Семинар 11. Решение двумерного уравнения Пуассона прямыми методами.

1. Постановка задачи.

$$\frac{\partial}{\partial x_1}\left(k_1(x_1,x_2)\frac{\partial u}{\partial x_1}\right)+\frac{\partial}{\partial x_2}\left(k_2(x_1,x_2)\frac{\partial u}{\partial x_2}\right)=-f(x_1,x_2),\quad (x_1,x_2)\in D=(0,1)\times(0,1),$$

$$u(x_1,x_2)=g(x_1,x_2),\quad (x_1,x_2)\in\partial D.$$

2. Конечно-разностная схема.

Равномерная сетка: $\Omega=\omega_{x_1}\times\omega_{x_2}$, $\omega_{x_\alpha}=\left\{x_{\alpha,i_\alpha}=i_\alpha h_\alpha,\, i_\alpha=0,\dots,N_\alpha,\, h_\alpha=1/N_\alpha\right\}$, $\alpha=1,2.$

Разностная схема:

$$\frac{1}{\hbar_1}\left\{k_{1,i_1+1/2,i_2}\frac{y_{i_1+1,i_2}-y_{i_1,i_2}}{h_1}-k_{1,i_1-1/2,i_2}\frac{y_{i_1,i_2}-y_{i_1-1,i_2}}{h_1}\right\}+$$

$$+\frac{1}{\hbar_2}\left\{k_{2,i_1,i_2+1/2}\frac{y_{i_1,i_2+1}-y_{i_1,i_2}}{h_2}-k_{2,i_1,i_2-1/2}\frac{y_{i_1,i_2}-y_{i_1,i_2-1}}{h_2}\right\}=-f_{i_1,i_2},\quad i_\alpha=1,\dots,N_\alpha-1,$$

$$y_{i_1,i_2}=g_{i_1,i_2},\quad i_\alpha=0,N_\alpha,\quad \alpha=1,2.$$

Эквивалентные уравнения:

$$\left(\frac{\hbar_2}{h_1}\left(k_{1,i_1+1/2,i_2}+k_{1,i_1-1/2,i_2}\right)+\frac{\hbar_1}{h_2}\left(k_{2,i_1,i_2+1/2}+k_{2,i_1,i_2-1/2}\right)\right)y_{i_1,i_2}-\frac{\hbar_2}{h_1}k_{1,i_1+1/2,i_2}y_{i_1+1,i_2}-\frac{\hbar_2}{h_1}k_{1,i_1-1/2,i_2}y_{i_1-1,i_2}-$$

$$-\frac{\hbar_1}{h_2}k_{2,i_1,i_2+1/2}y_{i_1,i_2+1}-\frac{\hbar_1}{h_2}k_{2,i_1,i_2-1/2}y_{i_1,i_2-1}=\hbar_1\hbar_2 f_{i_1,i_2},\quad i_\alpha=1,\dots,N_\alpha-1,$$

$$y_{i_1,i_2}=g_{i_1,i_2},\quad i_\alpha=0,N_\alpha,\quad \alpha=1,2.$$

3. Решение уравнения Пуассона с помощью дискретного преобразования Фурье (Пример 1).

Постановка задачи:

$$k_1(x_1,x_2)\equiv 1,\quad k_2(x_1,x_2)\equiv 1,\quad f(x_1,x_2)=2\pi^2\sin(\pi x_1)\sin(\pi x_2),$$

$$g(x_1,x_2)\equiv 0,\quad u(x_1,x_2)=\sin(\pi x_1)\sin(\pi x_2).$$

Эквивалентные уравнения:

$$2\left(\frac{h_2}{h_1}+\frac{h_1}{h_2}\right)y_{i_1,i_2}-\frac{h_2}{h_1}y_{i_1+1,i_2}-\frac{h_2}{h_1}y_{i_1-1,i_2}-\frac{h_1}{h_2}y_{i_1,i_2+1}-\frac{h_1}{h_2}y_{i_1,i_2-1}=h_1h_2 f_{i_1,i_2},\quad i_\alpha=1,\dots,N_\alpha-1,$$

$$y_{i_1,i_2}=0,\quad i_\alpha=0,N_\alpha,\quad \alpha=1,2.$$

Дискретное преобразование Фурье (ДПФ) по обеим координатам:

$$y_{i_1,i_2}=\sum_{k_1=1,k_2=1}^{N_1-1,N_2-1}c_{k_1k_2}\sqrt{2}\sin(\pi k_1x_{1,i_1})\sqrt{2}\sin(\pi k_2x_{2,i_2}),\quad f_{i_1,i_2}=\sum_{k_1=1,k_2=1}^{N_1-1,N_2-1}\varphi_{k_1k_2}\sqrt{2}\sin(\pi k_1x_{1,i_1})\sqrt{2}\sin(\pi k_2x_{2,i_2}),$$

$$c_{k_1k_2}=\frac{\varphi_{k_1k_2}}{\lambda_{k_1k_2}},\quad \lambda_{k_1k_2}=\frac{4}{h_1^2}\left(\sin\frac{\pi k_1h_1}{2}\right)^2+\frac{4}{h_2^2}\left(\sin\frac{\pi k_2h_2}{2}\right)^2=\frac{2}{h_1^2}\left(1-\cos(\pi k_1h_1)\right)+\frac{2}{h_2^2}\left(1-\cos(\pi k_2h_2)\right).$$

$$\varphi_{k_1k_2}=\sum_{i_1=1,i_2=1}^{N_1-1,N_2-1}f(x_{i_1},x_{i_2})2\sin(\pi k_1x_{1,i_1})\sin(\pi k_2x_{2,i_2})\,h_1h_2.$$

В итоге решение можно получить по формулам:

$$\gamma_{k_1k_2}=\frac{2\varphi_{k_1k_2}}{\lambda_{k_1k_2}}=\frac{2}{N_1N_2\left[N_1^2\left(1-\cos\frac{\pi k_1}{N_1}\right)+N_2^2\left(1-\cos\frac{\pi k_2}{N_2}\right)\right]}\sum_{i_1=1,i_2=1}^{N_1-1,N_2-1}f(x_{i_1},x_{i_2})\sin\left(\frac{\pi k_1i_1}{N_1}\right)\sin\left(\frac{\pi k_2i_2}{N_2}\right),$$

$$y_{i_1,i_2}=\sum_{k_1=1,k_2=1}^{N_1-1,N_2-1}\gamma_{k_1k_2}\sin\left(\frac{\pi k_1i_1}{N_1}\right)\sin\left(\frac{\pi k_2i_2}{N_2}\right).$$

Распараллеливание можно проводить по обоим индексам на решетке процессоров.

Если $N_\alpha=2^{m_\alpha}$, то можно применить быстрое дискретное преобразование Фурье (БДПФ).

Оптимизация вычислений:

$$\sin\left(\frac{\pi ki}{N}\right)=\begin{cases} +\sin\left(\dfrac{\pi m}{N}\right), & \text{если} \quad (2ki/N)\bmod 4=0, \quad m=(ki)\bmod(N/2), \\ +\sin\left(\dfrac{\pi m}{N}\right), & \text{если} \quad (2ki/N)\bmod 4=1, \quad m=(N/2)-(ki)\bmod(N/2), \\ -\sin\left(\dfrac{\pi m}{N}\right), & \text{если} \quad (2ki/N)\bmod 4=2, \quad m=(ki)\bmod(N/2), \\ -\sin\left(\dfrac{\pi m}{N}\right), & \text{если} \quad (2ki/N)\bmod 4=3, \quad m=(N/2)-(ki)\bmod(N/2). \end{cases}$$

В итоге получим оптимизированное ДПФ, которое легко распараллеливается.

4. Решение уравнения Пуассона комбинированным методом ДПФ-ЛГ (Пример 2).

Постановка задачи:

$$k_1(x_1,x_2)\equiv k(x_1)=1+(x_1-0.5)^2, \quad k_2(x_1,x_2)\equiv 1,$$

$$f(x_1,x_2)=-2(x_1-0.5)\pi\cos(\pi x_1)\sin(\pi x_2)+2\pi^2\sin(\pi x_1)\sin(\pi x_2),$$

$$g(x_1,x_2)\equiv 0, \quad u(x_1,x_2)=\sin(\pi x_1)\sin(\pi x_2).$$

Эквивалентные уравнения:

$$\left(\frac{h_2}{h_1}\left(k_{i_1+1/2}+k_{i_1-1/2}\right)+2\frac{h_1}{h_2}\right)y_{i_1,i_2}-\frac{h_2}{h_1}k_{i_1+1/2}y_{i_1+1,i_2}-\frac{h_2}{h_1}k_{i_1-1/2}y_{i_1-1,i_2}-$$

$$-\frac{h_1}{h_2}y_{i_1,i_2+1}-\frac{h_1}{h_2}y_{i_1,i_2-1}=h_1h_2f_{i_1,i_2}, \quad i_\alpha=1,...,N_\alpha-1, \quad y_{i_1,i_2}=0, \quad i_\alpha=0,N_\alpha, \quad \alpha=1,2.$$

Представление решения:

$$y_{i_1,i_2}=\sum_{k_2=1}^{N_2-1}c_{i_1k_2}\sqrt{2}\sin\left(\pi k_2x_{2,i_2}\right), \quad f_{i_1,i_2}=\sum_{k_2=1}^{N_2-1}\varphi_{i_1k_2}\sqrt{2}\sin(\pi k_2x_{2,i_2}), \quad i_\alpha=0,...,N_\alpha.$$

Задача для коэффициентов:

$$\frac{h_2}{h_1}\left(k_{i_1+1/2}+k_{i_1-1/2}\right)c_{i_1,k_2}-\frac{h_2}{h_1}k_{i_1+1/2}c_{i_1+1,i_2}-\frac{h_2}{h_1}k_{i_1-1/2}c_{i_1-1,i_2}+h_1h_2\lambda_{k_2}c_{i_1,k_2}=h_1h_2\varphi_{i_1,k_2},$$

$$i_\alpha=1,...,N_\alpha-1, \quad \alpha=1,2, \quad \lambda_{k_2}=\frac{4}{h_2^2}\left(\sin\frac{\pi k_2h_2}{2}\right)^2=\frac{2}{h_2^2}\left(1-\cos(\pi k_2h_2)\right),$$

$$c_{0,k_2}=0, \quad c_{N_1,k_2}=0, \quad k_2=1,...,N_2-1.$$

Для любого k_2 коэффициенты c_{i_1,k_2} находим методом прогонки.

5. Реализация примеров.

Пример 1. Решение уравнения Пуассона методом Фурье на решетке процессоров (ex15a.c).

```
#include <stdio.h> #include <stdlib.h> #include <string.h> #include <math.h>
#include "mycom.h" #include "mynet.h" #include "myio.h"
int np, mp, nl, ier, lp;
int np1, np2, mp1, mp2;
char pname[MPI_MAX_PROCESSOR_NAME];
char vname[10] = "ex15a"; char sname[20];
MPI_Status status;
union_t buf;
double tick, t1, t2, t3;
FILE *Fi = NULL;
FILE *Fo = NULL;
int n1, n2;
double pi, pi2;
double mysin(double x); double mysin(double x) {
  double s = sin(x); if (dabs(s)<1e-15) return s; else return 0.0; }
double f(double x1, double x2); double f(double x1, double x2) {
  return pi2*mysin(pi*x1)*mysin(pi*x2); }
double u(double x1, double x2); double u(double x1, double x2) {
  return mysin(pi*x1)*mysin(pi*x2); }
```

```
int main(int argc, char *argv[])
{
  int i1, i2, k1, k2, n1p, n2p, n12p;
  int n, m, i11, i12, i21, i22, nc1, nc2, nc1m, nc2m, nc12;
  double h1, h2, h12, h22, h0, s0, s1, s2; double *xx1, *xx2, *cv1, *cv2, *gg, *yy;
  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);
  fprintf(Fo,"Netsize: %d, process: %d, system: %s, tick=%12le\n",np,mp,pname,tick);
  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,"n1=%d\n",&n1); fscanf(Fi,"n2=%d\n",&n2); 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",&lp);
  }
  if (np>1) {
    if (mp==0) { buf.idata[0] = n1; buf.idata[1] = n2; buf.idata[2] = lp; }
    MPI_Bcast(buf.ddata,2,MPI_DOUBLE,0,MPI_COMM_WORLD);
    if (mp>0) { n1 = buf.idata[0]; n2 = buf.idata[1]; lp = buf.idata[2]; }
  }
  My2DGrid(np,mp,n1,n2,&np1,&np2,&mp1,&mp2);
  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; n1p = n1+1; n2p = n2+1; n12p = n1p*n2p;
  h1 = 1.0/n1; h2 = 1.0/n2; h12 = 1.0/(h1*h1); h22 = 1.0/(h2*h2); h0 = 2.0*h1*h2;
  pi = M_PI; pi2 = 2.0*pi*pi;
  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);
  fprintf(Fo,"n1=%d n2=%d h1=%le h2=%le\n",n1,n2,h1,h2);
  if (mp == 0) {
    fprintf(stderr,"n1=%d n2=%d h1=%le h2=%le\n",n1,n2,h1,h2);
    fprintf(stderr,"Grid=%dx%d\n",np1,np2);
  }
  t1 = MPI_Wtime();
  xx1 = (double*)(malloc(sizeof(double)*n1p));
  xx2 = (double*)(malloc(sizeof(double)*n2p));
  cv1 = (double*)(malloc(sizeof(double)*n1p));
  cv2 = (double*)(malloc(sizeof(double)*n2p));
  gg = (double*)(malloc(sizeof(double)*n12p));
  yy = (double*)(malloc(sizeof(double)*n12p));
  for (i1=0; i1<=n1; i1++) xx1[i1] = h1 * i1;
  for (i2=0; i2<=n2; i2++) xx2[i2] = h2 * i2;
  for (i1=0; i1<=n1; i1++) cv1[i1] = 1.0 - cos(pi*i1*h1);
  for (i2=0; i2<=n2; i2++) cv2[i2] = 1.0 - cos(pi*i2*h2);
// step 1:
  if (np<2) {
    for (i2=0; i2<=n2; i2++) {
      for (i1=0; i1<=n1; i1++) {
        m = n1p * i2 + i1; yy[m] = f(xx1[i1],xx2[i2]);
      }
    }
  }
  else {
    for (i2=0; i2<=n2; i2++) {
      for (i1=0; i1<=n1; i1++) {
        m = n1p * i2 + i1; s0 = 0.0;
        if ((i11<=i1) && (i1<=i12) && (i21<=i2) && (i2<=i22)) s0 = f(xx1[i1],xx2[i2]);
        gg[m] = s0;
      }
    }
    MPI_Allreduce(gg,yy,n12p,MPI_DOUBLE,MPI_SUM,MPI_COMM_WORLD);
  }
```

```
// step 2:
  if (np<2) {
    for (k2=0; k2<=n2; k2++) {
      for (k1=0; k1<=n1; k1++) {
        n = n1p * k2 + k1; s0 = 0.0;
        for (i2=1; i2<n2; i2++) {
          for (i1=1; i1<n1; i1++) {
            m = n1p * i2 + i1; s1 = mysin(pi*k1*xx1[i1]); s2 = mysin(pi*k2*xx2[i2]);
            s0 += yy[m]*s1*s2;
          }
        }
        s1 = h12*cv1[k1] + h22*cv2[k2]; s0 = s0 * h0 / s1; gg[n] = s0;
      }
    }
  }
  else {
    for (k2=0; k2<=n2; k2++) {
      for (k1=0; k1<=n1; k1++) {
        n = n1p * k2 + k1; s0 = 0.0;
        if ((i11<=k1) && (k1<=i12) && (i21<=k2) && (k2<=i22)) {
          for (i2=1; i2<n2; i2++) {
            for (i1=1; i1<n1; i1++) {
              m = n1p * i2 + i1; s1 = mysin(pi*k1*xx1[i1]); s2 = mysin(pi*k2*xx2[i2]);
              s0 += yy[m]*s1*s2;
            }
          }
          s1 = h12*cv1[k1] + h22*cv2[k2]; s0 = s0 * h0 / s1;
        }
        gg[n] = s0;
      }
    }
    MPI_Allreduce(gg,yy,n12p,MPI_DOUBLE,MPI_SUM,MPI_COMM_WORLD);
    for (n=0; n<n12p; n++) gg[n] = yy[n];
  }
// step 3:
  if (np<2) {
    for (i2=0; i2<=n2; i2++) {
      for (i1=0; i1<=n1; i1++) {
        m = n1p * i2 + i1; s0 = 0.0;
        for (k2=1; k2<n2; k2++) {
          for (k1=1; k1<n1; k1++) {
            n = n1p * k2 + k1; s1 = mysin(pi*k1*xx1[i1]); s2 = mysin(pi*k2*xx2[i2]);
            s0 += gg[n]*s1*s2;
          }
        }
        yy[m] = s0;
      }
    }
  }
  else {
    for (i2=0; i2<=n2; i2++) {
      for (i1=0; i1<=n1; i1++) {
        m = n1p * i2 + i1; s0 = 0.0;
        if ((i11<=i1) && (i1<=i12) && (i21<=i2) && (i2<=i22)) {
          for (k2=1; k2<n2; k2++) {
            for (k1=1; k1<n1; k1++) {
              n = n1p * k2 + k1; s1 = mysin(pi*k1*xx1[i1]); s2 = mysin(pi*k2*xx2[i2]);
              s0 += gg[n]*s1*s2;
            }
          }
        }
        yy[m] = s0;
      }
    }
    MPI_Allreduce(yy,gg,n12p,MPI_DOUBLE,MPI_SUM,MPI_COMM_WORLD);
    for (n=0; n<n12p; n++) yy[n] = gg[n];
  }
// fine:
```

```
  t1 = MPI_Wtime() - t1; s2 = 0.0;
  for (i2=0; i2<=n2; i2++) {
    for (i1=0; i1<=n1; i1++) {
      m = n1p * i2 + i1; s0 = u(xx1[i1],xx2[i2]); s1 = dabs(yy[m]-s0); s2 = dmax(s2,s1);
      if (lp>0) fprintf(Fo,"x1=%le x2=%le y=%le u=%le
d=%le\n",xx1[i1],xx2[i2],yy[m],s0,s1);
    }
  }
  fprintf(Fo,"t1=%le dmax=%le\n",t1,s2);
  if (mp==0)
    fprintf(stderr,"np=%d (%dx%d) n1=%8d n2=%8d dmax=%le
time=%le\n",np,np1,np2,n1,n2,s2,t1);
  if (mp==0 && lp>0) {
    sprintf(sname,"%s_%02d.dat",vname,np); OutFun2D(sname,0,n1p,n2p,xx1,xx2,yy);
  }
  MPI_Finalize();
  return 0;
}
```

Трансляция:

```
>mpicc -o ex15a.px -O2 ex15a.c mycom.c mynet.c myio.c -lm
```

Результаты расчетов:

```
>mpirun -np <1-16> -nolocal -machinefile hosts ex15a.px 128 128  (запуск в классе)
>mpirun -np <1-16> ex15a.px 128 128                               (запуск на сервере)
np= 1  (1x1) n1=     128 n2=     128 dmax=1.499661e-32 time=7.579066e+01
np= 2  (1x2) n1=     128 n2=     128 dmax=1.499661e-32 time=3.824831e+01
np= 3  (1x3) n1=     128 n2=     128 dmax=1.499661e-32 time=2.548009e+01
np= 4  (2x2) n1=     128 n2=     128 dmax=1.499661e-32 time=1.936292e+01
np= 5  (1x5) n1=     128 n2=     128 dmax=1.499661e-32 time=1.556806e+01
np= 6  (2x3) n1=     128 n2=     128 dmax=1.499661e-32 time=1.332567e+01
np= 7  (1x7) n1=     128 n2=     128 dmax=1.499661e-32 time=1.161343e+01
np= 8  (2x4) n1=     128 n2=     128 dmax=1.499661e-32 time=9.934915e+00
np= 9  (3x3) n1=     128 n2=     128 dmax=1.499661e-32 time=9.327514e+00
np=10  (2x5) n1=     128 n2=     128 dmax=1.499661e-32 time=8.118166e+00
np=11 (1x11) n1=     128 n2=     128 dmax=1.499661e-32 time=7.635249e+00
np=12  (3x4) n1=     128 n2=     128 dmax=1.499661e-32 time=7.074516e+00
np=13 (1x13) n1=     128 n2=     128 dmax=1.499661e-32 time=6.776237e+00
np=14  (2x7) n1=     128 n2=     128 dmax=1.499661e-32 time=6.199461e+00
np=15  (3x5) n1=     128 n2=     128 dmax=1.499661e-32 time=5.503648e+00
np=16  (4x4) n1=     128 n2=     128 dmax=1.499661e-32 time=5.311533e+00
```