

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЙ КОМПЛЕКС ПРОГРАММ ДЛЯ ПРОВЕДЕНИЯ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА ПРИ МОДЕЛИРОВАНИИ ТЕПЛООВОГО ПРОЦЕССА

Масина О. Н.

(Россия, Елец)

Разработан проблемно-ориентированный комплекс программ для проведения вычислительного эксперимента при численном моделировании теплового процесса, описываемого нелинейным уравнением теплопроводности.

Рассматривается динамическая система, в которой происходит нагревание закрытой комнаты, представляемой как однородный стержень, теплоизолированный с боков, техническим устройством с автоматической регуляцией температуры. При этом потребуются различные датчики, например, температуры воздуха, температуры, подаваемой устройством, и т.д. Для простоты будем использовать в качестве модели такой системы только техническое устройство, названное кондиционером и описываемое одномерным уравнением теплопроводности

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} + f(U, x, t), \quad (1)$$

с начальным условием $U(x, 0) = g(x)$, с граничными значениями $U(0, t) = h_1(t)$ и $U(A, t) = h_2(t)$ соответственно, где $U(x, t)$ — температура в области $G := \{(x, t): 0 \leq x \leq A, 0 \leq t \leq T\}$, функция $f(U, x, t)$ характеризует внешние источники тепла в момент времени t в точке x . Здесь $g(x)$, $f(U, x, t)$, $h_1(t)$ и $h_2(t)$ — заданные функции. Предполагается, что начальные и граничные условия согласованы.

Для численного моделирования соответствующего процесса построим в прямоугольной области G равномерную пространст-

венно-временную сетку $\{x_i = ih, t_j = jv, 0 \leq i \leq s, 0 \leq j \leq k\}$ с шагами $h = \frac{A}{s}$ и $v = \frac{T}{k}$ [1]. Значения функции в узлах сетки заменим соответствующими значениями сеточной функции $u(i, j) := u(x_i, t_j)$, а правую часть $f(U, x, t)$ — сеточной функцией $f(u(i, j), i, j) := f(u(x_i, t_j), x_i, t_j) := f(i, j)$. Тогда, заменяя производную $\frac{\partial U}{\partial t}$ в точке (x_i, t_j) разностным отношением $\frac{\partial U}{\partial t} \approx \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{v}$,

а производную $\frac{\partial^2 U}{\partial x^2}$ — второй разностной производной:
 $\frac{\partial^2 U}{\partial x^2} \approx \frac{u(x_{i-1}, t_{j+1}) - 2u(x_i, t_{j+1}) + u(x_{i+1}, t_{j+1})}{h^2}$, аппроксимируя на-

чальные и граничные условия в граничных узлах сетки, и вводя произвольный вещественный параметр σ (вес верхнего слоя $t = t_{j+1}$), получим однопараметрическое семейство схем с весами, являющееся обобщением явной и неявной разностных схем [1]:

$$\sigma \gamma u(i-1, j+1) - (1+2\sigma\gamma)u(i, j+1) + \gamma \sigma u(i+1, j+1) = -F^*(i), \quad (2)$$

$$0 < i < s, 0 \leq j \leq k-1,$$

$$u(0, j+1) = h_i(t_{j+1}), u(A, j+1) = h_i(t_{j+1}), 0 \leq j \leq k-1,$$

$$u(i, 0) = g(x_i), 0 \leq i \leq s,$$

где

$$F^*(i) := (1 - 2\gamma(1 - \sigma))u(i, j) + (1 - \sigma)\gamma(u(i+1, j) + u(i-1, j)) + v f(i),$$

$$\gamma = v / h^2.$$

После деления на $-(1+2\sigma\gamma)$, система (2) примет вид:

$$\left\{ \begin{array}{l} u(1) - a(1)u(2) = Fn(1), \\ -a(1)u(1) + u(2) - a(1)u(3) = Fn(2), \\ -a(1)u(2) + u(3) - a(1)u(4) = Fn(3), \\ \dots \\ -a(1)u(s-3) + u(s-2) - a(1)u(s-1) = Fn(s-2), \\ -a(1)u(s-2) + u(s-1) = Fn(s-1), \end{array} \right. \quad (3)$$

где

$$\begin{aligned} u(i) &::= u(i, j+1), \quad 0 < i < s, \\ a(1) &:= \sigma\gamma / (1 + 2\sigma\gamma), \\ Fn(1) &:= [\sigma\gamma u(0) + F^*(1)] / (1 + 2\sigma\gamma), \\ Fn(s-1) &:= [\sigma\gamma u(s) + F^*(s-1)] / (1 + 2\sigma\gamma), \\ Fn(i) &:= F^*(i) / (1 + 2\sigma\gamma), \quad 2 \leq i \leq s-2. \end{aligned}$$

Приводя систему (3) к ступенчатому виду, имеем:

$$\left\{ \begin{array}{l} u(1) - a(1)u(2) = F(1), \\ u(2) - a(2)u(3) = F(2), \\ u(3) - a(3)u(4) = F(3), \\ \dots \\ u(s-2) - a(s-2)u(s-1) = F(s-2), \\ u(s-1) = F(s-1), \end{array} \right.$$

где $a(i) := \frac{a(1)}{1 - a(1)a(i-1)}, F(i) := \frac{Fn(i) + a(1)Fn(i-1)}{1 - a(1)a(i-1)}, 2 \leq i < s,$

и получим решение по алгоритму, аналогичному методу прогонки [5].

Разностная схема с весами устойчива [5] при $\sigma \geq \frac{1}{2} - \frac{h^2}{4v}$.

Изменяя параметр $0 < \sigma < 1$, можно управлять устойчивостью и точностью разностной схемы с весами.

Задача. Реализовать алгоритм численного метода решения задачи теплопроводности, а точнее, схемы с весами, в форме проблемно-ориентированного комплекса программ в среде программирования Turbo Pascal 7.0. для организации работы уст-

ройства, производящего нагрев воздуха от минимальной температуры до максимальной с последовательным ее регулированием. Изучить влияние параметра устойчивости σ на точность вычислений.

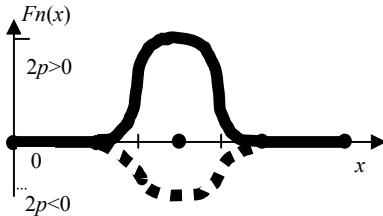


Рис. 1. График гладкой функции для задания теплового потока

больше величины 30^0 поток уменьшался вплоть до отрицательных значений, при уменьшении температуры α_2 ниже 10^0 поток увеличивался, а в интервале от 10^0 до 30^0 поток оставался без изменения. Тогда тепловой поток в каждый момент времени будет задан гладкой функцией с графиком (при $p > 0$ и при $p < 0$) на рис. 1.

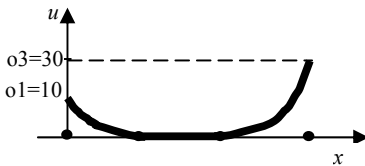


Рис. 2. График начального значения

Рассмотрим общий алгоритм решения. В качестве минимального и максимального значения температур будем использовать константы $\alpha_1 = 10$ и $\alpha_3 = 30$ соответственно. Для задания теплового потока введем функцию F_n таким образом, чтобы при средней температуре α_2

Данные расчета температуры u будем выводить в текстовом файле `otshet1.doc`. Вычисление значений температуры u организуем в процедуре `trote` следующим обра-

зом. Отрезок $\Delta x = 10$ разделим на $x = 128$ равных частей длины $h = \Delta x / x$ и интервал времени $T = 1$ на $t_u = 128$ частей величины $v = T / t_u$. Далее вычислим начальные значения температуры u и запишем их в файл вывода. Для этого обратимся к функции (подпрограмме) $G_n(i)$. График функции G_n для организации начального значения температуры представлен на рис. 2.

Затем организуем внешний цикл по времени t и найдем граничные значения слева и справа, обратившись к функциям

$HL(j)$ и $HR(j)$. Графики функций HL и HR для задания граничных температур слева и справа представлены на рис. 3.

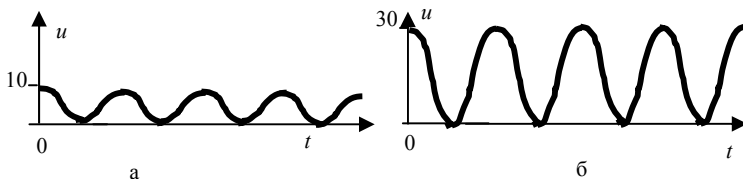


Рис. 3. Графики граничного значения: а) слева; б) справа

Во внутреннем цикле по x , следуя алгоритму, рассмотренному выше, просчитаем значения вспомогательной функции u_n и присвоим их искомой функции u . Допишем эти значения в файл вывода и закрытием внешнего цикла завершим работу процедуры `trote`.

Инициализируя графический режим работы, построим поле и основу графики в форме параллелепипеда. Округляя значения температуры до целых чисел, построим график начального значения. Воспользовавшись масштабными коэффициентами `mot` и `mom`, построим графики левого и правого граничных условий, используя для них разные цвета.

Для подсчета точности метода первоначально разделим x на 4 равные части, выполним процедуру `trote`, а затем удвоим число частей и повторно проделываем процедуру `trote`. Найдем модуль разности между значениями соответствующих температур и допишем полученный результат (точность) в файл вывода. После чего файл закроем. Затем прекратим работу графического режима и завершим работу программы.

Изменяя вручную значения константы σ , характеризующей параметр устойчивости σ , по данным файла вывода будем судить о точности вычислений.

Приведем текст программы, реализующей описанный выше алгоритм.

```
program TEPLO; {название программы}  
uses crt, graph; {подключение модулей crt и graph }
```

```
{параметр устойчивости, минимальная и максимальная температура, длины отрезков по пространству и времени, число частей по пространству и времени}
const o=0.5;o1=10;o3=30;Ax=10;T=1;x=128;tu=128;
{определение графических констант: координаты точки основания графики, коэффициенты пропорциональности}
const gx0=50;gx=128;gy0=330;gy=128;mot1=1;mom1=5;
var p,i,j,t1,x0,x1,L:integer;
    mom,mot,gHl,gHlo,gHR,gHRo,Gnc:integer;
    driver,mode,gx1,gx2,gx3,gy1:integer;
    o2,r,v,v1:real;
    ots:text;
    u,w,z,f,a,un:array[0..x] of real;
(*****начальное значение*****)
function Gn(i:integer):real;
begin
    Gn:=0;
    if i<=x0 then Gn:=o1*(1+sin((i-2*x0)*pi/(2*x0)));
    if i>=3*x0 then Gn:=o3*(1+sin(i*pi/(2*x0)));
end;
(*****граничное значение слева*****)
function HL(j:integer):real;
begin
    HL:=o1*(1+cos(j/5))/2
end;
(***8****граничное значение справа*****)
function HR(j:integer):real;
begin
    HR:=o3*(1+sin(j/5))/2
end;
(*****тепловой поток*****)
function Fn(i:integer):real;
BEGIN
    if o2>=o3 then p:=p-1;
    if o2<o1 then p:=p+1;
    if i<x0 then Fn:=0;
    if i>3*x0 then Fn:=0;
```

```

if xo<=i then begin if i<=2*xo then Fn:=p*(1+cos(pi*i/xo)) end;
if 2*xo<=i then begin if i<=3*xo then Fn:=p*(1+cos(pi*i/xo)) end
END;
function m (i:integer):real;
begin
m:=(1-2*v1*(1-o))*u[i]+(1-o)*v1*(u[i+1]+u[i-1])+v*Fn(i)
end;
(****процедура вычисления численного метода*****)
procedure trote(xo:integer);
var x1,t1:integer;
v,h:real;
BEGIN
x1:=4*xo;t1:=tu;
h:=Ax/x1; v:=T/t1; v1:=v/(h*h); o2:=0;
for i:=0 to x1 do
begin u[i]:=Gn(i);o2:=o2+u[i]/(x1+1); write(ots,u[i]) end;
write(ots,'');
for j:=0 to t1 do
BEGIN
w[0]:=HL(j); w[x1]:=HR(j); a[1]:=o*v1/(1+2*o*v1);
f[1]:=(v1*u[0]*o+m(1))/(1+2*o*v1);
f[x1-1]:=(v1*u[x1]*o+m(x1-1))/(1+2*o*v1);
for i:=2 to x1 do
begin f[i]:=m(i)/(1+2*o*v1);
a[i]:=a[1]/(1-a[1]*a[i-1]);w[i]:=(f[i]+a[1]*f[i-1])/(1-
a[1]*a[i-1]) end;
un[x1-1]:=w[x1-1]; o2:=0;
for i:=x1-2 downto 1 do un[i]:=w[i]+un[i+1]*a[i];
for i:=0 to x1 do
begin u[i]:=un[i]; write(ots,u[i]);o2:=o2+u[i]/(x1+1); end;
write(ots,''); o2:=0;
END;
write(ots,'*');
END;
(*****основная программа*****)
BEGIN
assign(ots,'d:\programs\bp\otshet1.doc'); reset(ots);rewrite(ots);

```

```
driver:=vga;mode:=vgaHi;
initgraph(driver,mode,'d:\programs\bp\bgi');
mox:=gx div x;
moy:=gy div tu;mot:=gy*mot1 div tu;
mom:=2*mot;
textbackground(1); clrscr; {построение графического поля}
{построение основы графики}
setfillstyle(1,14);bar(10,10,600,600);
moveto(gxo,gyo); setcolor(11);
gx1:=gxo+gx; gx2:=gxo+2*gx; gx3:=gxo+3*gx; gy1:=gyo+gy;
lineto(gx1,gyo); lineto(gx3,gy1); lineto(gx2,gy1);
lineto(gxo,gyo);
(*****график начального значения*****)
  xo:=x div 4; Gnc:=0;
  for i:=0 to x do
    begin u[i]:=Gn(i);Gnc:=trunc(u[i]);lineto(gxo+i,gyo-Gnc*mom);
  end;
(*****график граничного значения слева*****)
setcolor(3);
moveto(gxo,gyo); gHL0:=mom*trunc(HL(0));
for j:=0 to tu do
begin gHL:=mom*trunc(HL(j+1));
  lineto(gxo+j*mom,gyo+j*mot-gHL0);
  lineto(gxo+(j+1)*mom,gyo+(j+1)*mot-gHL);
  lineto(gxo+(j+1)*mom,gyo+(j+1)*mot);
  moveto(gxo+(j+1)*mom,gyo+(j+1)*mot-gHL);
  gHL0:=gHL
end;
(*****график граничного значения справа*****)
setcolor(13);
moveto(gx1,gyo); gHR0:=mom*trunc(HR(0));
for j:=0 to tu do
  begin gHR:=mom*trunc(HR(j+1));
    lineto(gx1+j*mom,gyo+j*mot-gHR0);
    lineto(gx1+(j+1)*mom,gyo+(j+1)*mot-gHR);
    lineto(gx1+(j+1)*mom,gyo+(j+1)*mot);
    moveto(gx1+(j+1)*mom,gyo+(j+1)*mot-gHR);
```



```
gHRo:=gHR
end;
p:=0; xo:=x div 4; trote(xo); L:=2*xo;
for i:=0 to L do z[i]:=un[2*i];
p:=0; xo:=x div 8; x1:=x div 2; trote(xo);
{точность вычислений}
r:=0;
for i:=0 to L do
begin
if r<abs(z[i]-un[i]) THEN r:=abs(z[i]-un[i]);
end;
write(ots,'tochnost=', r);
close(ots);
repeat until keypressed; closegraph
END.
```

Результатом работы этой программы является графическое изображение на экране (рис. 4) и данные расчета температур, находящиеся в файле `otshet1.doc`.

Начальные и граничные условия, заданные в программе в виде функций (подпрограмм), можно изменять. Тогда получим целый комплекс программ для моделирования теплового процесса и исследования устойчивости движения теплового потока.

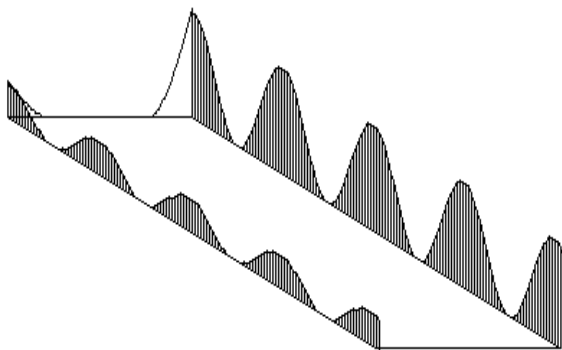


Рис. 4. Графическое изображение экрана в программе «ТЕПЛО»

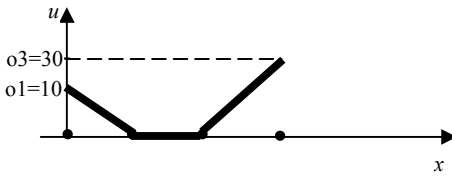


Рис. 5. График начального значения

Так, например, начальное значение температуры можно задать функцией, изображенным на рисунке 5, а граничные значения слева и

справа — функциями на рис.6. В этом случае получим уже другую программу с новыми начальными и граничными значениями.

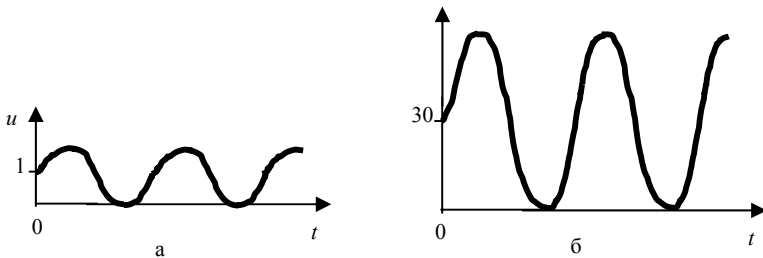


Рис. 6. Графики граничного значения: а) слева; б) справа

В результате работы комплекса получены следующие результаты:

1. при увеличении параметра устойчивости $\sigma \geq 1/2$ увеличивается точность вычислений (при $\sigma = 0,5$ точность равна 0,78957465043, а при $\sigma = 0,9$ изменяется до 0,038479249286);
2. решение непрерывно зависит от входных данных;
3. при использовании гладких функций увеличивается точность вычислений.

Разработанный проблемно-ориентированный комплекс программ может быть использован при решении многочисленных технических задач и на производстве тепловых сетей (в работе ТЭЦ, тепловых котлов, кондиционеров).

Список литературы:

1. Меренков Ю.Н. Численные методы решений одномерной модели, описываемой уравнением теплопроводности/Ю.Н.Меренков, О.Н.Масина // Вестник Елецкого государственного университета им. И. А. Бунина. Вып. 5: Серия «Математика. Физика». Елец: ЕГУ им. И. А. Бунина, 2004. С. 36–42.
2. Самарский А. А. Математическое моделирование: Идеи. Методы. Примеры / А.А.Самарский, А.П. Михайлов. — М.: ФИЗМАТЛИТ, 2002.
3. Самарский А. А. Теория разностных схем / А. А. Самарский. — М.: Наука, 1983.
4. Самарский А. А. Численные методы / А. А. Самарский, А.В. Гулин. — М.: Наука, 1989.
5. Тихонов А.Н. Уравнения математической физики / А.Н. Тихонов, А. А.Самарский. — Наука, 1996.

**THE PROBLEM-ORIENTED COMPLEX OF THE
PROGRAMS FOR REALIZATION OF COMPUTING
EXPERIMENT AT MODELING THERMAL PROCESS**

Masina O. N.

(Russia, Eletsck)

The problem-oriented complex of the programs for realization of computing experiment is developed at numerical modeling thermal process described by the nonlinear heat equation.