

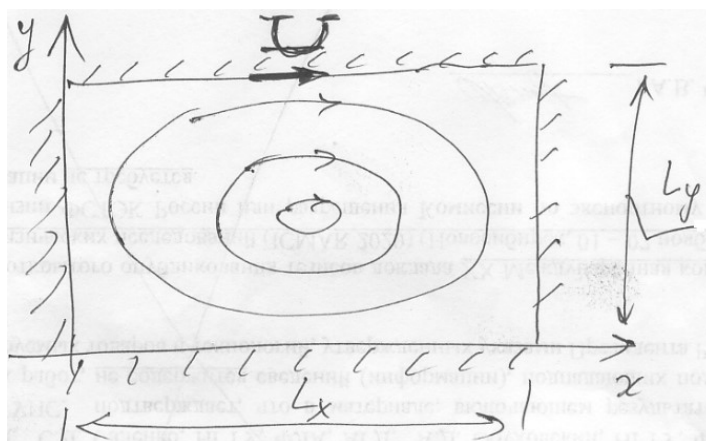
## Лабораторная работа №1

### «Обработка результатов CFD-эксперимента»

ЗАДАНИЕ. Обработать результаты расчета течения в каверне:

- построить изолинии модуля скорости  $V = |\vec{V}|$  и компонент скорости  $u$  и  $v$ , линии тока, поле векторов скорости;
- для заданной стенки построить графики коэффициента трения  $c_f(x)$  ( $c_f(y)$ ) и коэффициента давления  $c_p(x)$  ( $c_p(y)$ );
- определить действующие на заданную стенку аэродинамические силы и их коэффициенты.

Схема течения приведена на рисунке. Во всех вариантах скорость верхней стенки  $U = 10^{-3}$  м/с,  $L_x = 1$  м, плотность жидкости  $\rho = 10^3$  кг/м<sup>3</sup>, динамическая вязкость жидкости  $\mu = 10^{-3}$  Па·с, число узлов сетки  $N_x \times N_y = 81 \times 81$ . Варианты заданий даны в таблице.



Вариант	$L_y/L_x$	стенка	Вариант	$L_y/L_x$	стенка
1	2/3	нижняя	2	3/2	правая
3	1/2	верхняя	4	2	левая
5	3/5	нижняя	6	5/3	правая
7	3/7	верхняя	8	7/3	левая
9	3/4	нижняя	10	4/3	правая
11	4/5	верхняя	12	5/4	левая
13	2/5	нижняя	14	5/2	правая
15	5/6	верхняя	16	6/5	левая
17	5/7	нижняя	18	7/5	правая
19	5/8	верхняя	20	8/5	левая
21	5/9	нижняя	22	9/5	правая
23	3/8	верхняя	24	8/3	левая

Исходные данные размещены в текстовых файлах «var1.txt», ..., «var24.txt» следующей структуры:

$$N_x \quad N_y$$

$$x_1 \quad x_2 \dots x_{N_x}$$

$$y_1 \quad y_2 \dots y_{N_y}$$

$$\bar{u}_{1,1} \quad \bar{u}_{2,1} \dots \bar{u}_{N_x,1}$$

$$\bar{u}_{1,2} \quad \bar{u}_{2,2} \dots \bar{u}_{N_x,2}$$

$$\dots$$

$$\bar{u}_{1,N_y} \quad \bar{u}_{2,N_y} \dots \bar{u}_{N_x,N_y}$$

$$\bar{v}_{1,1} \quad \bar{v}_{2,1} \dots \bar{v}_{N_x,1}$$

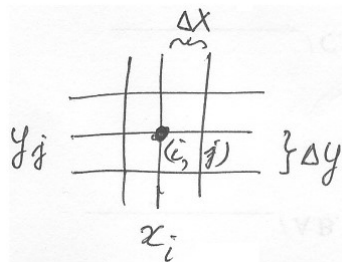
$$\bar{v}_{1,2} \quad \bar{v}_{2,2} \dots \bar{v}_{N_x,2}$$

$$\dots$$

$$\bar{v}_{1,N_y} \quad \bar{v}_{2,N_y} \dots \bar{v}_{N_x,N_y}$$

где  $\bar{u} = u/U$ ,  $\bar{v} = v/U$ .

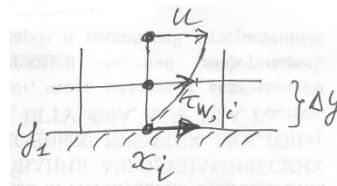
УКАЗАНИЯ. Пусть в узлах  $\{(x_i, y_j)\}$ ,  $i = 1, \dots, N_x$ ,  $j = 1, \dots, N_y$  равномерной разностной сетки задано поле скоростей  $\{(u_{i,j}, v_{i,j})\}$  несжимаемого вязкого течения.



Тогда можно определить:

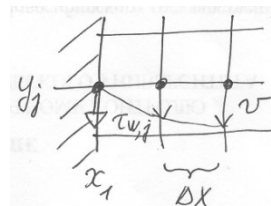
- касательное напряжение на стенке

$$\tau_{w,i} = \mu \left( \frac{\partial u}{\partial y} \right)_{i,1},$$



или

$$\tau_{w,j} = \mu \left( \frac{\partial v}{\partial x} \right)_{1,j},$$



где  $\mu$  – динамическая вязкость жидкости;

- давление на стенке

$$p_{w,i} = \mu \left( \frac{\partial v}{\partial y} \right)_{i,1} + const,$$

или

$$p_{w,j} = \mu \left( \frac{\partial u}{\partial x} \right)_{1,j} + const;$$

- коэффициент трения

$$c_f = \tau_w / (\rho U^2 / 2),$$

где  $\rho$  – плотность жидкости,  $U$  – характерная скорость;

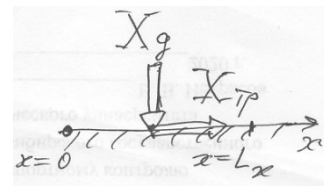
- коэффициент давления

$$c_p = (p_w - p_x) / (\rho U^2 / 2),$$

где  $p_x$  – характерное давление;

- действующие на стенку силы трения и давления

$$X_{\text{тр}} = \int_0^{L_x} \tau_w(x) dx, \quad X_d = \int_0^{L_x} p_w(x) dx$$



и их коэффициенты

$$c_{X_{\text{тр}}} = X_{\text{тр}} / (S \rho U^2 / 2), \quad c_{X_d} = X_d / (S \rho U^2 / 2),$$

где  $S$  – характерная площадь.

Для расчета значений частных производных на стенках (границах области) используются несимметричные формулы конечных разностей:

1-го порядка

$$\left( \frac{\partial u}{\partial y} \right)_{i,j} \approx \frac{u_{i,j+1} - u_{i,j}}{\Delta y}, \quad (1)$$

$$\left( \frac{\partial u}{\partial y} \right)_{i,j} \approx \frac{u_{i,j} - u_{i,j-1}}{\Delta y}, \quad (2)$$

2-го порядка

$$\left( \frac{\partial u}{\partial y} \right)_{i,j} \approx \frac{-3u_{i,j} + 4u_{i,j+1} - u_{i,j+2}}{2\Delta y}, \quad (3)$$

$$\left( \frac{\partial u}{\partial y} \right)_{i,j} \approx \frac{3u_{i,j} - 4u_{i,j-1} + u_{i,j-2}}{2\Delta y}. \quad (4)$$

Для расчета интегралов – формулы численного интегрирования, например, формула трапеций.

Для визуализации картины течения строятся изолинии скорости, поле векторов скорости, линии тока. Пример программы обработки приведен ниже.

```
program POST1 ! Постпроцессор
use GrafLib
implicit real*8(a-h,o-z)
parameter(nc=10,Npl=21) ! nc-кол-во уровней, Npl-кол-во линий тока
dimension z(nc),xpl(Npl),ypl(Npl)
allocatable:: x(:),y(:),u(:,,:),v(:,,:)
open(10,file='var0.txt')
read(10,*)Nx,Ny
```

```

allocate(x(Nx),y(Ny),u(Nx,Ny),v(Nx,Ny))
read(10,*)x
read(10,*)y
read(10,*)(u(:,j),j=1,Ny)
read(10,*)(v(:,j),j=1,Ny)
aLmax=dmax1(x(Nx),y(Ny))
print*, 'Flow statistics:'
xmin=x(1);print*, ' xmin=',xmin
xmax=x(Nx);print*, ' xmax=',xmax
ymin=y(1);print*, ' ymin=',ymin
ymax=y(Ny);print*, ' ymax=',ymax
umin=minval(u);print*, ' umin=',umin
umax=maxval(u);print*, ' umax=',umax
vmin=minval(v);print*, ' vmin=',vmin
vmax=maxval(v);print*, ' vmax=',vmax
! графика
xgmin=0.; xgmax=aLmax
ygmin=0.; ygmax=aLmax
call GRAFINIT(1, 'Contours')
ii=clickmenuqq(QWIN$TILE)
call PLOT2((/0.0d0,xmax,xmax,0.0d0,0.0d0/), & ! границы области
(/0.0d0,0.0d0,ymax,ymax,0.0d0/),0,1000,2d-3)
! изолинии V
Vmax=maxval(sqrt(u**2+v**2))
z=Vmax*dble((/0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.99/)) ! массив
уровней
call CPLOT(sqrt(u**2+v**2),1,Nx,1,Ny,x,y,nc,z)
! изолинии u
!z=dble((/-0.3,0.0,0.2,0.6,1.0/))
!call CPLOT(u,1,Nx,1,Ny,x,y,nc,z)
! изолинии v
!z=dble((/-0.4,-0.2,-0.1,0.05,0.17/))
!call CPLOT(v,1,Nx,1,Ny,x,y,nc,z)
ii=SAVEIMAGE_W('contours.bmp',& ! сохранение на диск
xgmin-0.1,ygmax+0.1,xgmax+0.1,ygmin-0.1)
call GrafInit(2, 'Velocity Vectors')
ii=clickmenuqq(QWIN$TILE)
ig=setactiveqq(2)
call PLOT2((/0.0d0,xmax,xmax,0.0d0,0.0d0/), & ! границы области
(/0.0d0,0.0d0,ymax,ymax,0.0d0/),0,1000,2d-3)
call Vectors1(Nx,Ny,x,y,u,v,0.04d0,1,3,1d-3,1) ! векторное поле
ii=SAVEIMAGE_W('vectors.bmp',xgmin,ygmax,xgmax,ygmin)
call GrafInit(3, 'Pathlines')
ii=clickmenuqq(QWIN$TILE)
ig=setactiveqq(3)
call PLOT2((/0.0d0,xmax,xmax,0.0d0,0.0d0/), & ! границы области
(/0.0d0,0.0d0,ymax,ymax,0.0d0/),0,1000,2d-3)
xpl=0.5*xmax ! xpl=const
do j=1,Npl; ypl(j)=ymin+(j-1)*(ymax-ymin)/(Npl-1); end do
!ypl=0.5*ymax ! ypl=const
!do i=1,Npl; xpl(i)=xmin+(i-1)*(xmax-xmin)/(Npl-1); end do
call PLINES(Nx,Ny,x,y,u,v,Npl,xpl,ypl,1d-2,10000) ! линии тока
ii=SAVEIMAGE_W('plines.bmp',xgmin,ygmax,xgmax,ygmin)
pause
end

subroutine CPLOT(d,ilb,iub,jlb,jub,x,y,nc,z)
real*8 d(ilb:iub,jlb:jub)

```

```

real*8 x(ilb:iub)
real*8 y(jlb:jub)
real*8 z(1:nc)
real*8 h(0:4),x1,x2,y1,y2
integer sh(0:4)
real*8 xh(0:4),yh(0:4)
integer im(1:4),jm(1:4)
integer cas
integer castab(-1:1,-1:1,-1:1)
integer p1,p2; logical SOLID
data im/0,1,1,0/
data jm/0,0,1,1/
data castab/0,0,9, 0,1,5, 7,4,8, &
0,3,6, 2,3,2, 6,3,0, &
8,4,7, 5,1,0, 9,0,0/
xsect(p1,p2) = (h(p2)*xh(p1)-h(p1)*xh(p2))/(h(p2)-h(p1))
ysect(p1,p2) = (h(p2)*yh(p1)-h(p1)*yh(p2))/(h(p2)-h(p1))
20 do 100 j=jub-1,jlb,-1
do 90 i=ilb,iub-1
dmin = min(d(i,j),d(i,j+1),d(i+1,j),d(i+1,j+1))
dmax = max(d(i,j),d(i,j+1),d(i+1,j),d(i+1,j+1))
if (dmax.ge.z(1) .and. dmin.le.z(nc)) then
do 80 k=1,nc
if (z(k).ge.dmin .and. z(k).le.dmax) then
do 22 m=4,0,-1
if (m.gt.0) then
h(m)=d(i+im(m),j+jm(m))-z(k)
xh(m)=x(i+im(m))
yh(m)=y(j+jm(m))
else
h(0)=0.25*(h(1)+h(2)+h(3)+h(4))
xh(0)=0.5*(x(i)+x(i+1))
yh(0)=0.5*(y(j)+y(j+1))
endif
if (h(m).gt.0.0) then
sh(m)=+1
else if (h(m).lt.0.0) then
sh(m)=-1
else
sh(m)=0
endif
22 continue
do 60 m=1,4
m1=m
m2=0
if (m.ne.4) then
m3=m+1
else
m3=1
endif
cas = castab(sh(m1),sh(m2),sh(m3))
if (cas.ne.0) then
goto (31,32,33,34,35,36,37,38,39),cas
31 x1=xh(m1)
y1=yh(m1)
x2=xh(m2)
y2=yh(m2)
goto 40

```

```

32 x1=xh(m2)
y1=yh(m2)
x2=xh(m3)
y2=yh(m3)
goto 40
33 x1=xh(m3)
y1=yh(m3)
x2=xh(m1)
y2=yh(m1)
goto 40
34 x1=xh(m1)
y1=yh(m1)
x2=xsect(m2,m3)
y2=ysect(m2,m3)
goto 40
35 x1=xh(m2)
y1=yh(m2)
x2=xsect(m3,m1)
y2=ysect(m3,m1)
goto 40
36 x1=xh(m3)
y1=yh(m3)
x2=xsect(m1,m2)
y2=ysect(m1,m2)
goto 40
37 x1=xsect(m1,m2)
y1=ysect(m1,m2)
x2=xsect(m2,m3)
y2=ysect(m2,m3)
goto 40
38 x1=xsect(m2,m3)
y1=ysect(m2,m3)
x2=xsect(m3,m1)
y2=ysect(m3,m1)
goto 40
39 x1=xsect(m3,m1)
y1=ysect(m3,m1)
x2=xsect(m1,m2)
y2=ysect(m1,m2)
goto 40
40 if(SOLID((x1+x2)/2,(y1+y2)/2))cycle
call vecout(x1,y1,x2,y2,k)
endif
60 continue
endif
80 continue
endif
90 continue
100 continue
End

```

```

subroutine vecout(x1,y1,x2,y2,nz)
use GrafLib
implicit real*8(a-h,o-z)
select case(nz)
case(1);icol=1
case(2);icol=9
case(3);icol=3

```

```

case(4);icol=11
case(5);icol=2
case(6);icol=10
case(7);icol=5
case(8);icol=13
case(9);icol=6
case(10);icol=4
end select
call PLOT2((/x1,x2/),(/y1,y2/),icol,50,2d-3)
end

```

```

subroutine Vectors1(Nx,Ny,x,y,Vx,Vy,c,icol,ivec,dr,ivar)
use Graflib
implicit real*8(a-h,o-z)
parameter (angle=0.3,tip=0.40)
dimension x(Nx),y(Ny),Vx(Nx,Ny),Vy(Nx,Ny)
logical SOLID
Vmax = sqrt(maxval(Vx**2+Vy**2)); np=50
do i=2,Nx-1,ivec
do j=2,Ny-1,ivec
if(SOLID(x(i),y(j))) then
cycle
else
if(ivar==1)then
Vmod=sqrt(Vx(i,j)**2+Vy(i,j)**2)
dx = c*Vx(i,j)/Vmod
dy = c*Vy(i,j)/Vmod
else
dx = c*Vx(i,j)/Vmax
dy = c*Vy(i,j)/Vmax
end if
! dx=c*Vx(i,j)/Vmax; dy=c*Vy(i,j)/Vmax ! -old
x2=x(i)+dx; y2=y(j)+dy
call Plot2((/x(i),x2/),(/y(j),y2/),icol,np,dr)
cs=cos(angle); sn=sin(angle)
dxi=-dx; dyi=-dy
x3=x2+( dxi*cs+dyi*sn)*tip
y3=y2+(-dxi*sn+dyi*cs)*tip
x4=x2+( dxi*cs-dyi*sn)*tip
y4=y2+( dxi*sn+dyi*cs)*tip
call Plot2((/x2,x3/),(/y2,y3/),icol,np,dr)
call Plot2((/x2,x4/),(/y2,y4/),icol,np,dr)
end if
end do
end do
end

```

```

subroutine PLINES(Nx,Ny,x,y,Vx,Vy,Npl,xpl,ypl,dt,Nt)
use Graflib
implicit real*8(a-h,o-z)
dimension x(Nx),y(Ny),Vx(Nx,Ny),Vy(Nx,Ny),xpl(Npl),ypl(Npl)
! расчет и построение ЛТ
do j=1,Npl ! цикл по номеру ЛТ
xpold=xpl(j);ypold=ypl(j) ! стартовая точка
do i=2,Nt ! интегрирование уравнений ЛТ
! метод Эйлера с пересчетом
call bilint(Nx,Ny,x,y,Vx,xpold,ypold,up)
call bilint(Nx,Ny,x,y,Vy,xpold,ypold,vp)

```

```

      xp1=xpold+dt*up
      yp1=ypold+dt*vp
      call bilint(Nx,Ny,x,y,Vx,xp1,yp1,up1)
      call bilint(Nx,Ny,x,y,Vy,xp1,yp1,vp1)
      xp=xpold+0.5*dt*(up+up1)
      yp=ypold+0.5*dt*(vp+vp1)
!   строим ЛТ
      call Plot((/xpold,xp/),(/ypold,yp/),1,0,0)
      xpold=xp
      ypold=yp
    end do
  end do
end

subroutine bilint(N1,N2,x,y,f,xint,yint,Fint) ! Билинейная
интерполяция
implicit real*8(a-h,o-z)
dimension x(N1),y(N2),f(N1,N2)
do i=1,N1-1
  do j=1,N2-1
    if((x(i)<=xint).and.(xint<=x(i+1)).and. &
      (y(j)<=yint).and.(yint<=y(j+1)))then
      S=f(i,j)*(x(i+1)-xint)*(y(j+1)-yint) &
        +f(i+1,j)*(xint-x(i))*(y(j+1)-yint) &
        +f(i,j+1)*(x(i+1)-xint)*(yint-y(j)) &
        +f(i+1,j+1)*(xint-x(i))*(yint-y(j))
      Fint=S/(x(i+1)-x(i))/(y(j+1)-y(j))
    end if
  end do
end do
end

logical function SOLID(x,y) ! внутренность тела
implicit real*8(a-h,o-z)
SOLID=.false.
!if((x>=0.5).and.(y>=0.3))SOLID=.true.
End

```

## Результаты для течения в квадратной каверне («var0.txt»):

```

Flow statistics:
  xmin=  0.0000000000000000E+000
  xmax=  1.0000000000000000
  ymin=  0.0000000000000000E+000
  ymax=  1.0000000000000000
  umin= -0.3583000000000000
  umax=  1.0000000000000000
  vmin= -0.6308000000000000
  vmax=  0.3493000000000000

```



