

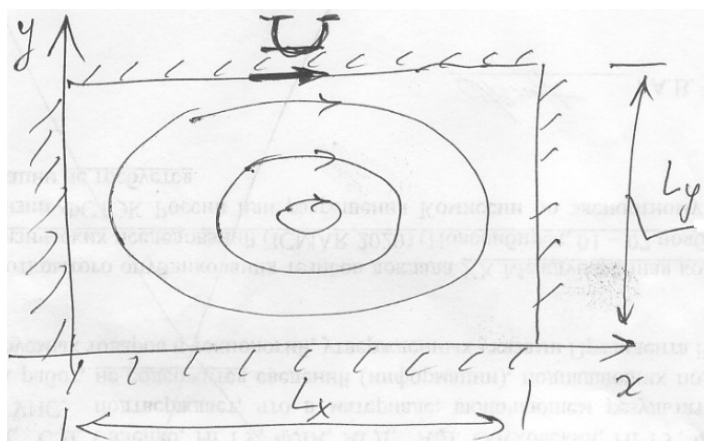
Лабораторная работа №1

«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 $c_f(x)$ или $c_f(y)$;
- определить действующую на заданную стенку силу трения и ее коэффициент.

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/м³, кинематическая вязкость жидкости $\nu = 10^{-6}$ м²/с, число узлов сетки $N_x \times N_y = 81 \times 81$. Варианты заданий даны в таблице.



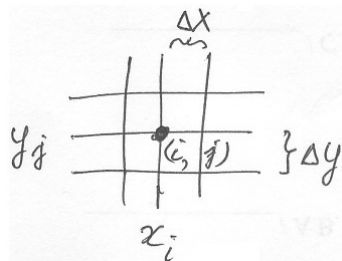
Вариант	L_y/L_x	стенка	Вариант	L_y/L_x	стенка
1	2/3	нижняя	2	3/2	правая
3	1/2	верхняя	4	2	левая
5	3/5	нижняя	6	5/3	правая
7	3/7	верхняя	8	7/3	левая
9	3/4	нижняя	10	4/3	правая
11	4/5	верхняя	12	5/4	левая
13	2/5	нижняя	14	5/2	правая
15	5/6	верхняя	16	6/5	левая
17	5/7	нижняя	18	7/5	правая
19	5/8	верхняя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

$$\begin{array}{l}
x_{1,1} \quad x_{1,2} \dots x_{1,N_x} \\
x_{2,1} \quad x_{2,2} \dots x_{2,N_x} \\
\dots \\
x_{N_y,1} \quad x_{N_y,2} \dots x_{N_y,N_x} \\
y_{1,1} \quad y_{1,2} \dots y_{1,N_x}, \\
y_{2,1} \quad y_{2,2} \dots y_{2,N_x} \\
\dots \\
y_{N_y,1} \quad y_{N_y,2} \dots y_{N_y,N_x} \\
\bar{u}_{1,1} \quad \bar{u}_{1,2} \dots \bar{u}_{1,N_x} \\
\bar{u}_{2,1} \quad \bar{u}_{2,2} \dots \bar{u}_{2,N_x} \\
\dots \\
\bar{u}_{N_y,1} \quad \bar{u}_{N_y,2} \dots \bar{u}_{N_y,N_x} \\
\bar{v}_{1,1} \quad \bar{v}_{1,2} \dots \bar{v}_{1,N_x} \\
\bar{v}_{2,1} \quad \bar{v}_{2,2} \dots \bar{v}_{2,N_x} \\
\dots \\
\bar{v}_{N_y,1} \quad \bar{v}_{N_y,2} \dots \bar{v}_{N_y,N_x}
\end{array}$$

где $\bar{u} = u/U$, $\bar{v} = v/U$.

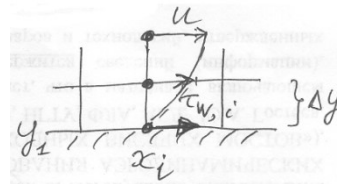
УКАЗАНИЯ. Пусть в узлах $\{(x_{j,i}, y_{j,i})\}$, $i = 1, \dots, N_x$, $j = 1, \dots, N_y$ равномерной разностной сетки задано поле скоростей $\{(u_{j,i}, v_{j,i})\}$ несжимаемого вязкого течения.



Тогда можно определить:

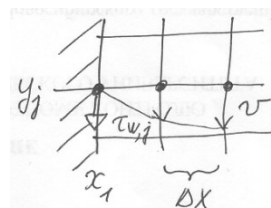
- касательное напряжение на стенке

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



или

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



где μ – динамическая вязкость жидкости;

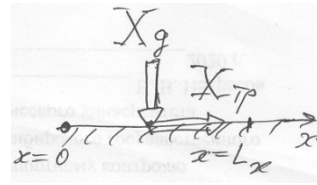
- коэффициент трения

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

где ρ – плотность жидкости, U – характерная скорость;

- действующую на стенку силу трения

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



и ее коэффициент

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

где S – характерная площадь.

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

1-го порядка

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

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

2-го порядка

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

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

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

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

```
import numpy as np
import matplotlib.pyplot as plt

data = np.loadtxt('var0.txt')
X,Y,U,V = np.split(data,4,axis=0)
print('x=',X,'\n','y=',Y,'\n','u/Uw=',U,'\n','v/Uw=',V)

anu, Uw, aLx, aLy = 1.e-6, 1e-3, 1.0, 1.0
ro = 1e3
amu = anu * ro
Nx, Ny = 81, 81

def Isolines(X,Y,Z,title):
    levels=np.linspace(np.min(Z),np.max(Z),10)
    plt.figure(figsize=(8,8))
    fig, ax = plt.subplots()
    ax.set_aspect('equal')
    plt.contour(X,Y,Z,levels=levels)
    plt.xlabel("x")
    plt.ylabel("y")
```

```

plt.title(title)
plt.colorbar()
plt.show()

Isolines(X,Y,U,'Изолинии u/Uw')
Isolines(X,Y,V,'Изолинии v/Uw')
Isolines(X,Y,np.sqrt(U**2+V**2),'Изолинии V/Uw')

plt.figure(figsize=(8,8))
fig, ax = plt.subplots()
ax.set_aspect('equal')
plt.streamplot(X, Y, U, V, density=2)
plt.show()

plt.figure(figsize=(8,8))
fig, ax = plt.subplots()
ax.set_aspect('equal')
step=4 # шаг построения векторов
plt.quiver(X[::step,::step],Y[::step,::step],U[::step,::step],V[::step,::step],units='x')
plt.show()

plt.plot(U[:,Nx//2+1],Y[:,0]/aLy,label='$u/U_{w}$')
plt.plot(V[:,Nx//2+1],Y[:,0]/aLy,label='$v/U_{w}$')
plt.legend(loc=4)
plt.xlabel('$u/U_{w}, v/U_{w}$'+' при '+'$x=L_{x}/2$')
plt.ylabel('$y/L_{y}$')
plt.grid(True)
plt.show()

dx=X[0,1]-X[0,0]
dy=Y[1,0]-Y[0,0]
dx1, dy1 = 2*dx, 2*dy
u, v = (np.zeros((Ny+1,Nx+1)) for k in range(2))
u[1:,1:] = U*Uw
v[1:,1:] = V*Uw

tauw = np.zeros(Nx+1)
tauw[1:Nx] = amu*(u[2,1:Nx]-u[1,1:Nx])/dy
plt.plot(X[0,:]/aLx,tauw[1:],label=r'$\tau_{w}$')
plt.legend(loc=4)
plt.xlabel('$x/L_{x}$')
plt.grid(True)
plt.show()

```

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

```

x= [[0.      0.0125 0.025   ... 0.975  0.9875 1.      ]
     [0.      0.0125 0.025   ... 0.975  0.9875 1.      ]
     [0.      0.0125 0.025   ... 0.975  0.9875 1.      ]
     ...
     [0.      0.0125 0.025   ... 0.975  0.9875 1.      ]
     [0.      0.0125 0.025   ... 0.975  0.9875 1.      ]
     [0.      0.0125 0.025   ... 0.975  0.9875 1.      ]]
y= [[0.      0.      0.      ... 0.      0.      0.      ]
     [0.0125 0.0125 0.0125 ... 0.0125 0.0125 0.0125]]

```

```

[0.025  0.025  0.025  ... 0.025  0.025  0.025 ]
...
[0.975  0.975  0.975  ... 0.975  0.975  0.975 ]
[0.9875 0.9875 0.9875 ... 0.9875 0.9875 0.9875]
[1.      1.      1.      ... 1.      1.      1.      ]]
u/Uw= [[ 0.00000000e+00  0.00000000e+00  0.00000000e+00 ...
0.00000000e+00
  0.00000000e+00  0.00000000e+00]
[ 0.00000000e+00  9.61942983e-05  3.81034481e-04 ...  4.19494804e-04
  6.12382556e-05  0.00000000e+00]
[ 0.00000000e+00  2.12078965e-04  6.89844579e-04 ...  1.01429938e-03
  2.63310240e-04  0.00000000e+00]
...
[ 0.00000000e+00 -2.30988534e-02 -2.94126490e-02 ...  2.48577965e-01
  8.44768049e-02  0.00000000e+00]
[ 0.00000000e+00  3.90832487e-02  9.13670551e-02 ...  4.63880279e-01
  3.45605802e-01  0.00000000e+00]
[ 1.00000000e+00  1.00000000e+00  1.00000000e+00 ...  1.00000000e+00
  1.00000000e+00  1.00000000e+00]]
v/Uw= [[ 0.00000000e+00  0.00000000e+00  0.00000000e+00 ...
0.00000000e+00
  0.00000000e+00  0.00000000e+00]
[ 0.00000000e+00 -9.60478697e-05 -2.10254418e-04 ...  2.49502717e-04
  5.76968127e-05  0.00000000e+00]
[ 0.00000000e+00 -3.81034481e-04 -6.85897472e-04 ...  9.93237174e-04
  4.19494804e-04  0.00000000e+00]
...
[ 0.00000000e+00  9.13670551e-02  1.03365771e-01 ... -2.18114627e-01
 -4.63880279e-01  0.00000000e+00]
[ 0.00000000e+00  9.92966115e-02  9.19460177e-02 ... -8.68663347e-02
 -2.99439839e-01  0.00000000e+00]
[ 0.00000000e+00  0.00000000e+00  0.00000000e+00 ...  0.00000000e+00
  0.00000000e+00  0.00000000e+00]]

```

