

**МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
И МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ
ТЕПЛОГАЗОСНАБЖЕНИЯ И ВЕНТИЛЯЦИИ**

ОГЛАВЛЕНИЕ

1. Решение уравнений вида $f(x)=0$.
2. Численное интегрирование.
 - 2.1. Метод трапеций.
 - 2.2. Метод Симпсона.
 - 2.3. Метод Гаусса.
3. Численное решение обыкновенных дифференциальных уравнений и их систем.
 - 3.1. Метод Эйлера.
 - 3.2. Метод Рунге-Кутты.
4. Метод наложения потоков.
 - 4.1. Течения воздуха вблизи всасывающих отверстий в плоской стенке.
 - 4.2. Расчет экранированных местных отсосов.
5. Построение траекторий движения пылевой частицы в воздушном потоке.
 - 5.1. Полет в плоскости.
 - 5.2. Полет в пространстве.
6. Метод конформных отображений.
 - 6.1. Некоторые сведения о функциях комплексного переменного.
 - 6.2. Общие сведения о конформных отображениях.
 - 6.3. Щелевой отсос в неограниченном пространстве.
 - 6.4. Щелевой отсос в плоской стенке.
7. Метод Н.Е. Жуковского.
 - 7.1. Общие сведения о методе Н.Е. Жуковского.
 - 7.2. Щелевой отсос в неограниченном пространстве.
 - 7.3. Щелевой отсос в плоской стенке.
8. Метод ГИУ для расчета плоских потенциальных течений.
9. Лабораторные работы

1. Решение уравнений вида $f(x)=0$

Пусть на промежутке $[a,b]$ задана непрерывно возрастающая или непрерывно убывающая функция $f(x)$, имеющей на этом промежутке единственный корень. Т.е. на этом промежутке существует такое число x для которого справедливо равенство $f(x)=0$. Существуют различные методы определения корня: метод касательных, метод хорд, метод итераций и др. Наиболее распространен в инженерной практике метод половинного деления, который хотя и обладает медленной сходимостью, но не имеет ограничений при его применении. Кроме того, его недостаток компенсируется высоким быстродействием современных ЭВМ.

Рассмотрим применение метода половинного деления к задаче определения корня на промежутке. Разделим промежуток $[a,b]$ пополам, средину которого обозначим c (рис 1.1).

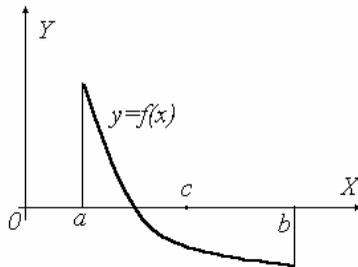


Рис.1.1

Выберем из полученных двух промежутков тот на котором есть корень. Если это промежуток $[a,c]$ то переместим точку b в точку c , если нет то точку a поместим в точку c . Затем опять разделим промежуток на котором существует корень точкой c пополам и вновь повторим вышеописанную процедуру. Так будем продолжать до тех пор пока величина промежутка $[a,b]$ будет меньше заданной точности.

Алгоритм вычисления корня будет иметь следующие шаги:

1. Вычисляются координаты точки c : $c = \frac{a+b}{2}$.
2. Если $f(a) - f(c) < 0$ (т.е. на промежутке $[a,c]$ есть корень) тогда необходимо определить координаты точки b по формуле $b = c$. Если

условие не выполняется то изменяются координаты точки a по формуле $a = c$.

3. Проверяется условие $|b - a| < \varepsilon$, где ε - заданная точность. При выполнении условия вычисления заканчиваются, а корень уравнения будет точка $c = \frac{a+b}{2}$. Если условие не выполняется то необходимо вернуться к пункту 1.

Замечание. Для того, чтобы применить этот алгоритм необходимо отделить корни уравнения. Т.е. найти промежутки, на которых существуют по одному корню. Это можно сделать графически, построив схематически график функции $y = f(x)$ либо аналитически. Для этого нужно задать некоторую точку x_0 на оси абсцисс и с малым шагом h начать двигаться вправо, проверяя каждый раз выполнение условия $f(x_0) - f(x_0 + h) < 0$. При выполнении условия один из промежутков будет найден.

Пример 1.1. Определить корни уравнения $x^3 + 12x - 2 = 0$ с точностью до 0,0001.

```

program otdlenie_kornej;
uses crt,printer; var a,b,x,h,c:real;
function f(x:real):real; begin f:=x*x*x+12*x-2 end;
{Процедура отделения корней}
procedure otdel(var a,b:real);
var x,h:real;
begin
x:=a; h:=0.1; {Начальное значение и шаг поиска}
while x<b do begin
if f(x)*f(x+h)<0 then begin a:=b; break;end; x:=x+h; end;
if a<>b then begin writeln('Корней нет'); halt end
else writeln('Границы промежутка',x:15:8,' ',(x+h):15:8);
writeln('f(a)*f(b)=',f(x)*f(x+h)); a:=x; b:=x+h;
end;
{Процедура вычисления корня c на промежутке [a,b]}
procedure dixotomi(a,b:real;var c:real);
begin
c:=(a+b)/2;

```

```

while abs(a-b)>0.00001 do
begin if f(a)*f(c)<0 then b:=c else a:=c; c:=(a+b)/2; end;
end;
{ Тело программы }
begin clrscr; a:=-10; b:=10; otdel(a,b); dixotomi(a,b,c);
writeln('x=',c:15:8,' f(x)=',f(c):15:8);
end.

```

Результаты счета: $x = 0,16628112$, $f(x) = -0,00002889$.

2. Численное интегрирование

2.1. Метод трапеций

Пусть на промежутке $[a, b]$ дана непрерывная функция $f(x)$ и необходимо вычислить интеграл $\int_a^b f(x)dx$. Поскольку геометрический смысл интеграла - это площадь криволинейной трапеции, ограниченной кривой $y=f(x)$ и прямыми $x=a$, $x=b$, $y=0$ то задача сводится к вычислению площади заштрихованной области рис 2.1.

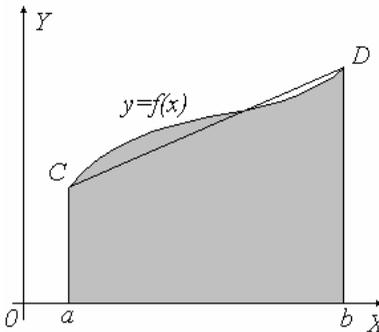


Рис.2.1.

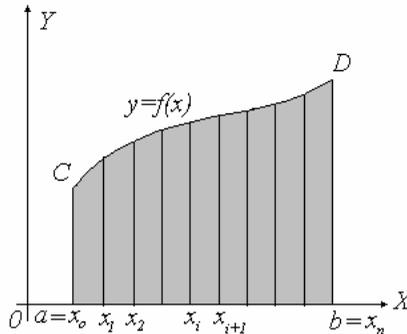


Рис.2.2.

Приближенно вычисляемый интеграл можно заменить площадью трапеции $abCD$

$$\int_a^b f(x)dx \approx S_{abCD} = \frac{|aC| + |bD|}{2} (b-a).$$

Учитывая, что $|aC| = f(a)$, $|bD| = f(b)$ и обозначая $h = b - a$ получим формулу трапеций

$$\int_a^b f(x)dx \approx S_{abCD} = \frac{f(a) + f(b)}{2} h.$$

Точность вычисления интеграла возрастает, если отрезок $[a, b]$ разделить на несколько частей и применить формулу трапеций к каждому отрезку X_i (рис.2.2).

Тогда

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx \frac{f(x_i) + f(x_{i+1})}{2} \Delta x_i.$$

Для простоты вычислений удобно делить отрезок $[a, b]$ на равные части, в этом случае длина каждого из отрезков разбиения есть $x_i = (b-a)/n$. Численное значение интеграла на отрезке (x_i, x_{i+1}) равно

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx \frac{b-a}{n} \cdot \frac{f(x_i) + f(x_{i+1})}{2},$$

а на всем отрезке $[a, b]$ соответственно

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \sum_{i=0}^{n-1} \frac{f(x_i) + f(x_{i+1})}{2} = \frac{b-a}{n} \sum_{i=0}^{n-1} \frac{y_i + y_{i+1}}{2}$$

Так как под знаком суммы величины y_i встречаются дважды (от $i = 1$ до $i = n-1$), то последнее равенство запишется следующим образом:

$$\int_a^b f(x)dx \approx \frac{b-a}{n} \left(\frac{y_0}{2} + y_1 + y_2 + \dots + y_{n-1} + \frac{y_n}{2} \right) = \frac{b-a}{n} \left[\sum_{i=1}^{n-1} y_i + \frac{y_0 + y_n}{2} \right]$$

Эта формула называется общей формулой трапеций.

Алгоритм численной реализации на языке Паскаль состоит из следующих этапов:

1. Описания переменных и функции интегрирования $f(x)$.
2. Ввода пределов интегрирования (a, b) и количества отрезков разбиения промежутка интегрирования n ;
3. Вычисления суммы

$$S = (f(x_1) + f(x_2) + \dots + f(x_{n-1}) + \frac{f(x_0) + f(x_n)}{2})h ;$$

4. Вывод на печать числа n и значения интеграла S .

Пример.2.1.

Составить программу вычисления интеграла $\int_a^b \frac{x - x^3}{x^4 + 2} dx$.

Предусмотреть ввод значения пределов интегрирования и количества отрезков разбиения с клавиатуры.

```

Program Трапеции;
uses crt;
var n,i:integer; a,b,h,s,x:real;

{Подинтегральная функция}
function f(x:real):real;
begin f:=(x-x*x*x)/(x*x*x*x+2) end;

begin
clrScr;
writeln('Ввести a,b,n'); readln(A,B,N);
{Ввод нижнего a и верхнего пределов интегрирования b, а также
количества отрезков разбиения b}
h:=(b-a)/n; {Шаг интегрирования}
x:=a;
s:=2*f(x); {Начальное значение суммы S}
for i:=1 to (n-1) do begin x:=x+h; s:=s+2*f(x); end;

```

```

s:=h/2*(s+f(b)); {Значение интеграла S}
writeln(n, ' ',s:0:6); {Вывод на экран результатов счета}
end.

```

2.2. Метод Симпсона (метод парабол)

Точность приближенного интегрирования заметно возрастает, если подинтегральную функцию $y = f(x)$ на отрезке $[a, b]$ заменить квадратичной функцией $y = \varphi(x)$ (рис.2.3), принимающей в узлах $x_0 = a$, x_1 , $x_2 = b$ значения $y_0 = f(x_0)$, $y_1 = f(x_1)$ и $y_2 = f(x_2)$.

Таким образом, будем иметь

$$\int_{x_0}^{x_2} f(x) dx \approx \int_{x_0}^{x_2} \varphi(x) dx . \quad (2.1)$$

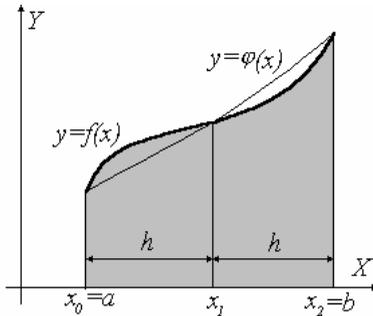


Рис.2.3.

Параболу $\varphi(x)$ представим в виде

$$\varphi(x) = A(x - x_0)^2 + B(x - x_0)(x - x_1) + C . \quad (2.2)$$

Если $x = x_0 \Rightarrow \varphi(x_0) = f(x_0) = C$.

Если

$$\begin{aligned} x = x_1 \Rightarrow \varphi(x_1) = f(x_1) &= A(x_1 - x_o)^2 + C = \\ &= Ah^2 + f(x_o) \Rightarrow A = \frac{f(x_1) - f(x_o)}{h^2}. \end{aligned}$$

Если

$$\begin{aligned} x = x_2 \Rightarrow \varphi(x_2) = f(x_2) &= A(x_2 - x_o)^2 + B(x_2 - x_o)(x_2 - x_1) + C = \\ &= \frac{f(x_1) - f(x_o)}{h^2} 4h^2 + B2h^2 + f(x_o) \Rightarrow B = \frac{f(x_2) - 4f(x_1) + 3f(x_o)}{2h^2}. \end{aligned}$$

Подставляя полученные значения постоянных A, B, C в (2.2), а затем в (2.1) получим следующее выражение

$$\begin{aligned} \int_a^b f(x)dx &\approx \int_a^b [A(x - x_o)^2 + B(x - x_o)(x - x_1) + C]dx = \\ &= \frac{h}{3} [f(x_o) + 4f(x_1) + f(x_2)]. \end{aligned} \quad (2.3)$$

Для повышения точности вычислений отрезок $[a, b]$ разбивают на n пар участков (рис.5) и к каждому применяют формулу Симпсона (2.3)

Тогда численное значение определенного интеграла на отрезке $[a, b]$ будет равно сумме интегралов, т.е.

$$\begin{aligned} \int_a^a f(x)dx &\approx \frac{h}{3} [f(x_0) + 2[f(x_2) + f(x_4) + \dots + f(x_{2n-2})] + \\ &+ 4[f(x_1) + f(x_3) + \dots + f(x_3) + \dots + f(x_{2n-1}) + f(x_{2n})]. \end{aligned} \quad (2.4)$$

Соотношение (2.4) называется общей формулой Симпсона.

Алгоритм численной реализации данного метода состоит из следующих этапов:

1. Описания переменных и подинтегральной функции;
2. Ввода пределов интегрирования a, b и количества отрезков разбиения n ;
3. Вычисления шага интегрирования

$$h = \frac{b-a}{2n} ;$$

4. Вычисления суммы S ;
5. Вывода на печать S и n .

Пример.2.2. Составить программу для вычисления интеграла

$$\int_1^{2.2} \frac{5,2x + 4,8}{4,5x^2 + 4,1} dx .$$

```

program simpson;
uses crt,printer;
var a,b,h,s,x:real; i,n:integer;

function f(x:real):real; {Подинтегральная функция}
begin f:=(5.2*x+4.8)/(4.5*x*x+4.1) end;

begin
clrscr;
a:=1; b:=2.2; {Нижний и верхний пределы интегрирования}
n:=12; {Количество отрезков разбиения}
h:=(b-a)/n/2; {Шаг интегрирования}
s:=0;{Начальное значение суммы}
x:=a; {Начальное значение переменной x}
for i:=1 to n do begin x:=x+2*h; s:=s+2*f(x)+4*f(x-h); end;
s:=h/3*(f(a)-f(b)+s); {Значение интеграла}
writeln('Вычисление интеграла по методу Симпсона');
writeln('n=',n);
writeln('Значение интеграла равно',s:0:6);
end.

```

2.3. Метод Гаусса

Для вычисления интегралов с повышенной точностью используется квадратурная формула Гаусса

$$\int_{-1}^1 f(x) dx = \sum_{k=1}^n f(x_k) A_k , \quad (2.5)$$

где x_k - узлы квадратурной формулы Гаусса; A_k – весовые множители; n – количество узлов.

Узлы и весовые множители берутся из специальных таблиц []. В табл. 2.1 представлены значения x_k, A_k для некоторых n .

Таблица 2.1

Весовые множители и узлы квадратурной формулы Гаусса

i	x_i	A_i
1	0,090617984593866399	0,23692688505618908751
2	0,538493101568309103	0,47862867049936645690
3	0.000000000000000000	0,5688888888888888889
4	-0,090617984593866399	0,23692688505618908751
5	-0,538493101568309103	0,47862867049936645690

Для того, чтобы вычислить интеграл с произвольными границами интегрирования необходимо его свести к $\int_{-1}^1 f(x)dx$ путём замены переменной.

$$\int_b^a f(t)dt = \left\{ t = \frac{a+b}{2} + \frac{b-a}{2}x \right\} = \frac{b-a}{2} \int_{-1}^1 f(x)dx \quad (2.6)$$

Т.е. необходим пересчет узлов квадратурной формулы Гаусса

$$x_i = \frac{a+b}{2} + \frac{b-a}{2}x_i, \quad i = 1, \dots, n$$

Вычисленная для полученных узлов x_i по формуле (2.5) сумма будет искомым интегралом, если ее умножить на $(b-a)/2$.

Пример.2.3. Вычислить интеграл $\int_0^{\sqrt{2}/2} \frac{\sqrt{2}}{2\pi} \left(x \cdot \arctg\left(\frac{\sqrt{2}-x}{x}\right) - \frac{x\pi}{4} \right) dx$

{ \$N+ }

program Gauss;

```
type vek=array[1..2]of real; reafun=function(x:real):real;
var h,a,b,x,integral:real; i,n,m:byte;
```

```
{ $F+ }
function podint (x:real):real;
begin          {Подинтегральная функция}
podint:=sqrt(2)/2/pi*(x*arctan((sqrt(2)-x)/x)-x*pi/4)
end;
{ $F- }
```

```
{Процедура вычисления интеграла по методу Гаусса}
procedure gaussint(n:byte;niz,ver:real;f:reafun;var int:real);
var a,x:array[1..20]of real; j:byte;
begin
if n=2 then
begin x[1]:=0.577350269189625764509148780502; a[1]:=1.0; end;
if n=4 then
begin
x[1]:=0.861136311594052575223946488893;
x[2]:=0.339981043584856264802665759103;
a[1]:=0.347854845137453857373063949222;
a[2]:=0.652145154862546142626936050778
end;
if n=6 then
begin
x[1]:=0.932469514203152027812301554494;
x[2]:=0.661209386466264513661399595020;
x[3]:=0.238619186083196908630501721681;
a[1]:=0.171324492379170345040296142173;
a[2]:=0.360761573048138607569833513838;
a[3]:=0.467913934572691047389870343990
end;
if n=8 then
begin
x[1]:=0.960289856497536231683560868569;
x[2]:=0.796666477413626739591553936476;
x[3]:=0.525532409916328985817739049189;
x[4]:=0.183434642495649804939476142360;
a[1]:=0.101228536290376259152531354310;
```

```

a[2]:=0.222381034453374470544355994426;
a[3]:=0.313706645877887287337962201987;
a[4]:=0.362683783378361982965150449277
end;
if n=10 then
begin
x[1]:=0.973906528517171720077964012084;
x[2]:=0.865063366688984510732096688423;
x[3]:=0.679409568299024406234327365115;
x[4]:=0.433395394129247190799265943166;
x[5]:=0.148874338981631210884826001130;
a[1]:=0.0666713443086881375935688098933;
a[2]:=0.149451349150580593145776339658;
a[3]:=0.219086362515982043995534934228;
a[4]:=0.269266719309996355091226921569;
a[5]:=0.295524224714752870173892994651
end;
    for j:=1 to n do
        begin x[n-(j-1)]:=-x[j]; a[n-(j-1)]:=a[j] end;
    for j:=1 to n do x[j]:=(ver+niz)/2+(ver-niz)/2*x[j];
    int:=0;
    for j:=1 to n do int:=int+a[j]*f(x[j]);
int:=(ver-niz)/2*int;
end;

begin
a:=0; {Нижний предел интегрирования}
b:=sqrt(2)/2; {Верхний предел интегрирования}
n:=10; {Количество узлов квадратурной формулы Гаусса}
m:=100;{Количество отрезков разбиения}
h:=(b-a)/m; {Шаг интегрирования}
x:=0;
for i:=1 to m do begin gaussint(n,a,a+h,podint,integral);
a:=a+h; x:=x+integral; end;
writeln('Интеграл=',x:0:5)
end.

```

Результаты расчета : интеграл равен 0,01727. Причем значение не изменяется при увеличении m от 1 до 200.

Пример.2.4. Вычислить интеграл $\int_2^{4.4} \frac{x-x^3}{x^4+2} dx$ различными методами

```

program integrirovanie;
const n=8; var a,b,x,integral,st,sa,ss:real;

function f(x:real):real;
begin          {Подинтегральная функция};
f:=(x-x*x*x)/(x*x*x*x+2) end;

procedure trapecii(a,b:real;n:integer;var s:real);
var h,x:real; i:integer;
begin h:=(b-a)/n; x:=a; s:=2*f(x);
for i:=1 to (n-1) do begin x:=x+h; s:=s+2*f(x); end; s:=h/2*(s+f(b)); end;

procedure simpson(a,b:real; N:INTEGER; var s:real);
var h,x:real; i:integer;
begin h:=(b-a)/n/2; s:=0; x:=a;
for i:=1 to n do begin x:=x+2*h; s:=s+2*f(x)+4*f(x-h); end;
s:=h/3*(f(a)-f(b)+s);
end;

{Процедура вычисления по методу Гаусса}
procedure gaussint(niz,ver:real;var int:real);
var
a,x:array[1..20]of real;
j:byte;
begin
x[1]:=0.960289856497536231683560868569;
x[2]:=0.796666477413626739591553936476;
x[3]:=0.525532409916328985817739049189;
x[4]:=0.183434642495649804939476142360;
a[1]:=0.101228536290376259152531354310;
a[2]:=0.222381034453374470544355994426;
a[3]:=0.313706645877887287337962201987;
a[4]:=0.362683783378361982965150449277;
for j:=1 to n do begin x[n-(j-1)]:=-x[j]; a[n-(j-1)]:=a[j] end;

```

```

for j:=1 to n do x[j]:=(ver+niz)/2+(ver-niz)/2*x[j];
int:=0; for j:=1 to n do int:=int+a[j]*f(x[j]); int:=(ver-niz)/2*int;
end;

```

```

begin
{ Тело программы }
a:=2; { Нижний предел интегрирования }
b:=4.4; { Верхний предел интегрирования }
sa:=1/(2*sqrt(2))*arctan(b*b/sqrt(2))-1/4*ln(b*b*b*b+2)-
1/(2*sqrt(2))*arctan(a*a/sqrt(2))+1/4*ln(a*a*a*a+2); { Аналитическое
значение интеграла }
gaussint(a,b,integral); { Вычисление интеграла по методу Гаусса }
trapecii(a,b,10,st); { Вычисление интеграла по методу трапеций }
simpson(a,b,10,ss); { Вычисление интеграла по методу Симпсона }
writeln('Интеграл=',integral:0:6,' ',st:0:6,' ',ss:0:6,' ',sa:0:6)
end.

```

Результаты расчета по методу Гаусса совпадает со значением интеграла вычисленного аналитически : -0,665972. По методу Симпсона интеграл равен -0,665971. Наибольшее отличие от аналитического значения у метода трапеций: -0,705684.

3. Численное решение обыкновенных дифференциальных уравнений и их систем

3.1. Методы Эйлера и Рунге-Кутта для одного дифференциального уравнения

Пусть дано обыкновенное дифференциальное уравнение (ОДУ) первого порядка

$$\frac{dy}{dx} = f(x, y) \quad (3.1)$$

при начальных условиях $x = x_0 \Rightarrow y = y_0$.

Сделаем небольшой шаг по x : $x = x_0 + h$ и запишем приближенный аналог уравнения (3.1)

$$\frac{y - y_0}{x - x_0} = f(x_0, y_0)$$

Учитывая, что $x - x_0 = h$ получим следующее уравнение

$$y = y_0 + f(x_0, y_0) \cdot h, \quad (3.2)$$

которое является основой для метода Эйлера.

Метод Эйлера имеет следующие этапы:

1. Делается шаг по x : $x = x_0 + h$.
2. Вычисляется y по формуле (3.2).
3. Вновь делается шаг по x : $x = x + h$.
4. Определяется y по формуле (3.2), где вместо x_0 , y_0 подставляются x и найденное на предыдущем этапе y .
5. Переход к п.3. Расчет производится до тех пор пока x не достигнет заданного значения.

Так как (3.2) представляет собой уравнение касательной метод Эйлера имеет наглядный геометрический смысл. В точке x_0 строится касательная, вдоль которой производится небольшой шаг. В полученной точке также строится касательная и вновь вдоль нее делается шаг и так до тех пор пока независимая переменная x не достигнет заданного значения.

Чем меньше шаг h , тем точнее вычисленное значение y .

Более точным и наиболее распространенным в инженерной практике является метод Рунге-Кутты, который имеет следующие этапы:

1. По данным начальным условиям задачи (3.1) вычисляются коэффициенты

$$\begin{aligned} k_1 &= h \cdot f(x_0, y_0), \\ k_2 &= h \cdot f\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1\right), \\ k_3 &= h \cdot f\left(x_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_2\right), \\ k_4 &= h \cdot f(x_0 + h, y_0 + k_3), \end{aligned}$$

где h - шаг интегрирования.

2. Вычисляется значение y по формуле

$$y = y_0 + (k_1 + 2k_2 + 2k_3 + k_4) / 6.$$

3. Вновь делается шаг по x : $x=x+h$.

4. Вместо x_0 , y_0 берутся x и найденное на предыдущем этапе y и производится переход к п.1. Расчет производится до тех пор пока x не достигнет заданного значения.

Демонстрация точности методов производится в примере 3.1.

Пример 3.1. Произвести расчет ОДУ $\frac{dy}{dx} = \frac{y^2 - y}{x}$ при заданных начальных условиях: $x_0 = 1 \Rightarrow y_0 = 2$. Использовать методы Рунге-Кутта, Эйлера и аналитическое решение.

```

program rechenie_ODU;
uses crt,printer; var xk,x,y,h,y1:real;j,n:integer;
{Функция аналитического решения дифференциального уравнения}
function fa(x:real):real; begin fa:=1/(1-0.5*x) end;
{Функция для численного решения дифференциального уравнения}
function fun(x:real; y:real):real; begin fun:=(y*y-y)/x; end;
{Процедура численного решения обыкновенного
дифференциального уравнения методом Рунге-Кутта}
procedure odurunge(h,x:real; var y:real);
{h-шаг интегрирования, x- переменная интегрирования, y-искомое
значение функции в точке x}
var w,k1,k2,k3,k4:real;
begin
w:=y; k1:=h*fun(x,y); y:=w+1/2*k1; k2:=h*fun(x+1/2*h,y);
y:=w+1/2*k2; k3:=h*fun(x+1/2*h,y);
y:=w+k3; k4:=h*fun(x+h,y); y:=w+(k1+2*k2+2*k3+k4)/6;
end;

{Процедура численного решения обыкновенного
дифференциального уравнения методом Эйлера}
procedure odujler(h,x:real; var y:real);
{h-шаг интегрирования, x- переменная интегрирования, y-искомое
значение функции в точке x}
begin y:=y+fun(x,y)*h; end;
{Основной блок программы}
begin clrscr;
x:=1; xk:=1.5; y:=2; y1:=2; n:=20; h:=(xk-x)/n;

```


где $i = 1, 2, \dots, N$ – номер каждой зависимой переменной y_i ; x – независимая переменная. Решение системы при заданных начальных условиях $x = x_0, y_1(x_0) = y_{01}, y_2(x_0) = y_{02}, \dots, y_N(x_0) = y_{0N}$ сводится к нахождению зависимостей (интегральных кривых) $y_1(x), y_2(x), \dots, y_i(x), \dots, y_N(x)$, проходящих через точки, заданные начальными условиями $(x_0, y_{01}), (x_0, y_{02}), \dots, (x_0, y_{0i}), \dots, (x_0, y_{0N})$.

Алгоритм решения системы обыкновенных дифференциальных уравнений (СОДУ) состоит из следующих этапов:

1. Вычисление коэффициентов

$$k_{1j} = h \cdot F_j(x_0, y_{01}, y_{02}, \dots, y_{0N}),$$

$$k_{2j} = h \cdot F_j(x_0 + \frac{1}{2}h, y_1, y_2, \dots, y_N), \text{ где } y_j = y_{0j} + \frac{1}{2}k_{1j} \text{ при } j=1, 2, \dots, N,$$

$$k_{3j} = h \cdot F_j(x_0 + \frac{1}{2}h, y_1, y_2, \dots, y_N), \text{ где } y_j = y_{0j} + \frac{1}{2}k_{2j} \text{ при } j=1, 2, \dots, N,$$

$$k_{4j} = h \cdot F_j(x_0 + h, y_1, y_2, \dots, y_N), \text{ где } y_j = y_{0j} + k_{3j} \text{ при } j=1, 2, \dots, N,$$

где h - шаг интегрирования.

2. Вычисляется значение y_j по формуле

$$y_j = y_{0j} + (k_{1j} + 2k_{2j} + 2k_{3j} + k_{4j}) / 6 \text{ при } j=1, 2, \dots, N.$$

3. Вновь делается шаг по x : $x = x + h$.

4. Вместо $x_0, y_{01}, y_{02}, \dots, y_{0N}$ берутся x и найденные на предыдущем этапе y_1, y_2, \dots, y_N и производится переход к п.1. Расчет производится до тех пор пока x не достигнет заданного значения.

Пример. Решить систему обыкновенных дифференциальных уравнений

$$\begin{cases} \frac{dy_1}{dx} = y_2, \\ \frac{dy_2}{dx} = \left(\frac{y_1}{x} - y_2\right) \frac{1}{x} - y_1. \end{cases}$$

с начальными условиями: $x_0 = 0,2$; $y_{01} = 0,09950083$; $y_{02} = 0,49235$.

```

program rechenie_SODU_metodom_Runge_Kutta;
uses crt,printer; const n=2;{Количество дифференциальных
уравнений}
type dim=array[1..n]of real;    var y:dim; x,za,ya:real;

procedure sodu(var x:real; var y:dim);
var w,k1,k2,k3,k4:dim; h:real; j:integer;

function fun(j:integer;x:real;y:dim):real;
begin if j=1 then fun:=y[2];    if j=2 then fun:=(y[1]/x-y[2])/x-y[1]
end;

begin h:=1/10;
for j:=1 to n do w[j]:=y[j]; for j:=1 to n do k1[j]:=h*fun(j,x,y);
for j:=1 to n do y[j]:=w[j]+1/2*k1[j]; for j:=1 to n do
k2[j]:=h*fun(j,x+1/2*h,y);
for j:=1 to n do y[j]:=w[j]+1/2*k2[j]; for j:=1 to n do
k3[j]:=h*fun(j,x+1/2*h,y);
for j:=1 to n do y[j]:=w[j]+k3[j]; for j:=1 to n do k4[j]:=h*fun(j,x+h,y);
for j:=1 to n do y[j]:=w[j]+(k1[j]+2*k2[j]+2*k3[j]+k4[j])/6;
x:=x+h;
end;

begin {Основной блок программы}
clrscr; x:=0.2; y[1]:=0.09950083; y[2]:=0.49235;
writeln('    x        y1        y2');
while x<=2.5 do begin writeln(x:15:8,y[1]:15:8,y[2]:15:8); sodu(x,y);end;
end.

```

С шагом $h = 0,1$ получены следующие результаты:

x	y_1	y_2
0,3	0,14830512	0,48310851
0,4	0,19600198	0,47022890
0,5	0,24223423	0,45384038
-	-	-
1,0	0,43997848	0,32508611
-	-	-
1,5	0,55784207	0,13984235
-	-	-
2,5	0,49700746	-0,24718087

4. Метод наложения потоков

4.1. Течения воздуха вблизи всасывающих отверстий в плоской стенке

Метод наложения потоков основывается на том, что вектор скорости сложного потенциального потока несжимаемой жидкости может быть представлен в виде суммы векторов скорости составляющих его потоков

$$\vec{v} = \vec{v}_1 + \vec{v}_2 + \vec{v}_3 + \dots + \vec{v}_n, \quad (3.1)$$

где n - количество складываемых потоков.

Это же утверждение справедливо для функций тока и потенциала сложного потока.

В инженерной практике встречается также иная модификация метода наложения потоков, не имеющего строгого математического и физического обоснования, однако имеющая неплохое совпадение с экспериментальными данными для задач о воздушно-струйных течениях. Квадрат (куб) координат скорости равен сумме квадратов (кубов) соответствующих координат скоростей складываемых потоков

$$v_i^2 = v_{i_1}^2 + v_{i_2}^2 + \dots + v_{i_n}^2, \quad (3.2)$$

$$v_i^3 = v_{i_1}^3 + v_{i_2}^3 + \dots + v_{i_n}^3, \quad (3.3)$$

где v_i - i -координата вектора скорости.

Изменение геометрической формы вытяжного отверстия приводит к изменению скорости во всасывающем факеле, увеличение которой приводит к уменьшению производительности системы аспирации, а соответственно к снижению энергозатрат. Используя метод наложения потоков определим форму отверстия имеющего наибольшую дальность.

Предполагаем, что скорость воздуха во всех точках равновеликих по площади вытяжных отверстия постоянна и равна V_0 .

Определим осевую скорость V_z у всасывающего отверстия в виде правильного n -угольника площадью S (рис.3.1).

Вычислим скорость V_{zOAB} в некоторой точке M , лежащей на оси OZ , вызываемой действием треугольного всасывающего отверстия OAB .

Так как угол $\angle BOA = \frac{\pi}{n}$, то площадь многоугольника

$$S = \frac{1}{2} |OB|^2 n \sin \frac{2\pi}{n} \text{ и длина}$$

$$|OA| = |OB| \cos \frac{\pi}{n} = \sqrt{\frac{S}{n} \operatorname{ctg} \frac{\pi}{n}}.$$

Вспользуемся полярной системой координат (полярная ось совпадает с OX) и выделим элементарную площадку $\rho d\rho d\varphi$ в плоскости ΔOAB , считая что на ней действует точечный сток. Тогда элементарный расход $dL = V_0 \rho d\rho d\varphi$, скорость $dV_z = dV \cos \alpha$

$$= \frac{dL}{2 \cdot \pi \cdot (\rho^2 + Z^2)} \cdot \frac{z}{\sqrt{\rho^2 + z^2}} \text{ и}$$

$$V_{zOAB} = \frac{V_0 \cdot Z}{2 \cdot \pi} \int_0^{\frac{\pi}{n}} d\varphi \int_0^{\frac{|OA|}{\cos \varphi}} \frac{\rho \cdot d\rho}{(\rho^2 + Z^2)^{\frac{3}{2}}}. \quad (3.4)$$

Интегрируя (3.4) и умножая, полученный результат на количество таких треугольников $2n$, получим зависимость для осевой скорости у правильного многоугольника

$$V_z = \frac{n \cdot V_0}{\pi} \left[\frac{\pi}{n} - \arcsin \frac{Z \cdot \sin \frac{\pi}{n}}{\sqrt{\frac{S}{n} \cdot \operatorname{ctg} \frac{\pi}{n} + Z^2}} \right]. \quad (3.5)$$

Устремляя $n \rightarrow \infty$ получим известную формулу для расчета осевой скорости у круглого всасывающего отверстия [29]

$$V_z = V_0 \cdot \left[1 - \frac{Z}{\sqrt{R^2 + Z^2}} \right], \quad (3.6)$$

где R - радиус круга.

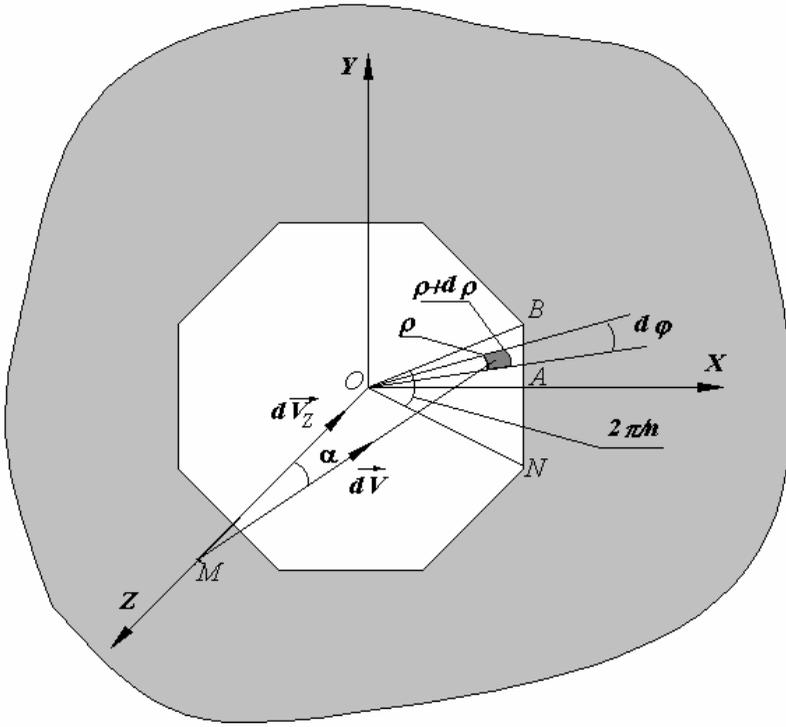


Рис. 3.1. Правильный многоугольник встроенный в плоскую безграничную стенку

При $n = 4$ имеем формулу для расчета скорости у квадратного отверстия, полученную И. А. Шепелевым,

$$V_z = \frac{2 \cdot V_0}{\pi} \cdot \operatorname{arctg} \frac{A^2 / 4}{\sqrt{A^2 / 2 + Z^2}}, \quad (3.7)$$

где A - длина стороны квадрата.

Приведем также формулы [12-29] для расчета осевых скоростей воздуха: у прямоугольного отверстия размером $2A \times 2B$

$$V_z = \frac{2 \cdot V_0}{\pi} \cdot \operatorname{arctg} \frac{B \cdot A}{Z \cdot \sqrt{B^2 + A^2 + Z^2}}, \quad (3.8)$$

кольцевого отверстия с внутренним радиусом R_1 и внешним R_2

$$V_z = Z \cdot V_0 \cdot \left(\frac{1}{\sqrt{R_1^2 + Z^2}} - \frac{1}{\sqrt{R_2^2 + Z^2}} \right) \quad (3.9)$$

и эллиптического

$$V_z = \frac{2 \cdot V_0}{\pi} \cdot \int_0^{\pi/2} \left[1 - \frac{Z}{\sqrt{\frac{a^2 \cdot e^2}{e^2 \cdot \cos^2 \varphi + a^2 \cdot \sin^2 \varphi} + Z^2}} \right] d\varphi, \quad (3.10)$$

где a - большая, e - малая полуоси эллипса.

Пример 4.1. Произвести расчет изменения осевой скорости при удалении от всасывающих отверстий различной геометрической формы. Найти скорость в точке z (на оси отсоса): z изменяется от 0 до 10 с шагом $\Delta z = 0,25$ для эллиптического отверстия с фокусами a, e при $e = 1$ и $a/e = 1+0,5i$, где i – номер варианта; прямоугольного отверстия, той же площади S , что и для эллипса (одна из сторон имеет длину 1); кольцевого отверстия, той же площади S , при внутреннем радиусе $R_{внут} = 1$; треугольного отверстия площадью S , при $n=3$; квадратного отверстия площадью S ; круглого отверстия площадью S .

Program Otverstie;

```
Uses printer; var variant:integer; i:integer; z26:real; vo,z,
a,b:real;{ Стороны прямоугольника } Sel,Rk,Rkr,rkp,Apr,int:real;
Vkol,Vpr,Vkv,Vel,Vkrug,Vtreug:real;
function dvel(z,phi:real):real;
Begin
dvel:=1-z/sqrt((sqrt(a*b))/(sqrt(b*cos(phi))+sqrt(a*sin(phi)))+z*z);
end;
```

```

procedure gaussint(z,niz,ver:real;var int:real);
const n=6; var a,x:array[1..20]of real; j:byte;
begin
  x[1]:=0.93246951420315; x[2]:=0.66120938646626;
x[3]:=0.23861918608319;
  a[1]:=0.17132449237917; a[2]:=0.36076157304813;
a[3]:=0.46791393457269;
  for j:=1 to n do begin x[n-(j-1)]:=-x[j]; a[n-(j-1)]:=a[j] end;
    for j:=1 to n do x[j]:=(ver+niz)/2+(ver-niz)/2*x[j];      int:=0;
    for j:=1 to n do int:=int+a[j]*dvel(z,x[j]);  int:=(ver-niz)/2*int;
end;
function arksin(x:real):real;
begin
if abs(x)=1 then arksin:=pi/2 else arksin:=arctan(x/sqrt(1-x*x));
end;
begin
write('Введите номер варианта ');
Read(variant);
vo:=1; b:=1; a:=b*(1+0.5*variant); sel:=pi*a*b; {Площадь эллипса}
apr:=sel/1; rk:=sqrt(sel/pi+1); rkp:=sqrt(sel/pi);
writeln('z  Прямоугольник Квадрат Эллипс Кольцо Круг
Треугольник ');
for i:=1 to 20 do begin
z:=i*0.25;
vpr:=2*vo/pi*arctan(apr/4/z/sqrt(1/4+apr*apr/4+z*z));
{Прямоугольник}
vkv:=2*vo/pi*arctan(sel/4/z/sqrt(sel/2+z*z));{Квадрат}
gaussint(z,0,pi/2,int);
Vel:=2*vo/pi*int;{Скорость у эллиптического отверстия }
Vkol:=z*vo*(1/sqrt(1+z*z)-1/sqrt(rk*rk+z*z));{Скорость у кольца}
Vkrug:=z*vo*(1/z-1/sqrt(rk*rk+z*z));{Круг}
Vtreug:=3*vo/pi*(pi/3-arksin(z*sin(pi/3)/ sqrt(sel/3*sqrt(1/3)+z*z)));
{Треуг.}
writeln(z:0:2, ' ,vpr:0:4,' ,vkv:0:4,' ,Vel:0:4,' ,Vkol:0:4,' ,Vkrug:0:4,'
',Vtreug:0:4)
end;
end.
Результаты расчета при номере варианта 40.

```

Z	Прямоугольник	Квадрат	Эллипс	Кольцо	Круг	Треугольник
0,25	0,7048	0,9447	0,8434	0,1893	0,9468	0,9421
.....
1,0	0,2950	0,7837	0,4975	0,4986	0,7915	0,7745
.....
3,0	0,1047	0,4479	0,1992	0,4099	0,4612	0,4350
.....
5,00	0,0627	0,2602	0,1184	0,2513	0,2707	0,2525

5. Построение траекторий движения пылевой частицы в воздушном потоке

На пылинку в воздушном потоке действуют силы тяжести и сопротивления воздуха (рис.5.1).

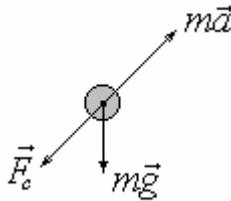


Рис. 5.1

Согласно второму закону Ньютона

$$m\vec{a} = \vec{F}_c + m\vec{g}, \quad (5.1)$$

где \vec{F}_c - сила сопротивления воздуха, $m\vec{g}$ - сила тяжести. Ускорение пылевой частицы \vec{a} является производной скорости \vec{v}_1 по времени

$$\vec{a} = \frac{d\vec{v}_1}{dt}.$$

Сила сопротивления воздуха представима в виде

$$\vec{F}_c = -\psi \frac{|\vec{v}_1 - \vec{v}| \cdot (\vec{v}_1 - \vec{v})}{2} \rho \chi S_M,$$

где ψ - коэффициент сопротивления среды, ρ - плотность воздуха, χ - коэффициент формы частицы ($\chi = 1$, если частица сферической формы; $\chi = 1,8$ для острозернистой частицы), \vec{v} - скорость воздуха, S_M - площадь миделевого сечения частицы, вычисляемой по формуле

$$S_M = \frac{\pi d_3^2}{4},$$

где d_3 - эквивалентный диаметр, т.е. диаметр равновеликого по объему шара.

Коэффициент сопротивления частицы размером от 10 мкм до 100 мкм вычисляется по формуле Стокса

$$\psi = \frac{24}{\text{Re}},$$

где $\text{Re} = \frac{|\vec{v}_1 - \vec{v}| \cdot d_3}{\nu}$ - число Рейнольдса, ν - коэффициент кинематической вязкости.

Подставляя все вышеуказанные формулы в выражение для силы сопротивления будем иметь

$$\vec{F}_c = - \frac{24}{|\vec{v}_1 - \vec{v}| \cdot d_g} \cdot \frac{|\vec{v}_1 - \vec{v}|(\vec{v}_1 - \vec{v})}{2} \rho \chi \frac{\pi d_g^2}{4} = -3\pi\nu\rho\chi d_g (\vec{v}_1 - \vec{v}).$$

Подставляя последнее выражение в (5.1) и учитывая, что $m = \rho_1 \frac{\pi d_g^3}{6}$ окончательно получим обыкновенное дифференциальное уравнение второго порядка, описывающее траекторию частицы пыли при стоксовском режиме обтекания ее воздушным потоком

$$\frac{\pi d_g^3}{6} \rho_1 \frac{d\vec{v}_l}{dt} = \frac{\pi d_g^3}{6} \rho_1 \vec{g} - 3\pi\nu\rho\chi d_g (\vec{v}_l - \vec{v}), \quad (5.2)$$

где ρ_1 - плотность частицы.

Уравнение (5.2) при $A = \rho \frac{\pi d_g^3}{6}$ и $B = 3\pi\nu\rho\chi d_g$ для плоского случая сводится к системе обыкновенных дифференциальных уравнений первого порядка

$$\left\{ \begin{array}{l} A \frac{dv_{1x}}{dt} = -B(v_{1x} - v_x), \\ \frac{dx}{dt} = v_{1x}, \\ A \frac{dv_{1y}}{dt} = -Ag - B(v_{1y} - v_y), \\ \frac{dy}{dt} = v_{1y}. \end{array} \right. \quad (5.3)$$

В пространстве имеем систему 6-ти дифференциальных уравнений

$$\left\{ \begin{array}{l} A \frac{dv_{1x}}{dt} = -B(v_{1x} - v_x), \\ \frac{dx}{dt} = v_{1x}, \\ A \frac{dv_{1y}}{dt} = -B(v_{1y} - v_y), \\ \frac{dy}{dt} = v_{1y}, \\ A \frac{dv_{1z}}{dt} = -Ag - B(v_{1z} - v_z), \\ \frac{dz}{dt} = v_{1z}. \end{array} \right. \quad (5.4)$$

Уравнения (5.3) и (5.4) решаются численно при заданных начальных условиях: координат точки вылета и величин составляющих скорости.

Пример. 5.1. Пусть в некоторой точке $M(a,b)$ находится сток воздуха единичной интенсивности (рис.5.2). Найти траекторию частицы, которая вылетает из точки $A(0,h)$. Входные данные для расчета: $(=1$ (частицы пыли сферической формы), $(=1,5(10^{-5}$ кг/(с(м), $(_1=2500$ кг/м³, $(=1,2$ кг/м³, $v_{1x}=1$ м/с, $v_{1y}=0$ м/с (начальная скорость частицы).

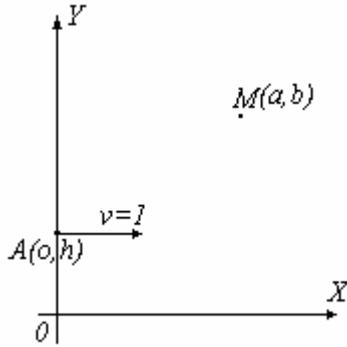


Рис.5.2

Величина скорости воздуха, вызываемой действием точечного стока в плоскости

$$v = \frac{Q}{2\pi r}.$$

Вычислим составляющие скорости воздуха

$$v_x = \frac{Q}{2\pi r} \cos \alpha = \frac{Q}{2\pi r} \cdot \frac{a-x}{r},$$

$$v_y = \frac{Q}{2\pi r} \sin \alpha = \frac{Q}{2\pi r} \cdot \frac{b-y}{r},$$

Q - расход воздуха удаляемый стоком, $r = \sqrt{(a-x)^2 + (b-y)^2}$ - расстояние от произвольной точки (x, y) до т. $M(a, b)$.

Ниже приведена программа расчета траектории частицы пыли при $a=1$, $b=1$, $h=0,1$.

```

program traektorii_metod_Runge_Kutta;
{Расчет траектории пылевой частицы в плоскости}
uses crt;
const n=4{Количество диф. уравнений}; d=40/1000000;{диаметр
частицы}
type vek=array[1..2]of real; dim=array[1..n]of real;
var a,b,l,Vpot,Vo,t,rel:real; y:dim; v:vek; i:longint; f:text;
procedure sodu(v:vek; var x:real; var y:dim);
const n=4; type dim1=array[1..4]of real;
var w,k1,k2,k3,k4:dim1; con,h:real; j:integer;
function fun(v:vek;j:integer;x:real;y:dim):real;
const g=9.81; ro=2500;{Плотность частицы} kf=1;{Коэффициент
формы} vjaz=1.5/100000;{Кинематическая вязкость} gov=1.2;{плотность
воздуха} var A,B:real;
begin
A:=pi*d*d*d/6*ro; B:=3*pi*kf*vjaz*d*rov;
if j=1 then fun:=y[3]; if j=3 then fun:=-B/A*(y[3]-v[1]);
if j=2 then fun:=y[4]; if j=4 then fun:=-g-B/A*(y[4]-v[2])
end;
begin h:=0.001; {Шаг интегрирования}
for j:=1 to n do w[j]:=y[j]; for j:=1 to n do k1[j]:=h*fun(v,j,x,y);
for j:=1 to n do y[j]:=w[j]+1/2*k1[j]; for j:=1 to n do
k2[j]:=h*fun(v,j,x+1/2*h,y);
for j:=1 to n do y[j]:=w[j]+1/2*k2[j]; for j:=1 to n do
k3[j]:=h*fun(v,j,x+1/2*h,y);
for j:=1 to n do y[j]:=w[j]+k3[j]; for j:=1 to n do k4[j]:=h*fun(v,j,x+h,y);
for j:=1 to n do y[j]:=w[j]+(k1[j]+2*k2[j]+2*k3[j]+k4[j])/6; x:=x+h;
end;
begin clrscr; t:=0;
{y[1]- абсцисса точки x, y[2]- ордината точки y, y[3]-горизонтальная
составляющая скорости, y[4]-вертикальная составляющая скорости}
l:=1; Vo:=1; y[1]:=0; y[2]:=0.1; y[3]:=Vo ; y[4]:=0;
writeln('          Траектория пылевой частицы ');
writeln('          Диаметр частицы  ',d*1000000:0:1,'мкм');
writeln('          '); writeln('          x          y          Vx          Vy');
a:=1; b:=1; i:=0;

```

```

while (y[2]>0)and(y[2]<=b) do begin if (i mod 3000=0)or(i=1) then
writeLn(y[1]:15:8,' ',y[2]:15:8,' ',y[3]:15:8,' ',y[4]:15:8);
v[1]:=1/2/pi/sqrt(sqr(y[2]-b)+sqr(y[1]-a))/(sqr(y[2]-b)+sqr(y[1]-a))*(a-
y[1]);
v[2]:=1/2/pi*sqr(sqr(y[2]-b)+sqr(y[1]-a))/(sqr(y[2]-b)+sqr(y[1]-a))*(b-
y[2]);
sodu(v,t,y); i:=i+1;
end; end.

```

Результаты расчета метода Рунге-Кутты при диаметре частицы 40 мкм и шаге по времени 0,001

x	y	Vx	Vy
0.00000000	0.10000000	1.00000000	0.00000000
0.00096315	0.09999942	0.92727902	-0.00113924
0.21591206	0.07928847	0.07053560	0.00000003
0.43455976	0.09977045	0.07489289	0.01360598
0.65920741	0.16013569	0.07288271	0.02631626
0.85350393	0.25367864	0.05312247	0.03503993
0.96516877	0.36413031	0.02158575	0.03780213
0.99722000	0.47807601	0.00314874	0.03804151
0.99998118	0.59220651	0.00004542	0.03804383

Видно, что частицы пыли улавливается стоком воздуха, так как ординат увеличивается и приближается к единице, т.е. к месту где находится сток (модель вытяжного отверстия).

Пример 5.2. Пусть в некоторой точке $M(a,b,c)$ находится сток воздуха единичной интенсивности (рис.5.2). Найти траекторию частицы, которая вылетает из точки $A(0,0,h)$. Входные данные для расчета: $(=1$ (частицы пыли сферической формы), $(=1,5(10^{-5}$ кг/(с·м), $(_1=2500$ кг/м³, $(=1,2$ кг/м³, $v_{1x}=1$ м/с, $v_{1y}=1$ м/с, $v_{1z}=0$ м/с (начальная скорость частицы).

Величина скорости воздуха, вызываемой действием точечного стока в плоскости

$$v = \frac{Q}{4\pi r^2}.$$

Составляющие скорости воздуха вычисляются по следующим выражениям

$$v_x = \frac{Q(a-x)}{4\pi r^3}, \quad v_y = \frac{Q(b-y)}{4\pi r^3}, \quad v_z = \frac{Q(c-z)}{4\pi r^3}.$$

Q - расход воздуха удаляемый стоком,

$r = \sqrt{(a-x)^2 + (b-y)^2 + (c-z)^2}$ - расстояние от произвольной точки (x,y,z) до т. $M(a,b,c)$.

```

program traektorii_metod_Runge_Kutta;
{Траектория пылевой частицы в пространстве}
uses crt;
const n=6{Количество диф. уравнений}; d=15/1000000;{Диаметр
частицы}
type vek=array[1..3]of real; dim=array[1..n]of real;
var r,c,a,b,l,Vpot,Vo,t,rel:real; y:dim; v:vek; i:longint;

procedure sodu(v:vek; var x:real; var y:dim);
type dim1=array[1..n]of real; var w,k1,k2,k3,k4:dim1; con,h:real;
j:integer;

function fun(v:vek;j:integer;x:real;y:dim):real;
const g=9.81; ro=2500;{Плотность частицы} kf=1;{Коэффициент
формы}
vjaz=1.5/100000;{Кинематическая вязкость} roV=1.2;{Плотность
воздуха}
var A,B:real;
begin A:=-18*vjaz*kf*roV/(d*d*ro);
if j=1 then fun:=y[4]; if j=2 then fun:=y[5]; if j=3 then fun:=y[6];
if j=4 then fun:=A*(y[4]-v[1]); if j=5 then fun:=A*(y[5]-v[2]);
if j=6 then fun:=-g+A*(y[6]-v[3]);
end;

begin
h:=0.0005; {Шаг интегрирования}
for j:=1 to n do w[j]:=y[j]; for j:=1 to n do k1[j]:=h*fun(v,j,x,y);
for j:=1 to n do y[j]:=w[j]+1/2*k1[j];

```

```

for j:=1 to n do k2[j]:=h*fun(v,j,x+1/2*h,y); for j:=1 to n do
y[j]:=w[j]+1/2*k2[j];
for j:=1 to n do k3[j]:=h*fun(v,j,x+1/2*h,y); for j:=1 to n do
y[j]:=w[j]+k3[j];
for j:=1 to n do k4[j]:=h*fun(v,j,x+h,y);
for j:=1 to n do y[j]:=w[j]+(k1[j]+2*k2[j]+2*k3[j]+k4[j])/6; x:=x+h;
end;
begin
clrscr; t:=0; Vo:=1; y[1]:=0; y[2]:=0; y[3]:=0.1; y[4]:=Vo; y[5]:=Vo;
y[6]:=0;
writeln('          Траектория пылевой частицы '); writeln(' ');
writeln('          Диаметр частицы ',d*1000000:0:1,'мкм');
writeln('    x      y      z      Vx      Vy      Vz');
a:=1; b:=1; c:=1; i:=0;
while (y[3]>0)and(y[3]<=c) do
begin if (i mod 1000=0)or(i=1) then
writeln(y[1]:12:6,y[2]:12:6,y[3]:12:6,y[4]:12:8,y[5]:12:6,y[6]:12:6);
r:=sqrt(sqrt(y[2]-b)+sqrt(y[1]-a)+sqrt(y[3]-c));
v[1]:=1/4/pi*(a-y[1])/t/r; v[2]:=1/4/pi*(b-y[2])/t/r;
v[3]:=1/4/pi*(c-y[3])/t/r; sodu(v,t,y); i:=i+1;
end;
end.

```

Результаты расчета траектории пылевой частицы диаметром 15мкм

x	y	z	Vx	Vy	Vz
0.000000	0.000000	0.100000	1.00000000	1.000000	0.000000
0.000436	0.000436	0.100000	0.75400459	0.754005	-0.000457
0.027440	0.027440	0.097988	0.01739201	0.017392	-0.000901
0.053907	0.053907	0.097340	0.01790606	0.017906	0.000052
0.081187	0.081187	0.098194	0.01847771	0.018478	0.001104
0.109375	0.109375	0.100710	0.01911902	0.019119	0.002273
0.138586	0.138586	0.105084	0.01984571	0.019846	0.003586
0.168965	0.168965	0.111555	0.02067868	0.020679	0.005075
0.200690	0.200690	0.120421	0.02164637	0.021646	0.006788
0.233991	0.233991	0.132062	0.02278866	0.022789	0.008789
0.269171	0.269171	0.146977	0.02416337	0.024163	0.011171
0.306640	0.306640	0.165836	0.02585799	0.025858	0.014077
0.346974	0.346974	0.189579	0.02801223	0.028012	0.017731

0.391023	0.391023	0.219592	0.03086526	0.030865	0.022521
0.440137	0.440137	0.258062	0.03486779	0.034868	0.029174
0.496667	0.496667	0.308797	0.04099634	0.040996	0.039264
0.565424	0.565424	0.379564	0.05190323	0.051903	0.057066
0.659692	0.659692	0.491781	0.07891092	0.078911	0.100807
0.932454	0.932454	0.891638	1.63821862	1.638219	2.610321

В данном случае пылевая частица улавливается точечным стоком.

6. Метод конформных отображений

6.1. Некоторые сведения о функциях комплексного переменного

Комплексные числа - это числа вида $z = x + iy$ (алгебраическая форма записи комплексного числа), где $i = \sqrt{-1}$ - мнимая единица, x и y действительные числа. Действительная часть комплексного числа $\operatorname{Re} z = x$, мнимая часть $\operatorname{Im} z = y$.

Комплексные числа могут изображаться точками на плоскости. Если по оси абсцисс откладывать действительную часть числа, а по оси ординат - мнимую часть, то комплексное число $z = x + iy$ можно изобразить точкой (x, y) в прямоугольной декартовой системе координат. Радиус-вектор \vec{r} комплексного числа $z = x + iy$ - это вектор с началом в центре координат и концом в точке (x, y) (рис.1). Длина радиус-вектора \vec{r} называется модулем комплексного числа. Обозначается $|\vec{r}|$, r или $\operatorname{mod}(z)$.

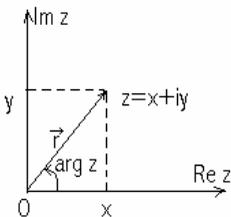


Рис.6.1. Изображение комплексных чисел.

Аргументом комплексного числа ($\operatorname{Arg} z$) называется угол, отсчитываемый от положительного направления оси абсцисс до радиус-вектора \vec{r} . Положительное направление отсчета - против часовой стрелке, а отрицательное - по часовой стрелке.

Аргументом комплексного числа в смысле главного значения ($\operatorname{arg} z$) называется угол, лежащий в промежутке от $-\pi$ до π ($-\pi < \operatorname{arg} z \leq \pi$) и отсчитываемый от

положительного направления оси абсцисс до радиус-вектора \vec{r} . Аргумент комплексного числа $z = x + iy$ может быть вычислен по формулам, указанным, в таблице.

Сумма комплексных чисел $z_1 = x_1 + iy_1$ и $z_2 = x_2 + iy_2$ называется число $z = z_1 + z_2 = (x_1 + x_2) + i(y_1 + y_2)$.

Произведение

$$z_1 \cdot z_2 = x_1 \cdot x_2 - y_1 \cdot y_2 + i(y_1 x_2 + y_2 x_1),$$

частное

$$\frac{z_1}{z_2} = \frac{x_1 x_2 + y_1 y_2}{x_2^2 + y_2^2} + i \frac{y_1 x_2 - x_1 y_2}{x_2^2 + y_2^2}.$$

Формулы вычисления аргумента комплексного числа

x, y	$\arg z$
$x > 0$	$\arctg y/x$
$x < 0, y > 0$	$\pi + \arctg y/x$
$x < 0, y < 0$	$-\pi + \arctg y/x$
$x = 0, y < 0$	$-\pi/2$
$x = 0, y > 0$	$\pi/2$
$x = 0, y = 0$	не определен

В тригонометрической форме записи комплексное число имеет вид $z = r(\cos \varphi + i \sin \varphi)$, где r - модуль, а φ - аргумент комплексного числа в смысле главного значения.

Если $z_1 = r_1(\cos \varphi_1 + i \sin \varphi_1)$ и $z_2 = r_2(\cos \varphi_2 + i \sin \varphi_2)$, то

$$z_1 \cdot z_2 = r_1 \cdot r_2 (\cos(\varphi_1 + \varphi_2) + i \sin(\varphi_1 + \varphi_2)),$$

$$z_1 / z_2 = r_1 / r_2 (\cos(\varphi_1 - \varphi_2) + i \sin(\varphi_1 - \varphi_2)).$$

Для возведения комплексного числа в n -ю степень необходимо воспользоваться формулой Муавра $z^n = r^n (\cos n\varphi + i \sin n\varphi)$, а для извлечения корня n -й степени выражением

$$\sqrt[n]{z} = \sqrt[n]{r} \left(\cos \frac{\varphi + 2k\pi}{n} + i \sin \frac{\varphi + 2k\pi}{n} \right), \text{ где } k=0,1,2,\dots,n-1.$$

Комплексное число может быть представлено в показательной форме $z = re^{i\varphi}$. Согласно формулам Эйлера $e^{i\varphi} = \cos \varphi + i \sin \varphi$ и $e^{-i\varphi} = \cos \varphi - i \sin \varphi$, откуда следует, что $e^z = e^{x+iy} = e^x \cdot e^{iy} = e^x (\cos y + i \sin y)$. Логарифм в смысле главного значения от комплексного числа может быть вычислен следующим образом $\ln z = \ln re^{i\varphi} = \ln r + i\varphi$, где $r = |z|$ и $\varphi = \arg z$. Основные свойства логарифмов от действительных чисел сохраняются и для логарифмов от комплексных чисел $\ln(z_1 \cdot z_2) = \ln z_1 + \ln z_2$, $\ln(z_1 / z_2) = \ln z_1 - \ln z_2$.

Рассмотрим простейшие функции от комплексных переменных и их отображения.

1. Функция $\omega = az$, где $a > 0$, $a \in R$. Если представить число z в показательной форме, то $\omega = are^{i\varphi}$, т. е. аргумент свое значение не изменяет, а модуль $|\omega| = ar$. Таким образом, данная функция осуществляет либо растяжение при $a > 1$, либо сжатие при $0 < a < 1$.

2. Функция $\omega = e^{i\alpha} \cdot z = e^{i\alpha} \cdot r \cdot e^{i\varphi} = r \cdot e^{i(\varphi+\alpha)}$ осуществляет поворот против часовой стрелки на угол α .

3. Отображение $\omega = z + b$, где $b = b_1 + b_2 \cdot i$ осуществляет сдвиг на величину b_1 по оси абсцисс и на b_2 по оси ординат.

4. Функция $\omega = az + b$, где $a = a_1 + a_2 \cdot i = |a|e^{i\theta}$ и $b = b_1 + b_2 \cdot i$ осуществляет линейное отображение (сжатие либо растяжение, поворот и сдвиг).

5. Функция $\omega = z^2 = r^2 e^{i2\varphi}$ возводит модуль в квадрат и увеличивает модуль в 2 раза. Например, внутренность первого квадранта отображает на верхнюю полуплоскость. Функция осуществляет взаимно-однозначное отображение при $0 < \varphi < \pi$, при изменении аргумента в больших пределах отображение неоднозначно.

6. Отображение $\omega = |z|z = r^2 e^{i\varphi}$ в отличие от предыдущей функции взаимно-однозначное.

7. Функция $\omega = \begin{cases} \frac{1}{1-|z|}, & |z| < 1 \\ \infty, & |z| = 1 \end{cases}$ осуществляет отображение

внутренности единичного круга на плоскость с разрезом вдоль оси абсцисс, начиная от 1 до ∞ .

8. Функция $\omega = \sqrt[3]{z}$ при $0 < \arg z < \pi$ многозначна. Она ставит в соответствие каждой точке z верхней полуплоскости три точки плоскости ω : $\omega_0 = \sqrt[3]{z}_0$, которая лежит в угле $0 < \arg \omega < \pi/3$; $\omega_1 = \sqrt[3]{z}_1$, которая лежит в угле $2\pi/3 < \arg \omega < \pi$; $\omega_2 = \sqrt[3]{z}_2$, которая лежит в угле $-\pi/3 < \arg \omega < -\pi/3$. Эти значения соответственно определяют в верхней полуплоскости z три однозначные функции $\omega_0(z), \omega_1(z), \omega_2(z)$, которые являются однозначными ветвями функции $\omega = \sqrt[3]{z}$.

6.2 Общие сведения о конформных отображений

Отображение области z на область ω , задаваемое функцией $\omega = f(z)$, называется конформным, если оно обладает свойствами постоянства растяжений и углов поворота. Это означает, что любой бесконечно малый элемент в области z изменяется в одно и тоже количество раз при отображении $\omega = f(z)$, причем коэффициент линейного растяжения $|f'(z)|$. Кривая в области z поворачивается на один и тот же угол $\alpha = \arg f'(z)$ в одном и том же направлении при отображении $\omega = f(z)$. Метод конформных отображений применяется для расчета потенциальных течений газа. Течение называется потенциальным, если существует такая функция ϕ , называемая потенциалом, для которой справедливо равенство

$$\vec{v} = \text{grad}\phi, \quad (6.1)$$

где \vec{v} - скорость течения. Горизонтальная и вертикальная составляющие скорости определяются соответственно по формулам

$$v_x = \frac{\partial \varphi}{\partial x}, v_y = \frac{\partial \varphi}{\partial y}. \quad (6.2)$$

Уравнение неразрывности для плоского потенциального течения газа сводится к уравнению Лапласа относительно потенциала

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0. \quad (6.3)$$

Линия на которой $\varphi = const$ называется эквипотенциалью. На эквипотенциали величина скорости одинакова во всех точках.

Кроме потенциала вводится также функция тока ψ

$$v_x = \frac{\partial \psi}{\partial y}, v_y = -\frac{\partial \psi}{\partial x}, \quad (6.4)$$

которая также удовлетворяет уравнению неразрывности.

Линия на которой функция ψ постоянна называется линией тока. Вектор скорости направлен по касательной в каждой точке линии тока. Эквипотенциали и линии тока перпендикулярны в точках их пересечения.

Комплексным потенциалом течения называется функция

$$w = \varphi + i\psi. \quad (6.5)$$

Если известна функция w в плоскости течения z , то определив комплексную скорость

$$\frac{dw}{dz} = \frac{\partial w}{\partial x} \cdot \frac{\partial x}{\partial z} = \frac{\partial \varphi}{\partial x} + i \frac{\partial \psi}{\partial x} = v_x - iv_y \quad (6.6)$$

можно найти вертикальную и горизонтальную составляющие скорости в любой точке области течения.

Определим, например, комплексный потенциал течения в верхней полуплоскости с точечным источником интенсивности Q , расположенном в начале координат (рис.6.1).

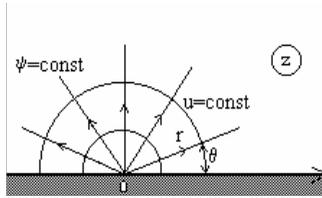


Рис. 6.1. Точечный источник в полуплоскости

В этом случае линии тока - это лучи, исходящие из точки 0, а эквипотенциали - полуокружности. Скорость вдоль линии тока

$$v_r = \frac{\partial \varphi}{\partial r} = \frac{Q}{\pi r}, \quad (6.7)$$

откуда $\varphi = \int \frac{Q}{\pi r} dr = \frac{Q}{\pi} \ln r + C$. Пусть $\varphi = 0$ при $r = 1$, тогда $C = 0$. Для того, чтобы определить функцию тока вычислим ее частную производную

$$\frac{\partial \psi}{\partial y} = v_x = v_r \cos \theta = \frac{Qx}{\pi r^2}. \quad (6.8)$$

Угол θ для первого квадранта $\theta = \arctg \frac{y}{x}$, а второго $\theta = \pi + \arctg \frac{y}{x}$. Поэтому $\frac{\partial \theta}{\partial y} = \frac{x}{r^2}$. Так как $\frac{d\psi}{d\theta} = \frac{\partial \psi}{\partial y} \frac{\partial y}{\partial \theta} = \frac{Q}{\pi}$ то $\psi = \frac{Q}{\pi} \theta$ ($const=0$, так как выбираем $\theta=0$ при $\psi=0$ и комплексный потенциал примет вид $w = \frac{Q}{\pi} \ln r + i \frac{Q}{\pi} \theta = \frac{Q}{\pi} \ln z$. Если точечный источник находится в точке z_0 то

$$w = \frac{Q}{\pi} \ln(z - z_0), \quad (6.9)$$

В случае точечного стока

$$w = -\frac{Q}{\pi} \ln(z - z_0). \quad (6.10)$$

Алгоритм расчета поля скоростей газа в заданной плоскости течения, называемой физической, состоит из следующих этапов:

1. Определение отображения физической плоскости течения z на верхнюю полуплоскость t (геометрическая плоскость течения, в качестве которой можно использовать и любую другую область с известным комплексным потенциалом)

$$z = f(t); \quad (6.11)$$

2. Задание точки z_0 в которой необходимо определить скорость течения и вычисление параметра t_0 из формулы (6.11);

3. Вычисление комплексной скорости

$$\frac{dw}{dz} = \frac{dw}{dt} \cdot \frac{dt}{dz}, \quad (6.12)$$

откуда определяются координаты вектора скорости.

Наиболее трудоемким является первый этап. Кроме того определение отображения (6.11) невозможно для многосвязных областей, несводящихся к односвязным, и поэтому метод конформных отображений для таких задач не применим.

Важным свойством конформных отображений является принцип симметрии. Если некоторая прямая делит физическую область течения на две симметричные подобласти то одну из них можно отбросить и произвести расчет поля скоростей для оставшейся подобласти. Полученные результаты расчета поля скоростей будут справедливы и для симметричных точек отброшенной подобласти.

Рассмотрим часто используемое в инженерной практике [89], отображение верхней полуплоскости на внутренность многоугольника (рис.6.2-6.3).

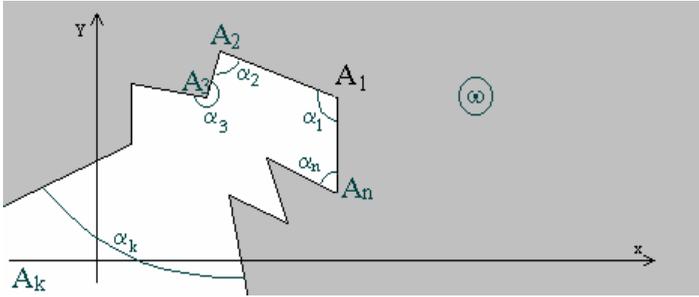


Рис.6.2. Многоугольная область

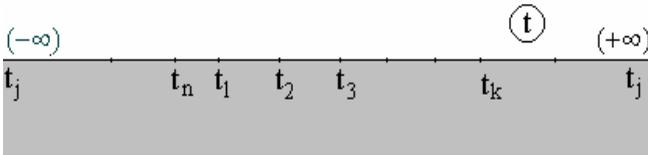


Рис.6.3. Верхняя полуплоскость

Направление обхода области выбрано таким образом, чтобы область все время находилась слева, если двигаться по границе области против часовой стрелки. Отображение верхней полуплоскости на многоугольную область осуществляется интегралом Кристоффеля-Шварца

$$\omega = C_1 \int_{t_k}^t (t-t_1)^{\frac{\alpha_1}{\pi}-1} (t-t_2)^{\frac{\alpha_2}{\pi}-1} \dots (t-t_n)^{\frac{\alpha_n}{\pi}-1} dt + C_2, \quad (6.13)$$

где t_1, t_2, \dots, t_n - n независимых параметров; $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$ - $(n-1)$ независимых параметров, причем $\alpha_1 + \alpha_2 + \dots + \alpha_{n-1} + \alpha_n = (n-2)\pi$; C_1, C_2 - комплексные постоянные, а значит 4 независимых параметра. Всего имеем $2n+3$ независимых параметра. Заметим, что если t_j находится в

∞ , то член $(t-t_j)^{\frac{\alpha_j}{\pi}-1}$ в интеграле отсутствует. При отображении

Кристоффеля-Шварца, при сохранении направления обхода, можно произвольно задавать абсциссы трех точек в плоскости t , остальные необходимо определять. Реперная точка t_k задается таким образом, чтобы легко определить постоянную C_2 .

6.3. Щелевой отсос в неограниченном пространстве

Направим ось Ox по оси симметрии щелевого отверстия шириной $2B$ со скоростью отсоса воздуха на ∞ равной 1 (рис.6.4). Определим скорость воздуха на оси Ox .

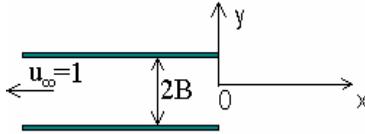


Рис.6.4. Щелевой отсос, свободно расположенный в пространстве

Поскольку данная область симметрична относительно оси абсцисс, то нижнюю часть рисунка можно отбросить.

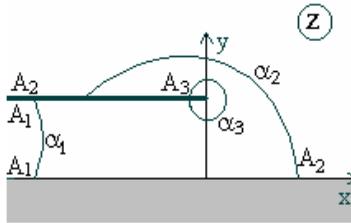


Рис.6.5. Физическая плоскость течения

Отобразим верхнюю полуплоскость (рис.6.6) на полученную треугольную область (рис.6.5). Точки A_1 , A_2 , A_3 отобразятся соответственно на t_1 , t_2 , t_3 .

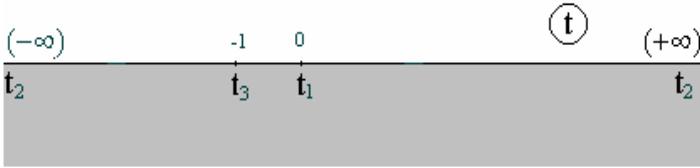


Рис.6.6. Геометрическая плоскость течения

Углы $\alpha_1 = 0, \alpha_3 = 2\pi$. Так как $\alpha_1 + \alpha_2 + \alpha_3 = \pi \Rightarrow \alpha_2 = \pi - 2\pi = -\pi$.

Интеграл Кристоффеля-Шварца для данного отображения примет вид

$$z = C_1 \int_{t_k}^t (t-0)^{0-1} (t+1)^{\frac{2\pi}{\pi}-1} dt + C_2. \quad (6.14)$$

В качестве реперной точки t_k лучше выбрать ту, которая дает возможность легко определить постоянную C_2 . В данном случае $t_k = t_3 = -1$. Тогда

$$z = C_1 \int_{-1}^t \frac{t+1}{t} dt + C_2. \quad (6.15)$$

Если принять $t = -1$ то $z = iB$ и $C_2 = iB$. Константу C_1 определим из соответствия точек A_1 и t_1 . Воспользуемся тем обстоятельством, что при переходе через точку t_1 по бесконечно малой окружности радиусом (с луча $t_1 t_2$ на отрезок $t_1 t_3$ происходит скачок на величину iB с $A_1 A_2$ на $A_1 A_3$ в плоскости z , то есть

$$z(+\epsilon) - z(-\epsilon) = 0 - iB = -iB. \quad (6.16)$$

Взяв интеграл (6.15) получим

$$z = C_1 (t + 1 + \ln t - i\pi) + iB.$$

Вычислим значения функции z в точках ε и $-\varepsilon$ при стремлении $\varepsilon \rightarrow 0$.

$$z(\varepsilon) = C_1(\varepsilon + 1 + \ln \varepsilon - i\pi) + iB, \quad z(-\varepsilon) = C_1(-\varepsilon + 1 + \ln \varepsilon + i\pi - i\pi) + iB.$$

Поэтому $\lim_{\varepsilon \rightarrow 0} (z(\varepsilon) - z(-\varepsilon)) = -i\pi C_1$. Используя (6.16) определяем

$$-iB = -i\pi C_1 \Rightarrow C_1 = B/\pi.$$

Таким образом искомое отображение имеет вид

$$z = \frac{B}{\pi}(t + 1 + \ln t). \quad (6.17)$$

На геометрической плоскости течения (рис.6.6) имеем точечный сток в точке t_1 . Поэтому комплексный потенциал этого течения

$w = -\frac{Q}{\pi} \ln t$. Комплексная скорость в физической плоскости (рис.6.5)

$$\frac{dw}{dz} = \frac{dw}{dt} \cdot \frac{dt}{dz} = -\frac{1}{t+1}. \quad (6.18)$$

Таким образом алгоритм расчета осевой скорости у щелевого отсоса, свободно расположенного в пространстве, состоит из четырех этапов:

1. Задается начальная координата $x=0$;
2. Из формулы $x = \frac{B}{\pi}(1+t + \ln t)$ (например, методом половинного деления) определяется параметр t ;
3. По формуле $u_x = -\frac{1}{t+1}$ рассчитывается осевая скорость воздуха в точке x ;
4. производится шаг $x = x + \Delta x$ и возврат к пункту 2. Расчет выполняется до заданной точки.

Пример 6.1. Произвести расчет осевой скорости у щелевого отсоса единичной полуширины и единичной скорости отсоса на бесконечности

```
program shel;
Uses crt,printer;
```

```

const k=1; {полуширина щели}
Var z,t,w:real;
procedure delenie(z:real; Var c:real); {Процедура вычисления
параметра методом половинного деления}
var a,b:real;
function f(t,z:real):real; begin f:=k/pi*(t+ln(t)+1)-z end;
begin
a:=0.0001; b:=30;
if f(a,z)*f(b,z)>0 then begin writeln('Корней нет'); Halt end; c:=(a+b)/2;
while abs(a-b)>0.000001 do begin if f(a,z)*f(c,z)<0 then b:=c else a:=c;
c:=(a+b)/2; end;
end;
{Основной блок программы}
begin clrscr; z:=0;
while z<=5*k do begin delenie(z,t); {Определение параметра t по
заданному z}
writeln('z=',z:0:3,' V=',-1/(1+t):0:3); z:=z+0.25*k; end;
end.

```

Табл.6.1.

Результаты расчета

x/B	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5
u	0,78	0,56	0,38	0,27	0,2	0,16	0,13	0,11	0,1	0,09	0,08

6.4. Щелевой отсос в плоской стенке

Также как и в предыдущем параграфе полагаем, что ширина щелевого отверстия $2B$, скорость отсоса $u_\infty = 1$ (рис.6.7). Определим скорость подтекания воздуха на оси абсцисс.

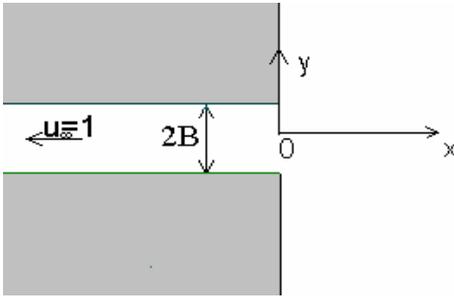


Рис.6.7. Щелевой отсос, встроенный в плоскую безграничную стенку

В силу симметрии области течения отбросим нижнюю полуплоскость (рис.6.8).

Отображение геометрической плоскости течения на физическую осуществляется интегралом Кристоффеля-Шварца

$$z = C_1 \int_0^t \frac{\sqrt{t}}{t-1} dt + C_2. \quad (6.18)$$

Постоянная $C_2 = iB$ так как $z = iB$ при $t = 0$. Проинтегрировав (6.18) получим

$$z = C_1 \left[2\sqrt{t} + \ln \left(\frac{\sqrt{t}-1}{\sqrt{t}+1} \right) - i\pi \right] + iB. \quad (6.19)$$

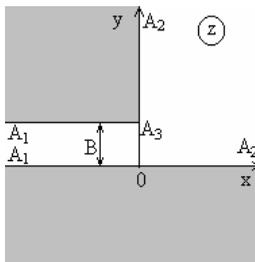


Рис.6.8. Физическая плоскость течения

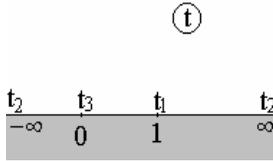


Рис. 6.9. Геометрическая плоскость течения

Постоянная C_1 определяется из соответствия точке t_1 и A_1 . При обходе точки t_1 по полуокружности радиуса $\varepsilon \rightarrow 0$ с луча $t_1 t_2$ на отрезок $t_3 t_1$ происходит скачок в физической плоскости на $\Delta z = -iB$.

Так как

$$\lim_{\varepsilon \rightarrow 0} [z(1+\varepsilon) - z(1-\varepsilon)] = -C_1 \cdot i\pi \Rightarrow C_1 = \frac{B}{\pi}$$

и искомое отображение примет вид

$$z = \frac{B}{\pi} \left[2\sqrt{t} + \ln \frac{\sqrt{t}-1}{\sqrt{t}+1} \right]. \quad (6.20)$$

Комплексный потенциал течения на геометрической плоскости

$$w = -\frac{Q}{\pi} \ln(t-1),$$

поэтому комплексная скорость

$$\frac{dw}{dz} = \frac{dw}{dt} \frac{dt}{dz} = -\frac{1}{\sqrt{t}}. \quad (6.21)$$

Алгоритм расчета состоит из следующих шагов:

1. задается начальная координата $x=0$;

2. из формулы $x = \frac{B}{\pi} \left(\sqrt{t} + \ln \frac{\sqrt{t}-1}{\sqrt{t}+1} \right)$ определяется параметр t ;

3. по формуле $u_x = -\frac{1}{\sqrt{t}}$ рассчитывается осевая скорость воздуха в точке x ;

4. производится шаг по $x = x + \Delta x$ и возврат к пункту 2. Расчет выполняется до заданной точки.

Пример 6.2. Произвести расчет осевой скорости у щелевого отсоса единичной полуширины и единичной скорости отсоса на бесконечности

```

program shel;
Uses crt,printer;
const k=1; { Полуширина щели}
Var z,t,w:real;
procedure delenie(z:real; Var c:real);
var a,b:real;
function f(t,z:real):real;
begin f:=z-k/pi*(2*sqrt(t)+ln((sqrt(t)-1)/(sqrt(t)+1))) end;
begin a:=1.00001; b:=3000;
if f(a,z)*f(b,z)>0 then begin writeln('Корней нет'); Halt end;
c:=(a+b)/2;
while abs(a-b)>0.000001 do begin if f(a,z)*f(c,z)<0 then b:=c else a:=c;
c:=(a+b)/2; end;
end;
{ Тело программы}
begin
clrscr; z:=0;
while z<=5 do begin
delenie(z,t); writeln(1st,'z=',z:0:3, ' V=',-1/sqrt(t):0:3); z:=z+0.25;
end;
end.

```

Табл. 6.2

Результаты расчета

x/B	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5
u	0,8 3	0,6 5	0,4 8	0,3 7	0,2 9	0,2 4	0,2	0,1 8	0,1 6	0,1 4	0,1 3

Из результатов расчета (табл.6.1-6.2) видно, что осевая скорость у щелевого отсоса, встроенного в плоскую безграничную стенку выше, чем у отсоса свободно расположенном в пространстве, что объясняется влиянием границ подтекания газа. В первом случае область подтекания меньше, чем во втором. Заметим, что если скорость отсоса воздуха $u_\infty \neq 1$ то все полученные результаты необходимо помножить на эту величину.

7. Метод Н.Е. Жуковского

7.1. Общие сведения о методе Н.Е. Жуковского

Примем следующие классические допущения, применяемые для расчета всасываемых воздушных течений: жидкость невесома, несжимаема, вихри отсутствуют, течение стационарное. Нормальная составляющая скорости на твердых стенках и неизвестных свободных линиях тока (образующихся при срыве потока) равна нулю. Кроме того на свободных линиях тока модуль скорости u_0 постоянен. В излагаемом подходе свободная линия тока - это граница между вихревой областью, возникающей у стенок и потенциальным течением, которое будем исследовать. Отсюда следует, что течение определено, если известен потенциал скоростей $w(z)$, где $z=x+iy$ - физическая плоскость течения воздуха.

Функция Жуковского имеет вид

$$\begin{aligned} \omega &= -\ln \frac{dw}{u_0 dz} = -\ln \frac{u_x - iu_y}{u_0} = -\ln \frac{u(\cos\theta - i \sin\theta)}{u_0} = \\ &= -\ln \left(\frac{u}{u_0} e^{-i\theta} \right) = -\ln \frac{u}{u_0} + i\theta, \end{aligned} \quad (7.1.1)$$

где θ - угол наклона скорости \vec{u} к положительному направлению оси Ox . Необходимо плоскость функции Жуковского ω отобразить на

параметрическую плоскость t , где известен комплексный потенциал течения, затем связать физическую плоскость z с этой параметрической плоскостью t . Если отображение $\omega = \omega(t)$ известно то из выражения

$\omega(t) = -\ln \frac{dw}{u_0 dz}$ определяем, что

$$z = \frac{1}{u_0} \int e^{\omega(t)} \frac{dw}{dt} dt . \quad (7.1.2)$$

Воспользовавшись этой формулой можно найти искомую комплексную скорость

$$\frac{dw}{dz} = \frac{dw}{dt} \frac{dt}{dz} = u_0 e^{-\omega(t)} . \quad (7.1.3)$$

7.2. Щелевой отсос в неограниченном пространстве

Пусть из щелевого отсоса шириной $2B$ (рис.7.5) воздух отсасывается со скоростью u_0 . Ширина струи на бесконечности 2δ . Необходимо определить величину скорости на оси OX .

В силу симметрии области течения (рис.7.5) нижнюю её часть отбрасываем.

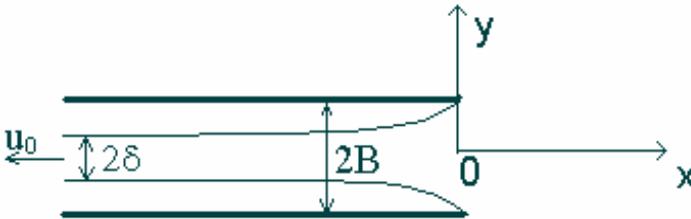


Рис.7.5. Отрыв струи у щелевого отсоса, свободно расположенном в пространстве

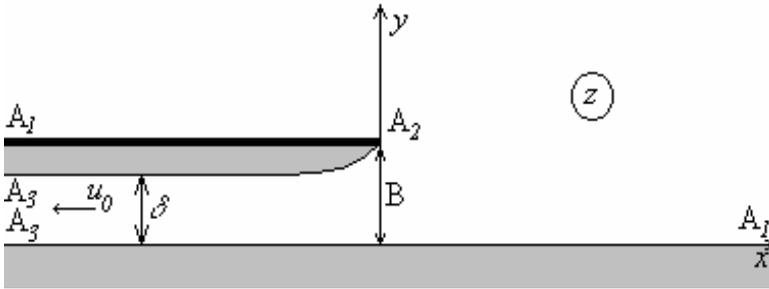


Рис. 7.6. Физическая плоскость течения

На луче A_1A_2 величина скорости изменяется от 0 до u_0 и действительная часть функции Жуковского $\operatorname{Re} \omega$ уменьшается от ∞ до 0. Направление скорости не изменяется, угол наклона с Ox $\theta = 0$ и $\operatorname{Im} \omega = 0$. На линии A_2A_3 скорость равна u_0 и $\operatorname{Re} \omega = 0$; угол наклона скорости к оси Ox и соответственно мнимая часть функции Жуковского $\operatorname{Im} \omega$ изменяется от 0 до $-\pi$. На прямой A_3A_1 $\operatorname{Re} \omega$ увеличивается от 0 до ∞ , а $\operatorname{Im} \omega = -\pi$. Таким образом область функции Жуковского - полуплоска (рис.7.7).

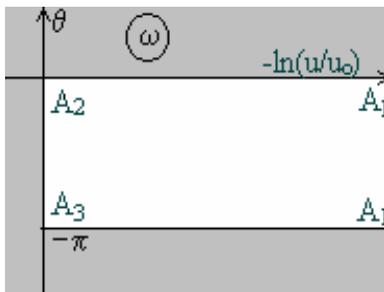


Рис.7.7. Область функции Жуковского

Отображение области ω на параметрическую плоскость t (рис.7.4) осуществляется интегралом Кристоффеля-Шварца

$$\omega = C_1 \int_0^t \frac{dt}{\sqrt{t}\sqrt{t+1}} + C_2. \quad (7.1.13)$$

Постоянная C_1 определяется из соответствия $t = 0 \Rightarrow \omega = -i\pi \Rightarrow C_1 = -i\pi$. Проинтегрировав (7.1.13) и используя соответствие $t = -1 \Rightarrow \omega = 0$ определим $C_1 = 1$. Таким образом

$$\omega = 2\ln(\sqrt{t} + \sqrt{t+1}) - i\pi. \quad (7.1.14)$$

Используя (7.1.2) найдем связь между физической плоскостью течения и параметрической

$$z = C_1 \frac{\delta}{\pi} \int_{-1}^t \frac{(\sqrt{t} + \sqrt{t+1})^2}{t} dt + C_2, \quad (7.1.15)$$

где $C_1 = iB$, так как при $t = -1 \Rightarrow z = iB$.

Интегрируя (7.1.15) и используя соответствие $t = +\varepsilon \rightarrow 0 \Rightarrow z = -\infty$ и $t = -\varepsilon \rightarrow 0 \Rightarrow z = -\infty + i\delta$ получим $C_1 = 1$, $\delta = B/2$ и поэтому

$$z = \frac{B}{2\pi} \left[(\sqrt{t} + \sqrt{t+1})^2 + 1 + \ln t + 2\ln(\sqrt{t} + \sqrt{t+1}) \right]. \quad (7.1.16)$$

Комплексная скорость

$$\frac{dw}{dz} = -\frac{u_0}{(\sqrt{t} + \sqrt{t+1})^2}. \quad (7.1.17)$$

Тогда алгоритм расчета осевой скорости у щелевого отсоса, свободно расположенного в пространстве, состоит из четырех этапов:

1. Задается начальная координата $x=0$;
2. Из формулы

$$x = \frac{B}{2\pi} \left[\left(\sqrt{t} + \sqrt{t+1} \right)^2 + 1 + \ln t + 2 \ln \left(\sqrt{t} + \sqrt{t+1} \right) \right]$$

определяется параметр t ;

3. По формуле $u_x = -\frac{u_0}{\left(\sqrt{t} + \sqrt{t+1} \right)^2}$ рассчитывается осевая скорость

воздуха в точке x ;

4. Производится шаг по $x = x + \Delta x$ и возврат к пункту 2. Расчет выполняется до заданной точки.

Из результатов расчета при $u_0 = 1$ (табл.7.1-7.2) видно, что как и для безотрывной модели течения, эффективнее отсос встроенный в плоскую стенку.

Пример 7.1. Произвести расчет осевой скорости у щелевого отсоса единичной поуширины и единичной скорости отсоса воздуха на бесконечности.

```

program shel;
Uses crt,printer;
const k=1; {Поуширина щели}
Var z,t,w:real;
procedure delenie(z:real; Var c:real);
var a,b:real;
function f(t,z:real):real;
begin
begin
k/2/pi*(2*(t+1)+ln(t)+2*sqrt(t+1)*sqrt(t)+2*ln(sqrt(t+1)+sqrt(t)))
end;
begin a:=0.000000001; b:=30;
if f(a,z)*f(b,z)>0 then begin writeln('Корней нет'); Halt end;
c:=(a+b)/2;
while abs(a-b)>0.0000000001 do
begin if f(a,z)*f(c,z)<0 then b:=c else a:=c; c:=(a+b)/2; end;
end;
{Тело программы}
begin clrscr; z:=0;
while z<=5*k do
begin delenie(z,t);
f:=z-

```

```
writeln(lst,'z=',z:0:3,' V=',-1/(sqrt(sqrt(t+1)+sqrt(t))):0:3);
z:=z+0.25*k;
end; end.
```

Табл. 7.1

Результаты расчета

x/B	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5
u	0,6 4	0,3 9	0,2 3	0,1 6	0,1 1	0,0 9	0,0 7	0,0 6	0,0 5	0,0 4	0,03 9

7.3. Щелевой отсос в плоской стенке.

Пусть воздух отсасывается со скоростью u_0 из щелевого отверстия шириной $2B$. С кромок отсоса происходит отрыв потока. Задача состоит в определении ширины потенциального течения 2δ на ∞ и скорости воздуха на оси OX (рис.7.1).

В силу симметрии области течения отбросим нижнюю полуплоскость (рис.7.2).

На линии A_1A_2 скорость изменяется от 0 до u_0 , а угол наклона $\theta = -\pi/2$ постоянен. Действительная часть функции Жуковского $\operatorname{Re} \omega$ при этом убывает от ∞ до 0, мнимая часть $\operatorname{Im} \omega = -\pi/2$. На A_2A_3 скорость постоянна и равна u_0 , а угол наклона скорости θ изменяется от $-\pi/2$ до $-\pi$, то есть $\operatorname{Re} \omega = 0$ и $\operatorname{Im} \omega$ уменьшается от $-\pi/2$ до $-\pi$. На прямой A_3A_1 скорость убывает от u_0 до 0 ($\operatorname{Re} \omega$ увеличивается от 0 до ∞), а угол $\theta = \operatorname{Im} \omega = -\pi$ постоянен. Таким образом область функции Жуковского - полулопа (рис.7.3).

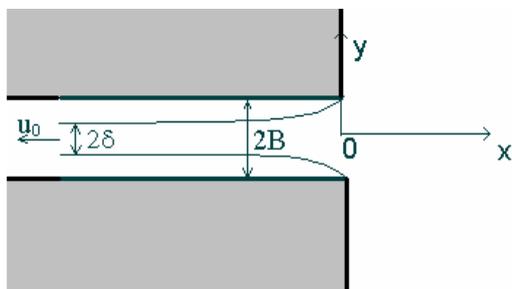


Рис.7.1. Отрывное течение газа у щелевого отсоса, встроенного в плоскую безграничную стенку

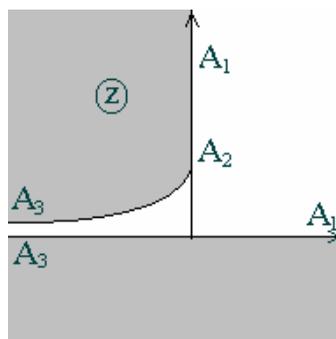


Рис.7.2. Физическая плоскость течения

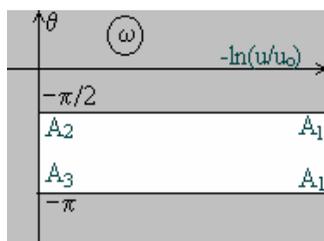


Рис.7.3. Плоскость функции Жуковского

Отобразим область функции Жуковского Н.Е. на параметрическую плоскость t (рис.7.3) с помощью интеграла Кристоффеля-Шварца

$$\omega = C_1 \int_{-1}^t \frac{dt}{\sqrt{t+1}\sqrt{t}} + C_2, \quad (7.2.4)$$

где $C_2 = -\pi/2 \cdot i$, так как при $t = -1 \Rightarrow \omega = -\pi/2 \cdot i$. Проинтегрировав получим $\omega = C_1 \cdot (2 \ln|\sqrt{t+1} + \sqrt{t}| - i\pi) - \pi/2 \cdot i$. Постоянную C_1 определим из соответствия точки A_3 . При $t = 0$ функция $\omega = -i\pi$, поэтому $C_1 = 1/2$ и

$$\omega = \ln(\sqrt{t+1} + \sqrt{t}) - i\pi. \quad (7.2.5)$$

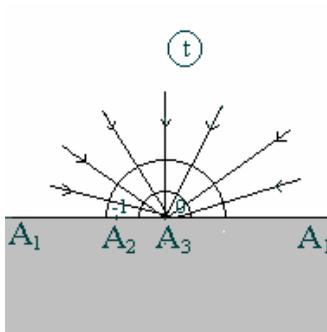


Рис.7.4. Параметрическая плоскость.

Найдем связь между физической плоскостью z и параметрической t воспользовавшись (7.2.2)

$$z = \frac{1}{u_0} C_1 \int_{-1}^t e^{\ln(\sqrt{t+1}+\sqrt{t})-i\pi} \cdot \frac{dw}{dt} dt + C_2. \quad (7.2.6)$$

Так как в плоскости t находится точечный сток в точке A_3 то комплексная скорость $\frac{dw}{dt} = -\frac{Q}{\pi t} = -\frac{u_0 \delta}{\pi t}$, поэтому после несложных преобразований (7.2.6) получим

$$z = C_1 \frac{\delta}{t} \int_{-1}^t \left(\frac{\sqrt{t+1}}{t} + \frac{1}{\sqrt{t}} \right) dt + C_2, \quad (7.2.7)$$

где $C_2 = iB$, так как при $t = -1 \Rightarrow z = iB$. Проинтегрировав получим

$$z = C_1 \frac{\delta}{\pi} \left[2(\sqrt{t+1} + \sqrt{t}) + \ln \frac{\sqrt{t+1}-1}{\sqrt{t+1}+1} - i(\pi+2) \right] + iB. \quad (7.2.8)$$

Постоянную C_1 определим из соответствия точки A_3 в плоскости t и z . При $t = \varepsilon \rightarrow 0$ функция $z = -\infty$ (рис.7.2 и рис.7.4). Тогда подставляя в (7.2.8) получим

$$-\infty = C_1 \frac{\delta}{\pi} [2 - \infty - i(\pi+2)] + iB. \quad (7.2.9)$$

Приравнявая мнимые части найдем $C_1 = \frac{B\pi}{(\pi+2)\delta}$. Осталось определить полуширину струи на бесконечности δ . При $t = -\varepsilon \rightarrow 0$ функция $z = -\infty + i\delta$ (рис.7.2, рис.7.4). Из (7.2.8) получим

$$-\infty + i\delta = C_1 \frac{\delta}{\pi} [2 - \infty + i\pi - i(\pi+2)] + iB. \quad (7.2.10)$$

Из равенства мнимых частей найдем $\delta = \frac{B\pi}{\pi + 2C_1}$ и используя выражение для C_1 окончательно определим $C_1 = 1$, $\delta = B \frac{\pi}{\pi + 2}$ и подставляя в (7.2.8) имеем

$$z = \frac{B}{\pi + 2} \left[2(\sqrt{t+1} + \sqrt{t}) + \ln \frac{\sqrt{t+1} - 1}{\sqrt{t+1} + 1} \right]. \quad (7.2.11)$$

Комплексная скорость в плоскости z

$$\frac{dw}{dz} = \frac{dw}{dt} \frac{dt}{dz} = -\frac{u_0}{(\sqrt{t+1} + \sqrt{t})}. \quad (7.2.12)$$

Алгоритм расчета осевой скорости у щелевого отсоса, встроенного в плоскую безграничную стенку, состоит из следующих шагов:

1. Задается начальная координата $x=0$;
2. Из формулы

$$x = \frac{B}{\pi + 2} \left[2(\sqrt{t+1} + \sqrt{t}) + \ln \left(\frac{\sqrt{t+1} - 1}{\sqrt{t+1} + 1} \right) \right]$$

определяется параметр t ;

3. По формуле

$$u_x = -\frac{u_0}{\sqrt{t+1} + \sqrt{t}}$$

рассчитывается осевая скорость воздуха в точке x ;

4. Производится шаг по $x = x + \Delta x$ и возврат к пункту 2, пока не будет достигнута заданная точка.

Если скорость отсоса воздуха задать $u_0 = 1$ то осевая скорость примет значения, приведенные в табл. 2.1.

Пример 7.2. Произвести расчет осевой скорости у щелевого отсоса единичной поуширины и единичной скорости отсоса воздуха на бесконечности.

```

program shel;
Uses crt,printer;
const
k=1; { Полуширина щели
Var z,t,w:real;
procedure delenie(z:real; Var c:real);
var a,b:real;
function f(t,z:real):real;
begin
f:=z-k/(pi+2)*(2*sqrt(t+1)+ln((sqrt(t+1)-
1)/(sqrt(t+1)+1))+2*sqrt(t))
end;

begin
a:=0.000000001; b:=300;
if f(a,z)*f(b,z)>0 then begin Halt end;
c:=(a+b)/2;
while abs(a-b)>0.0000001 do begin if f(a,z)*f(c,z)<0 then b:=c else a:=c;
c:=(a+b)/2; end;
end;

begin clrscr;
z:=0;
while z<=5*k do begin delenie(z,t);
writeln(1st,'z=',z:0:3, ' V=',-1/(sqrt(t)+sqrt(t+1)):0:3); z:=z+0.25*k;
end; end.

```

Табл. 7.2

Результаты расчета

x/B	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5
u	0,6 5	0,4 5	0,3 1	0,2 3	0,1 8	0,1 5	0,1 3	0,1 1	0,1	0,0 9	0,08

8. Метод ГИУ для расчета плоских потенциальных течений

Обобщенный метод наложения потоков или “особенностей”, названный позднее методом граничных интегральных уравнений (ГИУ) или непрямым методом граничных элементов, основывается на теореме выведенной Ламбом Г.: “...всякий вообще потенциальный поток может быть получен от определенной системы источников и стоков распределенных по границе области”. Решение уравнения Лапласа, записанного в интегральном виде, при заданных граничных условиях дает возможность такую систему определить.

Пусть дана некоторая область A течения воздуха, ограниченная контуром S (рис.8.1). Источники и стоки внутри области отсутствуют. На границе S задана нормальная составляющая скорости $\vec{v}_n(\xi)$, где ξ произвольная точка S (на твердых участках границы области течения нормальная составляющая скорости равна 0, т.е. задано условие непроницаемости).

Необходимо определить скорость воздуха во внутренних точках области вдоль любого заданного направления $\vec{v}_n(x)$, $x \in A$.

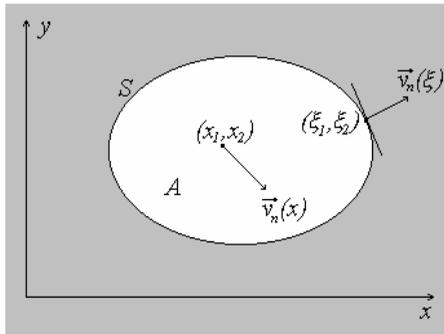


Рис.8.1. Схема течения воздуха в области A

Для потенциала ϕ справедливо уравнение Лапласа, которое запишем в полярной системе координат

$$\frac{\partial}{\partial r} \left(r \frac{\partial \phi}{\partial r} \right) + \frac{1}{r} \frac{\partial^2 \phi}{\partial \psi^2} = 0, \quad (8.1)$$

где r, ψ - полярные радиус и угол.

В случае когда потенциал не зависит от угла ψ (круговая симметрия) уравнение (8.1) упрощается

$$\frac{d}{dr} \left(r \frac{d\varphi}{dr} \right) = 0. \quad (8.2)$$

Дважды проинтегрировав (8.2) имеем

$$\varphi = C_1 \ln r + C_2. \quad (8.3)$$

Выражение (8.3) в курсах математической физики при $C_1 = -1$ и $C_2 = 0$ называется фундаментальным решением уравнения Лапласа в плоскости.

Если полюс полярной системы координат поместить в точку (x_1, x_2) , а в точку (ξ_1, ξ_2) источник (сток) интенсивностью $q(\xi)$, в качестве постоянных в (8.3) принять $C_1 = q(\xi)/2\pi$, $C_2 = 0$, то уравнение (8.3) преобразуется к виду

$$\varphi(x) = \frac{q(\xi)}{2\pi} \ln r, \quad (8.4)$$

где расстояние между точками x и ξ : $r = \sqrt{(x_1 - \xi_1)^2 + (x_2 - \xi_2)^2}$.

Таким образом, фундаментальное решение уравнения Лапласа (с точностью до постоянной) представляет собой значение потенциала в произвольной точке x , который обусловлен действием в некоторой точке ξ источника (стока) интенсивностью $q(\xi)$.

Величина скорости воздуха v_n в точке (x_1, x_2) вдоль любого заданного направления $\vec{n} \{n_1, n_2\}$, вызываемая действием источника (стока) в точке (ξ_1, ξ_2) определяется из следующего выражения

$$\begin{aligned} v_n = \vec{v} \cdot \vec{n} &= v_1 \cdot n_1 + v_2 \cdot n_2 = \frac{\partial \varphi}{\partial x_1} n_1 + \frac{\partial \varphi}{\partial x_2} n_2 = \\ &= \frac{q(\xi)}{2\pi r^2} \left[n_1(x_1 - \xi_1) + n_2(x_2 - \xi_2) \right] \end{aligned} \quad (8.5)$$

Пусть на границе S , исследуемой области A распределены источники (стоки) неизвестной интенсивностью $q(\xi)$, $\xi \in S$. Проинтегрировав их действие получим величину скорости воздуха во внутренней точке (x_1, x_2) области A вдоль заданного направления \vec{n}

$$v_n(x) = \int_S q(\xi) F(x, \xi) dS(\xi), \quad (8.6)$$

где $F(x, \xi) = \frac{1}{2\pi r^2} [n_1(x_1 - \xi_1) + n_2(x_2 - \xi_2)]$.

Для того чтобы определить неизвестную величину интенсивности $q(\xi)$ устремим точку $x(x_1, x_2)$ к точке x_0 границы S в направлении единичного вектора внешней нормали \vec{n} получим уравнение

$$v_n(x_0) = \int_S q(\xi) F(x_0, \xi) dS(\xi), \quad (8.7)$$

где интеграл является интегралом в смысле главного значения, так как функция $F(x_0, \xi)$ неограниченно возрастает при стремлении x_0 к ξ . Для того чтобы избавиться от этой особенности окружим точку ξ (рис.8.2) полуокружностью радиуса $\varepsilon \rightarrow 0$. Тогда интеграл в (8.7) можно представить в виде суммы двух

$$\int_S = \int_{S-S_\varepsilon} + \int_{S_\varepsilon}$$

где первый интеграл берется по контуру S без указанной полуокружности, второй - интеграл по дуге S_ε полуокружности.

Интегрируя по дуге полуокружности S_ε и предполагая, что величина $q(\xi)$ на этой дуге постоянна получим

$$\int_{S_\varepsilon} F(x_0, \xi) dS = \frac{1}{2\pi} \int_{S_\varepsilon} \frac{(x_{01} - \xi_1)n_1 + (x_{02} - \xi_2)n_2}{r^2} dS =$$

$$= \left\{ \begin{array}{l} x_{01} - \xi_1 = \varepsilon \cos \theta, \quad x_{02} - \xi_2 = \varepsilon \sin \theta, \quad r = \varepsilon \\ n_1 = -\cos \theta, \quad n_2 = -\sin \theta, \quad dS = \varepsilon d\theta. \end{array} \right\} = -\frac{1}{2}$$

и соответственно (8.7) можно записать в виде

$$v_n(x_0) = -\frac{1}{2}q(x_0) + \int_S q(\xi)F(x_0, \xi)dS(\xi). \quad (8.8)$$

Первое слагаемое (8.8) соответствует случаю когда x_0 совпадает с ξ , второе слагаемое - интеграл в обычном смысле, контур интегрирования которого не содержит точку $\xi = x_0$.

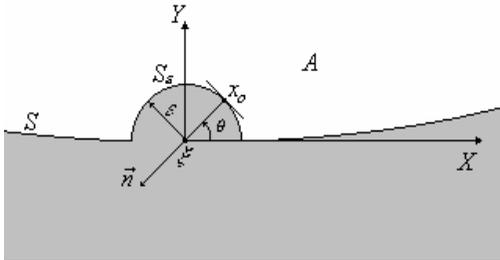


Рис. 8.2. Граница области без точки ξ

Если бы интеграл в (8.8) вычислялся аналитически, то разрешая данное уравнение относительно $q(\xi)$ и подставляя эту величину в (8.6) получили бы решение поставленной задачи. Однако в реальных задачах таких случаев практически нет и необходимо численное решение уравнения (8).

Разобьем границу S на N прямолинейных отрезков. Предполагаем, что на каждом из полученных отрезков, например k -м, интенсивность источников (стоков) постоянна и равна q^k . Суммируя действия всех таких источников на центр p -го отрезка получим дискретный аналог (8.8)

$$v_n(x_o^p) = -\frac{1}{2}q^p + \sum_{\substack{k=1, \\ k \neq p}}^N q^k \int_{S^k} F(x_o^p, \xi^k) dS(\xi^k), \quad (8.9)$$

где ξ^k - произвольная точка k -го отрезка.

Интегралы в (8.9) берутся по k -отрезкам (S^k). Скорость $v_n(x_o^p)$ известна из граничных условий поставленной задачи. Изменяя p от 1 до N получим систему N линейных уравнений с N неизвестными. Решая полученную систему (8.9) найдем $q^k, k=1,2,\dots,N$. Скорость во внутренней точке x вдоль заданного направления определяется по формуле

$$v_n(x) = \sum_{k=1}^N q^k \int_{S^k} F(x, \xi^k) dS(\xi^k). \quad (8.10)$$

Таким образом алгоритм решения плоской задачи состоит из 3-х этапов:

- 1) разбиение границы (дискретизация) исследуемой области течения воздуха на прямолинейные отрезки (граничные элементы);
- 2) решение системы линейных уравнений (8.9) и определение интенсивности источников $q^k, k=1,2,\dots,N$;
- 3) определение скорости воздуха в интересующей нас точке (8.10).

8.2. Дискретизация границы области

Дискретизация границы S области на граничные элементы осуществляется набором прямолинейных отрезков, заданных в глобальной прямоугольной декартовой системе координат (ГПДСК). Каждый граничный отрезок имеет свой номер и координаты начала “а” и конца “b”. Причем “а” и “b” выбираются так, чтобы единичный вектор нормали к данному отрезку был направлен вне области.

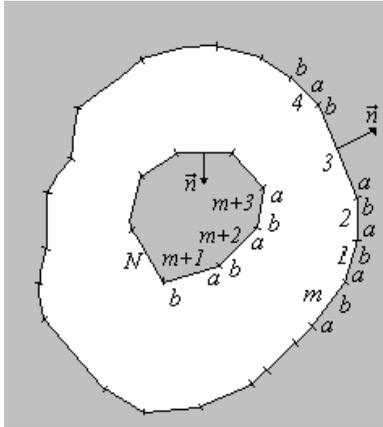


Рис. 8.3. Дискретизация границы области на N прямолинейных отрезков

8.3. Расчет интенсивностей источников (стоков)

Для того чтобы определить интенсивность источников (стоков) необходимо вначале сформировать матрицу

$$F = (F^{pk}) = \left(\int_{S^k} F(x_o^p, \xi^k) dS(\xi^k) \right),$$

составленной из коэффициентов при

неизвестных q^k в системе линейных алгебраических уравнений (8.9), где $p=1,2,\dots,N$ и $q=1,2,\dots,N$.

Интегралы F^{pk} вычисляются аналитически. Пусть центр p -го отрезка - точка x^p задана в ГПДСК XOY , центр k -го отрезка $[a,b]$ - точка c^k также задана в ГПДСК. Построим локальную прямоугольную декартову систему координат (ЛПДСК) с центром в точке c^k , ось ординат ξ_2 направим по вектору \vec{ab} , а ось абсцисс ξ_1 расположим так, чтобы полученная система координат была правой (рис.8.4). ЛПДСК $\eta_1\eta_2$ получается из $\xi_1\xi_2$ параллельным переносом в точку x^p .

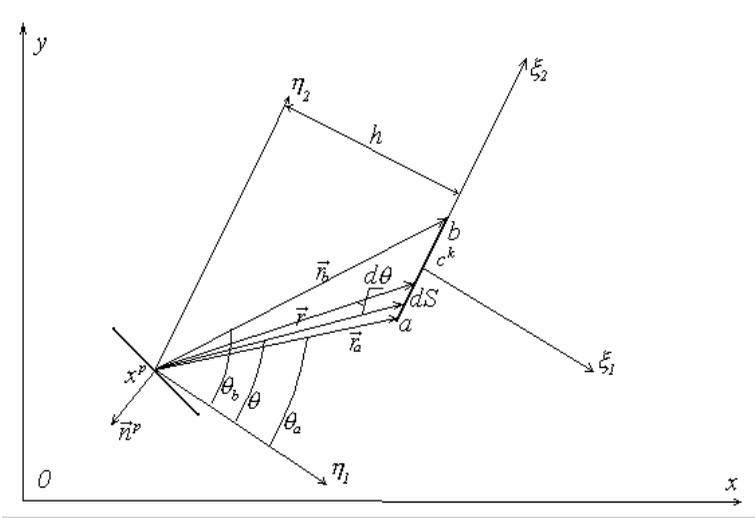


Рис. 8.4. К построению локальных систем координат

В ЛПДСК $\eta_1 \eta_2$ для координат произвольной точки ξ^k k -го отрезка справедливы равенства $\eta_1 = h$, $\eta_2 = h \cdot \operatorname{tg} \theta$, а для точки x^p : $\eta_1 = 0$, $\eta_2 = 0$. Обозначая $\{n_1, n_2\}$ координаты единичного вектора внешней нормали к p -му отрезку в системе координат $\eta_1 \eta_2$ получим для $F(x_o^p, \xi^k)$ следующее выражение

$$F(x_o^p, \xi^k) = \frac{(0-h)n_1 + (0-h \operatorname{tg} \theta)n_2}{2\pi((0-h)^2 + (0-h \operatorname{tg} \theta)^2)} = -\frac{n_1 + n_2 \operatorname{tg} \theta}{2\pi h} \cos^2 \theta.$$

Поэтому интеграл

$$F^{pk} = \int_{S^k} F(x_o^p, \xi^k) dS(\xi^k) = \left\{ dS = d\eta_2 = \frac{h}{\cos^2 \theta} d\theta \right\} = -\frac{1}{2\pi} \int_{\theta_a}^{\theta_b} (n_1 + n_2 \cdot \operatorname{tg} \theta) d\theta$$

Проинтегрировав получим

$$F^{pk} = -\frac{1}{2\pi} \left[n_1(\theta_b - \theta_a) + n_2 \cdot \ln \frac{r_b}{r_a} \right], \quad (8.11)$$

где r_b, r_a - длины соответствующих радиусов-векторов концов "a" и "b" k -го отрезка.

Рассмотрим более подробно вычисление параметров формулы (8.11).

Пусть даны координаты в ГПДСК отрезка $[a, b]$: $a(a_1, a_2)$; $b(b_1, b_2)$. Построим на этом отрезке ЛПДСК как показано на рис.8.4. и вычислим координаты ортов полученной системе координат. Так как

$$\vec{e}_2 = \frac{\vec{b} - \vec{a}}{|\vec{b} - \vec{a}|},$$

где \vec{a}, \vec{b} - радиус-векторы точек "a" и "b" в ГПДСК то

$$\begin{cases} e_{2x} = \frac{b_1 - a_1}{\sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}}, \\ e_{2y} = \frac{b_2 - a_2}{\sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}}. \end{cases} \quad (8.12)$$

Координаты орта оси абсцисс найдем из условий перпендикулярности векторов \vec{e}_1, \vec{e}_2 и равенства вектора $\vec{e}_1 \times \vec{e}_2$ орту \vec{k} ГПДСК, так как ЛПДСК правая

$$\begin{aligned} \vec{e}_1 \cdot \vec{e}_2 &= e_{1x} \cdot e_{2x} + e_{1y} \cdot e_{2y} = 0, \\ |\vec{e}_1 \times \vec{e}_2| &= e_{1x} \cdot e_{2y} - e_{1y} \cdot e_{2x} = 1. \end{aligned}$$

Откуда получим

$$e_{1x} = e_{2y}, e_{1y} = -e_{2x}. \quad (8.13)$$

Пусть известны координаты некоторого вектора $\vec{m} = \{m_1, m_2\}$ в ГПДСК. Определим его координаты в ЛПДСК. В ГПДСК

$$\begin{cases} \vec{e}_1 = e_{1x} \cdot \vec{i} + e_{1y} \cdot \vec{j}, \\ \vec{e}_2 = e_{2x} \cdot \vec{i} + e_{2y} \cdot \vec{j}. \end{cases}$$

Откуда

$$\begin{cases} \vec{i} = e_{1x} \cdot \vec{e}_1 + e_{2x} \cdot \vec{e}_2, \\ \vec{j} = e_{1y} \cdot \vec{e}_1 + e_{2y} \cdot \vec{e}_2. \end{cases}$$

Вектор

$$\begin{aligned} \vec{m} &= m_1 \vec{i} + m_2 \vec{j} = m_1 (e_{1x} \vec{e}_1 + e_{2x} \vec{e}_2) + m_2 (e_{1y} \vec{e}_1 + e_{2y} \vec{e}_2) = \\ &= (m_1 e_{1x} + m_2 e_{1y}) \vec{e}_1 + (m_1 e_{2x} + m_2 e_{2y}) \vec{e}_2. \end{aligned}$$

Таким образом, координаты вектора \vec{m} в ЛПДСК

$$\begin{cases} m'_1 = m_1 e_{1x} + m_2 e_{1y}, \\ m'_2 = m_1 e_{2x} + m_2 e_{2y}. \end{cases} \quad (8.14)$$

Пусть задана точка (a_1, a_2) в ГПДСК. Определим ее координаты в ЛПДСК $\eta_1 \eta_2$. При повороте глобальной системы координат, таким образом чтобы ось ординат стала сонаправленной с \vec{e}_2 , координаты точки преобразуются к виду

$$\begin{cases} a'_1 = a_1 e_{1x} + a_2 e_{1y}, \\ a'_2 = a_1 e_{2x} + a_2 e_{2y}. \end{cases} \quad (8.15)$$

При параллельном переносе в т. $x^P(x_1^P, x_2^P)$ координаты точки a станут соответственно

$$\begin{cases} a_1'' = a_1' - x_1^p, \\ a_2'' = a_2' - x_2^p. \end{cases} \quad (8.16)$$

Тогда с использованием формул (8.12-4.16) алгоритм вычисления F^{pk} будет состоять из следующих этапов:

1. Задается начальный номер p -го отрезка: $p=1$.
2. Вычисляются глобальные координаты ортов ЛПДСК, построенной на p -м отрезке

$$e_{2x}^p = \frac{b_1^p - a_1^p}{\sqrt{(b_1^p - a_1^p)^2 + (b_2^p - a_2^p)^2}}, \quad e_{2y}^p = \frac{b_2^p - a_2^p}{\sqrt{(b_1^p - a_1^p)^2 + (b_2^p - a_2^p)^2}},$$

$$e_{1x}^p = e_{2y}^p, \quad e_{1y}^p = -e_{2x}^p.$$

Тогда единичный вектор внешней нормали к p -му отрезку $\vec{n}^p = \vec{e}_1^p$, если \vec{e}_1^p направлен вне области течения, т.е. в случае когда при прохождении от “ a ” к “ b ” область течения находится слева. Если “ a ” и “ b ” выбраны таким образом, что при обходе по границе от “ a ” к “ b ” область находится справа то $\vec{n}^p = -\vec{e}_1^p$.

3. Задается начальный номер k -го отрезка: $k=1$.
4. Если $p=k$, то $F^{pk} = -0,5$ и переход к пункту 10.
5. Для нормального вектора \vec{n}^p вычисляются его координаты в ЛПДСК k -го отрезка. Для этого сначала определяются орты

$$e_{2x}^k = \frac{b_1^k - a_1^k}{\sqrt{(b_1^k - a_1^k)^2 + (b_2^k - a_2^k)^2}}, \quad e_{2y}^k = \frac{b_2^k - a_2^k}{\sqrt{(b_1^k - a_1^k)^2 + (b_2^k - a_2^k)^2}},$$

$$e_{1x}^k = e_{2y}^k, \quad e_{1y}^k = -e_{2x}^k, \quad (8.17)$$

а затем вычисляются искомые координаты

$$n_1^{pk} = n_1^p e_{1x}^k + n_2^p e_{1y}^k, \quad n_2^{pk} = n_1^p e_{2x}^k + n_2^p e_{2y}^k.$$

6. Вычисляются координаты радиус-вектора центра p -го отрезка в ПДСК

$$x_1^p = \frac{a_1^p + b_1^p}{2}, \quad x_2^p = \frac{a_2^p + b_2^p}{2}$$

и определяются его координаты в ЛПДСК k -го отрезка

$$x_1^{pk} = x_1^p e_{1x}^k + x_2^p e_{1y}^k, \quad x_2^{pk} = x_1^p e_{2x}^k + x_2^p e_{2y}^k.$$

7. Вычисляются координаты вершин k -отрезка a^k, b^k в системе координат p -го отрезка ($\eta_1 \eta_2$ на рис. 8.4)

$$\begin{cases} a_1^{kp} = a_1^k e_{1x}^k + a_2^k e_{1y}^k - x_1^{pk}, \\ a_2^{kp} = a_1^k e_{2x}^k + a_2^k e_{2y}^k - x_2^{pk}, \\ b_1^{kp} = b_1^k e_{1x}^k + b_2^k e_{1y}^k - x_1^{pk}, \\ b_2^{kp} = b_1^k e_{2x}^k + b_2^k e_{2y}^k - x_2^{pk}. \end{cases} \quad (8.18)$$

8. Определяются длины радиус-векторов точек a^k, b^k в системе координат p -го отрезка

$$r_a = \sqrt{(a_1^{kp})^2 + (a_2^{kp})^2}, \quad r_b = \sqrt{(b_1^{kp})^2 + (b_2^{kp})^2} \quad (8.19)$$

и углы

$$\theta_a = \begin{cases} \arctg \frac{a_2^{kp}}{a_1^{kp}}, & \text{если } a_1^{kp} \neq 0; \\ \frac{\pi}{2}, & \text{если } a_1^{kp} = 0, \end{cases} \quad \theta_b = \begin{cases} \arctg \frac{b_2^{kp}}{b_1^{kp}}, & \text{если } b_1^{kp} \neq 0; \\ \frac{\pi}{2}, & \text{если } b_1^{kp} = 0. \end{cases} \quad (8.20)$$

9. Вычисляется элемент матрицы

$$F^{pk} = -\frac{1}{2\pi} \left[n_1^{pk} (\theta_b - \theta_a) + n_2^{pk} \ln \frac{r_b}{r_a} \right].$$

10. Шаг по k : $k=k+1$ и переход к п. 4 до тех пор пока $k \leq N$.

11. Шаг по p : $p=p+1$ и переход к п. 2 до тех пор пока $p \leq N$.

После формирования элементов матрицы (F^{pk}) решается система уравнений

$$\begin{cases} F^{11}q^1 + F^{12}q^2 + \dots + F^{1N}q^N = v^1, \\ F^{21}q^1 + F^{22}q^2 + \dots + F^{2N}q^N = v^2, \\ \dots \\ F^{N1}q^1 + F^{N2}q^2 + \dots + F^{NN}q^N = v^N, \end{cases}$$

откуда определяются неизвестные интенсивности источников (стоков) q^1, q^2, \dots, q^N распределенные по N граничным отрезкам. Граничные значения нормальной составляющей скорости v^1, v^2, \dots, v^N заданы при постановке задачи.

8.4. Расчет скорости воздуха внутри области течения

Для того, чтобы определить скорость воздуха в произвольной точке $x(x_1, x_2)$ области течения A вдоль любого заданного направления $\vec{n} = \{n_1, n_2\}$ необходимо выполнить следующие шаги:

1. Задать $k=1$.

2. Вычислить координаты векторов \vec{e}_1^k, \vec{e}_2^k по формуле (8.17), а затем координаты вектора \vec{n} в ЛПДСК k -го отрезка

$$n_1^k = n_1 e_{1x}^k + n_2 e_{1y}^k, \quad n_2^k = n_1 e_{2x}^k + n_2 e_{2y}^k.$$

3. Вычисляются координаты т. x в ЛПДСК k -го отрезка

$$x_1^k = x_1 e_{1x}^k + x_2 e_{1y}^k, \quad x_2^k = x_1 e_{2x}^k + x_2 e_{2y}^k.$$

4. Определяются координаты т. a^k, b^k в ЛПДСК с центром в т. x и ортами \vec{e}_1^k, \vec{e}_2^k по формуле (8.18).

5. Находятся величины r_a, r_b по формулам (8.19) и углы θ_a, θ_b по формуле (8.20).

6. Вычисляется величина

$$f^k = -\frac{1}{2\pi} \left[n_1^k (\theta_b - \theta_a) + n_2^k \ln \frac{r_b}{r_a} \right].$$

7. Шаг по k : $k=k+1$ и переход к п.2. до тех пор пока $k \leq N$.

8. Вычисление скорости

$$v_n(x) = f^1 q^1 + f^2 q^2 + \dots + f^N q^N.$$

По приведенному алгоритму можно произвести расчет линий тока. Для этого необходимо задать точку, с которой будет начинаться расчет линии тока; вычислить горизонтальную v_x и вертикальную v_y составляющие скорости течения в выбранной ГПДСК; определить направление скорости течения \vec{v} ; сделать шаг в данном направлении и вновь выполнить изложенный порядок вычисления. Расчет прекращается, когда достигается заданное сечение (например, линия отсоса воздуха). Данный алгоритм расчета линий тока соответствует методу Эйлера решения обыкновенного дифференциального уравнения (ОДУ)

$$\frac{dx}{v_x} = \frac{dy}{v_y}$$

при заданных начальных условиях. Заметим однако, что расчет линий тока можно произвести и по более точным методам решения ОДУ (например, методом Рунге-Кутты).

Пример 8.1. Построить линии тока и произвести расчет траектории пылевой частицы при бурении скважин.

Наиболее эффективный [103] пылешламоприемник представляет собой насадок в виде кругового цилиндра высотой 0,08 м с боковым проемом равным диаметру буровой штанги (0,03 м) и отводящим патрубком для подключения к побудителю тяги. Задача состоит в определении объема аспирируемого воздуха, позволяющего эффективно улавливать пыли и шлам, образующихся при бурении. Подсосами воздуха в кольцевом зазоре вокруг буровой штанги пренебрегаем, что дает возможность свести задачу описания поля скоростей в пылешламоприемнике к плоскости.

Дискретизация границ области течения воздуха в пылешламоприемнике осуществлялась набором 492 граничных отрезков. Численная реализация метода ГИУ дала возможность построить линии тока и определить распределение скоростей в проеме пылешламоприемника и при удалении от него.

Экспериментально замеренная скорость истечения пыли и шлама из восстающих шпуров диаметром 40мм: $v_{эксн} = 6\text{м/с}$, скважин диаметром 75мм: $v_{эксн} = 7\text{м/с}$. Наибольшая крупность выбуриваемого материала 0,3-0,5мм, плотность $\rho_o = 3500\text{ кг/м}^3$.

Рассматривался трехмерный полет частицы. Плоский разрез пылешламоприемников представлен на рис.8.20. Ось OZ направлена вертикально вверх и проходит через ось симметрии буровой штанги. В виду большой крупности частицы стоковский закон ее обтекания не применим. Наиболее целесообразным представляется описание движения частицы по закону [81]

$$\frac{\pi d_p^3}{6} \rho_1 \frac{d\vec{v}_1}{dt} = \frac{\pi d_p^3}{6} \rho_1 \vec{g} - \psi S_m \rho \frac{|\vec{v}_1 - \vec{v}| \cdot (\vec{v}_1 - \vec{v})}{2}, \quad (8.21)$$

где \vec{v}, \vec{v}_1 - скорости среды и частицы соответственно; d_p - эквивалентный диаметр частицы; ρ, ρ_1 - плотность среды и частицы; $S_m = \pi d_p^2 / 4$ - площадь миделевого сечения.

В любой плоскости перпендикулярной буровой штанги предполагались равные поля составляющих v_x, v_y скорости воздуха, которые вычисляются по методу ГИУ. Вертикальная составляющая скорости воздуха вычисляется по формуле

$$v_z = -v_{\text{эксн}} + \sqrt{\frac{4\rho_1 d_3 g}{3\psi\rho}},$$

которая может быть получена из (8.21) в предположении, что частица пыли движется равномерно, т.е. $\frac{d\vec{v}_1}{dt} = 0$.

Коэффициент лобового сопротивления наиболее удобно представить в виде

$$\psi = \frac{24}{\text{Re}} + \psi_0 \quad (8.22)$$

где $\text{Re} = \frac{\rho d_3 |\vec{v}_1 - \vec{v}|}{\mu}$; ψ_0 - коэффициент найденный экспериментально для различных форм частицы. В наших расчетах $\psi_0 = 1,8$. Формула Озееновского типа (8.22), предложенная И.Н.Логачевым является наиболее приближенной к экспериментальным данным для изометрической формы частиц.

Уравнение (8.21) сводится к системе обыкновенных дифференциальных уравнений, которое решается численно, например, методом Рунге-Кутты.

$$\left\{ \begin{array}{l} A \frac{dv_{1x}}{dt} = -B(v_{1x} - v_x), \\ \frac{dx}{dt} = v_{1x}, \\ A \frac{dv_{1y}}{dt} = -B(v_{1y} - v_y), \\ \frac{dy}{dt} = v_{1y}, \\ A \frac{dv_{1z}}{dt} = -Ag - B(v_{1z} - v_z), \\ \frac{dz}{dt} = v_{1z}, \end{array} \right. \quad (8.23)$$

где

$$A = \rho_1 \frac{\pi d_2^3}{6},$$

$$B = -\Psi S_M \rho |\vec{v}_1 - \vec{v}| = -\Psi S_M \rho \sqrt{(v_{1x} - v_x)^2 + (v_{1y} - v_y)^2 + (v_{1z} - v_z)^2}$$

После удара о твердую поверхность частицы пыли тангенциальная $v_{2\tau}$ и нормальная v_{2n} скорости вычисляется по формулам [91]

$$\begin{aligned} v_{2n} &= -k \cdot v_{on}, \\ v_{2\tau} &= v_{o\tau} + \eta \cdot f \cdot (1+k) \cdot v_{on}, \end{aligned} \quad (8.24)$$

где $\eta = \min\left\{-\frac{2v_{o\tau}}{7f(1+k)v_{on}}, 1\right\}$, k - коэффициент восстановления при ударе (в наших расчетах $k=0,5$); f - коэффициент трения скольжения ($f=0,5$); $v_{on}, v_{o\tau}$ - нормальная и тангенциальная составляющие скорости частицы до удара. Исходные данные для расчета: плотность пылевой частицы $\rho_1 = 3500 \text{ кг/м}^3$, плотность воздуха $\rho = 1,205 \text{ кг/м}^3$; коэффициент динамической вязкости $\mu = 1,809 \cdot 10^{-5} \text{ Па} \cdot \text{с}$; $d_2 = 500 \text{ мкм}$; $g = 9,81 \text{ м/с}^2$.

Программы расчета плоского потенциального течения во всасывающем факеле пылешламоприемника диаметром 68 мм, написанные на языке Borland Pascal

1. Программа-модуль координат граничных отрезков, разбивающих границу области течения

```
{ $n+ }
unit korsk68;
interface type dim=array[1..800,'a'..'b',1..2] of real; var
kor:dim;ge:integer;
```

implementation type vek=array[1..2] of real;
 var a,b,c:vek; i,j,n,p,kseg,kotr:integer;
 al,rvnech,up,rvnut,u1,u2,rch,r,m,dl,l,h,ll,cc,d:extended;
 {2n - количество граничных элементов, dl - длина наименьшего
 отрезка;
 r - длина разбиваемого отрезка; a - массив координаты начала
 отрезка (a[1],a[2]);
 b - массив координат конца отрезка (b[1],b[2]); m - расстояние от
 начала отрезка а до рассматриваемой точки; kor - массив координат
 граничных элементов, например kor[i,'a',1] - абсцисса вершины “а” i -
 отрезка, kor[i,'b',2] - ордината вершины “b” i - отрезка; c - координаты
 середины отрезка; l - отношение длины отрезка от вершины “а” до
 рассматриваемой точки к длине отрезка от рассматриваемой точки до
 вершины “b” . }

{ Процедура неравномерного разбиения отрезка на граничные
 элементы.

p - номер граничного элемента, начиная с которого происходит
 разбиение,

a - координаты начала разбиваемого отрезка, b - координаты конца
 разбиваемого отрезка, n - количество граничных элементов в
 разбиваемом отрезке }

```

procedure razbotrez(p:integer; a,b:vek; var n:integer; var kor:dim);
var i,j:integer; var m,r,l,dl,g:extended; c:array[1..2] of extended;
begin
  r:=sqrt(sqrt(a[1]-b[1])+sqrt(a[2]-b[2])); dl:=0.001; c[1]:=(a[1]+b[1])/2;
c[2]:=(a[2]+b[2])/2;
  kor[p,'a',1]:=a[1]; kor[p,'a',2]:=a[2]; m:=dl*r; l:=m/(r-m);
  kor[p,'b',1]:=(kor[p,'a',1]+1*b[1])/(1+l);
kor[p,'b',2]:=(kor[p,'a',2]+b[2]*l)/(1+l);
  j:=1+p; m:=m*2; l:=m/(r-m); g:=0.05;
  { в начале разбиение отрезка производится неравномерно, длина
  первого dl*r, затем в 2 раза больше до тех пор пока не достигнет
  величины g*r, разбиение отрезка будет производится равномерно с
  длиной g*r, до тех пор пока не достигнет середины отрезка, затем
  разбиение отрезка просходит симметрично относительно середины }
  while m<=g*r do begin kor[j,'a']:=kor[j-1,'b'];
kor[j,'b',1]:=(a[1]+1*b[1])/(1+l);

```

```

kor[j,'b',2]:=(a[2]+l*b[2])/(1+l); m:=m*2;
l:=m/(r-m); j:=j+1 end;
  m:=g*r; l:=m/(r-m);
  while m<=0.5*r do begin kor[j,'a']:=kor[j-1,'b'];
kor[j,'b',1]:=(a[1]+l*b[1])/(1+l);
  kor[j,'b',2]:=(a[2]+l*b[2])/(1+l); m:=m+g*r; l:=m/(r-m); j:=j+1; end;
  if (abs(c[1]-kor[j-1,'b',1])<0.00001) and (abs(c[2]-kor[j-
1,'b',2])<0.00001) then j:=j-1;
  kor[j,'a']:=kor[j-1,'b']; kor[j,'b',1]:=c[1]; kor[j,'b',2]:=c[2];
  n:=j; i:=p;
  for j:=2*n-p+1 downto n+1 do begin kor[j,'a',1]:=(2*c[1]-kor[i,'b',1]);
  kor[j,'a',2]:=(2*c[2]-kor[i,'b',2]); kor[j,'b',1]:=(2*c[1]-kor[i,'a',1]);
  kor[j,'b',2]:=(2*c[2]-kor[i,'a',2]); i:=i+1; end; n:=2*n-p+1;
end;
{Процедура разбиения сегмента на граничные элементы}
procedure segment(p,n:integer; r,u1,u2:real; var kor:dim);
  {p - номер элемента с которого начинается разбиение; n - количество
элементов, на который разбивается сегмент; r - радиус сегмента; u1 -
угол который соответствует началу первого граничного элемента; u2 -
угол который соответствует концу последнего граничного элемента;}
  var i:integer; u,du:real;
  begin
    u:=u1; du:=(u2-u1)/n; i:=p;
    while i<=p+n-1 do begin kor[i,'a',1]:=r*cos(u); kor[i,'a',2]:=r*sin(u);
kor[i,'b',1]:=r*cos(u+du);
  kor[i,'b',2]:=r*sin(u+du); i:=i+1; u:=u+du end; end;

{Процедура равномерного разбиения отрезка на граничные
элементы}
procedure otrezok(p,kotr:integer;a,b:vek;var kor:dim);
  {p - начальный номер граничного элемента; kotr - количество
граничных элементов}
  var i:integer; dx:extended; kg:array[1..100,'a'..'b',1..2]of extended;
  begin
    for i:=1 to kotr do begin kg[i,'a',1]:=(a[1]+(i-1)/(kotr-(i-
1))*b[1])/(1+(i-1)/(kotr-(i-1)));
  kg[i,'a',2]:=(a[2]+(i-1)/(kotr-(i-1))*b[2])/(1+(i-1)/(kotr-(i-1))); if i=kotr
then begin kg[i,'b',1]:=b[1]; kg[i,'b',2]:=b[2] end else begin
kg[i,'b',1]:=(a[1]+i/(kotr-i)*b[1])/(1+i/(kotr-i)); kg[i,'b',2]:=(a[2]+i/(kotr-
i)*b[2])/(1+i/(kotr-i)); end; end;

```

```

j:=1;
for i:=p to p+kotr-1 do begin kor[i,'a',1]:=kg[j,'a',1];
kor[i,'a',2]:=kg[j,'a',2]; kor[i,'b',1]:=kg[j,'b',1]; kor[i,'b',2]:=kg[j,'b',2]; j:=j+1
end;
  {Процедура разбиения луча}
  procedure razbeskon(p:integer; a,b:vek; var n:integer; var kor:dim);
  {a,b - координаты начала и конца реперного отрезка}
  var i,j:integer; var m,r,l,dl,g:extended; c:array[1..2] of extended;
  begin
  r:=sqrt(sqrt(a[1]-b[1])+sqrt(a[2]-b[2])); {длина отрезка ab}
    dl:=0.001; c[1]:=(a[1]+b[1])/2; c[2]:=(a[2]+b[2])/2;
  kor[p,'a',1]:=a[1]; kor[p,'a',2]:=a[2]; m:=dl*r; l:=m/(r-m);
  kor[p,'b',1]:=(kor[p,'a',1]+1*b[1])/(1+l);
  kor[p,'b',2]:=(kor[p,'a',2]+b[2]*l)/(1+l);
  j:=1+p; m:=m*2; l:=m/(r-m);      g:=0.1;
  {В начале разбиение отрезка производится неравномерно, длина
  первого dl·r , затем в 2 раза больше до тех пор пока не достигнет
  величины g·r, разбиение будет производится равномерно с длиной g·r,
  до тех пор пока не достигнет середины отрезка, затем отрезки будут
  увеличиваться по длине пока не достигнут величины 100· r}
  while m<=g*r do begin kor[j,'a']:=kor[j-1,'b'];
  kor[j,'b',1]:=(a[1]+1*b[1])/(1+l);
  kor[j,'b',2]:=(a[2]+1*b[2])/(1+l); m:=m*2; l:=m/(r-m); j:=j+1 end;
  m:=g*r; l:=m/(r-m);
  while m<=0.5*r do begin kor[j,'a']:=kor[j-1,'b'];
  kor[j,'b',1]:=(a[1]+1*b[1])/(1+l);
  kor[j,'b',2]:=(a[2]+1*b[2])/(1+l); m:=m+g*r; l:=m/(r-m); j:=j+1; end;
  a:=b; b[1]:=b[1]+2*r; r:=sqrt(sqrt(a[1]-b[1])+sqrt(a[2]-b[2])); m:=r;
  while m<=100*r do begin kor[j,'a']:=kor[j-1,'b'];
  kor[j,'b',1]:=(a[1]+1*b[1])/(1+l);
  kor[j,'b',2]:=(a[2]+1*b[2])/(1+l); m:=m*2; l:=m/(r-m); j:=j+1; end;
n:=j-1;
end;
  {НАЧАЛО ОСНОВНОГО БЛОКА ПРОГРАММЫ}
begin p:=1;{Номер первого элемента} gch:=0.015; {радиус штанги}
rvnech:=0.038; {внешний радиус пылешламоприемника}
ksege:=20; {количество граничных элементов, на который разбивается
сегмент}

```

```

segment(p,kseg,rch,-pi,0,kor); {разбиение штанги на 20 граничных
элемента}
rvnut:=0.034; {внутренний радиус пылешламоприемника}
l:=0.02; {диаметр патрубка, подводимого к пылешламоприемнику}
u1:=arctan(l/sqrt(sqr(rvnut)-sqr(l))); a[1]:=rvnut*cos(u1)+2*1; b[1]:=a[1]-
2*1;
a[2]:=-1; b[2]:=-1; razbotrez(kseg+1,a,b,n,kor); {разбиение на элементы
от kseg+1 до n}
{разбиение внутреннего сегмента пылешламоприемника}
u2:=-pi+arctan(rch/sqrt(sqr(rvnut)-sqr(rch))); u1:=-
arctan(l/sqrt(sqr(rvnut)-sqr(l)));
p:=n+1; segment(p,3*kseg,rvnut,u1,u2,kor);
{разбиение стенки по толщине}
p:=p+3*kseg; kotr:=10; a[1]:=-rvnut*cos(pi-u2); a[2]:=-rch; b[1]:=-
sqrt(sqr(rvnech)-sqr(rch)); b[2]:=a[2]; otrezok(p,kotr,a,b,kor);
{разбиение внешнего сегмента пылешламоприемника}
p:=p+kotr; a[1]:=sqrt(sqr(rvnut)-sqr(l)); u1:=-
pi+arctan(rch/sqrt(sqr(rvnech)-sqr(rch)));
u2:=-arctan(sqrt(sqr(rvnech)-sqr(a[1]))/a[1]);
segment(p,3*kseg,rvnech,u1,u2,kor);
{разбиение отрезка внешней пылеотводящей трубки}
p:=p+3*kseg; b[1]:=a[1]+2*1; a[2]:=kor[p-1,'b',2]; b[2]:=a[2];
razbotrez(p,a,b,n,kor); p:=n+1;
{разбиение патрубка, уходящего в бесконечность}
a:=b; b[1]:=a[1]+2*1; b[2]:=a[2]; razbeskon(p,a,b,n,kor); p:=n+1;
for j:=p to 2*n do begin kor[j,'a',1]:=kor[j-(p-1),'b',1]; kor[j,'b',1]:=kor[j-
(p-1),'a',1];
kor[j,'b',2]:=-kor[j-(p-1),'a',2]; kor[j,'a',2]:=-kor[j-(p-1),'b',2] end;
{разбиение всасывающего отверстия}
p:=2*n+1; u1:=arctan(l/sqrt(sqr(rvnut)-sqr(l))); b[1]:=rvnut*cos(u1)+2*1;
a[1]:=b[1]; b[2]:=-1; a[2]:=1; razbotrez(p,a,b,n,kor); ge:=n; {общее
количество граничных элементов}
end.

```

2. Программа расчета интенсивностей источников (стоков), распределенных по границе облатси течения

```

{$n+}
program fi_skvagina;
uses crt,korsk68; label m1,m2; const ge=498;

```

```

{ге-число граничных элементов}{kol-число
прямолинейных участков области}
type vek=array[1..2]of real; matge=array[1..ge+1]of real;
  matr=array[1..ge]of ^matge; rez=array[1..ge]of real;
  var fl:file of rez; r1,r2,t1,t2,s,r,vvx,vg,vn:real;
хра,xpb,nor,xa,xb,no,ct,korct,xqa,xqb:vek;
  i,j,p,q,m,k1,n1,k,n:integer; kof:matr; koraq,korbq,x1,x2:vek; fi:rez;
  procedure baxis(xa,xb:vek;var x1,x2:vek); {Процедура вычисления
координат ортов ЛПДСК, построенной на отрезке с началом xa и
концом xb}
  begin
    x2[1]:=(xb[1]-xa[1])/sqrt(sqrt(xb[1]-xa[1])+sqrt(xb[2]-
xa[2]));
    x2[2]:=(xb[2]-xa[2])/sqrt(sqrt(xb[1]-xa[1])+sqrt(xb[2]-xa[2]));
    x1[1]:=x2[2]/(sqrt(x2[2])+sqrt(x2[1])); x1[2]:=-
x2[1]/(sqrt(x2[2])+sqrt(x2[1])); end;
  procedure preobrazvek(xm,xa,xb:vek;var ksi:vek); {Процедура
пересчета координат вектора xm, заданного в ГПДСК, в координаты ksi
в ЛПДСК}
  var fi:real; x1,x2:vek;
  begin
    baxis(xa,xb,x1,x2); ksi[1]:=x1[1]*xm[1]+x1[2]*xm[2];
ksi[2]:=x2[1]*xm[1]+x2[2]*xm[2] end;
  procedure perenos(xo,xm:vek;var ksi:vek);
  {xo - координаты точки, в которую параллельно переносится
система координат, xm - координаты заданной точки, ksi - координаты
точки в перенесенной системе координат}
  begin ksi[1]:=xm[1]-xo[1]; ksi[2]:=xm[2]-xo[2] end;
  {НАЧАЛО ОСНОВНОГО БЛОКА ПРОГРАММЫ}
  begin for j:=1 to ge do new(kof[j]);
  {Формирование элементов матриц  $F^{pq} = kof[p]^q$ }
  for p:=1 to ge do begin
    xpa[1]:=kor[p,'a',1];xpa[2]:=kor[p,'a',2];
xpb[1]:=kor[p,'b',1];xpb[2]:=kor[p,'b',2];
    baxis(xpa,xpb,x1,x2); nor[1]:=-x1[1];nor[2]:=-x1[2];
    for q:=1 to ge do begin if p=q then begin kof[p]^q[:]=-0.5;goto m1 end;
    xa[1]:=kor[q,'a',1];xa[2]:=kor[q,'a',2];xqa:=xa;
xpb[1]:=kor[q,'b',1];xpb[2]:=kor[q,'b',2];xqb:=xpb;
    preobrazvek(nor,xa,xb,no); ct[1]:=(xpb[1]-
xpa[1])/2+xpa[1];ct[2]:=(xpb[2]-xpa[2])/2+xpa[2];

```

```

preobrazvek(ct,xa,xb,korct);
preobrazvek(xqa,xa,xb,koraq);perenos(korct,koraq,koraq);
preobrazvek(xqb,xa,xb,korbq);perenos(korct,korbq,korbq);
r1:=sqrt(sqr(koraq[2])+sqr(koraq[1]));
r2:=sqrt(sqr(korbq[2])+sqr(korbq[1]));
if abs(koraq[1])<0.000001 then t1:=pi/2 else
t1:=arctan(koraq[2]/koraq[1]);
if abs(korbq[1])<0.000001 then t2:=pi/2 else
t2:=arctan(korbq[2]/korbq[1]);
if abs(no[1])<0.000001 then no[1]:=0; if abs(no[2])<0.000001 then
no[2]:=0;
kof[p]^q:=-1/2/pi*(no[1]*(t2-t1)-no[2]*ln(r1/r2));
m1: end; end;
{граничные условия для проекции скорости на внешнюю нормаль}
for j:=1 to 466 do kof[j]^ge+1:=0; for j:=467 to ge do kof[j]^ge+1:=1;
{Блок решения системы уравнений методом Гаусса}
m:=ge; n1:=m+1;
for k:=1 to m do begin k1:=k+1;s:=kof[k]^k;j:=k;
for i:=k1 to m do begin r:=kof[i]^k; if abs(r)>abs(s) then begin
s:=r;j:=i end end;
if s=0.0 then goto m2;
if j<>k then for i:=k to n1 do begin
r:=kof[k]^i;kof[k]^i:=kof[j]^i;kof[j]^i:=r end;
for j:=k1 to n1 do kof[k]^j:=kof[k]^j/s;
for i:=k1 to m do begin r:=kof[i]^k; for j:=k1 to n1 do
kof[i]^j:=kof[i]^j-kof[k]^j*r
end end;
if s<>0.0 then for i:=m downto 1 do begin s:=kof[i]^n1; for j:=i+1 to
m do s:=s-kof[i]^j*fi[j];
fi[i]:=s end;
m2: if s<>0.0 then begin for i:=1 to m do writeln('fi['i,']=',fi[i]) end else
writeln('det=0');
{Массив fi - интенсивности источников (стоков), распределенных по
границе области}
assign(fl,'fiskv68.dat'); rewrite(fl); write(fl,fi); close(fl);
end.

```

3. Программа расчета линий тока

```

{$n+}
program raschet_linii_toka;

```

```

uses crt,korsk68; const ge=498; label m1,m2;
type vek=array[1..2]of real; rez=array[1..ge]of real;
   pole=array[1..2,1..70]of real; lin=array[1..300]of real;
var fl:file of rez; flx,fly:file of lin; tgf,svch:lin; ltx,lty:lin;
skor,h,skor1,dl,dx,r1,r2,t1,t2,s,r:extended;
ct,xpa,xpb,nor,xa,xb,no,korct,xqa,xqb:vek;
i,j,p,l,q,m,k1,n1,k:integer; koraq,korbq,x1,x2:vek; cth:pole; fi,rech:rez;
procedure baxis(xa,xb:vek;var x1,x2:vek);
begin x2[1]:=(xb[1]-xa[1])/sqrt(sqr(xb[1]-xa[1])+sqr(xb[2]-xa[2]));
x2[2]:=(xb[2]-xa[2])/sqrt(sqr(xb[1]-xa[1])+sqr(xb[2]-xa[2]));
x1[1]:=x2[2]/(sqr(x2[2])+sqr(x2[1])); x1[2]:=-
x2[1]/(sqr(x2[2])+sqr(x2[1])); end;
procedure preobrazvek(xm,xa,xb:vek;var ksi:vek);
var fi:real; x1,x2:vek;
begin
baxis(xa,xb,x1,x2); ksi[1]:=x1[1]*xm[1]+x1[2]*xm[2];
ksi[2]:=x2[1]*xm[1]+x2[2]*xm[2] end;
procedure perenos(xo,xm:vek;var ksi:vek); begin ksi[1]:=xm[1]-xo[1];
ksi[2]:=xm[2]-xo[2] end;
      {НАЧАЛО ОСНОВНОГО БЛОКА ПРОГРАММЫ}
begin h:=1; assign(fl,'fiskv68.dat'); reset(fl); read(fl,fi); close(fl);
writeln(' Координаты линий тока '); writeln(' '); writeln(' x
'; y ');
ltx[1]:=-0.06; lty[1]:=0.001; dl:=0.001; l:=1;
while (ltx[1]<0.06) do begin
nor[1]:=1;nor[2]:=0; i:=0; ct[1]:=-ltx[1]; ct[2]:=-lty[1];
for p:=1 to 2 do begin
for q:=1 to ge do begin
xa[1]:=kor[q,'a',1];xa[2]:=kor[q,'a',2];xqa:=xa;
xb[1]:=kor[q,'b',1];xb[2]:=kor[q,'b',2];xqb:=xb;
preobrazvek(nor,xa,xb,no); preobrazvek(ct,xa,xb,korct);
preobrazvek(xqa,xa,xb,koraq);
perenos(korct,koraq,koraq);
preobrazvek(xqb,xa,xb,korbq);perenos(korct,korbq,korbq);
r1:=sqrt(sqr(koraq[2])+sqr(koraq[1]));
r2:=sqrt(sqr(korbq[2])+sqr(korbq[1]));
if abs(koraq[1])<0.000000001 then t1:=pi/2 else
t1:=arctan(koraq[2]/koraq[1]);

```

```

if abs(korbq[1])<0.000000001 then t2:=pi/2 else
t2:=arctan(korbq[2]/korbq[1]);
  if abs(no[1])<0.000000001 then no[1]:=0; if abs(no[2])<0.000000001
then no[2]:=0;
  rech[q]:=-1/2/pi*(no[1]*(t2-t1)-no[2]*ln(r1/r2));  m1:
end;                                skor:=0;
  for q:=1 to ge do skor:=skor+fi[q]*rech[q]; if p=1 then skor1:=skor;
nor[1]:=0; nor[2]:=1;
  if p=2 then begin tgf[1]:=skor/skor1; svch[1]:=lty[1]-tgf[1]*ltx[1];
dx:=dl/sqrt(1+sqr(tgf[1]));
  if skor1<0 then ltx[1+1]:=ltx[1]-dx else ltx[1+1]:=ltx[1]+dx;
lty[1+1]:=tgf[1]*ltx[1+1]+svch[1] end;
end;
  if (1 mod 1 =0) or (l=1) then writeln(1, ' ,ltx[1]:10:4, '  ',lty[1]:10:4);
inc(1);
end;
  assign(flx,'lix68_4.dat'); rewrite(flx); assign(fly,'liy68_4.dat');
rewrite(fly);
  write(flx,ltx); write(fly,lty); close(flx); close(fly)
end.

```

4. Программа расчета траектории пылевой частицы

```

{$n+}
program raschet_traektorii_v_skvachine;
uses crt, korukbp1;
const n=6; { количество дифференциальных уравнений, пишывающих
траекторию частицы }
ge=498 { количество граничных отрезков }; label m1,m2;
type vek=array[1..2]of real; rez=array[1..ge]of real;
pole=array[1..2,1..70]of real;
lin=array[1..1800]of real; dim=array[1..n]of real; vek3=array[1..3]of
real;
var fl:file of rez; t,vvx:real; y1,y:dim; v,v1:vek3; tgf,svch:rez; ltx,lty:lin;
skor,skor1,
  D,g,C,A,B,dl,dx,r1,r2,t1,t2,s,r,vg,h,ll,
kofotskoka,cc,ro,de,vjaz,Zo,kf,Vzo,f,nu:real;
  vch,ct,xpa,xpb,nor,xa,xb,no,korct,xqa,xqb:vek;
i,j,p,l,q,m,k1,n1,k,chet:integer;

```

```

koraq,korbq,x1,x2:vek; cth:pole; fi,rech:rez;
function arccos(x:real):real;
begin if x=0 then arccos:=(pi/2) else arccos:=arctan(sqrt(1-
sqr(x))/x)+pi*byte(x<0) end;
procedure povorotvek(ksi:vek;fi:real;var x:vek);
begin x[2]:=-ksi[1]*sin(fi)+ksi[2]*cos(fi);
x[1]:=ksi[1]*cos(fi)+ksi[2]*sin(fi) end;
procedure bacis(xa,xb:vek;var x1,x2:vek);
begin x2[1]:=(xb[1]-xa[1])/sqrt(sqr(xb[1]-xa[1])+sqr(xb[2]-xa[2]));
x2[2]:=(xb[2]-xa[2])/sqrt(sqr(xb[1]-xa[1])+sqr(xb[2]-xa[2]));
x1[1]:=x2[2]/(sqr(x2[2])+sqr(x2[1])); x1[2]:=-
x2[1]/(sqr(x2[2])+sqr(x2[1])); end;
procedure preobrazvek(xm,xa,xb:vek;var ksi:vek);
var fi:real; x1,x2:vek;
begin bacis(xa,xb,x1,x2); ksi[1]:=x1[1]*xm[1]+x1[2]*xm[2];
ksi[2]:=x2[1]*xm[1]+x2[2]*xm[2] end;
procedure perenos(xo,xm:vek;var ksi:vek);
begin ksi[1]:=xm[1]-xo[1]; ksi[2]:=xm[2]-xo[2] end;

procedure sodu(v:vek3; var x:real; var y:dim);
{ Процедура решения системы дифференциальных уравнений }
const n=6; type dim1=array[1..n]of real;
var w,k1,k2,k3,k4:dim1; con,h:real; j:integer;

function fun(v:vek3;j:integer;x:extended;y:dim):real;
const
kf=1;{коэффициент формы частицы} de=500/1000000;{диаметр
пылинки}
mu=1.809/100000;{коэффициент динамической вязкости} g=9.81;
ro=3500; {плотность частицы пыли} Zo=0.08; {высота
пылешламоприемника}
rov=1.205;{плотность воздуха} ksio=1.8;{коэффициент  $\Psi_o$ }
var A,Re,ksi,Sm,con:real;{Re-число Рейнольдса, Sm-площадь
миделевого сечения}
begin
Sm:=pi*de*de/4; A:=pi*de*de*de/6*ro;
Re:=rov*de/mu*sqrt(sqr(y[3]-v[1])+sqr(y[4]-v[2])+sqr(y[6]-v[3]));
ksi:=24/Re+ksio;
V[3]:=sqrt(g*2*A/ksi/Sm/rov)-7;

```

```

con:=ksi*Sm*rov/2/A*sqrt(sqrt(y[3]-v[1])+sqrt(y[4]-v[2])+sqrt(y[6]-v[3]))
;
if j=1 then fun:=y[3]; if j=2 then fun:=y[4]; if j=5 then fun:=y[6];
if j=6 then fun:=-g-con*(y[6]-v[3]); if j=3 then fun:=-con*(y[3]-v[1]);
if j=4 then fun:=-con*(y[4]-v[2]);
end;
begin
h:=0.00005; { шаг по времени }
for j:=1 to n do w[j]:=y[j]; for j:=1 to n do k1[j]:=h*fun(v,j,x,y);
for j:=1 to n do y[j]:=w[j]+1/2*k1[j]; for j:=1 to n do
k2[j]:=h*fun(v,j,x+1/2*h,y);
for j:=1 to n do y[j]:=w[j]+1/2*k2[j]; for j:=1 to n do
k3[j]:=h*fun(v,j,x+1/2*h,y);
for j:=1 to n do y[j]:=w[j]+k3[j]; for j:=1 to n do
k4[j]:=h*fun(v,j,x+h,y);
for j:=1 to n do y[j]:=w[j]+(k1[j]+2*k2[j]+2*k3[j]+k4[j])/6; x:=x+h;
end;
begin {НАЧАЛО ОСНОВНОГО БЛОКА
ПРОГРАММЫ}
clrscr; assign(fl,'fiskv68.dat'); reset(fl); read(fl,fi); close(fl);
writeln(' t x y z Vx Vy Vz'); chet:=1;
y[1]:=-0.034; y[2]:=0.014; y[3]:=0; y[4]:=0; y[5]:=0.08; y[6]:=-7; t:=0;
{y[1],y[2],y[5]- абсцисса, ордината, аппликата точки начала
движения; y[3],y[4],y[6]-x,y,z составляющие скорости пылевой частицы;
v[1],v[2],v[3]-x,y,z составляющие скорости воздуха }
i:=0;
while (y[5]>0)and(y[1]<=0.034) do
begin
inc(i); kofotskoka:=0.5; {коэффициент восстановления при ударе
частицы о стенку}
f:=0.5; { коэффициент трения скольжения }
if (( (sqrt(y[1])+sqrt(y[2]))>sqrt(0.034-0.0001))and
(y[1]>kor[112,'b',1])and
(y[1]<=kor[53,'a',1])or((sqrt(y[1])+sqrt(y[2])<sqrt(0.015+0.0001))and(y[1]
>0.016)
and(y[1]<0.016))
then
begin y:=y1; vch[1]:=y[3]; vch[2]:=y[4];
writeln(sqrt(sqrt(vch[1])+sqrt(vch[2])));

```

```

povorotvek(vch,arctan(-y[1]/y[2]),vch);
if k=1 then vch[2]:=-kofotskoka*vch[2] else
begin if -2*vch[1]/(7*f*(1+kofotskoka)*vch[2])<1 then
nu:=-2*vch[1]/(7*f*(1+kofotskoka)*vch[2]) else nu:=1;
vch[1]:=vch[1]+nu*f*(1+kofotskoka)*vch[2]; vch[2]:=-
kofotskoka*vch[2]; end;
povorotvek(vch,-arctan(-y[1]/y[2]),vch); y[3]:=vch[1]; y[4]:=vch[2];
writeln(sqrt(sqr(vch[1])+sqr(vch[2])));
end;
if (y[1]>kor[52,'b',1])and(abs(y[2])>-kor[52,'a',2])
then begin y:=y1;
if -2*y[3]/(7*f*(1+kofotskoka)*y[4])<1 then nu:=-
2*y[3]/(7*f*(1+kofotskoka)*y[4]) else nu:=1;
y[3]:=y[3]+nu*f*(1+kofotskoka)*y[4]; y[4]:=-kofotskoka*y[4];
y[3]:=y[3]; y[4]:=-kofotskoka*y[4]
end;
nor[1]:=1;nor[2]:=0; ct[1]:=y[1]; ct[2]:=y[2];
for p:=1 to 2 do
begin
for q:=1 to 2 do
begin
xa[1]:=kor[q,'a',1];xa[2]:=kor[q,'a',2];xqa:=xa;xb[1]:=kor[q,'b',1];xb[2]:=kor[
q,'b',2];xqb:=xb;
preobrazvek(nor,xa,xb,no);
preobrazvek(ct,xa,xb,korct);preobrazvek(xqa,xa,xb,koraq);

perenos(korct,koraq,koraq);preobrazvek(xqb,xa,xb,korbq);perenos(korct,korb
q,korbq);
r1:=sqrt(sqr(koraq[2])+sqr(koraq[1]));
r2:=sqrt(sqr(korbq[2])+sqr(korbq[1]));
if abs(koraq[1])<0.000000001 then t1:=pi/2 else
t1:=arctan(koraq[2]/koraq[1]);
if abs(korbq[1])<0.000000001 then t2:=pi/2 else
t2:=arctan(korbq[2]/korbq[1]);
if abs(no[1])<0.000000001 then no[1]:=0; if abs(no[2])<0.000000001
then no[2]:=0;
rech[q]:=-1/2/pi*(no[1]*(t2-t1)-no[2]*ln(r1/r2));
end;

```

```

skor:=0; for q:=1 to ge do skor:=skor+fi[q]*rech[q];
skor:=37*skor; if p=1 then skor1:=skor;
nor[1]:=0; nor[2]:=1;
end;
v[1]:=skor1;
v[2]:=skor;
y1:=y;
if (chet=1) or (chet mod 1 =0) then
begin
writeln(t:8:4,',y[1]:8:4,',y[2]:8:4,',y[5]:8:4,',y[3]:8:4,',y[4]:8:4,'
',y[6]:8:4,',v[1]:8:4,',v[2]:8:4);
end;
ltx[i]:=y[1];
lty[i]:=y[2];
chet:=chet+1;
sodu(v,t,y);
end;
end.

```

На рис. 8.1. представлены линии тока (верхняя часть рисунка) в пылешламоприемниках и траектории частиц крупностью 0,5мм при различных точках ее возможного вылета. Скорость отсоса воздуха v_0 подбиралась таким образом, чтобы частица “успела” за 0,08м своего вертикального падения попасть в отводящий патрубок . Наиболее неблагоприятная точка для захвата пыли - точка лежащая на оси симметрии воздушного потока. Такая пылинка следуя расчетным данным не может быть уловлена. Однако в реальных условиях применения пылешламоприемника такой случай маловероятен. Поэтому от оси отсоса сделано отступление 0,0025м. Таким образом для локализации пылевыведений необходимо создать скорость в отсосе для пылешламоприемника внутренним диаметром 68мм - 80м/с; для пылешламоприемника внутренним диаметром 40мм - 27 м/с.

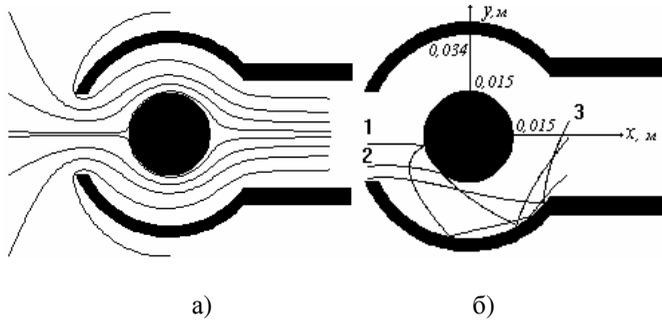


Рис.8.5. Линии тока и траектории пылевой частицы диаметром 0,5мм при различных положениях точек ее вылета (1 - 0,0025м от оси симметрии отсоса; 2- 0,01м; 3 - 0,014м) в пылешламоприемниках диаметром 68мм.

9. Лабораторные работы

Лабораторная работа №1.

Численное интегрирование методами трапеций, Симпсона и Гаусса

Цель работы: используя различные методы численного интегрирования (трапеций, Симпсона и Гаусса) вычислить заданный определенный интеграл.

Содержание отчёта

1. Условие лабораторной работы;
2. Постановка задачи;
3. Аналитическое вычисление данного интеграла;
4. Алгоритм численного решения задачи в виде программы для ЭВМ;
5. Результаты решения задачи в виде таблицы или графика;
6. Выводы о точности используемых методов численного интегрирования.

Постановка задачи

1. Вычислить заданный интеграл аналитически;
2. Составить и отладить программу вычисления заданного интеграла методом трапеции (колонка 2 табл. 9.1)
3. Составить и отладить программу вычисления заданного интеграла методом Симпсона (колонка 3 табл. 9.1)
4. Составить и отладить программу вычисления заданного интеграла методом Гаусса (колонка 4 табл. 9.1)
5. Составить и отладить программу вычисления интеграла, где в виде процедур оформлены программы численного интегрирования методами трапеций, Симпсона и Гаусса (колонка 5 табл.9.1). В программе предусмотреть вывод на печать результатов счёта в виде следующей таблицы.
- 6.

Количество отрезков разбиения промежутков интегрирования n	Результаты вычисления по методу			
	Трапеции	Симпсона	Гаусса	По аналитической формуле

Для метода Гаусса взять фиксированное число квадратурных узлов, равных 8.

7. Определить погрешности вычисления различными методами.

Варианты заданий

Таблица 9.1

№	Метод трапеции	Метод Симпсона	Метод Гаусса	Общая программа
1	2	3	4	5
1	$\int_0^1 \frac{4\arctg x - x}{1+x^2} dx$	$\int_0^1 x^2 e^{3x} dx$	$\int_1^3 \sqrt{x+1} dx$	$\int_0^{\pi/2} \frac{\cos x dx}{1+\sin x - \cos x}$
2	$\int_0^{1/2} \frac{8x - \arctg 2x}{1+4x^2} dx$	$\int_1^e \sqrt{x} \ln^2 x dx$	$\int_{1/2}^1 \frac{dx}{\sqrt{1-x^4}}$	$\int_0^{\pi/2} \frac{\sin x dx}{1+\cos x + \sin x}$
3	$\int_0^1 \frac{x}{1+x^4} dx$	$\int_1^{e^2} \frac{\ln^2 x}{\sqrt{x}} dx$	$\int_{3/4}^{4/3} \frac{dx}{\sqrt{x^2+1}}$	$\int_0^2 \frac{\cos x dx}{1+\cos x + \sin x}$

4	$\int_{\sqrt{3}}^{\sqrt{8}} \frac{x-1/x}{\sqrt{x^2+1}} dx$	$\int_{-1}^1 x^2 e^{-x/2} dx$	$\int_0^4 \frac{dx}{1+\sqrt{x}}$	$\int_0^{\pi/4} \frac{dx}{\cos x(1+\cos x)}$
5	$\int_0^{\sqrt{3}} \frac{x-(\arctg x)^4}{1+x^2} dx$	$\int_{-2}^0 (x^2+2)e^{x/2} dx$	$\int_0^{\ln 2} \sqrt{e^x-1} dx$	$\int_0^{\pi/2} \frac{dx}{(1+\sin x+\cos x)^2}$
6	$\int_0^{\sin 1} \frac{(\arcsin x)^2+1}{\sqrt{1-x^2}} dx$	$\int_1^8 \frac{\ln^2 x}{\sqrt[3]{x^2}} dx$	$\int_1^2 \frac{\sqrt{x^2+1}}{x} dx$	$\int_0^{2\pi/3} \frac{\cos^2 x dx}{(1+\sin x+\cos x)^2}$
7	$\int_{\sqrt{3}}^{\sqrt{8}} \frac{dx}{x\sqrt{x^2+1}}$	$\int_1^2 x \ln^2 x dx$	$\int_{-1}^1 \frac{dx}{(1+x^2)^2}$	$\int_{-2\pi/3}^0 \frac{\cos^2 x dx}{(1+\cos x-\sin x)^2}$
8	$\int_{\sqrt{2}}^2 \frac{dx}{x\sqrt{x^2-1}}$	$\int_0^{\pi/2} (1-5x^2)\sin x dx$	$\int_0^1 \operatorname{ctg}(2x+1) dx$	$\int_0^{\pi/2} \frac{\sin x dx}{(1+\sin x+\cos x)^2}$
9	$\int_0^1 \frac{x}{\sqrt{x^4+x^2+1}} dx$	$\int_0^{\pi/2} x \cos x dx$	$\int_{1/2}^1 \frac{dx}{\sqrt{x}}$	$\int_{-\pi/2}^0 \frac{\cos x dx}{(1+\cos x-\sin x)^2}$

Продолжение табл.9.1

1	2	3	4	5
10	$\int_0^1 \frac{(x^2+1)}{(x^3+3x+1)^2} dx$	$\int_0^1 \frac{\arcsin x}{\sqrt{1+x}} dx$	$\int_0^1 \frac{dx}{1+x}$	$\int_0^{\pi/2} \frac{\sin x dx}{5+3\sin x}$
11	$\int_0^2 \frac{x^3}{4+x^2} dx$	$\int_0^{\pi} e^x \sin 2x dx$	$\int_{-2}^{-1} \frac{dx}{x^3}$	$\int_0^{\pi/2} \frac{\sin x dx}{2+\sin x}$
12	$\int_0^{\pi/4} \frac{2\cos x+3\sin x}{(2\sin x-3\cos x)^3} dx$	$\int_0^{\pi} x^3 \sin x dx$	$\int_{-1}^{-1} \frac{dx}{8+x^2}$	$\int_0^{\pi/2} \frac{\sin^2 x dx}{(1+\cos x+\sin x)^2}$
13	$\int_1^4 \frac{1/(2\sqrt{x})+1}{(\sqrt{x}+x)^2} dx$	$\int_1^2 \frac{\ln x}{x^5} dx$	$\int_0^1 \sqrt{x^2+4} dx$	$\int_{-\pi/2}^0 \frac{\sin x dx}{(1+\cos x-\sin x)^2}$
14	$\int_{\sqrt{3}}^{\sqrt{8}} \frac{x+1/x}{\sqrt{x^2+1}} dx$	$\int_0^1 \arcsin x dx$	$\int_0^{\pi/4} \operatorname{tg} x dx$	$\int_0^{\pi/2} \frac{\sin x dx}{(1+\sin x)^2}$

15	$\int_0^{\sqrt{3}} \frac{\arctg x + x}{1+x^2} dx$	$\int_1^2 (3x+2) \ln x dx$	$\int_{\pi/2}^{\pi} 2^8 \sin^8 x dx$	$\int_0^{\pi/2} \frac{\cos x dx}{(1+\cos x + \sin x)^2}$
16	$\int_0^1 \frac{x^3}{1+x^2} dx$	$\int_0^1 x \ln(1+x^2) dx$	$\int_0^{2\pi} \sin^4 x \cos^4 x dx$	$\int_{-2\pi/3}^0 \frac{\cos x dx}{1+\cos x - \sin x}$
17	$\int_1^3 \frac{1-\sqrt{x}}{\sqrt{x(x+1)}} dx$	$\int_{\pi/4}^{\pi/3} \frac{xdx}{\sin^2 x}$	$\int_0^{2\pi} \cos^8 x dx$	$\int_0^{3/2} \frac{x^2 dx}{\sqrt{9-x^2}}$
18	$\int_0^1 \frac{x^3}{(1+x^2)^2} dx$	$\int_0^e (1-\ln x)^2 dx$	$\int_0^{2\pi} \cos^8(x/4) dx$	$\int_0^1 \frac{x^2 dx}{\sqrt{4-x^2}}$
19	$\int_0^{\pi/4} \operatorname{tg} x \ln(\cos x) dx$	$\int_0^1 (\arcsin x)^2 dx$	$\int_{\pi/2}^0 2^8 \cos^8 x dx$	$\int_0^{\sqrt{2}} \frac{x^4 dx}{(4-x^2)^{3/2}}$
20	$\int_0^1 \frac{(x^3+x) dx}{x^4+1}$	$\int_0^{\pi/4} \ln(1+\operatorname{tg} x) dx$	$\int_0^{\pi} 2^4 \sin^8 x dx$	$\int_0^2 \frac{dx}{(4+x^2)\sqrt{4+x^2}}$
21	$\int_2^9 \frac{xdx}{\sqrt[3]{x-1}}$	$\int_0^1 x^2 \arctg x dx$	$\int_0^{16} \sqrt{256-x^2} dx$	$\int_0^1 \frac{x^4 dx}{(2-x^2)^{3/2}}$
22	$\int_{-1}^0 \frac{\operatorname{tg}(x+1)}{\cos^2(1+x)} dx$	$\int_0^{2\pi} 7x^2 \cos 2x dx$	$\int_0^1 \sqrt{4-x^2} dx$	$\int_0^{4\sqrt{3}} \frac{dx}{\sqrt{(64-x^2)^3}}$
23	$\int_{\sqrt{2}}^{\sqrt{3}} \frac{x}{\sqrt{x^4-x^3-1}} dx$	$\int_0^{2\pi} (2x^2-15) \cos 3x dx$	$\int_0^1 x^2 \sqrt{1-x^2} dx$	$\int_0^2 \frac{x^4 dx}{\sqrt{(8-x^2)^3}}$
24	$\int_{\pi/4}^{\pi/2} \frac{x \cos x + \sin x}{(x \sin x)^2} dx$	$\int_{-2}^0 x^2 \cos 3x dx$	$\int_{-2}^3 x^2 \sqrt{9-x^2} dx$	$\int_{-2}^3 x^2 \sqrt{9-x^2} dx$

Лабораторная работа №2

Определение осевой скорости воздуха вблизи всасывающих отверстий, встроенных в плоскую безграничную стенку

Цель работы: Произвести расчет изменения осевой скорости при удалении от всасывающих отверстий различной геометрической формы.

Подобрать такую форму отверстия, при которой осевая скорость воздуха будет максимальной.

Содержание отчета.

1. Условие лабораторной работы.
2. Постановка задачи.
3. Программа расчета.
4. Результаты расчета в виде таблицы или графика.
5. Вывод.

Задание к лабораторной работе №2

Найти скорость в точки z (на оси отсоса): z изменяется от 0 до 10 с шагом $\Delta z = 0,25$ для:

1. эллиптического отверстия a, b – фокусы, причем $b = 1/a/b = 1+0,5i$, где i – номер варианта.
2. прямоугольного отверстия, той же площади S , что и для эллипса (одна из сторон имеет длину 1).
3. кольцевого отверстия, той же площади S , при внутреннем радиусе
4. $R_{\text{внут}} = 1$.
5. треугольного отверстия площадью S , при $n = 3$.
6. квадратного отверстия площадью S .
7. круглого отверстия площадью S .

Результаты расчета представить в виде таблицы.

z	$V_{\text{элл}}$	$V_{\text{пр}}$	$V_{\text{кв}}$	$V_{\text{кон}}$	$V_{\text{кр}}$	$V_{\text{тр}}$

Лабораторная работа №3

Решение системы обыкновенных дифференциальных уравнений в ОДУ первого рода методом Рунге – Кутта

Цель работы: решить данную систему обыкновенных дифференциальных уравнений первого порядка методом Рунге – Кутта при заданных начальных условиях.

Содержание отчета

1. Условие лабораторной работы.
2. Постановка задачи.
3. Программа расчета для ЭВМ.
4. Результаты счета.
5. Выводы.

Постановка задачи

1. Составить программу расчета.
2. Произвести расчет интегральной кривой, начиная с x_0 (начальная точка) до x_0+1 , с шагом $h = 1/n$, $n = 2,4,8,16,\dots,N$, где N – число, при котором разница между численным решением и аналитическим не превышает 0,0001.

Варианты заданий

Вар		Аналитическое решение
1	2	3
1	$\begin{cases} \frac{dy}{dx} = z - 3y, \\ \frac{dz}{dx} = z - y. \end{cases}$ <p>Начальные условия: при $x = 0, y = z = 1$</p>	$\begin{aligned} y &= (A - B + Bx) e^{-2x}; \\ z &= (-A - Bx) e^{-2x}; \quad A = -1, B = -2. \end{aligned}$

2	$\begin{cases} \frac{dx}{dt} + 7x - y = 0, \\ \frac{dy}{dt} + 2x + 5y = 0. \end{cases}$ <p>Начальные условия: при $t = 0$, $x = y = 1$</p>	$\begin{aligned} x &= (A \cos t + B \sin t) e^{-6t}; \\ y &= [(A+B) \cos t - (A-B) \sin t] e^{-6t}; \\ A &= 1, B = 0. \end{aligned}$
3	$\begin{cases} \frac{dx}{dt} = z + y - x, \\ \frac{dy}{dt} = z + x - y, \\ \frac{dz}{dt} = x + y + z. \end{cases}$ <p>Начальные условия: при $t = 0$, $x = 1$, $y = z = 0$</p>	$\begin{aligned} x &= Ae^{-t} + Be^{2t} + Ce^{-2t}; \\ y &= Ae^{-t} + Be^{2t} - Ce^{-2t}; \\ z &= -Ae^{-t} + 2Be^{2t}; \\ 3A &= 6B = 2C = 1. \end{aligned}$
4	$\begin{cases} \frac{dx}{dt} = z + y, \\ \frac{dy}{dt} = z + x, \\ \frac{dz}{dt} = x + y. \end{cases}$ <p>Начальные условия: при $t = 0$, $x = -1$, $y = 1, z = 0$</p>	$\begin{aligned} x &= -e^{-t}; \\ y &= e^{-t}; \\ z &= 0. \end{aligned}$
5	$\begin{cases} \frac{dy}{dx} = z, \\ \frac{dz}{dx} = -y. \end{cases}$ <p>начальные условия: при $x = 0$, $y = z = 1$</p>	$\begin{aligned} y &= c_1 \cos x + c_2 \sin x; \\ z &= c_2 \cos x - c_1 \sin x. \end{aligned}$

6	$\begin{cases} \frac{dy}{dx} = y + 5z, \\ \frac{dz}{dx} + y + 3z = 0. \end{cases}$ <p>Начальные условия: при $x = 0, y = 1, z = 5$</p>	$y = e^{-x} (c_1 \cos x + c_2 \sin x);$ $z = 1/5 e^{-x} [(c_2 - 2c_1) \cos x - (c_1 + 2c_2) \sin x].$
7	$\begin{cases} \frac{dy}{dx} = -3y - z, \\ \frac{dz}{dx} = y - z. \end{cases}$ <p>Начальные условия: при $x = 0, y = 1, z = 2$</p>	$y = e^{-2x} (c_1 - c_2 - c_1 x);$ $z = e^{-2x} (c_1 x + c_2).$
8	$\begin{cases} \frac{dx}{dt} = y, \\ \frac{dy}{dt} = z, \\ \frac{dz}{dt} = x. \end{cases}$ <p>Начальные условия: при $t = 0,$ $x = y = z = 1$</p>	$x = c_1 e^t + e^{-1/2} (c_2 \cos \frac{\sqrt{3}}{2} t + c_2 \sin \frac{\sqrt{3}}{2} t);$ $y = c_1 e^t + e^{-1/2} (\frac{c_3 \sqrt{3} - c_2}{2} \cos \frac{\sqrt{3}}{2} t - \frac{c_2 \sqrt{3} + c_3}{2} \sin \frac{\sqrt{3}}{2} t)$ $z = c_1 e^t + e^{-1/2} (\frac{-c_3 \sqrt{3} - c_2}{2} \cos \frac{\sqrt{3}}{2} t + \frac{c_2 \sqrt{3} - c_3}{2} \sin \frac{\sqrt{3}}{2} t)$
9	$\begin{cases} \frac{dy}{dx} = y + z, \\ \frac{dz}{dx} = y + z + x. \end{cases}$ <p>Начальные условия: при $x = 0, y = 2,$ $z = -1/4$</p>	$y = c_1 + c_2 e^{2x} - 1/4 (x^2 + x);$ $z = c_2 e^{2x} - c_1 - 1/4 (x^2 - x - 1).$

10	$\begin{cases} \frac{dy}{dx} + 2y + z = \sin x, \\ \frac{dz}{dx} - 4y - 2z = \cos x. \end{cases}$ <p>Начальные условия: при $x = 0, y = 0,$ $z = -2$</p>	$y = c_1 + c_2x + 2\sin x;$ $z = -2c_1 - c_2(2x + 1) - 3\sin x - 2\cos x.$
11	$\begin{cases} \frac{dy}{dx} + 3y + 4z = 2x, \\ \frac{dz}{dx} - y - z = x. \end{cases}$ <p>Начальные условия: при $x = y = 0, z = 0$</p>	$y = (c_2 - 2c_1 - 2c_2x) e^{-x} - 6x + 14;$ $z = (c_1 + c_2x) e^{-x} + 5x - 9.$

1	2	3
12	$\begin{cases} \frac{dx}{dt} - 4x - y + 36t = 0, \\ \frac{dy}{dt} + 2x - y + 2e^t = 0. \end{cases}$ <p>Начальные условия: при $t = 0, x = 0, y = 1$</p>	$X = 10e^{2t} - 8e^{3t} - e^t + 6t - 1;$ $y = -20e^{2t} + 8e^{3t} + 3e^t + 12t + 10.$
13	$\begin{cases} \frac{dy}{dx} = y^2/z, \\ \frac{dz}{dx} = 1/2 y. \end{cases}$ <p>Начальные условия: при $x = 0, y = 2, z = 1$</p>	$y = \frac{2c_1}{(c_2 - x)^2};$ $z = \frac{c_1}{c_2 - x}.$

14	$\begin{cases} \frac{dy}{dx} + z = 1, \\ \frac{dz}{dx} + 2/x^2 y = \ln x. \end{cases}$ <p>Начальные условия: при $x = 1, y = 2,$ $z = 1/9$</p>	$y = c_1 x + \frac{c_2}{x} - \frac{x^2}{18} (3 \ln^2 - 2 \ln x);$ $z = 1 - 2c_1 x + \frac{c_2}{x^2} + \frac{x}{9} (3 \ln^2 + \ln x - 1).$
15	$\begin{cases} \frac{dy}{dx} + 2y + 4z = 1 + 4x \\ \frac{dz}{dx} + y - z = 3/2x^2. \end{cases}$ <p>Начальные условия: при $x = 0, y = 2,$ $z = -3/4$</p>	$y = c_1 e^{2x} + c_2 e^{-3x} + x^2 + x;$ $z = -c_1 e^{2x} + \frac{c_2}{4} e^{-3x} - \frac{1}{2} x^2.$
16	$\begin{cases} \frac{dx}{dt} = y, \\ \frac{dy}{dt} = x + e^t + e^{-t}. \end{cases}$ <p>Начальные условия: при $t = 0, x = 1, y = 0$</p>	$x = 1/2 [(t + A)e^t + (-t + B)e^{-t}];$ $y = 1/2 [(t + A + 1)e^t + (t - B - 1)e^{-t}].$
17	$\begin{cases} \frac{dx}{dt} + 5x - 2y = e^t, \\ \frac{dy}{dt} - x + 6y = e^{2t}. \end{cases}$ <p>Начальные условия: при $t = 0, x = 7/40 + 1/27$ $y = 1/40 + 7/54$</p>	$x = 2Ae^{-4t} + Be^{-7t} + 7/40e^t + 1/27e^{3t};$ $y = Ae^{-4t} - Be^{-7t} + 1/40e^t + 7/54e^{2t}.$

18	$\begin{cases} \frac{dy}{dx} = 3z - y, \\ \frac{dz}{dx} = z + y + e^x. \end{cases}$ <p>Начальные условия: $x = 0, y = 2, z = 0$</p>	$\begin{aligned} y &= Ae^{2x} + Be^{-2x} - e^x; \\ z &= Ae^{2x} - 1/3Be^{-2x} + 2/3e^x. \end{aligned}$
19	$\begin{cases} \frac{dy}{dx} - y + z = 3x^2, \\ \frac{dz}{dx} + 4y + 2z = 2 + 8x. \end{cases}$ <p>Начальные условия: $x = 0, y = 1, z = -1$</p>	$\begin{aligned} y &= -x^2 + c_1e^{2x} + c_2e^{-3x}; \\ z &= 2x^2 + 2x - c_1e^{2x} + 4c_2e^{-3x}. \end{aligned}$
20	$\begin{cases} \frac{dx}{dt} + 4y = \cos 2t, \\ \frac{dy}{dt} + 4x = \sin 2t. \end{cases}$ <p>Начальные условия: при $t = 0, x = 1, y = 9/10$</p>	$\begin{aligned} x &= Ae^{4t} + Be^{-4t} + 3/10 \sin 2t; \\ y &= -Ae^{4t} + Be^{-4t} + 1/10 \cos 2t. \end{aligned}$
21	$\begin{cases} x^2 \frac{dy}{dx} + z = x^2, \\ \frac{dz}{dx} + y = x. \end{cases}$ <p>Начальные условия: При $x = 1,$ $y = 1 - \frac{1 - \sqrt{5}}{2}$</p>	$\begin{aligned} y &= x - ?_1 c_1 x^{-?_2} - ?_2 c_2 x^{-?_1}; \\ y &= c_1 x^{?_2} - c_2 x^{?_1}; \\ ?_1 &= \frac{1 - \sqrt{5}}{2}, ?_2 = \frac{1 + \sqrt{5}}{2}. \end{aligned}$

22	$\begin{cases} \frac{dy}{dx} + z = 1, \\ x^2 \frac{dz}{dx} + 2y = x^2 \ln x. \end{cases}$ <p>Начальные условия: $x = 1, y = 2, z = -2\frac{1}{9}$</p>	$y = -1/6x^2 \ln^2 x + 1/9x^2 \ln x + Ax^{-1} + Bx^3;$ $z = 1 - y.$
23	$\begin{cases} t \frac{dx}{dt} = t - 2x, \\ t \frac{dy}{dt} = t(x+y) + 2x - t. \end{cases}$ <p>Начальные условия: при $t = 0, x = 0, y = 1$</p>	$x + y = c_1 e^t;$ $3xt^2 = t^3 + c_2.$
24	$\begin{cases} \frac{dy}{dx} = z, \\ \frac{dz}{dx} = y. \end{cases}$ <p>Начальные условия: $x = \pi/2, y = 1, z = -1$</p>	$y = c_1 \cos x + c_2 \sin x;$ $z = c_2 \cos x - c_1 \sin x.$
25	$\begin{cases} \frac{dy}{dx} = -3y - z, \\ \frac{dz}{dx} = y - z. \end{cases}$ <p>Начальные условия: $x = 0, y = 0, z = 2$</p>	$y = (c_1 - c_2 - c_1 x) e^{-2x};$ $z = (c_1 x + c_2) e^{-2x}.$

**Лабораторная работа № 4. Полет частицы пыли в плоскости,
при стоксовском ее обтекании воздухом**

Цель работы: найти максимальный диаметр пылевой частицы, улавливаемой стоком воздуха. Построить траекторию частицы пыли d_{\max} и $d_{\max} - 5$ мкм.

Содержание отчета.

1. Условие лабораторной работы.

2. Постановка задачи.
3. Программа расчета.
4. График траектории частиц пыли d_{\max} и $d_{\max} - 5$.
5. Выводы.

Постановка задачи

Пусть воздух стекает к точке $M(a, b)$. Из точки A вылетает частица пыли со скоростью v_0 . Найти максимальный диаметр частицы пыли d_{\max} , улавливаемой вытяжным устройством (стоком), расположенным в точке M . Построить две траектории частиц диаметрами d_{\max} , $d_{\max} - 5$ мкм.

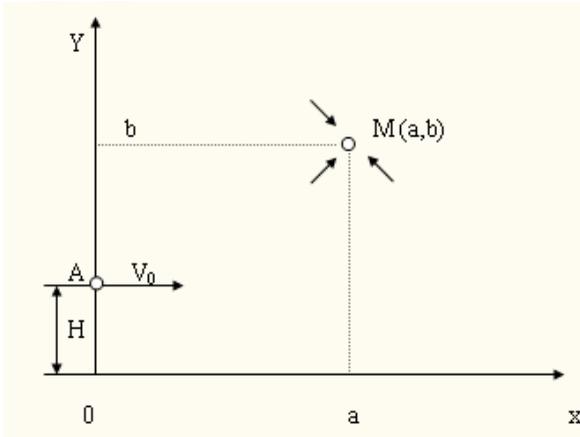


Рис.9.1

Плотность частицы $\rho = 2500 \text{ кг/м}^3$; вязкость воздуха $1,776 \cdot 10^{-5} \text{ м}^2/\text{с}$; коэффициент формы $\chi = 1$; $H = 0,1 \text{ м}$, $a = 0,5 + i \cdot 0,01 \text{ м}$, $b = 0,5 + i \cdot 0,01 \text{ м}$, где i – номер варианта; $V_0 = 1 \text{ м/с}$.

Лабораторная работа № 5

Трехмерный полет частицы пыли при стоксовском ее обтекании

Цель работы: найти максимальный диаметр частицы пыли, улавливаемой вытяжным устройством, с точностью 5 мкм.

Содержание отчета

1. Условие лабораторной работы.
2. Постановка задачи.

3. Программа расчета.
4. Таблица координат траектории частиц пыли d_{\max} и d_{\max} - 5мкм.
5. Выводы.

Постановка задачи

Пусть сток находится в точке $M(a, b, c)$, рис.11), где $a=0,5+0,01i$, $b=0,5+0,01i$, $c=0,5+0,01i$, i – номер варианта. Начальная скорость частицы $\vec{V} = \{1, 1, 0\}$. Плотность частицы $\rho_1=2500 \text{ кг/м}^3$, кинематическая вязкость воздуха $\nu=1.776 \cdot 10^{-5} \text{ м}^2/\text{с}$, $H=0.1 \text{ м}$.

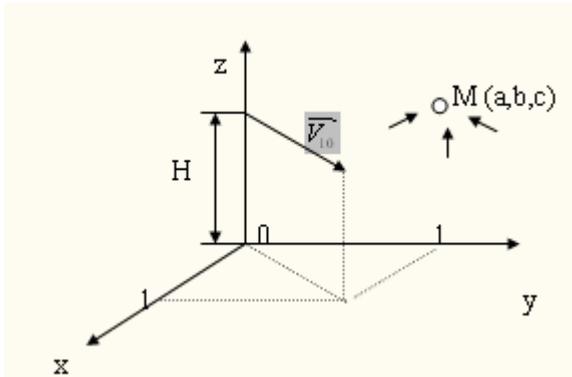


Рис.9.2

Лабораторная работа № 6.

Определение максимального диаметра пылевой частицы, улавливаемой отсосом воздуха от аспирационного укрытия

Цель работы: определить максимальный диаметр частицы, улавливаемой отсосом от аспирационного укрытия.

Содержание

1. Исходные данные.
2. Результаты расчета, представленные в табличном виде.
3. Результаты расчета, представленные в графическом виде.

Постановка задачи

1. По данным преподавателем программам рассчитать максимальный диаметр частицы пыли улавливаемой отсосом с точностью 5 мкм.
2. Найти траектории частиц пыли диаметром d_{\max} и $d_{\max} - 5$ мкм.
3. Построить найденные траектории частиц пыли.

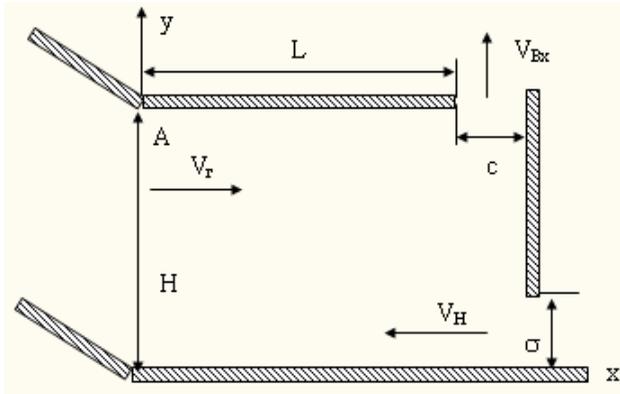


Рис.9.3

Аспирационное укрытие имеет следующие параметры: коэффициент динамической формы частицы пыли $\chi = 1$; коэффициент динамической вязкости $\mu = 1.776 \cdot 10^{-5} \text{ Па} \cdot \text{с}$

Варианты заданий 1 – 9 (рис.9.3)

Варианты	$V_G, \text{м/с}$	$V_{Вк}, \text{м/с}$	$V_H, \text{см/с}$	$L, \text{м}$	$H, \text{м}$	$c, \text{м}$	$\sigma, \text{м}$	$\rho, \text{кг/м}^3$
1	1,11	1,44	0,54	0,3	0,3	0,25	0,05	2600
2	1,41	1,56	0,9	0,78	0,3	0,3	0,05	2600
3	0,67	0,74	0,6	0,6	0,45	0,45	0,05	2600
4	3,4	2,92	0,56	0,9	0,55	0,65	0,05	2600
5	3,5	4,85	0,47	0,4	0,7	0,51	0,05	3500
6	3,73	1,97	1,3	1,15	0,35	0,7	0,05	1400
7	1,11	1,44	0,54	0,3	0,3	0,25	0,05	1000
8	1,41	1,56	0,9	0,78	0,3	0,3	0,05	2000
9	0,67	0,74	0,6	0,6	0,45	0,45	0,05	3500

Вариант 10 – 17.

Частицы вытекают из т. А (0, h_0), рис. 9.4

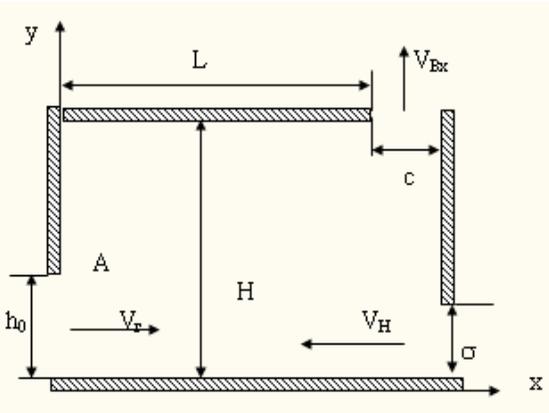


Рис.9.4

Варианты	$V_Г, \text{м/с}$	$V_{Вх}, \text{м/с}$	$V_Н, \text{см/с}$	$L, \text{м}$	$H, \text{м}$	$c, \text{м}$	$\sigma, \text{м}$	$h_0, \text{м}$	$\rho, \text{кг/м}^3$
10	3,33	1,44	0,54	0,2	0,3	0,25	0,05	2/3	2600
11	4,23	1,56	0,9	0,2	0,3	0,3	0,05	2/3	2600
12	2,01	0,74	0,6	0,2	0,45	0,45	0,05	2/3	2600
13	10,2	2,92	0,56	0,2	0,55	0,65	0,05	2/3	2600
14	10,5	4,85	0,47	0,2	0,7	0,51	0,05	2/3	3500
15	11,19	1,97	1,3	0,2	0,35	0,7	0,05	2/3	1400
16	3,33	1,44	0,54	0,2	0,3	0,25	0,05	2/3	1000
17	3,33	1,44	0,54	0,2	0,3	0,25	0,05	2/3	4000

Вариант 18 – 25

Частицы вылетают из т. А (0, H), рис. 9.5

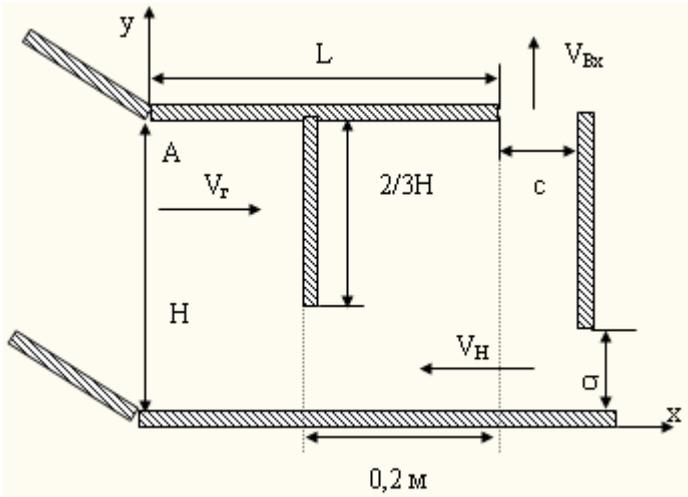


Рис.9.5

Вариант	$V_T, \text{м/с}$	$V_{Вх}, \text{м/с}$	$V_H, \text{см/с}$	L, м	H, м	c, м	σ , м	ρ , кг/м
18	1,11	1,44	0,54	0,3	0,3	0,25	0,05	2600
19	1,41	1,56	0,9	0,78	0,3	0,3	0,05	2600
20	0,67	0,74	0,6	0,6	0,45	0,45	0,05	2600
21	3,4	2,92	0,56	0,9	0,55	0,65	0,05	2600
22	3,5	4,85	0,47	0,4	0,7	0,51	0,05	3500
23	3,73	1,97	1,3	1,15	0,35	0,7	0,05	1400
24	1,11	1,44	0,54	0,3	0,3	0,25	0,05	1000
25	1,41	1,56	0,9	0,78	0,3	0,3	0,05	2000

Литература

1. Логачев И.Н., Логачев К.И. Аэродинамические основы аспирации. Санкт-Петербург: Химиздат. – 2005. – 659с.
2. Логачев К.И. Аэродинамика всасывающих факелов. Белгород: Изд-во БелГТАСМ, 2000. - 175с.
3. Бенерджи П., Баттерфилд Р. Метод граничных элементов в прикладных науках. - М.:Мир,1984.-486 с.

4. Фабрикант Н.Я. Аэродинамика - М.:Наука, 1964. - 814 с.
5. Нейков О.Д., Логачев И.Н. Аспирация и обеспыливание воздуха при производстве порошков. - М.: Металлургия, 1981. - 192 с.
6. Минко В.А. Обеспыливание технологических процессов производства строительных материалов. - Воронеж: Изд-во Воронежского университета, 1981. - 176 с.