```
function vt = vperp(x,glr,glt,glp,sigrl,sigtl,sigpl,vrotl,L,gsr,gst,gsp,sigrs,sigts,sigps,vrots)

global cosb cosbl sinb L0 sinl vsp vsr vst vlr vlp vlt Ro


%------------------
% VITESSE DU SOLEIL
%------------------

vxsol=0; vzsol=0; vysol=-200e3;

%-----------------------------------------------------------
% CALCUL DES COORDONNEES PAR RAPPORT AU CENTRE DE LA GALAXIE
%-----------------------------------------------------------

[R, z, th] = toGC(x.*L);
r  = sqrt(R.*R+z.*z);

%-----------------------------------------------------------------------
% Conversion des g en vitesse. g varie entre 0 et 1, et le v correspondant
% a une distribution gaussienne de centre 0 et de dispertion sig
%-----------------------------------------------------------------------

vlr = sigrl.*erfinv(2.*glr-1);
vlt = sigtl.*erfinv(2.*glt-1);
vlp = sigpl.*erfinv(2.*glp-1)+vrotl;

%---------------------------------------------
% calcul des angles pour conversion en cartesien
%---------------------------------------------

sth = R./r;        cth = z./r;        sph = x.*L*cosb.*sinl./R;
cph = -(Ro-x.*L.*cosbl)./R;

%----------------------------------
% calcul de la vitesse en cartesien
%----------------------------------

vlx = vlr.*sth.*cph + vlt.*cth.*cph - vlp.*sph;
vly = vlr.*sth.*sph + vlt.*cth.*sph + vlp.*cph;
vlz = vlr.*cth - vlt.*sth;

%-------------------------------------------------------------------------
% on tient compte maintenant de la vitesse du Soleil et de la
source%------------------------------------------------------------------------
```

```
%---------------------------------------------
% Meme calcul mais pour la source cette fois-ci
%---------------------------------------------

[R, z, th] = toGC(L);  % utilisation des memes variables pour economiser
r  = sqrt(R.*R+z.*z);  % la memoire

vsr = sigrs.*erfinv(2.*gsr-1);
vst = sigts.*erfinv(2.*gst-1);
vsp = sigps.*erfinv(2.*gsp-1)+vrots;

sth = R./r;        cth = z./r;         sph = L*cosb.*sinl./R;
cph = -(Ro-L.*cosbl)./R;

vsx = vsr.*sth.*cph + vst.*cth.*cph - vsp.*sph;
vsy = vsr.*sth.*sph + vst.*cth.*sph + vsp.*cph;
vsz = vsr.*cth - vst.*sth;

%-----------------------------
% CALCUL DE LA VITESSE RELATIVE
%-----------------------------

vlx = vlx - (1-x).*vxsol + x.*vsx;
vly = vly - (1-x).*vysol + x.*vsy;
vlz = vlz - (1-x).*vzsol + x.*vsz;

%---------------------------------------------
% Vitesse projetée le long de la ligne de visée
%---------------------------------------------

vr = cosbl.*vlx + cosb.*sinl.*vly +sinb.*vlz;
v  = vlx.*vlx + vly.*vly + vlz.*vlz;

%---------------------------------------------------------
% Norme de la vitesse perpendiculairement à la ligne de visée
%---------------------------------------------------------

vt = sqrt(v-vr.*vr);
```

```
% probabilite de masse des lentilles.

function res = mPm(m)

res = m.*proba(m);
```

```
disp(['bigsim']);
load evenements.bigsim.txt
load para.bigsim.txt

%----------------------------------
% Constantes physiques (unites SI)
%----------------------------------
G=6.672e-11; pc=3.08567802e16;
kpc=pc*1e3;   Msol=1.989e30;
c=299792458; GMsol=1.32712497e20;

%--------------------------------------
%recuperation des evenements parametres
%--------------------------------------

n=para(1,1);
nbsimul=para(2,1);
tau=para(3,1);
Gammax=para(4,1);
uT=para(5,1);
AT=para(6,1);

%----------------------------------------
%recuperation des evenements selectionnes
%----------------------------------------

te=evenements(:,5);
te=te';

clear evenements;

histoB
```

```
%-----------------------------------
%-----------------------------------
% application du facteur d'efficacite
%-----------------------------------
%-----------------------------------

%----------------------
% Efficacités MACHO bulbe
%----------------------

tmachob = [ 0, 0.56, 0.7049, 0.8872, 1.116, 1.405, 1.769, 2.227, 2.803, 3.540, 4.456, 5.609,
7.060, 8.887, 11.18, 14.08, 17.72, 22.3, 28.08, 35.34, 44.49, 56, 70.49, 89.02, 112.05,
141.04 ];
tmachob = tmachob/2.;
stdeffmachob = [ 0, 0, 0, 0.00091407, 0.016453, 0.0338, 0.05758, 0.08135, 0.11791, 0.14899,
0.1718, 0.2065, 0.2404, 0.2815, 0.3071, 0.3436, 0.3839, 0.4076, 0.4095, 0.4369, 0.4716,
0.4561, 0.4570, 0.4360, 0.4140, 0.3884 ];
clpeffmachob = [ 0, 0.0054844, 0.001828, 0.01005, 0.08866, 0.1809, 0.2568, 0.3171, 0.4085,
0.4515, 0.5018, 0.5237, 0.5840, 0.6087, 0.6590, 0.6636, 0.7129, 0.7248, 0.8089, 0.7385,
0.7888, 0.7641, 0.6974, 0.6663, 0.6069, 0.5475 ];

%--------------------
% Choix de l'efficacite
%--------------------

eff = clpeffmachob;
teff = tmachob;
teffmax = max(teff);
teffmin = min(teff);

%----------------------------------------------------------------------
% Tirage d'un nombre aleatoire qui servira a decider si l'evt est garde ou non
%----------------------------------------------------------------------

ra = rand(1,length(te))*max(eff);

%-----------------------------------------------------------------------------------------
% Interpolation lineaire de l'efficacite pour determiner la probabilite qu'un evt a d'etre garde
%-----------------------------------------------------------------------------------------

i1 = find((te<=teffmax)&(te>=teffmin));
effsim = zeros(1,length(te));  % applique une efficacite nulle aux durees superieures et
inferieures
effsim(i1) = interp1(teff,eff,te(i1));
```

```
%-------------------------------------------------------------------------------------------------------------
------------
% compare le nombre aleatoire precedent a l'efficacite que l'on vient de calculer afin de
decider si l'evt est garde ou non
%-------------------------------------------------------------------------------------------------------------
------------

i1 = find(ra-effsim<=0);

clear ra

teobs=te(i1);

%-----------------------------------------------------
%comparaison des temps moyens ponderes par l'efficacite
%-----------------------------------------------------

sumTe=sum(teobs./effsim(i1));

clear teobs i1 effsim

fraction=0.4

for ii=1:100,

taux=ii/100
relecturetau

rap6(ii)=rapport;

end

fraction=0.5

for ii=1:100,

taux=ii/100
relecturetau

rap5(ii)=rapport;

end

fraction=0.6

for ii=1:100,
taux=ii/100
relecturetau

rap4(ii)=rapport;

end

fraction=0

for ii=1:100,

taux=ii/100
relecturetau

rap1(ii)=rapport

end



fid = fopen('tauCG.txt','w');
for ii=1:100,
rapo=[rap0(ii),rap4(ii),rap5(ii),rap6(ii)];
fprintf(fid,'%12.8f %12.8f  %12.8f  %12.8f  \n',rapo);
end

fclose(fid);

clear te

%-------
%bigsim2
%-------

disp(['bigsim2']);
load evenements.bigsim2.txt
load para.bigsim2.txt

%----------------------------------
% Constantes physiques (unites SI)
%----------------------------------
G=6.672e-11; pc=3.08567802e16;
kpc=pc*1e3;   Msol=1.989e30;
c=299792458; GMsol=1.32712497e20;
```

```
%-------------------------------------
%recuperation des evenements parametres
%-------------------------------------

n=para(1,1);
nbsimul=para(2,1);
tau=para(3,1);
Gammax=para(4,1);
uT=para(5,1);
AT=para(6,1);

%---------------------------------------
%recuperation des evenements selectionnes
%---------------------------------------

te=evenements(:,5);
te=te';

clear evenements;

%-----------------------------------
%-----------------------------------
% application du facteur d'efficacite
%-----------------------------------
%-----------------------------------

%------------------------
% Efficacités MACHO bulbe
%------------------------

tmachob = [ 0, 0.56, 0.7049, 0.8872, 1.116, 1.405, 1.769, 2.227, 2.803, 3.540, 4.456, 5.609,
7.060, 8.887, 11.18, 14.08, 17.72, 22.3, 28.08, 35.34, 44.49, 56, 70.49, 89.02, 112.05,
141.04 ];
tmachob = tmachob/2.;
stdeffmachob = [ 0, 0, 0, 0.00091407, 0.016453, 0.0338, 0.05758, 0.08135, 0.11791, 0.14899,
0.1718, 0.2065, 0.2404, 0.2815, 0.3071, 0.3436, 0.3839, 0.4076, 0.4095, 0.4369, 0.4716,
0.4561, 0.4570, 0.4360, 0.4140, 0.3884 ];
clpeffmachob = [ 0, 0.0054844, 0.001828, 0.01005, 0.08866, 0.1809, 0.2568, 0.3171, 0.4085,
0.4515, 0.5018, 0.5237, 0.5840, 0.6087, 0.6590, 0.6636, 0.7129, 0.7248, 0.8089, 0.7385,
0.7888, 0.7641, 0.6974, 0.6663, 0.6069, 0.5475 ];

%----------------------
% Choix de l'efficacite
%----------------------

eff = clpeffmachob;
```

```
teff = tmachob;
teffmax = max(teff);
teffmin = min(teff);

%-----------------------------------------------------------------------------
% Tirage d'un nombre aleatoire qui servira a decider si l'evt est garde ou non
%-----------------------------------------------------------------------------

ra = rand(1,length(te))*max(eff);

%-----------------------------------------------------------------------------------------------
% Interpolation lineaire de l'efficacite pour determiner la probabilite qu'un evt a d'etre garde
%-----------------------------------------------------------------------------------------------

i1 = find((te<=teffmax)&(te>=teffmin));
effsim = zeros(1,length(te));  % applique une efficacite nulle aux durees superieures et
inferieures
effsim(i1) = interp1(teff,eff,te(i1));

%--------------------------------------------------------------------------------------------------------------
------------
% compare le nombre aleatoire precedent a l'efficacite que l'on vient de calculer afin de
decider si l'evt est garde ou non
%--------------------------------------------------------------------------------------------------------------
------------

i1 = find(ra-effsim<=0);

clear ra

teobs=te(i1);

%------------------------------------------------------
%comparaison des temps moyens ponderes par l'efficacite
%------------------------------------------------------

sumTe=sum(teobs./effsim(i1));

clear teobs i1 effsim

fraction=0.5

for ii=1:10,

taux=ii/10
relecturetau
```

```
rap52(ii)=rapport;

end

clear te

%-------
%bigsim3
%-------

disp(['bigsim3']);
load evenements.bigsim3.txt
load para.bigsim3.txt

%---------------------------------
% Constantes physiques (unites SI)
%---------------------------------
G=6.672e-11; pc=3.08567802e16;
kpc=pc*1e3;   Msol=1.989e30;
c=299792458;GMsol=1.32712497e20;

%-------------------------------------
%recuperation des evenements parametres
%-------------------------------------

n=para(1,1);
nbsimul=para(2,1);
tau=para(3,1);
Gammax=para(4,1);
uT=para(5,1);
AT=para(6,1);

%---------------------------------------
%recuperation des evenements selectionnes
%---------------------------------------

te=evenements(:,5);
te=te';

clear evenements;

%-----------------------------------
%-----------------------------------
% application du facteur d'efficacite
%-----------------------------------
```

```
%-----------------------------------

%------------------------
% Efficacités MACHO bulbe
%------------------------

tmachob = [ 0, 0.56, 0.7049, 0.8872, 1.116, 1.405, 1.769, 2.227, 2.803, 3.540, 4.456, 5.609,
7.060, 8.887, 11.18, 14.08, 17.72, 22.3, 28.08, 35.34, 44.49, 56, 70.49, 89.02, 112.05,
141.04 ];
tmachob = tmachob/2.;
stdeffmachob = [ 0, 0, 0, 0.00091407, 0.016453, 0.0338, 0.05758, 0.08135, 0.11791, 0.14899,
0.1718, 0.2065, 0.2404, 0.2815, 0.3071, 0.3436, 0.3839, 0.4076, 0.4095, 0.4369, 0.4716,
0.4561, 0.4570, 0.4360, 0.4140, 0.3884 ];
clpeffmachob = [ 0, 0.0054844, 0.001828, 0.01005, 0.08866, 0.1809, 0.2568, 0.3171, 0.4085,
0.4515, 0.5018, 0.5237, 0.5840, 0.6087, 0.6590, 0.6636, 0.7129, 0.7248, 0.8089, 0.7385,
0.7888, 0.7641, 0.6974, 0.6663, 0.6069, 0.5475 ];

%---------------------
% Choix de l'efficacite
%---------------------

eff = clpeffmachob;
teff = tmachob;
teffmax = max(teff);
teffmin = min(teff);

%-------------------------------------------------------------------------
% Tirage d'un nombre aleatoire qui servira a decider si l'evt est garde ou non
%-------------------------------------------------------------------------

ra = rand(1,length(te))*max(eff);

%------------------------------------------------------------------------------------------
% Interpolation lineaire de l'efficacite pour determiner la probabilite qu'un evt a d'etre garde
%------------------------------------------------------------------------------------------

i1 = find((te<=teffmax)&(te>=teffmin));
effsim = zeros(1,length(te));  % applique une efficacite nulle aux durees superieures et
inferieures
effsim(i1) = interp1(teff,eff,te(i1));

%----------------------------------------------------------------------------------------------------
-----------
% compare le nombre aleatoire precedent a l'efficacite que l'on vient de calculer afin de
decider si l'evt est garde ou non
```

```
%------------------------------------------------------------------------------------------------------------------------------

i1 = find(ra-effsim<=0);

clear ra

teobs=te(i1);

%--------------------------------------------------------
%comparaison des temps moyens ponderes par l'efficacite
%--------------------------------------------------------

sumTe=sum(teobs./effsim(i1));

clear teobs i1 effsim


fraction=0.5

for ii=1:10,

taux=ii/10
relecturetau

rap53(ii)=rapport;

end

clear te



fid = fopen('tauCGbis.txt','w');
for ii=1:10,
rapobis=[rap5(ii),rap52(ii),rap53(ii)];
fprintf(fid,'%12.8f %12.8f  %12.8f   \n',rapobis);
end

fclose(fid);
```

```
figure(2)

tx=[0.01:0.01:1];

plot(tx,rap4,'b-')
hold on
plot(tx,rap5,'r:')
hold on
plot(tx,rap6,'k--')
hold on
plot(tx,rap1,'k-.')
hold on


tx=[0.1:0.1:1];
plot(tx,rap52,'bv')
hold on
plot(tx,rap53,'bo')
hold on

legend('k-.','f_B=0','b-','f_B=0.4','r:','f_B=0.5','k--','f_B=0.6','v','model 1','o','model 2')

xlabel('alpha_B')
ylabel('true optical depth / observed optical depth')

x=[0,0,0];
y=[0.6667,1,1.5];

hold on

plot(x,y,'r*')


axis([0.5 1.6 0 1])

print tauCG.ps

exit
```

```
function res = denssource(x)

global mmeande mmeandm mmeanbu mmeanh

[R, z, th] = toGC(x);

res=zeros(size(R));

%---------------
% source : bulbe
%---------------

res = res + rhobulbe(R,z,th)*mmeanbu;

%----------------------
% source : disque mince
%----------------------

res = res + rhodm(R,z,th)*mmeandm;

%----------------------
% source : disque epais
%----------------------

res = res + rhode(R,z,th)*mmeande;

%--------------
% source : halo
%--------------

%res = res + rhohalo(R,z,th)*mmeanh;
```