\

□□□□□□□□□□□□□□

پیوست آ

برنامه نوشته شده برای نرم‌افزار فلوئنت

```
#include ”udf.h”

/*Constant Parameter Definition*/
#define J0 11.9281
#define B0 1.0
#define a 0.01
#define Pi 3.14159
#define Th 0.12
#define Ch 1.0
#define D 5.0
#define K 0.5
#define ep 0.0000009

/*Define Source Term; X-Momentum Equation*/
DEFINE_SOURCE(xmom, c, t, dS, eqn)
{
real Z[ND_ND];
real Airfoil_Tet,Y_Airfoil;
real Xloc,Yloc,X,Y,R,Teta,Angle;
real Alpha,TIME,XDC,XC;
real source,Yprim,Eta;
FILE *fp2;
TIME=CURRENT_TIME;
Alpha=0.1745329-D*0.01745329*sin(2.0*K*TIME);
C_CENTROID(Z, c, t);
X=Z[0];
Y=Z[1];
R=sqrt(pow(X,2)+pow(Y,2));
Teta=atan(Y/X);
if (X<=0.0)
{
Angle=fabs(Teta)-fabs(Alpha);
```

```
Xloc=R*cos(Angle);
Xloc=-Xloc;
* Yloc=R*sin(Angle);
}
if (X>0.0)
{
if (Teta<(15.77777-Alpha))
{
Angle=fabs(Teta)+fabs(Alpha);
Xloc=R*cos(Angle);
Yloc=R*sin(Angle);
}
if (Teta==(15.77777-Alpha))
{
Xloc=0.0;
Yloc=R;
}
if (Teta>(15.77777-Alpha))
{
Angle=fabs(Teta)-fabs(Alpha);
Xloc=R*cos(Angle);
Xloc=-Xloc;
Yloc=R*sin(Angle);
}
}
XDC=(Xloc+0.25)/Ch;
XC=(Xloc+0.25)*Ch;
Y_Airfoil=(Th*Ch/0.2)*(0.2969*pow(XDC,0.5)-0.1260*XDC-0.3516*pow(XDC,2)
+0.2843*pow(XDC,3)-0.1015*pow(XDC,4));
Airfoil_Tet=(Th*Ch/0.2)*(((0.2969/2.0)*sqrt(1.0/XC))-(0.1260/Ch)-(2.0*0.3516*XDC)
+(3.0*0.2843*pow(XDC,2))-(4.0*0.1015*pow(XDC,3)));
Yprim=Yloc-Y_Airfoil;
Eta=(-Pi)/a;
source=J0*B0*exp(Eta*Yprim)*cos(fabs(Airfoil_Tet-Alpha));
dS[eqn]=J0*B0*Eta*exp(Eta*Yprim)*cos(fabs(Airfoil_Tet-Alpha));
return source;
}

/*Define Source Term; Y-Momentum Equation*/
DEFINE_SOURCE(xmom, c, t, dS, eqn)
{
real Z[ND_ND];
real Airfoil_Tet,Y_Airfoil;
real Xloc,Yloc,X,Y,R,Teta,Angle;
```

```
real Alpha,TIME,XDC,XC;
real source,Yprim,Eta;
FILE *fp2;
TIME=CURRENT_TIME;
Alpha=0.1745329-D*0.01745329*sin(2.0*K*TIME);
C_CENTROID(Z, c, t);
X=Z[0];
Y=Z[1];
R=sqrt(pow(X,2)+pow(Y,2));
Teta=atan(Y/X);
if (X<=0.0)
{
Angle=fabs(Teta)-fabs(Alpha);
Xloc=R*cos(Angle);
Xloc=-Xloc;
Yloc=R*sin(Angle);
}
if (X>0.0)
{
if (Teta<(15.77777-Alpha))
{
Angle=fabs(Teta)+fabs(Alpha);
Xloc=R*cos(Angle);
Yloc=R*sin(Angle);
}
if (Teta==(15.77777-Alpha))
{
Xloc=0.0;
Yloc=R;
}
if (Teta>(15.77777-Alpha))
{
Angle=fabs(Teta)-fabs(Alpha);
Xloc=R*cos(Angle);
Xloc=-Xloc;
Yloc=R*sin(Angle);
}
}
XDC=(Xloc+0.25)/Ch;
XC=(Xloc+0.25)*Ch;
Y_Airfoil=(Th*Ch/0.2)*(0.2969*pow(XDC,0.5)-0.1260*XDC-0.3516*pow(XDC,2)
+0.2843*pow(XDC,3)-0.1015*pow(XDC,4));
Airfoil_Tet=(Th*Ch/0.2)*(((0.2969/2.0)*sqrt(1.0/XC))-(0.1260/Ch)-(2.0*0.3516*XDC)
+(3.0*0.2843*pow(XDC,2))-(4.0*0.1015*pow(XDC,3)));
Yprim=Yloc-Y_Airfoil;
```

```
Eta=(-Pi)/a;
source=J0*B0*exp(Eta*Yprim)*sin(fabs(Airfoil_Tet-Alpha));
dS[eqn]=J0*B0*Eta*exp(Eta*Yprim)*sin(fabs(Airfoil_Tet-Alpha));
return source;
}

/*Motion of Oscillatory Airfoil*/
DEFINE_CG_MOTION(airfoil, dt, vel, omega, time, dtime)
{
Thread *t;
FILE *fp1;
int zone_id;
real alpha;
t=DT_THREAD(dt);
zone_id=THREAD_ID(t);
omega[2]=0.0174532*D*2.0*K*cos(2.0*K*time);
}
```