XCAS au service des mathématiques dans l'enseignement secondaire

Séminaire IREM/APMEP - Bordeaux

Guillaume CONNAN

IREM de Nantes

Jeudi 24 janvier 2013

Sommaire

Sommaire

Sommaire

- Bernard Parisse
- Giac C++
- XCAS
- Renée De Graeve

- Bernard Parisse
- Giac C++

- Bernard Parisse
- Giac C++
- XCAS
- Renee De Graeve

- Bernard Parisse
- Giac C++
- XCAS
- Renée De Graeve

- Bernard Parisse
- Giac C++
- XCAS
- Renée De Graeve

- Bernard Parisse
- Giac C++
- XCAS
- Renée De Graeve

- Bernard Parisse
- Giac C++
- XCAS
- Renée De Graeve

Sommaire

- calcul formel ;
- géométrie (dynamique) dans le plan et l'espace ;
- programmation

- calcul formel ;
- géométrie (dynamique) dans le plan et l'espace ;
- programmation français/anglais

- calcul formel ;
- géométrie (dynamique) dans le plan et l'espace ;
- programmation français/anglais impérative un peu fonctionnelle ;

- calcul formel ;
- géométrie (dynamique) dans le plan et l'espace ;
- programmation français/anglais impérative un peu fonctionnelle ;
- tableur dynamique ;

- calcul formel ;
- géométrie (dynamique) dans le plan et l'espace ;
- programmation français/anglais impérative un peu fonctionnelle ;
- tableur dynamique ;

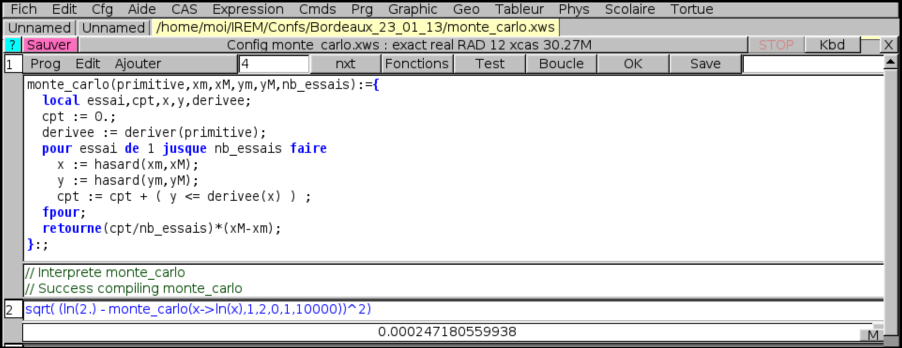
- calcul formel ;
- géométrie (dynamique) dans le plan et l'espace ;
- programmation français/anglais impérative un peu fonctionnelle ;
- tableur dynamique ;
- combinaisons de tout ça...

- calcul formel ;
- géométrie (dynamique) dans le plan et l'espace ;
- programmation français/anglais impérative un peu fonctionnelle ;
- tableur dynamique ;
- combinaisons de tout ça...

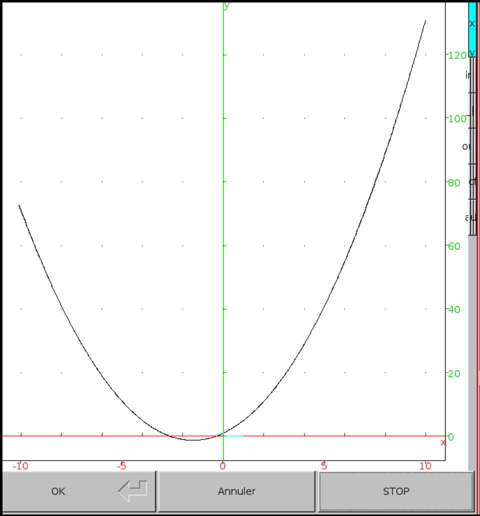
- calcul formel ;
- géométrie (dynamique) dans le plan et l'espace ;
- programmation français/anglais impérative un peu fonctionnelle ;
- tableur dynamique ;
- combinaisons de tout ça...

Sommaire

Interface XCAS



Terminal




```
Fichier Éditer Affichage Terminal Aller Aide
moi@info-po-ens6254:~$ giac
// Using locale /usr/share/locale/
// fr_FR.UTF-8
// /usr/share/locale/
// giac
// UTF-8
// Maximum number of parallel threads 4
// Using keyword file /usr/share/giac/doc/fr/keywords
Help file /usr/share/giac/doc/fr/aide_cas not found
Added 165 synonyms
Welcome to giac readline interface
(c) 2001,2011 B. Parisse & others
Homepage http://www-fourier.ujf-grenoble.fr/~parisse/giac.html
Released under the GPL license 3.0 or above
See http://www.gnu.org for license details
May contain BSD licensed software parts (lapack, atlas, tinymt)
-------------------------------------------------
Press CTRL and D simultaneously to finish session
Type ?commandname for help
0>> f := x -> x^2 + 3*x + 1
// Success
// End defining f
 (x)->x^2+3*x+1
// Time 0
1>> resoudre(f(x)=0 , x)
list[1/2*(-3-sqrt(5)),1/2*(-3+sqrt(5))]
// Time 0.01
2>> graphe(f(x))
```

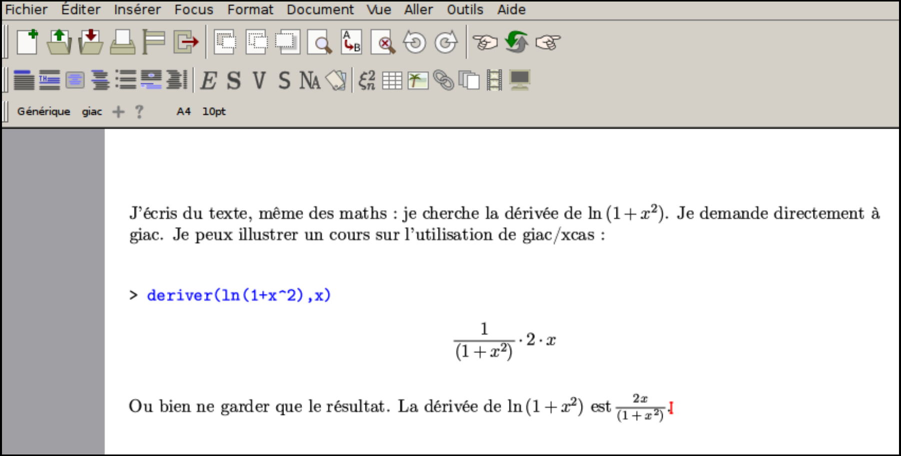
Interface Python

```
Python 2.7.3 (default, Sep 26 2012, 21:51:14)
[GCC 4.7.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from giacpy import *
>>> x = giac('x')
>>> (pi/6).cos()
sqrt(3)/2
>>> f = x.cos()
>>> f.int(x,0,2)
sin(2)
>>> f.diff()
-sin(x)
>>> (f.diff()).diff()
-cos(x)
```

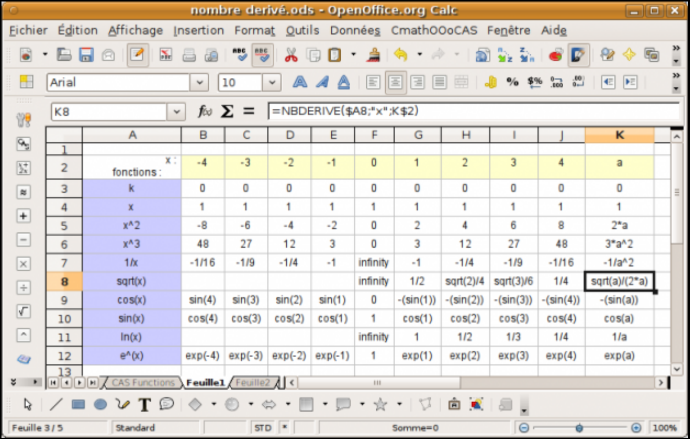
Interface Sage

```
----------------------------------------------------------------------
| Sage Version 5.4.1, Release Date: 2012-11-15                       |
| Type "notebook()" for the browser-based notebook interface.        |
| Type "help()" for help.                                            |
----------------------------------------------------------------------
sage: solve(abs(11+2*I) - abs(x+1+5*I),x)
[abs(x + 5*I + 1) == 5*sqrt(5)]
sage: giac.resoudre(abs(11+2*I) - abs(x+1+5*I),x)
list[-11,9]
sage: %giac

  --> Switching to Giac <--

,,

giac: resoudre(abs(11+2*I) - abs(x+1+5*I),x)
list[-11,9]
```

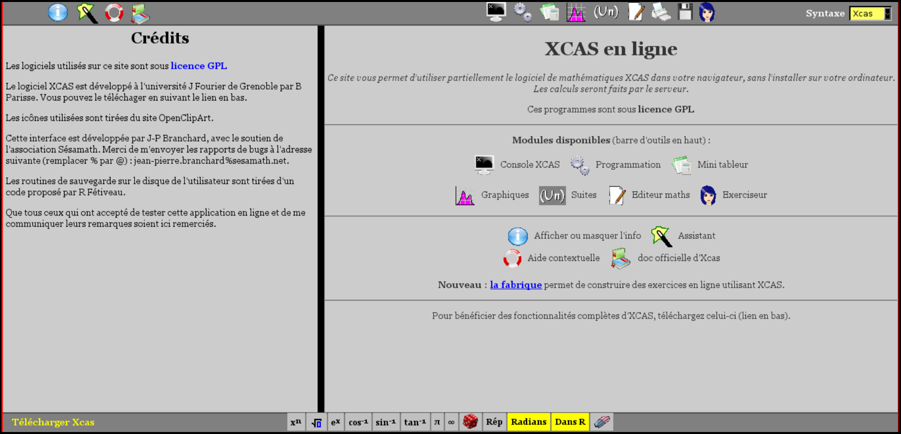
Interface TEXmacs



Interface Open Office



XCAS en ligne



etc...

libgiac

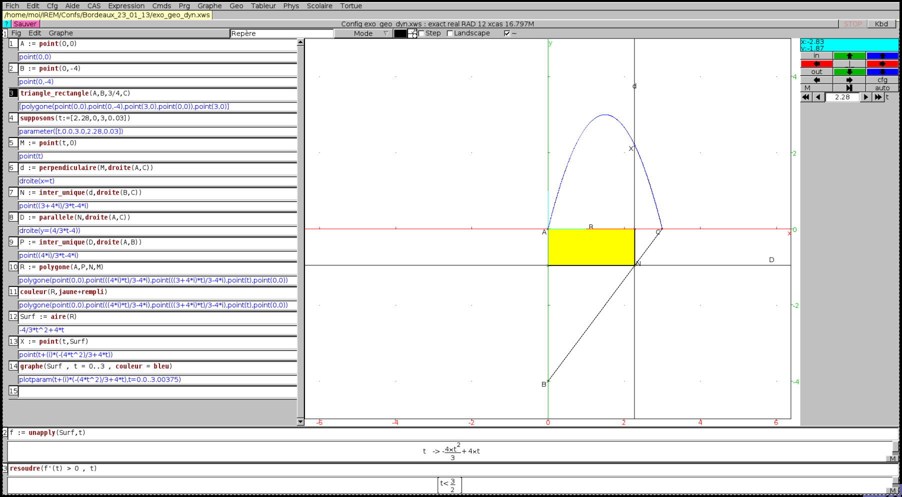
Sommaire

Un exercice de seconde

Exercice 1

On considère un triangle ABC rectangle en A tel que $AC = 3$ et $AB = 4$. Soit M un point quelconque du segment $[AC]$. On construit le rectangle $AMNP$ tel que N appartienne à $[BC]$ et P à $[AB]$. Étudiez l'aire du rectangle $AMNP$ en fonction de la position de M.

Un exercice de seconde



Un exercice de seconde

Commençons par ouvrir une fenêtre de géométrie en tapant simultanément sur Alt + G puis définissons les points A et B dans un repère judicieusement choisi. On utilise `point(x,y)` (ou même `point(z)` pour les terminales) :

```
A := point(0,0) /* plaçons A */
B := point(0,-4) /* plaçons B tel que AB = 4 en laissant le 1er quadrant
    libre pour y tracer la courbe à la fin de la séance */
```

Définissons ensuite le point C tel que le triangle ABC soit direct, rectangle en A et que l'on ait $AC = \frac{3}{4}AB$ à l'aide de la commande `triangle_rectangle` :

```
triangle_rectangle(A,B,3/4,C)
```

Un exercice de seconde

Commençons par ouvrir une fenêtre de géométrie en tapant simultanément sur Alt + G puis définissons les points A et B dans un repère judicieusement choisi. On utilise `point(x,y)` (ou même `point(z)` pour les terminales) :

```
A := point(0,0) /* plaçons A */
B := point(0,-4) /* plaçons B tel que AB = 4 en laissant le 1er quadrant
    libre pour y tracer la courbe à la fin de la séance */
```

Définissons ensuite le point C tel que le triangle ABC soit direct, rectangle en A et que l'on ait $AC = \frac{3}{4}AB$ à l'aide de la commande `triangle_rectangle` :

```
triangle_rectangle(A,B,3/4,C)
```

Un exercice de seconde

Créons maintenant un réel quelconque de $[0;3]$ que l'on puisse faire varier à la souris à l'aide de la commande
`supposons (t := [valeur de départ,mini,maxi])` :

```
supposons(t := [1,0,3])
M := point(t,0)
```

Pour définir N, commençons par définir la perpendiculaire en M à la droite (AC). La syntaxe est tout à fait naturelle grâce à `perpendiculaire(Point,Droite)`. C'est bien ce que l'élève doit tracer sur sa feuille de papier pour tracer le rectangle :

```
d := perpendiculaire(M,droite(A,C))
N := inter_unique(d,droite(B,C))
```

Un exercice de seconde

Créons maintenant un réel quelconque de [0; 3] que l'on puisse faire varier à la souris à l'aide de la commande `supposons (t := [valeur de départ,mini,maxi])` :

```
supposons(t := [1,0,3])
M := point(t,0)
```

Pour définir N, commençons par définir la perpendiculaire en M à la droite (AC). La syntaxe est tout à fait naturelle grâce à `perpendiculaire(Point,Droite)`. C'est bien ce que l'élève doit tracer sur sa feuille de papier pour tracer le rectangle :

```
d := perpendiculaire(M,droite(A,C))
N := inter_unique(d,droite(B,C))
```

Un exercice de seconde

```
D := parallele(N,droite(A,C))
P := inter_unique(D,droite(A,B))
R := polygone(A,P,N,M)
couleur(R,jaune+rempli)
```

```
Surf := aire(R)
                        -4/3*t^2+4*t
X:=point(t,Surf)
graphe(Surf,t=0..3,couleur=bleu)
```

Un exercice de seconde

```
D := parallele(N,droite(A,C))
P := inter_unique(D,droite(A,B))
R := polygone(A,P,N,M)
couleur(R,jaune+rempli)
```

```
Surf := aire(R)
                        -4/3*t^2+4*t
X:=point(t,Surf)
graphe(Surf,t=0..3,couleur=bleu)
```

Un exercice de seconde

```
longueur(A,P),longueur(A,M)
                       -(t*4)/3+4,t
evalf(longueur(A,P) , longueur(A,M))
                       2.0,1.5
f := unapply(Surf,t)
                       (t)->-4*t^2/3+4*t
resoudre(f'(t) > 0 , t)
                       list[t<(3/2)]
```

Sommaire

Les coniques ne sont plus au programme de terminale mais sont encore sources d'inspiration pour des activités. Nous allons ici nous occuper d'un théorème concernant le tracé de la tangente à une conique et le prouver à l'aide de XCAS ce qui n'est possible qu'avec un logiciel de calcul formel.

Théorème 1 (Construction de la tangente à une conique)

Soit Γ une conique de foyer F et de directrice associée D. La tangente à Γ en tout point M qui n'appartient pas à l'axe focal coupe D en un point T tel que l'on ait $\overrightarrow{FM} \cdot \overrightarrow{FT} = 0$

Théorème 1 (Construction de la tangente à une conique)

Soit Γ une conique de foyer F et de directrice associée D. La tangente à Γ en tout point M qui n'appartient pas à l'axe focal coupe D en un point T tel que l'on ait $\overrightarrow{FM} \cdot \overrightarrow{FT} = 0$

Exercice 2 (Énoncé adapté au théorème précédent)

Soient a et b deux réels strictement positifs tels que $a > b$ et f la fonction définie sur $[-a, a]$ par

$$f(x) = b\sqrt{1 - \frac{x^2}{a^2}}$$

Notons Γ la courbe représentative de f dans un repère orthonormé. Soit F le point de coordonnées $(\sqrt{a^2 - b^2}\,;0)$ et D la droite d'équation $x = \frac{a^2}{\sqrt{a^2 - b^2}}$.
Montrez que la tangente à Γ en tout point M d'ordonnée non nulle coupe D en un point T tel que l'on ait $\overrightarrow{FM} \cdot \overrightarrow{FT} = 0$

Théorème 1 (Construction de la tangente à une conique)

Soit Γ une conique de foyer F et de directrice associée D. La tangente à Γ en tout point M qui n'appartient pas à l'axe focal coupe D en un point T tel que l'on ait $\overrightarrow{FM} \cdot \overrightarrow{FT} = 0$

Exercice 2 (Énoncé adapté au théorème précédent)

Soient a et b deux réels strictement positifs tels que $a > b$ et f la fonction définie sur $[-a, a]$ par

$$f(x) = b\sqrt{1 - \frac{x^2}{a^2}}$$

Notons Γ la courbe représentative de f dans un repère orthonormé. Soit F le point de coordonnées $(\sqrt{a^2 - b^2}\,; 0)$ et D la droite d'équation $x = \frac{a^2}{\sqrt{a^2-b^2}}$.
Montrez que la tangente à Γ en tout point M d'ordonnée non nulle coupe D en un point T tel que l'on ait $\overrightarrow{FM} \cdot \overrightarrow{FT} = 0$

```
f := x -> b*sqrt(1 - (x^2/a^2))
```

```
supposons(a > 0)
supposons(t > -a) et supposons(t < a)
M := point(t,f(t))
```

```
d := droite(y = f'(abscisse(M))*(x - abscisse(M)) + ordonnee(M))
F := point(sqrt(a^2-b^2),0)
D := droite(x = a^2/sqrt(a^2-b^2))
T := inter_unique(d,D)
```

```
p := produit_scalaire(vecteur(M,F),vecteur(T,F))
```

```
f := x -> b*sqrt(1 - (x^2/a^2))
```

```
supposons(a > 0)
supposons(t > -a) et supposons(t < a)
M := point(t,f(t))
```

```
d := droite(y = f'(abscisse(M))*(x - abscisse(M)) + ordonnee(M))
F := point(sqrt(a^2-b^2),0)
D := droite(x = a^2/sqrt(a^2-b^2))
T := inter_unique(d,D)
```

```
p := produit_scalaire(vecteur(M,F),vecteur(T,F))
```

```
f := x -> b*sqrt(1 - (x^2/a^2))
```

```
supposons(a > 0)
supposons(t > -a) et supposons(t < a)
M := point(t,f(t))
```

```
d := droite(y = f'(abscisse(M))*(x - abscisse(M)) + ordonnee(M))
F := point(sqrt(a^2-b^2),0)
D := droite(x = a^2/sqrt(a^2-b^2))
T := inter_unique(d,D)
```

```
p := produit_scalaire(vecteur(M,F),vecteur(T,F))
```

```
f := x -> b*sqrt(1 - (x^2/a^2))
```

```
supposons(a > 0)
supposons(t > -a) et supposons(t < a)
M := point(t,f(t))
```

```
d := droite(y = f'(abscisse(M))*(x - abscisse(M)) + ordonnee(M))
F := point(sqrt(a^2-b^2),0)
D := droite(x = a^2/sqrt(a^2-b^2))
T := inter_unique(d,D)
```

```
p := produit_scalaire(vecteur(M,F),vecteur(T,F))
```

```
afficher(p)
```

$$(-t+\sqrt{a^2-b^2})(\sqrt{a^2-b^2}-\frac{-a^2\sqrt{a^2-t^2}\sqrt{-b^2+a^2}}{(\sqrt{a^2-t^2}b^2+-a^2\sqrt{a^2-t^2})})-\sqrt{1-\frac{t^2}{a^2}}b-\frac{1}{(\sqrt{a^2-t^2}b^2-a^2\sqrt{a^2-t^2})}(b^3a+b-a^3+\sqrt{-b^2+a^2}tba)$$

```
simplifier(p)
```

0

```
est_orthogonal(droite(M,F),droite(T,F))
```

1

```
afficher(p)
```

$$(-t+\sqrt{a^2-b^2})(\sqrt{a^2-b^2}-\frac{-a^2\sqrt{a^2-t^2}\sqrt{-b^2+a^2}}{(\sqrt{a^2-t^2}b^2+-a^2\sqrt{a^2-t^2})})-\sqrt{1-\frac{t^2}{a^2}}b-\frac{1}{(\sqrt{a^2-t^2}b^2-a^2\sqrt{a^2-t^2})}(b^3a+b-a^3+\sqrt{-b^2+a^2}tba)$$

```
simplifier(p)
```

0

```
est_orthogonal(droite(M,F),droite(T,F))
```

1

```
afficher(p)
```

$$(-t+\sqrt{a^2-b^2})(\sqrt{a^2-b^2}-\frac{-a^2\sqrt{a^2-t^2}\sqrt{-b^2+a^2}}{(\sqrt{a^2-t^2}b^2+-a^2\sqrt{a^2-t^2})})-\sqrt{1-\frac{t^2}{a^2}}b-\frac{1}{(\sqrt{a^2-t^2}b^2-a^2\sqrt{a^2-t^2})}(b^3a+b-a^3+\sqrt{-b^2+a^2}tba)$$

```
simplifier(p)
```

$$0$$

```
est_orthogonal(droite(M,F),droite(T,F))
```

$$1$$

```
afficher(p)
```

$$(-t+\sqrt{a^2-b^2})(\sqrt{a^2-b^2}-\frac{-a^2\sqrt{a^2-t^2}\sqrt{-b^2+a^2}}{(\sqrt{a^2-t^2}b^2+-a^2\sqrt{a^2-t^2})})-\sqrt{1-\frac{t^2}{a^2}}b-\frac{1}{(\sqrt{a^2-t^2}b^2-a^2\sqrt{a^2-t^2})}(b^3a+b-a^3+\sqrt{-b^2+a^2}tba)$$

```
simplifier(p)
```

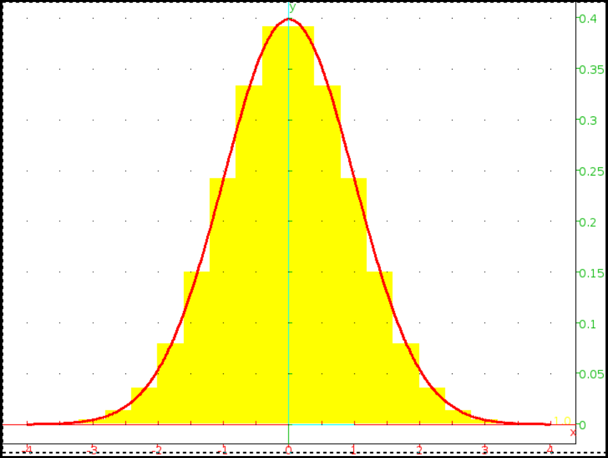
0

```
est_orthogonal(droite(M,F),droite(T,F))
```

1

Sommaire

```
supposons(xm := [40,-100,100])
supposons(N := [20,0,100])
tracer_aire(loi_normale(0,1,x),x = -xm..xm,N,point_milieu,couleur=jaune)
```



```
approx_mil(f,a,b,N):={
  local h,S,k;
  h := evalf((b-a)/N);
  S := 0;
  pour k de 1 jusque N faire
    S := S + h * f(a + (k-0.5)*h)
  fpour;
  retourne S
}:;
```

```
approx_mil(x -> loi_normale(0,1,x), -xm,xm,iPart(N))
                      0.985616238639
```

```
approx_mil(f,a,b,N):={
  local h,S,k;
  h := evalf((b-a)/N);
  S := 0;
  pour k de 1 jusque N faire
    S := S + h * f(a + (k-0.5)*h)
  fpour;
  retourne S
}:;
```

```
approx_mil(x -> loi_normale(0,1,x), -xm,xm,iPart(N))
                      0.985616238639
```

Sommaire

```
monte_carlo(primitive,xm,xM,ym,yM,nb_essais):={
  local essai,cpt,x,y,derivee;
  cpt := 0.;
  derivee := deriver(primitive);
  pour essai de 1 jusque nb_essais faire
    x := alea(xm,xM);
    y := alea(ym,yM);
    cpt := cpt + ( y <= derivee(x) ) ;
  fpour;
  retourne(cpt/nb_essais)*(xM-xm) }
```

```
monte_carlo_sigma(primitive,xm,xM,ym,yM,nb_essais):={
  local k,liste;
  liste:=[seq(primitive(xM)-primitive(xm)-monte_carlo(primitive,xm,xM,ym,
       yM,nb_essais) , k = 1..1000)];
  retourne ecart_type(liste) }
```

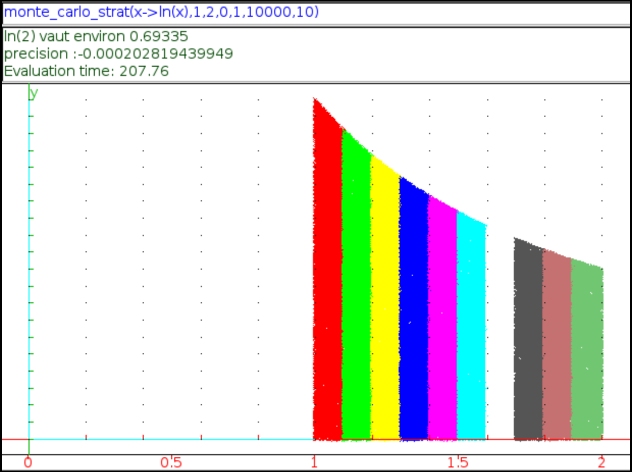
```
monte_carlo_sigma(x->ln(x),1,2,0,1,10000)
   Evaluation time: 112.6
        0.00433957839425
```

```
monte_carlo_sigma(primitive,xm,xM,ym,yM,nb_essais):={
  local k,liste;
  liste:=[seq(primitive(xM)-primitive(xm)-monte_carlo(primitive,xm,xM,ym,
      yM,nb_essais) , k = 1..1000)];
  retourne ecart_type(liste) }
```

```
monte_carlo_sigma(x->ln(x),1,2,0,1,10000)
   Evaluation time: 112.6
        0.00433957839425
```

```
monte_carlo_strat(primitive,xm,xM,ym,yM,nb_essais,echant):={
  h := evalf((xM-xm)/echant);
  P := NULL; app := 0;  xh := xm;
  derivee := deriver(primitive);
  pour tranche de 1 jusque echant faire
    cpt := 0;
    pour essai de 1 jusque nb_essais faire
      x := alea(xh,xh + h);
      y := alea(ym,yM);
      si  y <= derivee(x)  alors
        P := P, point([x,y],couleur=tranche);
        cpt := cpt +1;
      fsi;
    fpour;
    app := app + (cpt/nb_essais)*h;
    xh := xh + h;
  fpour;
  afficher(primitive(xM)-primitive(xm) + " vaut environ " + app);
  afficher("precision :" + (primitive(xM)-primitive(xm) - app));
  retourne(P) }
```

```
monte_carlo_strat(x->ln(x),1,2,0,1,10000,10)
```



Sommaire

Sommaire

On lance trois fois de suite une pièce de monnaie. On veut compter combien de fois on obtient pile (ou face).

```
frequences([(alea(2) + alea(2) + alea(2)) $ (k=1..1000000)])
```

En 5 secondes, un million de simulations donnent :

```
[[0,0.125177],[1,0.374565],[2,0.375099],[3,0.125159]]
```

En effet, on peut vérifier avec `binomial(n,k,p)` qui renvoie $\binom{n}{k}p^k(1-p)^{n-k}$:

```
seq(binomial(3,k,0.5),k=0..3)
```

renvoie `(0.125,0.375,0.375,0.125)`

On lance trois fois de suite une pièce de monnaie. On veut compter combien de fois on obtient pile (ou face).

```
frequences([(alea(2) + alea(2) + alea(2)) $ (k=1..1000000)])
```

En 5 secondes, un million de simulations donnent :

```
[[0,0.125177],[1,0.374565],[2,0.375099],[3,0.125159]]
```

En effet, on peut vérifier avec `binomial(n,k,p)` qui renvoie $\binom{n}{k}p^k(1-p)^{n-k}$:

```
seq(binomial(3,k,0.5),k=0..3)
```

renvoie `(0.125,0.375,0.375,0.125)`

On lance trois fois de suite une pièce de monnaie. On veut compter combien de fois on obtient pile (ou face).

```
frequences([(alea(2) + alea(2) + alea(2)) $ (k=1..1000000)])
```

En 5 secondes, un million de simulations donnent :

```
[[0,0.125177],[1,0.374565],[2,0.375099],[3,0.125159]]
```

En effet, on peut vérifier avec `binomial(n,k,p)` qui renvoie $\binom{n}{k}p^k(1-p)^{n-k}$:

```
seq(binomial(3,k,0.5),k=0..3)
```

renvoie (0.125,0.375,0.375,0.125)

On lance trois fois de suite une pièce de monnaie. On veut compter combien de fois on obtient pile (ou face).

```
frequences([(alea(2) + alea(2) + alea(2)) $ (k=1..1000000)])
```

En 5 secondes, un million de simulations donnent :

```
[[0,0.125177],[1,0.374565],[2,0.375099],[3,0.125159]]
```

En effet, on peut vérifier avec `binomial(n,k,p)` qui renvoie $\binom{n}{k}p^k(1-p)^{n-k}$:

```
seq(binomial(3,k,0.5),k=0..3)
```

renvoie `(0.125,0.375,0.375,0.125)`

On lance trois fois de suite une pièce de monnaie. On veut compter combien de fois on obtient pile (ou face).

```
frequences([(alea(2) + alea(2) + alea(2)) $ (k=1..1000000)])
```

En 5 secondes, un million de simulations donnent :

```
[[0,0.125177],[1,0.374565],[2,0.375099],[3,0.125159]]
```

En effet, on peut vérifier avec `binomial(n,k,p)` qui renvoie $\binom{n}{k}p^k(1-p)^{n-k}$:

```
seq(binomial(3,k,0.5),k=0..3)
```

renvoie `(0.125,0.375,0.375,0.125)`

Sommaire

C'est présenté un peu antimathématiquement dans les programmes mais bon...
On peut illustrer la convergence en loi de la loi binomiale $\mathcal{B}(n,p)$ vers le loi normale $\mathcal{N}(np, \sqrt{np(1-p)})$.

On modélise tout d'abord la loi binomiale à partir d'une loi uniforme :

```
binom(n,p):={
  somme([(alea(0,1) <= p) $ (k = 1..n)])
}:;
```

On compare ensuite graphiquement :

```
convergence(n,p):={
  m := n*p;
  s := sqrt(n*p*(1-p));
  histogram([binom(n,p) $ (k=1..20000)],0,s*0.3), graphe(loi_normale(m,s,x
      ),x = m - 3*s..m + 3*s,couleur=bleu)
}:;
```

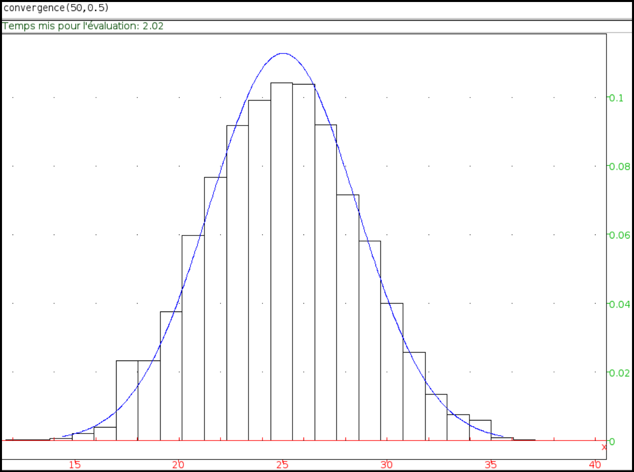
On modélise tout d'abord la loi binomiale à partir d'une loi uniforme :

```
binom(n,p):={
  somme([(alea(0,1) <= p) $ (k = 1..n)])
}:;
```

On compare ensuite graphiquement :

```
convergence(n,p):={
  m := n*p;
  s := sqrt(n*p*(1-p));
  histogram([binom(n,p) $ (k=1..20000)],0,s*0.3), graphe(loi_normale(m,s,x
      ),x = m - 3*s..m + 3*s,couleur=bleu)
}:;
```

Pour $\mathcal{B}(n,p)$ on obtient :



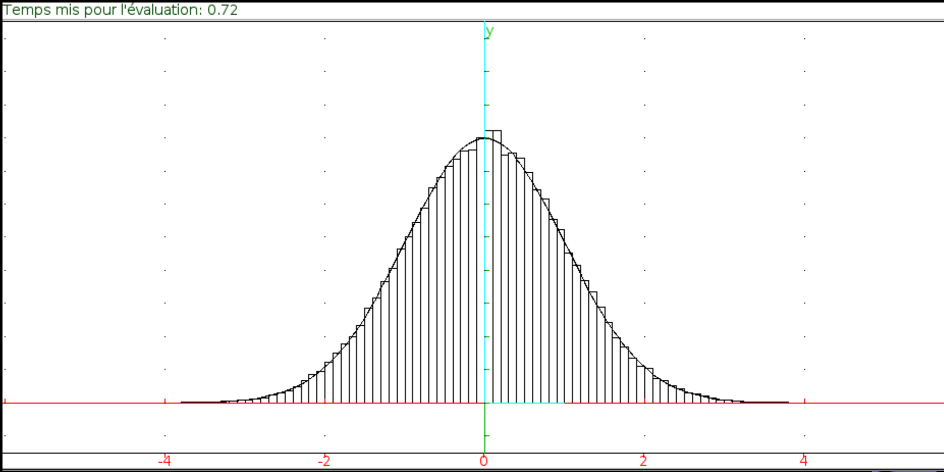
Sommaire

C'est aussi un peu malvenu d'en parler en terminale, mais bon...
On part de deux va indépendantes. Alors

$$\frac{1}{\sqrt{2\pi}^2} e^{-\frac{x^2+y^2}{2}}\,\mathrm{d}x\,\mathrm{d}y = \frac{1}{2\pi} e^{-\frac{r^2}{2}}\, r\,\mathrm{d}r\,\mathrm{d}\theta = \left(\frac{1}{2} e^{-\frac{s}{2}}\mathrm{d}s\right)\left(\frac{1}{2\pi}\mathrm{d}\theta\right)$$

Alors S suit une loi exponentielle de paramètre 1/2 et Θ une moi uniforme sur $[0,2\pi]$.
Alors S peut être générée par $-\frac{\ln U_1}{1/2}$ et Θ par $2\pi U_2$ si U_1 et U_2 sont deux var indépendantes suivant une loi uniforme sur $[0,1]$.
On projette : $X = R\cos(\Theta) = \sqrt{-2\ln U_1}\cos(2\pi U_2)$.

```
histogram([(sqrt(-2*ln(alea(0,1)))*cos(2*pi*alea(0,1))) $ (k=1..100000)
    ],0,0.1),graphe(loi_normale(0,1,x))
```



```
moyenne([(sqrt(-2*ln(alea(0,1)))*cos(2*pi*alea(0,1))) $ (k=1..100000)])
                  -0.00124374528487
```

```
ecart_type([(sqrt(-2*ln(alea(0,1)))*cos(2*pi*alea(0,1))) $ (k=1..100000)])
                  1.00039929494
```

```
moyenne([(sqrt(-2*ln(alea(0,1)))*cos(2*pi*alea(0,1))) $ (k=1..100000)])
                   -0.00124374528487
```

```
ecart_type([(sqrt(-2*ln(alea(0,1)))*cos(2*pi*alea(0,1))) $ (k=1..100000)])
                   1.00039929494
```

Sommaire

Voici un tableau qui donne l'extension de la banquise au minimum de septembre de 1979 à 2012 (NSIDC *National Snow and Ice Data Center*) :

An	79	80	81	82	83	84	85	86	87	88	89	90
km^2	7.2	7.85	7.25	7.45	7.52	7.17	6.93	7.54	7.48	7.49	7.04	6.24
An	91	92	93	94	95	96	97	98	99	00	01	02
km^2	6.55	7.55	6.5	7.18	6.13	7.88	6.74	6.56	6.24	6.32	6.75	5.96
An	03	04	05	06	07	08	09	10	11	12		
km^2	6.15	6.05	5.57	5.92	4.3	4.68	5.63	4.9	4.34	3.6		

Définition 2 (moyenne mobile)

On appelle moyenne mobile d'ordre k la moyenne arithmétique d'un terme avec les k termes voisins.

- si k est impair, c'est-à-dire s'écrit sous la forme $k = 2p+1$ alors on remplace chaque terme par la moyenne de ce terme, des p termes suivants et des p termes précédents ;
- Si k est pair, c'est-à-dire s'écrit sous la forme $k = 2p$, c'est moins évident : on remplace chaque terme par la moyenne de ce terme, des p termes suivants et des p termes précédents mais les termes extrêmes sont affectés d'un coefficient $1/2$.

Définition 3 (Lissage par moyennes mobiles)

Lisser une série chronologique c'est remplacer la série initiale par la série des moyennes mobiles.

Définition 2 (moyenne mobile)

On appelle moyenne mobile d'ordre k la moyenne arithmétique d'un terme avec les k termes voisins.

- si k est impair, c'est-à-dire s'écrit sous la forme $k = 2p + 1$ alors on remplace chaque terme par la moyenne de ce terme, des p termes suivants et des p termes précédents ;
- Si k est pair, c'est-à-dire s'écrit sous la forme $k = 2p$, c'est moins évident : on remplace chaque terme par la moyenne de ce terme, des p termes suivants et des p termes précédents mais les termes extrêmes sont affectés d'un coefficient $1/2$.

Définition 3 (Lissage par moyennes mobiles)

Lisser une série chronologique c'est remplacer la série initiale par la série des moyennes mobiles.

Définition 2 (moyenne mobile)

On appelle moyenne mobile d'ordre k la moyenne arithmétique d'un terme avec les k termes voisins.

- si k est impair, c'est-à-dire s'écrit sous la forme $k = 2p + 1$ alors on remplace chaque terme par la moyenne de ce terme, des p termes suivants et des p termes précédents ;
- Si k est pair, c'est-à-dire s'écrit sous la forme $k = 2p$, c'est moins évident : on remplace chaque terme par la moyenne de ce terme, des p termes suivants et des p termes précédents mais les termes extrêmes sont affectés d'un coefficient $1/2$.

Définition 3 (Lissage par moyennes mobiles)

Lisser une série chronologique c'est remplacer la série initiale par la série des moyennes mobiles.

Définition 2 (moyenne mobile)

On appelle moyenne mobile d'ordre k la moyenne arithmétique d'un terme avec les k termes voisins.

- si k est impair, c'est-à-dire s'écrit sous la forme $k = 2p + 1$ alors on remplace chaque terme par la moyenne de ce terme, des p termes suivants et des p termes précédents ;
- Si k est pair, c'est-à-dire s'écrit sous la forme $k = 2p$, c'est moins évident : on remplace chaque terme par la moyenne de ce terme, des p termes suivants et des p termes précédents mais les termes extrêmes sont affectés d'un coefficient $1/2$.

Définition 3 (Lissage par moyennes mobiles)

Lisser une série chronologique c'est remplacer la série initiale par la série des moyennes mobiles.

Par exemple, reprenons les données du tableau et effectuons un lissage par moyennes mobiles d'ordre 5. Il y a 30 termes au départ. L'ordre 5 étant impair, on va remplacer chaque terme par la moyenne de ce terme, des 2 précédents et des deux suivants : on est donc obligé de commencer par le troisième terme et de terminer par le vingt-huitième.
On remplacera alors t_3 par

$$t'_3 = \frac{t_1 + t_2 + t_3 + t_4 + t_5}{5} = \frac{5,3 + 5,5 + 4,95 + 5,13 + 5,37}{5} = 5,25$$

```
lmm(L,k):={
  X  := L[0];  Y := L[1];
  Xm := NULL; Ym := NULL;
  imax := size(X)-1;
  si k mod 2 == 0 alors
    p := k/2;
    pour j de p jusque imax-p faire
      Ym := Ym , (0.5*Y[j-p] + somme(Y[l],l,j-p+1,j+p-1) + 0.5*Y[j+p])/k;
      Xm := Xm , X[j];
    fpour;
  sinon
    p := (k-1)/2;
    pour j de p jusque imax-p faire
      Ym := Ym , (somme(evalf(Y[l]),l,j-p,j+p))/k;
      Xm := Xm , X[j];
    fpour;
  fsi;
  C :=  couleur(ligne_polygonale(X,Y),rouge):;
  Cm := couleur(ligne_polygonale([Xm],[Ym]),bleu):;
  retourne C,Cm }:;
```

8
7.5
7
6.5
6
5.5
5
4.5
4
3.5
1980
1985
1990
1995
2000
2005
2010

Sommaire

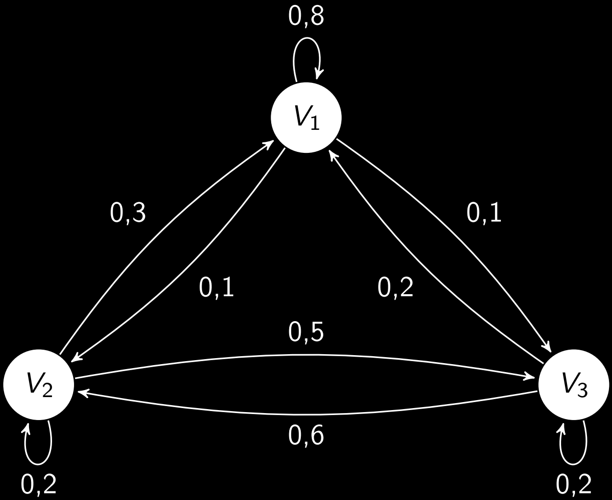

ABBA
WWW.UNIVERSALSTEREO.COM.MX

Zlot, Brzxxz et Morzgniouf sont trois villes situées respectivement en Syldavie, Bordurie et Bouzoukstan. Des trafiquants de photos dédicacées du groupe ABBA prennent leur marchandise le matin dans n'importe laquelle de ces villes pour l'apporter le soir dans n'importe quelle autre. On notera pour simplifier V_1, V_2 et V_3 ces villes et p_{ij} la probabilité qu'une marchandise prise le matin dans la ville V_i soit rendue le soir dans la ville V_j. La matrice $P = (p_{ij})_{\substack{1 \leqslant i \leqslant 3 \\ 1 \leqslant j \leqslant 3}}$ est appelée *matrice de transition* de la chaîne de Markov.

Définition 4 (Matrice stochastique)

Une matrice dont tous les coefficients sont positifs ou nuls et dont la somme des coefficients de chaque ligne vaut 1 est appelée **matrice stochastique**. Si, de plus, la somme des colonnes vaut 1, la matrice est dite **bistochastique**.

Voici le diagramme de transition des trafiquants suédophiles :



La matrice de transition est :

$$\begin{pmatrix} 0,8 & 0,1 & 0,1 \\ 0,3 & 0,2 & 0,5 \\ 0,2 & 0,2 & 0,6 \end{pmatrix}$$

On notera $x_i^{(k)}$ la proportion de trafiquants qui se trouvent au matin du jour k dans la ville V_i.

Définition 5 (Vecteur d'état)

On appelle *vecteur d'état* tout élément $(x_1, \dots, x_n)$ de $\mathbb{R}^n$ tel que $x_1 + \cdots + x_n = 1$.

Ainsi, $x^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}\right)$ est un vecteur d'état.

On démontre que

$$\forall k \in [\![1, n]\!], \quad x^{(k)} = x^{(k-1)} \cdot P$$

puis que

$$\forall k \in [\![1, n]\!], \quad x^{(k)} = x^{(0)} \cdot P^k$$

On notera $x_i^{(k)}$ la proportion de trafiquants qui se trouvent au matin du jour k dans la ville V_i.

Définition 5 (Vecteur d'état)

On appelle *vecteur d'état* tout élément $(x_1,...,x_n)$ de $\mathbb{R}^n$ tel que $x_1+\cdots+x_n=1$.

Ainsi, $x^{(k)}=\left(x_1^{(k)},x_2^{(k)},x_3^{(k)}\right)$ est un vecteur d'état.

On démontre que

$$\forall k \in [\![1,n]\!],\quad x^{(k)}=x^{(k-1)}\cdot P$$

puis que

$$\forall k \in [\![1,n]\!],\quad x^{(k)}=x^{(0)}\cdot P^k$$

On notera $x_i^{(k)}$ la proportion de trafiquants qui se trouvent au matin du jour k dans la ville V_i.

Définition 5 (Vecteur d'état)

On appelle *vecteur d'état* tout élément $(x_1, ..., x_n)$ de $\mathbb{R}^n$ tel que $x_1 + \cdots + x_n = 1$.

Ainsi, $x^{(k)} = \left(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}\right)$ est un vecteur d'état.
On démontre que

$$\forall k \in [\![1, n]\!], \quad x^{(k)} = x^{(k-1)} \cdot P$$

puis que

$$\forall k \in [\![1, n]\!], \quad x^{(k)} = x^{(0)} \cdot P^k$$

Supposons que le chef de la mafia locale dispose de 1000 trafiquants qui partent tous le matin du jour 0 de la ville de Zlot. Quelle sera la proportion de trafiquants dans chacune des villes au bout d'une semaine ? d'un an ?

Il s'agit donc de calculer des puissances successives de P.

On obtient les proportions au bout d'une semaine en calculant $x^{(0)} \cdot p^7$ avec $x^{(0)} = (1000, 0, 0)$.

Nous allons utiliser XCAS pour les longs calculs.

Supposons que le chef de la mafia locale dispose de 1000 trafiquants qui partent tous le matin du jour 0 de la ville de Zlot. Quelle sera la proportion de trafiquants dans chacune des villes au bout d'une semaine ? d'un an ?
Il s'agit donc de calculer des puissances successives de P.
On obtient les proportions au bout d'une semaine en calculant $x^{(0)} \cdot p^7$ avec $x^{(0)} = (1000, 0, 0)$.
Nous allons utiliser XCAS pour les longs calculs.

Supposons que le chef de la mafia locale dispose de 1000 trafiquants qui partent tous le matin du jour 0 de la ville de Zlot. Quelle sera la proportion de trafiquants dans chacune des villes au bout d'une semaine ? d'un an ?
Il s'agit donc de calculer des puissances successives de P.
On obtient les proportions au bout d'une semaine en calculant $x^{(0)} \cdot p^7$ avec $x^{(0)} = (1000, 0, 0)$.
Nous allons utiliser XCAS pour les longs calculs.

Supposons que le chef de la mafia locale dispose de 1000 trafiquants qui partent tous le matin du jour 0 de la ville de Zlot. Quelle sera la proportion de trafiquants dans chacune des villes au bout d'une semaine ? d'un an ?
Il s'agit donc de calculer des puissances successives de P.
On obtient les proportions au bout d'une semaine en calculant $x^{(0)} \cdot p^7$ avec $x^{(0)} = (1000, 0, 0)$.
Nous allons utiliser XCAS pour les longs calculs.

```
A := [[4/5,1/10,1/10],
      [3/10,1/5,1/2],
      [1/5,1/5,3/5]]

X := [1000,0,0]

X * A
                 [800,100,100]
evalf(X * A^7)
                [546.9038,144.5711,308.5251]
evalf(X * A^365)
                [536.585365854,146.341463415,317.073170732]
evalf(A^10000)
                [[0.536585365854,0.146341463415,0.317073170732],
                 [0.536585365854,0.146341463415,0.317073170732],
                 [0.536585365854,0.146341463415,0.317073170732]]
v := evalf(A^10000)[0]
                [0.536585365854,0.146341463415,0.317073170732]
v * A
                [0.536585365854,0.146341463415,0.317073170732]
```

Troublant....En fait, ce résultat se généralise.

Définition 6 (Matrice de transition régulière)

Une matrice de transition A est dite régulière si, et seulement si, il existe un entier naturel r tel que A^r a ses coefficients tous **strictement** positifs.

Théorème 7 (Théorème des chaînes de Markov régulières)

Soit A une matrice régulière. Il existe une matrice stochastique :

$$A_\infty = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ a_1 & a_2 & \cdots & a_n \\ \vdots & \vdots & \vdots & \vdots \\ a_1 & a_2 & \cdots & a_n \end{pmatrix}$$

telles que $\lim_{n \to +\infty} A^n = A_\infty$.

De plus, $v = (a_1, a_2, \cdots, a_n)$ *est l'unique vecteur d'état stationnaire (tel que* $v \cdot A = v$*).*

Troublant....En fait, ce résultat se généralise.

Définition 6 (Matrice de transition régulière)

Une matrice de transition A est dite régulière si, et seulement si, il existe un entier naturel r tel que A^r a ses coefficients tous **strictement** positifs.

Théorème 7 (Théorème des chaînes de Markov régulières)

Soit A une matrice régulière. Il existe une matrice stochastique :

$$A_\infty = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ a_1 & a_2 & \cdots & a_n \\ \vdots & \vdots & \vdots & \vdots \\ a_1 & a_2 & \cdots & a_n \end{pmatrix}$$

telles que $\lim\limits_{n\to+\infty} A^n = A_\infty$.

De plus, $v = (a_1, a_2, \cdots, a_n)$ *est l'unique vecteur d'état stationnaire (tel que* $v \cdot A = v$*).*

Troublant....En fait, ce résultat se généralise.

Définition 6 (Matrice de transition régulière)

Une matrice de transition A est dite régulière si, et seulement si, il existe un entier naturel r tel que A^r a ses coefficients tous **strictement** positifs.

Théorème 7 (Théorème des chaînes de Markov régulières)

Soit A une matrice régulière. Il existe une matrice stochastique :

$$A_\infty = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ a_1 & a_2 & \cdots & a_n \\ \vdots & \vdots & \vdots & \vdots \\ a_1 & a_2 & \cdots & a_n \end{pmatrix}$$

telles que $\lim\limits_{n\to+\infty} A^n = A_\infty$.
De plus, $v = (a_1, a_2, \cdots, a_n)$ *est l'unique vecteur d'état stationnaire (tel que* $v \cdot A = v$*).*

Sommaire

Intégrons l'équation différentielle $u'(t) = f(t, u(t))$ sur l'intervalle I_n :

$$u(t_{n+1}) - u(t_n) = \int_{t_n}^{t_{n+1}} f(s, u(s))\,\mathrm{d}s$$

On utilise alors des méthodes d'intégration numériques classiques comme la méthode des rectangles ou la méthode du point médian. On choisit un pas constant de h.

- $u(t_{n+1}) - u(t_n) = h \times f(t_n, u(t_n))$;
- $u(t_{n+1}) - u(t_n) = h \times f(t_n + h/2, u(t_n + h/2))$. Pour calculer $u(t_n + h/2)$, on utilise une approximation du premier ordre : $u(t_n + h/2) - u(t_n) = (h/2) \times f(t_n, u(t_n))$.

Intégrons l'équation différentielle $u'(t) = f(t, u(t))$ sur l'intervalle I_n :

$$u(t_{n+1}) - u(t_n) = \int_{t_n}^{t_{n+1}} f(s, u(s))\,\mathrm{d}s$$

On utilise alors des méthodes d'intégration numériques classiques comme la méthode des rectangles ou la méthode du point médian. On choisit un pas constant de h.

- $u(t_{n+1}) - u(t_n) = h \times f(t_n, u(t_n))$;
- $u(t_{n+1}) - u(t_n) = h \times f(t_n + h/2, u(t_n + h/2))$. Pour calculer $u(t_n + h/2)$, on utilise une approximation du premier ordre :
  $u(t_n + h/2) - u(t_n) = (h/2) \times f(t_n, u(t_n))$.

```
Euler1(F,N,a,b,yo):={
local S,X,Y,h,P,f,k;
S := NULL;
X := a;
Y := yo;
h := evalf((b-a)/N);
pour k de 1 jusque N faire
   Y := h*F(X,Y) + Y;
   X := X + h;
   P := point(X,Y);
   S := S,P;
fpour;
couleur(polygone_ouvert(S),bleu);
}:;
```

```
Euler2(F,N,a,b,yo):={
local S,X,Y,h,P,f,k;
S := NULL;
X := a;
Y := yo;
h := evalf((b-a)/N);
pour k de 1 jusque N faire
   Y := h*F(X + 0.5*h,Y + 0.5*h*F(X,Y)) + Y;
   X := X + h;
   P := point(X,Y);
   S := S,P;
fpour;
couleur(polygone_ouvert(S),vert);
}:;
```

Considérons le problème :

$$\begin{cases} u'(t) = 5u(t) - 5, \quad t \in [0,50] \\ u(0) = 1 \end{cases}$$

et imaginons une petite perturbation de la valeur initiale : $u(0) = 1 + \varepsilon$. On peut faire estimer $\widetilde{u}(50) - u(50)$ pour $\varepsilon = 10^{-12}$.