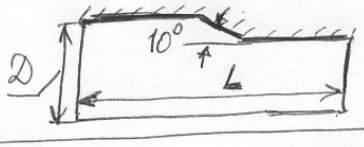
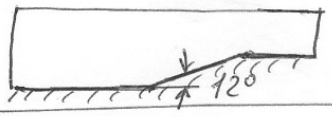
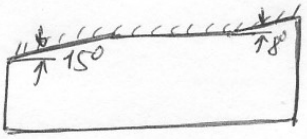
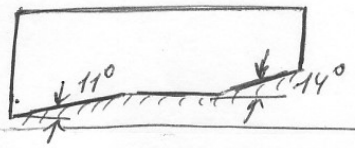
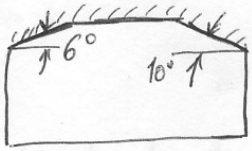
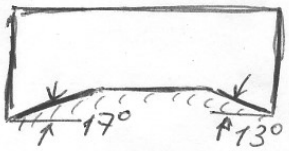
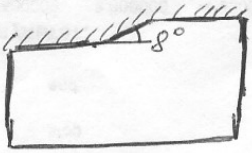
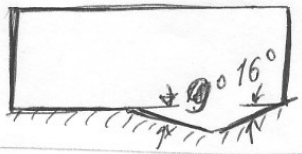
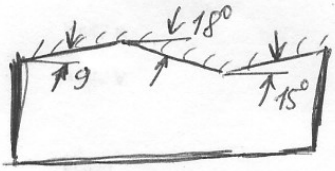
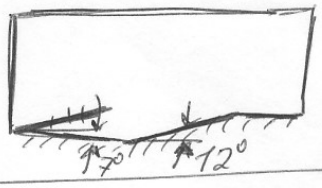
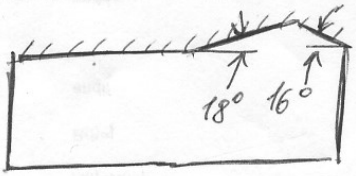
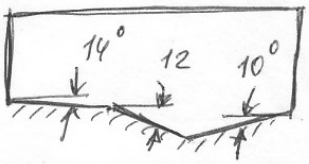
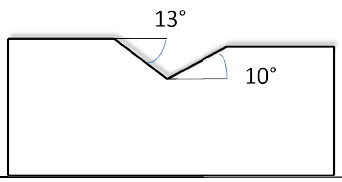
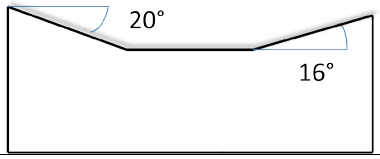
Лабораторная работа №3

«Генерация двумерной сетки»

ЗАДАНИЕ. Сгенерировать в заданной области структурированные конечно-разностную (КР) и конечно-объемную (КО) сетки. Варианты задания приведены в таблице:

Вар.	Схема области	Вар.	Схема области
1		2	
3		4	
5		6	
7		8	
9		10	
11		12	

№ вар.	Схема области	№ вар.	Схема области
13		14	

15		16	Перевернутая схема варианта №1, причем и верхняя и нижняя границы – стенки
17	Перевернутая схема варианта №2, причем и верхняя и нижняя границы – стенки	18	Перевернутая схема варианта №3, причем и верхняя и нижняя границы – стенки
19	Перевернутая схема варианта №4, причем и верхняя и нижняя границы – стенки	20	Перевернутая схема варианта №5, причем и верхняя и нижняя границы – стенки
21	Перевернутая схема варианта №6, причем и верхняя и нижняя границы – стенки	22	Перевернутая схема варианта №7, причем и верхняя и нижняя границы – стенки

Во всех вариантах $D = 1$ м, $L = 3D$.

УКАЗАНИЯ. 1) Для создания КР-сетки используйте как основу программу «meshgen2», которая генерирует структурированную сетку в расчетной области с верхней $y_{\text{в}}(x) = 0.5 + 0.1x^2$ и нижней $y_{\text{н}}(x) = 0.05 + 0.05\sin(3\pi x)$ границами типа «стенка» комбинацией метода сгущения к обеим границам

$$\begin{cases} x = \xi, \\ \dfrac{y}{H} = \dfrac{(2\alpha+\beta)B^{\frac{\eta-\alpha}{1-\alpha}} + 2\alpha - \beta}{(2\alpha+1)\left(B^{\frac{\eta-\alpha}{1-\alpha}} + 1\right)}, \quad B = \dfrac{\beta+1}{\beta-1}, \quad 1 < \beta < \infty \end{cases}$$

при $\alpha = 1/2$ ($\alpha = 0$ – сгущение только к верхней границе) и алгебраического метода

$$\begin{cases} x = x_{\text{н}}(\xi)(1-\eta) + x_{\text{в}}(\xi)\eta, \\ y = y_{\text{н}}(\xi)(1-\eta) + y_{\text{в}}(\xi)\eta. \end{cases}$$

Для сгущения сетки только к нижней границе используйте отображение

$$\begin{cases} x = \xi, \\ \dfrac{y}{H} = \dfrac{(\beta+1) - (\beta-1)B^{1-\eta}}{1 + B^{1-\eta}}. \end{cases}$$

Текст программы «meshgen2» приведен ниже.

```
from math import pi,sin
import matplotlib.pyplot as plt
import numpy as np

def ytop(x): # верх. граница
    ytop = 0.5+0.1*x**2
    return ytop

def ybot(x): # ниж. граница
    ybot = 0.05+0.05*sin(3*pi*x)
    return ybot

def MeshBotTop(Nx, Ny, xi, et, alf, bet, y): # сгущение к верх. и ниж. границам
    B = (bet + 1) / (bet - 1)
    for i in range(1,Nx+1):
```

```
        for j in range(1,Ny+1):
            etalf = (et[j] - alf) / (1 - alf)
            yH = ((bet+2*alf)*B**etalf+2*alf-bet)/(2*alf+1)/(1+B**etalf)
            y[i][j] = ybot(xi[i]) * (1 - yH) + ytop(xi[i]) * yH

def Jacobian(Nx, Ny, xi, et, x, y, Jac): # расчет якобиана J=xi_x*et_y
    for i in range(1,Nx+1):
        for j in range(1,Ny+1):
            xi_x = 1
            if j == 1:
                det = et[2] - et[1]
                dy = y[i][2] - y[i][1]
            elif j == Ny:
                det = et[Ny] - et[Ny-1]
                dy = y[i][Ny] - y[i][Ny-1]
            else:
                det = et[j+1] - et[j-1]
                dy = y[i][j+1] - y[i][j-1]
            et_y = det / dy
            Jac[i][j] = xi_x * et_y

Nx = 51
Ny = 51
x,xi,et = ([0] * (Nx+1) for k in range(3))
y,Jac = ([[0] * (Ny+1) for _ in range(Nx+1)] for k in range(2))
dxi = 1./(Nx-1)
for i in range(1,Nx+1):
    xi[i] = (i-1)*dxi
x = xi.copy()
det = 1./(Ny-1)
for j in range(1,Ny+1):
    et[j] = (j-1)*det
bet = 1.02
alf = 0.5

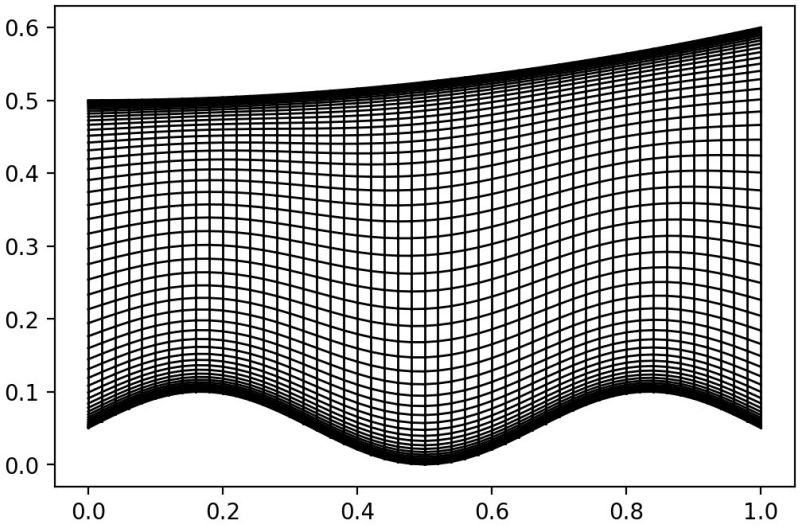
MeshBotTop(Nx, Ny, xi, et, alf, bet, y)
hxmin,hxmax = dxi,dxi
hymin,hymax = 1e3,-1e3
for i in range(1,Nx+1):
    for j in range(1,Ny):
        hy = y[i][j+1] - y[i][j]
        if hy < hymin: hymin = hy
        if hy > hymax: hymax = hy
print(' Mesh statistics:')
print(Nx*Ny, ' nodes')
print((Nx-1)*(Ny-1), ' cells')
print(' hx_min=', hxmin, ' hx_max=', hxmax)
print(' hy_min=', hymin, ' hy_max=', hymax)
hy1min = min(y[i][2]-y[i][1] for i in range(1,Nx+1))
hy1max = max(y[i][2]-y[i][1] for i in range(1,Nx+1))
print(' hy1_min=',hy1min,' hy1_max=',hy1max)
hx1hy1min = min(dxi/(y[i][2] - y[i][1]) for i in range(1,Nx+1))
hx1hy1max = max(dxi/(y[i][2] - y[i][1]) for i in range(1,Nx+1))
print(' (hx1/hy1)_min=',hx1hy1min,' (hx1/hy1)_max=',hx1hy1max)
Jacobian(Nx, Ny, xi, et, x, y, Jac)
Jmin = min(Jac[i][j] for j,i in zip(range(1,Ny+1),range(1,Nx+1)))
Jmax = max(Jac[i][j] for j,i in zip(range(1,Ny+1),range(1,Nx+1)))
print(' J_min=',Jmin)
print(' J_max=',Jmax)

lw=1
fig = plt.figure(dpi=200)
y1 = np.array(y)
# Горизонтальные линии (постоянный j)
for j in range(1, Ny + 1):
```

```
    plt.plot(x[1:], y1[1:,j], "k", linewidth=lw)
# Вертикальные линии (постоянный i)
for i in range(1, Nx + 1):
    x1=[x[i]]*(Ny)
    plt.plot(x1, y1[i,1:], "k", linewidth=lw)
plt.savefig('mesh2',dpi=fig.dpi,bbox_inches='tight',pad_inches=0.1)
plt.show()
```

Результаты запуска:

```
 Mesh statistics:
2601  nodes
2500  cells
 hx_min= 0.02  hx_max= 0.02
 hy_min=          0.0008067553212950246
hy_max= 0.02581754815411852
 hy1_min=         0.0008067553212950246
hy1_max= 0.0011019641866606908
 (hx1/hy1)_min=         18.14941015515803
(hx1/hy1)_max= 24.790663875505007
 J_min= 0.811555800237311
 J_max= 22.182612411859722
```



2) Для создания структурированной КО-сетки используйте модули DesignModeler и Meshing программного комплекса ANSYS. Изучите и повторите решение примера из пособия [1] (стр. 13-15) – см. ниже. Затем перейдите к построению расчетной области и сетки для вашего варианта. Расчетную область при этом разбивайте на блоки, каждый из которых может быть отображен на четырехугольник. Число разбиений граней задавайте согласованным с КР-сеткой.

Рассмотрим пример создания расчетной сетки в изображенной на рис. 1 области с размерами V1=1 м, V3=2 м, V5=1 м, H2=3 м, H4=2 м. Здесь верхняя граница является стенкой, нижняя – осью симметрии, левая и правая – входом и выходом соответственно.

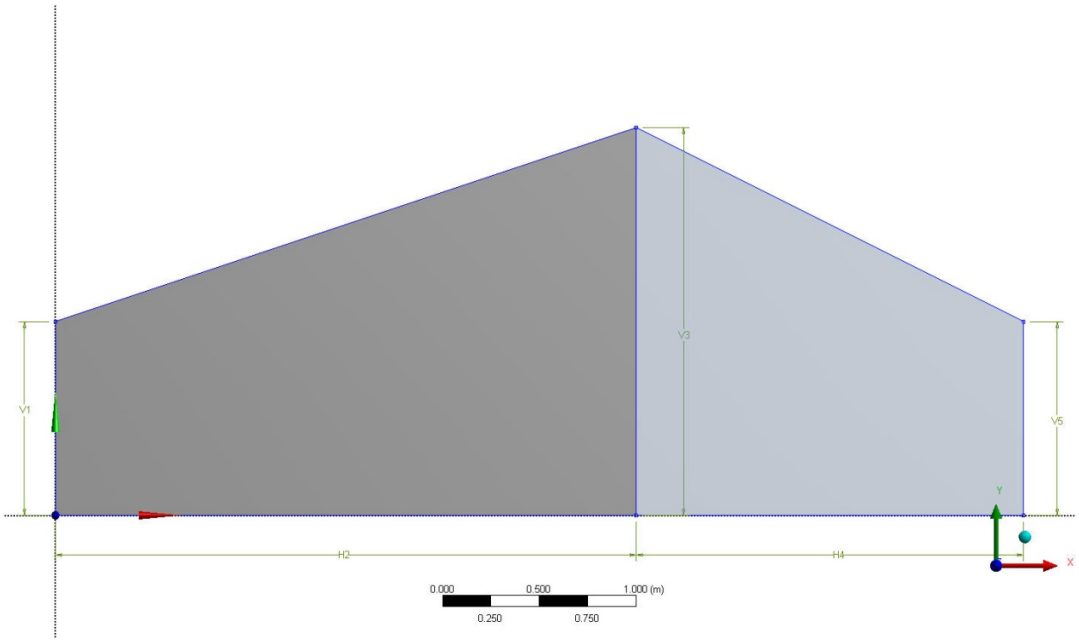


Рис. 1

1. Создание проекта (программа *Workbench*). После запуска *ANSYS Workbench* раскройте в «Toolbox» закладку «Analyses Systems» и выберите «Mesh» (двойной клик “мыши”, либо перетаскивание на поле «Project Schematic»). Сохраните проект (кнопка «Save») в специально созданной рабочей папке. Путь к рабочей папке не должен содержать букв кириллицы. В свойствах «Properties of Schematic A2: Geometry» раскройте пункт «Advanced Geometry Options» и выберите «Analysis Type» = «2D».

2. Создание геометрической модели (*DesignModeler*). Запустите *DesignModeler.*

Строим геометрическую модель в виде двух блоков четырехугольной формы (см. рис. 1). С этой целью создаем на плоскости XY **два** эскиза (чертежа): один – с **замкнутым** контуром (границей) левого блока, другой – правого блока. Затем с помощью инструмента «Concept» -> «Surfaces From Sketches» формируем поверхности. **Важно!** В свойствах «Details View» для второй поверхности указываем «Operation» = «Add Frozen». Созданные поверхностные тела («Surface Body») помещаем в одну часть (ПКМ (правая кнопка мыши) над выделенной группой, затем «Form New Part»), что обеспечит совпадение сеточных узлов на общей границе блоков.

3. Генерация структурированной сетки (*Meshing*). В пункте «Defaults» свойств сетки «Details of “Mesh”» задайте «Physics Preference» = «CFD», «Solver Preference» = «Fluent».

Выберите для обоих блоков отображаемую сетку («Mesh» (ПКМ) -> «Insert» -> «Mapped Face Meshing»).

На верхней и нижней границах левого блока задайте количество разбиений «Number of Divisions» = «100», на аналогичных границах правого – «70».

Вдоль вертикальных границ распределение узлов должно быть неравномерным, со сгущением к стенке. Поэтому для этих границ задайте «Number of Divisions» = «50», выберите подходящий тип сгущения «Bias Type» и введите «Bias Factor» = «10».

Нажмите кнопку «Generate Mesh». В разделе «Statistics» свойств сетки «Details of “Mesh”» выберите «Mesh Metric» = «Orthogonal Quality» и отобразите диаграмму распределения элементов сетки по ортогональности. Полученная сетка приведена на рис. 2.

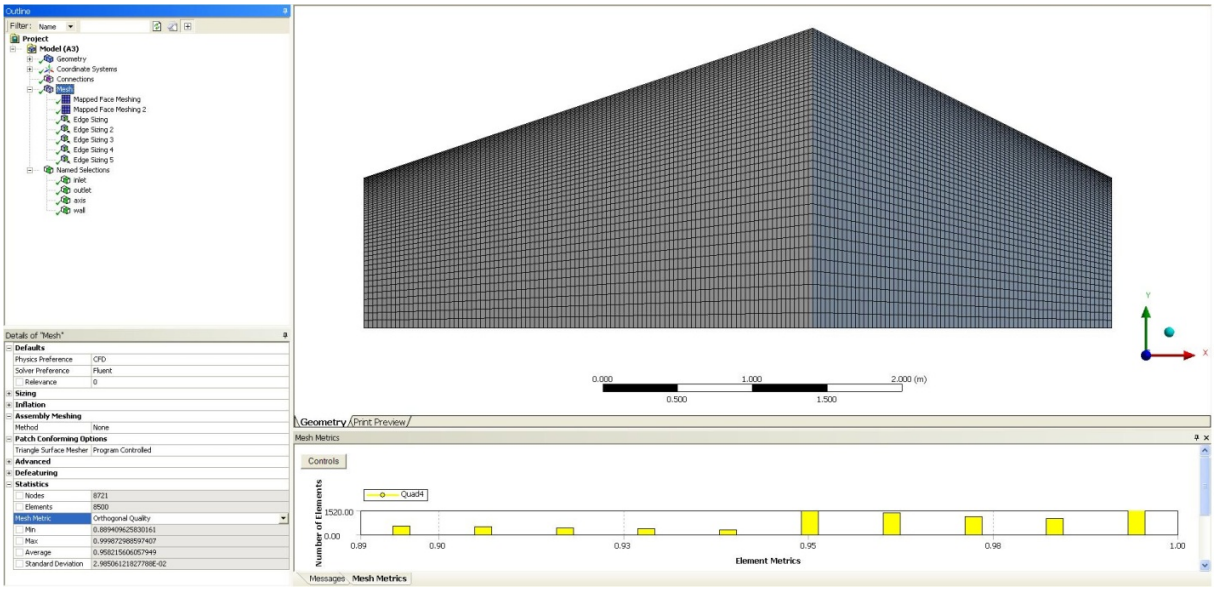


Рис. 2

Создайте граничные поверхности: выделяя граничное ребро (или группу ребер с помощью зажатой клавиши CTRL), вызывайте контекстное меню (ПКМ) -> «Create Named Selection». Входная граница («вход») должна именоваться «inlet», «выход» – «outlet», «стенка» – «wall», «ось» – «axis». Перед сохранением проекта выполните 

.

Далее выберите пункты меню «File» -> «Export», имя и тип файла «Fluent Input Files (*.msh)» и нажмите кнопку «Сохранить».

Литература

1. Гостеев Ю.А. Создание двумерных сеток в программе ANSYS MESHING. Лабораторный практикум: методическое пособие [для 3-4 курсов]. Новосибирск: Изд-во НГТУ, 2017.