Chap. V : Équations

Laurent Poinsot

17 janvier 2009

Plan

Plan

Plan

Pour résoudre **formellement** une équation, on utilise la commande `solve` :

```
> solve(2*ln(x/3)=-1,x);
```

$$3e^{(-1/2)}$$

Pour résoudre **formellement** une équation, on utilise la commande `solve` :

```
> solve(2*ln(x/3)=-1,x);
```

$$3e^{(-1/2)}$$

Pour résoudre **formellement** une équation, on utilise la commande `solve` :

```
> solve(2*ln(x/3)=-1,x);
```

$$3e^{(-1/2)}$$

Si l'équation admet plusieurs solutions, `Maple` les affiche séparées par des virgules :

```
> eq := a*x^2+b*x+c=0;
```

$$eq := ax^2 + bx + c = 0$$

```
>solve(eq,x);
```

$$\frac{1}{2}\frac{-b+\sqrt{b^2-4ac}}{a}, \frac{1}{2}\frac{-b-\sqrt{b^2-4ac}}{a}$$

Si l'équation admet plusieurs solutions, `Maple` les affiche séparées par des virgules :

```
> eq := a*x^2+b*x+c=0;
```

$$eq := ax^2 + bx + c = 0$$

```
>solve(eq,x);
```

$$\frac{1}{2}\,\frac{-b+\sqrt{b^2-4ac}}{a},\ \frac{1}{2}\,\frac{-b-\sqrt{b^2-4ac}}{a}$$

Si l'équation admet plusieurs solutions, `Maple` les affiche séparées par des virgules :

```
> eq := a*x^2+b*x+c=0;
```

$$eq := ax^2 + bx + c = 0$$

```
>solve(eq,x);
```

$$\frac{1}{2}\frac{-b+\sqrt{b^2-4ac}}{a}, \frac{1}{2}\frac{-b-\sqrt{b^2-4ac}}{a}$$

Si l'équation admet plusieurs solutions, `Maple` les affiche séparées par des virgules :

```
> eq := a*x^2+b*x+c=0;
```

$$eq := ax^2 + bx + c = 0$$

```
>solve(eq,x);
```

$$\frac{1}{2}\frac{-b+\sqrt{b^2-4ac}}{a}, \frac{1}{2}\frac{-b-\sqrt{b^2-4ac}}{a}$$

Si l'équation admet plusieurs solutions, `Maple` les affiche séparées par des virgules :

```
> eq := a*x^2+b*x+c=0;
```

$$eq := ax^2 + bx + c = 0$$

```
>solve(eq,x);
```

$$\frac{1}{2}\frac{-b+\sqrt{b^2-4ac}}{a}, \frac{1}{2}\frac{-b-\sqrt{b^2-4ac}}{a}$$

Pour résoudre **formellement** un système d'équations, c'est encore `solve` qu'il faut utiliser :

```
> eq1 :=a*x+3*y=2;
```

$$eq1 := ax + 3y = 2$$

```
> eq2 :=x^2+y=1;
```

$$eq2 := x^2 + y = 1$$

```
>solve({eq1,eq2},{x,y});
```

$$\{x = RootOf(3\_Z^2 - a\_Z - 1),$$

$$y = RootOf(3\_Z^2 - a\_Z - 1) + \frac{2}{3}\}$$

Pour résoudre **formellement** un système d'équations, c'est encore `solve` qu'il faut utiliser :

```
> eq1 :=a*x+3*y=2;
```

$$eq1 := ax + 3y = 2$$

```
> eq2 :=x^2+y=1;
```

$$eq2 := x^2 + y = 1$$

```
>solve({eq1,eq2},{x,y});
```

$$\{x = RootOf(3\_Z^2 - a\_Z - 1),$$

$$y = RootOf(3\_Z^2 - a\_Z - 1) + \frac{2}{3}\}$$

Pour résoudre **formellement** un système d'équations, c'est encore `solve` qu'il faut utiliser :

```
> eq1 :=a*x+3*y=2;
```

$$eq1 := ax + 3y = 2$$

```
> eq2 :=x^2+y=1;
```

$$eq2 := x^2 + y = 1$$

```
>solve({eq1,eq2},{x,y});
```

$$\{x = RootOf(3\_Z^2 - a\_Z - 1),$$

$$y = RootOf(3\_Z^2 - a\_Z - 1) + \frac{2}{3}\}$$

Pour résoudre **formellement** un système d'équations, c'est encore `solve` qu'il faut utiliser :

```
> eq1 :=a*x+3*y=2;
```

$$eq1 := ax + 3y = 2$$

```
> eq2 :=x^2+y=1;
```

$$eq2 := x^2 + y = 1$$

```
>solve({eq1,eq2},{x,y});
```

$$\{x = RootOf(3\_Z^2 - a\_Z - 1),$$

$$y = RootOf(3\_Z^2 - a\_Z - 1) + \frac{2}{3}\}$$

Pour résoudre **formellement** un système d'équations, c'est encore `solve` qu'il faut utiliser :

```
> eq1 :=a*x+3*y=2;
```

$$eq1 := ax + 3y = 2$$

```
> eq2 :=x^2+y=1;
```

$$eq2 := x^2 + y = 1$$

```
>solve({eq1,eq2},{x,y});
```

$$\{x = RootOf(3\_Z^2 - a\_Z - 1),$$

$$y = RootOf(3\_Z^2 - a\_Z - 1) + \frac{2}{3}\}$$

Pour résoudre **formellement** un système d'équations, c'est encore `solve` qu'il faut utiliser :

```
> eq1 :=a*x+3*y=2;
```

$$eq1 := ax + 3y = 2$$

```
> eq2 :=x^2+y=1;
```

$$eq2 := x^2 + y = 1$$

```
>solve({eq1,eq2},{x,y});
```

$$\{x = RootOf(3\_Z^2 - a\_Z - 1),$$

$$y = RootOf(3\_Z^2 - a\_Z - 1) + \frac{2}{3}\}$$

Pour résoudre **formellement** un système d'équations, c'est encore `solve` qu'il faut utiliser :

```
> eq1 :=a*x+3*y=2;
```

$$eq1 := ax + 3y = 2$$

```
> eq2 :=x^2+y=1;
```

$$eq2 := x^2 + y = 1$$

```
>solve({eq1,eq2},{x,y});
```

$$\{x = RootOf(3\_Z^2 - a\_Z - 1),$$

$$y = RootOf(3\_Z^2 - a\_Z - 1) + \frac{2}{3}\}$$

Pour résoudre **numériquement** une équation, on utilise la commande `fsolve` :

```
> eq :=x^3+3*x^2=exp(x);
```

$$eq := x^3 + 3x^2 = e^x$$

```
>solve(eq,x);
```

$$RootOf(e^{\_Z} - \_Z^3 - 3\_Z^2)$$

Le système n'a pas été en mesure de calculer formellement les solutions de l'équation.

Pour résoudre **numériquement** une équation, on utilise la commande `fsolve` :

```
> eq :=x^3+3*x^2=exp(x);
```

$$eq := x^3 + 3x^2 = e^x$$

```
>solve(eq,x);
```

$$RootOf(e^{\_Z} - \_Z^3 - 3\_Z^2)$$

Le système n'a pas été en mesure de calculer formellement les solutions de l'équation.

Pour résoudre **numériquement** une équation, on utilise la commande `fsolve` :

```
> eq :=x^3+3*x^2=exp(x);
```

$$eq := x^3 + 3x^2 = e^x$$

```
>solve(eq,x);
```

$$RootOf(e^{\_Z} - \_Z^3 - 3\_Z^2)$$

Le système n'a pas été en mesure de calculer formellement les solutions de l'équation.

Pour résoudre **numériquement** une équation, on utilise la commande `fsolve` :

```
> eq :=x^3+3*x^2=exp(x);
```

$$eq := x^3 + 3x^2 = e^x$$

```
>solve(eq,x);
```

$$RootOf(e^{\_Z} - \_Z^3 - 3\_Z^2)$$

Le système n'a pas été en mesure de calculer formellement les solutions de l'équation.

Pour résoudre **numériquement** une équation, on utilise la commande `fsolve` :

```
> eq :=x^3+3*x^2=exp(x);
```

$$eq := x^3 + 3x^2 = e^x$$

```
>solve(eq,x);
```

$$RootOf(e^{\_Z} - \_Z^3 - 3\_Z^2)$$

Le système n'a pas été en mesure de calculer formellement les solutions de l'équation.

Pour résoudre **numériquement** une équation, on utilise la commande `fsolve` :

```
> eq :=x^3+3*x^2=exp(x);
```

$$eq := x^3 + 3x^2 = e^x$$

```
>solve(eq,x);
```

$$RootOf(e^{\_Z} - \_Z^3 - 3\_Z^2)$$

Le système n'a pas été en mesure de calculer formellement les solutions de l'équation.

La commande `fsolve` permet de chercher une valeur approchée des solutions de l'équation :

```
> fsolve(eq,x=-4..-2);
```

$$-2.994416364$$

```
> fsolve(eq,x=-1..0);
```

$$-.4935166498$$

```
> fsolve(eq,x=0..1);
```

$$.7518696378$$

La commande `fsolve` permet de chercher une valeur approchée des solutions de l'équation :

```
> fsolve(eq,x=-4..-2);
```

$$-2.994416364$$

```
> fsolve(eq,x=-1..0);
```

$$-.4935166498$$

```
> fsolve(eq,x=0..1);
```

$$.7518696378$$

La commande `fsolve` permet de chercher une valeur approchée des solutions de l'équation :

```
> fsolve(eq,x=-4..-2);
```

$$-2.994416364$$

```
> fsolve(eq,x=-1..0);
```

$$-.4935166498$$

```
> fsolve(eq,x=0..1);
```

$$.7518696378$$

La commande `fsolve` permet de chercher une valeur approchée des solutions de l'équation :

```
> fsolve(eq,x=-4..-2);
```

$$-2.994416364$$

```
> fsolve(eq,x=-1..0);
```

$$-.4935166498$$

```
> fsolve(eq,x=0..1);
```

$$.7518696378$$

La commande `fsolve` permet de chercher une valeur approchée des solutions de l'équation :

```
> fsolve(eq,x=-4..-2);
```

$$-2.994416364$$

```
> fsolve(eq,x=-1..0);
```

$$-.4935166498$$

```
> fsolve(eq,x=0..1);
```

$$.7518696378$$

La commande `fsolve` permet de chercher une valeur approchée des solutions de l'équation :

```
> fsolve(eq,x=-4..-2);
```

$$-2.994416364$$

```
> fsolve(eq,x=-1..0);
```

$$-.4935166498$$

```
> fsolve(eq,x=0..1);
```

$$.7518696378$$

La commande `fsolve` permet de chercher une valeur approchée des solutions de l'équation :

```
> fsolve(eq,x=-4..-2);
```

$$-2.994416364$$

```
> fsolve(eq,x=-1..0);
```

$$-.4935166498$$

```
> fsolve(eq,x=0..1);
```

$$.7518696378$$

Pour résoudre une équation différentielle, on utilise `dsolve` :

```
> eq :=diff(y(x),x)-y(x)=exp(x);
```

$$eq := (\frac{\partial}{\partial x} y(x)) - y(x) = e^x$$

```
> dsolve(eq,y(x));
```

$$y(x) = (x + \_C1)e^x$$

La constante d'intégration est notée $\_C1$.

Pour résoudre une équation différentielle, on utilise `dsolve` :

```
> eq :=diff(y(x),x)-y(x)=exp(x);
```

$$eq := (\frac{\partial}{\partial x} y(x)) - y(x) = e^x$$

```
> dsolve(eq,y(x));
```

$$y(x) = (x + \_C1)e^x$$

La constante d'intégration est notée $\_C1$.

Pour résoudre une équation différentielle, on utilise `dsolve` :

```
> eq :=diff(y(x),x)-y(x)=exp(x);
```

$$eq := (\frac{\partial}{\partial x} y(x)) - y(x) = e^x$$

```
> dsolve(eq,y(x));
```

$$y(x) = (x + \_C1)e^x$$

La constante d'intégration est notée $\_C1$.

Pour résoudre une équation différentielle, on utilise `dsolve` :

```
> eq :=diff(y(x),x)-y(x)=exp(x);
```

$$eq := (\frac{\partial}{\partial x} y(x)) - y(x) = e^x$$

```
> dsolve(eq,y(x));
```

$$y(x) = (x + \_C1)e^x$$

La constante d'intégration est notée $\_C1$.

Pour résoudre une équation différentielle, on utilise `dsolve` :

```
> eq :=diff(y(x),x)-y(x)=exp(x);
```

$$eq := (\frac{\partial}{\partial x}y(x)) - y(x) = e^x$$

```
> dsolve(eq,y(x));
```

$$y(x) = (x + \_C1)e^x$$

La constante d'intégration est notée $\_C1$.

Pour résoudre une équation différentielle, on utilise `dsolve` :

```
> eq :=diff(y(x),x)-y(x)=exp(x);
```

$$eq := (\frac{\partial}{\partial x}y(x)) - y(x) = e^x$$

```
> dsolve(eq,y(x));
```

$$y(x) = (x + \_C1)e^x$$

La constante d'intégration est notée $\_C1$.

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x+1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x+1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x+1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x+1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x+1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x+1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x+1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x+1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x+1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x + 1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x + 1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x + 1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x + 1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x + 1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x + 1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x + 1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x + 1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x + 1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x + 1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x + 1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x + 1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x + 1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x + 1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x + 1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x + 1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x + 1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x + 1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x + 1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x + 1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x + 1)e^x$$

Pour résoudre l'équation différentielle avec des **conditions initiales**, par exemple avec $y(0) = 1$, on doit les rajouter à la commande `dsolve` :

```
> dsolve({eq,y(0)=1},y(x));
```

$$y(x) = (x+1)e^x$$

`Maple` renvoie un objet qui est une équation. Le membre de gauche est $y(x)$. Le membre de droite est $(x+1)e^x$. Pour stocker ce membre de droite, on utilise la commande `rhs` (Right Hand Side) :

```
> sol1 :=rhs(%);
```

$$sol1 := (x+1)e^x$$

On procède de même pour obtenir d'autres solutions avec des conditions initiales différentes :

```
> dsolve({eq,y(0)=0}, y(x));
```

$$y(x) = e^x x$$

```
> sol2 := rhs(%);
```

$$sol2 := e^x x$$

```
> dsolve({eq,y(0)=-1}, y(x));
```

$$y(x) = (x-1)e^x$$

```
> sol3 :=rhs(%);
```

$$sol3 := (x-1)e^x$$

On procède de même pour obtenir d'autres solutions avec des conditions initiales différentes :

```
> dsolve({eq,y(0)=0}, y(x));
```

$$y(x) = e^x x$$

```
> sol2 := rhs(%);
```

$$sol2 := e^x x$$

```
> dsolve({eq,y(0)=-1}, y(x));
```

$$y(x) = (x-1)e^x$$

```
> sol3 :=rhs(%);
```

$$sol3 := (x-1)e^x$$

On procède de même pour obtenir d'autres solutions avec des conditions initiales différentes :

```
> dsolve({eq,y(0)=0}, y(x));
```

$$y(x) = e^x x$$

```
> sol2 := rhs(%);
```

$$sol2 := e^x x$$

```
> dsolve({eq,y(0)=-1}, y(x));
```

$$y(x) = (x-1)e^x$$

```
> sol3 :=rhs(%);
```

$$sol3 := (x-1)e^x$$

On procède de même pour obtenir d'autres solutions avec des conditions initiales différentes :

```
> dsolve({eq,y(0)=0}, y(x));
```

$$y(x) = e^x x$$

```
> sol2 := rhs(%);
```

$$sol2 := e^x x$$

```
> dsolve({eq,y(0)=-1}, y(x));
```

$$y(x) = (x-1)e^x$$

```
> sol3 :=rhs(%);
```

$$sol3 := (x-1)e^x$$

On procède de même pour obtenir d'autres solutions avec des conditions initiales différentes :

```
> dsolve({eq,y(0)=0}, y(x));
```

$$y(x) = e^x x$$

```
> sol2 := rhs(%);
```

$$sol2 := e^x x$$

```
> dsolve({eq,y(0)=-1}, y(x));
```

$$y(x) = (x - 1)e^x$$

```
> sol3 :=rhs(%);
```

$$sol3 := (x - 1)e^x$$

On procède de même pour obtenir d'autres solutions avec des conditions initiales différentes :

```
> dsolve({eq,y(0)=0}, y(x));
```

$$y(x) = e^x x$$

```
> sol2 := rhs(%);
```

$$sol2 := e^x x$$

```
> dsolve({eq,y(0)=-1}, y(x));
```

$$y(x) = (x-1)e^x$$

```
> sol3 :=rhs(%);
```

$$sol3 := (x-1)e^x$$

On procède de même pour obtenir d'autres solutions avec des conditions initiales différentes :

```
> dsolve({eq,y(0)=0}, y(x));
```

$$y(x) = e^x x$$

```
> sol2 := rhs(%);
```

$$sol2 := e^x x$$

```
> dsolve({eq,y(0)=-1}, y(x));
```

$$y(x) = (x-1)e^x$$

```
> sol3 :=rhs(%);
```

$$sol3 := (x-1)e^x$$

On procède de même pour obtenir d'autres solutions avec des conditions initiales différentes :

```
> dsolve({eq,y(0)=0}, y(x));
```

$$y(x) = e^x x$$

```
> sol2 := rhs(%);
```

$$sol2 := e^x x$$

```
> dsolve({eq,y(0)=-1}, y(x));
```

$$y(x) = (x-1)e^x$$

```
> sol3 :=rhs(%);
```

$$sol3 := (x-1)e^x$$

On procède de même pour obtenir d'autres solutions avec des conditions initiales différentes :

```
> dsolve({eq,y(0)=0}, y(x));
```

$$y(x) = e^x x$$

```
> sol2 := rhs(%);
```

$$sol2 := e^x x$$

```
> dsolve({eq,y(0)=-1}, y(x));
```

$$y(x) = (x - 1)e^x$$

```
> sol3 :=rhs(%);
```

$$sol3 := (x - 1)e^x$$

Pour résoudre une équation récurrente, on utilise la commande `rsolve` :

```
> eq :=u(n)=2*u(n-1);
```

$$eq := u(n) = 2u(n-1)$$

```
> rsolve(eq,u(n));
```

$$u(0)2^n$$

Pour résoudre une équation récurrente, on utilise la commande `rsolve` :

```
> eq :=u(n)=2*u(n-1);
```

$$eq := u(n) = 2u(n-1)$$

```
> rsolve(eq,u(n));
```

$$u(0)2^n$$

Pour résoudre une équation récurrente, on utilise la commande `rsolve` :

```
> eq :=u(n)=2*u(n-1);
```

$$eq := u(n) = 2u(n-1)$$

```
> rsolve(eq,u(n));
```

$$u(0)2^n$$

Pour résoudre une équation récurrente, on utilise la commande `rsolve` :

```
> eq :=u(n)=2*u(n-1);
```

$$eq := u(n) = 2u(n-1)$$

```
> rsolve(eq,u(n));
```

$$u(0)2^n$$

Pour résoudre une équation récurrente, on utilise la commande `rsolve` :

```
> eq :=u(n)=2*u(n-1);
```

$$eq := u(n) = 2u(n-1)$$

```
> rsolve(eq,u(n));
```

$$u(0)2^n$$

Pour résoudre une équation récurrente avec **conditions initiales**, on doit les sajouter à la commande `rsolve` :

```
> eq :=u(n)=n*u(n-1);
```

$$eq := u(n) = nu(n-1)$$

```
> rsolve({eq,u(0)=1}, u(n));
```

$$\Gamma(n+1)$$

La fonction Γ est le prolongement à $\mathbb{C}$ de la factorielle $n!$ définie sur $\mathbb{N}$.

Pour résoudre une équation récurrente avec **conditions initiales**, on doit les sajouter à la commande `rsolve` :

```
> eq :=u(n)=n*u(n-1);
```

$$eq := u(n) = nu(n-1)$$

```
> rsolve({eq,u(0)=1}, u(n));
```

$$\Gamma(n+1)$$

La fonction Γ est le prolongement à $\mathbb{C}$ de la factorielle $n!$ définie sur $\mathbb{N}$.

Pour résoudre une équation récurrente avec **conditions initiales**, on doit les sajouter à la commande `rsolve` :

```
> eq :=u(n)=n*u(n-1);
```

$$eq := u(n) = nu(n-1)$$

```
> rsolve({eq,u(0)=1}, u(n));
```

$$\Gamma(n+1)$$

La fonction Γ est le prolongement à $\mathbb{C}$ de la factorielle $n!$ définie sur $\mathbb{N}$.

Pour résoudre une équation récurrente avec **conditions initiales**, on doit les sajouter à la commande `rsolve` :

```
> eq :=u(n)=n*u(n-1);
```

$$eq := u(n) = nu(n-1)$$

```
> rsolve({eq,u(0)=1}, u(n));
```

$$\Gamma(n+1)$$

La fonction Γ est le prolongement à $\mathbb{C}$ de la factorielle $n!$ définie sur $\mathbb{N}$.

Pour résoudre une équation récurrente avec **conditions initiales**, on doit les sajouter à la commande `rsolve` :

```
> eq :=u(n)=n*u(n-1);
```

$$eq := u(n) = nu(n-1)$$

```
> rsolve({eq,u(0)=1}, u(n));
```

$$\Gamma(n+1)$$

La fonction Γ est le prolongement à $\mathbb{C}$ de la factorielle $n!$ définie sur $\mathbb{N}$.

Pour résoudre une équation récurrente avec **conditions initiales**, on doit les sajouter à la commande `rsolve` :

```
> eq :=u(n)=n*u(n-1);
```

$$eq := u(n) = nu(n-1)$$

```
> rsolve({eq,u(0)=1}, u(n));
```

$$\Gamma(n+1)$$

La fonction Γ est le prolongement à $\mathbb{C}$ de la factorielle $n!$ définie sur $\mathbb{N}$.

Pour résoudre une équation récurrente avec **conditions initiales**, on doit les sajouter à la commande `rsolve` :

```
> eq :=u(n)=n*u(n-1);
```

$$eq := u(n) = nu(n-1)$$

```
> rsolve({eq,u(0)=1}, u(n));
```

$$\Gamma(n+1)$$

La fonction Γ est le prolongement à $\mathbb{C}$ de la factorielle $n!$ définie sur $\mathbb{N}$.