Лабораторная работа №2

«Численное решение уравнения Хопфа»

Квазилинейное уравнение переноса

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{1а}$$

или в дивергентной форме

$$\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = 0,\ F(u) = u^2/2, \tag{1б}$$

называется *уравнением Хопфа* и относится к основным модельным уравнениям вычислительной гидроаэродинамики, на которых тестируются различные разностные схемы.

В случае разрывных начальных данных

$$u(x,0) = \varphi(x) = \begin{cases} u_L\,, & x \le 0 \\ u_R\,, & x \ge 0 \end{cases} \tag{2}$$

задача (1) – (2) называется *задачей Римана* и имеет точное решение

$$u_{ex} = \begin{cases} u_L\,, & x \le Dt \\ u_R\,, & x \ge Dt \end{cases}$$

где $D = (u_L + u_R)/2$ – скорость распространения сильного разрыва (УВ).

Для численного решения (1) – (2) можно использовать одну из разностных схем:

КИР

$$\frac{v_i^{n+1} - v_i^n}{\tau} + \frac{1}{h}\begin{Bmatrix} F_{i+1}^n - F_i^n, & v_i^n < 0 \\ F_i^n - F_{i-1}^n, & v_i^n > 0 \end{Bmatrix} = 0, \tag{3}$$

Мак-Кормака 1

$$\frac{\tilde{v}_i - v_i^n}{\tau} + \frac{F_{i+1}^n - F_i^n}{h} = 0, \tag{4а}$$

$$\frac{v_i^{n+1} - (v_i^n + \tilde{v}_i)/2}{\tau} + \frac{\tilde{F}_i - \tilde{F}_{i-1}}{2h} = 0, \tag{4б}$$

Мак-Кормака 2

$$\frac{\tilde{v}_i - v_i^n}{\tau} + \frac{F_i^n - F_{i-1}^n}{h} = 0, \tag{5а}$$

$$\frac{v_i^{n+1} - (v_i^n + \tilde{v}_i)/2}{\tau} + \frac{\tilde{F}_{i+1} - \tilde{F}_i}{2h} = 0, \tag{5б}$$

Лакса-Вендроффа

$$\frac{v_i^{n+1} - v_i^n}{\tau} + \frac{F_{i+1}^n - F_{i-1}^n}{2h} -$$

$$\frac{\tau}{4h^2}\left[(v_{i+1}^n + v_i^n)(F_{i+1}^n - F_i^n) - (v_i^n + v_{i-1}^n)(F_i^n - F_{i-1}^n)\right] = 0, \tag{6}$$

неявная 1

$$\frac{v_i^{n+1} - v_i^n}{\tau} + \frac{F_i^{n+1} - F_{i-1}^{n+1}}{h} = 0 \tag{7}$$

неявная 2

$$\frac{v_i^{n+1} - v_i^n}{\tau} + \frac{1}{2}\left[\frac{F_i^{n+1} - F_{i-1}^{n+1}}{h} + \frac{F_i^n - F_{i-1}^n}{h}\right] = 0 \tag{8}$$

TVD

$$\frac{v_i^{n+1}-v_i^n}{\tau}+\frac{1}{h}\begin{Bmatrix} F_{i+1}^n - F_i^n, & \alpha_i < 0 \\ F_i^n - F_{i-1}^n, & \alpha_i > 0 \end{Bmatrix} = 0, \qquad (9)$$

где

$$\alpha_i = \begin{cases} v_i^n, & \Delta v \equiv v_{i+1}^n - v_i^n = 0 \\ \dfrac{F_{i+1}^n - F_i^n}{\Delta v}, & \Delta v \neq 0. \end{cases}$$

ЗАДАНИЕ. Методом конечных разностей решить для (1б) задачу Римана (2). Исследовать зависимость максимальной погрешности

$$\delta_{max} = \max_{1\le i\le N, 0\le n\le M} |v_i^n - u_{ex}(x_i, t_n)|$$

приближенного решения $\{v_i^n\}$ от шага сетки h (при фиксированном шаге по времени τ) и от шага по времени τ (при фиксированном шаге по пространству h).

Варианты задания приведены в таблице:

№ вар.	№ схемы	u_L	u_R	№ вар.	№ схемы	u_L	u_R
1	3	3/2	1/2	2	4	3/2	1/2
3	5	2	1	4	6	2	1
5	7	5/2	3/2	6	8	5/2	3/2
7	9	4/3	1/3	8	3	4/3	1/3
9	4	5/4	1/4	10	5	5/4	1/4
11	6	6/5	2/5	12	7	6/5	2/5
13	8	3/2	1/2	14	9	3/2	1/2
15	3	2	1	16	4	2	1
17	5	5/2	3/2	18	6	5/2	3/2
19	7	4/3	1/3	20	8	4/3	1/3
21	9	5/4	1/4	22	3	5/4	1/4
23	4	6/5	2/5	24	5	6/5	2/5

УКАЗАНИЯ. В качестве основы использовать программу «CONV1» (см. ниже), в которой для решения задачи (1) – (2) используется схема Лакса

$$\frac{v_i^{n+1}-(v_{i+1}^n+v_{i-1}^n)/2}{\tau}+\frac{F_{i+1}^n-F_{i-1}^n}{2h}=0. \qquad (10)$$

Выразив из (10) решение на новом временном слое v_i^{n+1}, получим следующий алгоритм:

$$v_1^{n+1} = u_L, \quad i = 1$$

$$v_i^{n+1} = \frac{v_{i+1}^n + v_{i-1}^n}{2} - \frac{\tau}{2h}(F_{i+1}^n - F_{i-1}^n), \quad i = 2, \dots, N_x.$$

Для расчета потока $F_{N_x+1}^n$ на правой границе в узле $i = N_x + 1$ применяем экстраполяцию

$$v_{N_x+1}^n = 2v_{N_x}^n - v_{N_x-1}^n. \qquad (11)$$

Этот прием применяем для всех схем 2-го порядка (4) - (6).

Для регуляризации численного решения, полученного по схемам 2-го порядка (4) - (6), используем явное сглаживание

$$v_i \to \hat{v}_i = (1-2\alpha)v_i + \alpha(v_{i-1}+v_{i+1}), \;\; i = 2, \ldots, N_x - 1, \;\; \alpha \sim 10^{-2} \ldots 10^{-1}.$$

Значение параметра α подбираем так, чтобы минимизировать одновременно численное «размазывание» и численную «рябь».

Текст программы «CONV1» и дополнительных процедур:

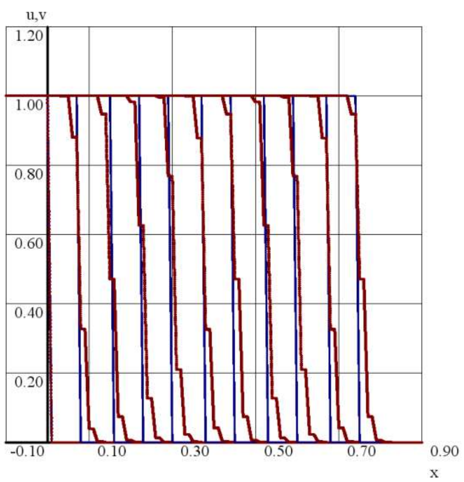
```
program Conv1
use GrafLib
parameter(Nx=101) ! chislo uzlov
dimension x(Nx),u(Nx),v(Nx),v1(1:Nx+1)
common /par1/uL,uR
pi2=8*atan(1.)
xL=-0.1; xR=0.9
h=(xR-xL)/(Nx-1)
tmax=1.5; gam=1.0 ! chislo Curanta
print*,' h=',h
print*,' gam=',gam
uL=1; uR=0 ! nach. dannie
t=0.
do i=1,Nx
 x(i)=xL+(i-1)*h
 u(i)=Phi(x(i),t)
 v(i)=u(i)
end do
tau=gam*h/maxval(abs(v)) ! nach. shag po vremeni
Nt=tmax/tau+0.5 ! chislo shagov po vremeni
nout=Nt/10  ! chastota vivoda
xgmin=dble(xL); xgmax=dble(xR)
ygmin=minval(u); ygmax=1.2*maxval(u)
call GrafInit(1,'Convection')
call Axis('x','u,v',0,2d-3)
call GRID(5,6,0)
ii=clickmenuqq(QWIN$TILE)
delmax=0
write(*,100)
do n=0,Nt ! ------
 del=sqrt(h*sum((u-v)*(u-v)))  !  porgeshnost  na
pred. sloe
 if(del>delmax)delmax=del
 if(mod(n,nout)==0)then ! vivod dannih
  call Xshock(Nx,x,v,xsh) ! koordinata YB
  write(*,200)n,t,tau,del,xsh
  call Plot2(dble(x),dble(u),1,500,2d-3)
  call Plot2(dble(x),dble(v),4,100,3d-3)
 end if
 tauh=tau/h
! Shema Laxa
 v1(1:Nx)=v ! zapominaem staroe reshenie
 v1(Nx+1)=2*v1(Nx)-v1(Nx-1)  !  extrapolyacia  za
pravuyu granicu
 do i=2,Nx  ! novoe reshenie vo vnutrennih uzlah
  v(i)=0.5*(v1(i+1)+v1(i-1)) &
     -0.5*tauh*(Flux(v1(i+1))-Flux(v1(i-1)))
 end do
 v(1)=uL ! novoe reshenie na levoi granice
! call SMOOTHING(v,Nx,3e-1) ! sglazhivznie
 t=t+tau
 do i=1,Nx ! tochnoe reshenie
  u(i)=Uex(x(i),t)
 end do
 tau=gam*h/maxval(abs(v)) ! novii shag po vremeni
end do ! ------
write(*,*)' delmax=',delmax ! maks. porgeshnost
ii=SAVEIMAGE_W('v(x,t).bmp',xgmin-
0.1,ygmax+0.1,xgmax+0.1,ygmin-0.1)
100 format(4x,'n',6x,'t',10x,'tau',9x,'del',8x,'xsh')
200 format(1x,i4,5(1x,1pe11.4))
pause
end

function Phi(x,t) ! nach. dannie
common /par1/uL,uR
 Phi=uL
 if(x>0) Phi=uR
end

function Uex(x,t) ! tochn. reshenie
common /par1/uL,uR
D=(uL+uR)/2.
if(x<=D*t) then
 Uex=uL
else
 Uex=uR
end if
end

function Flux(u) ! potok
Flux=u*u/2.
end

subroutine Xshock(Nx,x,v,xsh) ! rashet koord. YB
dimension x(Nx),v(Nx)
dvmax=0.0
do i=2,Nx
```

```
 dv=v(i)-v(i-1)
 if(abs(dv)>dvmax)then
  imax=i
  dvmax=abs(dv)
 end if
end do
xsh=x(imax)
end
subroutine SMOOTHING(v,Nx,alf) ! sglazhivanie
dimension v(Nx),v1(Nx)
v1=v
do i=2,Nx-1
 v(i)=(1-2*alf)*v1(i)+alf*(v1(i-1)+v1(i+1))
end do; end
```

Назначение процедур:
«Phi» – расчет начальных данных $\varphi(x)$, «Uex» – точного решения $u_{ex}(x,t)$, «Flux» – потока $F(u)$, «Xshock» – координаты конечно-разностной УВ, «SMOOTHING» – сглаживание численного решения.

Результаты запуска:

```
h=  9.9999998E-03
  gam=   1.000000
   n     t                 tau            del              xsh
   0  0.0000E+00  1.0000E-02  0.0000E+00  1.0000E-02
  15  1.5000E-01  1.0000E-02  4.9583E-02  8.0000E-02
  30  3.0000E-01  1.0000E-02  7.2001E-02  1.5000E-01
  45  4.5000E-01  1.0000E-02  7.5193E-02  2.4000E-01
  60  6.0000E-01  1.0000E-02  8.5655E-02  3.1000E-01
  75  7.5000E-01  1.0000E-02  4.9680E-02  3.8000E-01
  90  9.0000E-01  1.0000E-02  6.7818E-02  4.5000E-01
 105  1.0500E+00  1.0000E-02  7.5193E-02  5.4000E-01
 120  1.2000E+00  1.0000E-02  8.5655E-02  6.1000E-01
 135  1.3500E+00  1.0000E-02  4.9680E-02  6.8000E-01
 150  1.5000E+00  1.0000E-02  6.7818E-02  7.5000E-01
  delmax=  8.5970469E-02
```



Содержание отчета:

1) Постановка задачи с записью уравнения, НУ и ГУ для вашего варианта.
2) Разностная схема и численный алгоритм на ее основе.
3) Тексты главной программы и дополнительных процедур (кроме графического модуля).
4) Текстовые и графические результаты одного запуска программы для каждого из уравнений (1а) и (1б).
5) Таблицы и графики зависимостей $\delta_{max}(h)$ и $\delta_{max}(\tau)$. Их анализ и вывод о порядках точности схемы по времени и по пространству.
6) Определение скорости УВ из результатов численного решения, сравнение с точным значением.