

Задача 5

Температурное поле в полуограниченном пространстве $y \in [0, \infty]$ при условиях $T(0, t) = 1$, $T(\infty, t) = 0$, $T(y, 0) = 0$ описывается выражением

$$T(x) = \operatorname{erfc}(x),$$

где $x = y / (2 \cdot \sqrt{a \cdot t})$, a – коэффициент теплопроводности, y , t – пространственная и временная координаты.

Определить значение x , при котором

$$T(x) = 0.5.$$

Решение.

Уравнение $\operatorname{erfc}(x) - 0.5 = 0$, с учетом $\operatorname{erfc}(x) = 1 - \operatorname{erf}(x)$, преобразуется к виду

$$\operatorname{erf}(x) = 0.5, \quad (5.1)$$

где

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (5.2)$$

Пояснения к решению.

Для решения уравнения (5.1) можно воспользоваться методом деления отрезка пополам (метод бисекции), который был рассмотрен ранее.

Для нахождения значения интеграла (5.2) можно воспользоваться формулой трапеции для вычисления площади фигуры, ограниченной некоторой функцией. Для интеграла (5.2) фигура ограничена прямыми $x=0$ и $x=x^*$ (x^* – верхнее значение интеграла), осью абсцисс и кривой подынтегральной функции.

Квадратурная формула трапеции для численного интегрирования

$$I = \int_a^b f(x) dx = h * \left(\frac{f(a) + f(b)}{2} + \sum_{i=1}^{n-1} f(x_i) \right)$$

Простая программа для вычисления интеграла (5.2) может быть представлена в Octave. Например, для вычисления $\operatorname{erf}(0.5)$

```
y = erf (0.5);  
printf(" \t Интеграл = %f \t \n", y);
```

В файле с именем erf.m записана функция

```
function erfx = erf (x)  
function retval = ef (t)  
retval = exp (-t^2);  
endfunction  
  
n=1000;  
dt=(x-0)/(n-1);  
s=0;  
xti=0+dt;  
for i=1:n-1  
s = s+ef(xti);  
xti=xti+dt;  
endfor  
erfx = (2/pi)*dt*((ef(0)+ef(x))/2 + s);
```

endfunction

В этом фрагменте функции erf(x) используется внутренняя функция ef(t), отвечающая подынтегральной функции.

В данном случае, листинг с ответом, имеет вид:

Интеграл = 0.293909

Программа решения уравнения (C++)

```
#include <iostream>
#include <math.h>
using namespace std;
const double PI = 3.14159265;

double ff (double x){ // Определение значения функции
    double u = erf(x)-0.5; return u; }

// Решение нелинейного уравнения методом половинного деления
double fx (double a, double b , double eps) {
    double fa=ff(a), fb=ff(b);
    if (fa*fb>0)
        printf("\t Необходимо изменить интервал, т.к. fa*fb > 0 \n" );
    else {
        double z=0;
        for (int i=0;i<100;i++){ // Предельное число итераций = 100
            z=(a+b)/2;
            if (ff(a) * ff(z)<0) b=z; else a=z;
            if (abs(b-a)<eps) return z; }
    }
    return 0;}

// Численное интегрирование в диапазоне от 0 до 10
double erf(double xx)
{
    double xt1=0, ddx=0.0001, xt2=0, xfin=10;
    double integ=0, kk= 2.0/sqrt(PI);
    do {
        xt1=xt2; xt2=xt1+ddx;
        integ+=0.5*(exp(-xt1*xt1)+ exp(-xt2*xt2) );}
    while (xt2<xx);
    double zz = kk*integ*ddx;
    return zz;
}

int main() // Решение уравнение
{
```

```
double a=0.1, b=2.5, eps=1.0E-9;
double xx=fx(a,b,eps);           // Метод поиска корня уравнения
double nev = abs(ff(xx));         // Проверка точности вычисления корня
printf("\t Решение x = %f    Невязка = %f\n", xx, nev);
return 0;
}
```

Листинг с ответом

```
Решение x = 0.476900    Невязка = 0.000069
```

Если воспользоваться таблицей значений функции ошибок

Таблица значений функции ошибок

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

x	erf(x)	x	erf(x)	x	erf(x)
0.00	0.00000	1.00	0.8427	2.00	0.9953
0.10	0.1125	1.10	0.8802	2.10	0.9970
0.2	0.2227	1.20	0.9103	2.20	0.9981
0.3	0.3286	1.30	0.9340	2.30	0.9989
0.4	0.4284	1.40	0.9523	2.40	0.9993
0.5	0.5205	1.50	0.9661	2.50	0.9996
0.6	0.6039	1.60	0.9764	2.60	0.9998
0.7	0.6778	1.70	0.9838	2.70	0.9999
0.8	0.7421	1.80	0.9891		
0.90	0.7969	1.90	0.9928		

Программа решения задачи с учетом значений таблицы функции ошибок (Java) -

```
package erf;
```

```
public class erfz {
    public static void main(String[] args) {
        double a=0.1, b=2.5, eps=1e-7;
        double xx=fx(a,b,eps);           // Метод поиска корня уравнения
```

```

double nev = Math.abs(ff(xx));
System.out.println (" Решение =" + xx + " Невязка = " + nev);
    }

    // Метод половинного деления
public static double fx (double a, double b , double eps) {
    double fa=ff(a), fb=ff(b);
    if (fa*fb>0)System.out.println("Необходимо изменить интервал, т.к. fa*fb > 0");
    double z=0;
    for (int i=0;i<100;i++){ // Предельное число итераций
        z=(a+b)/2;
        if (ff(a) * ff(z)<0) b=z; else a=z;
        if (Math.abs(b-a)<eps) return z; }
    return 0;}

    // Метод определения значения функции
public static double ff (double xd){
    double u = erfxd(xd)-0.5; return u; }

    // Метод определения значения функции erf(x), используя таблицу
    public static double erfxd(double xx){
double [] x={ 0.0, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90,
    1.00, 1.10, 1.20, 1.30, 1.40, 1.50, 1.60, 1.70, 1.80, 1.90,
    2.00, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7};
double [] erff={0.0, 0.1125, 0.2227, 0.3286, 0.4284, 0.5205,
    0.6039, 0.6778, 0.7421, 0.7969, 0.8427, 0.8802, 0.9103,
    0.9340, 0.9523, 0.9661, 0.9764, 0.9838, 0.9891 ,0.9928,
    0.9953, 0.9970, 0.9981, 0.9989, 0.9993, 0.9996 ,0.9998, 0.9999};
double fx=-10000;
    if (xx<=x[0]) return erff[0];
    if (xx>=x[x.length-1]) return erff[erff.length-1];
    for(int i=0; i<erff.length-1;i++)
    {
        if ((x[i]<=xx) && (x[i+1]>xx))
        {
            fx=erff[i] + ((xx-x[i])/(x[i+1] - x[i]))*(erff[i+1]-erff[i]);
            return fx;}
        }
    return 0;}
}

```

Листинг с ответом –

Решение =0.4777416467666627 Невязка = 5.667209634907522E-8