La dérivée en animation

© Pierre Lantagne
Enseignant retraité du Collège de Maisonneuve

Ce document est une révision de celui produit en 2000.

Bonne lecture à tous !

* Ce document Maple est exécutable avec la version 2020.1

Animations de la dérivée

```
> restart;
  with(plots,display, setoptions):
  setoptions( axesfont=["TIMES",roman,8],size=[300,300]);
```

L'initialisation suivante permettra d'avoir plus de lisibilité des nombres décimaux en supprimamt les zéros non significatifs à la fin d'un nombre.

```
> interface(typesetting=extended); # Pour s'assurer le niveau de
  composition étendue
  Typesetting:-Settings(striptrailing=true);
```

extended

false **(1.1)**

Initialisation

Le bloc d'instructions ci-dessous compose une procédure nommée **`sécante`**.
La procédure **`sécante`** permet de réaliser une animation illustrant la tangente comme position limite de différentes sécantes au graphique d'une fonction f.

Positionnez le curseur clignotant dans le bloc ci-dessous et exécutez tout le bloc d'instructions en pressant la touche entrée.

```
> sécante := proc(f::procedure, t1::name=realcons, t2::name=
  realcons,
             séquences::name=posint, pavée::name=list,dir::name)
   local i,g,n,p,y0,y1,dx,vuex,vuey,x0,x1,xs,Tangente,Courbe,A,B,C;
   if not member(dir,{gauche, droite}) then
     ERROR("Le sixième argument «dir» doit être l'une\ndes
  modalitéss suivantes: gauche, droite.") fi;
   x0:=rhs(t1);
   x1:=rhs(t2);
   n:=rhs(séquences);
   vuex:=op(1,rhs(pavée));
```

```
 vuey:=op(2,rhs(pavée));
 if evalf(x0)>=evalf(x1) then
  ERROR("t0 doit être strictement inférieure à t1")
 fi;
 dx := (x1-x0)/n;
 if dir=droite then
   y0 := f(x0); # point de tangence
   for i from 1 to n do
   xs:= x1-dx*(i-1);
   g[i]:=student[makeproc]([x0,y0],[xs,f(xs)]);
   p[i]:=plot([[[x0,0],[x0,y0]],f(x),g[i](x),[[x0,0],[x0,y0],[xs,
y0],[xs,f(xs)]]],x=vuex,vuey,
        color=[orange,navy,khaki,khaki],linestyle=[2,1,1,2],
numpoints=60):
   od;
  Tangente:=plot([x,D(f)(x0)*(x-x0)+y0,x=vuex],color=orange,
thickness=2):
  Courbe:=plot([x,f(x),x=vuex],color=navy):
  A:=plot([[x0,0],[x0,y0]],color=orange,linestyle=2):
  B:=plots[display](A,Courbe,Tangente):
  C:=plots[display](seq(p[i],i=1..n),insequence=true):
  plots[display]([C,B],insequence=true,view=rhs(pavée)):
 else
  y1:=f(x1); # point de tangence
  for i from 1 to n do
  xs:=x0+dx*(i-1);
  g[i]:=student[makeproc]([xs,f(xs)],[x1,f(x1)]);
  p[i]:=plot([[[x0,0],[x0,y0]],f(x),g[i](x),[[x1,0],[x1,f(x1)],
[x1,f(xs)],[xs,f(xs)]]],
        x=vuex,vuey,color=[orange,navy,khaki,khaki],linestyle=[2,
1,1,2],numpoints=60)
  od:
 Tangente:=plot([x,D(f)(x1)*(x-x1)+f(x1),x=vuex],color=orange,
thickness=2):
 Courbe:=plot([x,f(x),x=vuex],color=navy):
 A:=plot([[x1,0],[x1,f(x1)]],color=orange,linestyle=2):
 B:=display(A,Courbe,Tangente):
 C:=display(seq(p[i],i=1..n),insequence=true):
 display([C,B],insequence=true,view=rhs(pavée)):
fi
end:
```

Même si Maple n'a rien affiché dans la zone des résultats, vous disposerez, pour toute la durée de cette sessoin Maple,
d'une macro-commande qui servira à réaliser des animations illustrant la position limite de différentes sécantes à une courbe.

La macro-commande sécante

```
sécante(f, t1 = u, t2 = v, séquences=n, view=[a..b,c..d], dir)
```

avec $t_1 < t_2$.

anime dans le plan la position limite des sécantes à une fonction f passant par les points de coordonnées [t_1,$f(t_1)$] et [t_2,$f(t_2)$].

Le point [t_1,$f(t_1)$] ou le point [t_2,$f(t_2)$] sera considéré comme le point fixe (point de tangence) selon la valeur dir:

- si la modalité de dir est droite, [t_1,$f(t_1)$] sera considéré comme le point fixe, donc, une animation illustrant la dérivée à droite de $[t_1,f(t_1)]$]
- si la modalité de dir est gauche, [t_2,$f(t_2)$] sera considéré comme le point fixe, donc, une animation illustrant la dérivée à gauche de [t_2,$f(t_2)$]

Le nombre de séquences (frames) de l'animation est précisé par la valeur *n* de séquences.
La vue de l'animation est présentée sur le pavé [*a*, *b*] x [*c*, *d*].

La tangente comme position limite d'une suite de sécantes

Soit la fonction f dont la règle est $f(x) = 2.5 - \sqrt{4 - x^2}$.
À l'aide de la macro-commande **sécante**, animez les différentes sécantes au graphe de f passant par le point [1,2; f(1,2)].

Créez d'abord la fonction f avec l'opérateur flèche.

```
> f:=x->2.5-sqrt(4-x^2);
```

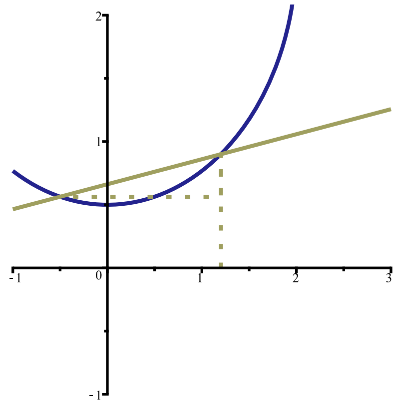
$$f := x \mapsto 2.5 - \sqrt{4 - x^2} \quad \textbf{(1.3.1)}$$

Réalisez ensuite une première animation avec un nombre de séquences (frames) égale à 10 et avec une vue sur le pavé [-1, 3] x [-1, 3].

Attention: Dans la macro-commande **sécante**, il faut indiquer obligatoirement un dernier argument ***droite*** ou ***gauche*** :

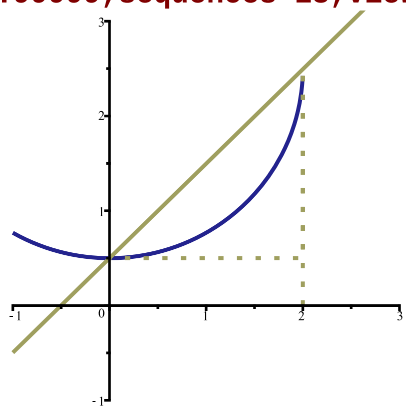
- droite pour spécifier $(t_1,f(t_1))$ comme point de tangence et donc pour des animations approchant t_1 par la droite à partir de t_2
- gauche pour spécifier $(t_2,f(t_2))$ comme point de tangence et donc pour des animations approchant t_2 par la gauche à partir de t_1

```
> sécante(f,x1=-0.5,x2=1.2,séquences=18,view=[-1..3,-1..2],gauche);
```



Pour lancez l'animation, cliquez sur le tracé obtenu puis cliquez sur le bouton de départ (start) de la barre de menu contextuel .

```
> sécante(f,x1=0,x2=1.99999,séquences=18,view=[-1..3,-1..3],gauche);
```



Lorsque que le graphique seront affichés, cliquez-les.
Ensuite lancez l'animation en cliquant sur le bouton de départ (start) du menu contextuel.

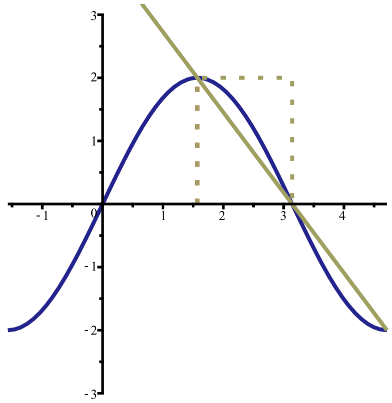
Soit maintenant la fonction g définie par $g(t) = 2\sin(t)$.
Animez les différentes sécantes au graphe de g en traitant le point $\left[\frac{\pi}{2}, g\left(\frac{\pi}{2}\right)\right]$ comme point de tangence.

```
> g:=t->2*sin(t);
```

$$g := t \mapsto 2\cdot\sin(t) \qquad \textbf{(1.3.2)}$$

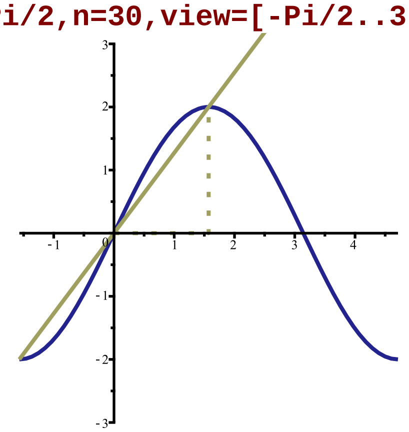
Animation sur l'intervalle de la forme $\left[\frac{\pi}{2}, t\right]$.

```
> sécante(g,t1=Pi/2,t2=Pi,n=15,view=[-Pi/2..3*Pi/2,-3..3],droite);
```



Animation sur l'intervalle de la forme $\left[t, \frac{\pi}{2}\right]$.

```
> sécante(g,t1=0,t2=Pi/2,n=30,view=[-Pi/2..3*Pi/2,-3..3],droite);
```



Animation du tracé du graphique de la fonction dérivée d'une fonction

(*Auteur: Dr.G.Bitsch, du Kepler-Gymnasium*)

La procédure suivante annime de façon remarquable le tracé du graphique de la fonction dérivée. L'intérêt de cette animation repose sur l'interprétation graphique de la dérivée qui est illustrée de façon très dynamique.

C'est donc avec un grand plaisir que je vous propose la traduction et l'adaptation du dévelopement du Dr. G.Bitsch, professeur au Kepler-Gymnasium.

La procédure **diffpic** appellera deux procédures devant préalablement être définies: **tangente** et **tang**.

La première procédure **tangente** définit en fait une macro-commande permettant d'établir l'équation de la tangente au point $P\left(x_0, y_0\right)$ d'une fonction f. La fonction doit être passée à la procédure par son son nom lui-même et non pas par son expression f(x) .

```
> tangente:=proc(x0::realcons,f::procedure)
  D(f)(x0)*(x-x0)+f(x0);
  end:
```

La deuxième procédure **tang** définit une macro-commande permettant d'obtenir les coordonnées de deux points "symétriquement" de part et d'autre du point de tangence

$P(x_0, y_0)$ de la fonction f le long de la droite tangente.

La valeur de h est utilisée pour obtenir les abscisses de ces points et donc détermine la longueur de la tangente.

```
> tang:=proc(x0::numeric,h::numeric,f::procedure)
  [[x0-h,evalf(subs(x=x0-h,tangente(x0,f)))],[x0+h,evalf(subs(x=x0+
  h,tangente(x0,f)))]];
  end:
```

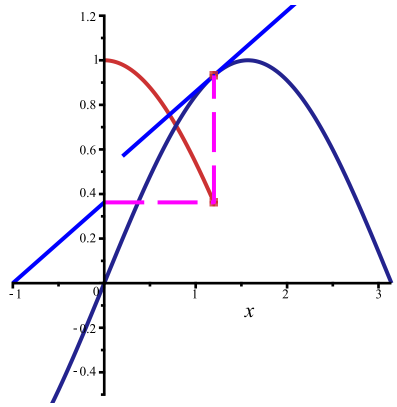
C'est avec un souci pédagogique que ces deux procédures **tangente** et **tang** n'ont pas été directement intégrées
à la procédure principale **diffpict**.

La troisième procédure **diffpict** est une une macro-commande permettant d'obtenir le tracé du graphique de la dérivée d'une fonction f sur un intervalle donné dont la borne de droite doit être l'abscisse du point de tangence. En traçant, à partir du point (-1,0), un segment de droite parallèle à la tangente passant par le point $P(x_0, y_0)$, l'ordonnée de l'extrémité de ce segment correspondra à la valeur de la pente de cette tangente au point $P(x_0, y_0)$: ce qui permet de visualiser le tracé du point P'(x_0,f'(x_0)).

```
> diffpic:=proc(x0::numeric,h::numeric,f::procedure,vue::name=list)
  local p1,p2,p3,p4,p5,xr,yr,y0,y1:
   xr:=op(1,rhs(vue));
   yr:=op(2,rhs(vue));
   y0:=evalf(f(x0)):
   y1:=evalf(D(f)(x0)):
   p1:=plot(f(x),x=xr,yr,color=navy):
   p2:=plot({tang(x0,h,f),[[-1,0],[0,y1]]},x=xr,yr,color=blue):
   p3:=plot({[x0,y0],[x0,y1]},style=point, symbol=circle,color=
  orange):
   p4:=plot([[0,y1],[x0,y1],[x0,y0]],style=line,linestyle=3,color=
  magenta):
   p5:=plot(diff(f(x),x),x=0..x0,yr,color=orange):
   display({p1,p2,p3,p4,p5},view=[xr,yr]);
  end:
```

Exemple:

```
> diffpic(1.2,1,sin,view=[-1..Pi,-0.5..1.2]);
```



Finalement, la dernière procédure **ani_diff** est une une macro-commande permettant d'animer le tracé de la dérivée d'une fonction f sur un intervalle [a,b]. Le dernier argument de cette procédure précise le nombre

de séquences (frames) à bâtir pour réaliser l'animation.

Nous allons maintenant inclure les procédures *tang* et *tangente* dans la procédure *ani_diff*. En conséquence, il ne sera plus nécessaire de les exécuter séparément.

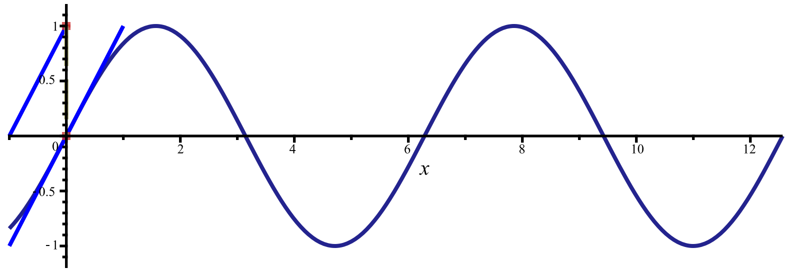
```
> ani_diff:=proc(h::name=realcons,f::procedure,vue::name=list,
  n::name=posint)
  local Opts,tang, tangente, p1,p2,p3,p4,p5,p6,i,f1,fx0,f1x0,dx,x0,
  xr,yr:
  Opts:=[args[5..nargs]];
  tang:=proc(x0::numeric,h::numeric,f::procedure)
    [[x0-h,evalf(subs(x=x0-h,tangente(x0,f)))],[x0+h,evalf(subs(x=
  x0+h,tangente(x0,f)))]]
    end proc:
    tangente:=proc(x0::realcons,f::procedure)
     D(f)(x0)*(x-x0)+f(x0);
     end proc:
   xr:=op(1,rhs(vue));
   yr:=op(2,rhs(vue));
   dx:=op(2,xr)/rhs(n);
   for i from 0 to rhs(n) do
     x0:=evalf(i*dx);
     fx0:=evalf(f(x0));
     f1x0:=evalf(D(f)(x0));
     p1[i]:=plot(f(x),x=xr,yr,color=navy):
     p2[i]:=plot({tang(x0,rhs(h),f),[[-1,0],[0,f1x0]]},x=xr,yr,
            color=BLUE):
     p3[i]:=plot({[x0,fx0],[x0,f1x0]},style=point,
     symbol=circle,color=orange):
     p4[i]:=plot([[0,f1x0],[x0,f1x0],[x0,fx0]],style=line,
            linestyle=3,color=khaki):
     p5[i]:=plot([x,D(f)(x),x=0..x0],x=xr,yr,color=orange,thickness=
```

```
2):
   p6[i]:=plots[display]({p1[i],p2[i],p3[i],p4[i],p5[i]},view=[xr,
yr]):
 od:
plots[display]([seq(p6[i],i=0..rhs(n))],insequence=true,op(Opts));
end:
```

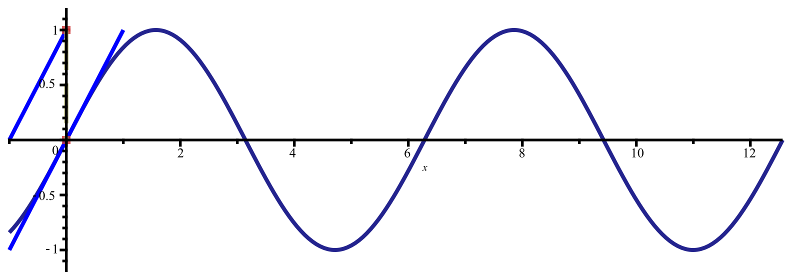
Graphique de la dérivée de la fonction f définie par $f(x) = \sin(x)$

.

```
> ani_diff(h=1,sin,view=[-1..4*Pi,-1.2..1.2],n=24,size=[600,200]);
```

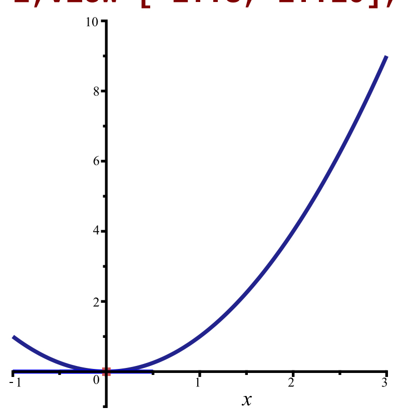



```
> T:=ani_diff(h=1,sin,view=[-1..4*Pi,-1.2..1.2],n=24):
  plots[display](T,insequence=true,font=["TIMES",roman,6],size=[600,
  200]);
```



Graphique de la dérivée de la fonction f définie par $f(x) = x^2$

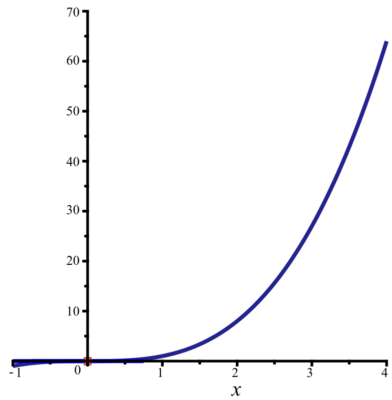
```
> ani_diff(h=1/2,x->x^2,view=[-1..3,-1..10],n=18,size=[300,300]);
```



Graphique de la dérivée de la fonction f définie par $f(x) = x^3$

```
> f:=x->x^3;
  ani_diff(h=3/4,f,view=[-1..4,-1..70],n=18,size=[300,300]);
```

$$f := x \mapsto x^3$$



70
60
50
40
30
20
10
-1
0
1
2
3
4
x