Codes Scilab pour l'exemple de texte

I. Modélisation : temps de descente d'une bille

Première simulation : temps de descente de la bille pour trois formes de toboggan, à l'aide d'une discrétisation Euler explicite sur l'EDO donnant l'abscisse de la bille.

```
clear()

function y=f1(x)
    y=sqrt(10*x)
endfunction

function y=f2(x)
    y=sqrt(20*sqrt(x))/sqrt(1+(1/(4*x)))
endfunction

function y=g(x)
    y=(x.^(3/2)).*(5-3*x)/2
endfunction

function y=h(x)
    y=((3/2)*(x.^(1/2)).*(5-3*x)-3*x^(3/2))/2
endfunction

function y=f3(x)
    y=sqrt(20*g(x))/(sqrt(1+h(x)^2))
endfunction

//dt = 0.01
//x = 0.001
//L=[x]
//for i=1:10
//    x=x+dt*f3(x)
//    L=[L,x]
//end
//
//X=0:0.1:1
//for i=1:10
//    plot(L(i),-sqrt(L(i)),'o')
//
//end
```

```
//
//X=0:0.1:1
//x = 0.0001;
//t = 0;
//Tf = 10;
//
//while t < Tf & x<1
//    t = t+dt;
//    x=x+dt*f(x)
//    drawlater();
//    clf();
//    a = gca();
//    a.data_bounds = [0 -1;1 0];
//    plot(x,-x,'o',X,-X);
//    drawnow();
//end
//
//
//
X=0:0.01:1
x = 0.0001;
y=0.0001;
z=0.0001;
t = 0;
Tf = 10;
dt=0.01;

//while t < Tf & x<1 & y<1
//    t = t+dt;
//    x=x+dt*f2(x)
//    y=y+dt*f1(y)
//    drawlater();
//    clf();
//    a = gca();
//    a.data_bounds = [0 -1;1.1 0];
//    subplot(2,1,1)
//    plot(y,-y,'o',X,-X);
//    b=gca();
//    b.data_bounds = [0 -1;1.1 0];
//    subplot(2,1,2)
//    plot(x,-sqrt(x),'o',X,-sqrt(X));
//    drawnow();
//end

while t < Tf & z<1 & x<1 & y<1
    drawlater();
    clf();
    a = gca();
    a.data_bounds = [0 -1;1 0];
    subplot(3,1,1)
    plot(y,-y,'o',X,-X);
    xtitle('Descente du toboggan plan', '$x$', '$y$')
    subplot(3,1,2)
    plot(x,-sqrt(x),'o',X,-sqrt(X));
    xtitle('Descente du toboggan raide', '$x$', '$y$')
    subplot(3,1,3)
    plot(z,-g(z),'o',X,-g(X));
    xtitle('Descente du toboggan doux', '$x$', '$y$')
```

```
    drawnow();
    t = t+dt;
    x=x+dt*f2(x)
    y=y+dt*f1(y)
    z=z+dt*f3(z)
end
```

Deuxième simulation : temps de descente de la bille pour le toboggan "doux", à l'aide d'une discrétisation Euler explicite sur l'EDO donnant l'abscisse de la bille.

(Non montré lors de la présentation)

```
clear()

function y=g(x)
    y=(x.^(3/2)).*(5-3*x)/2
endfunction

function y=h(x)
    y=((3/2)*(x.^(1/2)).*(5-3*x)-3*x^(3/2))/2
endfunction

function y=f1(x)
    y=sqrt(20*g(x))/(sqrt(1+h(x)^2))
endfunction

X=0:0.01:1
x = 0.0001;
t = 0;
Tf = 10;
dt = 0.005 ;

while t < Tf & x<1
    drawlater();
    clf();
    a = gca();
    a.data_bounds = [0 -1;1 0];
    chaine=['Temps=',string(t)]
    xstring(0.7,-0.2,chaine)
    plot(x,-g(x),'o',X,-g(X));
    xtitle('Descente du toboggan doux','x','y')
    drawnow();
    t = t+dt;
    x=x+dt*f1(x)
end
```

Troisième simulation : temps de descente de la bille pour le toboggan "plan", à l'aide d'une discrétisation Euler explicite sur l'EDO donnant l'abscisse de la bille.

(Non montré lors de la présentation)

```
clear()

function y=f1(x)
    y=sqrt(10*x)
endfunction

X=0:0.1:1
x = 0.0001;
t = 0;
Tf = 10;
dt = 0.005 ;

while t < Tf & x<1
    drawlater();
    clf();
    a = gca();
    a.data_bounds = [0 -1;1 0];
    chaine=['Temps=',string(t)]
    xstring(0.7,-0.2,chaine)
    xtitle('Descente du toboggan plan','x','y')
    plot(x,-x,'o',X,-X);
    drawnow();
    t = t+dt;
    x=x+dt*f1(x)
end
```

Quatrième simulation : temps de descente de la bille pour le toboggan "raide", à l'aide d'une discrétisation Euler explicite sur l'EDO donnant l'abscisse de la bille.

(Non montré lors de la présentation)

```
clear()

function y=f1(x)
    y=sqrt(20*sqrt(x))/sqrt(1+(1/(4*x)))
endfunction

X=0:0.01:1
x = 0.0001;
```

```
t = 0;
Tf = 10;
dt = 0.005 ;

while t < Tf & x<1
    drawlater();
    clf();
    a = gca();
    a.data_bounds = [0 -1;1 0];
    chaine=['Temps=',string(t)]
    xstring(0.7,-0.2,chaine)
    xtitle('Descente du toboggan raide','x','y')
    plot(x,-sqrt(x),'o',X,-sqrt(X));
    drawnow();
    t = t+dt;
    x=x+dt*f1(x)
end
```

II. Méthode classique d'approximation d'intégrale mise en défaut

Cinquième simulation : Illustration de la dégénérescence de l'ordre de la méthode du point milieu lorsque les fonctions intégrées présentent une singularité en zéro.

```
clear()

//Point milieu

N = 5000 ;
dx = 1/N
x = [0:dx:1-dx] ;

function y=f1(x)
    y=(1+x.^2).*x.^(-3/4)
endfunction

function y=f2(x)
    y=x.^(-3/4)
endfunction

function y=f3(x)
    y=x.^(-5/6)
endfunction


//I = (1/N) * sum(f(x+(dx/2))) ;
```

```
L1=[]
L2=[]
L3=[]
nmin = 100 ;
nmax = 1000 ;

for N=nmin:nmax
    dx = 1/N
    x= [0:dx:1-dx]
    I1 = (1/N) * sum(f1(x+(dx/2)))
    I2 = (1/N) * sum(f2(x+(dx/2)))
    I3 = (1/N) * sum(f3(x+(dx/2)))
    L1=[L1,I1]
    L2=[L2,I2]
    L3=[L3,I3]
end

plot(log(nmin:1:nmax) , log(abs(4-L2)))
xtitle('Estimation erreur point milieu avec singularité pour f(x)=x.^(-3/4)',
 '$\log(N)$','$\log(I-I_N)$')
[a2,b,r]=reglin(log(nmin:1:nmax) , log(abs(4-L2))) ;
chaine2=['Pente=', string(a2)] ;
xstring(6,-0.3, chaine2)

xclick()
close


plot(log(nmin:1:nmax) , log(abs((40/9)-L1)))
xtitle('Estimation erreur point milieu avec singularité pour f(x)=(1+x.^2).*x.^(-
3/4)', '$\log(N)$','$\log(I-I_N)$')
[a1,b,r]=reglin(log(nmin:1:nmax) , log(abs(40/9-L1))) ;
chaine1=['Pente=', string(a1)] ;
xstring(6,-0.3, chaine1)

xclick()
close


plot(log(nmin:1:nmax) , log(abs(6-L3)))
xtitle('Estimation erreur point milieu avec singularité pour f(x)=x^-5/6', '$\log
(N)$','$\log(I-I_N)$')
[a3,b,r]=reglin(log(nmin:1:nmax) , log(abs(6-L3))) ;
chaine3=['Pente=', string(a3)] ;
xstring(6,0.65, chaine3)

function y=f4(x)
    y=sqrt(10*x).^(-1)
endfunction

function y=f5(x)
    y=((sqrt(1+(4*x).^(-1)))).*sqrt(20*sqrt(x)).^(-1)
endfunction

function y=g(x)
    y=1/2.*(x.^(3/2)).*(5-3*x)
endfunction
```

```
function y=h(x)
    y=((3/2)*(x.^(1/2)).*(5-3*x)-3*x.^(3/2))/2
endfunction

function y=f6(x)
    y=(sqrt(1+h(x).^2)).*((sqrt(20*g(x)))).^(-1)
endfunction

N1= 1000
dx1 = 1/N1
x1= [0:dx1:1-dx1]

I4 = (1/N1) * sum(f4(x1+(dx1/2)))
I5 = (1/N1) * sum(f5(x1+(dx1/2)))
I6 = (1/N1) * sum(f6(x1+(dx1/2)))

disp(N1,'Pour un nombre de points N=')
disp(I4,'Methode point milieu : Temps de descente pour le toboggan doux')
disp(I5,'Methode point milieu : Temps de descente pour le toboggan raide')
disp(I6,'Methode point milieu : Temps de descente pour le toboggan doux')
```

Sixième simulation : Illustration de la dégénérescence de l'ordre de la méthode de rectangle à droite lorsque les fonctions intégrées présentent une singularité en zéro.

```
clear()

//Rectangle à droite
function y=f(x)
    y=x.^(-3/4)
endfunction

L=[]
n = 1000 ;
for N=100:n
    dx = 1/N
    x= [0:dx:1-dx]
    I = (1/N) * sum(f(x+dx))
    L=[L,I]
end

plot(log(100:1:n) , log(abs(4-L))) ;
xtitle('Estimation erreur rectangle à droite avec singularité pour f(x)=x^-3/4',
 '$\log(N)$','$\log(I-I_N)$')
[a,b,r]=reglin(log(100:1:n) , log(abs(4-L))) ;
chaine=['Pente=', string(a)] ;
xstring(6,0, chaine)

function y=f4(x)
    y=sqrt(10*x).^(-1)
```

```
endfunction

function y=f5(x)
    y=((sqrt(1+(4*x).^(-1)))).*sqrt(20*sqrt(x)).^(-1)
endfunction

function y=g(x)
    y=1/2.*(x.^(3/2)).*(5-3*x)
endfunction

function y=h(x)
    y=((3/2)*(x.^(1/2)).*(5-3*x)-3*x.^(3/2))/2
endfunction

function y=f6(x)
    y=(sqrt(1+h(x).^2)).*((sqrt(20*g(x)))).^(-1)
endfunction

N1= 1000
dx1 = 1/N1
x1= [0:dx1:1-dx1]

I4 = (1/N1) * sum(f4(x1+dx1))
I5 = (1/N1) * sum(f5(x1+dx1))
I6 = (1/N1) * sum(f6(x1+dx1))

disp(N1,'Pour un nombre de points N=')
disp(I4,'Methode rectangle à droite : Temps de descente pour le toboggan doux')
disp(I5,'Methode rectangle à droite : Temps de descente pour le toboggan raide')
disp(I6,'Methode rectangle à droite : Temps de descente pour le toboggan doux')
```

III. Une méthode hybride pour prendre en compte la singularité en zéro de la fonction intégrée

Septième simulation : Illustration de la convergence de la méthode hybride (phénomène de saturation, balance de la compétition)

```
clear()

//Methode hybride

function y=f(x)
    y=(1+x.^2).*x.^(-3/4)
endfunction

//I = (h^(1-b)/(1-b)) * f(h*u)*(h*u)^b + ((1-h)/N)* sum(f(h+(x+dx/2)*(1-h)))
```

```
//
//
h = 10^(-1) ;
b = 3/4
u = (1-b)/(2-b)

L=[]
nmin = 100 ;
nmax = 1000 ;
for N=nmin:nmax
    dx = 1/N
    x= [0:dx:1-dx]
    I = (h^(1-b)/(1-b)) * f(h*u)*(h*u)^b + ((1-h)/N)* sum(f(h+(x+dx/2)*(1-h)))
    L=[L,I]
end

plot(log(nmin:1:nmax) , log(abs(40/9-L))) ;
xtitle('Estimation erreur methode hybride avec singularité pour h=10^-1', '$\log
(N)$','$\log(I-I_N)$')
//
//[a,b,r]=reglin(log(nmin:1:nmax) , log(abs(4-L))) ;
//chaine=['Pente=', string(a)] ;
//xstring(5.8,-0.4, chaine)
//
//h = 10^(-3) ;
//b = 3/4
//u = (1-b)/(2-b)
//
h2=10^(-3)
L1=[]
for N=nmin:nmax
    dx = 1/N
    x= [0:dx:1-dx]
    I = (h2^(1-b)/(1-b)) * f(h2*u)*(h2*u)^b + ((1-h2)/N)* sum(f(h2+(x+dx/2)*(1-h
2)))
    L1=[L1,I]
end


xclick()
close()

plot(log(nmin:1:nmax) , log(abs(40/9-L1))) ;
xtitle('Estimation erreur methode hybride avec singularité pour h=10^-3', '$\log
(N)$','$\log(I-I_N)$')
xclick()
plot(log(nmin:1:nmax), -2*(log(nmin:1:nmax)-log(nmin))+log(abs(40/9-L1(1))),'g')
 ;


L2=[]
for N=nmin:nmax
    h3=1/sqrt(N)
    dx = 1/N
    x= [0:dx:1-dx]
    I = (h3^(1-b)/(1-b)) * f(h3*u)*(h3*u)^b + ((1-h3)/N)* sum(f(h3+(x+dx/2)*(1-h
3)))
    L2=[L2,I]
```

```
end

xclick()
close

plot(log(nmin:1:nmax) , log(abs(40/9-L2))) ;
xtitle('Estimation erreur methode hybride avec singularité pour h=1/sqrt(N)',
 '$\log(N)$','$\log(I-I_N)$')
[a,b,r]=reglin(log(nmin:1:nmax) , log(abs(40/9-L2))) ;
chaine=['Pente=', string(a)] ;
xstring(5.8,-7, chaine)

function y=f1(x)
    y=sqrt(10*x).^(-1)
endfunction



b1 = 1/2
u1 = (1-b1)/(2-b1)

N=1000
h3=1/sqrt(N)
dx = 1/N
x= [0:dx:1-dx]
I = ((h3^(1-b1))/(1-b1)) * f1(h3*u1)*(h3*u1)^b1 + ((1-h3)/N)* sum(f1(h3+(x+dx/2)*
(1-h3)))


disp(N,'pour un nombre de points N=',)
disp(I,'Methode hybride : Temps de descente pour le toboggan plan')


function y=f2(x)
    y=((sqrt(1+(4*x).^(-1)))).*sqrt(20*sqrt(x)).^(-1)
endfunction



b1 = 3/4
u1 = (1-b1)/(2-b1)

N=100
h3=1/sqrt(N)
dx = 1/N
x= [0:dx:1-dx]
I = ((h3^(1-b1))/(1-b1)) * f2(h3*u1)*(h3*u1)^b1 + ((1-h3)/N)* sum(f2(h3+(x+dx/2)*
(1-h3)))



disp(I,'Methode hybride : Temps de descente pour le toboggan raide')


function y=g(x)
    y=1/2.*(x.^(3/2)).*(5-3*x)
endfunction
```

```
function y=h(x)
    y=((3/2)*(x.^(1/2)).*(5-3*x)-3*x.^(3/2))/2
endfunction

function y=f3(x)
    y=(sqrt(1+h(x).^2)).*((sqrt(20*g(x)))).^(-1)
endfunction



b1 = 3/4
u1 = (1-b1)/(2-b1)

N=1000
h3=1/sqrt(N)
dx = 1/N
x= [0:dx:1-dx]
I = ((h3^(1-b1))/(1-b1)) * f3(h3*u1)*(h3*u1)^b1 + ((1-h3)/N)* sum(f3(h3+(x+dx/2)*
(1-h3)))



disp(I,'Methode hybride : Temps de descente pour le toboggan doux')
```

I - Modélisation : descente du toboggan

II - Méthode classique (échec)

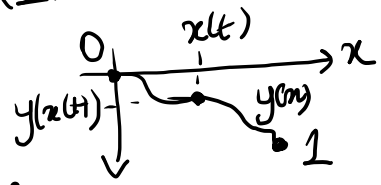
III - Méthode hybride

I- Modélisat°

1 Présentation

But: trouver la forme du toboggan tq tps de descente min.

Inconnue: fct $y : [0,1] \to \mathbb{R}$ tq $y(0)=0$ et $y(1)=1$



Hyp (descente)

1) bille ≃ pt matériel
$\vec{u}(t) = (x(t), y(x(t))$, à l'instant t
2) bille de masse m
3) bille part de $(0,0)$ sans vitesse initiale
4) la bille ne subit pas de frottements

□

2- Expression théorique du tps de descente

Def: $T(y)$ / $\vec{u}(T(y)) = (1,1)$

But: Comprendre T °/o à y

Sous-but: Expression de T °/o à y

$$T(y) = \int_0^{T(y)} dt$$

$$= \int_0^1 \frac{dz}{x'(x^{-1}(z))}$$

Pb: $x' \longleftrightarrow y$

chgt de variables
$z = x(t)$
de ss $x \in C^1$-difféo (par ex $x' \neq 0$)

$t=0$: $z = x(0) = 0$
$t = T(y)$: $z = x(T(y)) = 1$
$dz = x'(t)\,dt$

Apparté : trouver x'

Conservat° de l'e.m :

$E_{méca}(t) \overset{(*)}{=} E_{cin}(t) + E_{pot}(t) = E_{méca}(0)$

- $E_{cin}(t) = \frac{1}{2} \times m \times \left\| \frac{d\vec{u}}{dt} \right\|^2$
  $\frac{1}{2} \times m \times$ vitesse

$\vec{u} = \begin{pmatrix} x(t) \\ y^*(x(t)) \end{pmatrix}$

- $E_{pot}(t) = m \times g \times -y(x(t))$
  masse $\times$ $g \times$ hauteur

- $E_{méca}(0) = 0$

$(*) \Rightarrow x'(t) = \frac{\sqrt{2gy(x(t))}}{\sqrt{1 + y'(x(t))}}$

Conclusion :

1) si y ne s'annule qu'en 0,
$x' \neq 0$, $x \in C^1$-difféo $[0, T(y)] \to [0,1]$

2) Finalement,

$$T(y) = \int_0^1 \frac{\sqrt{1 + y'(z)^2}}{\sqrt{2g\, y(z)}}\, dz$$

$T : C^1 \longrightarrow$

$E_{frotte}(t) = -\alpha\, \ell(x(t))$

$\ell'(x(t)) = \sqrt{1 + \dot{y}(x(t))}$

II - Méthode classique (point milieu)

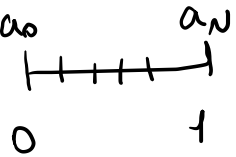
Méthode :

- Quadrature élémentaire :

$$\int_a^b f(t)dt \simeq (b-a) f\left(\frac{a+b}{2}\right)$$

- Quadrature composée :

$$\int_0^1 f(x)dx = \sum_{i=0}^{N-1} \underbrace{\int_{a_i}^{a_{i+1}} f(x)dx}_{\simeq (a_{i+1}-a_i) f\left(\frac{a_i+a_{i+1}}{2}\right)}$$



Théorème : Pour des fcts régulières (par ex $f \in C^2$), la méthode est d'ordre 2.

Théorème : Pour une fct $f(x) = x^{-\beta} + g(x)$ $(g \in C^2)$ $(0 < \beta < 1)$, la méthode est d'ordre $1-\beta$.

preuve :

$a_i = \frac{i}{N} = ih$

$$|I - I_N| \leq \underbrace{\left| \int_0^{1/N} f(x)dx - \frac{1}{N} f\left(\frac{1}{2N}\right) \right|}_{=\epsilon_1} + \underbrace{\left| \int_{1/N}^1 f(x)dx - \frac{1}{N} \sum_{i=1}^{N-1} f\left(\frac{2i+1}{2N}\right) \right|}_{=\epsilon_2}$$

▷ **Estimat° ϵ_1**

$$\epsilon_1 = \mathcal{O}(h^{1-\beta})$$

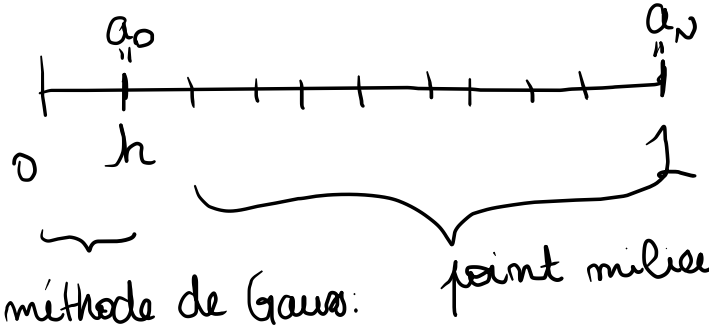
▷ **Estimat° ϵ_2**

$$\epsilon_2 \leq \frac{h^3}{24} \sum_{i=0}^{N-1} \underbrace{\|f''\|_{[a_i, a_{i+1}]}}_{a_i^{-\beta-2}} + \|g''\|_{[0,1]}$$

$$\leq \frac{h^2}{24} \|g''\| + \underbrace{\frac{h^3}{24} h^{-\beta-2}}_{=\mathcal{O}(h^{-\beta+1})} \sum_{i=0}^{+\infty} i^{-\beta-2}$$

III - Méthode hybride

Idée :



méthode de Gauss : poids absorbe la singularité

point milieu

* $\int_0^h f(x)\,dx \simeq \lambda_0 f(x_0)$

• où x_0 racine de π_1, 2ième poly U pour le poids $x^{-\beta}$

$$P_1 = x - hu \quad , \quad u = \frac{1-\beta}{2-\beta}$$

• $\lambda_0 = \int_0^h \underbrace{L_0(x)}_{=1}\, x^{-\beta}\,dx = \frac{h^{1-\beta}}{1-\beta}$

* $\int_h^1 f(x)\,dx \simeq \sum_{i=0}^{N-1} (a_{i+1}-a_i) f\left(\frac{a_i+a_{i+1}}{2}\right)$

$a_i = h + \frac{i}{N}(1-h)$

Théorème : Si $f(x) = x^{-\beta}$, $x \in [0,1]$.

Alors $\exists\, C_1, C_2 > 0$,

$$\left| \int_0^1 f(x)\,dx - I_{h,N} \right| \leq C_1 h^{-\beta+3} + C_2 \frac{h^{-\beta-1}}{N^2}.$$

compétition

min : $h = \frac{1}{\sqrt{N}}$

preuve :

• P

$$|E| \leq C_1 h^{-\beta+3} + C_2 \frac{h^{-\beta-1}}{N^2}$$

- Près de zéro :

$$\left| \int_0^h f(x)dx - \frac{h^{1-\beta}}{1-\beta} g(hu) \right| \leq \int_0^h |x-hu|^2 x^{-\beta} dx$$

$$\left(\int_0^1 (x-hu) x^{-\beta} dx \right) \leq h^{-\beta+3}$$

- En dehors de 0 :

$$\left| \int_h^1 f(x)dx - \frac{1-h}{N} \sum_{i=0}^{N-1} f\left(\frac{a_i + a_{i+1}}{2}\right) \right| \star$$

$$\leq \sum \underbrace{\| f'' \|_{[a_i, a_{i+1}]}}_{a_i^{-\beta-2}} \frac{(1-h)^3}{N^3}$$

$$\leq \frac{C}{N^2} \times \frac{1-h}{N} \sum_{i=0}^{N-1} \underbrace{\left(h + (1-h)\frac{i}{N}\right)^{-\beta-2}}$$

On (comparaison série / intégrale)

$$\frac{1-h}{N} \sum_{i=0}^{N-1} \left(h + (1-h)\frac{i}{N}\right)^{-\beta-2} \leq C \int_h^1 x^{-\beta-2} dx$$

$$\leq C h^{-\beta-1}$$

$$\star \leq C \frac{h^{-\beta-1}}{N^2}$$

□

Il existe une

$$\int_0^1 f(x)\,\omega(x)\,dx \simeq \sum_{i=0}^{N-1} \lambda_i f(x_i)$$

Il existe une unique méthode à $l+1$ points de degré $2l+1$.

$x_i \simeq$ racine de π_{l+1}

$$\lambda_i = \int_0^1 L_i(x)\,\omega(x)\,dx .$$