

# Une application du langage Maple à la physique : La simulation d'interférences lumineuses

## *Première partie : Cas de deux sources ponctuelles*

par Maurice DOURLENT  
Lycée Technique Régional Jean Dupuy  
1, rue Aristide Bergès  
65016 Tarbes

---

### **RÉSUMÉ**

*A l'aide du langage de manipulation symbolique MAPLE, on peut simuler en trois dimensions la figure théorique d'interférences non localisées produite par deux sources lumineuses ponctuelles synchrones. En jouant sur l'orientation de l'axe contenant les deux sources par rapport au plan de cette figure, on peut visualiser toutes les situations avec le même programme (franges rectilignes, franges annulaires et autres cas). En utilisant la programmation trichromatique de l'image vidéo, on peut aussi simuler les colorations variées obtenues en lumière polychromatique.*

### **INTRODUCTION**

Lorsque la source primaire n'est pas un faisceau laser, la réalisation expérimentale d'interférences lumineuses non localisées visibles simultanément par l'ensemble des élèves d'une classe est délicate pour plusieurs raisons, la principale étant la faible luminosité obtenue sur l'écran où l'on recueille la figure d'interférences. Ces expériences sont alors effectuées de préférence en travaux pratiques, sans utiliser un écran, mais en visant à travers un oculaire, lorsque chaque élève peut disposer d'un montage individuel nécessitant peu de lumière.

Avec un programme de calcul formel assez court, il est possible d'afficher en perspective à trois dimensions sur le moniteur d'un ordinateur, la figure d'interférences prévue par un calcul théorique, ce qui permet une confrontation immédiate avec l'expérience. Pour cela, il faut relier l'émission lumineuse d'une petite zone de l'écran du moniteur à l'éclairage théorique calculé pour la zone correspondante de l'espace physique d'observation des interférences.

Dans ce qui suit, on décrit et commente un programme utilisant le langage MAPLE, qui permet de simuler la figure d'interférences non localisées produite par deux sources ponctuelles synchrones.

## ORGANISATION DU PROGRAMME

### *Options prises pour le repérage dans l'espace géométrique*

- L'origine O du repère orthonormé cartésien Oxyz servant aux calculs est au milieu du segment joignant les deux sources.
- La droite qui les joint est repérée par son orientation angulaire (angles  $\theta$  et  $\varphi$  des coordonnées sphériques) par rapport à ces axes.
- L'écran où est recueillie la figure d'interférences est dans un plan perpendiculaire à l'axe Ox du repère.

### *Données initiales à introduire au choix de l'utilisateur*

- La distance a entre les deux sources, qui sert de référence pour les autres longueurs utilisées.
- Les angles  $\theta$  et  $\varphi$  (notés respectivement theta et phi dans les listings figurant en annexe).
- La distance OE de O au centre E de l'écran.
- Le nombre approximatif d'interfranges à visionner dans la configuration «Young», c'est-à-dire quand les sources appartiennent à l'un des deux axes autres que Ox (par exemple  $\theta = \varphi = \pi / 2$  quand les sources sont sur l'axe Oy).
- La longueur d'onde  $\lambda$  de la lumière émise par les sources (notée lambda dans le listing).
- La grille du tracé, qui conditionne sa résolution et ce que l'on pourrait appeler le rapport «*qualité-prix*», avec :

*qualité* = degré d'exactitude du dessin par rapport aux résultats attendus. Il s'améliore en augmentant le nombre de mailles de la grille.

*prix* = temps de calcul. Il augmente aussi avec le nombre de mailles de la grille.

***Ce réglage est très important, car un choix inadapté peut aboutir à une simulation sans rapport avec l'expérience*** (voir figure 2).

### ***Déroulement des calculs***

L'algorithme choisi est sommairement explicité dans les commentaires accompagnant le listing du programme de base figurant en fin de texte (voir Annexe). Après le calcul préliminaire de quelques paramètres à utiliser ultérieurement, une boucle permet de calculer la longueur optique de chaque trajet entre son point d'émission et le point M de l'écran où il y a interférence. On peut alors en déduire la différence de marche entre les deux trajets, puis l'éclairement relatif calculé en M. Cet éclairement est introduit comme variable informatique dans l'outil «color» du package «plot3d» de MAPLE, permettant ainsi la gestion de l'intensité lumineuse et de la couleur de l'écran du moniteur.

Pour terminer, une autre boucle sert à tracer deux points simulant chacun une source, un rayon particulier issu de chacune, qui interfère avec l'autre «en M».

*Remarque* : Dans la méthode de calcul de l'éclairement, il n'a pas été tenu compte de la décroissance de l'amplitude de l'onde en fonction de la distance de M à la source car, en optique, l'étendue transversale du champ d'interférences est suffisamment faible pour que l'on puisse négliger cette dépendance.

*Résultats* : Voir les figures 1 à 5, relatives à deux sources monochromatiques avec un affichage vidéo monochrome en noir et blanc, obtenu en donnant la même valeur aux trois variables R (rouge), G (vert), B (bleu) de l'outil «color». Dans ces figures, à l'exception de la figure 3, le repère n'est pas orthonormé.

## **COMMENTAIRES**

### ***A propos de l'algorithme***

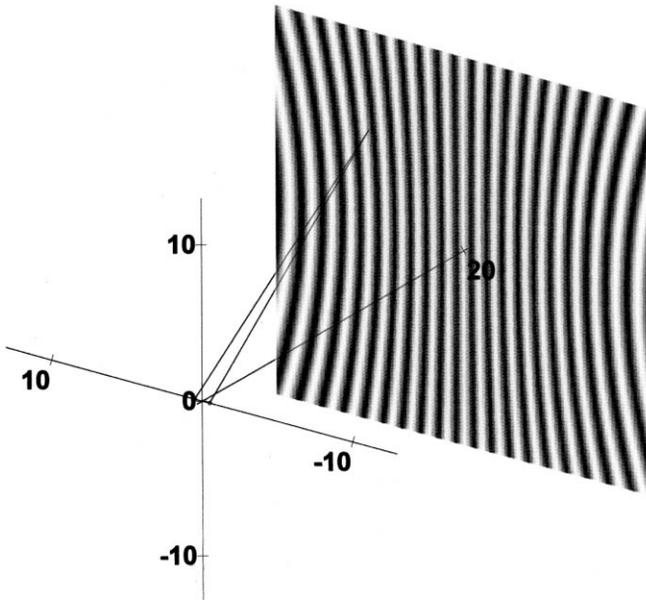
A titre indicatif, pour un choix de maillage de grille de  $(120 \times 120)$  points, il faut compter environ deux minutes pour obtenir l'image vidéo avec un processeur Pentium fonctionnant à 150 MHz. On peut diminuer ce délai en choisissant un algorithme ne comportant aucune «boucle de calcul», ce qui avait été fait au cours des premiers essais. Les avantages qui ont fait préférer la version qui figure en annexe sont les suivants :

- Quelques adjonctions permettent de l'adapter rapidement au cas où les sources ne sont pas monochromatiques et comportent :
  - soit un doublet de raies proches, pour simuler le brouillage des franges (voir figure 6 et variante n° 1 de l'annexe) ;
  - soit de la lumière polychromatique, modélisée par trois fréquences, une pour chaque faisceau électronique du moniteur. On donne alors une valeur calculée spécifique à chaque variable R (rouge), G (vert), B (bleu) de l'outil «color». Les résultats obtenus sont à la figure 7 et le programme est la variante n° 2 de l'annexe.

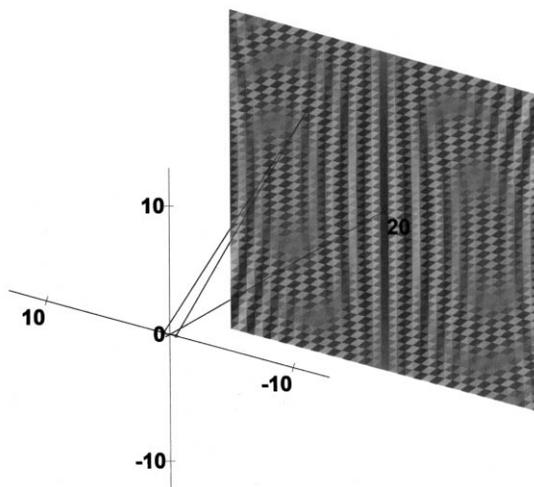
- Elle se prête à une extension au cas d'ondes multiples à condition d'adapter la formule de calcul de l'éclairement relatif au point courant de l'écran. Cette extension est décrite dans la deuxième partie.

### *A propos des résultats*

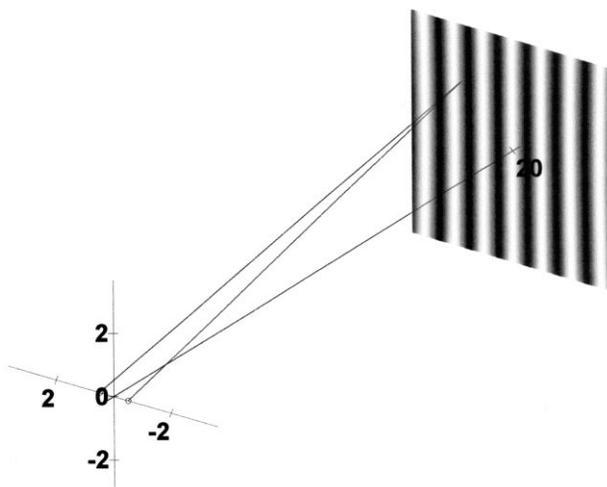
Il va de soi qu'une simulation ne doit en aucun cas se substituer à la véritable expérience, lorsque celle-ci est réalisable, mais elle peut servir d'outil pédagogique pour aider à transmettre certaines idées à l'auditoire. Par exemple, la figure 1 montre bien que pour la configuration «Young», les franges théoriques ne sont pas rectilignes, sauf près du centre de l'écran (voir figure 3) ou selon les ordres de grandeurs relatifs entre  $\lambda$ ,  $a$  et  $OE$  (voir figure 6). On peut ainsi plus facilement faire comprendre que dans ce cas, on ne modifie pas la figure d'interférences en remplaçant les trous sources par des fentes parallèles à  $Oz$ , dans le but d'améliorer la luminosité grâce à une augmentation du flux lumineux total émis par les sources.



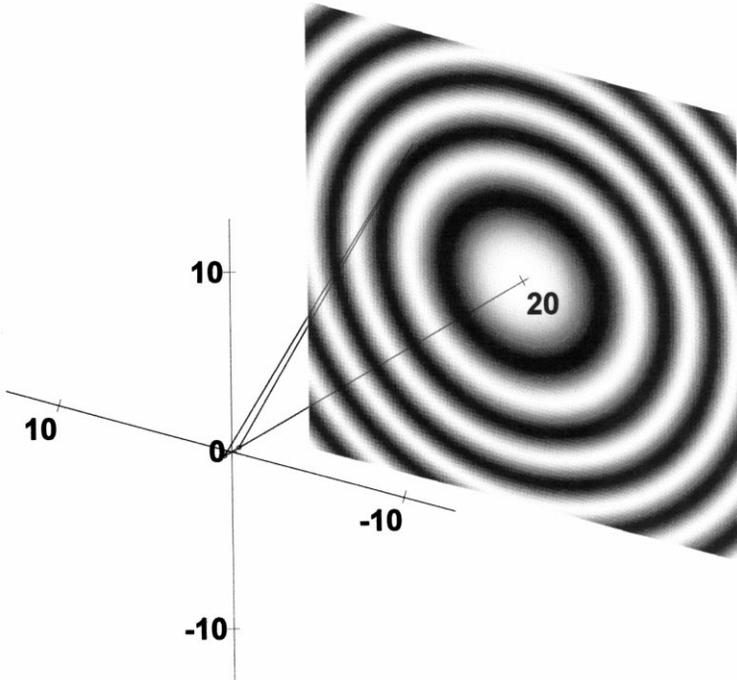
**Figure 1** : Franges d'Young monochromatiques  
Données initiales : celles du listing du programme de base.  
Grille choisie pour le dessin : (120×120) points.



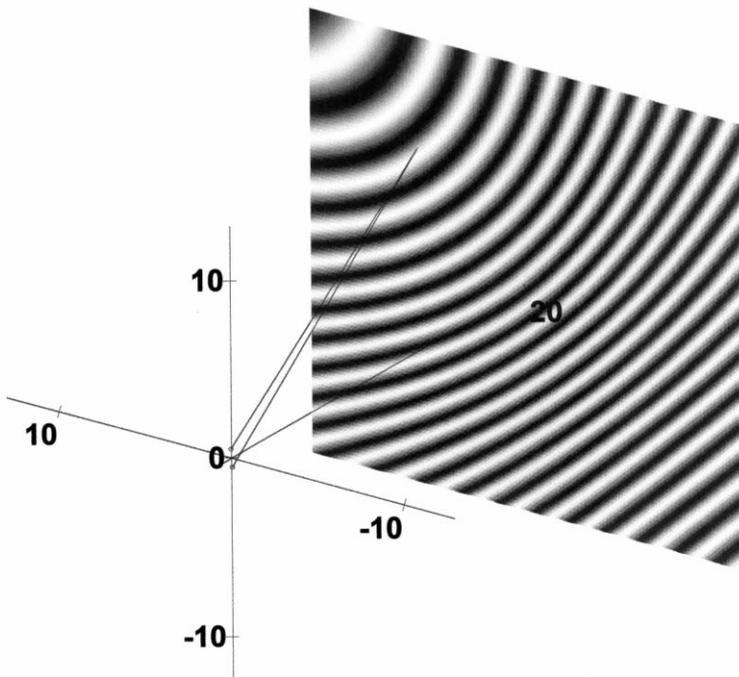
**Figure 2 :** Franges d'Young monochromatiques  
Mêmes données initiales que pour la figure 1  
sauf : grille choisie pour le tracé :  $(40 \times 40)$  points.



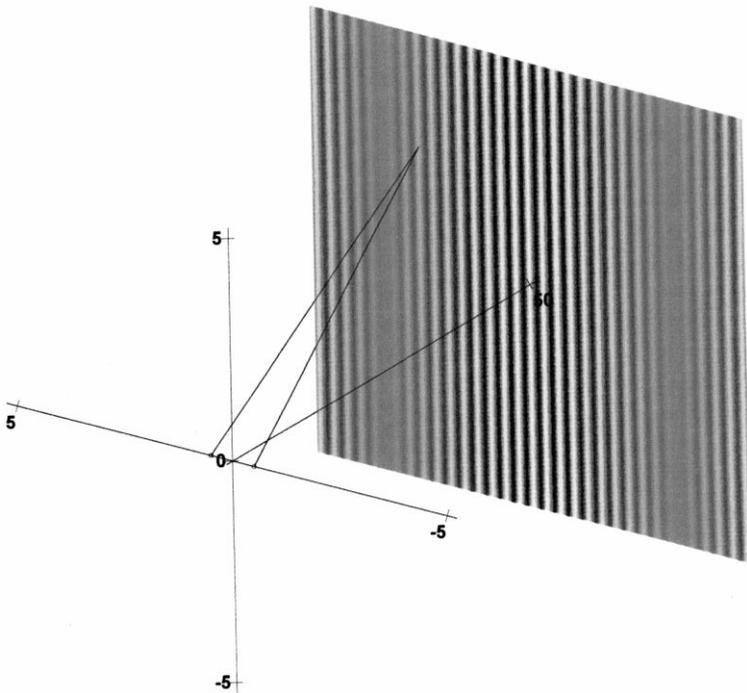
**Figure 3 :** Franges d'Young monochromatiques  
Données initiales : celles du listing du programme de base,  
sauf : nombre approché d'interfranges pour le dessin = 7.



**Figure 4** : Franges annulaires monochromatiques  
Mêmes données initiales que pour la figure 1,  
sauf :  $\phi = 0$ .



**Figure 5** : Interférences de deux ondes monochromatiques  
Données initiales : celles du listing du logiciel de base,  
Sauf :  $\theta = \arccos(1/\sqrt{3})$  et  $\phi = \pi/4$   
ce qui correspond à une même inclinaison de la ligne joignant les deux sources  
par rapport à chacun des trois axes du repère cartésien.



**Figure 6** : Franges d'Young pour deux ondes dichromatiques de fréquences proches (doublet)  
Données initiales : voir listing  
grille choisie pour le dessin :  $(200 \times 200)$  points.

## *Annexe*

---

Les commentaires sont en italique.

### PROGRAMME DE BASE INTERFÉRENCES À DEUX ONDES MONOCHROMATIQUES

> with(plots,spacecurve,display):

> with(linalg):

#### *Données initiales au choix de l'utilisateur*

> a:=1:	<i># distance entre deux sources voisines</i>
> theta:= Pi/2:phi:= Pi/2:	<i># orientation ligne contenant les sources</i>
> OE:=20*a:	<i># distance entre origine O et centre E de l'écran</i>
> NbF:=25:	<i># nombre approché de franges à dessiner en configuration Young</i>
> lambda:=a/20:	<i># longueur d'onde émise par les sources</i>
> Grille:=120:	<i># réglage de la finesse du maillage du dessin</i>

#### *Calculs préliminaires*

> k:=2*Pi/lambda:	<i># norme vecteur d'onde</i>
> Champ:=NbF*lambda*OE/(2*a):	<i># étendue zone de tracé des interférences</i>
> PAS:=vector([a*sin(theta)*cos(phi), a*sin(theta)*sin(phi),a*cos(theta)]):	<i># vecteur directeur de la ligne joignant les sources</i>

**Calculs pour le tracé des interférences**

> M:=vector([OE,y,z]):	# coordonnées du point courant M de l'écran
> for i from 1 to 2 do > facteur:=i-3/2: > S[i]:=vector([PAS[1]*facteur, PAS[2]*facteur,PAS[3]*facteur]): > V:=evalm(M-S[i]): > L[i]:=sqrt(V[1]^2+V[2]^2+V[3]^2): > od:	# calcul trajet L[i] de chaque rayon
> EclR:= (1+cos(k*(L[2]-L[1])))/2:	# éclairement relatif en M
> Franges:=plot3d([OE,y,z],y=-Champ..Champ,z=-Champ..Champ, grid=[Grille,Grille],color=COLOR(RGB,EclR,EclR,EclR), orientation=[-145,65],style=patchngrid,axes=normal, tickmarks=[2,3,3],axesfont=[HELVETICA,BOLD,6]):	

**Calculs pour le tracé de deux rayons particuliers**

> P:=vector([OE,Champ/2,Champ/2]):	# définition du point particulier de l'écran
> for i from 1 to 2 do > SP:=evalm(P-S[i]): > r[i]:=spacecurve([S[i][1]+SP[1]*u, S[i][2]+SP[2]*u,S[i][3]+SP[3]*u], u=0..1,numpoints=2,color=BLACK): > od:	# équation paramétrique de chaque r[i]
> Rayons:=convert(r,set):	
> Sources:=spacecurve([PAS[1]*(u-3/2), PAS[2]*(u-3/2),PAS[3]*(u-3/2)], u=1..2,numpoints=2,style=POINT, symbol=CIRCLE,color=BLACK):	# point de départ de r[i]

**Dessin**

> display(Sources,Rayons,Franges);

## Variante n° 1 :

### interférences de deux ondes dichromatiques de fréquences proches (doublet)

---

$$(\lambda_1 = 0,0065 \cdot a \text{ et } \lambda_2 = 0,0062 \cdot a)$$

- > with(plots,spacecurve,display):
- > with(linalg):

#### *Données initiales au choix de l'utilisateur*

> a:=1:	# distance entre deux sources voisines
> theta:= Pi/2;phi:=Pi/2:	# orientation ligne contenant les sources
> OE:=50*a:	# distance entre origine O et centre E de l'écran
> NbF:=31:	# nombre approché de franges à dessiner en configuration Young
> lambda:=array(1..2,[0.0065*a,0.0062*a]):	# longueurs d'ondes émises par les sources
> Grille:=200:	# réglage de la finesse du maillage du dessin

#### *Calculs préliminaires*

> for m from 1 to 2 do > k[m]:=2*Pi/lambda[m]: > od:	# normes des deux vecteurs d'onde
> Champ:=NbF*lambda[1]*OE/(2*a):	# étendue zone de tracé des interférences
> PAS:=vector([a*sin(theta)*cos(phi), a*sin(theta)*sin(phi),a*cos(theta)]):	# vecteur directeur de la ligne joignant les sources

**Calculs pour le tracé des interférences**

> M:=vector([OE,y,z]):	# coordonnées du point courant M de l'écran
> for i from 1 to 2 do > facteur:=i-3/2: > S[i]:=vector([PAS[1]*facteur, PAS[2]*facteur,PAS[3]*facteur]): > V:=evalm(M-S[i]): > L[i]:=sqrt(V[1]^2+V[2]^2+V[3]^2): > od:	# calcul trajet L[i] de chaque rayon
> EclR:=0:	# initialisation éclaircissement relatif en M
> for m from 1 to 2 do > EclR:= EclR+(1+cos(k[m]*(L[2]-L[1])))/4: > od:	# éclaircissement relatif en M
> Franges:=plot3d([OE,y,z],y=-Champ..Champ,z=-Champ..Champ, grid=[Grille,Grille],color=COLOR(RGB,EclR,EclR,EclR), orientation=[-145,65],style=patchnograd,axes=normal, tickmarks=[2,3,3],axesfont=[HELVETICA,BOLD,6]):	

**Calculs pour le tracé de deux rayons particuliers**

> P:=vector([OE,Champ/2,Champ/2]):	# définition du point particulier
> for i from 1 to 2 do > SP:=evalm(P-S[i]): > r[i]:=spacecurve([S[i][1]+SP[1]*u, S[i][2]+SP[2]*u,S[i][3]+SP[3]*u], u=0..1,numpoints=2,color=BLACK): > od:	# équation paramétrique de chaque r[i]
> Rayons:=convert(r,set):	
> Sources:=spacecurve([PAS[1]*(u-3/2),PAS[2]*(u-3/2) PAS[3]*(u-3/2)],u=1..2,numpoints=2, style=POINT,symbol=CIRCLE, color=BLACK):	# point de départ de r[i]

**Dessin**

> display(Sources,Rayons,Franges);

## Variante n° 2 :

### cas de deux ondes trichromatiques

---

$$(\lambda_1 = 0,0065 \cdot a ; \lambda_2 = 0,0054 \cdot a ; \lambda_3 = 0,0047 \cdot a)$$

- > with(plots,spacecurve,display):
- > with(linalg):

#### *Données initiales au choix de l'utilisateur*

> a:=1:	# distance entre deux sources voisines
> theta:= Pi/2:phi:=0:	# orientation ligne contenant les sources
> OE:=20*a:	# distance entre origine O et centre E de l'écran
> NbF:=55:	# nombre approché de franges à dessiner en configuration Young
> lambda:=array(1..3,[0.0065*a,0.0054*a,0.0047*a]):	# longueurs d'ondes émises par les sources
> Grille:=200:	# réglage de la finesse du maillage du dessin

#### *Calculs préliminaires*

> for m from 1 to 3 do > k[m]:=2*Pi/lambda[m]: > od:	# normes vecteurs d'onde pour chaque radiation
> Champ:=NbF*lambda[1]*OE/(2*a):	# étendue zone de tracé des interférences
> PAS:=vector([a*sin(theta)*cos(phi), a*sin(theta)*sin(phi),a*cos(theta)]):	# vecteur directeur de la période spatiale des sources

**Calculs pour le tracé des interférences**

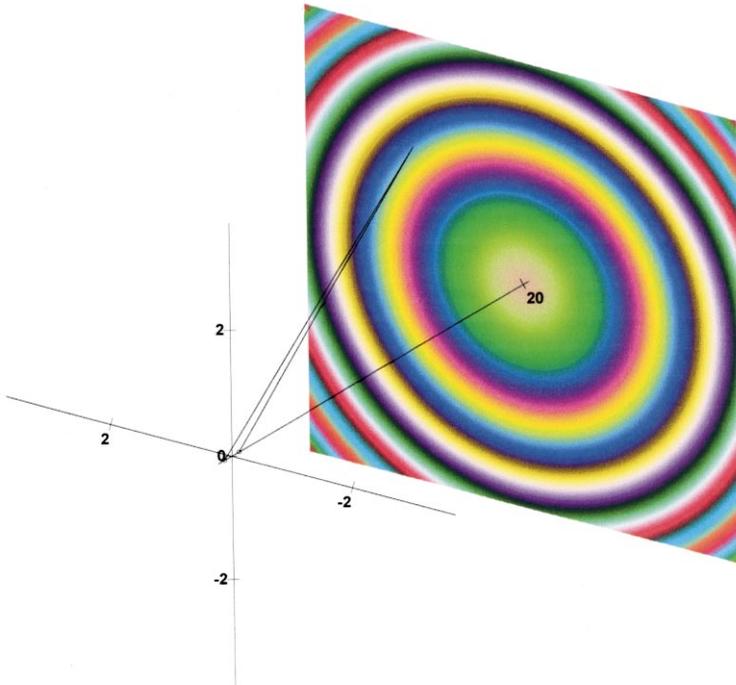
> M:=vector([OE,y,z]):	# coordonnées du point courant M de l'écran
> for i from 1 to 2 do > facteur:=i-3/2: > S[i]:=vector([PAS[1]*facteur, PAS[2]*facteur,PAS[3]*facteur]): > V:=evalm(M-S[i]): > L[i]:=sqrt(V[1]^2+V[2]^2+V[3]^2): > od:	# calcul trajet L [i] de chaque rayon
> for m from 1 to 3 do > EclR[m]:=((1+cos(k[m]*(L[2]-L[1]))))/2: > od:	# éclaircissement relatif en M pour la radiation N° m
> Franges:=plot3d([OE,y,z],y=-Champ..Champ,z=-Champ..Champ, grid=[Grille,Grille],color=COLOR(RGB,EclR[1],EclR[2],EclR[3]), orientation=[-145,65],style=patchnograd,axis=normal, tickmarks=[2,3,3],axesfont=[HELVETICA,BOLD,6]):	

**Calculs pour le tracé de deux rayons particuliers**

> P:=vector([OE,Champ/2,Champ/2]):	# définition du point particulier
> for i from 1 to 2 do > SP:=evalm(P-S[i]): > r[i]:=spacecurve([S[i][1]+SP[1]*u, S[i][2]+SP[2]*u,S[i][3]+SP[3]*u], u=0..1,numpoints=2,color=BLACK): > od:	# équation paramétrique de chaque r[i]
> Rayons:=convert(r,set):	
> Sources:=spacecurve([PAS[1]*(u-3/2),PAS[2]*(u-3/2), PAS[3]*(u-3/2)],u=1..2,numpoints=2, style=POINT,symbol=CIRCLE,color=BLACK):	# point de départ de r[i]

**Dessin**

> display(Sources,Rayons,Franges);



**Figure 7** : Interférences annulaires de deux sources trichromatiques identiques et ponctuelles  
Données initiales : celle du listing «Variante N° 2» de l'Annexe  
*grille choisie pour le dessin : (200×200) points.*