

Université de BATNA - INSTITUT DE GENIE MECANIQUE

APPLICATION AUX ECOULEMENTS POTENTIELS

Superposition d'un écoulement autour d'un cylindre et une circulation

Proposé par: Dr L. MESSAOUDI

Préparé par: N.

BESSANANE & C. MESSAOUDI

Introduction

Ce mini projet nous a été destiné pour le but d'avoir une initiation au logiciel Maple, à travers une application de la théorie des écoulements potentiels ; Cette application se résume à la représentation graphique de la combinaison d'un écoulement autour d'un cylindre et d'un vortex, en utilisant le logiciel courant (Maple), après avoir quelques notions sur ce dernier on a pu élaborer un petit programme qui permet à son utilisateur de faire sa propre combinaison des écoulements élémentaires (écoulement plan uniforme, source, puit et vortex).

Explication du contenu du programme

Ce programme contient implicitement les équations sous forme complexe des écoulements élémentaires cités dessus, le choix de certains ou la totalité de ces écoulements et le nombre ainsi que les données de chaque, se fait à travers des fenêtres appelées Maplets, qui sont des sous-routines dans Maple, modifiables selon le choix, après la saisie de toutes les données, les types d'écoulement choisis affectés à son nombre et à ses propres données chacun d'eux, sont sommés dans une seule équation sous forme complexe dont on peut tirer les fonctions de courant et les lignes de courant ainsi que le champ de vitesses qui seront représentés graphiquement comme résultat.

Mode de fonctionnement du programme

Quand on lance l'exécution du programme une fenêtre apparaît dont il faut remplir ses cases vides par le nombre de chaque type d'écoulement, si on veut omettre un type de ces écoulements il faut mettre tout simplement un zéro à la case correspondante ; Après avoir finir la première étape d'autres fenêtres apparaissent, leur nombre est selon le choix du nombre de chaque type d'écoulements mentionné à la première étape, et qui ont pour objet acquérir les données de chaque composant de la combinaison, après l'achèvement de cette seconde étape on trouvera le résultat graphique représenté par : les fonctions de courant, les lignes de courant, la superposition des lignes et des fonctions de courant pour vérifier leur orthogonalité et en fin la superposition des lignes de courant et le champ de vitesses.

Pour obtenir une combinaison d'un écoulement autour d'un cylindre et d'un vortex comme dans notre cas on a introduit les données suivantes :

Première étape :

Le nombre des écoulements plans = 1

Le nombre des sources = 1

Le nombre des puits = 1

Le nombre des vortex = 1

Deuxième étape :

Débit volumique pour l'écoulement plan $Q_0=10$

L'angle d'inclinaison $\text{Alpha}=15^\circ$

Débit volumique de la source $Q_v=150$

Coordonnées du centre de la source $X_0=1$, $Y_0=-1$

Débit volumique du puit $Q_v=150$

Coordonnées du centre du puit $X_0=3$, $Y_0=-1$

La circulation du vortex $\text{Gamma}=100$

Coordonnées du centre du vortex $X_0=2$, $Y_0=-1$

Remarque : Les lignes horizontales qui apparaissent sur les graphes démarrants du centre du puit ou de la source vers la gauche dans les lignes de courant et du centre du vortex vers la gauche dans les fonctions de courant sont dûent à la dicontinuité de la fonction tangente qui apparaît à l'inversion des équations de celles ci.

Contenu du progamme

```
> restart;  
> with(Maplets[Elements]):  
    maplet := Maplet(  
        Window( 'title' = "Application aux écoulements
```

```

potentiels",
    [
        ["Donner le nombre des écoulements
plans:",TextField['T1']( 3 )],
        ["Donner le nombre des sources:
",TextField['T2']( 3 )],
        ["Donner le nombre des puits:
",TextField['T3']( 3 )],
        ["Donner le nombre des vortex:
",TextField['T4']( 3 )],
        Button( "OK", Shutdown( ['T1,T2,T3,T4'] )
    )
    ]
)
):
result := Maplets[Display](maplet):
Nf := parse( result[1] ):
Ns := parse( result[2] ):
Np := parse( result[3] ):
Nv := parse( result[4] ):
> assume(x,real,y,real):
z:=x+I*y:
> if (Nf = 0) then
    Ff:=0:
end if:
if (Nf > 0) then
    if (Nf = 1) then Vf := "Veuillez saisir les données
concernant l'écoulement uniforme" end if:
    for i from 1 to Nf do
        if (Nf > 1) and (i = 1) then
            Vf := "Veuillez saisir les données concernant
le premier écoulement uniforme" end if:
            if (Nf > 1) and (i > 1) then
                Vf := "Veuillez saisir les données concernant
l'écoulement uniforme suivant" end if:
            donf := Maplet(
                Window( 'title' = "Acquisition des données",
                    [
                        [Vf],
                        [""],
                        ["Débit volumique
", TextField
['TF1']( 6 )],
                        [""],
                        ["L'angle que forme l'écoulement avec le
plan horizontal (en degré)
", TextField['TF2']( 6 )],
                        Alpha =

```

```

        Button( "OK", Shutdown( ['TF1,TF2'] ) )
    ]
)
):
    dof := Maplets[Display](donf):
    Qo[i] := parse( dof[1] ):
    alphad[i] := parse( dof[2] ):
    alpha[i] := alphad[i]*Pi/180:
    ff[i]:=evalc(Qo[i]*z*exp(-I*alpha[i])):
end do:
Ff:=sum(ff[j],j=1..Nf):
end if:
> if (Ns = 0) then
    Fs:=0:
    Xsmi:=100:Ysmi:=100:
    Xsma:=-100:Ysma:=-100:
    Qvms:=0:
end if:
if (Ns > 0) then
    if (Ns = 1) then Vs := "Veuillez saisir les données
concernant la source" end if:
    for i from 1 to Ns do
        if (Ns > 1) and (i = 1) then
            Vs := "Veuillez saisir les données concernant
la première source" end if:
            if (Ns > 1) and (i > 1) then
                Vs := "Veuillez saisir les données concernant
la source suivante" end if:
            dons := Maplet(
                Window( 'title' = "Acquisition des données",
                    [
                        [Vs],
                        [""],
                        ["Débit volumique
Qv =", TextField['TS1'] ( 6 )],
                        [""],
                        ["Les coordonnées du centre de la source
Xo =", TextField['TS2'] ( 4 )," Yo =
",TextField['TS3'] ( 4 )],
                        Button( "OK", Shutdown( ['TS1,TS2,TS3'] ) )
                    ]
                )
            )
        )
    ):
    dos := Maplets[Display](dons):

```

```

        Qvs[i] := parse( dos[1] ):
        xos[i] := parse( dos[2] ):
        yos[i] := parse( dos[3] ):
        zos[i]:=xos[i]+I*yos[i]:
        fs[i]:=evalc((Qvs[i]/(2*Pi))*ln(z-zos[i])):
    end do:
    Fs:=sum(fs[k],k=1..Ns):
    Xsmi:=min(seq(xos[k],k=1..Ns)):
    Ysmi:=min(seq(yos[k],k=1..Ns)):
    Xsma:=max(seq(xos[k],k=1..Ns)):
    Ysma:=max(seq(yos[k],k=1..Ns)):
    Qvms:=max(seq(Qvs[k],k=1..Ns)):
end if:
> if (Np = 0) then
    Fp:=0:
    Xpmi:=100:Ypmi:=100:
    Xpma:=-100:Ypma:=-100:
    Qvmp:=0:
end if:
if (Np > 0) then
    if (Np = 1) then Vp := "Veuillez saisir les données
concernant le puit" end if:
    for i from 1 to Np do
        if (Np > 1) and (i = 1) then
            Vp := "Veuillez saisir les données concernant
le premier puit" end if:
        if (Np > 1) and (i > 1) then
            Vp := "Veuillez saisir les données concernant
le puit suivant" end if:
        donp := Maplet(
            Window( 'title' = "Acquisition des données",
                [
                    [Vp],
                    [""],
                    ["Débit volumique

Qv =          ", TextField['TP1']( 6 )],
                    [""],
                    ["Les coordonnées du centre du puit
Xo =", TextField['TP2']( 4 ),"          Yo = ",
TextField['TP3']( 4 )],

                    Button( "OK", Shutdown( ['TP1,TP2,TP3'] )
                )
            ]
        )
    ):
    dop := Maplets[Display](donp):

```

```

        Qvp[i] := parse( dop[1] ):
        xop[i] := parse( dop[2] ):
        yop[i] := parse( dop[3] ):
        zop[i]:=xop[i]+I*yop[i]:
        fp[i]:=evalc(-(Qvp[i]/(2*Pi))*ln(z-zop[i]])):
    end do:
    Fp:=sum(fp[l],l=1..Np):
    Xpmi:=min(seq(xop[l],l=1..Np)):
    Ypmi:=min(seq(yop[l],l=1..Np)):
    Xpma:=max(seq(xop[l],l=1..Np)):
    Ypma:=max(seq(yos[l],l=1..Np)):
    Qvmp:=max(seq(Qvp[l],l=1..Np)):
end if:
> if (Nv = 0) then
    Fv:=0:
    Xvmi:=100:Yvmi:=100:
    Xvma:=-100:Yvma:=-100:
    Qvmv:=0:
end if:
if (Nv > 0) then
    if (Nv = 1) then Vv := "Veuillez saisir les données
concernant le vortex" end if:
    for i from 1 to Nv do
        if (Nv > 1) and (i = 1) then
            Vv := "Veuillez saisir les données concernant
le premier vortex" end if:
            if (Nv > 1) and (i > 1) then
                Vv := "Veuillez saisir les données concernant
le vortex suivant" end if:
            donv := Maplet(
                Window( 'title' = "Acquisition des données",
                    [
                        [Vv],
                        [""],
                        ["La circulation
Gamma = ", TextField['TV1']( 6 )],
                        [""],
                        ["Les coordonnées du centre du vortex
Xo = ", TextField['TV2']( 4 )," Yo =
",TextField['TV3']( 4 )],
                        Button( "OK", Shutdown( ['TV1,TV2,TV3'] )
                    )
                ]
            )
        )
    ):
    dov := Maplets[Display](donv):

```

```

        Gamma[i] := parse( dov[1] ):
        xov[i] := parse( dov[2] ):
        yov[i] := parse( dov[3] ):
        zov[i]:=xov[i]+I*yov[i]:
        fv[i]:=evalc(I*Gamma[i]/(2*Pi)*ln(z-zov[i])):
    end do:
    Fv:=sum(fv[m],m=1..Nv):
    Xvmi:=min(seq(xov[m],m=1..Nv)):
    Yvmi:=min(seq(yov[m],m=1..Nv)):
    Xvma:=max(seq(xov[m],m=1..Nv)):
    Yvma:=max(seq(yov[m],m=1..Nv)):
    Qvmv:=max(seq(Gamma[m],m=1..Nv)):
end if:
> Fg:=evalc(Ff+Fs+Fp+Fv):
psi:=Im(Fg):
phi:=Re(Fg):
> if (Ns=0) and (Np=0) and (Nv=0) then
    X1:=-5:X2:=5:Y1:=-5:Y2:=5 else
    Xmi:=min(Xsmi,Xpmi,Xvmi):Ymi:=min(Ysmi,Ypmi,Yvmi):
    Xma:=max(Xsma,Xpma,Xvma):Yma:=max(YSma,Ypma,Yvma):
    Qmax:=max(Qvms,Qvmp,Qvmv):
    X1:=Xmi-(Qmax*0.027):X2:=Xma+(Qmax*0.027):
    Y1:=Ymi-(Qmax*0.027):Y2:=Yma+(Qmax*0.027):
end if:

```

Résultats Graphiques

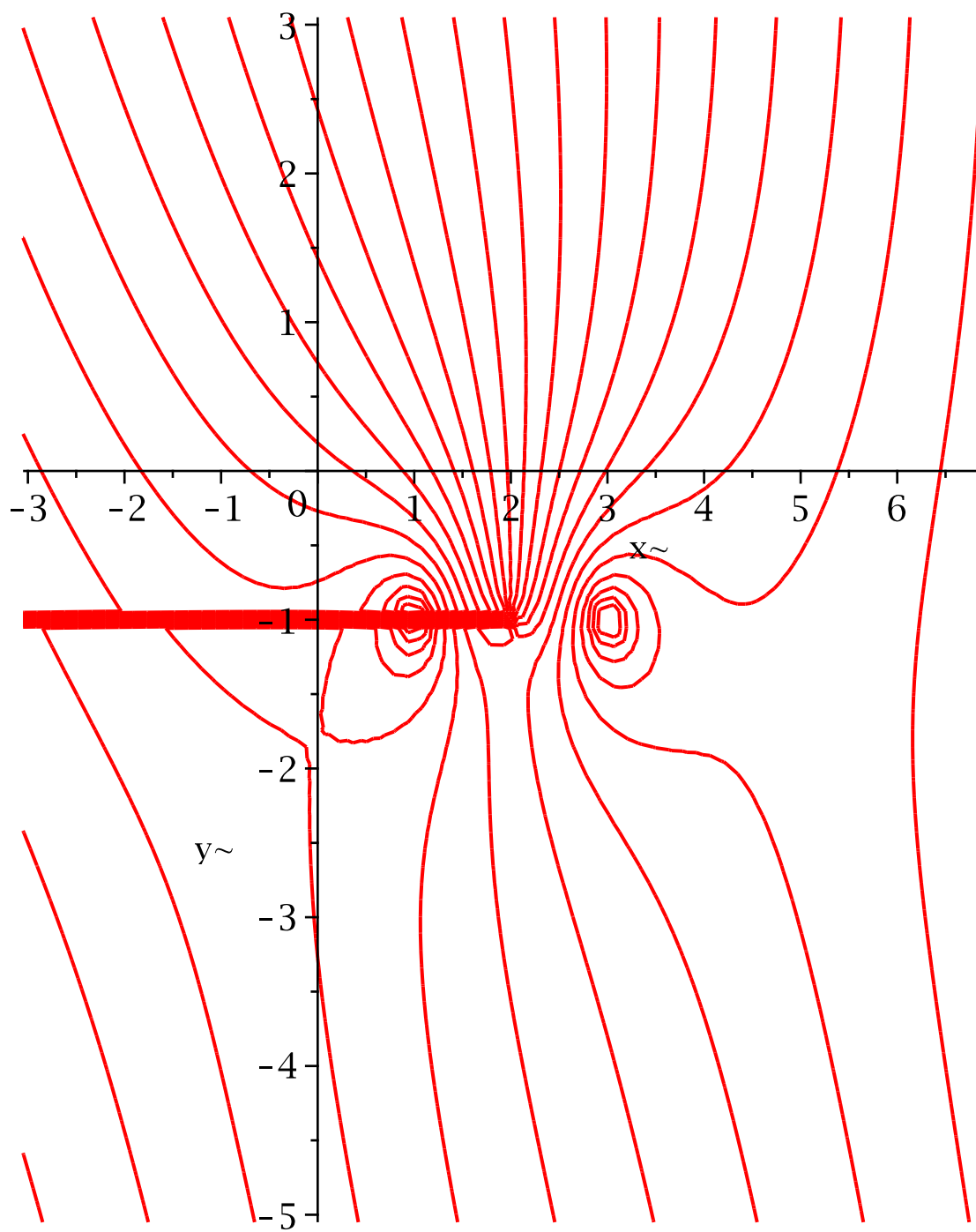
```

> with(plots):
g0:= fieldplot([diff(phi,x),-diff(psi,x)],x=X1..X2,y=
Y1..Y2,color=red):
g1:= contourplot(phi,x=X1..X2,y=Y1..Y2,numpoints=5000,
contours=25,color=red):
g2:= contourplot(psi,x=X1..X2,y=Y1..Y2,numpoints=5000,
contours=25,color=blue):
display(g1,title="Fonctions de courant");
display(g2,title="Lignes de courant");
display(g1,g2,title="Vérification de l'orthogonalité
des fonctions et des lignes de courant");
display(g2,g0,title="Superposition du champ de vitesses
et des lignes de courant");

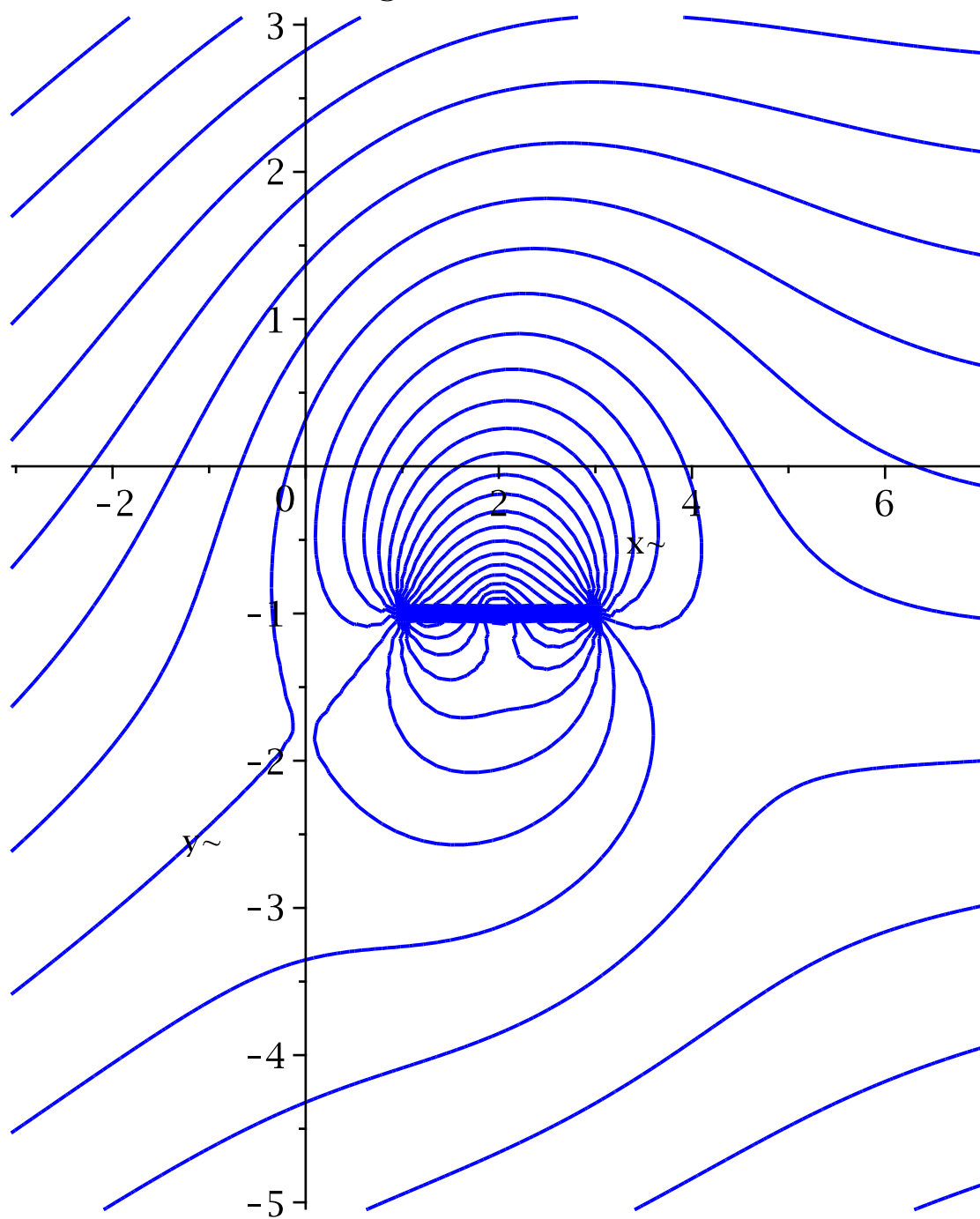
```

Warning, the name changecoords has been redefined

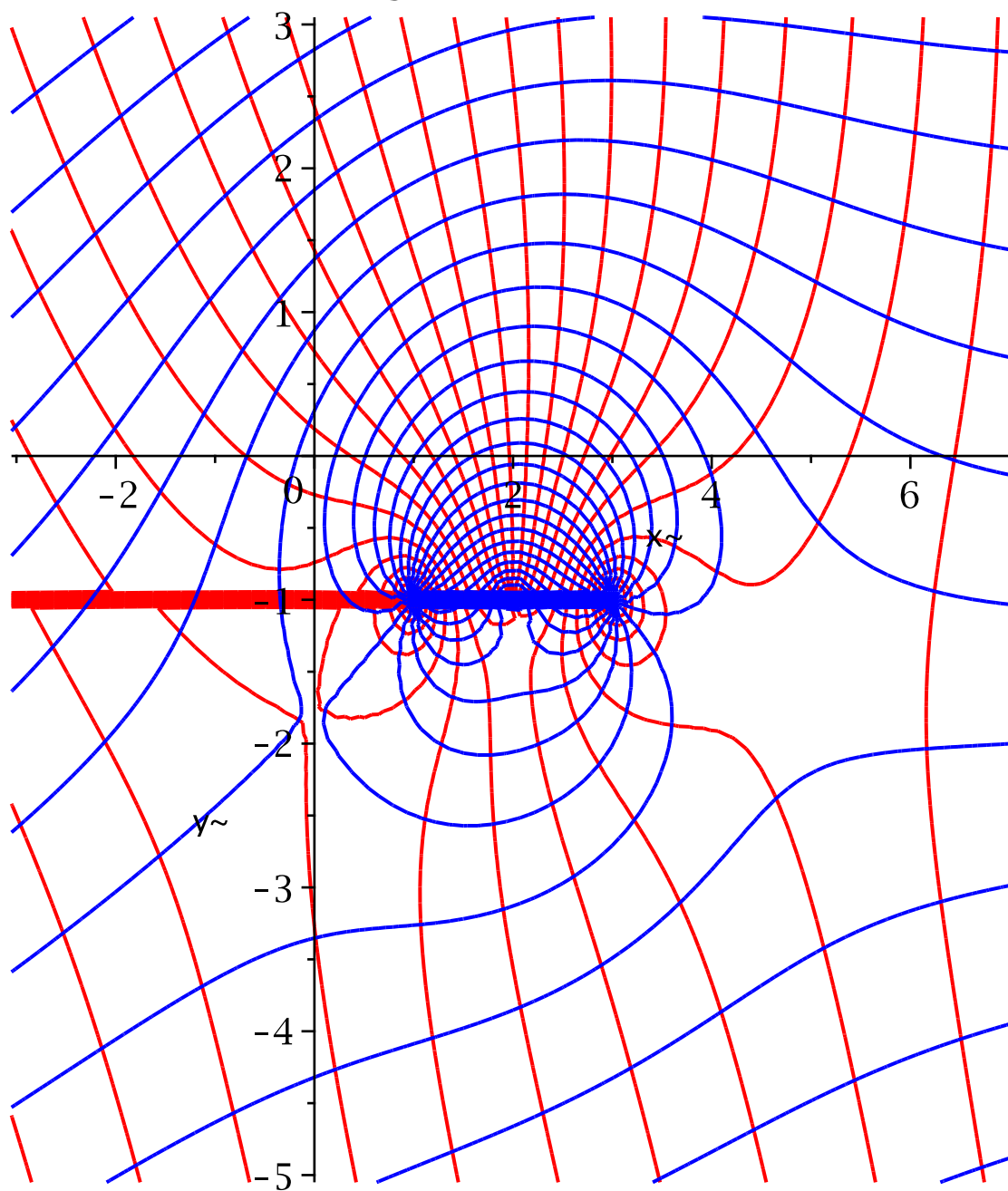
Fonctions de courant



Lignes de courant



Vérification de l'orthogonalité des fonctions et des
lignes de courant



Superposition du champ de vitesses et des lignes de courant

