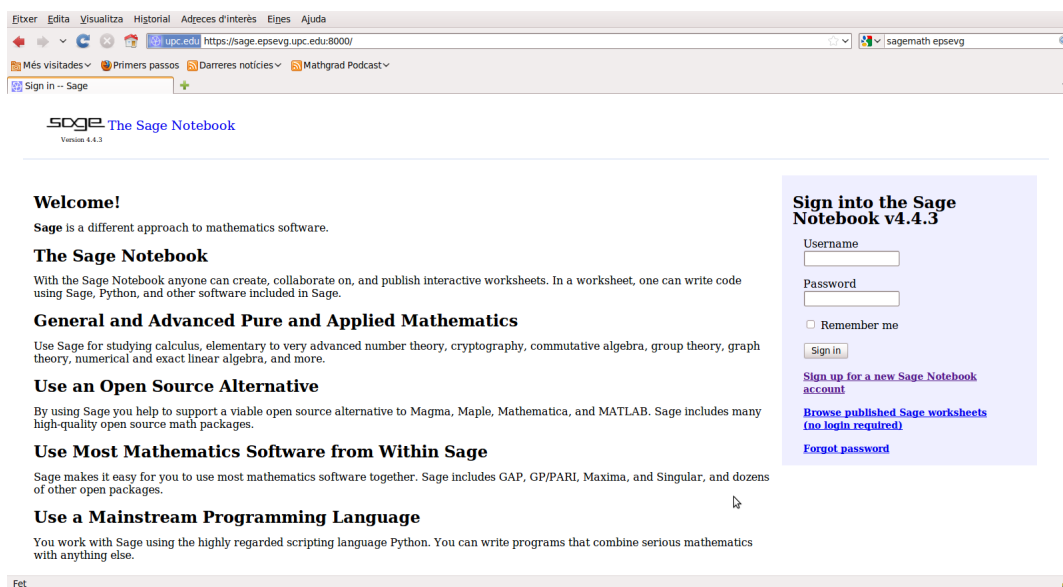


Figura 1: Pantalla d'accés a `https://sage.epsevg.upc.edu:8000`

 **Aneu amb cura si enganxeu codi directament d'aquest pdf a un full de Sage, ja que les cometes ' no es copien correctament.**

2 Com accedir a Sage

La Figura 1 mostra la pantalla d'inici del servidor de Sage de l'EPSEVG, al què s'accedeix per `https://sage.epsevg.upc.edu:8000`. Si ja teniu un compte creat podeu accedir-hi directament introduint el **Username** i **Password** i prement el **Sign in**, però si no cal que passeu primer pel formulari de **Sign up**, que apareix a la Figura 2.

Podeu triar el **Username** que vulgueu, però millor que feu servir quelcom del tipus

Nxynomcognom,

on `xy` és el vostra grup de FOMA, per exemple `N13joancollades`.

3 La interfície web de Sage

Una vegada fet el **Sign in**, apareixerà la pantalla de la Figura 3. Si és la primera vegada que entreu, o si no heu salvat cap full en sessions anteriors, veureu que a la carpeta per defecte (**Active**) no hi ha cap full de càlcul. Podeu crear un nou full de càlcul amb qualsevol dels dos enllaços [create a new worksheet](#) (en el centre) o [New Worksheet](#) (a dalt a la dreta). Apareixerà

The screenshot shows a web browser window with the URL `https://sage.epsevg.upc.edu:8000/register`. The page title is "Sign up for a Sage Notebook account". It contains four numbered steps for registration:

- 1. Create a username**
Your username must start with a letter and be between 3 and 64 characters long. You may only use letters, numbers, underscores, @, and dots.
- 2. Create a good password**
Your password must be between 4 and 32 characters long. Your password can not contain your username nor spaces.
- 3. Re-type your password**
- 4. Enter your email address**
Your email address is required for account confirmation and recovery. You will be emailed a confirmation link right after you successfully sign up.

At the bottom of the form are two buttons: "Create account" and "Cancel".

Figura 2: Formulari per crear un compte de Sage al servidor.

The screenshot shows the Sage Notebook home page for user "carlesbatlle". The URL is `https://sage.epsevg.upc.edu:8000/home/carlesbatlle/`. The page header includes the Sage logo and version "4.4.3", and navigation links: "Home", "Published", "Log", "Settings", "Help", "Report a Problem", and "Sign out".

Below the header, there are buttons for "New Worksheet", "Upload", and "Download All Active". A search bar labeled "Search Worksheets" is also present.

The main content area shows a table with columns "Active Worksheets", "Owner / Collaborators", and "Last Edited". The "Current Folder" is set to "Active". Below the table, a welcome message reads: "Welcome to Sage! You can [create a new worksheet](#), view [published worksheets](#), or read the [documentation](#)."

Figura 3: Pantalla inicial una vegada s'ha engegat la sessió.

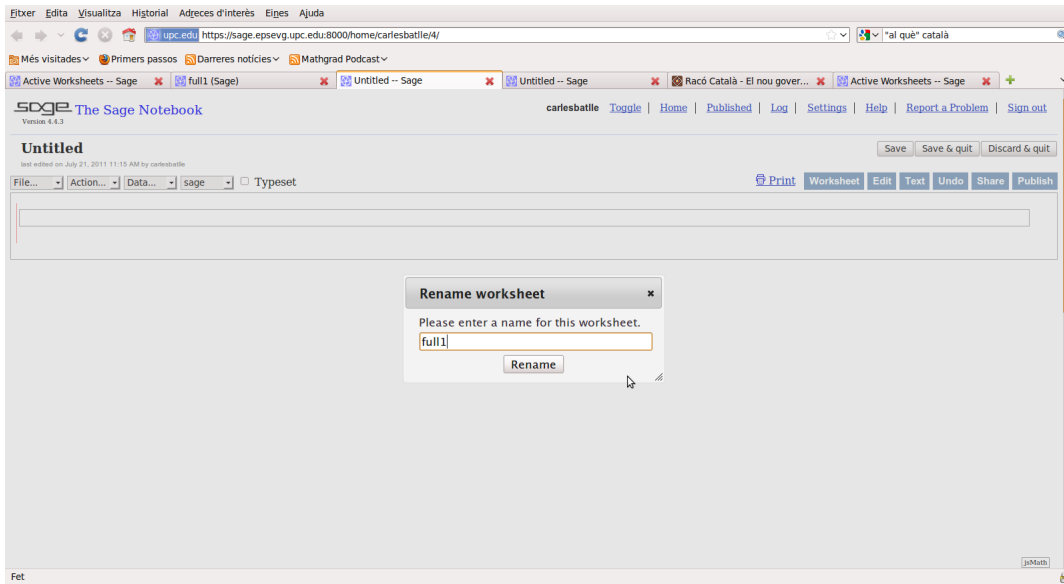


Figura 4: Formulari per anomenar el full de càlcul.

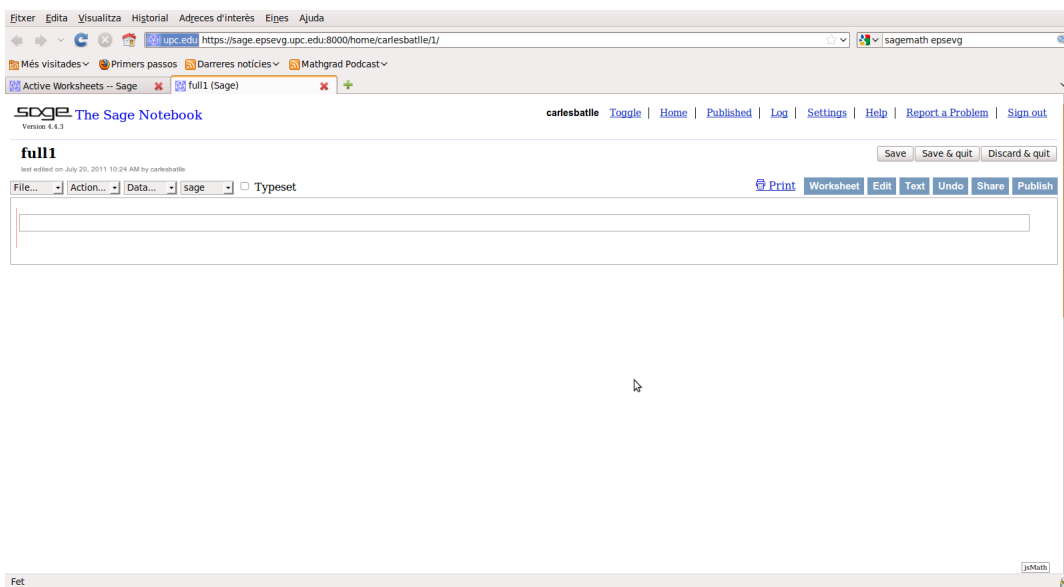
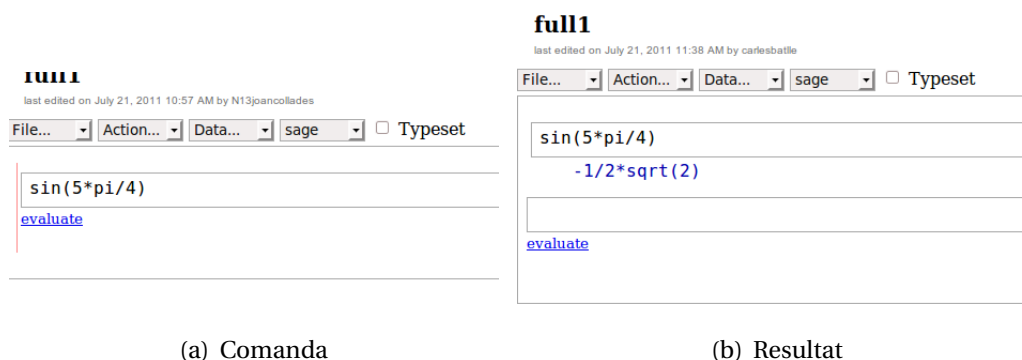
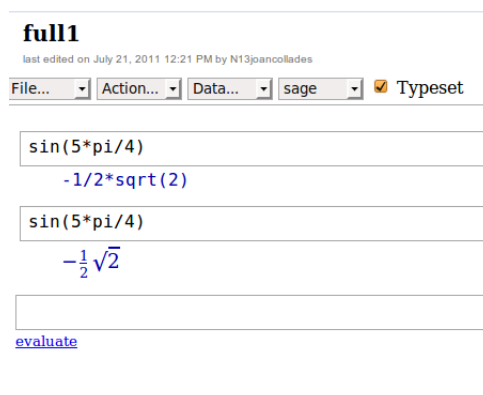


Figura 5: Full de càlcul buit.

Figura 6: Càlcul de $\sin 5\pi/4$.Figura 7: El càlcul de $\sin 5\pi/4$ amb l'opció `Typeset` activada.

la pantalla de la Figura 4, que demana el nom del nou full. Podeu posar el que vulgueu, per exemple `full1` i anireu a parar a la pantalla de la Figura 5.

Podeu entrar una comanda en el rectangle que hi ha al costat de la ratlla vertical vermell. Per exemple, si voleu calcular $\sin 5\pi/4$ heu d'entrar el que apareix a la Figura 6(a) (quan comenceu a introduir quelcom apareix l'opció `evaluate` a sota) i tot seguit avaluar-ho. Això es pot fer prement `evaluate` o, més ràpidament, amb la combinació de tecles `Shift+Enter` (amb `Enter` sols es crea una nova línia a la caixa d'entrada de comandes, en la que podeu escriure una comanda nova). En qualsevol cas, el resultat apareix a sota, tal com mostra la Figura 6(b). Fixeu-vos que dóna el resultat *exacte* $-\frac{\sqrt{2}}{2}$, i no cap aproximació numèrica.

És convenient que activeu l'opció `Typeset`, que mostra els resultats en forma visualment més atractiva, tal com mostra la Figura 7.

Si voleu afegir un nou càlcul entre altres dos, passeu el cursor entre les dues caixes fins que aparegui una ratlla horitzontal blava, tal com mostra la Figura 8(a), i cliqueu (Figura 8(b)).

És possible també posar més d'una comanda en la mateixa ratlla, separant-les amb `;`, tal

The screenshot shows the Sage Notebook interface in a browser. The address bar shows the URL `https://sage.epsevg.upc.edu:8000/home/N13joancollades/1/`. The notebook title is "full1" and it was last edited on July 21, 2011. The workspace contains two input boxes and two output boxes. The first input box contains `sin(5*pi/4)` and the first output box contains `-1/2*sqrt(2)`. The second input box contains `sin(5*pi/4)` and the second output box contains $-\frac{1}{2}\sqrt{2}$. The interface includes a menu bar with options like "File...", "Action...", "Data...", "sage", and "Typeset". There are also buttons for "Print", "Worksheet", "Edit", "Text", "Undo", "Share", and "Publish".

(a) La ratlla blava

This screenshot shows the same Sage Notebook interface as in (a), but with a new calculation added below the first one. The first calculation remains the same: input `sin(5*pi/4)` and output `-1/2*sqrt(2)`. Below it, there is a new input box containing `sin(5*pi/4)` and a new output box containing $-\frac{1}{2}\sqrt{2}$. The "evaluate" button is visible between the two calculations. The interface elements like the menu bar and buttons are the same as in (a).

(b) La nova caixa

Figura 8: Com afegir un nou càlcul entre altres dos.

com mostra l'entrada següent

```
sin(3*pi/5);e^(cos(pi))
```

$$\sin\left(\frac{3}{5}\pi\right)$$

$$e^{(-1)}$$

Finalment, és possible recuperar el resultat de la darrera operació en un nou càlcul amb el símbol `_`:

```
2*cos(pi/6)-2
```

$$\sqrt{3}-2$$

```
_*(4-4/5+1/12)
```

$$\frac{197}{60}\sqrt{3}-\frac{197}{30}$$

La interfície web té moltes opcions que podeu explorar lliurement. Podeu, per exemple, guardar el full de càlcul en un fitxer (Sage salva els fulls amb l'extensió `sws`, que després pot carregar-se en un altre servidor Sage (o en el de l'EPSEVG) amb el botó [Upload](#). Això és útil si instal·leu Sage en una màquina pròpia (preferentment sota Ubuntu). Si simplement voleu salvar els vostres càlculs en el servidor, cal que aneu al botó `Save` o `Save & quit`, i la propera vegada que entreu al servidor trobareu el full tal com l'heu deixat, però amb els càlculs sense realitzar.

Quan acabeu és convenient que feu `Save & quit`, ja que en cas contrari el full continua actiu encara que feu un [Sign out](#), i això consumeix recursos innecessàriament. Recordeu, però, que amb `Save & quit` els resultats dels càlculs realitzats desapareixen, i si voleu recuperar-los s'han de executar de nou les comandes la següent vegada que treballeu amb el full.

Part II

Matemàtiques amb Sage

Les funcions de Sage adopten el format `objecte.funció()` de Python. Així, si volem calcular el determinant de la matriu `A`, podem fer `A.det()`. Com que això no és la notació matemàtica habitual, Sage accepta també `det(A)`. Això passa per la majoria de funcions matemàtiques, però no sempre. Per exemple, el càlcul de la matriu inversa sols admet la forma `A.inverse()`.

4 Operacions elementals

Sage pot utilitzar-se a nivell elemental com una calculadora avançada, amb la particularitat que, a més de poder representar els reals mitjançant aproximacions decimals, és capaç de tractar-los com a tals, sense aproximació. Per exemple

```
pi+2
```

$\pi + 2$

Si voleu un resultat aproximat heu d'emprar la funció `numerical_approx`, que es pot abreujar a `N` o `n`:

```
N(pi+2)
```

5.14159265358979

Per defecte apareixen 16 dígits, però això es pot canviar:

```
n(e, digits=30)
```

2.71828182845904523536028747135

L'assignació de variables es fa amb `=`

```
a=5
```

```
sqrt(a)
```

$\sqrt{5}$

mentre que `==` avalua la condició

```
3==2
```

False

Podeu eliminar el valor d'una variable amb `del`

```
del a
```

```
sqrt(a)
```

Traceback (click to the left of this block for traceback)

...

NameError: name 'a' is not defined

Si voleu eliminar totes les variables podeu fer-ho amb `reset()`.

L'exponenciació es pot fer amb `**`, com a Python, o emprant el sinònim `^` (no vàlid a Python)

```
4**5
```

1024

```
3.2^4
```

104.857600000000

Podeu també fer operacions exactes amb racionals i obtenir la forma més simplificada. Per exemple, per calcular

$$\frac{3}{\frac{1}{8} - \frac{1}{5}} + \frac{27}{12} \left(\frac{1}{4} - \frac{5}{9} \right)$$

heu de fer

```
3 / (1/8 - 1/5) + 27/12 * (1/4 - 5/9)
```

$$-\frac{651}{16}$$

i es pot operar també amb complexos, emprant I com a unitat imaginària:

```
(3+I) / (2-3*I)
```

$$\frac{11}{13}I + \frac{3}{13}$$

A més d'aquests tipus d'operacions, Sage és especialment potent manipulant i operant nombres naturals. Tot seguit es mostren exemples d'algunes funcions d'interès.

divisors Divisors d'un natural donat

```
divisors(124)
```

$$[1, 2, 4, 31, 62, 124]$$

factorial Factorial $n! = n \cdot (n-1) \cdot (n-2) \cdots 3 \cdot 2 \cdot 1$ d'un nombre natural

```
factorial(21)
```

$$51090942171709440000$$

lcm Mínim comú múltiple d'una llista de naturals

```
lcm([3, 22, 11, 54, 101])
```

$$59994$$

gcd Màxim comú divisor d'una llista de naturals

```
gcd([45, 123, 12])
```

$$3$$

primes_first_n La llista dels n primers nombres primers (podeu emprar-ho per veure si la numeració de bústies de personal de l'EPSEVG és correcta)

```
primes_first_n(25)
```

$$[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]$$

eratosthenes Retorna una llista de tots els primers més petits o iguals que n

```
eratosthenes(100)
```

$$[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]$$

5 Manipulacions algebraiques

Sage permet manipular expressions simbòliques. Per exemple, si volem expandir $(a + b + c)^3$, primer declarem a , b i c com a variables simbòliques

```
var('a b c')
```

i definim una nova variable $d = (a + b + c)^3$

```
d=(a+b+c)^3
```

Això, al ser una assignació, no treu cap resultat, però ara podem fer

```
expand(d)
```

$$a^3 + 3a^2b + 3a^2c + 3ab^2 + 6abc + 3ac^2 + b^3 + 3b^2c + 3bc^2 + c^3$$

L'operació recíproca d'expandir la fa `factor`

```
factor(1/(a+b)+2/(a-b)+a/(a+2*b))
```

$$\frac{a^3 - ab^2 + 3a^2 + 7ab + 2b^2}{(a-b)(a+b)(a+2b)}$$

Les simplificacions de funcions trigonomètriques s'han de fer requerint expressament les identitats que aquestes obeeixen.

```
var('a')
```

a

```
p=sin(3*a)
```

```
show(p)
```

$\sin(3a)$

```
p.expand_trig()
```

$$-\sin(a)^3 + 3\sin(a)\cos(a)^2$$

Aquí, emprant la notació d'objectes de Python, s'ha fet servir el mètode `expand_trig()` de la classe a la que pertany l'objecte p . L'altre mètode important és `simplify_trig()`:

```
q=(cos(a))^2+(sin(a))^2
```

```
q.simplify_trig()
```

1

6 Nombres complexos, polinomis i funcions racionals

6.1 Nombres complexos

Com ja hem dit, I representa la unitat imaginària j , però també es pot emprar i , encara que per als resultats sempre es mostra I :

```
exp(pi*I)+exp(pi*i/2)
```

$$I - 1$$

En alguns casos Sage dóna directament la forma cartesiana simplificada

```
(3+I) * (5-I)
```

$$2I + 16$$

però normalment cal obligar-lo a simplificar:

```
I*e^(I*pi/4)
```

$$I e^{(\frac{1}{4}I)\pi}$$

```
simplify(I*e^(I*pi/4))
```

$$\left(\frac{1}{2}I - \frac{1}{2}\right) \sqrt{2}$$

La llista següent mostra les operacions més importants que es poden fer amb complexos.

conjugate Calcula el complex conjugat

```
conjugate(3+I)
```

$$-I + 3$$

abs Calcula el mòdul

```
abs(3+I)
```

$$\sqrt{10}$$

real Calcula la part real

```
real((3+I)^5)
```

$$-12$$

imag Calcula la part imaginària

```
imag((3+I)/(2-I))
```

$$1$$

Hom pot obtenir l'argument, o angle, d'un nombre complex emprant aquestes funcions i la funció `atan2`, que és l'arctangent però calcula correctament el quadrant tenint en compte els signes de la part real i imaginària. Per exemple, si volem calcular l'argument de $-1 + j$,

```
z=-1+I
```

```
atan2(imag(z), real(z))
```

$$\frac{3}{4}\pi$$

i la forma exponencial de $-1 + j$ seria

```
abs(z) * e^(I * atan2(imag(z), real(z)))
```

$$\sqrt{2} e^{(\frac{3}{4}I)\pi}$$

Es pot passar de la forma exponencial a la cartesiana emprant el mètode `expand_trig()`:

```
(2 * exp(2 * pi / 3 * I)).expand_trig()
```

$$I\sqrt{3} - 1$$

6.2 Polinomis

Sage permet definir polinomis usant comandes molt potents, però de cara als problemes que apareixen a FOMA definirem els polinomis com a funcions normals.

Veurem en primer lloc com efectuar una descomposició en arrels. Sigui el polinomi de cinquè grau

$$P(x) = x^5 + 9x^4 + 36x^3 + 80x^2 + 96x + 48.$$

El definim a Sage amb

```
P(x) = x^5 + 9*x^4 + 36*x^3 + 80*x^2 + 96*x + 48
```

Veiem que $P(-2) = 0$

```
P(-2)
```

$$0$$

i per tant $P(x)$ és divisible per $x + 2$:

```
(P(x) / (x+2)).simplify_full()
```

$$x^4 + 7x^3 + 22x^2 + 36x + 24$$

Aquí hem fet servir el procediment `simplify_full()` per aconseguir que realment faci la divisió. Ara assignem el resultat a un nou polinomi $Q(x)$, emprant el símbol `_`, que identifica el darrer resultat:

```
Q(x) = _
```

Si calculem $Q(-2)$ veiem que -2 també és arrel

```
Q(-2)
```

$$0$$

Tornem a dividir, i assignem directament el resultat a un polinomi, $M(x)$:

```
M(x) = (Q(x) / (x+2)).simplify_full()
```

Com que les assignacions no mostren el resultat en pantalla, demanem explícitament que ens mostri $M(x)$:

```
M(x)
```

$$x^3 + 5x^2 + 12x + 12$$

Aquest polinomi torna a ser divisible per $x + 2$, ja que es pot veure que $M(-2) = 0$. Tornem per tant a dividir

```
N(x) = (M(x) / (x+2)) . simplify_full()
```

```
N(x)
```

$$x^2 + 3x + 6$$

Calculem ara les arrels d'aquest polinomi utilitzant la funció `solve`, i emprant la notació de diccionaris de Python (això és per a que sigui més fàcil manipular els resultats)

```
solve(N(x)==0, x, solution_dict=True)
```

$$\left\{ \left\{ x: \left(-\frac{1}{2}I\right) \sqrt{15} - \frac{3}{2} \right\}, \left\{ x: \left(\frac{1}{2}I\right) \sqrt{15} - \frac{3}{2} \right\} \right\}$$

$N(x)$ té dues arrels complexes conjugades. Posem el diccionari resultant dins una variable `sols` i tot seguit assignem els valors de les arrels a dues variables `a` i `b` (en Sage els índex comencen per zero)

```
sols=_
```

```
a=sols[0][x]; b=sols[1][x]
```

Podem ara escriure les descomposicions de $P(x)$ sobre els reals

```
PR(x) = (x+2)^3 * N(x); PR(x)
```

$$(x+2)^3(x^2 + 3x + 6)$$

i sobre els complexos

```
PC(x) = (x+2)^3 * (x-a) * (x-b); PC(x)
```

$$\frac{1}{4}(x+2)^3(2x - I\sqrt{15} + 3)(2x + I\sqrt{15} + 3)$$

(aquí Sage ha tret un $\frac{1}{2}$ de cada factor complex; en realitat hauria d'estar dins els factors, de manera que el coeficient de x sigui 1). Podem comprovar que el resultat és correcte, tant per al cas real com per al complex:

```
expand(PR(x)); simplify(expand(PC(x)))
```

$$x^5 + 9x^4 + 36x^3 + 80x^2 + 96x + 48$$

$$x^5 + 9x^4 + 36x^3 + 80x^2 + 96x + 48$$

Anem ara a veure com es poden calcular totes les arrels d'un polinomi en casos més complexos. Hi ha fórmules generals per a les arrels de polinomis fins a grau 4, però en general les solucions són força complicades. Per exemple, les arrels de $P(x) = x^4 + x^3 + 2x + 1$ es poden calcular amb

```
sols=solve(x^4+x^3+2*x+1==0, x, solution_dict=True)
```

i podem fer que ens les ensenyi amb

```
for i in range(4): sols[i][x]
```

El resultat, però, és tant complicat que no val la pena reproduir-lo aquí. El que sí que podem fer és que s'ens mostri una aproximació numèrica d'aquest càlcul exacte:

```
for i in range(4): (sols[i][x]).n()
-1.55897987798175
-0.472212951658699
0.515596414820225 - 1.04524701808165i
0.515596414820225 + 1.04524701808165i
```

Veiem per tant que hi ha dues arrels reals i dues de complexes conjugades. Tal com hem dit, excepte en casos especials, no es poden calcular les arrels d'un polinomi de grau més gran o igual que 5 de forma exacte. Per exemple, si $T(x) = x^5 + x^3 + x + 2$,

```
sols=solve(x^5+x^3+x+2==0, x, solution_dict=True)
```

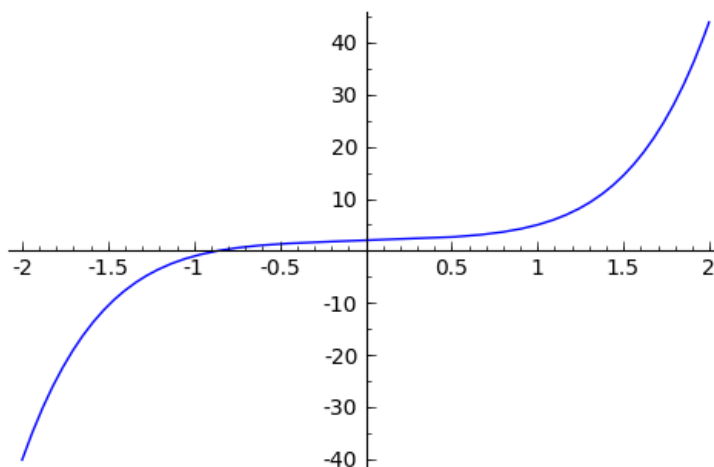
no retorna res

```
sols
```

```
[{0: x5 + x3 + x + 2}]
```

En aquests casos, les arrels s'han de calcular numèricament de manera aproximada. Això és fàcil en el cas d'arrels reals. Per veure quantes n'hi ha (com que és un polinomi de grau senar com a mínim n'hi ha una) dibuixem la seva gràfica en l'interval $[-2, 2]$ (tenint en compte el grau i els coeficients del polinomi, es pot veure que no pot haver-hi cap arrel real fora d'aquest interval):

```
plot(T(x), (x, -2, 2))
```



Per calcular l'arrel, cridem la funció `find_root`, dient-li que la busqui entre -2 i 0 :

```
find_root(T(x)==0, -2, 0)
```

```
-0.865582429789
```

El càlcul numèric de les arrels reals s'ha de fer una per una, donant l'interval on es creu que estan. El càlcul numèric de les arrels complexes és molt més complicat.

6.3 Funcions racionals

Veurem aquí com descomposar una funció racional en fraccions simples sobre els reals. Sigui per exemple

$$M(x) = \frac{x^5 + 1}{x^4 - 2x^3 + 2x^2 - 2x + 1}.$$

La introduïm a Sage amb

```
P(x)=x^5+1; Q(x)=x^4-2*x^3+2*x^2-2*x+1; M(x)=P(x)/Q(x)
```

```
M(x)
```

$$\frac{x^5+1}{x^4-2x^3+2x^2-2x+1}$$

Observem en primer lloc que el numerador és de grau igual o més gran que el denominador. L'expressió de $M(x)$ en termes de fraccions simples contindrà, per tant, a més de les fraccions, un polinomi de grau 1. En segon lloc, els zeros del denominador són

```
solve(Q(x)==0, x)
```

$$[x = (-1), x = 1, x = 1]$$

Com que hi ha les arrels complexes conjugades $\pm j$ i l'arrel real 1, i com que hi ha d'haver en total quatre arrels i el polinomi té coeficients reals, l'arrel 1 té multiplicitat 2. En la descomposició en fraccions simples hi apareixeran $\frac{1}{x-1}$ i $\frac{1}{(x-1)^2}$, a més de $\frac{1}{x^2+1}$, ja que $(x-j)(x+j) = x^2 + 1$. Tot això es pot obtenir, juntament amb els coeficients de la descomposició, amb el procediment `partial_fraction` aplicat a $M(x)$:

```
(M(x)).partial_fraction(x)
```

$$x + \frac{x-1}{2(x^2+1)} + \frac{3}{2(x-1)} + \frac{1}{(x-1)^2} + 2$$

La descomposició de funcions racionals en fraccions simples és un tema complicat per als manipuladors algebraics en el cas que hi hagi arrels reals no racionals. Per exemple, les arrels del polinomi $x^4 + 3x^3 - 3x^2 - 15x - 10$ són $\pm\sqrt{5}$, -1 , -2 . Com que Sage treballa per defecte sobre els racionals la descomposició sols dóna el resultat final per a les arrels -1 i -2 :

```
R(x)=x^4+3*x^3-3*x^2-15*x-10
```

```
(1/R(x)).partial_fraction(x)
```

$$\frac{3x-7}{4(x^2-5)} + \frac{-1}{4(x+1)} + \frac{1}{x+2}$$

El càlcul no descomposa la fracció amb el denominador $x^2 - 5$, que finalment donarà lloc a fraccions amb denominadors $x - \sqrt{5}$ i $x + \sqrt{5}$. Aquest darrer pas s'ha de fer a ma, però és molt més senzill que fer-ho tot manualment des del començament. Val a dir també que, de cara a les aplicacions, i si hom està disposat a treballar amb les funcions hiperbòliques, no

cal descomposar aquest tipus de fraccions, de la mateixa manera que no es descomposen les fraccions amb denominadors de grau 2 amb arrels complexes conjugades.

7 Àlgebra lineal

7.1 Matrius, determinants i sistemes d'equacions

Les matrius s'introdueixen en Sage per files, amb la instrucció `matrix`. Per exemple, la matriu

$$\begin{pmatrix} 2 & 2 & 0 \\ 2 & 3 & -1 \\ 0 & 3 & 4 \end{pmatrix}$$

es defineix amb

```
A=matrix([[2,2,0],[2,3,-1],[0,3,4]])
```

El determinant d'una matriu quadrada es calcula amb `det`:

```
det(A)
```

14

Com que el determinant és diferent de zero, qualsevol sistema d'equacions de la forma

$$Ax = b$$

té solució única, que es pot calcular amb `solve_right` aplicat a A i amb argument igual al terme independent b . Per exemple, si volem solucionar el sistema

$$\begin{pmatrix} 2 & 2 & 0 \\ 2 & 3 & -1 \\ 0 & 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ 0 \\ -1 \end{pmatrix}$$

és a dir,

$$\begin{cases} 2x_1 + 2x_2 & = \frac{1}{2} \\ 2x_1 + 3x_2 - x_3 & = 0 \\ 3x_2 + 4x_3 & = -1, \end{cases}$$

hem de fer

```
x=A.solve_right(vector([1/2,0,-1]));x
```

$(\frac{19}{28}, -\frac{3}{7}, \frac{1}{14})$

Quan la matriu del sistema té determinant no nul és possible també obtenir la solució actuant amb la matriu inversa, encara que això no és aconsellable, degut als errors numèrics que introdueix, per a sistemes de dimensió molt gran:

```
x=A.inverse()*vector([1/2,0,-1]);x
```

$$\left(\frac{19}{28}, -\frac{3}{7}, \frac{1}{14}\right)$$

Si el sistema és compatible però indeterminat (infinites solucions), Sage en dóna una de particular. Per exemple, el sistema de dues equacions amb tres incògnites

$$\begin{cases} x_1 + 2x_2 & = \frac{1}{2} \\ 2x_1 + 3x_2 - x_3 & = 0 \end{cases}$$

és

```
B=matrix([[1, 2, 0], [2, 3, -1]])
```

```
B.solve_right(vector([1/2, 0]))
```

$$\left(-\frac{3}{2}, 1, 0\right)$$

Finalment, si el sistema és incompatible, com ara el donat per

$$\begin{cases} x_1 - x_2 & = 1 \\ & 3x_2 + 4x_3 = 0 \\ x_1 + 2x_2 + 4x_3 & = 2 \end{cases}$$

el sistema retorna un missatge d'error:

```
C=matrix([[1, -1, 0], [0, 3, 4], [1, 2, 4]])
```

```
C.solve_right(vector([1, 0, 2]))
```

[Traceback \(click to the left of this block for traceback\)](#)

...

[ValueError: matrix equation has no solutions](#)

Que el darrer sistema no té solució es pot veure calculant el rang de C i el rang de la matriu ampliada

$$aC = \left(\begin{array}{ccc|c} 1 & -1 & 0 & 1 \\ 0 & 3 & 4 & 0 \\ 1 & 2 & 4 & 2 \end{array} \right).$$

Aquesta matriu es construeix amb el procediment `augment`, afegint-li una darrera columna que conté el vector que forma el terme independent del sistema. Com que Sage afegeix els vectors a les matrius com si fossin files, s'ha de transposar:

```
aC=C.augment(transpose(vector([1, 0, 2])))
```

El rang d'una matriu es calcula amb `rank`:

```
rank(C)
```

2

```
rank(aC)
```

3

Com que $\text{rang}(aC) > \text{rang}(C)$ el sistema és incompatible. Podem fer més explícita la incompatibilitat efectuant operacions amb files fins a obtenir una forma reduïda, amb zeros per sota de la (pseudo)diagonal principal. Donada una matriu A , les operacions bàsiques són

- Multiplicar la fila i per un escalar $a \neq 0$: $A.\text{rescale_row}(i, a)$.
- Afegir a la fila i la fila j multiplicada per un escalar a : $A.\text{add_multiple_of_row}(i, j, a)$.
- Permutar les files i i j : $A.\text{swap_rows}(i, j)$.

Per exemple, donada la matriu aC , podem començar permutant les files 2 i 3 (recordem que en Python i en Sage els índexs comencen per zero i per tant, de fet, hem de permutar les files d'índex 1 i 2):

```
aC.swap_rows(1, 2); aC
```

$$\begin{pmatrix} 1 & -1 & 0 & 1 \\ 1 & 2 & 4 & 2 \\ 0 & 3 & 4 & 0 \end{pmatrix}$$

A la fila 2 li podem restar la 1:

```
aC.add_multiple_of_row(1, 0, -1); aC
```

$$\begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 3 & 4 & 1 \\ 0 & 3 & 4 & 0 \end{pmatrix}$$

Finalment, a la fila 3 li podem sotstreure la 2:

```
aC.add_multiple_of_row(2, 1, -1); aC
```

$$\begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 3 & 4 & 1 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

La darrera fila és una contradicció i això demostra la incompatibilitat.

És possible resoldre diversos sistemes d'equacions amb matriu A comú, però diferent vector b , simultàniament. Per exemple, suposem que volem calcular la solució de

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 4 & 5 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = b$$

per als vectors b

$$\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 2 \\ 3 \end{pmatrix}.$$

Per fer-ho, introduïm els vectors com a columnes d'una matriu b :

```
A=matrix([[ -1, 0, 1], [-2, 4, 5], [0, 1, 1]])
```

```
b=transpose(matrix([[1, 0, 1], [0, 2, 3]]))
```

on hem aplicat `transpose` ja que `matrix` construeix la matriu per files. Com que $\det A = -1 \neq 0$ el sistema té solució única per a qualsevol terme independent, i calculem les dues solucions amb¹

```
sols=A.solve_right(b);sols
```

$$\begin{pmatrix} 5 & 10 \\ -5 & -7 \\ 6 & 10 \end{pmatrix}$$

Podem seleccionar una de les dues solucions amb `matrix_from_columns`. Per exemple, la solució corresponent al primer vector és

```
soll=sols.matrix_from_columns([0]);soll
```

$$\begin{pmatrix} 5 \\ -5 \\ 6 \end{pmatrix}$$

i, efectivament,

```
A*soll
```

$$\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

Per a sistemes compatibles determinats, és possible calcular components individuals de la solució emprant la regla de Cramer. Per exemple, sigui

$$\begin{pmatrix} 1 & 2 & -1 \\ 0 & -2 & 3 \\ 3 & 0 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 3 \end{pmatrix}.$$

Introduïm la matriu A com

```
A=matrix([[1, 2, -1], [0, -2, 3], [3, 0, -1]])
```

i el vector columna b com

```
b=transpose(vector([1, -1, 3]))
```

Com que

```
detA=det(A);detA
```

¹En els exemples anterior havíem passat a `solve_right` un vector sense transposar, però si el vector que se li passa és un vector columna, o un conjunt de vectors columna, `solve_right` ho interpreta correctament i retorna el resultat també en forma de vector columna.

el sistema és compatible i determinat. Si volem calcular x_2 , hem de formar la matriu que s'obté substituint a A la segona columna pel vector columna b . La manera més fàcil de fer això és construint primer la matriu ampliada i després seleccionant les columnes que volem, i en l'ordre que volem, amb `matrix_from_columns`:

```
aA=A.augment(b);aA
```

$$\begin{pmatrix} 1 & 2 & -1 & 1 \\ 0 & -2 & 3 & -1 \\ 3 & 0 & -1 & 3 \end{pmatrix}$$

```
num2=aA.matrix_from_columns([0,3,2]);num2
```

$$\begin{pmatrix} 1 & 1 & -1 \\ 0 & -1 & 3 \\ 3 & 3 & -1 \end{pmatrix}$$

Llavors, emprant la regla de Cramer

```
x2=det(num2)/detA;x2
```

$$-\frac{1}{7}$$

7.2 Vectors, combinacions lineals i bases

Al començament d'aquesta subsecció presentem els resultats de Sage suposant que el Typeset no està activat. Si el teniu activat la presentació pot ser una mica diferent, probablement amb més informació.

Els vectors es defineixen amb `vector`. Per exemple, si volem introduir els vectors

$$u_1 = \begin{pmatrix} 1 \\ -3 \\ 5 \end{pmatrix}, \quad u_2 = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

hem de fer

```
u1=vector([1,-3,5]);u2=vector([0,1,2])
```

La suma de vectors i la multiplicació per escalars es fa llavors de forma natural:

```
u1+u2
```

$$(1, -2, 7)$$

```
3*u1-5*u2
```

$$(3, -14, 5)$$

El conjunt de combinacions lineals d'un conjunt de vectors, o subespai generat per aquest conjunt, es calcula amb `span`:

```
W=span([u1,u2])
```

Podem saber si un vector pertany al subespai generat per u_1 i u_2 preguntant-ho directament. Per exemple, si

$$u_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

fem

```
u3=vector([1,1,1])
```

```
u3 in W
```

False

En canvi,

```
u4=vector([-1,2,-7])
```

```
u4 in W
```

True

Podem calcular la combinació lineal de u_1 i u_2 que dóna u_4 escrivint totes les combinacions lineals de u_1 i u_2

```
var('a b'); a*u1+b*u2
```

(a, b)

(a, -3*a + b, 5*a + 2*b)

igualant el resultat a les components de u_4 i solucionant el sistema:

```
solve([a==-1, -3*a+b==2, 5*a+2*b==-7], a, b)
```

[[a == -1, b == -1]]

Per tant, $u_4 = -u_1 - u_2$. Si demaneu informació sobre W obtindreu

```
W
```

Free module of degree 3 and rank 2 over Integer Ring

Echelon basis matrix:

[1 0 11]

[0 1 2]

En un llenguatge una mica avançat, això està dient que el subespai W està generat per vectors de 3 components i té dimensió 2, i que una base del mateix, en una certa forma especial, és

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 11 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}.$$

Un altre exemple, a \mathbb{R}^4 , és

```
u1=vector([-1,0,1,1]); u2=vector([0,1,2,1])
```

```
u3=vector([1,-1,1,-1]);u4=vector([1,1,1,0])
```

```
W=span([u1,u2,u3,u4])
```

```
W
```

Free module of degree 4 and rank 3 over Integer Ring

Echelon basis matrix:

```
[1 0 3 0]
```

```
[0 1 2 1]
```

```
[0 0 4 1]
```

Es tracte per tant d'un subespai de dimensió 3 amb vectors de 4 components, amb base

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 3 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 1 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 0 \\ 0 \\ 4 \\ 1 \end{pmatrix}.$$

Si sols volem la dimensió de W podem fer-ho amb `dimension`:

```
dimension(W)
```

```
3
```

La indicació **over Integer Ring** que apareix cada vegada que demanem informació sobre el subespai generat pel vectors ens adverteix que sols s'estan considerant combinacions lineals amb coeficients enters. Això és important, ja que, per exemple, en aquest espai el vector

$$v = \begin{pmatrix} 1/4 \\ 0 \\ 3/4 \\ 0 \end{pmatrix}$$

no és combinació lineal de

$$w = \begin{pmatrix} 1 \\ 0 \\ 3 \\ 0 \end{pmatrix}$$

donat que el coeficient $1/4$ que dona v en termes de w no és un enter:

```
w=vector([1,0,3,0])
```

```
v=vector([1/4,0,3/4,0])
```

```
W=span([w])
```

```
v in W
```

```
False
```

Com que això no és el que volem normalment, cal dir-ho al definir els vectors, bé posant coeficients expressament racionals o reals, o indicant que estem en els racionals o els reals. Per exemple, si volem treballar amb els racionals, hem de fer servir l'argument $\mathbb{Q}\mathbb{Q}$:

```
u=vector([1, 1/3, 0, -2/5], QQ); u
```

```
(1, 1/3, 0, -2/5)
```

```
U=span([u]); U
```

```
Vector space of degree 4 and dimension 1 over Rational Field
```

```
Basis matrix:
```

```
[1 1/3 0 -2/5]
```

En canvi, sobre els reals cal emprar $\mathbb{R}\mathbb{R}$:

```
u=vector([1, 1/3, 0, -2/5], RR); u
```

```
(1.000000000000000, 0.333333333333333, 0.000000000000000, -0.400000000000000)
```

```
U=span([u]); U
```

```
Vector space of degree 4 and dimension 1 over Real Field with 53 bits of precision
```

```
Basis matrix:
```

```
[1.000000000000000 0.333333333333333 0.000000000000000 -0.400000000000000]
```

(els 53 bits es corresponen a una representació amb 16 decimals)

Encara que nosaltres podem considerar que els dos vectors que hem definit són el mateix, en realitat són objectes diferents i Sage els tracte com a tals. En la majoria d'aplicacions d'enginyeria (però no en totes, per exemple en les relacionades amb alguns àmbits TIC i d'expressió gràfica, on sols els enters tenen sentit) caldria treballar sobre els reals, però com que això introdueix la representació decimal, que és molt llarga i poc estètica, treballarem sobre els racionals \mathbb{Q} , entesos com un subconjunt dels reals \mathbb{R} que els aproxima amb precisió tant gran com calgui.

Re-activem ara el Typeset

Sigui el conjunt de vectors de \mathbb{R}^4 (de fet de \mathbb{Q}^4)

```
u1=vector([1/4, 1, -1, 2], QQ); u2=vector([0, 1, -2, 1], QQ)
```

```
u3=vector([1, 1/3, -1, 1], QQ); u4=vector([1, 1, 0, 1], QQ)
```

```
W=span([u1, u2, u3, u4])
```

```
dimension(W)
```

```
4
```

Com que la dimensió del subespai generat pels quatre vectors és 4, són linealment independents, i per tant formen una base de \mathbb{R}^4 . Suposem ara que volem calcular les components del vector

```
v=vector([1, 4, -2, 0], QQ)
```

en aquesta base. Una manera de fer-ho és escriure una combinació lineal qualsevol dels vectors u_1, u_2, u_3 i u_4 i igualar-la a x . Això, però, és equivalent a resoldre el sistema

$$Ax = v$$

on x és el vector amb les components que cerquem i A és la matriu que té per columnes els vectors u_i . Amb Sage és fàcil crear una matriu a partir de vectors, però per defecte la crea posant els vectors com a files, que no és el que volem. Cal per tant transposar el resultat:

```
A=transpose(matrix([u1, u2, u3, u4]));A
```

$$\begin{pmatrix} \frac{1}{4} & 0 & 1 & 1 \\ 1 & 1 & \frac{1}{3} & 1 \\ -1 & -2 & -1 & 0 \\ 2 & 1 & 1 & 1 \end{pmatrix}$$

Una vegada tenim la matriu A , cal aplicar-li el procediment `solve_right` per trobar el vector x que, multiplicat per la dreta d' A , doni v :

```
x=A.solve_right(v)
```

```
x
```

$$\left(-\frac{5}{2}, \frac{27}{8}, -\frac{9}{4}, \frac{31}{8}\right)$$

Podem veure que, efectivament, això dóna la solució correcta:

```
x[0]*u1+x[1]*u2+x[2]*u3+x[3]*u4
```

$$(1, 4, -2, 0)$$

7.3 Aplicacions lineals, imatges i nuclis

Donada una matriu associada a una aplicació lineal, podem calcular el subespai imatge i el subespai nucli amb els procediments `column_space()` i `right_kernel()`. Per exemple, sigui l'aplicació de \mathbb{R}^3 en \mathbb{R}^2 donada per la matriu

$$A = \begin{pmatrix} 1 & 3 & 4 \\ 2 & 3 & -1 \end{pmatrix}.$$

Igual que passava al discutir els espais vectorials, cal indicar a on pertanyen els elements de la matriu (en cas contrari, i com que aquesta matriu sols té enters, Sage suposarà que sols volem treballar amb enters). Per les raons ja discutides, volem treballar sobre els racionals, i escrivim

```
A=matrix(QQ, [[1, 3, 4], [2, 3, -1]]);A
```

$$\begin{pmatrix} 1 & 3 & 4 \\ 2 & 3 & -1 \end{pmatrix}$$

Per calcular el subespai imatge fem

```
imatge=A.column_space()
```

```
imatge
```

$$\text{RowSpan}_{\mathbb{Q}} \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right)$$

Això vol dir que una base de la imatge està formada pels vectors (que aquí apareixen com a **files**, fet indicat per RowSpan)

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

i per tant la imatge és tot \mathbb{R}^2 , fet que també pot obtenir-se directament amb

```
rank(A)
```

2

Per calcular el nucli fem

```
nucli=A.right_kernel()
```

```
nucli
```

$$\text{RowSpan}_{\mathbb{Q}} \left(1 \quad -\frac{3}{5} \quad \frac{1}{5} \right)$$

que vol dir que el nucli té dimensió 1 i una base del mateix és

$$\begin{pmatrix} 1 \\ -\frac{3}{5} \\ \frac{1}{5} \end{pmatrix}.$$

Si no haguéssim definit la matriu sobre \mathbb{Q} , hauria donat el vector amb components

$$\begin{pmatrix} 5 \\ -3 \\ 1 \end{pmatrix}.$$

Quan calculem la imatge amb `column_space()` Sage ens proporciona una base de la imatge formada, normalment, per vectors diferents dels que apareixen a les columnes de la matriu. Si volem tenir una base formada per vectors columna originals de la matriu, podem emprar la funció `pivots()`, que retorna una llista amb els índexs (començant com sempre per 0) de les columnes que són linealment independents. Per exemple, si

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

$$\begin{pmatrix} 1 & 1 & 2 & -1 & 4 \\ 2 & 0 & 2 & 3 & -2 \\ -1 & -1 & -2 & 5 & 0 \end{pmatrix}$$

```
A.pivots()
```

[0, 1, 3]

que indica que els vectors

$$\begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 3 \\ 5 \end{pmatrix}$$

formen una base de la imatge.

Per a sistemes compatibles indeterminats la solució general es pot calcular, a més de posant el sistema en forma reduïda i prenent les variables *sobrants* com a paràmetres, calculant una solució particular i afegint-t'hi tots els vectors del nucli de l'aplicació lineal associada. Sigui per exemple el sistema de 3 equacions amb 5 incògnites

$$\begin{pmatrix} -1 & 5 & -1 & -1 & 6 \\ 1 & -3 & 3 & 4 & -1 \\ 0 & 1 & 4 & 6 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}$$

que introduïm amb

```
A = matrix([[ -1, 5, -1, -1, 6], [1, -3, 3, 4, -1], [0, 1, 4, 6, 6]])
```

```
b=transpose(vector([1, 0, -2]))
```

Tenim que

```
rank(A)
```

3

i per tant el sistema és compatible. Com que l'espai de sortida és de dimensió 5, el nucli és de dimensió 2:

```
A.right_kernel()
```

```
RowSpanZ  $\begin{pmatrix} 1 & 0 & -3 & 2 & 0 \\ 0 & 4 & -1 & 3 & -3 \end{pmatrix}$ 
```

Sage dóna el resultat posant Z , que indica combinacions amb coeficients enters, degut a que tots els elements d' A i b són enters, però en realitat els pensem com nombres reals. En qualsevol cas, una base del nucli és la formada pels vectors

$$\begin{pmatrix} 1 \\ 0 \\ -3 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 4 \\ -1 \\ 3 \\ -3 \end{pmatrix}.$$

Una solució particular del sistema ve donada per

```
xp=A.solve_right(b);xp
```

$$\begin{pmatrix} \frac{13}{2} \\ 2 \\ -\frac{5}{6} \\ 0 \\ 0 \end{pmatrix}$$

Per tant, la solució general del sistema és

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} \frac{13}{2} \\ 2 \\ -\frac{5}{6} \\ 0 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 0 \\ -3 \\ 2 \\ 0 \end{pmatrix} + \mu \begin{pmatrix} 0 \\ 4 \\ -1 \\ 3 \\ -3 \end{pmatrix},$$

amb λ i μ paràmetres arbitraris (per tant, essencialment, amb x_4 i x_5 donant els dos graus de llibertat de la solució).

7.4 Vectors i valors propis

Sigui ara la matriu quadrada

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 3 \end{pmatrix}$$

que introduïm a Sage amb la instrucció

```
A = matrix(QQ, [[1,1,0],[0,2,0],[0,1,3]])
```

Podem calcular els seus valors propis amb el procediment `eigenvalues()`:

```
A.eigenvalues()
```

`[3,2,1]`

Els valors propis són per tant 1, 2 i 3. Els vectors propis corresponents es poden obtenir amb `eigenspaces_right()`:

```
A.eigenspaces_right()
```

`[(3, RowSpan_Q(0 0 1)), (2, RowSpan_Q(1 1 -1)), (1, RowSpan_Q(1 0 0))]`

Això vol dir que els vectors propis són

$$v_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ amb vap } \lambda_1 = 3, \quad v_2 = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} \text{ amb vap } \lambda_2 = 2, \quad v_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \text{ amb vap } \lambda_3 = 1.$$

Per exemple, si fem Av_1 tenim

```
A*vector([0,0,1])
```

```
(0,0,3)
```

que confirma que és un vector propi amb valor propi 3.

Sigui ara la matriu

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ -1 & 0 & 1 & 2 & 2 \\ 1 & 3 & 4 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 3 & 1 & 2 & 3 \end{pmatrix},$$

que introduïm com

```
B=matrix([[1,1,1,1,1],[-1,0,1,2,2],[1,3,4,-1,0],[0,1,0,0,1],[1,3,1,2,3]])
```

Si intentem calcular els valors propis, obtenim

```
B.eigenvalues()
```

```
-2.0195051?, -0.34894014?, 0.80500090?, 3.4736551?, 6.0897892?
```

(hem escrit menys decimals dels que dóna Sage). L'interrogant al final de cada valor propi indica que Sage ha hagut de renunciar a calcular-los exactament (ha de resoldre aquí un polinomi de grau 5, les solucions del qual no es poden obtenir, en general, de forma tancada) i ho ha fet numèricament. De fet, el polinomi característic que ha solucionat es pot obtenir amb charpoly:

```
B.charpoly(x)
```

```
 $x^5 - 8x^4 + 5x^3 + 44x^2 - 20x - 12$ 
```

7.5 Rotacions

La matriu que dóna una rotació d'angle t en sentit antihorari en el pla és

$$R_2 = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}.$$

Podem definir aquesta matriu en Sage, amb angle arbitrari, mitjançant

```
var('t')
```

```
R2=matrix([[cos(t),-sin(t)],[sin(t),cos(t)])]
```

```
R2
```

```
 $\begin{pmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{pmatrix}$ 
```

Podem obtenir matrius de rotacions específiques substituint t per angles concrets:

```
A=R2(t=pi/2);A
```

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

B=R2 (t=2*pi/3);B

$$\begin{pmatrix} -\frac{1}{2} & -\frac{1}{2}\sqrt{3} \\ \frac{1}{2}\sqrt{3} & -\frac{1}{2} \end{pmatrix}$$

La matriu A correspon a una rotació de 90° i la B a una de 120° . Podem comprovar això fent que les matrius actuïn sobre el vector $u_x = (1, 0)$:

A*vector ([1, 0])

$$(0, 1)$$

B*vector ([1, 0])

$$\left(-\frac{1}{2}, \frac{1}{2}\sqrt{3}\right)$$

Els valors propis d'aquestes matrius són complexos, de la forma $\cos t \pm j \sin t$:

A.eigenvalues ()

$$[-i, i]$$

B.eigenvalues ()

$$\left[-\frac{1}{2}i\sqrt{3} - \frac{1}{2}, \frac{1}{2}i\sqrt{3} - \frac{1}{2}\right]$$

Les matrius de rotació a l'espai tenen una forma general més complicada, però sí que és senzill comprovar si una determinada matriu és de rotació, mirant si al multiplicar-la per la seva trasposta dóna la identitat. Sigui per exemple

$$R = \begin{pmatrix} 0 & -1 & 0 \\ -1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} \end{pmatrix}.$$

que introduïm com

q=sqrt (2)

R=matrix ([[0, -1, 0], [-1/q, 0, -1/q], [1/q, 0, -1/q]])

És una matriu de rotació ja que

R*transpose (R)

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Els valors propis de R són

R.eigenvalues ()

$$\left[-\frac{1}{8}\sqrt{16\sqrt{2}-40} - \frac{1}{4}\sqrt{2} - \frac{1}{2}, \frac{1}{8}\sqrt{16\sqrt{2}-40} - \frac{1}{4}\sqrt{2} - \frac{1}{2}, 1\right]$$

Al ser una matriu de rotació, hi ha un valor propi 1, mentre que els altres dos són complexos conjugats ($16\sqrt{2} - 40 < 0$). Podem veure els valors numèrics amb

```
ee=R.eigenvalues()
for i in range(3): ee[i].n()
-0.853553390593274 - 0.521005383279987i
-0.853553390593274 + 0.521005383279987i
1.000000000000000
```

La part real dels valors complexos dóna el cosinus de l'angle de rotació, i la part imaginària el sinus, però això ho calcularem més endavant d'una altra manera.

Per tal de calcular l'eix de rotació, cerquem els vectors propis de valor propi 1. Per fer això hem de construir la matriu que s'obté de R restant-li 1 per la identitat:

```
R1=R-identity_matrix(3)
```

```
R1
```

$$\begin{pmatrix} -1 & -1 & 0 \\ -\frac{1}{2}\sqrt{2} & -1 & -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} & 0 & -\frac{1}{2}\sqrt{2}-1 \end{pmatrix}$$

i hem de calcular el seu nucli:

```
nucli=R1.right_kernel()
```

```
nucli
```

$$\text{RowSpan}_{\text{SR}} \left(1 \quad -1 \quad -\frac{1}{2}(\sqrt{2}-2)\sqrt{2} \right)$$

Això vol dir que l'eix de rotació està en la direcció del vector

$$v = \begin{pmatrix} 1 \\ -1 \\ -\frac{1}{2}(\sqrt{2}-2)\sqrt{2} \end{pmatrix}.$$

Donat que

$$-\frac{1}{2}(\sqrt{2}-2)\sqrt{2} = \sqrt{2}-1,$$

introduïm v a Sage com

```
v=vector([1,-1,sqrt(2)-1])
```

i un vector perpendicular a aquest és, per exemple,

```
u=vector([1,1,0])
```

```
u*v
```

```
0
```

L'acció de la rotació sobre aquest vector és

```
Ru=R*u
```

```
Ru
```

$$\left(-1, -\frac{1}{2}\sqrt{2}, \frac{1}{2}\sqrt{2}\right)$$

i ara calculem el cosinus de l'angle format pel vector original u i el rotat:

```
k= (u*Ru/(u.norm()*Ru.norm())) .simplify_full()
```

```
expand(k)
```

$$-\frac{1}{4}\sqrt{2}-\frac{1}{2}$$

El valor numèric de k és

```
k.n()
```

$$-0.853553390593274$$

que es correspon amb el que havíem vist al calcular els valors propis de la matriu de rotació. Finalment, cal notar que, al ser una rotació, el vector imatge Ru té la mateixa norma que u

```
u.norm();Ru.norm()
```

$$\sqrt{2}$$

$$\sqrt{2}$$

i que el vector rotat continua sent perpendicular a l'eix de rotació:

```
(Ru*v).full_simplify()
```

$$0$$

8 Càlcul

8.1 Funcions, dominis, gràfiques, i límits

Tal com hem vist ja en el cas dels polinomis, la definició de funcions a Sage, i la seva representació gràfica, és molt senzilla. Per exemple, si

$$f(x) = (x+1)e^{-x}$$

tindrem (podem emprar e^{\wedge} o la funció `exp`)

```
f(x)=(x+1)*e^(-x)
```

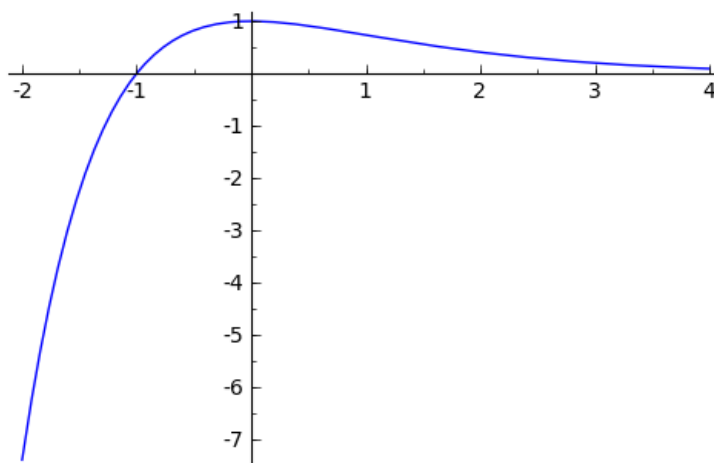
```
f(3)
```

$$4e^{(-3)}$$

```
n(f(3))
```

$$0.199148273471456$$

```
plot(f(x),(x,-2,4))
```



Encara que el domini d'una funció s'ha d'estudiar detingudament de forma manual, la gràfica dóna una idea de si els càlculs s'han fet bé o no. Per exemple, considerem

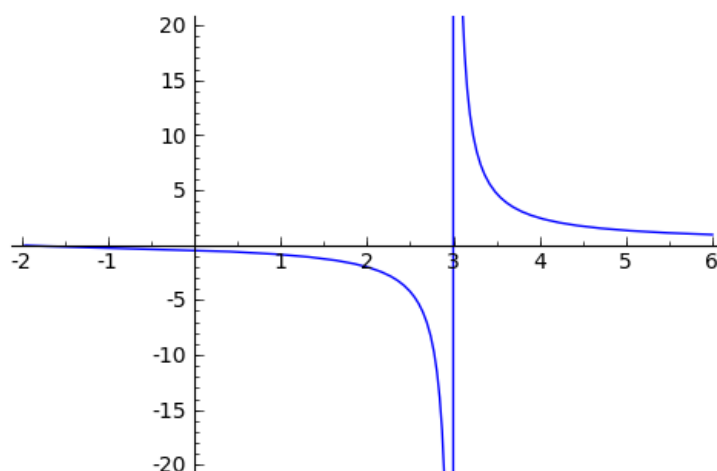
$$f(x) = \frac{\sqrt{x+2}}{x-3}.$$

Aquesta funció té domini

$$\text{Dom}(f(x)) = \{x \in \mathbb{R} \mid x \geq -2, x \neq 3\}.$$

Si dibuixem la gràfica, i posem les opcions de `ymin` i `ymax` per evitar que els valors al voltant de l'asímtota vertical amaguin la resta de valors, obtenim

```
plot(sqrt(x+2)/(x-3), (x, -5, 6), ymin=-20, ymax=20)
```



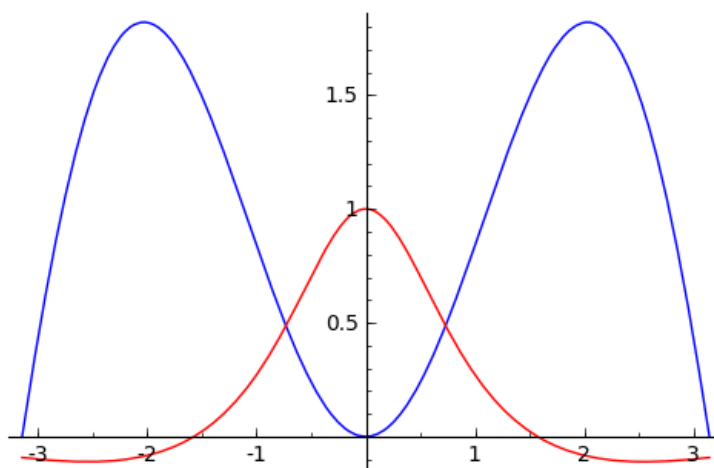
Encara que hem dit que dibuixi des de `-5` fins a `6`, sols ho ha fet de `-2` fins a `6`, posant a més l'asímtota vertical. De fet, juntament amb la gràfica, surt un missatge indicant que Sage no ha pogut avaluar la funció en tota una sèrie de punts, que no pertanyen al domini.

És possible dibuixar diverses gràfiques simultànies amb la funció `show`. La idea és assignar el resultat de `plot` a una variable, de manera que no ho representi, i forçar després la representació amb `show`.

```
p1=plot(x*sin(x), (x, -pi, pi))
```

```
p2=plot(cos(x)/(x^2+1), (x, -pi, pi))
```

```
show(p1+p2)
```



Encara que és possible representar funcions definides a trossos sumant *plots* amb diferents rangs d'abscisses, es pot fer de manera més elegant amb la funció `Piecewise`. Per exemple, si

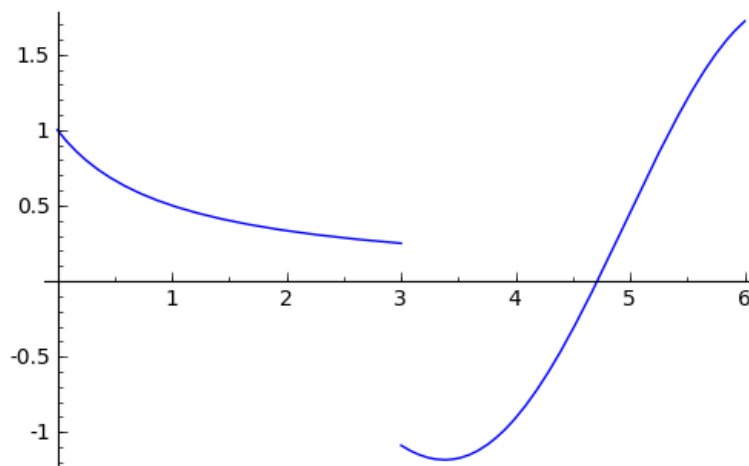
$$f(x) = \begin{cases} \frac{1}{x+1} & \text{si } x \in [0, 3) \\ \log x \cos x & \text{si } x \in (3, 6] \end{cases}$$

s'ha de fer

```
f= Piecewise([[ (0, 3), 1/(x+1) ], [ (3, 6), log(x)*cos(x) ]])
```

(noteu que s'ha de posar `f` i no `f(x)`), i llavors

```
plot(f)
```



El càlcul de límits és senzill i amb força opcions. Per exemple

$$\lim_{x \rightarrow 0} \frac{\sin 2x}{x}$$

es pot calcular amb

$$f(x) = \sin(2 * x) / x$$

$$f.\text{limit}(x=0)$$

2

També és possible utilitzar les formes directes $\text{limit}(f(x), x=a)$ o $\text{lim}(f(x), x=a)$.

Per exemple

$$\text{limit}(x * \sin(x) / (1 - \cos(x)), x=0)$$

2

$$\text{lim}((1 + x - \pi/2)^{\tan(x)}, x = \pi/2)$$

$e^{(-1)}$

$$\text{limit}(x * \log(1/x), x=0)$$

0

A continuació es presenten alguns exemples variats, incloent-hi límits per l'esquerra, per la dreta, a l'infinit i amb resultats variats:

- Límits laterals

$$\lim_{x \rightarrow 3^\pm} e^{\frac{1}{x-3}}$$

$$f(x) = \exp(1 / (x - 3))$$

El límit quan $x \rightarrow 3$ no existeix, degut a que no coincideixen els límits laterals. Sage indica aquesta situació amb el resultat `und`:

```
f.limit(x=3)
```

und

Per l'esquerra

```
f.limit(x=3, dir='minus')
```

0

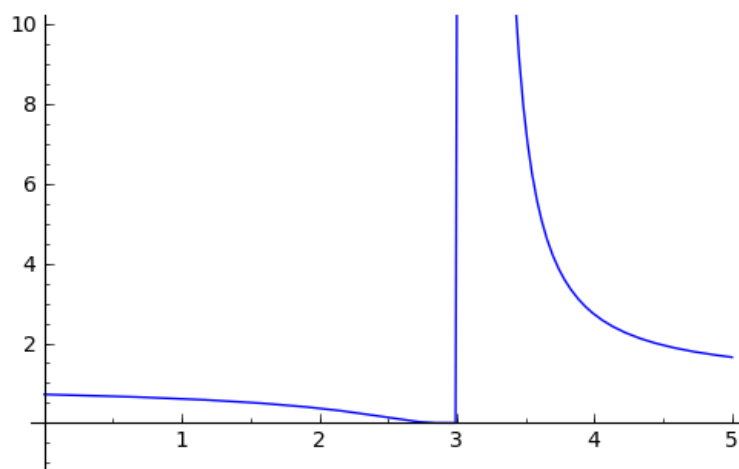
Per la dreta

```
f.limit(x=3, dir='plus')
```

$+\infty$

Gràfica

```
plot(f(x), (x, 0, 5), ymin=-1, ymax=10)
```



- Límits a l'infinit

$$\lim_{x \rightarrow \pm\infty} \arctan x$$

```
f(x) = atan(x)
```

Límit a $+\infty$ (el símbol d'infinit a Sage són dues lletres o)

```
f.limit(x=+oo)
```

$\frac{1}{2}\pi$

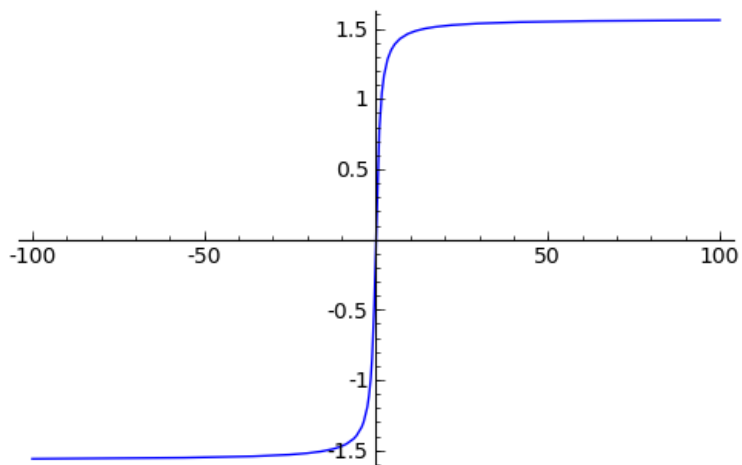
Límit a $-\infty$

```
f.limit(x=-oo)
```

$$-\frac{1}{2}\pi$$

Gràfica

```
plot(f(x), (x, -100, 100))
```



- Límit lateral que no existeix. Sage indica aquesta situació amb `ind`, que és diferent del cas de límits laterals no coincidents (`und`).

$$\lim_{x \rightarrow 0^+} \sin \frac{1}{x}$$

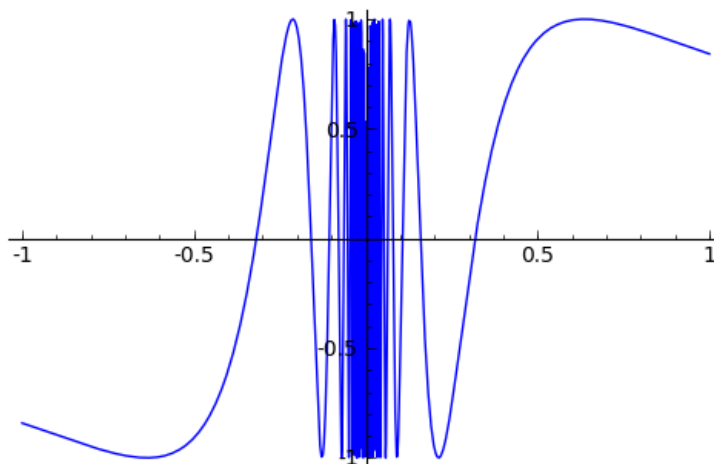
```
f(x)=sin(1/x)
```

```
f.limit(x=0, dir='plus')
```

`ind`

La funció oscil.la arbitràriament a l'acostar-se a $x = 0$ (tant per la dreta com per l'esquerra)

```
plot(f(x), (x, -1, 1))
```



8.2 Derivades i polinomis de Taylor

El càlcul de la funció derivada d'una funció és fa amb la funció `diff`. Si $f(x) = x^3 \sin 3x$, el càlcul de $f'(x)$ es pot fer amb

```
f(x)=x^3*sin(3*x)
```

```
diff(f(x),x)
```

$$3x^3 \cos(3x) + 3x^2 \sin(3x)$$

i el càlcul de la cinquena derivada de $f(x)$, $f^{(5)}(x)$ dóna

```
diff(f(x),x,5)
```

$$243x^3 \cos(3x) + 1215x^2 \sin(3x) - 1620x \cos(3x) - 540 \sin(3x)$$

Poden utilitzar-se també la forma alternativa `derivative(f(x),x)` i la notació de Python `f(x).diff(x)`:

```
(sin(x)*e^x).diff(x)
```

$$e^x \sin(x) + e^x \cos(x)$$

A vegades és interessant tenir en una mateixa gràfica la funció i la seva derivada:

```
f(x)=x^4*sin(x)/(x^4+1)
```

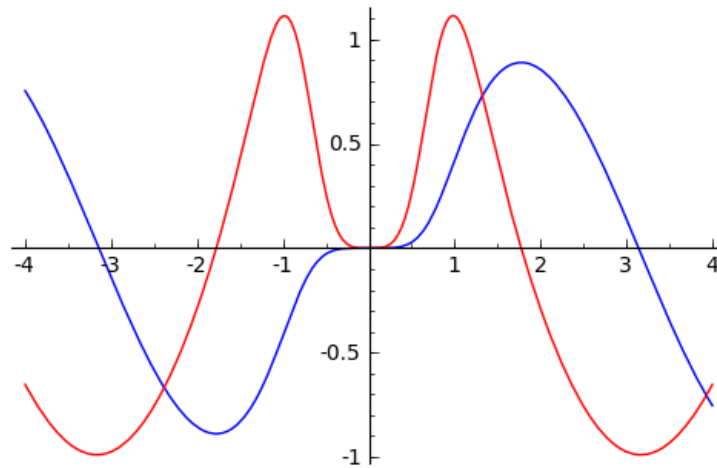
```
g(x)=diff(f(x),x);g(x)
```

$$\frac{-4x^7 \sin(x)}{(x^4+1)^2} + \frac{x^4 \cos(x)}{x^4+1} + \frac{4x^3 \sin(x)}{x^4+1}$$

```
pf=plot(f(x),(x,-4,4),color='blue')
```

```
pg=plot(g(x),(x,-4,4),color='red')
```

```
show(pf+pg)
```



El càlcul de polinomis de Taylor de grau qualsevol és fa amb la funció `taylor`. Si

$$f(x) = \frac{\sin x}{x-1},$$

```
f(x)=sin(x)/(x-1)
```

el polinomi de grau $n = 2$ al voltant de $a = 0$ seria

```
P2(x)=taylor(f(x),x,0,2);P2(x)
```

$$-x^2 - x$$

i el polinomi de grau $n = 6$ al voltant del mateix punt

```
P6(x)=taylor(f(x),x,0,6);P6(x)
```

$$-\frac{101}{120}x^6 - \frac{101}{120}x^5 - \frac{5}{6}x^4 - \frac{5}{6}x^3 - x^2 - x$$

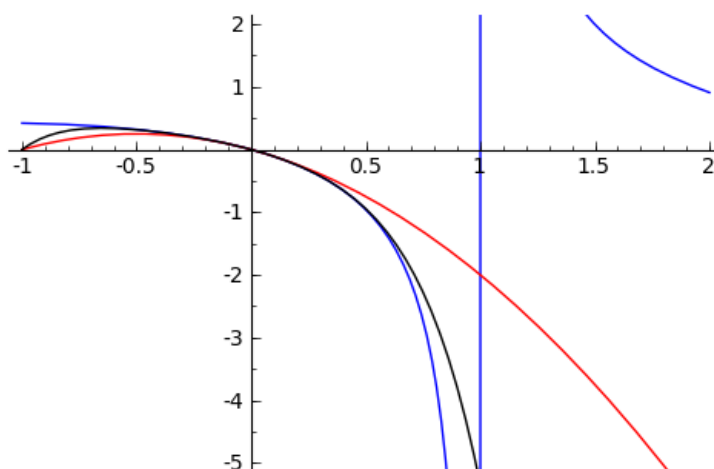
Podem representar els dos polinomis i la funció original en una mateixa gràfica amb

```
p0=plot(f(x),(x,-1,2),ymin=-5,ymax=2,color='blue')
```

```
p2=plot(P2(x),(x,-1,2),ymin=-5,ymax=2,color='blue')
```

```
p6=plot(P6(x),(x,-1,2),ymin=-5,ymax=2,color='blue')
```

```
show(p0+p2+p6)
```



Fixeu-vos que la funció, en blau, té una asymptota vertical en $x = 1$ que els polinomis, per ser polinomis, no tenen. De tota manera, el polinomi de grau 6 (en negre) aproxima la funció millor que el polinomi de grau 2 (en vermell), encara que tots dos divergeixen molt ràpidament de la funció al allunyar-nos de $x = 0$ (i especialment després de passar l'asímtota).

El polinomi de Taylor també es pot calcular al voltant de l'origen o d'altres punts (sempre que la funció hi sigui derivable prou vegades):

```
taylor(cos(x), x, pi, 8)
```

$$\frac{1}{40320}(\pi - x)^8 + \frac{1}{720}(\pi - x)^6 - \frac{1}{24}(\pi - x)^4 + \frac{1}{2}(\pi - x)^2 - 1$$

```
taylor(e^x, x, 0, 6)
```

$$\frac{1}{720}x^6 + \frac{1}{120}x^5 + \frac{1}{24}x^4 + \frac{1}{6}x^3 + \frac{1}{2}x^2 + x + 1$$

```
taylor(1/(1-x), x, 0, 7)
```

$$x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$$

```
taylor(log(1+x), x, 0, 4)
```

$$-\frac{1}{4}x^4 + \frac{1}{3}x^3 - \frac{1}{2}x^2 + x$$

```
taylor(x*sin(x) - x*tan(x) + 1/2*x^4, x, 0, 4)
```

0

El darrer exemple mostra que una funció pot tenir polinomis de Taylor de grau força gran nuls. De fet, el primer polinomi de Taylor al voltant de $x = 0$ no nul d'aquesta funció és el de grau 6, ja que en $x = 0$ s'anul·len la funció i les seves 5 primeres derivades:

```
f(x) = x*sin(x) - x*tan(x) + 1/2*x^4
```

```
for i in range(10): [i, (diff(f(x), x, i)).substitute(x=0)]
```

[0,0]

[1,0]

```
[2,0]
[3,0]
[4,0]
[5,0]
[6,-90]
[7,0]
[8,-2184]
[9,0]
```

Un dels avantatges de Sage és que permet construir aplicacions interactives, anomenades *sagelets*, amb paràmetres variables per l'usuari.

La *sagelet* següent (©Harald Schilly, <http://wiki.sagemath.org/interact/calculus>) permet variar interactivament el grau del polinomi de Taylor, de $n = 1$ fins a $n = 12$, de la funció $f(x) = \sin x e^{-x}$ al voltant de $x_0 = 0$:

```
var('x, x0')
x0 = 0
f = sin(x)*e^(-x)
p = plot(f, -1, 5, thickness=2)
dot = point((x0, f(x0)), pointsize=80, rgbcolor=(1, 0, 0))
@interact
def _(order=(1..12)):
    ft = f.taylor(x, x0, order)
    pt = plot(ft, -1, 5, color='green', thickness=2)
    html('$f(x)\;=\;\%s\;%\latex(f)')
    html('$\hat{f}(x;\%s)\;=\;\%s+\mathcal{O}(x^{\%s})\;%\;(x0, latex(ft), order+1)')
    show(dot + p + pt, ymin = -.5, ymax = 1)
```

Podeu escriure aquest codi en una caixa d'entrada de Sage i experimentar.

8.3 Integrals

El càlcul de primitives es fa amb la funció `integrate`, en la forma normal `integrate(f(x), x)` o en notació de Python `(f(x)).integrate(x)`:

```
integrate(x*sin(x), x)
```

$$-x \cos(x) + \sin(x)$$

```
(sin(2*x)).integrate(x)
```

$$-\frac{1}{2} \cos(2x)$$

Si Sage no sap calcular una primitiva de forma explícita, la retorna tal qual

```
integrate(sqrt(sin(x)), x)
```

$$\int \sqrt{\sin(x)} dx$$

Podem integrar una funció i derivar-la per comprovar el resultat

```
f(x) = x * exp(x) * cos(x) ; f(x)
```

$$x e^x \cos(x)$$

```
g(x) = integrate(f(x), x) ; g(x)
```

$$\frac{1}{2}(x-1)e^x \sin(x) + \frac{1}{2}x e^x \cos(x)$$

```
(f(x) - diff(g(x), x)).simplify_full()
```

$$0$$

Es pot integrar també funcions que depenen de paràmetres, que s'han de definir però prèviament:

```
var('a')
```

$$a$$

```
integrate(a*x^2*cos(a*x), x)
```

$$\frac{2ax \cos(ax) + (a^2 x^2 - 2) \sin(ax)}{a^2}$$

La integral definida

$$\int_a^b f(x) dx$$

es fa amb `integrate(f(x), x, a, b)`:

```
integrate(x*sin(x), x, 0, pi)
```

$$\pi$$

Es poden calcular integrals definides amb extrems dependents de paràmetres però és possible que Sage demani saber el signe del paràmetre:

```
var('a')
```

$$a$$

```
integrate(x*exp(-x), x, 0, a)
```

Traceback (click to the left of this block for traceback)

...

Is a positive, negative, or zero?

El signe d' a es pot especificar amb `assume`:

```
assume(a>0)
```

```
integrate(x*exp(-x), x, 0, a)
```

$$-(a+1)e^{-a} + 1$$