```
> # Let a rotation curve of a spiral galaxy. This program made the
calculus of the inverse of the matrix of forces in a maximal disk view
as a set of stars (or points) with usual symmetry. After, it realizes
the calculus of masses and of the surfacic density of the disk.

> k:=100;l:=k/2;Digits:=30:with(LinearAlgebra):
```

$$k := 100$$

$$l := 50$$

```
> #Tabular: distances and angles
> c:=seq(evalf(cos(Pi*n/l)),n=1..2*l):
d:=seq(evalf((i/(k))^2),i=1..k+1):
> # it is a good choice to simulate a galaxy by point masses
# the number of masses is k^2+1
# we calculate now the forces between the masses
> s:=proc(i,j)
local num,dist,F,n,u;
 u:=0;
 for n to 2*l do
       dist:=evalf(d[i]^2+d[j]^2-2*d[i]*d[j]*c[n]);
       num:=evalf(d[i]-d[j]*c[n]);
       if (n=2*l)and (i=j) then F:=0
       else F:=evalf(num/(dist^(3/2)))
       fi;
       u:=u+F
  od end:
A:=Matrix([seq([seq(s(ii,jj),jj=1..k),evalf(1/d[ii]^2)],ii=1..k),
[seq(2*l,jj=1..k),1]]):
> #with k=400, 50 mn. and with k=200, 300 seconds
> #The fundamental matrix A (n-body problem) is established.
#So the result (the inverse matrix) is:
> invA:=MatrixInverse(A);#plus 50 secondes pour k=200
```

$$invA := \begin{bmatrix} \text{101 x 101 Matrix} \\ \text{Data Type: sfloat} \\ \text{Storage: rectangular} \\ \text{Order: Fortran\_order} \end{bmatrix}$$

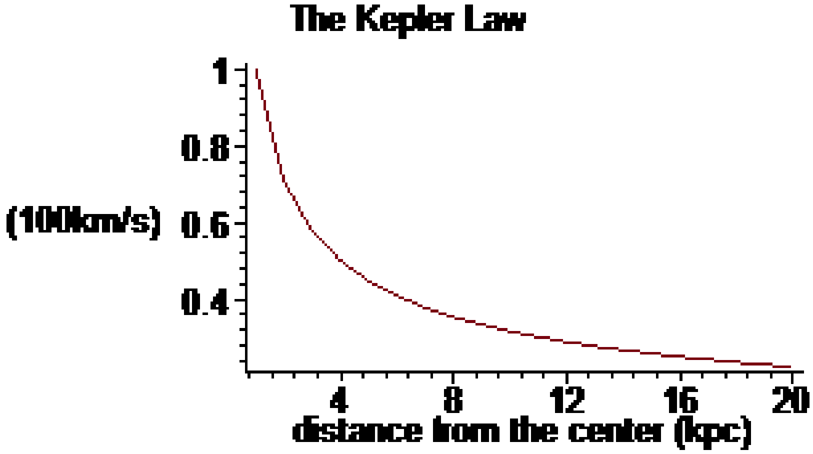
```
>
#precision:=max(seq(abs(evalf(evalm(A&*invA)[1,j],25)),j=2..k));#
it's long for k>50;for k=50 precision=.1e-19
> #a test
> X:=evalm(A&*invA):
> X[1,1];X[1,2],X[2,10];X[5,5];#well
```

1.000000000000000000000000006

$$-0.6\ 10^{-22},\ 0.4\ 10^{-20}$$

0.999999999999999999999999994

```
>
> #We could verify the third Kepler's law.
> fv:=proc(x) 1/sqrt(x) end:V:=1:R:=20:
> plot([[y,fv(y)]$y=1..R],title=`The Kepler Law`,
labels=[`distance from the center (kpc)`,`(100km/s)`]);
```




```
>  B:=Vector([seq(w*fv(d[i])^2/d[i],i=1..k),1]):
> C:=evalf(MatrixVectorMultiply(invA,B),20):
> wnul:=seq(evalf(solve(C[i]=0,w)),i=1..k+1):
n1:=0:n2:=0:
for j to k+1 do
 N:=seq(subs(w=wnul[j],C[i]),i=1..k+1):truc:=1:
 for i to k+1 do
   if N[i]<-10^(-8) then truc:=0 fi:
 od;
 if truc=1 then if n1=0 then n1:=j else n2:=j fi fi;
od:
if n1=0 then print(`il_y_a_des_masses_negatives`) else
 if wnul[n1]>wnul[n2] then wmax:=wnul[n1];wminim:=wnul[n2]
   else wmax:=wnul[n2];wminim:=wnul[n1] fi;
 fi:
wmax:=evalf(wmax,25);wminim:=evalf(wminim,25);
wm:=evalf((wmax+wminim)/2,18);
```

$$wmax := 1.000000000000000000622757$$

$$wminim := 1.$$

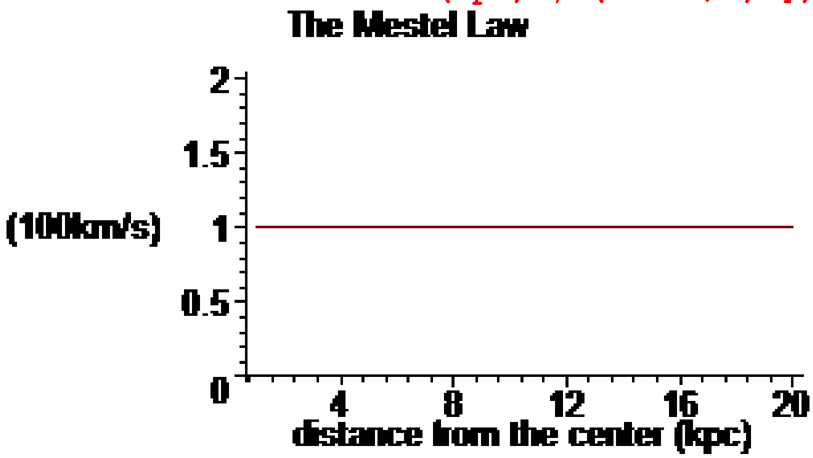
$$wm := 1.00000000000000000$$

```
> M:=seq(subs(w=wm,evalf(C[i],18)),i=1..k+1);
```

M := -0.1000000000000 10^{-19}, 0.1000000000000 10^{-19}, -0.2000000000000 10^{-20}, 0.1000000000000 10^{-20}, 0.2000000000000 10^{-20}, 0., 0.1000000000000 10^{-20}, 0., 0.1000000000000 10^{-20}, 0., 0., 0.1000000000000 10^{-20}, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.1000000000000 10^{-20}, 0., 0.1000000000000 10^{-20}, -0.1000000000000 10^{-20}, 0., 0.,

0., -0.1000000000000 10^{-20}, 0., 0., 0., 0., 0., 0., 0., 0., -0.1000000000000 10^{-20}, 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., -0.1000000000000 10^{-19}, 0., 0., 1.000000000000000000000000000000

```
> #very well, it's remarquable!
>
> #For a constant curve (Mestel):
> fv:=proc(x) 1 end:V:=1:R:=20:
> plot([[y,fv(y)]$y=1..R],title=`The Mestel Law`,
labels=[`distance from the center (kpc)`,`(100km/s)`]);
```




```
>  B:=Vector([seq(w*fv(d[i])^2/d[i],i=1..k),1]):
> C:=evalf(MatrixVectorMultiply(invA,B),20):
> wnul:=seq(evalf(solve(C[i]=0,w)),i=1..k+1):
n1:=0:n2:=0:
for j to k+1 do
 N:=seq(subs(w=wnul[j],C[i]),i=1..k+1):truc:=1:
 for i to k+1 do
   if N[i]<-10^(-15) then truc:=0 fi:
 od;
 if truc=1 then if n1=0 then n1:=j else n2:=j fi fi;
od:
if n1=0 then print(`il_y_a_des_masses_negatives`) else
 if wnul[n1]>wnul[n2] then wmax:=wnul[n1];wminim:=wnul[n2]
   else wmax:=wnul[n2];wminim:=wnul[n1] fi;
 fi:
wmax:=evalf(wmax,25);wminim:=evalf(wminim,25);
wm:=evalf((wmax+wminim)/2,15);
```

$$wmax := 1.548558023337066564356272$$

$$wminim := 1.547577169025207608551954$$

$$wm := 1.54806759618114$$

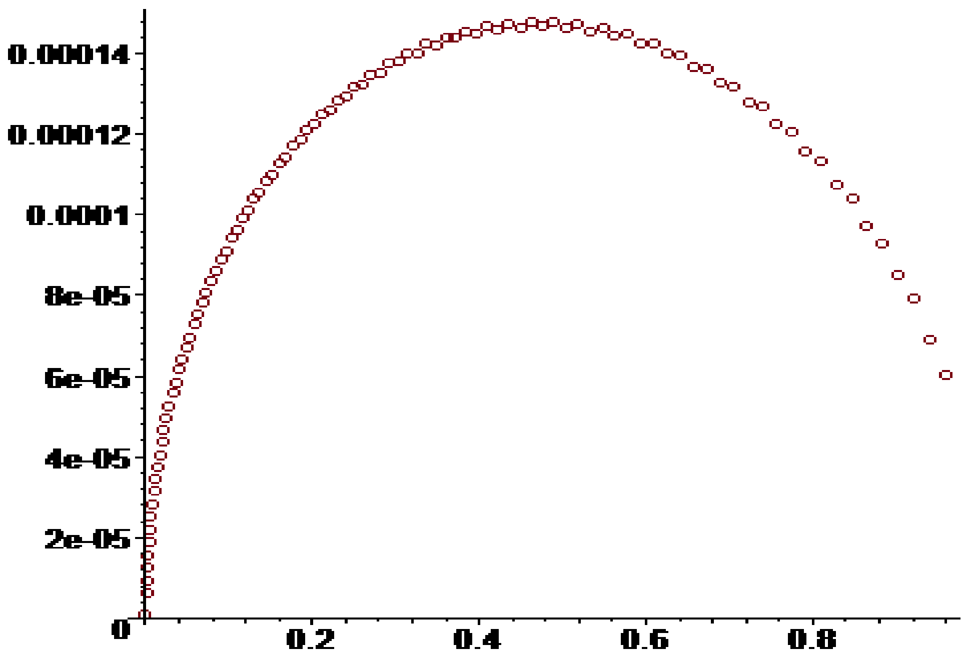
\>

```
M:=seq(subs(w=wm,evalf(C[i],15)),i=1..k):M[1..4];MM:=[seq([d[i],subs(w=wm,evalf(C[i],15))],i=1..k-2)]:
```

$0.10686197178094986212770\ 10^{-5}, 0.65563603831555670439570\ 10^{-5},$

$0.90082219593916625803042\ 10^{-5}, 0.0000127974854546157679906612$

```
> plot(MM,style=point,symbol=circle);
```



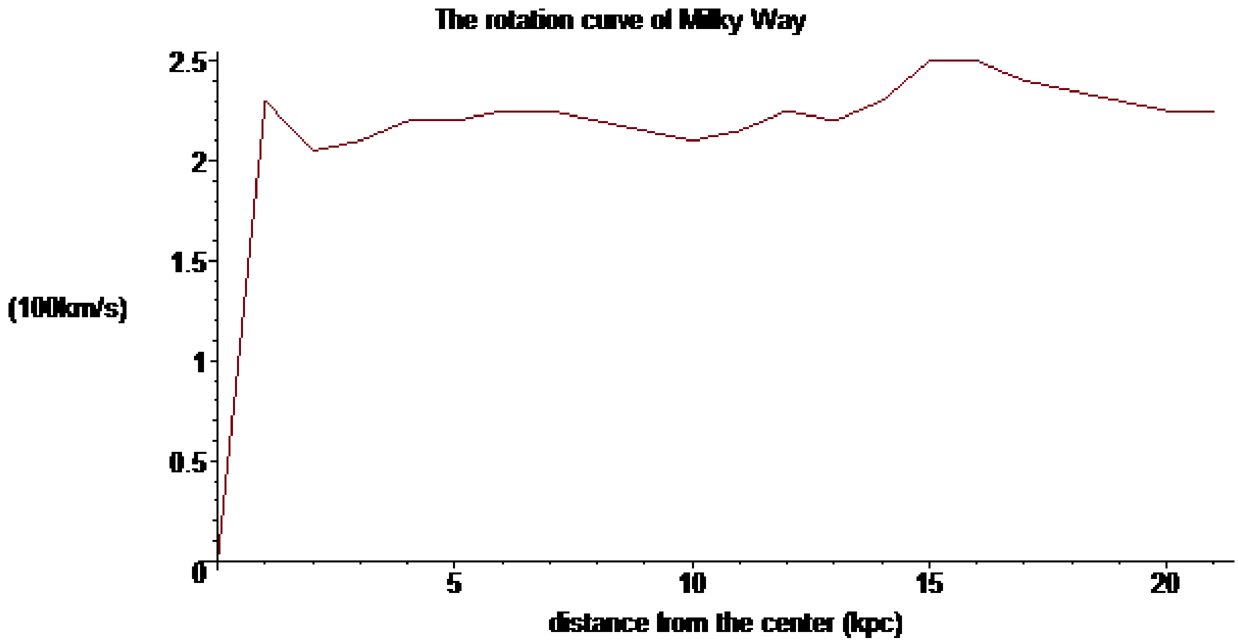

```
> #well, beautiful!
>
> #For Milky Way, with R=21 kpc and we spline the curve of velocities.
> R:=21;
f:=[0,2.3,2.05,2.1,2.2,2.2,2.25,2.25,2.2,2.15,2.1,2.15,2.25,2.2,2.3,2.5,2.5,2.4,2.35,2.30,2.25,2.25]:
a:=nops(f)-2;V:=f[a+2];
fv:=proc(x)

(f[trunc(a*x)+1]+(a*x-trunc(a*x))*(f[trunc(a*x)+2]-f[trunc(a*x)+1]))/V
end:
```

$$R := 21$$

$$a := 20$$

$$V := 2.25$$

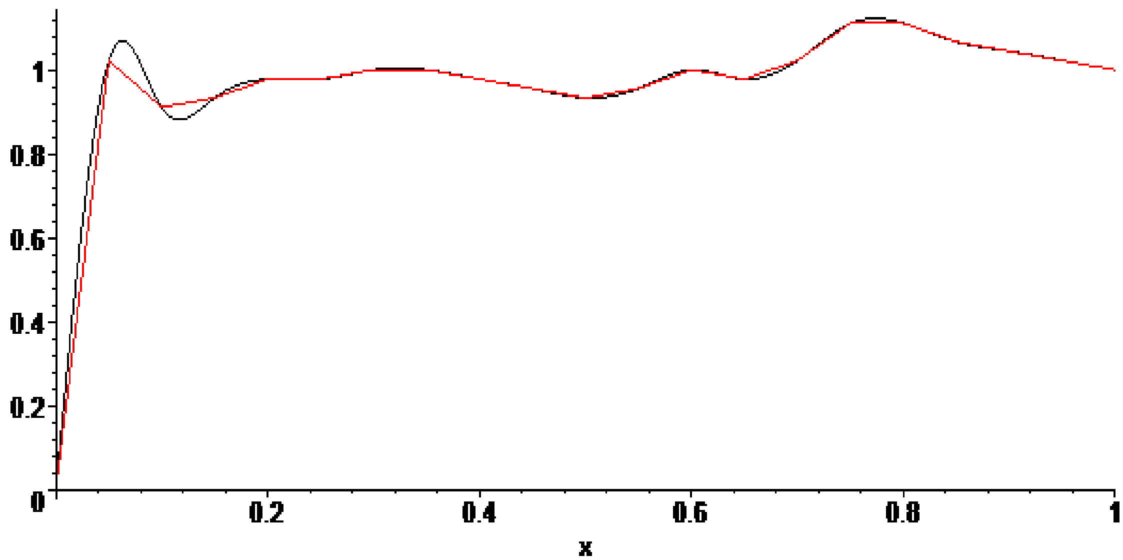
```
> plot([[y,f[y+1]]$y=0..21],title=`The rotation curve of Milky Way`,labels=[`distance from the center (kpc)`,`(100km/s)`]);
```




```
> f:=[0,2.3,2.05,2.1,2.2,2.2,2.25,2.25,2.2,2.15,2.1,2.15,2.25,2.2,
2.3,2.5,2.5,2.4,2.35,2.30,2.25,2.25]:nops(f);F:=[seq([(i-1)/20,f[
i]/V],i=1..21)]:
```

22

```
> with(CurveFitting):
> g:=x->Spline(F,x):
> plot({g(x),fv(x)},x=0..1,color=[red,black]);
```




```
> #The second member of the linear system, coming from observed
velocities.
   BB:=Vector([seq(w*g(d[i])^2/d[i],i=1..k),1]):
#w as the meaning of the inverse of the mass M of the galaxy
> #calculus of mass as function of w
C:=evalf(MatrixVectorMultiply(invA,BB),30):
```

```
> #search of w=wmin and w=wmax such that all the mass are >=0
wnul:=seq(evalf(solve(C[i]=0,w)),i=1..k+1):
n1:=0:n2:=0:
for j to k+1 do
 N:=seq(subs(w=wnul[j],C[i]),i=1..k+1):truc:=1:
 for i to k+1 do
   if N[i]<-10^(-5) then truc:=0 fi:
 od;
 if truc=1 then if n1=0 then n1:=j else n2:=j fi fi;
od:
if n1=0 then print(`il_y_a_des_masses_negatives`) else
 if wnul[n1]>wnul[n2] then wmax:=wnul[n1];wminim:=wnul[n2]
   else wmax:=wnul[n2];wminim:=wnul[n1] fi;
 fi:
wmax:=evalf(wmax,15);wminim:=evalf(wminim,15);
wm:=evalf((wmax+wminim)/2,15);wmax-wminim;
```

$$wmax := 1.48804031757250$$

$$wminim := 1.48803988768331$$

$$wm := 1.48804010262790$$

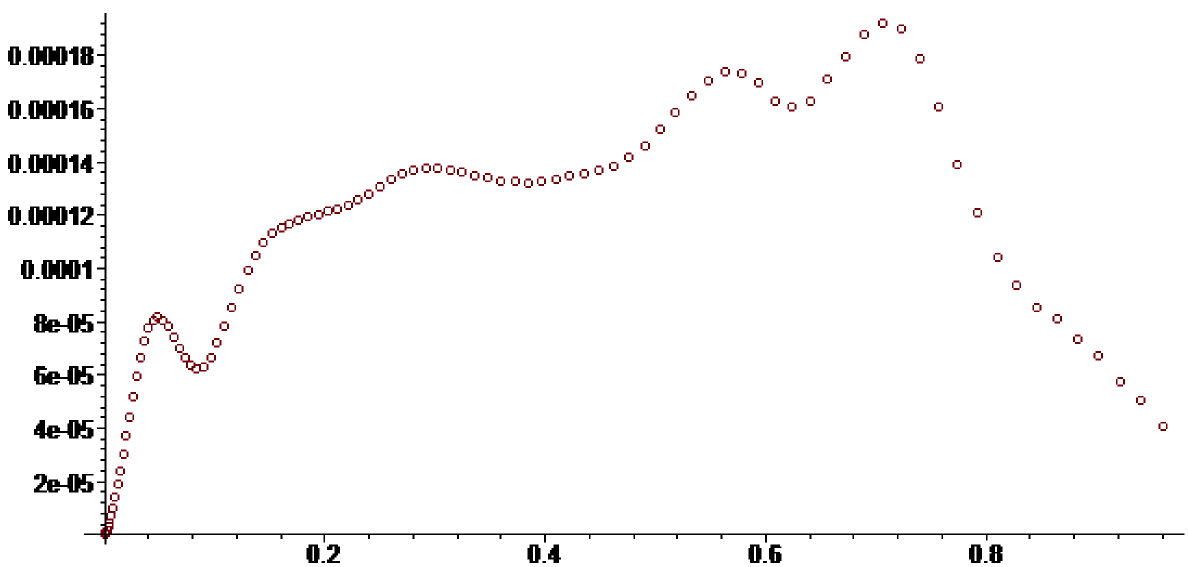
$$0.42988919\ 10^{-6}$$

```
>
M:=seq(subs(w=wm,evalf(C[i],15)),i=1..k):M[1..3];MM:=[seq([d[i],s
ubs(w=wm,evalf(C[i],15))],i=1..k-2)]:
```

$0.487094160163034235\ \ 10^{-9}, 0.382897052811240109 76\ \ 10^{-7},$

$0.235262995506385957 6434\ \ 10^{-6}$

```
> plot(MM,style=point,symbol=circle);
```



```
> MasseGalaxie:=evalf(0.23*10^10*V^2*R/wm,5):
print(`Masse_de_la_Galaxie`,MasseGalaxie,`en_Masses_Solaires`);
```

Masse_de_la_Galaxie, $0.16432\ 10^{12}$, *en_Masses_Solaires*

```
>
evalf(0.23*10^10*V^2*R/wmax,8),evalf(0.23*10^10*V^2*R/wminim,8);
```

$0.16432267\ 10^{12}, 0.16432271\ 10^{12}$

```
>
> #Now for and extended Milky Way, (always without spherical halo)
> R:=27;
f:=[0,2.3,2.05,2.1,2.2,2.2,2.25,2.25,2.2,2.15,2.1,2.15,2.25,2.2,
2.3,2.35,2.4,2.35,2.30,2.25,2.20,2.15,2.1,2.05,2.,2.,1.95,1.90]:
a:=nops(f)-2;
V:=f[a+2];
fv:=proc(x)

(f[trunc(a*x)+1]+(a*x-trunc(a*x))*(f[trunc(a*x)+2]-f[trunc(a*x)+1
]))/V
end:
```

$R := 27$

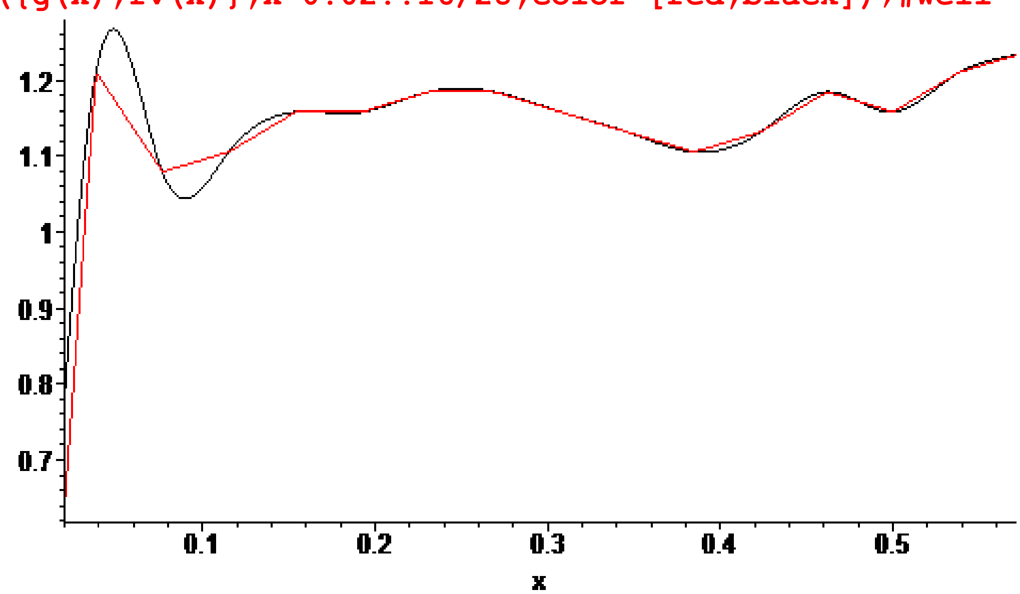
$a := 26$

$V := 1.90$

```
> nops(f);F:=[seq([(i-1)/26,f[i]/V],i=1..27)]:
```

28

```
> g:=x->Spline(F,x):
> plot({g(x),fv(x)},x=0.02..16/28,color=[red,black]);#well
```




```
> #The second member of the linear system, coming from observed
velocities.
```

```
   BB:=Vector([seq(w*g(d[i])^2/d[i],i=1..k),1]):
#w as the meaning of the inverse of the mass M of the galaxy
> #calculus of mass as function of w
C:=evalf(MatrixVectorMultiply(invA,BB),30):
> #search of w=wmin and w=wmax such that all the mass are >=0
wnul:=seq(evalf(solve(C[i]=0,w)),i=1..k+1):
n1:=0:n2:=0:
for j to k+1 do
 N:=seq(subs(w=wnul[j],C[i]),i=1..k+1):truc:=1:
 for i to k+1 do
   if N[i]<-10^(-5) then truc:=0 fi:
 od;
 if truc=1 then if n1=0 then n1:=j else n2:=j fi fi;
od:
if n1=0 then print(`il_y_a_des_masses_negatives`) else
 if wnul[n1]>wnul[n2] then wmax:=wnul[n1];wminim:=wnul[n2]
   else wmax:=wnul[n2];wminim:=wnul[n1] fi;
 fi:
wmax:=evalf(wmax,15);wminim:=evalf(wminim,15);
wm:=evalf((wmax+wminim)/2,15);wmax-wminim;
```

$$wmax := 1.26275816916535$$

$$wminim := 1.26275759935312$$

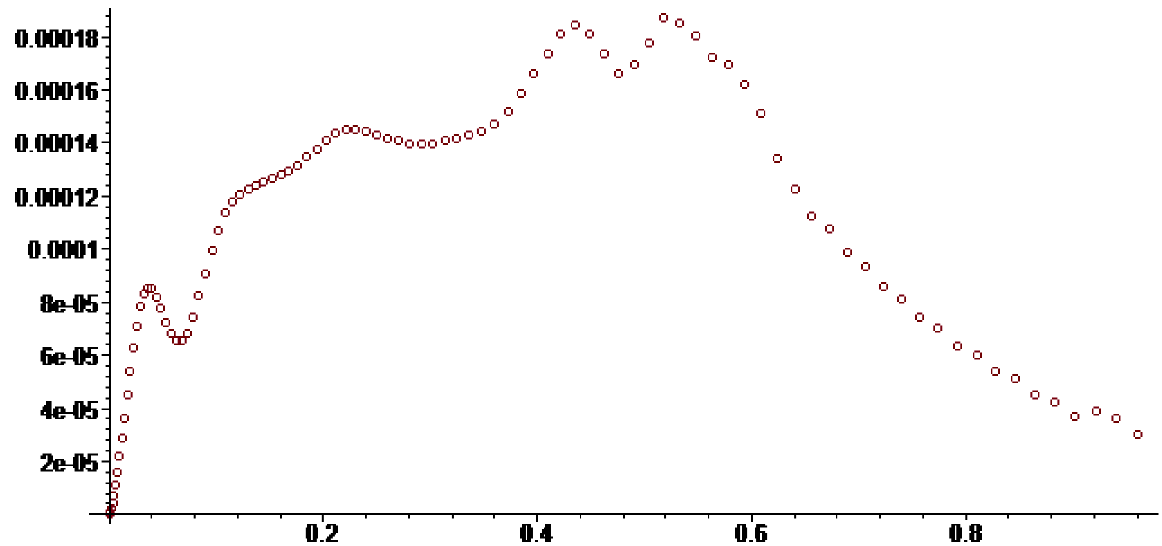
$$wm := 1.26275788425924$$

$$0.56981223\ 10^{-6}$$

```
>
M:=seq(subs(w=wm,evalf(C[i],15)),i=1..k):M[1..3];MM:=[seq([d[i],s
ubs(w=wm,evalf(C[i],15))],i=1..k-2)]:
```

$$0.760821489033722714170 0\ 10^{-9}, 0.598061789276472948321192\ 10^{-7},$$
$$0.367439492984531070742506 24\ 10^{-6}$$

```
> plot(MM,style=point,symbol=circle);
```




```
> MasseGalaxie:=evalf(0.23*10^10*V^2*R/wm,5):
print(`Masse_de_la_Galaxie`,MasseGalaxie,`en_Masses_Solaires`);
```

Masse_de_la_Galaxie, $0.17753\ 10^{12}$, *en_Masses_Solaires*

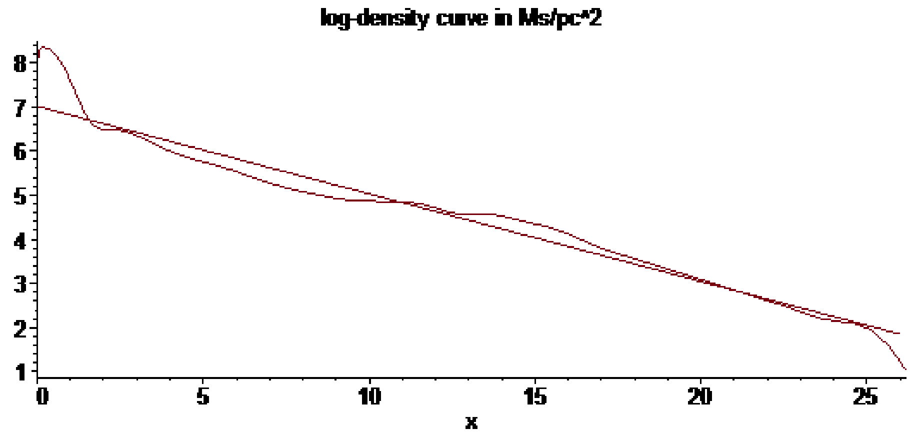
```
>
evalf(0.23*10^10*V^2*R/wmax,8),evalf(0.23*10^10*V^2*R/wminim,8);
```

$0.17753280\ 10^{12}, 0.17753289\ 10^{12}$

```
> Mgal:=MasseGalaxie:
> #mean surfacic density
rho:=[seq(Mgal*evalf(8*l*(M[i]+M[i+1])/Pi/((d[i+1]+d[i+2])^2
-(d[i]+d[i-1])^2)/(R*10^3)^2),i=2..(k-1))]:

> #log-density curve
i:='i':Rho:=[seq(ln(rho[i]),i=1..(k-2))]:
i:='i':courbelog:=plot([[R*(d[i]+d[i+1])/2,Rho[i]]$i=2..k-2],
title=`log-density curve in Ms/pc^2`):
> with(stats):Digits:=5:s:=trunc(k/6);
liste:=[[seq(R*(d[i]+d[i+1])/2,i=s..(k-5))],[seq(Rho[i+1],i=s-1..
(k-6))]]:
eqfit:=fit[leastsquare[[x,y]]](liste);
eqfonction:=unapply(rhs(eqfit),x):
courberegr:=plot(eqfonction(x),x=0..R-1):
with(plots):
display({courberegr,courbelog});Digits:=25:
```

$s := 16$

$eqfit := y = -0.19832\ x + 7.0080$



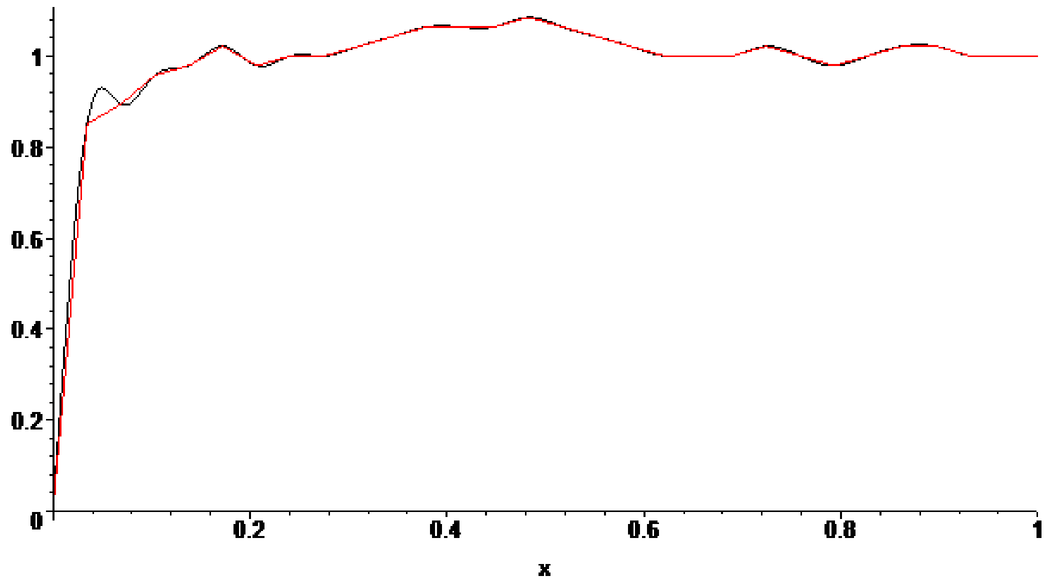

```
>
> #For Andromeda NE
R:=30;V:=2.35;
f:=[0,2.,2.1,2.25,2.3,2.4,2.3,2.35,2.35,2.4,2.45,
2.5,2.5,2.5,2.55,2.5,2.45,2.4,2.35,2.35,2.35,
2.4,2.35,2.3,2.35,2.4,2.4,2.35,2.35,2.35,2.35]:
a:=nops(f)-2;
fv:=proc(x)

(f[trunc(a*x)+1]+(a*x-trunc(a*x))*(f[trunc(a*x)+2]-f[trunc(a*x)+1
]))/V
end:
```

$$R := 30$$

$$V := 2.35$$

$$a := 29$$

```
> #with spline
> F:=[seq([(i-1)/29,f[i]/V],i=1..30)]:
> g:=x->Spline(F,x):
> plot({g(x),fv(x)},x=0..1,color=[red,black]);#well
```




```
> #The second member of the linear system, coming from observed
velocities.
   BB:=Vector([seq(w*g(d[i])^2/d[i],i=1..k),1]):
#w as the meaning of the inverse of the mass M of the galaxy
> #calculus of mass as function of w
C:=evalf(MatrixVectorMultiply(invA,BB),30):
> #search of w=wmin and w=wmax such that all the mass are >=0
wnul:=seq(evalf(solve(C[i]=0,w)),i=1..k+1):
n1:=0:n2:=0:
for j to k+1 do
 N:=seq(subs(w=wnul[j],C[i]),i=1..k+1):truc:=1:
 for i to k+1 do
   if N[i]<-10^(-5) then truc:=0 fi:
 od;
 if truc=1 then if n1=0 then n1:=j else n2:=j fi fi;
od:
if n1=0 then print(`il_y_a_des_masses_negatives`) else
 if wnul[n1]>wnul[n2] then wmax:=wnul[n1];wminim:=wnul[n2]
   else wmax:=wnul[n2];wminim:=wnul[n1] fi;
 fi:
wmax:=evalf(wmax,15);wminim:=evalf(wminim,15);
wm:=evalf((wmax+wminim)/2,15);wmax-wminim;
```

$$wmax := 1.51817067998510$$

$$wminim := 1.51817018976190$$

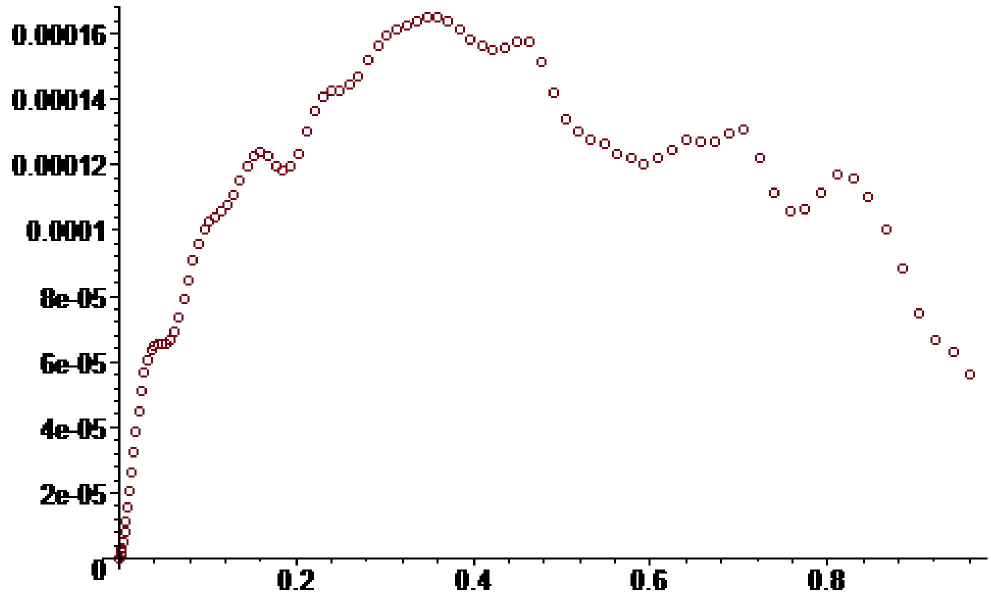
$$wm := 1.51817043487350$$

$$0.49022320\ 10^{-6}$$

```
>
M:=seq(subs(w=wm,evalf(C[i],15)),i=1..k):M[1..3];MM:=[seq([d[i],s
ubs(w=wm,evalf(C[i],15))],i=1..k-2)]:
```

$0.5444329393942434\,45\ \ 10^{-9}, 0.4279645908037062\,3722\ \ 10^{-7},$
$0.2629349610224073370741\ \ 10^{-6}$

```
> plot(MM,style=point,symbol=circle);
```




```
> MasseGalaxie:=evalf(0.23*10^10*V^2*R/wm,5):
print(`Masse_d'Andromède NE`,MasseGalaxie,`en_Masses_Solaires`);
```

Masse_d'Andromède SW, $0.25100\ 10^{12}$, *en_Masses_Solaires*

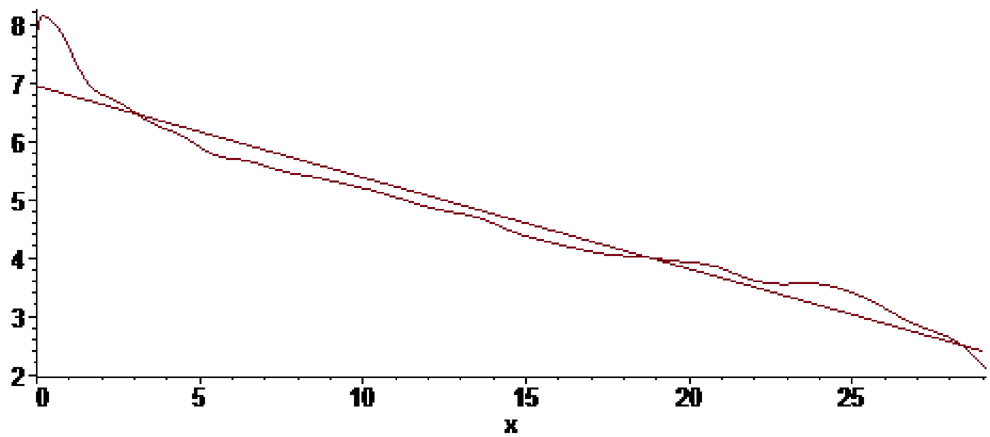
```
>
evalf(0.23*10^10*V^2*R/wmax,8),evalf(0.23*10^10*V^2*R/wminim,8);
```

$0.25099449\ \ 10^{12}, 0.25099458\ \ 10^{12}$

```
> Mgal:=MasseGalaxie:
> #mean surfacic density
rho:=[seq(Mgal*evalf(8*l*(M[i]+M[i+1])/Pi/((d[i+1]+d[i+2])^2
-(d[i]+d[i-1])^2)/(R*10^3)^2),i=2..(k-1))]:

> #log-density curve
i:='i':Rho:=[seq(ln(rho[i]),i=1..(k-2))]:
i:='i':courbelog:=plot([[R*(d[i]+d[i+1])/2,Rho[i]]$i=2..k-2],
title=`log-density curve in Ms/pc^2`):
> with(stats):Digits:=5:s:=trunc(k/6);
liste:=[[seq(R*(d[i]+d[i+1])/2,i=s..(k-5))],[seq(Rho[i+1],i=s-1..
(k-6))]]:
eqfit:=fit[leastsquare[[x,y]]](liste);
eqfonction:=unapply(rhs(eqfit),x):
courberegr:=plot(eqfonction(x),x=0..R-1):
with(plots):
display({courberegr,courbelog});Digits:=25:
```

$s := 16$

$$eqfit := y = -0.15648\,x + 6.9432$$



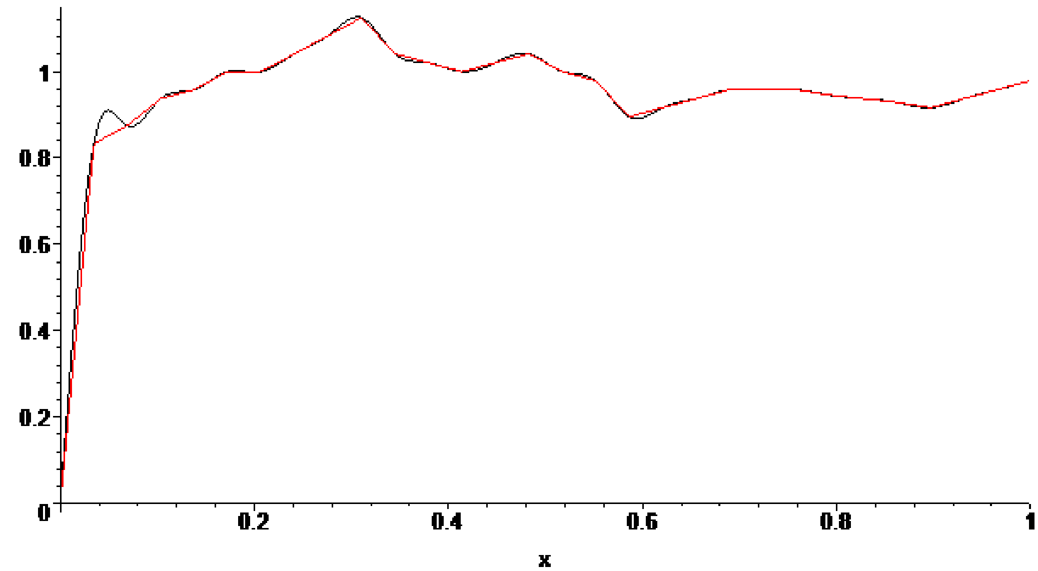

```
>
>
> #For Andromede SW :
#The observed rotation curve for Andromeda SW
R:=30;V:=2.4;
f:=[0,2.,2.1,2.25,2.3,2.4,2.4,2.5,2.6,2.7,2.5,
2.45,2.4,2.45,2.5,2.4,2.35,2.15,2.2,2.25,2.3,
2.3,2.3,2.27,2.25,2.23,2.2,2.25,2.3,2.35,2.4]:
a:=nops(f)-2;
fv:=proc(x)

(f[trunc(a*x)+1]+(a*x-trunc(a*x))*(f[trunc(a*x)+2]-f[trunc(a*x)+1
]))/V
end:
```

$$R := 30$$

$$V := 2.4$$

$$a := 29$$

```
>
> #with spline
> F:=[seq([(i-1)/29,f[i]/V],i=1..30)]:
> g:=x->Spline(F,x):
> plot({g(x),fv(x)},x=0..1,color=[red,black]);#well
```




```
> #The second member of the linear system, coming from observed
velocities.
   BB:=Vector([seq(w*g(d[i])^2/d[i],i=1..k),1]):
#w as the meaning of the inverse of the mass M of the galaxy
> #calculus of mass as function of w
C:=evalf(MatrixVectorMultiply(invA,BB),30):
> #search of w=wmin and w=wmax such that all the mass are >=0
wnul:=seq(evalf(solve(C[i]=0,w)),i=1..k+1):
n1:=0:n2:=0:
for j to k+1 do
 N:=seq(subs(w=wnul[j],C[i]),i=1..k+1):truc:=1:
 for i to k+1 do
   if N[i]<-10^(-5) then truc:=0 fi:
 od;
 if truc=1 then if n1=0 then n1:=j else n2:=j fi fi;
od:
if n1=0 then print(`il_y_a_des_masses_negatives`) else
 if wnul[n1]>wnul[n2] then wmax:=wnul[n1];wminim:=wnul[n2]
   else wmax:=wnul[n2];wminim:=wnul[n1] fi;
 fi:
wmax:=evalf(wmax,15);wminim:=evalf(wminim,15);
wm:=evalf((wmax+wminim)/2,15);wmax-wminim;
```

$$wmax := 1.66581549990064$$

$$wminim := 1.66581493199076$$

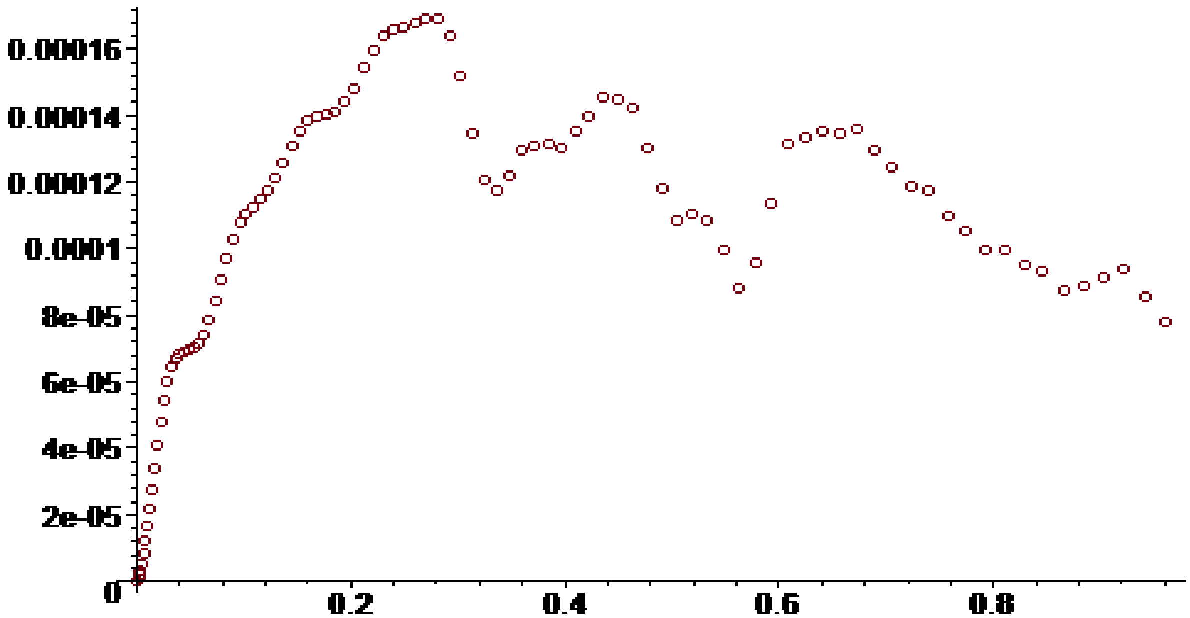
$$wm := 1.66581521594570$$

$$0.56790988\ 10^{-6}$$

```
>
M:=seq(subs(w=wm,evalf(C[i],15)),i=1..k):M[1..3];MM:=[seq([d[i],s
ubs(w=wm,evalf(C[i],15))],i=1..k-2)]:
```

$0.574809134325484802\ 10^{-9}, 0.45184259018604191514\ 10^{-7},$
$0.2776054369956981328435\ 10^{-6}$

```
> plot(MM,style=point,symbol=circle);
```




```
> #Question : where are the spiral arms ?
> MasseGalaxie:=evalf(0.23*10^10*V^2*R/wm,5):
print(`Masse_d'Andromède SW`,MasseGalaxie,`en_Masses_Solaires`);
```

Masse_d'Andromède SW, $0.23858\ 10^{12}$, *en_Masses_Solaires*

```
>
evalf(0.23*10^10*V^2*R/wmax,8),evalf(0.23*10^10*V^2*R/wminim,8);
```

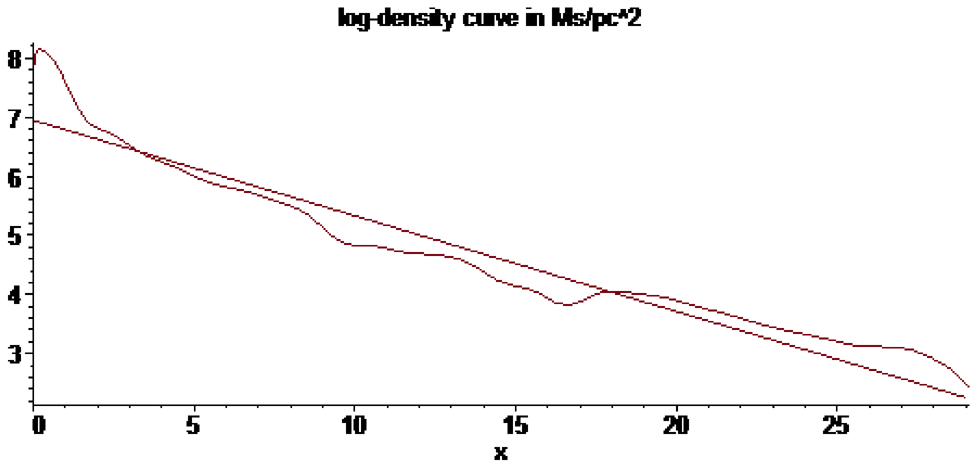
$0.23858585\ 10^{12}, 0.23858592\ 10^{12}$

```
> Mgal:=MasseGalaxie:
> #mean surfacic density
rho:=[seq(Mgal*evalf(8*l*(M[i]+M[i+1])/Pi/((d[i+1]+d[i+2])^2
-(d[i]+d[i-1])^2)/(R*10^3)^2),i=2..(k-1))]:

> #log-density curve
i:='i':Rho:=[seq(ln(rho[i]),i=1..(k-2))]:
i:='i':courbelog:=plot([[R*(d[i]+d[i+1])/2,Rho[i]]$i=2..k-2],
title=`log-density curve in Ms/pc^2`):
> with(stats):Digits:=5:s:=trunc(k/6);
liste:=[[seq(R*(d[i]+d[i+1])/2,i=s..(k-5))],[seq(Rho[i+1],i=s-1..
(k-6))]]:
eqfit:=fit[leastsquare[[x,y]]](liste);
eqfonction:=unapply(rhs(eqfit),x):
courberegr:=plot(eqfonction(x),x=0..R-1):
with(plots):
```

```
display({courberegr,courbelog});Digits:=25:
```

$$s := 16$$

$$eqfit := y = -0.16191\ x + 6.9493$$



>

>

> #Thus : no need of a spherical halo to explain the flatness problem. Moreover the planes of dwarf galaxy satellites could be understand.

> #This program is robust and fast! If the galaxy is now in a universe, it is not difficult to modify this program (a correction from 1% to 3% for the mass which is less than the uncertainties coming from velocities).

>