

```

PROGRAM resolutionProgramme
PRINT *, "Ce programme calcul un décollage pour un profil NACA 4415"
CALL decollage
END PROGRAM

!Sous-programme de résolution globale
SUBROUTINE decollage
IMPLICIT NONE
DOUBLE PRECISION :: v0, v1, v2, dt, vVent, d, t, P, rechercheZeroDichoto,
Cl, gravity
DOUBLE PRECISION :: rG, m, Cx, distanceMax, rho, A, Re, Ri, S, N,
PousseeTurboprop
DOUBLE PRECISION :: TempComb, BPR, Tsol, Ve1, Ve2, Mach, PousseeTurbofan,
Trainee, VTemp
DOUBLE PRECISION :: Palti, Talti, altitude,vinit, Pext, Text,
rapportCompression
DOUBLE PRECISION :: Temp3, VeInit, TempSortie, RVe, NCFM, SFC1, SFC2,
consommation, corde
DOUBLE PRECISION :: angle1, angle2,altitude1, altitude2,angle0
DOUBLE PRECISION :: temp1, temp2, Saile,volets
DOUBLE PRECISION, PARAMETER :: pi=3.1415926, debut=0.0, fin=1000.0,
Cp=1004, Rair=287, gamma=1.4
INTEGER :: i

!Demande des données du prob
CALL demandeDonneeProb(P,S,rG,m,vVent,Re,Ri,N,PousseeTurbofan, &
Cx,BPR, SFC2,vinit,dt,distanceMax, altitude, Tsol, rapportCompression, &
TempComb, RVe, NCFM, SFC1, corde, Saile,volets,angle0)
Pext=Palti(altitude)
Text=Talti(Tsol, altitude)
rho=Pext/(Text*287.0)
A= 3.14159265*(Re*Re-Ri*Ri)
TempSortie=TempComb*RapportCompression**((1.0-gamma)/gamma)

!initialisation données
v0=vinit
v1=vinit
v2=vinit
d=0
t=0
i=0
VeInit=0.001
consommation=0
altitude2=altitude
altitude1=altitude
angle1=0.0
angle2=0.0

OPEN(10,FILE='PousseeAcceleration.dat')
OPEN(11,FILE='Autres.dat')

WRITE(10,*) "Distance parcourue", "Temps
",&
"Poussee", "Acceleration"
WRITE(11,*) "Distance parcourue", "Temps", "&
"Vitesse Appareil", "Consommation", "&
"Vitesse echappement", "altitude", "&
"angle d'attaque", "Vitesse au sol", "&

```

"Mach appareil"

!
_____!
_____!

!résolution de l'équation au premier pas de temps

!Calcul de la traînée au temps i-1

Trainee=-Cx*rho/2.0*(v0+vVent)**2*S

!Calcul de la vitesse du fluide en dehors de la vitesse d'échappement au temps i-1

Vtemp= rechercheZeroDichoto(debut,fin, v0+vVent, P,rG,A,rho)

IF (Vtemp<0.0) THEN

d=distanceMax+1

PRINT *, "Erreur dans le calcul (1)"

ELSE

!Calcul de la vitesse du fluide d'échappement au temps i-1

Temp3=2.0*Cp*TempComb*RapportCompression**((1.0-gamma)/gamma)*((gamma-1.0)*Mach(gamma,TempSortie,Rair,VeInit)/2.0)

Vel=sqrt(Temp3*RVe)

!Calcul de la vitesse au temps i=dt

v1=v0+dt/m*(Trainee+N*PousseeTurboprop(vtemp,v0+vVent,vel,BPR,A,rho)+NCFM*PousseeTurbofan)

!Ecriture des résultats dans un fichier de données

WRITE(10,*) d, t, N*PousseeTurboprop(vtemp,v0+vVent,vel,BPR,A,rho)+NCFM*PousseeTurbofan, &

Trainee+N*PousseeTurboprop(vtemp,v0+vVent,vel,BPR,A,rho)+NCFM*PousseeTurbofan

d=d+(v1*cos(angle1*PI/180))*dt

t=t+dt

consommation= consommation +

(dt/3600.0)*(SFC1*P*N+NCFM*pousseeTurbofan*SFC2)*0.001

WRITE(11,*) d, t, v1, consommation, Vel, altitude, angle1,v1*cos(angle1*PI/180), Mach(gamma,Text,Rair,v1)

Pext=Palti(altitude)

Text=Talti(Tsol, altitude)

rho=Pext/(Text*287.0)

!Suivi de l'avancement du programme sur le terminal

i=i+1

PRINT *, "pas",i

END IF

!
_____!
_____!

!résolution pour les autres pas de temps

DO WHILE (d<distanceMax)

!Calcul de la traînée au temps i-1

Trainee=-Cx*rho/2*(v1+vVent)**2*S

!Calcul de la vitesse du fluide en dehors de la vitesse d'échappement au temps i-1

Vtemp= rechercheZeroDichoto(debut,fin, v1+vVent,P,rG,A,rho)

IF (Vtemp<0.0) THEN

PRINT *, "Erreur dans le calcul (2)"

d=distanceMax+1

```

ELSE
    !Calcul de la vitesse du fluide d'échappement au temps i-1
    Temp3=2.0*Cp*TempComb*RapportCompression**((1.0-
gamma)/gamma)*((gamma-1.0)*Mach(gamma,TempSortie,Rair,Vel)/2.0)
    Ve2=sqrt(Temp3*RVe)
    !Calcul de la vitesse au temps i, dépend de l'angle d'attaque et si
la portance est suffisamment élevée

angle2=angle1+180/PI*dt/(m*v1)*(Cl(angle1+angle0)*volets/2*rho*V1*V1*Sail
e-m*gravity(altitude1)*cos(angle1*PI/180))
    IF (angle2<0) THEN
        angle2=0.0
    END IF
    temp1=Trainee+N*PousseeTurboprop(vtemp,v1+vVent,ve2,BPR,A,
rho)+NCFM*PousseeTurbofan-m*gravity(altitude1)*sin(angle1*PI/180)
    temp2=dt/(m)*temp1
    v2=v1+temp2
    altitude2=altitude1+dt*v1*sin(angle1*PI/180)
    !Ecriture des résultats dans un fichier de données
    WRITE(10,*) d, t, N*PousseeTurboprop(vtemp,v1+vVent,ve2,BPR,A,
rho)+NCFM*PousseeTurbofan, &

Trainee+N*PousseeTurboprop(vtemp,v1+vVent,ve1,BPR,A,rho)+NCFM*PousseeTurb
ofan
    d=d+(v2*cos(angle2*PI/180))*dt
    t=t+dt
    consommation= consommation +
(dt/3600.0)*(SFC1*P*N+NCFM*pousseeTurbofan*SFC2)*0.001
    WRITE(11,*) d, t, v2, consommation, Ve2,altitude2, angle2,
v2*cos(angle2*PI/180),Mach(gamma,Text,Rair,v2)
    v0=v1
    v1=v2
    Vel=Ve2
    altitude1=altitude2
    angle1=angle2
    Pext=Palti(altitude2)
    Text=Talti(Tsol, altitude2)
    rho=Pext/(Text*287.0)
    !Suivi de l'avancement du programme sur le terminal
    i=i+1
    PRINT *, "pas",i
END IF
END DO

!Fin de programme et fermeture des fichiers
CLOSE(10)
CLOSE(11)
END SUBROUTINE

```

!

```

!les deux equations permettant de calculer la poussee
FUNCTION PousseeTurboprop(v1,v0,ve,BPR,A, rho)
IMPLICIT NONE
DOUBLE PRECISION:: PousseeTurboprop, v1,v0,ve,BPR,A, rho,temp, resultat
temp=rho*A/(2*(1+BPR))*(v0+v1)

```

```

resultat=temp*((1+BPR)*(v1-v0)+ve-v1)
PousseeTurboprop=resultat
END FUNCTION

```

```

Function Vcalcul(v,v0,P,Ng,A,rho)
IMPLICIT NONE
DOUBLE PRECISION :: v,v0,P,Ng,A,rho, Vcalcul
Vcalcul=(rho/2)*((v*v)-(v0*v0))*(v+v0)-(2*P*nG)/A
END FUNCTION

```

! _____ !

```

!Fonction permettant de déterminer le coefficient de portance (profil
naca4415)

```

```

FUNCTION Cl(angle)
IMPLICIT NONE
DOUBLE PRECISION :: Cl, angle, resultat
resultat=0.0
IF (angle<=10) THEN
    resultat=0.50+0.12*(angle)
ELSE
    IF(angle<=20) THEN
        resultat=1.8
    ELSE
        resultat=0.0
    END IF
END IF
Cl=resultat
END FUNCTION

```

```

!Fonction permettant de calculer la gravité

```

```

FUNCTION gravity(alti)
DOUBLE PRECISION, PARAMETER :: rayonTerre=6.6378d6, masseTerre=5.972d24,
G=6.6738d-11
DOUBLE PRECISION :: gravity, alti
gravity=G*masseTerre/((rayonTerre+alti)**2)
END FUNCTION

```

! _____ !

```

!Pression et température en fonction de l'altitude

```

```

FUNCTION Palti(z)
DOUBLE PRECISION :: Palti,z
!Pression en fonction de l'altitude
Palti = 101325.0*((288-0.0065*z)/288.0)**(5.255)
END FUNCTION

```

```

FUNCTION Talti(Tsol,z)

```

```

DOUBLE PRECISION :: Talti,Tsol, z, resultat
!Température en fonction de l'altitude, Tsol=15°C par défaut!
IF (z<11000) THEN
    resultat = Tsol-((6.5*z)/1000.0)
    IF (resultat<(273.15-56.5)) THEN
        resultat=273.15-56.5
    END IF
ELSE

```

```

    resultat=273.15-56.5
END IF
Talti=resultat
END FUNCTION

```

! _____ !

```

!Fonction calculant le nombre de mach
FUNCTION Mach(gamma,T,R,V)
IMPLICIT NONE
DOUBLE PRECISION :: gamma, V, R, T, Mach
Mach=V/sqrt(gamma*R*T)
END FUNCTION

```

```

!Fonction calculant le reynolds
FUNCTION Re(rho,V,c)
DOUBLE PRECISION, PARAMETER :: mu=1.8*10**(-5) !Mu air supposé constant
en fonction de T
DOUBLE PRECISION :: rho, V, c, Re
Re= rho*V*c/mu
END FUNCTION

```

! _____ !

```

!Fonction de calcul
FUNCTION rechercheZeroDichoto(debutVrai,finVrai,v0,P,Ng,A,rho)
IMPLICIT NONE
DOUBLE PRECISION:: rechercheZeroDichoto,v0,P,Ng,A,rho, debutVrai, finVrai
DOUBLE PRECISION :: debut,fin, milieu, Fmilieu, Fdeb, Ffin
DOUBLE PRECISION, PARAMETER :: seuil=0.0001
INTEGER :: i

```

```

!Initialisation
debut=debutVrai
Fin=finVrai
milieu=(debut+fin)/2
!Cas ou f(debut) ou f(fin)=0
Ffin=(rho/2)*((fin*fin)-(v0*v0))*(fin+v0)-(2*P*nG)/A
Fdeb=(rho/2)*((debut*debut)-(v0*v0))*(debut+v0)-(2*P*nG)/A
!Verification d'éventuelle erreur
IF ((Fdeb*Ffin)>0.0) THEN
rechercheZeroDichoto=-5
PRINT *, "Erreur de dichotomie (1)", Fdeb, Ffin
ELSE
    IF (abs(Fdeb)<seuil) THEN
        milieu=debut
    ELSE IF (abs(Ffin)<seuil) THEN
        milieu=fin
    END IF
    i=0
    !Boucle
    Fmilieu=(rho/2)*((milieu*milieu)-(v0*v0))*(milieu+v0)-(2*P*nG)/A
    DO WHILE (abs(Fmilieu)>seuil)
        IF ((Fdeb*Fmilieu)>0.0) THEN
            debut=milieu
            Fdeb=(rho/2)*((debut*debut)-(v0*v0))*(debut+v0)-(2*P*nG)/A

```

```

ELSE
  fin=milieu
  Ffin=(rho/2)*((fin*fin)-(v0*v0))*(fin+v0)-(2*P*nG)/A
END IF
milieu=(debut+fin)/2
i=i+1
Fmilieu=(rho/2)*((milieu*milieu)-(v0*v0))*(milieu+v0)-(2*P*nG)/A
!Vérification d'éventuelle erreur
IF ((Fdeb*Ffin)>0.0) THEN
  PRINT *, "Erreur de dichotomie (2)",i, Fdeb,Fmilieu
  Fmilieu=seuil/2.0
  milieu=-5
END IF
END DO
!Sortie du résultat
rechercheZeroDichoto=milieu
END IF
END FUNCTION

```

! _____ !

```

!Sous-Programme demandant les données du problèmes
SUBROUTINE demandeDonneeProb(P,S,rG,m,vVent,Re,Ri,N,PousseeTurbofan, &
Cx,BPR, SFC2,vinit,dt,distanceMax, altitude, Tsol, rapportCompression, &
TempComb, RVe, NCFM, SFC1, corde, Saile,volets, angle0)
IMPLICIT NONE
DOUBLE PRECISION :: vinit, rG, m, Cx,dt, distanceMax, Tsol,
rapportCompression, P, vVent,SFC2
DOUBLE PRECISION :: Re, Ri, S, N, PousseeTurbofan, BPR, altitude,
TempComb, RVe, NCFM,SFC1
DOUBLE PRECISION :: corde, Saile, L,volets, angle0
INTEGER :: i
!données du problème préenregistrées:
PRINT *, "Importer des données préenregistrées 1=BEE-Plane (TP400),", &
"2=Bee-Plane (NK12MV), 3=ATR72, 4=A321, 5=Données par défaut,
autre=données manuelles?"
READ *, i
SELECT CASE(i)
CASE(1)
P=7971000d0 !puissance du TP400 au décollage
BPR=8d-1 ! by-pass-ratio TP400
Cx=2.9d-3 ! Coefficient trainée Bee-Plane
Re=5.33/2 !rayon extérieur de l'hélice TP400
Ri=0.92/2 ! rayon intérieur de l'hélice TP400
S=890d0 ! surface bee-plane
TempComb=1500d0 !Température de combustion du TP400 ou turbine inlet
temperature
rapportCompression=25d0 ! Rapport de compression du TP400
SFC1= 0.238d0 ! Specific fuel consumption (consommation spécifique du
turbofan)
N=2
m=1.2d5
PRINT *, "Corde de l'aile?"
READ *, corde
PRINT *, "Longueur d'une l'aile ?"
READ *,L
Saile=L*corde*2

```

```

PRINT *, "Ajout de portence par les volets (en %) ?"
READ *,volets
volets=1+volets/100

CASE(2)
Cx=2.9d-3 ! Coefficient trainée Bee-Plane
S=890d0 ! surface bee-plane
P=11033000d0 ! Puissance NK-12
Re=6.2d0/2 ! rayon exterieur des hélices du Kouznetsov NK12-MA
Ri=1.2d0/2 ! rayon intérieur des hélices Kouznetsov NK-12M
N=2
m=1.2d5
PRINT *, "Rapport compression ?"
READ *, rapportCompression
PRINT *, "Température combustion ?"
READ *, TempComb
PRINT *, "SFC turbopropulseur (en kg/kW/h)?"
READ *, SFC2
PRINT *, "By-Pass-Ratio ?"
READ *, BPR
PRINT *, "Corde de l'aile?"
READ *, corde
PRINT *, "Longeur d'une l'aile ?"
READ *,L
Saile=L*corde*2
PRINT *, "Ajout de portence par les volets (en %) ?"
READ *,volets
volets=1+volets/100

CASE(3)
N=2d0
Re=4d0/2 ! hélice ATR72
Ri=0.680d0/2 ! hélice ATR 72
m=24000d0 ! masse de l'ATR 72
P=1881000d0 ! Puissance PW124B
S=27.05*10.79 ! Surface du ATR72
PRINT *, "By-Pass-Ratio ?"
READ *, BPR
PRINT *, "SFC turbopropulseur (en kg/kW/h)?"
READ *, SFC1
PRINT *, "Drag coefficient ?"
READ *, Cx
rapportCompression=17.0 ! Rapport de compression
TempComb=816.0+273.0 !! Turbine Inlet temperature PW124B
PRINT *, "Corde de l'aile?"
READ *, corde
PRINT *, "Longeur d'une l'aile ?"
READ *,L
Saile=L*corde*2
PRINT *, "Ajout de portence par les volets (en %) ?"
READ *,volets
volets=1+volets/100

CASE(4)
N=0.0d0
m=93500d0 !masse de l'A321
P=7971000.0d0!pas de turboprop
BPR=0.8d0 !

```

```

Re=10d0/2 !
Ri=0.1/2d0 !
TempComb=1500d0!
rapportCompression=25d0!
S=44.5*34.1
NCFM=2.0
PousseeTurbofan=148000
PRINT *, "Drag coefficient ?"
READ *, Cx
rapportCompression=0.1
TempComb=0.1
PRINT *, "TSFC turboréacteur (en kg/kN/h)?"
READ *, SFC2
Saile=130
PRINT *, "Ajout de portance par les volets (en %) ?"
READ *,volets
volets=1+volets/100
rG=0.85d0
PRINT *, "vitesse du vent (défaut 0)?"
READ *, vVent
PRINT *, "vitesse initiale de l'appareil (0 au décollage) ?"
READ *, vinit
PRINT *, "altitude (0 par défaut) ?"
READ *, altitude
PRINT *, "Température au sol (en K)?"
READ *, Tsol
dt=-1
  DO WHILE(dt<0)
    PRINT *, "Quel est le pas de temps?"
    READ *, dt
  END DO
distanceMax=-1
DO WHILE(distanceMax<0)
  PRINT *, "Quel est la distance maximal (arret du calcul)?"
  READ *, distanceMax
END DO
Rve=0.16

```

```

CASE(5)
P=7971000.0d0!puissance du TP400 au décollage
BPR=0.8d0 ! by-pass-ratio TP400
Cx=0.0029d0 ! Coefficient trainée Bee-Plane
Re=5.33/2d0 !rayon extérieur de l'hélice TP400
Ri=0.92/2d0 ! rayon intérieur de l'hélice TP400
S=890d0 ! surface bee-plane
TempComb=1500d0
rapportCompression=25d0
N=10000d0
m=1d5
rG=0.85d0
vVent=0d0
vinit=0d0
altitude=0d0
Tsol=300d0
dt=1.0d-2
distanceMax=4.5d3
Rve=0.16

```



```
NCFM=0.0
SFC2=41.4! consommation d'un LF 507
PousseeTurbofan=30.62d3 !poussee d'un Honeywell HTF7000
PousseeTurbofan=120000
SFC1=0.238
corde=5d0
Saile=180d0
volets=2.0
```

```
CASE DEFAULT
PRINT *, "Puissance moteurs ?"
READ *, P
PRINT *, "Rayon exterieur hélice ?"
READ *, Re
PRINT *, "Rayon interieur hélice ?"
READ *, Ri
PRINT *, "Surface contact ?"
READ *, S
PRINT *, "Nombre de moteurs ?"
READ *, N
PRINT *, "Masse de l'appareil (défaut 1e5) ?"
READ *, m
PRINT *, "By-Pass-Ratio ?"
READ *, BPR
PRINT *, "Drag coefficient ?"
READ *, Cx
PRINT *, "Rapport compression ?"
READ *, rapportCompression
PRINT *, "Température combustion ?"
READ *, TempComb
PRINT *, "SFC turbopropulseur (en kg/kW/h)?"
READ *, SFC1
PRINT *, "Nombre de Turboréacteur ?"
READ *, NCFM
PRINT *, "TSFC turboréacteur (en kg/kN/h)?"
READ *, SFC2
PRINT *, "Poussée turboréacteur (en N)?"
READ *, PousseeTurbofan
PRINT *, "Corde de l'aile?"
READ *, corde
PRINT *, "Longeur d'une l'aile ?"
READ *, L
Saile=L*corde
PRINT *, "Ajout de portence par les volets (en %) ?"
READ *,volets
volets=1+volets/100
```

```
END SELECT
IF (i==5) THEN
ELSE
PRINT *, "Angle des ailes à plats ?"
READ *, angle0
PRINT *, "Ajout de portence par les volets (en %) ?"
READ *,volets
volets=1+volets/100
PRINT *, "Surface Alaire ?"
READ *,Saile
```

```
END IF
IF (i==5 .OR. i==4) THEN
ELSE
  PRINT *, "Rendement turbopropulseur ?"
  READ *, rG
  PRINT *, "vitesse du vent (défaut 0)?"
  READ *, vVent
  PRINT *, "vitesse initiale de l'appareil (0 au décollage) ?"
  READ *, vinit
  PRINT *, "altitude (0 par défaut) ?"
  READ *, altitude
  PRINT *, "Température au sol (en K)?"
  READ *, Tsol
  PRINT *, "Rendement de turbine (environ 0.2)?"
  READ *, RVe
  dt=-1
  DO WHILE(dt<0)
    PRINT *, "Quel est le pas de temps?"
    READ *, dt
  END DO
  distanceMax=-1
  DO WHILE(distanceMax<0)
    PRINT *, "Quel est la distance maximal (arret du calcul)?"
    READ *, distanceMax
  END DO
END IF
END SUBROUTINE
```