

Classes de Fit Paczynski

R. Ansari ansari@lal.in2p3.fr
É. Aubourg aubourg@hep.saclay.cea.fr
É. Lesquoy lesquoy@hep.saclay.cea.fr
C. Magneville cmv@hep.saclay.cea.fr

Table des matières

1	Fit de courbes de lumière micro-lensing (Paczynski)	3
	• Class InitPaczyn	4
	• Class Paczyn0	5
	• Class Paczyn	6
	• Class PaczynA	7
	• Class InvPaczyn	8
	• Class LogPaczyn	9
	• Class LogInvPaczyn	10
	• Class PaczynPC0	11
	• Class PaczynPC	12
	• Class PaczynBlend	13
	• Class PaczynBlend2	14
	• Class PaczynPar	15
	• Class PaczynParBl	16
	• Class PaczynPCMod	17
	• Class PaczynBlendMod	18
	• Class PaczynParBlMod	19
	• Class PaczynTF	20
	• Class BinLens	21
	• Class PaczynDiff	22
	• Class PaczynDiffMod	23
	• Class PaczynParallaxe	24
	• Class PaczynParallaxe3	26
	• Class PaczynSourceTailleFinie	27
	• Class PaczynParallaxeTF	28
	• Class PaczynSourceBin	29
	• Class PaczynSourceBinRot	30
	• Class SourceBinaireRotation	31
	• Class PaczynSourceBinPar	32

Chapitre 1

Fit de courbes de lumière micro-lensing (Paczynski)

Lib : Outils++

Include : paczfit.h

Pour initialiser les parametres du fit d'une fct de Paczynski.

InitPaczyn : :InitPaczyn(GeneralFunction *fun)

double InitPaczyn : :Init(GeneralFitData& data ,int niter ,double u01,double u02,int nu0st,double t01,double t02,int nt0st ,double tau1,double tau2,int ntaust,double f1,double f2,int nfst ,int l)

Pour initialiser les parametres des fonctions de Paczynski.

data : structure de donnees

niter : nombre d'iterations (<=0 -> 1)

u01,u02,nu0st : u0 varie de u01 a u02 avec nu0st pas.
pas = (pmax-pmin)/nstep, nstep force pair.

t01,t02,nt0st : idem pour t0

tau1,tau2,ntaust : idem pour tau

f1,f2,nfst : idem pour le flux ou la magnitude

Retourne: le Khi2 minimum, <0 si probleme.

Remplit ParMin : valeur trouvee pour l'initialisation des parametres

([0]=u0, [1]=t0, [2]=tau, [3]=flux ou mag)

Si pmax<=pmin ou nstep<=0, alors le parametre correspondant est fixe a la valeur pmin et il n'y a pas d'iteration. Les parametres u0 et tau sont contraints a etre >0 : si ils sont donnees <=0 la borne inferieur est mise a pas/100. La methode consiste a quadriller l'espace a 4 parametres dans les bornes indiquees et selon le pas prescrit. Pour chaque iteration, le point de la maille ayant le Khi2 minimum est retenu et l'iteration suivante change le bornes de variation entre cette valeur +/- pas, le pas est lui meme recalcule comme (pmax_New-pmin_New)/nstep.

double InitPaczyn : :GetParIni(int i) const

Pour recuperer l'initialisation du parametre i. ([0]=u0, [1]=t0, [2]=tau, [3]=f).

inline double* InitPaczyn : :GetParIni()

Pour recuperer le pointeur sur le tableau des initialisations des parametres.

Class

Paczyn0

Lib : Outils++

Include : paczfit.h

fonction de Paczynski * fond au temps t pour U0, T0, Tau, fond et derivees de la fonction par rapport aux parametres [0]=U0, [1]=T0, [2]=Tau, [3]=fond

Paczyn0 : :Paczyn0()

Lib : Outils++
Include : paczfit.h
heritage : InitPaczyn

fonction de Paczynski * fond au temps t pour U0, T0, Tau, fond et derivees de la fonction par rapport aux parametres [0]=U0, [1]=T0, [2]=Tau, [3]=fond

u0Seuil et tauSeuil sont des valeurs permettant d'empêcher l'apparition de divergence quand $u_0, \tau \rightarrow 0$

- Par défaut ils sont initialisés à zéro.
- Sinon la fonction fitée est:
$$A = \frac{(u^2+2)}{(u\sqrt{u^2+4})}$$
avec $u^2 = u_0^2 + u_0\text{Seuil} + (t-t_0)^2/(\tau^2+\tau\text{Seuil})$
- Attention, u0Seuil, tauSeuil ne sont pas élevés au carré pour permettre une initialisation avec valeur négative. Même remarque pour tauSeuil. Mais typiquement:
 $u_0\text{Seuil} = 1.e-6$
tauseuil = valeur petite par rapport à la plus petite taille de bosse voulue. Si une valeur mini est donnée pour tau, tauSeuil peut rester à zéro.

Paczyn : :Paczyn(double u0seuil,double tauseuil)

Class

PaczynA

Lib : Outils++
Include : paczfit.h
heritage : InitPaczyn

fonction de Paczynski * fond au temps t pour Amax, T0, Tau, fond et derivees de la fonction par rapport aux parametres [0]=Amax, [1]=T0, [2]=Tau, [3]=fond

PaczynA : :PaczynA()

Class

InvPaczyn

Lib : Outils++
Include : paczfit.h
heritage : InitPaczyn

fonction de inverse Paczynski : fond/Ampli au temps t pour U0, T0, Tau, fond et
derives de la fonction par rapport aux parametres [0]=U0, [1]=T0, [2]=Tau, [3]=fond

InvPaczyn : :InvPaczyn()

Class

LogPaczyn

Lib : Outils++
Include : paczfit.h
heritage : InitPaczyn

fonction de Paczynski en -magnitude c-a-d $2.5 \cdot \log_{10}(\text{Paczyn})$ au temps t pour U0, T0, Tau, magfond et derivees de la fonction par rapport aux parametres [0]=U0, [1]=T0, [2]=Tau, [3]=magfond

LogPaczyn : :LogPaczyn()

Class

LogInvPaczyn

Lib : Outils++
Include : paczfit.h
heritage : InitPaczyn

fonction de Paczynski a l'envers en -magnitude c-a-d $2.5 \cdot \log_{10}(\text{Paczyn})$ au temps t pour $U0$, $T0$, Tau , magfond et derivees de la fonction par rapport aux parametres $[0]=U0$, $[1]=T0$, $[2]=\text{Tau}$, $[3]=\text{magfond}$

LogInvPaczyn : :LogInvPaczyn()

Class

PaczynPC0

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond fit simultane de n couleurs (cf commentaire dans PaczynPC).

PaczynPC0 : :PaczynPC0(int ncol)

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond fit simultane de n couleurs

ncol = nombre de couleurs

(u0Seuil et tauSeuil cf. commentaire PacZyn)

La fonction est structuree F[c,t] :

ou c =0.,1.,... represente la couleur

et t represente le temps comme pour Paczin

[0]=U0, [1]=T0, [2]=Tau, [3]=fond coul 1, [4]=fond coul 2, etc...

PaczynPC : :PaczynPC(int ncol,double u0seuil,double tauseuil)

Class

PaczynBlend

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec blending

ncol = nombre de couleurs

La fonction est structuree F[c,t] :

ou c =0.,1.,... represente la couleur

et t represente le temps comme pour Paczin

[0]=U0, [1]=T0, [2]=Tau,

[3]=fond coul 1, [4]=coeff de blending coul 1,

[5]=fond coul 2, [6]=coeff de blending coul 2, etc...

PaczynBlend : :PaczynBlend(int ncol)

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec blending mais avec contrainte sur le Tau par rapport au blending.

ncol = nombre de couleurs

La fonction est structuree F[c,t] :

ou c =0.,1.,... represente la couleur

et t represente le temps comme pour Paczin

[0]=U0, [1]=T0, [2]=Tau (apparent),

[3]=fond coul 1, [4]=fonc coul2 ...

[3+ncol]=coeff de blending coul 1

[3+ncol+1]=coeff de blending coul 2 - coef coul 1

[3+ncol+2]=coeff de blending coul 3 - coef coul 1 ...

PaczynBlend2 : :PaczynBlend2(int ncol)

double PaczynBlend2 : :CalcTau(double tauObs, double f)

CalcTau fournit le vrai tau a partir du tau apparent et du coef blending de couleur 0.

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec effet de parallaxe

ncol = nombre de couleurs

La fonction est structuree F[c,t] :

ou c =0.,1.,... represente la couleur

et t represente le temps comme pour Paczin

Voir la these de Nathalie p71 pour les formules et la signification des parametres.

Parametres :

0 : U0, 1 : T0, 2 : Tau,

3 : du, 4 : theta, 5 : beta

6 : talpha

7->7+ncol-1 : flux non amplifies

talpha et beta sont a priori connus d'avance et fixes : talpha est defini par

$\alpha = \omega(t_0 - t_{\alpha})$

C'est donc le jour ou longitude ecliptique terre = longitude ecliptique ligne de visee.

Pour SMC et LMC, talpha = 4 aout 1997 = jour EROS 2772

Pour SMC, beta = 25.4 degres

Pour LMC, beta = 4.65 degres

PaczynPar : :PaczynPar(int ncol)

Class

PaczynParBl

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec effet de parallaxe et blending

ncol = nombre de couleurs

La fonction est structuree $F[c,t]$:

ou $c = 0., 1., \dots$ represente la couleur

et t represente le temps comme pour Paczin

Voir la these de Nathalie p71 pour les formules et la signification des parametres.

Parametres :

0 : U_0 , 1 : T_0 , 2 : τ ,

3 : du , 4 : θ , 5 : β

6 : α

7->7+2*ncol-1 : flux non amplifies, coef blending en alternance

α et β sont a priori connus d'avance et fixes : α est defini par

$\alpha = \omega(t_0 - \alpha)$

C'est donc le jour ou longitude ecliptique terre = longitude ecliptique ligne de visee - $\pi/2$.

Pour SMC et LMC, $\alpha = 4$ aout 1997 -365/4 = jour EROS 2681

Pour SMC, $\beta = 25.4$ degres

Pour LMC, $\beta = 4.65$ degres

PaczynParBl : :PaczynParBl(int ncol)

Class

PaczynPCMod

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec modulation sinusoidale.

ncol = nombre de couleurs

La fonction est structuree F[c,t] :

ou c =0.,1.,... represente la couleur

et t represente le temps comme pour Paczin

[0]=U0, [1]=T0, [2]=Tau, [3]=fond coul 1, [4]=fond coul 2, etc...

3+ncol : periode, 3+ncol+1 : phase a T=0,

3+ncol+2... : amplitudes (facteur *)

nombre de params : 5+2*ncol

PaczynPCMod : :PaczynPCMod(int ncol)

Class

PaczynBlendMod

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec blending et modulation sinusoidale

ncol = nombre de couleurs

La fonction est structuree F[c,t] :

ou c =0.,1.,... represente la couleur

et t represente le temps comme pour Paczin

[0]=U0, [1]=T0, [2]=Tau,

[3]=fond coul 1, [4]=coeff de blending coul 1,

[5]=fond coul 2, [6]=coeff de blending coul 2, etc...

3+2*ncol : periode (jours), 3+2*ncol+1 : phase a T=0,

3+2*ncol+2... : amplitudes (facteur *)

PaczynBlendMod : :PaczynBlendMod(int ncol, bool blmodul)

Class

PaczynParBIMod

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec effet de parallaxe, blending et modulation sinusoidale

ncol = nombre de couleurs

La fonction est structuree F[c,t] :

ou c = 0.,1.,... represente la couleur

et t represente le temps comme pour Paczin

Voir la these de Nathalie p71 pour les formules et la signification des parametres.

Parametres :

0 : U0, 1 : T0, 2 : Tau,

3 : du, 4 : theta, 5 : beta

6 : talpha

7->7+2*ncol-1 : flux non amplifies, coef blending en alternance

7+2*ncol : periode (jours), 7+2*ncol+1 : phase a T=0,

7+2*ncol+2+... : amplitude... (facteur *)

talpha et beta sont a priori connus d'avance et fixes : talpha est defini par

$\alpha = \omega(t_0 - \text{talpha})$

C'est donc le jour ou longitude ecliptique terre = longitude ecliptique ligne de visee - $\pi/2$.

Pour SMC et LMC, talpha = 4 aout 1997 - 365/4 = jour EROS 2681

Pour SMC, beta = 25.4 degres

Pour LMC, beta = 4.65 degres

PaczynParBIMod : :PaczynParBIMod(int ncol, bool blmodul)

Lib : Outils++
Include : paczfit.h

fonction de Paczynski * fond au temps t pour U0, T0, Tau, U et fond [0]=U0, [1]=T0, [2]=Tau, [3]=U, [4]=fond

Amplification paczynsky dans le cas d'une source non ponctuelle.

(integration analytique partie interieure et
integration numerique partie exterieure)

- Integration exterieure par methode de Gauss (Higher order terms).

u = parametre d'impact en rayon d'Einstein en fct du temps

U = rayon de l'etoile projete dans le plan de la lentille exprime

en rayon d'Einstein: $U = (x * R_s) / R_e$.

nseq = nombre de pas pour decomposer l'integrale

deg_gaus = degre de l'integration par la methode de Gauss

PaczynTF : :PaczynTF(int nseq, int deg_gaus)

void PaczynTF : :SetParInteg(int nseq, int deg_gaus)

Pour redefinir les parametres d'integration.

Lib : Outils++
Include : paczfit.h

Courbe de lumière pour déflecteur binaire, avec blending.

ncol = nombre de couleurs

La fonction est structurée $F[c,t]$:

ou $c = 0., 1., \dots$ représente la couleur

et t représente le temps comme pour Paczin

Paramètres :

0 : b (paramètre d'impact / centre de masse, en Re)

1 : t_0 (temps d'approche minimal du centre de masse)

2 : τ (durée de traversée d'un Re)

3 : a (séparation des étoiles, en Re)

4 : q (rapport de masses)

5 : θ (angle trajectoire / axe étoiles, radians)

6 : rayon étoile

7 : param limb darkening

8- \rightarrow 8+2*ncol-1 : flux non amplifiés, coef blending en alternance

BinLens : :BinLens(int ncol)

Lib : Outils++
Include : paczfit.h

fonction de Paczynski pour phot diff. fit simultane de n couleurs

ncol = nombre de couleurs

La fonction est structuree F[c,t] :

ou c =0.,1.,... represente la couleur

et t represente le temps comme pour Paczin

[0]=U0, [1]=T0, [2]=Tau,

[3]=flux coul 1, [4]=ligne de base coul 1

[5]=flux coul 2, [6]=ligne de base coul 2

PaczynDiff : :PaczynDiff(int ncol)

Class

PaczynDiffMod

Lib : Outils++
Include : paczfit.h

fonction de Paczynski pour phot diff. fit simultane de n couleurs et modulation sinusoidale

ncol = nombre de couleurs

La fonction est structuree F[c,t] :

ou c =0.,1.,... represente la couleur

et t represente le temps comme pour Paczin

[0]=U0, [1]=T0, [2]=Tau,

[3]=flux coul 1, [4]=ligne de base coul 1.

[5]=flux coul 2, [6]=ligne de base coul 2, etc.

3+2*ncol : periode (jours), 3+2*ncol+1 : phase a T=0,

3+2*ncol+2 : amplitude sur amplification (facteur *)

3+2*ncol+3 : amplitude sur base (facteur *)

PaczynDiffMod : :PaczynDiffMod(int ncol)

Lib : Outils++
Include : fdpaczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec effet de parallaxe
 l'excentricite de l'orbite terrestre est prise en compte d'apres ThM-hse M. Dominik
 (page 55) - F. Derue

F. Derue 1999.

Parametres :

```
[0]    U0 : parametre d'impact
[1]    t0 : instant du maximum
[2]    dt : duree caracteristique de l'evenement
[3]    du : a terre (1-x)/ Re, demi-grand axe de l'orbite
[4]          terrestre projetee dans le plan lentille
[5] epsilon : excentricite de l'orbite terrestre (va-
leur fixee)
[6]    tp : passage au perihelie (valeur fixee)
[7]    tv : passage au point vernal (valeur fixee)
[8]    beta : co-latitude ecliptique de l'etoile (valeur fixee)
[9]    lambda : longitude ecliptique de l'etoile (valeur fixee)
[10] theta : angle entre la trajectoire de la lentille
          dans le plan de la lentille et le demi-grand axe
          de l'orbite terrestre proje-
tee dans le plan de la lentille
[11+ncol] fond : ligne de base
```

NB : le lien avec la classe PaczynPar est le suivant epsilon est fixee a 0 $\alpha = tv + (2\pi/T) (\lambda + \pi/2)$

Géometrie :

```
Xd = dt cos(theta) - u0 sin(theta)
Yd = dt sin(theta) + u0 cos(theta)
os = coeff sin(xi)
os = coeff cos(xi) cos(beta)
p = (2Pi/T) ( t-tp)
= du (1-epsilon cos(Omegatp)
= Omegatp + 2 epsilon sin(Omegatp) -
  (lambda + (2Pi/T)(tv-tp) + epsilon sin((2Pi/T)(tv-tp)))
```

date de l'equinoxe de printemps (21/mars)

21/03/1996 -> 2273.

21/03/1997 -> 2638.

21/03/1998 -> 3002
passage de la terre au perihelie (03/01/1997 et 03/01/1998
03/01/1996 -> 2195.
03/01/1997 -> 2560.
03/01/1998 -> 2925

Donnees 2D :

x = couleur (0..ncol-1)

y = temps

PaczynParallaxe : :PaczynParallaxe(int ncol)

Constructeur. ncol = nombre de couleurs La fonction est structuree

F[c,t] : ou c =0.,1.,... represente la couleur et t represente le temps
comme pour Paczin

Lib : Outils++
Include : fdpaczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec effet de parallaxe
fonction utilisee dans la these de F. Derue

ncol = nombre de couleurs

La fonction est structuree F[c,t] : ou c =0.,1,... represente la couleur et t represente le temps comme pour Paczin

F. Derue - 1999

Parametres :

[0] U0 : parametre d'impact
[1] t0 : instant du maximum
[2] dt : duree caracteristique de l'evenement
[3] du : Rterre (1-x)/ Re, demi-grand axe de l'orbite
terrestre projetee dans le plan lentille
[4] tv : passage au point vernal (valeur fixee)
[5] beta : latitude ecliptique de l'etoile (valeur fixee)
[6] lambda : longitude ecliptique de l'etoile (valeur fixee)
[7] theta : un angle de rotation
[8+ncol] fond : ligne de base
[10+ncol] blcoef : coefficient de blending

B: le lien avec la classe PaczynPar est le suivant
epsilon est fixee a 0
 $\lambda = (2\pi/T)(\alpha - t_v) - (\pi/2)$
en imposant epsilon a 0 on retombe sur le fit PaczynPar

Geometrie :

$X_d = dt \cos(\theta) - u_0 \sin(\theta)$
 $Y_d = dt \sin(\theta) + u_0 \cos(\theta)$
os = coeff sin(xi)
os = coeff cos(xi)

PaczynParallaxe3(int ncol)

Constructeur

Class

PaczynSourceTailleFinie

Lib : Outils++
Include : fdpaczfit.h

Classe de Paczynski d'une source avec effet de taille finie et blending meme parametres que classique avec en plus R rayon de l'etoile projete dans le plan lentille en unites de rayon d'Einstein projete

Parametres

ncol = nombre de couleurs

La fonction est structuree $F[c,t]$:

ou $c = 0., 1., \dots$ represente la couleur

et t represente le temps comme pour Paczin

[0] u0: parametre d'imapct

[1] : instant du maximum

[2]: duree caracteristique

[3]: Rayon de la source (en unites de rayon d'Einstein projete)

[4+ncol]: ligne de base

[5+ncol] : coefficient de blending

F. Derue 1999

PaczynSourceTailleFinie(int ncol)

Constructeur

Class

PaczynParallaxeTF

Lib : Outils++
Include : fdpaczfit.h

fonction de Paczynski * fond fit simultane de n couleurs avec effet de parallaxe et taille fine ncol = nombre de couleurs La fonction est structuree F[c,t] : ou c =0.,1,... represente la couleur et t represente le temps comme pour Paczin

F. Derue 1999

Parametres :

[0] U0 : parametre d'impact
[1] t0 : instant du maximum
[2] dt : duree caracteristique de l'evenement
[3] U = rayon de la source projete
[4] du : $R_{\text{terre}} (1-x) / R_e$, demi-grand axe de l'orbite terrestre projete dans le plan lentille
[5] tv : passage au point vernal (valeur fixee)
[6] beta : latitude ecliptique de l'etoile (valeur fixee)
[7] lambda : longitude ecliptique de l'etoile (valeur fixee)
[8] theta : un angle de rotation
[9+ncol] fond : ligne de base
[11+ncol] blcoef : coefficient de blending

B: le lien avec la classe PaczynPar est le suivant
epsilon est fixee a 0
 $\lambda = (2\pi/T)(\alpha - t_v) - (\pi/2)$
en imposant epsilon a 0 on retombe sur le fit PaczynPar

Geometrie :

$X_d = dt \cos(\theta) - u_0 \sin(\theta)$
 $Y_d = dt \sin(\theta) + u_0 \cos(\theta)$
 $X_{los} = \text{coeff} \sin(\xi)$
 $Y_{los} = \text{coeff} \cos(\xi)$

PaczynParallaxeTF(int ncol)

Cosntructeur

Class

PaczynSourceBin

Lib : Outils++
Include : fdpaczfit.h

Classe de Paczynski d'une source binaire ou le mouvement de rotation des composantes est négligé soit un coefficient $\omega = L_2/(L_1+L_2)$ donnant le rapport des luminosités, la source binaire est décrite par deux valeurs de t_0 et u_{\min} , la durée de l'événement étant la même $Abs(u_1(t), u_2(t)) = (1-\omega)Ass(u_1) + \omega Ass(u_2)$

F. Derue 1999

Paramètres :

`ncol` = nombre de couleurs

La fonction est structurée `F[c,t]` :

ou `c = 0., 1., ...` représente la couleur

et `t` représente le temps comme pour Paczin

[0] : `u01` paramètre d'impact pour la première étoile à `t01`

[1] : `t01` instant du maximum pour la première étoile

[2] : `dt` durée caractéristique

[3] : `u02` paramètre d'impact pour la deuxième étoile à `t02`

[4] : `t02` instant du maximum pour la deuxième étoile

[5] : `omega` : fraction de luminosité de la seconde composante

[6+`ncol`] : ligne de base dans chaque couleur

PaczynSourceBin(int ncol)

Constructeur

Class

PaczynSourceBinRot

Lib : Outils++
Include : fdpaczfit.h

Classe de Paczynski d'une source binaire en rotation une composante est lumineuse, l'autre pas l'orbite des 2 étoiles est circulaire

F. Derue 1999

Parametres :

ncol = nombre de couleurs

La fonction est structuree $F[c,t]$:

ou $c = 0., 1., \dots$ represente la couleur

et t represente le temps comme pour Paczin

[0] u0: parametre d'impact

[1] t0: instant du maximum

[2] dt : duree caracteristique

[3] rho : demi-grand axe de l'orbite de la source en unites de rayon
d'Einstein projete

[4] alpha = angle de la direction du demi-
grand axe de l'orbite

de la source et la trajectoire de la lentille

[5]: perbin -> periode de la binaire

[6]: phasebin -> phase de la binaire a $t=t_0$

[7]: inc -> inclinaison de l'orbite dans le plan lentille

[8+ncol] : fond

PaczynSourceBinRot(int ncol)

Constructeur

Lib : Outils++
Include : fdpaczfit.h

Classe de fonction de Paczynski avec une source binaire la parametrisation est celle de M. Dominik (pg 52 de sa these) Les differents parametres sont les suivants :

F. Derue 1999

Parametres :

ncol = nombre de couleurs

La fonction est structuree $F[c,t]$:

ou $c = 0., 1., \dots$ represente la couleur

et t represente le temps comme pour Paczin

[0] u_0 = parametre d'impact minimal entre la lentille et le centre de gravite projete du systeme double dans le plan de la lentille

[1] t_0 = date deu minimum d'approche entre la lentille et le centre de gravite projete du systeme double dans le plan de la lentille

[2] dt = Temps de traversee d'un rayon d'Einstein

[3] θ = angle entre la trajectoire de la lentille et le demi-grand

axe de l'orbite projete de la source

NB : oui mais laquelle des deux sources !!!!!

[4] ω = fraction de luminosite de la seconde etoile

[5] m_1 = fraction de masse de la seconde etoile

[6] ρ = separation entree les deux composantes en unites de rayon d'Einstein projete

[7] η = angle entre la normale a l'orbite de la source et la ligne de visee

[8] γ = un angle de rotation γ

[9] P = la periode de la binaire

[10] e = excentricite de l'orbite de la binaire

[11] ϕ_0 = la phase a $t = t_0$

NB: dans le code on retranche aussitot $\pi/2$ a la cete phase pour etre a la fois compatible

avec les equations de Dominik et

avoir une initialisation identique de ϕ_0 dans cette classe et dans PaczynSourceBinRot (pas propre mais ca marche!)

$12+ncol$] : ligne de base

SourceBinaireRotation(int ncol)

Constructeur

Class

PaczynSourceBinPar

Lib : Outils++
Include : fdpaczfit.h

Classe de Paczynski d'une source binaire en rotation une composante est lumineuse, l'autre pas l'orbite des 2 etoiles est circulaire on prend en compte la parallaxe terrestre

F. Derue 1999.

Parametres :

ncol = nombre de couleurs

La fonction est structuree $F[c,t]$:

ou $c = 0., 1., \dots$ represente la couleur

et t represente le temps comme pour Paczin

[0] U_0 : parametre d'impact
[1] t_0 : instant du maximum
[2] dt : duree caracteristique de l'evenement
[8] du : $R_{terre} (1-x) / R_e$, demi-grand axe de l'orbite
terrestre projetee dans le plan lentille
[9] tv : passage au point vernal (valeur fixee)
[10] β : latitude ecliptique de l'etoile (valeur fixee)
[11] λ : longitude ecliptique de l'etoile (valeur fixee)
[12] θ : un angle de rotation
[13+ncol] fond : ligne de base

PaczynSourceBinPar(int ncol)

Constructeur

Index

BinLens, [21](#)

InitPaczyn, [4](#)

InvPaczyn, [8](#)

LogInvPaczyn, [10](#)

LogPaczyn, [9](#)

Paczyn, [6](#)

Paczyn0, [5](#)

PaczynA, [7](#)

PaczynBlend, [13](#)

PaczynBlend2, [14](#)

PaczynBlendMod, [18](#)

PaczynDiff, [22](#)

PaczynDiffMod, [23](#)

PaczynPar, [15](#)

PaczynParallaxe, [24](#)

PaczynParallaxe3, [26](#)

PaczynParallaxeTF, [28](#)

PaczynParBl, [16](#)

PaczynParBlMod, [19](#)

PaczynPC, [12](#)

PaczynPC0, [11](#)

PaczynPCMod, [17](#)

PaczynSourceBin, [29](#)

PaczynSourceBinPar, [32](#)

PaczynSourceBinRot, [30](#)

PaczynSourceTailleFinie, [27](#)

PaczynTF, [20](#)

SourceBinaireRotation, [31](#)