Семинар 13. Решение волнового уравнения в двумерном случае.

1. Постановка задачи.

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x_1}\left(k(x_1,x_2)\frac{\partial u}{\partial x_1}\right) + \frac{\partial}{\partial x_2}\left(k(x_1,x_2)\frac{\partial u}{\partial x_2}\right) + f(x_1,x_2,t), \quad (x_1,x_2) \in D = (a_1,b_1)\times(a_2,b_2),$$

$$u(x_1,x_2,0) = \varphi(x_1,x_2), \quad u_t'(x_1,x_2,0) = \psi(x_1,x_2),$$

$$u(a_1,x_2,t) = g_{11}(t), \quad u(b_1,x_2,t) = g_{12}(t), \quad u(x_1,a_2,t) = g_{21}(t), \quad u(x_1,b_2,t) = g_{22}(t),$$

$$k(x_1,x_2) = \begin{cases} k_1, & (x_1,x_2) \in D_0, \\ k_2, & (x_1,x_2) \notin D_0, \end{cases} \quad D_0 = [x_{11},x_{12}]\times[x_{21},x_{22}],$$

$$f(x_1,x_2,t) = \sum_{m=1}^{2} Q_m \exp\left[-(x_1 - x_{1,m}^*)^2 / r_m^2 - (x_2 - x_{2,m}^*)^2 / r_m^2 \right] \sin(\omega_m t),$$

$$\varphi(x_1,x_2) = 0, \quad \psi(x_1,x_2) = 0, \quad g_{11}(t) = 0, \quad g_{12}(t) = 0, \quad g_{21}(t) = q_0\left[1 - \exp[-\omega_0 t]\right], \quad g_{22}(t) = 0.$$

2. Численный метод.

МКР на равномерной сетке $\Omega = \bar{\omega}_{x_1} \times \bar{\omega}_{x_2} \times \bar{\omega}_t$, $\bar{\omega}_{x_\alpha} = \left\{ x_{\alpha,i_\alpha} = i_\alpha h_\alpha, i_\alpha = 0,\ldots,N_\alpha, h_\alpha = \frac{b_\alpha - a_\alpha}{N_\alpha} \right\}$, $\alpha = 1,2,$

$$\bar{\omega}_t = \left\{ t_j = j\tau,\ j = 0,\ldots,N_t,\ \tau = \frac{t_{\max}}{N_t} \right\}.$$

Схема с весами: $\dfrac{y_{i_1i_2}^{j+1} - 2y_{i_1i_2}^{j} + y_{i_1i_2}^{j-1}}{\tau^2} =$

$$= \sigma\left[\frac{1}{\hbar_1}\left\{ k_{i_1+1/2,i_2}\frac{y_{i_1+1,i_2}^{j+1} - y_{i_1i_2}^{j+1}}{h_1} - k_{i_1-1/2,i_2}\frac{y_{i_1i_2}^{j+1} - y_{i_1-1,i_2}^{j+1}}{h_1} \right\} + \frac{1}{\hbar_2}\left\{ k_{i_1,i_2+1/2}\frac{y_{i_1,i_2+1}^{j+1} - y_{i_1i_2}^{j+1}}{h_2} - k_{i_1,i_2-1/2}\frac{y_{i_1i_2}^{j+1} - y_{i_1,i_2-1}^{j+1}}{h_2} \right\} + f_{i_1i_2}^{j+1} \right] +$$

$$+(1-2\sigma)\left[\frac{1}{\hbar_1}\left\{ k_{i_1+1/2,i_2}\frac{y_{i_1+1,i_2}^{j} - y_{i_1i_2}^{j}}{h_1} - k_{i_1-1/2,i_2}\frac{y_{i_1i_2}^{j} - y_{i_1-1,i_2}^{j}}{h_1} \right\} + \frac{1}{\hbar_2}\left\{ k_{i_1,i_2+1/2}\frac{y_{i_1,i_2+1}^{j} - y_{i_1i_2}^{j}}{h_2} - k_{i_1,i_2-1/2}\frac{y_{i_1i_2}^{j} - y_{i_1,i_2-1}^{j}}{h_2} \right\} + f_{i_1i_2}^{j} \right] +$$

$$+ \sigma\left[\frac{1}{\hbar_1}\left\{ k_{i_1+1/2,i_2}\frac{y_{i_1+1,i_2}^{j-1} - y_{i_1i_2}^{j-1}}{h_1} - k_{i_1-1/2,i_2}\frac{y_{i_1i_2}^{j-1} - y_{i_1-1,i_2}^{j-1}}{h_1} \right\} + \frac{1}{\hbar_2}\left\{ k_{i_1,i_2+1/2}\frac{y_{i_1,i_2+1}^{j-1} - y_{i_1i_2}^{j}}{h_2} - k_{i_1,i_2-1/2}\frac{y_{i_1i_2}^{j-1} - y_{i_1,i_2-1}^{j-1}}{h_2} \right\} + f_{i_1i_2}^{j-1} \right],$$

$$0 < i_\alpha < N_\alpha, \quad \alpha = 1,2, \quad 1 \le j < N_t - 1,$$

$$y_{i_1i_2}^0 = \varphi_{i_1i_2}, \quad y_{i_1i_2}^1 = y_{i_1i_2}^0 + \tau\psi_{i_1i_2},$$

$$y_{0,i_2}^j = g_{11}(t_j), \quad y_{N_1,i_2}^j = g_{12}(t_j), \quad y_{i_1,0}^j = g_{21}(t_j), \quad y_{i_1,N_2}^j = g_{22}(t_j),$$

$$k_{i_1\pm1/2,i_2} = \frac{2k_{i_1i_2}k_{i_1\pm1,i_2}}{k_{i_1i_2} + k_{i_1\pm1,i_2}}, \quad k_{i_1,i_2\pm1/2} = \frac{2k_{i_1i_2}k_{i_1,i_2\pm1}}{k_{i_1i_2} + k_{i_1,i_2\pm1}}, \quad k_{i_1i_2} = k(x_{1,i_1},x_{2,i_2}), \quad f_{i_1i_2}^j = f(x_{1,i_1},x_{2,i_2},t_j), \quad \hbar_\alpha = \begin{cases} 0.5h_\alpha, & i = 0,N_\alpha, \\ h_\alpha, & 1 < i < N_\alpha. \end{cases}$$

Расчетные формулы для схемы с весами:

$$y_{i_1i_2}^{j+1} - \sigma\left[B_{1i_1i_2}\left(y_{i_1+1,i_2}^{j+1} - y_{i_1i_2}^{j+1} \right) - A_{1i_1i_2}\left(y_{i_1i_2}^{j+1} - y_{i_1-1,i_2}^{j+1} \right) + B_{2i_1i_2}\left(y_{i_1,i_2+1}^{j+1} - y_{i_1i_2}^{j+1} \right) - A_{2i_1i_2}\left(y_{i_1i_2}^{j+1} - y_{i_1,i_2-1}^{j+1} \right) \right] =$$

$$= 2y_{i_1i_2}^{j} - y_{i_1i_2}^{j-1} + (1-2\sigma)\left[B_{1i_1i_2}\left(y_{i_1+1,i_2}^{j} - y_{i_1i_2}^{j} \right) - A_{1i_1i_2}\left(y_{i_1i_2}^{j} - y_{i_1-1,i_2}^{j} \right) + B_{2i_1i_2}\left(y_{i_1,i_2+1}^{j} - y_{i_1i_2}^{j} \right) - A_{2i_1i_2}\left(y_{i_1i_2}^{j} - y_{i_1,i_2-1}^{j} \right) \right] +$$

$$+\sigma\left[B_{1i_1i_2}\left(y_{i_1+1,i_2}^{j-1} - y_{i_1i_2}^{j-1} \right) - A_{1i_1i_2}\left(y_{i_1i_2}^{j-1} - y_{i_1-1,i_2}^{j-1} \right) + B_{2i_1i_2}\left(y_{i_1,i_2+1}^{j-1} - y_{i_1i_2}^{j-1} \right) - A_{2i_1i_2}\left(y_{i_1i_2}^{j-1} - y_{i_1,i_2-1}^{j-1} \right) \right] +$$

$$+\sigma\tau^2 f_{i_1i_2}^{j+1} + (1-2\sigma)\tau^2 f_{i_1i_2}^{j} + \sigma\tau^2 f_{i_1i_2}^{j-1},$$

$$A_{1i_1i_2} = \gamma_1 k_{i_1-1/2,i_2}, \quad B_{1i_1i_2} = \gamma_1 k_{i_1+1/2,i_2}, \quad A_{2i_1i_2} = \gamma_2 k_{i_1,i_2-1/2}, \quad B_{2i_1i_2} = \gamma_2 k_{i_1,i_2+1/2}, \quad \gamma_1 = \frac{\tau^2}{\hbar_1 h_1}, \quad \gamma_2 = \frac{\tau^2}{\hbar_2 h_2}.$$

В этом случае можно ограничиться явной схемой:

$$y_{i_1i_2}^{j+1} = 2y_{i_1i_2}^{j} - y_{i_1i_2}^{j-1} + \left[B_{1i_1i_2}\left(y_{i_1+1,i_2}^{j} - y_{i_1i_2}^{j} \right) - A_{1i_1i_2}\left(y_{i_1i_2}^{j} - y_{i_1-1,i_2}^{j} \right) + B_{2i_1i_2}\left(y_{i_1,i_2+1}^{j} - y_{i_1i_2}^{j} \right) - A_{2i_1i_2}\left(y_{i_1i_2}^{j} - y_{i_1,i_2-1}^{j} \right) \right] + \tau^2 f_{i_1i_2}^{j},$$

$$y_{i_1i_2}^0 = 0, \quad y_{i_1i_2}^1 = 0, \quad y_{0,i_2}^{j+1} = 0, \quad y_{N_1,i_2}^{j+1} = 0, \quad y_{i_1,0}^{j+1} = g_{21}\left(t_{j+1}\right), \quad y_{i_1,N_2}^{j} = 0.$$

3. Параллельная реализация.

См. уравнение теплопроводности и уравнение Пуассона.

4. Реализация примеров.

Пример 1. Реализация явной схемы на решетке процессоров (ex17a.c).

```
#include <stdio.h> #include <stdlib.h> #include <string.h> #include <unistd.h>
#include <math.h> #include "mycom.h" #include "mynet.h" #include "myio.h"
int np, mp, nl, ier, lp; static int np1, np2, mp1, mp2;
int mp_l, mp_r, mp_b, mp_t; static char pname[MPI_MAX_PROCESSOR_NAME];
char vname[10] = "ex17a"; static char sname[20];
MPI_Status status; static union_t buf;
double tick, t1, t2, t3; static FILE *Fi = NULL; static FILE *Fo = NULL;
int n1, n2, ntp, ntm, ntv; static double a1, b1, a2, b2;
double x11, x12, x21, x22, k1, k2;
double x13, x14, x23, x24, r1, r2, q0, q1, q2;
double w0, w1, w2, tmax, epst; static double tv, gt, ymin, ymax;
double k(double x1, double x2); double k(double x1, double x2) {
  if ((x11<=x1) && (x1<=x12) && (x21<=x2) && (x2<=x22)) return k1; else return k2; }
double f(double x1, double x2, double t); double f(double x1, double x2, double t) {
  double s1 = (x1-x13) / r1; double s2 = (x2-x23) / r1;
  double s3 = (x1-x14) / r2; double s4 = (x2-x24) / r2;
  double s5 = w1 * t; double s6 = w2 * t;
  return q1*dexp(-s1*s1-s2*s2)*dsin(s5)+q2*dexp(-s3*s3-s4*s4)*dsin(s6);
}
double phi(double x1, double x2); double phi(double x1, double x2) { return 0; }
double psi(double x1, double x2); double psi(double x1, double x2) { return 0; }
double g11(double t); double g11(double t) { return 0.0; }
double g12(double t); double g12(double t) { return 0.0; }
double g21(double t); double g21(double t) { return q0*(1.0-dexp(-w0*t)); }
double g22(double t); double g22(double t) { return 0.0; }
int main(int argc, char *argv[])
{
  int m, i1, i2, j1, j2, i11, i12, i21, i22, nc1, nc2, nc1m, nc2m, nc12;
  double h1, h12, h2, h22, tau, tau2, gam1, gam2, s0, s1, s2, s3, s4, s5;
  double *xx1, *xx2, *aa1, *bb1, *aa2, *bb2, *yy0, *yy1, *yy2;
  double *ss_l, *rr_l, *ss_r, *rr_r, *ss_b, *rr_b, *ss_t, *rr_t;
  MyNetInit(&argc,&argv,&np,&mp,&nl,pname,&tick);
  fprintf(stderr,"Netsize: %d, process: %d, system: %s, tick=%12le\n",np,mp,pname,tick);
  sleep(1);
  sprintf(sname,"%s.p%02d",vname,mp);
  ier = fopen_m(&Fo,sname,"wt");
  if (ier!=0) mpierr("Protocol file not opened",1);
  if (mp==0) {
    sprintf(sname,"%s.d",vname);
    ier = fopen_m(&Fi,sname,"rt");
    if (ier!=0) mpierr("Data file not opened",2);
    fscanf(Fi,"a1=%le\n",&a1); fscanf(Fi,"b1=%le\n",&b1);
    fscanf(Fi,"a2=%le\n",&a2); fscanf(Fi,"b2=%le\n",&b2);
    fscanf(Fi,"x11=%le\n",&x11); fscanf(Fi,"x12=%le\n",&x12);
    fscanf(Fi,"x21=%le\n",&x21); fscanf(Fi,"x22=%le\n",&x22);
    fscanf(Fi,"k1=%le\n",&k1); fscanf(Fi,"k2=%le\n",&k2);
    fscanf(Fi,"x13=%le\n",&x13); fscanf(Fi,"x23=%le\n",&x23);
    fscanf(Fi,"x14=%le\n",&x14); fscanf(Fi,"x24=%le\n",&x24);
    fscanf(Fi,"r1=%le\n",&r1); fscanf(Fi,"r2=%le\n",&r2);
    fscanf(Fi,"q0=%le\n",&q0); fscanf(Fi,"q1=%le\n",&q1); fscanf(Fi,"q2=%le\n",&q2);
    fscanf(Fi,"w0=%le\n",&w0); fscanf(Fi,"w1=%le\n",&w1); fscanf(Fi,"w2=%le\n",&w2);
    fscanf(Fi,"tmax=%le\n",&tmax); fscanf(Fi,"epst=%le\n",&epst);
    fscanf(Fi,"n1=%d\n",&n1); fscanf(Fi,"n2=%d\n",&n2);
    fscanf(Fi,"ntp=%d\n",&ntp); fscanf(Fi,"ntm=%d\n",&ntm);
    fscanf(Fi,"lp=%d\n",&lp);
    fclose_m(&Fi);
    if (argc>1) sscanf(argv[1],"%d",&n1); if (argc>2) sscanf(argv[2],"%d",&n2);
    if (argc>3) sscanf(argv[3],"%d",&ntp); if (argc>4) sscanf(argv[4],"%d",&ntm);
  }
  if (np>1) {
    if (mp==0) {
      buf.ddata[0]  = a1; buf.ddata[1]  = b1; buf.ddata[2]  = a2; buf.ddata[3]  = b2;
      buf.ddata[4]  = x11; buf.ddata[5]  = x12; buf.ddata[6]  = x21; buf.ddata[7]  = x22;
      buf.ddata[8]  = k1; buf.ddata[9]  = k2;
      buf.ddata[10] = x13; buf.ddata[11] = x23; buf.ddata[12] = x14; buf.ddata[13] = x24;
```

```
      buf.ddata[14] = r1; buf.ddata[15] = r2; buf.ddata[16] = q0; buf.ddata[17] = q1;
      buf.ddata[18] = q2; buf.ddata[19] = w0; buf.ddata[20] = w1; buf.ddata[21] = w2;
      buf.ddata[22] = tmax; buf.ddata[23] = epst; buf.idata[48] = n1; buf.idata[49] = n2;
      buf.idata[50] = ntp; buf.idata[51] = ntm; buf.idata[52] = lp;
    }
    MPI_Bcast(buf.ddata,27,MPI_DOUBLE,0,MPI_COMM_WORLD);
    if (mp>0) {
      a1 = buf.ddata[0]; b1 = buf.ddata[1]; a2 = buf.ddata[2]; b2 = buf.ddata[3];
      x11 = buf.ddata[4]; x12 = buf.ddata[5]; x21 = buf.ddata[6]; x22 = buf.ddata[7];
      k1 = buf.ddata[8]; k2 = buf.ddata[9]; x13 = buf.ddata[10]; x23 = buf.ddata[11];
      x14 = buf.ddata[12]; x24 = buf.ddata[13]; r1 = buf.ddata[14]; r2 = buf.ddata[15];
      q0 = buf.ddata[16]; q1 = buf.ddata[17]; q2 = buf.ddata[18]; w0 = buf.ddata[19];
      w1 = buf.ddata[20]; w2 = buf.ddata[21]; tmax = buf.ddata[22]; epst = buf.ddata[23];
      n1 = buf.idata[48]; n2 = buf.idata[49]; ntp = buf.idata[50]; ntm = buf.idata[51];
      lp = buf.idata[52];
    }
  }
  fprintf(Fo,"Netsize: %d, process: %d, system: %s, tick=%12le\n",np,mp,pname,tick);
  fprintf(Fo,"a1=%le b1=%le a2=%le b2=%le\n",a1,b1,a2,b2);
  fprintf(Fo,"x11=%le x12=%le x21=%le x22=%le k1=%le k2=%le\n",x11,x12,x21,x22,k1,k2);
  fprintf(Fo,"x13=%le x23=%le r1=%le q1=%le w1=%le\n",x13,x23,r1,q1,w1);
  fprintf(Fo,"x14=%le x24=%le r2=%le q2=%le w2=%le\n",x14,x24,r2,q2,w2);
  fprintf(Fo,"q0=%le w0=%le\n",q0,w0);
  fprintf(Fo,"tmax=%le epst=%le\n",tmax,epst);
  fprintf(Fo,"n1=%d n2=%d ntp=%d ntm=%d lp=%d\n",n1,n2,ntp,ntm,lp);
  t1 = MPI_Wtime();
  h1 = (b1-a1)/n1; h12 = h1 * h1; h2 = (b2-a2)/n2; h22 = h2 * h2;
  tau = 0.5 * dmin(h1,h2) / sqrt(dmax(k1,k2));
  tau = dmin(tau,1.0/dabs(q1)); tau = dmin(tau,1.0/dabs(q2)); tau2 = tau * tau;
  gam1 = tau2 / h12; gam2 = tau2 / h22;
  s0 = dmin(tmax/tau,1000000000.0); ntm = imin(ntm,(int)s0);
  fprintf(Fo,"h1=%le h2=%le tau=%le ntm=%d\n",h1,h2,tau,ntm);
  My2DGrid(np,mp,n1,n2,&np1,&np2,&mp1,&mp2);
//
// mp = np1 * mp2 + mp1
//
  if (mp1 ==     0) mp_l = -1; else mp_l = mp - 1;
  if (mp1 == np1-1) mp_r = -1; else mp_r = mp + 1;
  if (mp2 ==     0) mp_b = -1; else mp_b = mp - np1;
  if (mp2 == np2-1) mp_t = -1; else mp_t = mp + np1;
  MyRange(np1,mp1,0,n1,&i11,&i12,&nc1); nc1m = nc1-1;
  MyRange(np2,mp2,0,n2,&i21,&i22,&nc2); nc2m = nc2-1;
  nc12 = nc1 * nc2;
  fprintf(Fo,"Grid=%dx%d coord=(%d,%d)\n",np1,np2,mp1,mp2);
  fprintf(Fo,"i11=%d i12=%d nc1=%d\n",i11,i12,nc1);
  fprintf(Fo,"i21=%d i22=%d nc2=%d\n",i11,i12,nc2);
  if (mp == 0) {
    fprintf(stderr,"n1=%d n2=%d h1=%le h2=%le tau=%le ntm=%d\n",n1,n2,h1,h2,tau,ntm);
    fprintf(stderr,"Grid=%dx%d\n",np1,np2);
  }
  xx1 = (double*)(malloc(sizeof(double)*nc1));
  xx2 = (double*)(malloc(sizeof(double)*nc2));
  yy0 = (double*)(malloc(sizeof(double)*nc12));
  yy1 = (double*)(malloc(sizeof(double)*nc12));
  yy2 = (double*)(malloc(sizeof(double)*nc12));
  aa1 = (double*)(malloc(sizeof(double)*nc12));
  bb1 = (double*)(malloc(sizeof(double)*nc12));
  aa2 = (double*)(malloc(sizeof(double)*nc12));
  bb2 = (double*)(malloc(sizeof(double)*nc12));
  if (mp_l>=0) {
    rr_l = (double*)(malloc(sizeof(double)*nc2));
    ss_l = (double*)(malloc(sizeof(double)*nc2));
  }
  if (mp_r>=0) {
    rr_r = (double*)(malloc(sizeof(double)*nc2));
    ss_r = (double*)(malloc(sizeof(double)*nc2));
  }
  if (mp_b>=0) {
```

```
    rr_b = (double*)(malloc(sizeof(double)*nc1));
    ss_b = (double*)(malloc(sizeof(double)*nc1));
  }
  if (mp_t>=0) {
    rr_t = (double*)(malloc(sizeof(double)*nc1));
    ss_t = (double*)(malloc(sizeof(double)*nc1));
  }
  for (i1=0; i1<nc1; i1++) xx1[i1] = a1 + h1 * (i11 + i1); // grid for x1
  for (i2=0; i2<nc2; i2++) xx2[i2] = a2 + h2 * (i21 + i2); // grid for x2
// Coefficients:
  for (i2=0; i2<nc2; i2++) {
    j2 = i21 + i2;
    for (i1=0; i1<nc1; i1++) {
      j1 = i11 + i1;
      m = nc1 * i2 + i1;
      if ((j1==0) || (j1==n1)) {
        aa1[m] = 0.0; bb1[m] = 0.0;
      }
      else {
        s0 = k(xx1[i1],xx2[i2]); s1 = k(xx1[i1]-h1,xx2[i2]); s2 = k(xx1[i1]+h1,xx2[i2]);
        aa1[m] = gam1 * 2.0 * s0 * s1 / (s0 + s1);
        bb1[m] = gam1 * 2.0 * s0 * s2 / (s0 + s2);
      }
      if ((j2==0) || (j2==n2)) {
        aa2[m] = 0.0; bb2[m] = 0.0;
      }
      else {
        s0 = k(xx1[i1],xx2[i2]); s1 = k(xx1[i1],xx2[i2]-h2); s2 = k(xx1[i1],xx2[i2]+h2);
        aa2[m] = gam2 * 2.0 * s0 * s1 / (s0 + s1);
        bb2[m] = gam2 * 2.0 * s0 * s2 / (s0 + s2);
      }
    }
  }
// Initial values:
  ntv = 1;
  tv = tau;
  gt = 1.0;
  for (i2=0; i2<nc2; i2++)
    for (i1=0; i1<nc1; i1++) {
      m = nc1 * i2 + i1;
      yy1[m] = phi(xx1[i1],xx2[i2]); yy2[m] = yy1[m] + tau * psi(xx1[i1],xx2[i2]);
    }
// Time loop:
  do {
    ntv++;
    tv += tau;
    // update solution:
    for (m=0; m<nc12; m++) yy0[m] = yy1[m];
    for (m=0; m<nc12; m++) yy1[m] = yy2[m];
    // data exchange:
    if (np>1) {
      if (mp_l>=0) {
        i1 = 0; for (i2=0; i2<nc2; i2++) { m = nc1 * i2 + i1; ss_l[i2] = yy1[m]; }
      }
      if (mp_r>=0) {
        i1 = nc1m; for (i2=0; i2<nc2; i2++) { m = nc1 * i2 + i1; ss_r[i2] = yy1[m]; }
      }
      if (mp_b>=0) {
        i2 = 0; for (i1=0; i1<nc1; i1++) { m = nc1 * i2 + i1; ss_b[i1] = yy1[m]; }
      }
      if (mp_t>=0) {
        i2 = nc2m; for (i1=0; i1<nc1; i1++) { m = nc1 * i2 + i1; ss_t[i1] = yy1[m]; }
      }
      BndAExch2D(mp_l,nc2,ss_l,rr_l,
                 mp_r,nc2,ss_r,rr_r,
                 mp_b,nc1,ss_b,rr_b,
                 mp_t,nc1,ss_t,rr_t);
    }
```

```
    // computation of new solution:
    for (i2=0; i2<nc2; i2++) {
      j2 = i21 + i2;
      for (i1=0; i1<nc1; i1++) {
        j1 = i11 + i1;
        m = nc1 * i2 + i1;
        s0 = 2.0 * yy1[m] - yy0[m];
        if      (j1== 0) yy2[m] = g11(tv);
        else if (j1==n1) yy2[m] = g12(tv);
        else if (j2== 0) yy2[m] = g21(tv);
        else if (j2==n2) yy2[m] = g22(tv);
        else {
          if (i1==0)    s1 = aa1[m] * (yy1[m] - rr_l[i2]);
          else          s1 = aa1[m] * (yy1[m] - yy1[m-1]);
          if (i1==nc1m) s2 = bb1[m] * (rr_r[i2] - yy1[m]);
          else          s2 = bb1[m] * (yy1[m+1] - yy1[m]);
          if (i2==0)    s3 = aa2[m] * (yy1[m] - rr_b[i1]);
          else          s3 = aa2[m] * (yy1[m] - yy1[m-nc1]);
          if (i2==nc2m) s4 = bb2[m] * (rr_t[i1] - yy1[m]);
          else          s4 = bb2[m] * (yy1[m+nc1] - yy1[m]);
          s5 = tau2 * f(xx1[i1],xx2[i2],tv-tau);
          yy2[m] = s0 + s2 - s1 + s4 - s3 + s5;
        }
      }
    }
    // check output:
    if (ntv % ntp == 0) {
      gt = 0.0; ymin = yy2[0]; ymax = yy2[0];
      for (m=0; m<nc12; m++) {
        s0 = dabs(yy2[m]-yy1[m]); s1 = dabs(yy1[m]);
        if (s1>1e-15) s0 = s0 / s1; else s0 = 0;
        gt = dmax(gt,s0); ymin = dmin(ymin,yy2[m]); ymax = dmax(ymax,yy2[m]);
      }
      gt = gt / tau;
      if (np>1) {
        buf.ddata[0] = gt; buf.ddata[1] = - ymin; buf.ddata[2] = ymax;
        MPI_Allreduce(buf.ddata,buf.ddata+3,3,MPI_DOUBLE,MPI_MAX,MPI_COMM_WORLD);
        gt = buf.ddata[3]; ymin = - buf.ddata[4]; ymax = buf.ddata[5];
      }
      if (mp == 0) {
        t2 = MPI_Wtime() - t1;
        fprintf(stderr,"ntv=%d tv=%le gt=%le ymin=%le ymax=%le tcpu=%le\n",
          ntv,tv,gt,ymin,ymax,t2);
      }
    }
    if (lp>0) {
      fprintf(Fo,"ntv=%d tv=%le gt=%le\n",ntv,tv,gt);
      for (i2=0; i2<nc2; i2++) {
        j2 = i21 + i2;
        for (i1=0; i1<nc1; i1++) {
          j1 = i11 + i1;
          m = nc1 * i2 + i1;
          fprintf(Fo,"i1=%8d i2=%8d x1=%12le x2=%12le y2=%12le\n",
            j1,j2,xx1[i1],xx2[i2],yy2[m]);
        }
      }
    }
  } while (ntv<ntm);
  t1 = MPI_Wtime() - t1;
  sprintf(sname,"%s_%02d.dat",vname,np);
  OutFun2DP(sname,np,mp,nc1,nc2,xx1,xx2,yy2);
  fprintf(Fo,"ntv=%d tv=%le gt=%le time=%le\n",ntv,tv,gt,t1);
  if (mp == 0)
    fprintf(stderr,"Grid=%dx%d n1=%d n2=%d ntv=%d tv=%le gt=%le ymin=%le ymax=%le
tcpu=%le\n",np1,np2,n1,n2,ntv,tv,gt,ymin,ymax,t1);
  ier = fclose_m(&Fo);
  MPI_Finalize();
  return 0;}
```

Компиляция:

```
>mpicc -o ex17a.px -O2 -lm -static ex17a.c mycom.c mynet.c myio.c
```

Результаты:

```
>mpirun -np <1-16> -nolocal -machinefile hosts ex17a.px
1x1 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=8.317579e+01
2x2 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=2.967879e+01
3x3 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=1.745440e+01
4x4 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=1.407800e+01
5x5 n1=100 n2=100 ntv=20000 tv=1e+02 ymin=-3.985695e-01 ymax=3.981306e-01 tcpu=1.373613e+01
```