Министерство образования Республики Беларусь

Учреждение образования «Белорусский Государственный Университет Информатики и Радиоэлектроники»

Кафедра «Информатика»

Лабораторная работа № 5

«Моделирование параболических уравнений в частных производных используя методы расщепления»

по курсу «Методы численного анализа»

**Выполнил:**

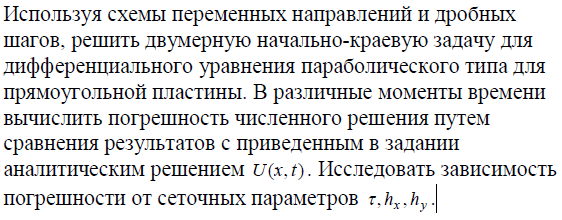
Волчецкий А. М. гр. 253504

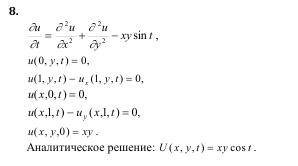
**Проверила:**

Ероминек К.Р.

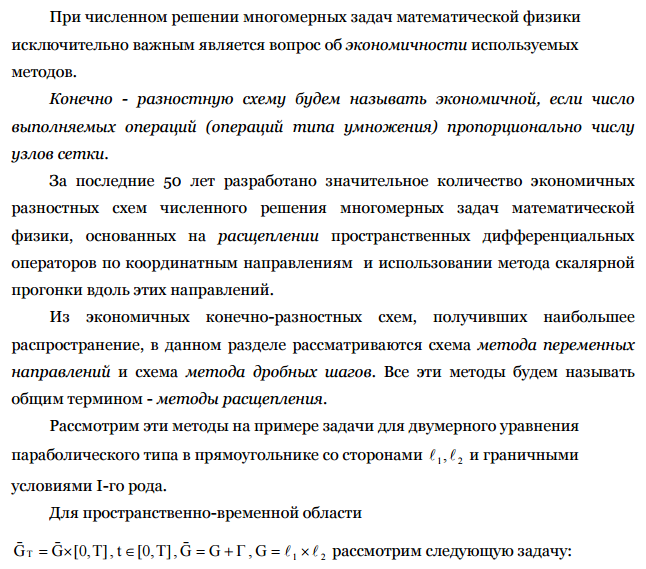
Минск 2015

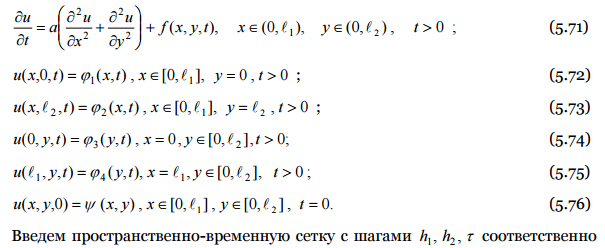
**Задание**

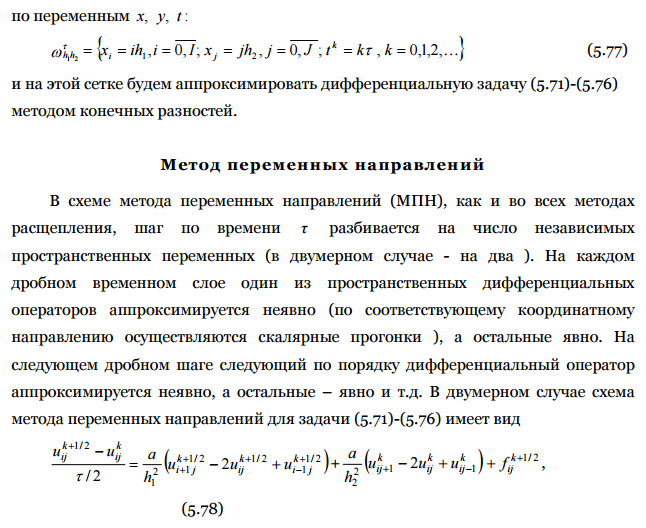


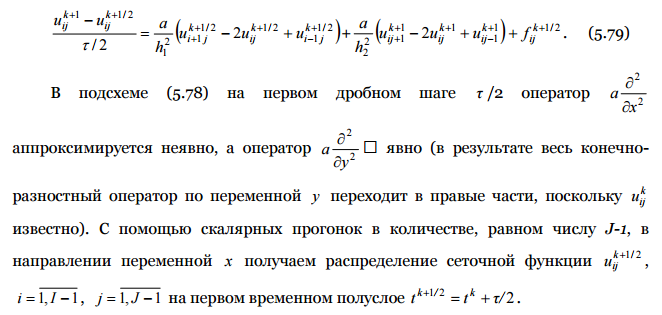


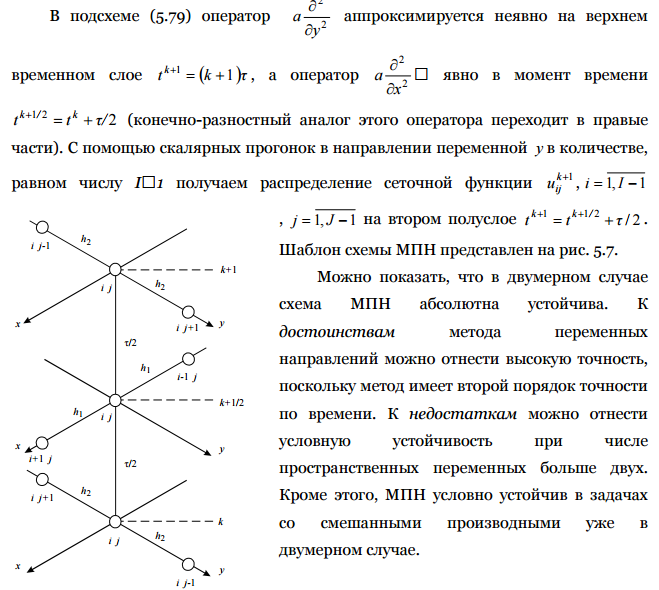
Теоретические сведения





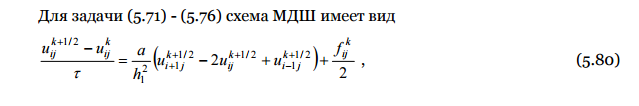


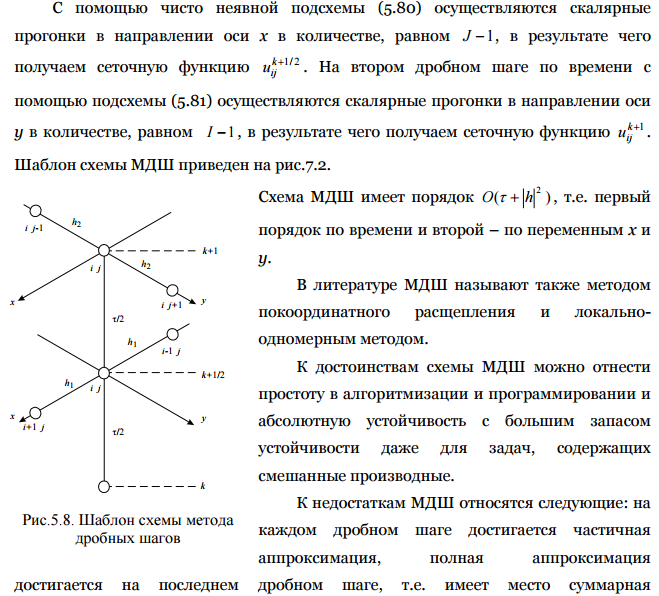




Метод дробных шагов

В отличие от МПН метод дробных шагов (МДШ) использует только неявные конечно-разностные операторы, что делает его абсолютно устойчивым в задачах, не содержащих смешанные производные. Он обладает довольно значительным запасом устойчивости и в задачах со смешанными производными.







Код программы

var g;

var graph='u';

var scheme='impl';

var tb\_tmax;

var a;

var b;

var mu;

var method='val\_dir';

var cut='y';

var Nx=100;

var Ny=100;

var K=100;

var tau;

var t;

var decimalplaces=3;

var x0=0.0;

var y0=0.0;

var lx=Math.PI;

var ly=Math.PI;

var t0=0.0;

var U=Matrix3(Nx+1,Ny+1);

var omega=1.5;

var log='';

//var nval\_dir;

function fi1(y,t)

{

return 0;

}

function fi2(y,t)

{

return -Math.sin(y)\*Math.sin(mu\*t);

}

function fi3(x,t)

{

return 0;

}

function fi4(x,t)

{

return -Math.sin(x)\*Math.sin(mu\*t);

}

function ksi(x,y)

{

return 0;

}

function f(x,y,t)

{

return Math.sin(x)\*Math.sin(y)\*(mu\*Math.cos(mu\*t)+(a+b)\*Math.sin(mu\*t));

}

function U\_real(x,y,t)

{

return Math.sin(x)\*Math.sin(y)\*Math.sin(mu\*t);

}

function Matrix(n,m)

{

var M=[];

for (var i=0;i<n;i++)

{

M[i]=[];

for (var j=0;j<m;j++)

M[i][j]=0;

}

return M;

}

function step\_norm()

{

try{

hx=lx/Nx;

hy=ly/Ny;

}catch(e){alert(e);}

}

function a\_change()

{

try{

a=parseFloat(tb\_a.value);

step\_norm();

}catch(e){alert(e);}

}

function b\_change()

{

try{

b=parseFloat(tb\_b.value);

step\_norm();

}catch(e){alert(e);}

}

function mu\_change()

{

try{

mu=parseFloat(tb\_mu.value);

step\_norm();

}catch(e){alert(e);}

}

function Nx\_change()

{

try{

Nx=parseFloat(tb\_Nx.value);

tb\_i.max=Nx;

}catch(e){alert(e);}

}

function Ny\_change()

{

try{

Ny=parseFloat(tb\_Ny.value);

tb\_j.max=Ny;

}catch(e){alert(e);}

}

function tau\_change()

{

try{

tau=parseFloat(tb\_tau.value);

}catch(e){alert(e);}

}

function t\_change()

{

try{

t=parseFloat(tb\_t.value);

if (t>K)

{

t=K;

tb\_t.value=t;

}

draw();

}catch(e){alert(e);}

}

function tmax\_change()

{

try{

K=parseInt(tb\_tmax.value);

}catch(e){alert(e);}

}

function omega\_change()

{

try{

omega=parseFloat(tb\_omega.value);

}catch(e){alert(e);}

}

function i\_change()

{

try{

draw();

}catch(e){alert(e);}

}

function j\_change()

{

try{

draw();

}catch(e){alert(e);}

}

function rb\_x\_click()

{

try{

cut='x';

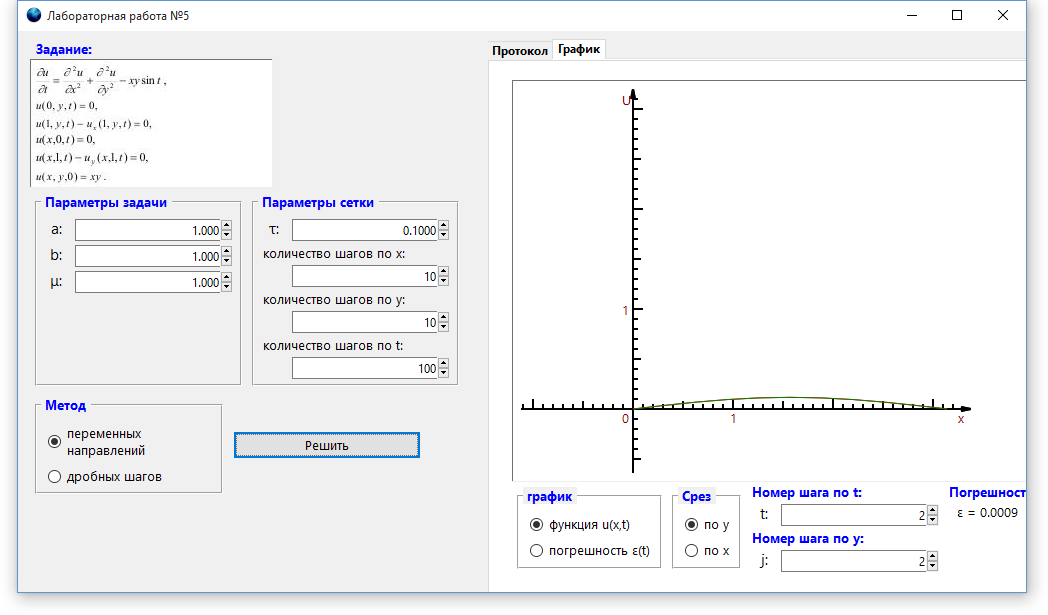
vb\_j.hidden=true;

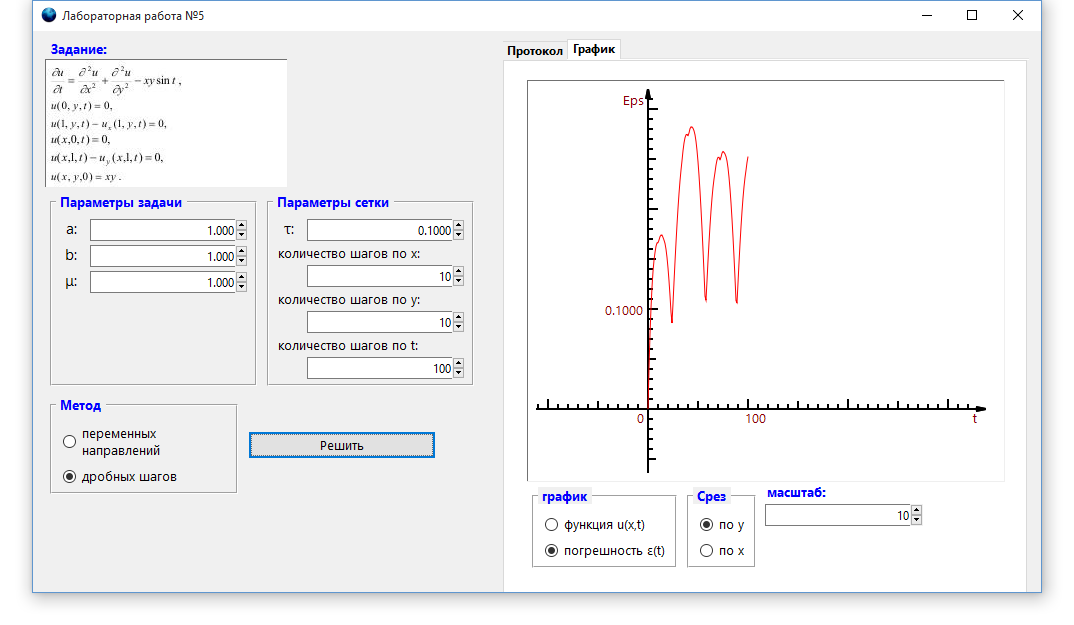
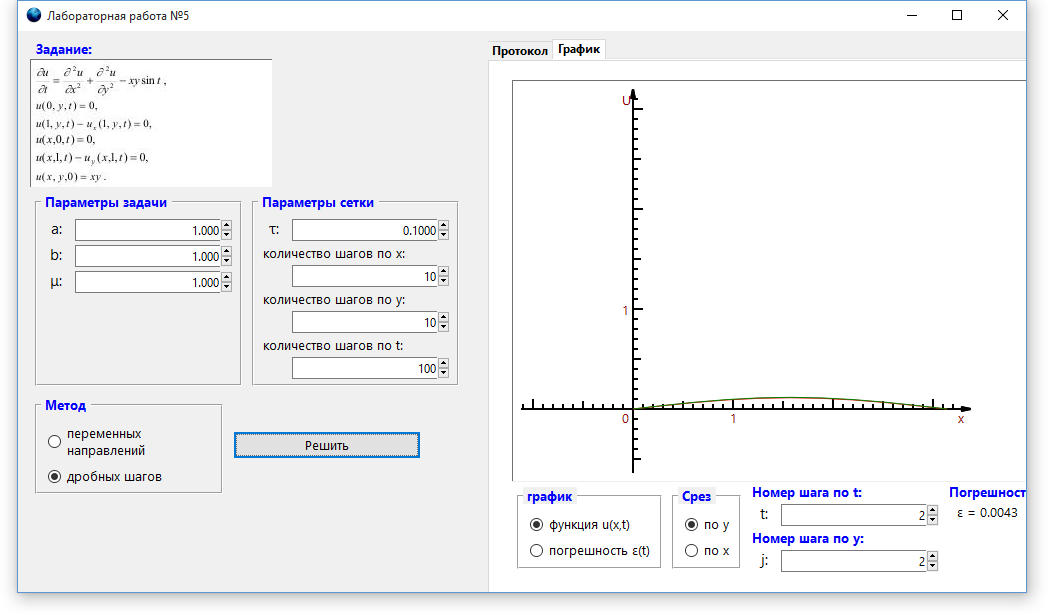
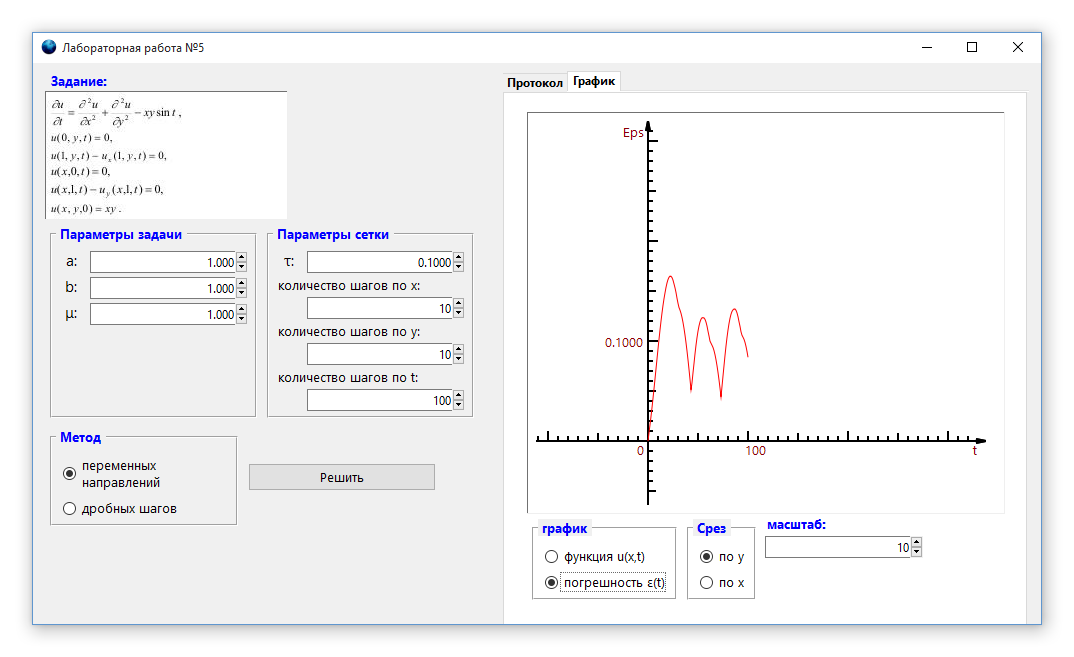
vb\_i.hidden=false;

draw();

}catch(e){alert(e);}

}





Результаты выполнения алгоритма показали, что погрешность решения методом переменных направлений меньше погрешности решения методом дробных шагов. Наибольшее влияние на погрешность оказывает параметр ῖ - шаг по времени.

**Вывод**

В данной работе были рассмотрены методы переменных направлений и дробных шагов для решения двумерной начально-краевой задачи для дифференциального уравнения параболического типа. Прежде всего, было дано понятие двумерной начально-краевой задачи для параболического дифференциального уравнения и выделены основные идеи методов переменных направлений и дробных шагов, приведен их алгоритм в среде xulrunner и построен график решения. Результаты показали, что на решение задачи влияют размер шага по оси x, размер шага по оси y, шаг по временной оси и взятые начальные и граничные значение. К достоинствам метода переменных направлений можно отнести высокую точность. К недостаткам можно отнести условную устойчивость при числе пространственных переменных больше двух. К достоинствам схемы МДШ можно отнести простоту в алгоритмизации и программировании и абсолютную устойчивость с большим запасом устойчивости даже для задач, содержащих смешанные производные. К недостаткам МДШ относятся следующие: на каждом дробном шаге достигается частичная аппроксимация, полная аппроксимация достигается на последнем дробном шаге, т.е. имеет место суммарная аппроксимация.