Calcul formel : Maxima

Institut Polytechnique de Saint-Louis (IPSL)
Université Gaston Berger (UGB)

Classes préparatoires intégrées: Deuxième année

Plan du cours

Plan

Motivations

Le calcul formel, encore appelé calcul symbolique, donne les moyens

- *de manipuler les nombres,*
- *d'effectuer des calculs*
- *de réaliser des représentations graphiques sophistiquées,*
- *de mener des calculs algébriques,*
- *de représenter symboliquement des objets mathématiques complexes (racine carrée $\sqrt{x}$, par exemple)*

Exemples de calculs formels

- Division euclidienne
- Calcul matriciel
- Familles libres, génératrices, bases
- Études de fonctions réelles d'une variable réelle

Les Logiciels de Calculs formels

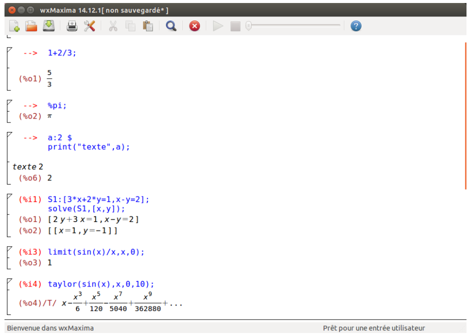
- Logiciels propriétaires :
  - Maple
  - Mathematica
  - ...
- Logiciel libre, gratuit et multiplateform (Windows, Linux, Mac..):
  - **Maxima** : Compilateur
  - **Wxmaxima**: Interface graphique

Pour télécharger le logiciel Maxima, il faut aller sur le site

http://maxima.sourceforge.net/download.html

Plan

Pour le lancer, il faut lancer le fichier exécutable de wxMaxima.



1. Lancer **wxmaxima**
2. Presser la touche Entrée
3. Saisir une instruction, par exemple 2+4 ou exp(1.0)
4. Terminer la ligne de commande par le caractère ; ou $.
5. Presser simultanément les deux touches Shift ⇑ et Entrée

NB: Appuyer juste sur la touche Entrée permet de passer à la ligne sans évaluer votre expression

- **Cell** : Aller au niveau de $\boxed{\text{Cell}}$:

```
Titre : Cours calcul Formel
Section: Motivations ...
```

pour créer un Titre, Section, etc.

- **Police** : Aller au niveau de $\boxed{\text{Edition −> configurer}}$:

```
Police par défaut :  DEjaVu Serif (15)
Police mathématique :  DEjaVu Serif (14)
```

- **Groupement d'instructions**:

```
1+2;  2^(1/2);  2.0^(1/2);  (2*3)^2;
```

- **Commentaires** :

```
1+2; /* un premier calcul tres simple */
/*2.0^(1/2) ; n'est pas prise en compte */
```


- **Variables et affectation**: Maxima utilise le symbole
  - **:** pour affecter une valeur à une variable (Exemple, a:2;)
  - **:=** pour définir une fonction (Ex, $f(x) := 2 * x$;).

1. La fonction valeur absolue (d'un nombre réel) peut être définie comme suit:

```
k(x):= if (x>=0) then x else -x;
k(1);
k(-1);
```

2. Si on a défini une expression, on peut définir une fonction à l'aide de la commande **f(x):=''(expr)** ou **define('f(x),expr)**. Par exemple

```
expr :  2/( x^2 -1);
f(x):=''( expr );  f(5);
define ('g(x),expr );   g(5);
/* Noter la différence entre logcontract et log */
```


- **Noms de variables**: commence par une lettre + caractère, sauf (%, :, ;,=,$), qui jouent un rôle particulier dans maxima.
- **Opérateur dito** Le symbole **%** désigne l'opérateur dito qui permet d'obtenir le dernier résultat calculé par l'interpréteur (lorsqu'il existe) :

```
a :1+2*3;
%;
```

- **Constantes numériques usuelles et fonctions**:
  Une liste des constantes usuellement utilisées dans Maxima :

  → $\%e$: constante de Néper e
  → $\%pi$: nombre π
  → $\%phi$: nombre d'or $\varphi = \frac{1+\sqrt{5}}{2}$
  → $\%i$: unité imaginaire i
  → **inf** infinis réel positif $+\infty$
  → **minf** infinis réel négatif $-\infty$

Une liste des fonctions usuellement utilisées dans Maxima:

→ **sin** sinus, **cos** cosinus, **tan** tangente, **cot** cotangente, **asin** arcsinus, **acos** arccosinus, **atan** arctangente, **acot** arccotangente, **sinh** sinus hyperbolique, **cosh** cosinus hyperbolique, **tanh** tangente hyperbolique, **asinh** arcsinus hyperbolique, **acosh** arccosinus hyperbolique, **atanh** arctangente hyperbolique,

→ **log** logarithme naturel, **log10** logarithme décimale, **exp** exponentiel

→ **sqrt** racine carrée

→ **abs** valeur absolue

→ **float** conversion en floating point

→ **entier** partie entière d'un réel

- **Nettoyage de la mémoire de travail** :

```
a:(x+%beta)*(x-%beta);
b:5-x+%alpha;
a;
b;
kill (all);
a;
b;
```

- **Désaffecter une variable**:

```
a:3;
a;
a:'a;
a;
```


- **Utilisation de l'aide en ligne**: Pour obtenir de l'aide à propos d'une fonction il suffit de sélectionner la fonction et appuyer sur la touche **F1**

- **Parenthèses, crochets, accolades**: Pour entourer une expression de parenthèses ou crochets ou guillemets, sélectionnez-là et tapez «**(**» ou «**[**» ou «{ ».

- **Arrêter l'exécution** : Si un calcul est trop long à évaluer, on peut essayer les commandes

  → «**Maxima** → **Interrompre**» ou

  → «**Maxima** → **RedémarrerMaxima**».

Plan

Calculs symboliques, numériques, arithmétique

- **factor** factorise un nombre ou un polynôme;

```
factor(30!) ;
factor(x^2 + x -6);
factor(sin(x^2 -1));
scanmap(factor,sin(x^2 -1));
```

- **expand** développe une expression ;

```
expand((x+3)^4);
```

- **divide** calcule le quotient et le reste de la division euclidienne ;

```
divide(100 ,7);
divide(3693 ,3);
divide(x^3-1,x -1);
divide(x^4+x^2-3 ,x ^2+1);
```

- **partfrac** décompose une fraction en éléments simples ;

```
partfrac((x ^2+8* x +4) /(x^2 -4) ,x);
```

- **ratsimp** simplifie des expressions rationnelles ;

```
ratsimp((x^2 -1) /(x+1));
```

Par défaut, ratsimp ne simplifie pas les termes avec des racines carrées. Mettre algebraic à true impose ces simplifications ;

```
q1:1/( sqrt (5) -1);
ratsimp (q1);
algebraic : true ;
ratsimp (q1);
```

- **trigsimp** simplifie des expressions trigonométrique ;

```
trigsimp(2*cos(x)^2 + sin(x)^2) ;
```


- **trigexpand** développe des expressions trigonométrique ;

```
trigsimp (2* cos (x)^2 + sin(x)^2) ;
```

Par défaut, Maxima donne la valeur exacte. La commende numer permet d'obtenir des résultat à la virgule près;

```
cos (% pi/4);
cos (% pi/5);
cos (% pi/5), numer;
```

- **trigexpand** développe des expressions trigonométrique ;

```
trigexpand (sin (2*x)+cos (2*x));
```


- **exponentialize** forme trigonométrique en exponentielle ;

```
exponentialize(cos(x));

exponentialize(tanh(x));
```

(-) $\dfrac{e^{i\,x}+e^{-i\,x}}{2}$

(-) $\dfrac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

- **logcontract** développe des expressions logarithmiques ;

```
logcontract (log (x)+ log(x+1));

logcontract (log (x)-log(x+1));

logcontract (10* log(x +1));
```

(-) $\log\left(x\ (x+1)\right)$

(-) $\log\left(\dfrac{x}{x+1}\right)$

(-) $\log\left((x+1)^{10}\right)$

- **solve**: Résolution d'équations et systèmes;

```
solve(x^3=1,x);
solve ([x -2*y=14,x+3*y=9],[x,y]);
```

(-) $[x=\dfrac{\sqrt{3}\,i-1}{2}, x=-\dfrac{\sqrt{3}\,i+1}{2}, x=1]$
(-) $[[x=12, y=-1]]$
Il est préférable d'affecter l'ensemble des solutions et d'accéder à une des valeurs par [n] ;

```
sol : solve (x^2-4,x);
sol [1];
```

(-) $[x=-2, x=2]$
(-) $x=-2$

Autres exemples :

```
trigsimp(solve([cos(x)^2-x=2-sin(x)^2],[x]));
```

(-) $[x=-1]$

- **solve_rec** : calcul le terme général d'une suite récurrente linéaire

```
load("solve_rec")$
solve_rec(u(n)=n*u(n-1)/(1+ n),u(n));
solve_rec(u(n)=n*u(n-1)/(1+ n),u(n),u(1)=2);
solve_rec(u(n)=u(n-1)+u(n -2),u(n));
solve_rec(u(n)=u(n-1)+u(n -2),u(n),u(0)=0,u(1)=1);
```

(-) $\mathrm{u}(n) = \frac{\%k_1}{n+1}$

(-) $\mathrm{u}(n) = \frac{4}{n+1}$

(-) $\mathrm{u}(n) = \frac{\left(\sqrt{5}-1\right)^n \%k_1 (-1)^n}{2^n} + \frac{\left(\sqrt{5}+1\right)^n \%k_2}{2^n}$

(-) $\mathrm{u}(n) = \frac{\left(\sqrt{5}+1\right)^n}{\sqrt{5}\, 2^n} - \frac{\left(\sqrt{5}-1\right)^n (-1)^n}{\sqrt{5}\, 2^n}$

- **Sommes**: **sum**(exp, i, a, b) est la somme $\sum_{i=a}^{b} exp$, **simpsum** calcule cette somme;

```
sum (k, k, 1, n);

%, simpsum ;

sum (1/k^4, k, 1, inf ), simpsum ;
```

(-) $\sum_{k=1}^{n} k$

(-) $\frac{n^2+n}{2}$

(-) $\frac{\pi^4}{90}$

On peut ainsi générer un polynôme à partir des coefficients :

```
c:[5 ,4 ,3 ,2];
sum (c[i +1]* x^i,i ,0,
          length (c) -1);
```

(-) $[5, 4, 3, 2]$

(-) $2\,x^3 + 3\,x^2 + 4\,x + 5$

- **Produits**: **product**(exp, i, a, b) est la somme $\prod_{i=a}^{b} exp$, **simpproduct** calcule ce produit;

```
product(i^3,i,1,10);

product(z(i),i,1,minf);

product(x+i*(i+1)/2,i,1,3);

product (k, k, 1, n), simpproduct;

product (if k <= 5 then a^k else
          b^k, k, 1, 10);
```

(-) $477847258398720000($

(-) $\prod_{i=1}^{\infty} \mathrm{z}(i)$

(-) $(x+1)(x+3)(x+6$

(-) $n!$

(-) $a^{15}\, b^{40}$

- **subst**(valeur,variable,expression): remplace dans expression la variable par valeur ;

```
eq:2*x^2+1/x-(x+1)^3/(1-x)$
subst (3,x,eq); %, numer;
```

(-) $\frac{151}{3}, 50.333333333$

- **ratsubst**(variableNEW,variableOLD,expression): substitue une variable à une autre variable

```
eq:2*x^2+1/x-(x+1)^3/(1-x)$
ratsubst(y,1-x,eq);
```

(-) $\frac{3\,y^4 - 13\,y^3 + 24\,y^2 - 23\,y + 8}{y^2 - y}$

- **Hypothèses**:

  On peut imposer une condition pour permettre à Maxima de simplifier une expression ;

```
abs((x -1)*x*(x+1));
assume(x < -1);
abs((x -1)*x*(x+1));
```

- **Limites: limit**$(f(x), x, x_0)$: pour calculer la limite $\lim_{x \to x_0} f(x)$;

```
f:exp(x)$
'limit(f,x ,0)=limit(f,x ,0);
limit(f,x,inf);
limit(f,x,minf );
h:sin(1/x);
limit(h,x,0);
```

(-) $\lim_{x \to 0} e^x = 1$

(-) ∞

(-) 0

(-) ind

- **Limite d'une fonction**

```
f:exp(x)$
'limit(f,x ,0)=limit(f,x ,0);
limit(f,x,inf);
limit(f,x,minf );
h:sin(1/x);
limit(h,x,0);
```

(-) $\lim_{x \to 0} e^x = 1$

(-) ∞

(-) 0

(-) ind

On peut indiquer aussi la direction : x_0^+ ou x_0^-

```
f:1/x$
limit (f,x ,0,plus );
limit (f,x ,0,minus);
```

(-) ∞

(-) $-\infty$

- **Limite d'une suite**

```
u(n):=(1+2/n)^sqrt(n);
limit (u(n),n,inf );
```


Dérivées: La fonction dérivée

- **Dérivés: diff**$(f(x), x)$: dérivées

```
f:x^2*exp (x);
'diff(f,x)
= diff(f,x);
```

```
f(x):x^2*exp(x);
'diff(f(x),x)
= diff(f(x),x);
```

(-) $x^2\, e^x$

(-) $\frac{d}{d\,x}\left(x^2\, e^x\right) = x^2\, e^x + 2\,x\, e^x$

- **diff**$(f(x), x, n)$: pour les dérivées d'ordre n

```
f(x):=x ^5+1/ x;
diff(f(x),x ,1);
diff(f(x),x ,2);
```

(-) $\mathrm{f}(x) := x^5 + \frac{1}{x}$

(-) $5\,x^4 - \frac{1}{x^2}$

(-) $20\,x^3 + \frac{2}{x^3}$

Formule de Taylor

- **taylor**$(f(x), x, x_0, n)$: formule de Taylor et développements en série

```
taylor ( sqrt (1+x),x ,0 ,4);
```

(-) $/T/\ 1 + \frac{x}{2} - \frac{x^2}{8} + \frac{x^3}{16} - \frac{5\,x^4}{128} + \ldots$

Le symbole principal /T/ indique un développement de Taylor.

La résultante, représentée par l'expression rationnelle, peut être utilisée dans d'autres calculs. Il est indiqué par le symbole /R/.

```
kill(all)$
ratprint:false$
T(x):= ''(taylor(exp(x),x,0,3));
T(3);
T(y);
```

(-) $/T/ \quad T(x) := 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \ldots$

(-) $/R/ \quad 13$

(-) $/R/ \quad \frac{y^3 + 3y^2 + 6y + 6}{6}$

- **integrate**$(f(x), x)$: pour calculer une primitive

```
f:x^2* exp (x);
Inte:integrate(f,x);
expand(diff(Inte ,x));
```

(-) $x^2\, e^x$

(-) $\left(x^2 - 2\,x + 2\right)\, e^x$

(-) $x^2\, e^x$

- **integrate**$(f(x), x, a, b)$: pour calculer une intégrale

```
f:x^2*exp (x)$
'integrate(f,x,-2,0)=integrate(f,x,-2,0);
integrate(1/x, x, 1, inf);
integrate(1/x, x, 1, -2,-1);
```

(-) $\int_{-2}^{0} x^2\, e^x dx = 2 - 10\, e^{-2}$

(-) defint: integral is divergent.

(-) $-\,log(2)$

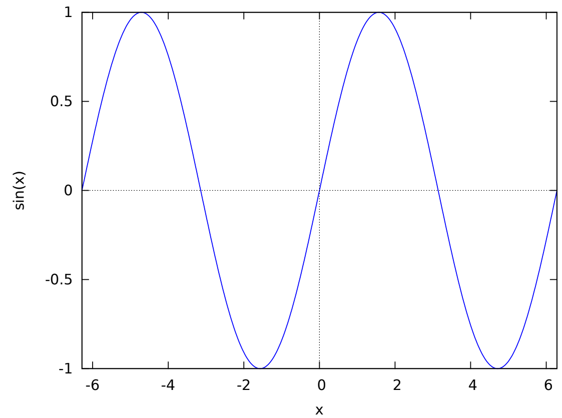
Graphes: draw2d et wxdraw2d

En général, il existe trois types de graphes: les graphes explicites, implicites et paramétriques.

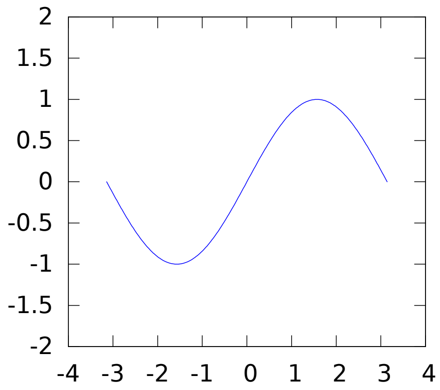
- **Explicite:** La fonction est donnée en termes de valeurs d'entrée. Par exemple, les fonctions $y = f(x) = x^2$ définie dans $\mathbb{R}$, ou $z = f(x, y) = \sin(x, y)$ définie sur un rectangle $[-2, 2] \times [-3, 2]$.
- **Implicite:** La fonction est implicitement donnée en résolvant une équation de la forme $F(x, y) = 0$. Par exemple, $x^2 + y^2 - 1 = 0$ dans le carrée unité $[0, 1] \times [0, 1]$
- **Paramétrique:** La courbe tracée est donnée comme un chemin, c'est-à-dire une carte $(0, 1) \to \mathbb{R}^2$. Par exemple, la parabole $y = x^2$ peut-être paramétré en utilisant un paramètre noté t, et posé $x = t$ et $y = t^2$.

Graphes 2D: Explicite

```
draw2d(explicit(sin(x), x,-2*%pi,2*%pi));
```

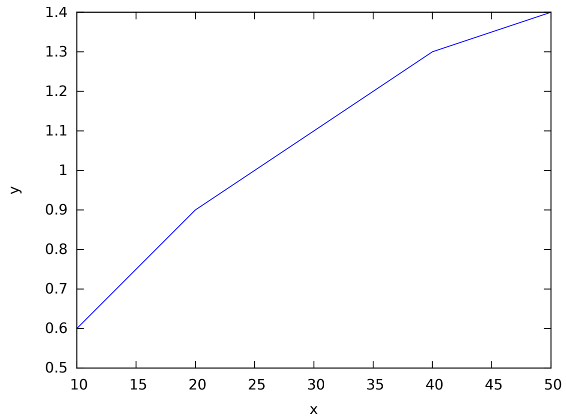


```
draw2d(
explicit(sin(x),x,-%pi,%pi), xrange=[-4,4], yrange=[-2,2],
terminal='png,
file_name="sinus");
```

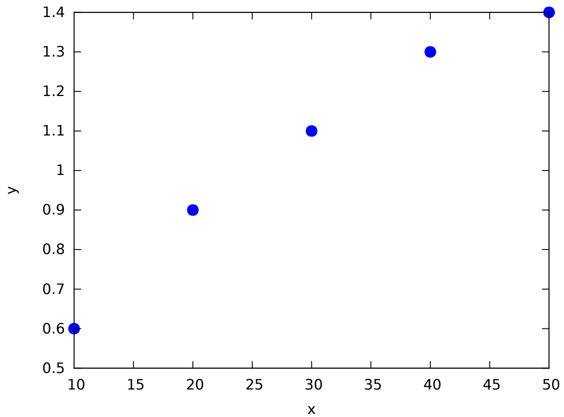


Il faut noter que les courbes peuvent également être exportés en d'autres formats, en remplaçant le *png* par *pdf*, *eps*, *gif*...

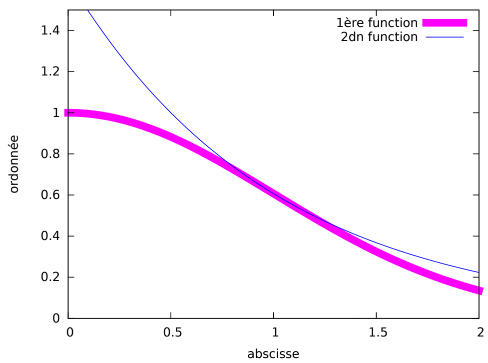

```
draw2d(color=blue, point_type=none, points_joined=true,
points([10,20,30,40,50],[.6,.9,1.1,1.3,1.4]));
```




```
draw2d(color=blue, point_type=filled_circle,
point_size=2, points([[10,.6],[20,.9],[30,1.1],
[40,1.3],[50,1.4]]));
```

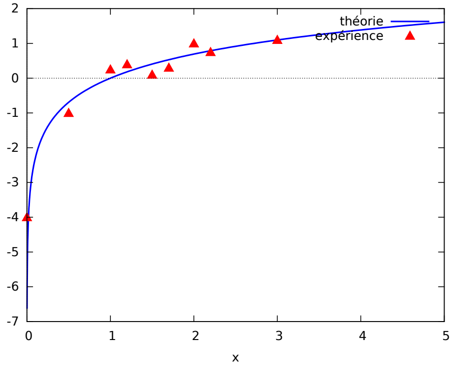



```
draw2d(nticks=100, yrange=[0,1.5],
xlabel="abscisse", ylabel="ordonnée",
color=magenta, line_width=10, key="1ère function",
explicit(exp(-x^2/2), x,0,2),
color=blue, line_width=1,key="2nd function",
explicit(sqrt(%e)*exp(-x), x,0,2));
```



```
datalist: [[0,-4], [0.5,-1], [1,0.25], [1.2,0.4],
[1.5,0.1],[1.7,0.3], [2,1], [2.2,0.75], [3,1.1]]$

draw2d(color=blue, key="théorie", explicit(log(x), x,0,5),
color=red, point_type=filled_up_triangle, point_size=3,
key="experience", points(datalist));
```



- **Suites ou Séquences**: *makelist* **et** *for-do* ;

```
? makelist <1>
```

Exemple 1: Utilisons $makelist$ pour générer la séquence $L = 1; 3; 5;\ ..., 49; 51$, à l'aide de la formule $2n - 1$ avec $n = 1...26$.

```
L:makelist(2*n-1,n,1,26);
```

Le dixième élément de L s'obtient par: `L[10];`

Exemple 2: Utilisons $makelist$ pour générer une série de 20 couple (x, y), ordonnées sur une ligne avec $y = 2x$ pour $x = 0.0; 0.1; 0.2;\ ...1.9; 2$. Représentons ensuite cette liste de couples ordonnées dans $wxdraw2d$.


```
POINTS:makelist([0.1*k,2*(0.1*k)],k,0,20);
```

```
wxdraw2d(
grid=true,
xaxis=true,
yaxis=true,
xrange=[-1,3],
yrange=[-1,5],
point_type=7,
color=red,
points(POINTS)
);
```


Exercice 3 : Soit la fonction

$$y(x) = \frac{x^4}{2} + 5x + c,$$

où c est une constant arbitraire. Tracez la famille de courbes correspondant à plusieurs valeurs de c.

```
declare(c,constant)$
y(x):=x^4/2+5*x+c$

FAMILLE:makelist(explicit(y(x),x,-3,3),c,-20,20)$

wxdraw2d(
grid=true, xrange=[-3,3], yrange=[-5,5],
xaxis=true, yaxis=true, color=red, FAMILLE
);
```

Exercice 4 :

Trajectoire d'une roquette qui suit la fonction
$f(t) = 4 + 1.01 * t - 0.5 * 9.81 * t^2$

```
f(t):=4+1.01*t-0.5*9.81*t^2$
L:makelist(gr2d(
        line_width=2, color=red,
        xrange=[0,2], yrange=[-13.6,4],
        point_type=filled_circle, point_size=2,
        points([[i,f(i)]]),
        line_width=1, line_type=dots, color=black,
        explicit(f(x),x,0,i)), i,0,2,0.01)$
load(draw)$
draw(
    delay     = 1,
    file_name = "vov",
    terminal  = 'animated_gif,
    L)$
```


- **integrate**: Représentation graphique

```
f(t):=t^2*exp(t)$
wxdraw2d(
grid=true,
xrange=[-3,1],
yrange=[-1,1],
fill_color =blue,
filled_func=0,
explicit(f(x),x,-2,0),
filled_func=false,
explicit(f(x),x,-3,3)
);
```


- **integrate**: Représentation graphique

```
f: sqrt (x);
g:x^(3/2) ;
draw2d (
filled_func =f,
fill_color =red ,
explicit (g,x ,0 ,1) ,
filled_func =false ,
explicit (f,x ,0 ,1),
explicit (g,x ,0 ,1));
solve (f=g);
```


Exercices

Calculer l'aire comprise entre le graphe de la fonction $f(x)$ et le graphe de la $g(x)$:

1. $f(x) = -x^2 + x + 2$ et $g(x) = x^2 - 3x + 2$
2. $f(x) = \frac{x^3}{4}$ et $g(x) = x^2 - x$
3. $f(x) = \cosh x$ et $g(x) = 5 \cos 3x$

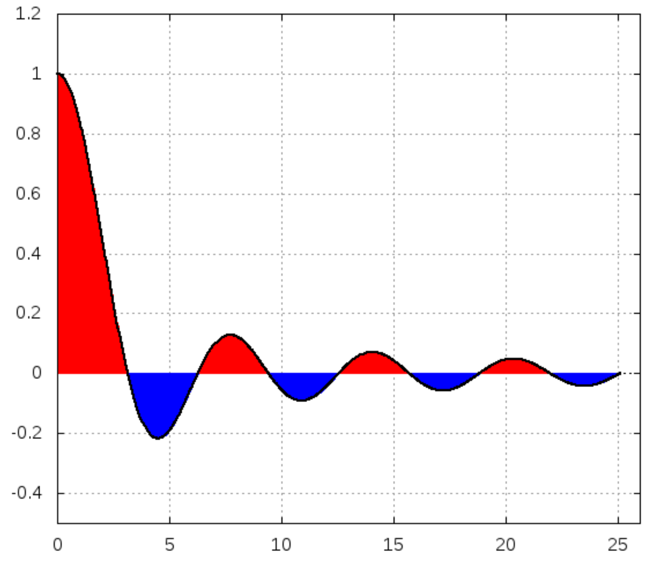
- **Solution**: Exercice 3

```
f(x):=cosh(x)$
g(x):=5*cos(3*x)$

x1:find_root(f(x)-g(x),0,1);
x2:find_root(f(x)-g(x),1.5,2);
x3:find_root(f(x)-g(x),2,2.5);

A1:float(integrate(g(x)-f(x),x,0,x1));
A2:float(integrate(f(x)-g(x),x,x1,x2));
A3:float(integrate(g(x)-f(x),x,x2,x3));
```


Exercice: Représenter ce graphe, intégrale de $\dfrac{sin(x)}{x}$



- **ode2**(eq, y, x) : Équations différentielles ordinaires de 1er ordre

```
eq :'diff(y,x)+(3*x^2+ 1)*y -x^2* exp(-x);
solu :ode2(eq,y,x);
ic1 (solu,x=0,y=1);
```

(-) $\dfrac{d}{d\,x}\,y+\left(3\,x^2+1\right)\,y-x^2\,e^{-x}$

(-) $y=e^{-x^3-x}\left(\dfrac{e^{x^3}}{3}+\%c\right)$

(-) $y=\dfrac{e^{-x^3-x}\,\left(e^{x^3}+2\right)}{3}$

- **ode2**(eq, y, x) : Équations différentielles ordinaires de 2nd ordre

```
edo :'diff (y,x ,2)+2*
'diff(y,x)+y=(1 -x)*exp(-x);
solu:ode2(edo,y,x);
ic2(solu,x=0,y=1,'diff(y,x)=2);
```

(-) $\dfrac{d^2}{d\,x^2}\,y+2\,\left(\dfrac{d}{d\,x}\,y\right)+y=(1-x)\,e^{-x}$

(-) $y = (\%k_2 x + \%k_1)\,e^{-x} - \dfrac{(x^3-3x^2)}{6}e^{-x}$

(-) $y=(3\,x+1)\,e^{-x}-\dfrac{\left(x^3-3\,x^2\right)\,e^{-x}}{6}$

- **desolve** : Équations différentielles ordinaires d'ordre 2

```
kill(all);
eqn: diff(y(t),t,2)= 2*diff(y(t),t,1)-2*y(t);
atvalue(y(t), t=0, 1)$
atvalue(diff(y(t),t), t=0, 3)$
desolve([eqn],[y(t)]);
```


- **desolve** : Systèmes d'ODE linéaires Considérons ce système D'ODE à résoudre,

$$\begin{cases} x'(t) = x(t) - 2y(t) \\ y'(t) = -x(t) + y(t) \end{cases}$$

```
eqn1: diff(x(t),t)=x(t)-2*y(t);
eqn2: diff(y(t),t)=-x(t)+y(t);

atvalue(x(t), t=0, 1)$
atvalue(y(t), t=0, 1)$

desolve([eqn1,eqn2],[x(t),y(t)]);
```


Exercices

Calculer et afficher la solution des problèmes différentiels suivants:

1. $\begin{cases} y'(t) = -ty + 2t \\ y(0) = 3 \end{cases}$
2. $\begin{cases} y'(t) = \cos(t)y + e^{\sin(t)} \\ y(0) = 3 \end{cases}$
3. $\begin{cases} y'(t) = -\dfrac{y}{t} \\ y(1) = 3 \end{cases}$
4. $\begin{cases} y'(t) = y^2 - 1 \\ y(0) = 0 \end{cases}$

Transformée de Laplace

Soit f une fonction d'une variable réel $t > 0$, à valeur réel ou complexe et soit s un paramètre réel ou complexe. La transformée de Laplace de f est une fonction notée F de s est définie comme suit :

$$F(s) = \mathcal{L}(f(t)) \int_0^{+\infty} e^{-st} f(t)\, dt = \lim_{\tau \to +\infty} \int_0^{\tau} e^{-st} f(t)\, dt.$$

Si cette limite existe et est finie, on dit que l'intégrale converge et la Transformée de Laplace est définie. Sinon, elle n'est pas définie.

Exemple 1 : On considère $s > 0$.

1. Pour $f(t) = 1, \quad F(s) = \dfrac{1}{s}$.
2. Pour $f(t) = t, \quad F(s) = \dfrac{1}{s^2}$.
3. Pour $f(t) = t^n, \quad F(s) = \dfrac{n!}{s^{n+1}}$.

Transformée de Laplace

Exemple 2 : On considère $s > 0$.

1. Pour $f(t) = \begin{cases} t & \text{si } 0 \leq t \leq 1 \\ 1 & \text{si } t > 1, \end{cases}$

$$\begin{aligned} F(s) &= \int_0^{+\infty} e^{-st} f(t)\,dt = \int_0^1 e^{-st} f(t)\,dt + \int_1^{+\infty} e^{-st} f(t)\,dt \\ &= \frac{1 - e^{-s}}{s^2}. \end{aligned}$$

2. Pour $f(t) = \begin{cases} 1 & \text{si } t \geq a \\ 0 & \text{si } t < a, \end{cases}$ $\quad F(s) = \dfrac{e^{-as}}{s}.$

Propriétés de la Transformée de Laplace

1. $\mathcal{L}$ est une application linéaire :

$$\mathcal{L}(f(t) + \lambda g(t)) = \mathcal{L}(f(t)) + \lambda \mathcal{L}(g(t)).$$

2. Transformée de Laplace de la dérivée nième $f^{(n)}$:

$$\mathcal{L}(f^{(n)}(t)) = s^n \mathcal{L}(f(t)) - s^{n-1} f(0^+) - s^{n-2} f'(0^+) - \cdots - f^{(n-1)}(0^+).$$

3. Si $\mathcal{L}(f(t)) = F(s)$, alors l'inverse de la transformée de Laplace est noté $\mathcal{L}^{-1}$ est telle que :

$$\mathcal{L}^{-1}(F(s)) = f(t), \quad \forall t > 0.$$

4. $\mathcal{L}^{-1}$ est une application linéaire.

Application : Résolution EDO

Exemple : On considère l'EDO suivante

$$\frac{d^2y}{dt^2} + \omega^2 y = 1, \quad y(0) = a, y'(0) = b,$$

où $\omega > 0$; a et b sont des constantes. En appliquant la transformée de laplace sur la première équation, on obtient :

$$\mathcal{L}\left\{\frac{d^2y}{dt^2}\right\} = s^2 F(s) - sa - b \qquad \text{et} \qquad \mathcal{L}\{1\}\} = 1/s.$$

Ce qui conduit à l'équation $s^2 F(s) - sa - b + \omega^2 F(s) = 1/s$, qui a pour solution :

$$F(s) = \frac{sa + b}{s^2 + \omega^2} + \frac{1}{s(s^2 + \omega^2)}.$$

On en déduit $y(t)$ à l'aide de la transformée de Laplace inverse :

$$y(t) = \left(a - \frac{1}{\omega^2}\right)\cos(\omega t) + \frac{b}{\omega}\sin(\omega t) + \frac{1}{\omega^2}.$$

Transformée de Laplace avec Maxima

```
laplace(1,t,s);
laplace(t,t,s);

assume(n>0)$
laplace(t^n,t,s);
forget(n>0)$

laplace(cos(w*t),t,s);
laplace(sin(w*t),t,s);
```

A l'aide de la fonction laplace de maxima, essayer de calculer le Laplacien de $f(t) = \begin{cases} 1 & \text{si } t \geq a \\ 0 & \text{si } t < a. \end{cases}$

Transformée de Laplace inverse avec Maxima

```
ilt((s+1)/(s^2*(s-1)), s, t);
ilt(1/s^2, s, t);

assume(a>0)$
ilt(a/(s^2+a^2),s,t);
forget(a>0)$

assume(w>0)$
ilt(w/((s-a)^2+w^2),s,t);
forget(w>0)$

ilt(exp(-2*s)/(s^2+1),s,t);
```


Résolution EDO avec Transformée de Laplace

```
kill(all)$
assume(w>0)$
eq:diff(u(t),t,2)+w^2*u(t)=1$
atvalue(u(t), t=0, a)$
atvalue(diff(u(t),t), t=0, b)$
ro:laplace(eq,t,s)$
rt:subst(z,laplace(u(t),t,s),ro)$
rr:solve(rt,z)$
define(y(t),ilt(rhs(rr[1]), s, t));
forget(w>0)$
```


Plan

- **Vecteurs et Listes** : On définit un vecteur (resp. une liste) en donnant ses coordonnées (resp. éléments) entre crochets ;

```
u:[1 ,2 ,3];
v:[-1 ,2 ,3];
u+v;
u-v;
```

(-) $[1, 2, 3]$
(-) $[-1, 2, 3]$
(-) $[0, 4, 6]$
(-) $[2, 0, 0]$

- **makelist**: Pour générer automatiquement le éléments du vecteur ou de la liste.

```
makelist (n^2,n ,3 ,7);
```

(-) $[9, 16, 25, 36, 49]$

- Le produit scalaire de deux vecteurs se note avec un point précédé et suivi d'un espace ;

```
u . v;
```

(-) 12

- **matrix** : définit une matrice et prend **ligne par ligne** en entrées

```
M:matrix([1,2],[3,4]) ;
```

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

Pour extraire une ligne ou une colonne d'une matrice

```
M[2];
col(M,2);
```

(-) $[3,4]$

(-) $\begin{pmatrix} 2 \\ 4 \end{pmatrix}$

- **transpose**: Pour transposer une matrice

```
M:matrix([1,2],[3,4])$
transpose (M);
```

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

- $<< * >>$: pour multiplier des matrices élément par élément.

```
M*M; M^2;
```

(-) $\begin{pmatrix} 1 & 4 \\ 9 & 16 \end{pmatrix}$

- $<< \cdot >>$: pour multiplier des matrices.

```
M^^2; M.M;
```

(-) $\begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$

- **charpoly, eigenvalues et eigenvectors**: Pour obtenir le polynôme caractéristique, les valeurs propres et vecteurs propres d'une matrice

```
M:matrix([1,2,0],[0,3,0],[2,-4,2])$
charpoly(M,x);
solve(charpoly(M,x)=0,x);
eigenvalues(M);
eigenvectors(M);
```

- **part**: Pour extraire un vecteur

```
M:matrix([1,2,0],[0,3,0],
[2,-4,2])$
part(M,3);
[vp,vcp]:eigenvectors(M)$
vcp;
vp1:part(vcp,3);
```

(-) $[2, -4, 2]$

(-) $[[[1, 0, -2]], [[0, 0, 1]], [[1, 1, -$

(-) $[1, 1, -2]$

- **rank, determinant, invert [ou ^^(-1)]** : Pour trouver, respectivement, le rang, le déterminant et l'inverse.

```
rank(M);

determinant(M);

invert(M);

M^^(-1);
```

(-) 2

(-) -2

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

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

- **genmatrix** (suite utilisée, nombre de lignes, nombre de colonnes) : Pour générer automatiquement une matrice

```
f[i,j]:= i^2+j ^2;

genmatrix(f,2,4);
```

(-) $f_{i,j} := i^2 + j^2$

(-) $\begin{pmatrix} 2 & 5 & 10 & 17 \\ 5 & 8 & 13 & 20 \end{pmatrix}$

- **makelist**: Pour générer automatiquement le éléments d'une matrice

```
a(i,j):=i^2+j^2;
A:apply(
          'matrix,
makelist(makelist(a(i,j),i,1,4),
        j,1,2));
```