Семинар 7. Решение линейных пространственно одномерных краевых задач.

1. Постановка задачи.

$$\frac{d}{dx}\left(k(x)\frac{du}{dx}\right)-q(x)u=-f(x),\quad a<x<b.$$

Тестовый пример 1.

Коэффициенты:

$$k(x)=1+\left(\frac{x-a}{b-a}\right)^2,\quad q(x)=1+\left(\frac{b-x}{b-a}\right)^2,\quad f(x)=q(x)u(x)-k'(x)u'(x)-k(x)u''(x).$$

Граничные условия 1-го рода:

$$u(a)=u_a,\quad u(b)=u_b.$$

Тестовая функция:

$$u(x)=u_a\cos\left(\frac{\pi(x-a)}{2(b-a)}\right)+u_b\sin\left(\frac{\pi(x-a)}{2(b-a)}\right).$$

Тестовый пример 2.

Коэффициенты: те же.

Граничные условия 2-го рода:

$$u'(a)=u_a,\quad u'(b)=u_b.$$

Тестовая функция:

$$u(x)=\frac{2(b-a)}{\pi}\left[u_a\sin\left(\frac{\pi(x-a)}{2(b-a)}\right)-u_b\cos\left(\frac{\pi(x-a)}{2(b-a)}\right)\right].$$

Тестовый пример 3.

Коэффициенты: те же.

Граничные условия 3-го рода:

$$u'(a)=+\eta\left(u-1\right),\quad u'(b)=-\eta\left(u-1\right).$$

Тестовая функция:

$$u(x)=\frac{1}{2}\left[\exp\left(-\eta\left(x-a\right)\right)+\exp\left(-\eta\left(b-x\right)\right)\right],\quad u'(x)=\frac{1}{2}\eta\left[-\exp\left(-\eta\left(x-a\right)\right)+\exp\left(-\eta\left(b-x\right)\right)\right].$$

Тестовый пример 4.

Коэффициенты:

$$k(x)=1+\left[\sin\left(\frac{\pi\left(x-a\right)}{(b-a)}\right)\right]^2,\quad q(x)=1+\left[\cos\left(\frac{\pi\left(x-a\right)}{(b-a)}\right)\right]^2,$$

$$f(x)=q(x)u(x)-k'(x)u'(x)-k(x)u''(x).$$

Периодические граничные условия:

$$u(a)=u(b),\quad u'(a)=u'(b).$$

Тестовая функция:

$$u(x)=\sin\left(\frac{\pi\left(x-a\right)}{(b-a)}\right),\quad u'(x)=\frac{\pi}{(b-a)}\cos\left(\frac{\pi\left(x-a\right)}{(b-a)}\right).$$

Тестовый пример 5.

Коэффициенты:

$$k(x)=1+\exp\left(-\frac{x-a}{b-a}\right),\quad q(x)=1+\exp\left(\frac{b-x}{b-a}\right),$$

$$f(x)=q(x)u(x)-k'(x)u'(x)-k(x)u''(x).$$

Интегральная граничные условия:

$$\int_a^b u(x)dx\equiv\int_a^b u(x)\rho_0(x)dx=\left(b-a\right)\left[\mathrm{ch}(1)-\mathrm{ch}(0)\right]\equiv c_0,$$

$$\int_a^b u(x)(x-a)dx\equiv\int_a^b u(x)\rho_1(x)dx=\left(b-a\right)^2\left[\mathrm{ch}\left(1\right)-\mathrm{sh}\left(1\right)\right]\equiv c_1.$$

Тестовая функция:

$$u(x)=\mathrm{sh}\left(\frac{(x-a)}{(b-a)}\right),\quad u'(x)=\frac{1}{(b-a)}\mathrm{ch}\left(\frac{(x-a)}{(b-a)}\right).$$

2. Численный алгоритм.

Разностная схема на равномерной сетке $\omega_x=\left\{x_i=a+ih_x,\, i=0,...,N_x,\, h_x=\dfrac{b-a}{N_x}\right\}$:

$$\frac{1}{\hbar_x}\left\{k_{i+1/2}\frac{y_{i+1}-y_i}{h_x}-k_{i-1/2}\frac{y_i-y_{i-1}}{h_x}\right\}-q_iy_i=-f_i,\quad 1\le i\le N_x-1;$$

$$k_{i\pm1/2}=\frac{k_i+k_{i\pm1}}{2},\quad k_i=k(x_i),\quad q_i=q(x_i),\quad f_i=f(x_i),\quad \hbar_x=\begin{cases}0.5h_x, & i=0,N_x,\\ h_x, & 1<i<N_x.\end{cases}$$

Разностные уравнения на границе для теста 1:

$$y_0=u_a,\quad y_{N_x}=u_b.$$

Разностные уравнения на границе для теста 2:

$$\frac{2}{h_x}\left\{k_{1/2}\frac{y_1-y_0}{h_x}-k_0u_a\right\}-q_0y_0=-f_0;$$

$$\frac{2}{h_x}\left\{k_{N_x}u_b-k_{N_x-1/2}\frac{y_{N_x}-y_{N_x-1}}{h_x}\right\}-q_{N_x}y_{N_x}=-f_{N_x}.$$

Разностные уравнения на границе для теста 3:

$$\frac{2}{h_x}\left\{k_{1/2}\frac{y_1-y_0}{h_x}-k_0\eta\left(y_0-1\right)\right\}-q_0y_0=-f_0;$$

$$\frac{2}{h_x}\left\{-k_{N_x}\eta\left(y_{N_x}-1\right)-k_{N_x-1/2}\frac{y_{N_x}-y_{N_x-1}}{h_x}\right\}-q_{N_x}y_{N_x}=-f_{N_x}.$$

Разностные уравнения для теста 4:

$$\frac{1}{h_x}\left\{k_{i+1/2}\frac{y_{i+1}-y_i}{h_x}-k_{i-1/2}\frac{y_i-y_{i-1}}{h_x}\right\}-q_iy_i=-f_i,\quad 1\le i\le N_x-2;$$

$$k_{i\pm1/2}=\frac{k_i+k_{i\pm1}}{2},\quad k_i=k(x_i),\quad q_i=q(x_i),\quad f_i=f(x_i).$$

Уравнения на границах:

$$\frac{1}{h_x}\left\{k_{1/2}\frac{y_1-y_0}{h_x}-k_{N_x-1/2}\frac{y_0-y_{N_x-1}}{h_x}\right\}-q_0y_0=-f_0,\quad i=0;\quad y_{N_x}=y_0;$$

$$\frac{1}{h_x}\left\{k_{N_x-1/2}\frac{y_0-y_{N_x-1}}{h_x}-k_{N_x-3/2}\frac{y_{N_x-1}-y_{N_x-2}}{h_x}\right\}-q_{N_x-1}y_{N_x-1}=-f_{N_x-1},\quad i=N_x-1.$$

Разностные уравнения на границе для теста 5:

$$\sum_{j=0}^{N}y_j\rho_{0,j}\hbar_j=c_0,\quad \sum_{j=1}^{N}y_j\rho_{1,j}\hbar_j=c_1.$$

3. Параллельная реализация.

Канонический вид алгебраической задачи:

$$-A_iy_{i-1}+C_iy_i-B_iy_{i+1}=F_i,\quad 1\le i\le N-1,$$

$$C_0y_0-B_0y_1=F_0,\quad -A_Ny_{N-1}+C_Ny_N=F_N.$$

Как он получается:

$$\left(-\hbar_xh_x\right)\left[\frac{1}{\hbar_x}\left\{k_{i+1/2}\frac{y_{i+1}-y_i}{h_x}-k_{i-1/2}\frac{y_i-y_{i-1}}{h_x}\right\}-q_iy_i\right]=\left(-\hbar_xh_x\right)\left[-f_i\right],$$

$$k_{i+1/2}\left(y_i-y_{i+1}\right)+k_{i-1/2}\left(y_i-y_{i-1}\right)+\hbar_xh_xq_iy_i=\hbar_xh_xf_i,$$

$$-k_{i-1/2}y_{i-1} + \left(k_{i+1/2} + k_{i-1/2} + \hbar_x h_x q_i\right) y_i - k_{i+1/2} y_{i+1} = \hbar_x h_x f_i,$$

$$-A_i y_{i-1} + C_i y_i - B_i y_{i+1} = F_i.$$

Аналогично получаем уравнения на границах, если задавались граничные условия 2-го или 3-го рода.

Значения коэффициентов в тесте 1:

$$A_i = \begin{cases} 0, & i = 0, N, \\ 0.5(k_i + k_{i-1}), & 1 \le i \le N-1, \end{cases} \quad B_i = \begin{cases} 0, & i = 0, N, \\ 0.5(k_i + k_{i+1}), & 1 \le i \le N-1, \end{cases} \quad C_i = \begin{cases} 1, & i = 0, N, \\ A_i + B_i + h^2 q_i, & 1 \le i \le N-1, \end{cases}$$

$$F_0 = u_a, \quad F_N = u_b, \quad F_i = h^2 f_i, \quad 1 \le i \le N-1.$$

Значения коэффициентов в тесте 2:

$$A_i = \begin{cases} 0, & i = 0, \\ 0.5(k_i + k_{i-1}), & 1 \le i \le N-1, \\ k_{N-1/2}, & i = N, \end{cases} \quad B_i = \begin{cases} k_{1/2}, & i = 0, \\ 0.5(k_i + k_{i+1}), & 1 \le i \le N-1, \\ 0, \end{cases} \quad C_i = \begin{cases} B_0 + 0.5h^2 q_0, & i = 0, \\ A_i + B_i + h^2 q_i, & 1 \le i \le N-1, \\ A_N + 0.5h^2 q_N, & i = N, \end{cases}$$

$$F_0 = hk_0 u_a + 0.5h^2 f_0, \quad F_N = hk_N u_b + 0.5h^2 f_N, \quad F_i = h^2 f_i, \quad 1 \le i \le N-1.$$

Значения коэффициентов в тесте 3:

$$A_i = \begin{cases} 0, & i = 0, \\ 0.5(k_i + k_{i-1}), & 1 \le i \le N-1, \\ k_N \eta, & i = N, \end{cases} \quad B_i = \begin{cases} k_0 \eta, & i = 0, \\ 0.5(k_i + k_{i+1}), & 1 \le i \le N-1, \\ 0, & i = N, \end{cases} \quad C_i = \begin{cases} B_0 + 0.5h^2 q_0, & i = 0, \\ A_i + B_i + h^2 q_i, & 1 \le i \le N-1, \\ A_N + 0.5h^2 q_N, & i = N, \end{cases}$$

$$F_0 = hk_0 \eta + 0.5h^2 f_0, \quad F_N = hk_N \eta + 0.5h^2 f_N, \quad F_i = h^2 f_i, \quad 1 \le i \le N-1.$$

Значения коэффициентов в тесте 4:

$$A_i = \begin{cases} 0.5(k_0 + k_{N-1}), & i = 0, \\ 0.5(k_i + k_{i-1}), & 1 \le i \le N-2, \\ 0.5(k_{N-1} + k_{N-2}), & i = N-1, \end{cases} \quad B_i = \begin{cases} 0.5(k_1 + k_0), & i = 0, \\ 0.5(k_i + k_{i+1}), & 1 \le i \le N-2, \\ 0.5(k_0 + k_{N-1}), & i = N-1, \end{cases}$$

$$C_i = A_i + B_i + h^2 q_i, \quad F_i = h^2 f_i, \quad 0 \le i \le N-1.$$

Значения коэффициентов в тесте 5:

$$A_i = \begin{cases} 0, & i = 0, N, \\ 0.5(k_i + k_{i-1}), & 1 \le i \le N-1, \end{cases} \quad B_i = \begin{cases} 0, & i = 0, N, \\ 0.5(k_i + k_{i+1}), & 1 \le i \le N-1, \end{cases} \quad C_i = \begin{cases} 1, & i = 0, N, \\ A_i + B_i + h^2 q_i, & 1 \le i \le N-1, \end{cases}$$

$$F_i = h^2 f_i, \quad 1 \le i \le N-1;$$

$$\sum_{j=0}^{N} y_j \rho_{0,j} \hbar_j = c_0 = F_0, \quad \sum_{j=1}^{N} y_j \rho_{1,j} \hbar_j = c_1 = F_N.$$

Алгоритм решения во всех случаях – соответствующая параллельная прогонка.

4. Реализация примеров.

Пример 1. Решение линейной первой краевой задачи методом правой прогонки (ex11a.c).

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <math.h>
#include "mpi.h"
#include "mycom.h"
#include "mynet.h"
#include "myprog.h"
int np, mp, nl, ier, lp;
char pname[MPI_MAX_PROCESSOR_NAME];
char sname[10] = "ex11a.p00";
MPI_Status status;
union_t buf;
double tick, t1, t2, t3;
FILE *Fi = NULL;
FILE *Fo = NULL;
int nx;
double xa, xb, ua, ub, ak, px, px2;
```

```
double k(double x);
double k(double x) {
  return 1.0 + ak*(x-xa)*(x-xa);
}
double k1(double x);
double k1(double x) {
  return ak*2.0*(x-xa);
}

double q(double x);
double q(double x) {
  return 1.0 + ak*(xb-x)*(xb-x);
}

// Test function
double u(double x);
double u(double x) {
  return ua*cos(px*(x-xa)) + ub*sin(px*(x-xa));
}

double u1(double x);
double u1(double x) {
  return px*(-ua*sin(px*(x-xa)) + ub*cos(px*(x-xa)));
}

double u2(double x);
double u2(double x) {
  return -px2*(ua*cos(px*(x-xa)) + ub*sin(px*(x-xa)));
}

double f(double x);
double f(double x) {
  return - k1(x)*u1(x) - k(x)*u2(x) + q(x)*u(x);
}

int main(int argc, char *argv[])
{
  int i, j, i1, i2, nc, ncm, ncp, ncx;
  double hx, hx2, s0, s1, s2, a0, b0, c0, f0, a1, b1, c1, f1;
  double *xx, *aa, *bb, *cc, *dd, *ee, *ff, *al, *y1, *y2, *y3, *y4;

  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+7, "%02d", mp);
  ier = fopen_m(&Fo, sname, "wt");
  if (ier!=0) mpierr("Protocol file not opened",1);

  if (mp==0) {
    fprintf(stderr,"Usage: mpirun -np <process number> ex11a.px [<grid cells
number>]\n");

    ier = fopen_m(&Fi,"ex11a.d","rt");
    if (ier!=0) mpierr("Data file not opened",2);
    i = fscanf(Fi,"xa=%le\n",&xa);
    i = fscanf(Fi,"xb=%le\n",&xb);
    i = fscanf(Fi,"ua=%le\n",&ua);
    i = fscanf(Fi,"ub=%le\n",&ub);
    i = fscanf(Fi,"nx=%d\n",&nx);
    i = fscanf(Fi,"lp=%d\n",&lp);
    fclose_m(&Fi);

    if (argc>1) sscanf(argv[1],"%d",&nx);
  }
```

```
if (np>1) {
  if (mp==0) {
    buf.ddata[0] = xa; buf.ddata[1] = xb;
    buf.ddata[2] = ua; buf.ddata[3] = ub;
    buf.idata[8] = nx; buf.idata[9] = lp;
  }
  MPI_Bcast(buf.ddata,5,MPI_DOUBLE,0,MPI_COMM_WORLD);
  if (mp>0) {
    xa = buf.ddata[0]; xb = buf.ddata[1];
    ua = buf.ddata[2]; ub = buf.ddata[3];
    nx = buf.idata[8]; lp = buf.idata[9];
  }
}

fprintf(Fo,"Netsize: %d, process: %d, system: %s, tick=%12le\n",np,mp,pname,tick);
fprintf(Fo,"xa=%le xb=%le ua=%le ub=%le nx=%d lp=%d\n",xa,xb,ua,ub,nx,lp);

t1 = MPI_Wtime();
ak = 1.0/((xb-xa)*(xb-xa)); px = 0.5*pi/(xb-xa); px2 = px*px;
hx = (xb-xa)/nx; hx2 = hx * hx;

// Domain decomposition
MyRange(np, mp, 0, nx, &i1, &i2, &nc);

ncm = nc-1; ncp = 2*(np-1); ncx = imax(nc,ncp);
fprintf(Fo,"np=%d mp=%d nx=%d --> i1=%d i2=%d nc=%d ncm=%d ncp=%d ncx=%d\n",
  np, mp, nx, i1, i2, nc, ncm, ncp, ncx);

xx = (double*)(malloc(sizeof(double)*nc));
aa = (double*)(malloc(sizeof(double)*ncx));
bb = (double*)(malloc(sizeof(double)*ncx));
cc = (double*)(malloc(sizeof(double)*ncx));
ff = (double*)(malloc(sizeof(double)*ncx));
al = (double*)(malloc(sizeof(double)*ncx));
y1 = (double*)(malloc(sizeof(double)*nc));

for (i=0; i<nc; i++)
  xx[i] = xa + hx * (i1 + i);

if (mp==0) {
  aa[0] = 0.0; bb[0] = 0.0; cc[0] = 1.0; ff[0] = ua;
}
else {
  s0 = k(xx[0]); s1 = k(xx[0]-hx); s2 = k(xx[0]+hx);
  aa[0] = 0.5 * (s0 + s1);
  bb[0] = 0.5 * (s0 + s2);
  cc[0] = hx2 * q(xx[0]) + aa[0] + bb[0];
  ff[0] = hx2 * f(xx[0]);
}

for (i=1; i<ncm; i++) {
  s0 = k(xx[i]); s1 = k(xx[i-1]); s2 = k(xx[i+1]);
  aa[i] = 0.5 * (s0 + s1);
  bb[i] = 0.5 * (s0 + s2);
  cc[i] = hx2 * q(xx[i]) + aa[i] + bb[i];
  ff[i] = hx2 * f(xx[i]);
}

if (mp==np-1) {
  aa[ncm] = 0.0; bb[ncm] = 0.0; cc[ncm] = 1.0; ff[ncm] = ub;
}
else {
  s0 = k(xx[ncm]); s1 = k(xx[ncm]-hx); s2 = k(xx[ncm]+hx);
  aa[ncm] = 0.5 * (s0 + s1);
  bb[ncm] = 0.5 * (s0 + s2);
  cc[ncm] = hx2 * q(xx[ncm]) + aa[ncm] + bb[ncm];
  ff[ncm] = hx2 * f(xx[ncm]);
}
```

```
if (np<2) {
  ier = prog_right(nc,aa,bb,cc,ff,al,y1);
  if (ier!=0) mpierr("Bad solution 1",1);
  t2 = 0.0;
}
else {
  y2 = (double*)(malloc(sizeof(double)*nc));
  y3 = (double*)(malloc(sizeof(double)*nc));
  y4 = (double*)(malloc(sizeof(double)*ncp));
  dd = (double*)(malloc(sizeof(double)*4*ncp));
  ee = (double*)(malloc(sizeof(double)*4*ncp));

  a0 = aa[0];   b0 = bb[0];   c0 = cc[0];   f0 = ff[0];
  a1 = aa[ncm]; b1 = bb[ncm]; c1 = cc[ncm]; f1 = ff[ncm];

  if (mp==0) {
    aa[ncm] = 0.0; bb[ncm] = 0.0; cc[ncm] = 1.0; ff[ncm] = 0.0;
    ier = prog_right(nc,aa,bb,cc,ff,al,y1); if (ier!=0) mpierr("Bad solution 1",1);
    for (i=0; i<ncm; i++) ff[i] = 0.0; ff[ncm] = 1.0;
    ier = prog_right(nc,aa,bb,cc,ff,al,y2); if (ier!=0) mpierr("Bad solution 2",2);
  }
  else if (mp<np-1) {
    aa[0] = 0.0; bb[0] = 0.0; cc[0] = 1.0; ff[0] = 0.0;
    aa[ncm] = 0.0; bb[ncm] = 0.0; cc[ncm] = 1.0; ff[ncm] = 0.0;
    ier = prog_right(nc,aa,bb,cc,ff,al,y1); if (ier!=0) mpierr("Bad solution 1",1);
    for (i=0; i<ncm; i++) ff[i] = 0.0; ff[ncm] = 1.0;
    ier = prog_right(nc,aa,bb,cc,ff,al,y2); if (ier!=0) mpierr("Bad solution 2",2);
    ff[0] = 1.0; for (i=1; i<=ncm; i++) ff[i] = 0.0;
    ier = prog_right(nc,aa,bb,cc,ff,al,y3); if (ier!=0) mpierr("Bad solution 3",3);
  }
  else {
    aa[0] = 0.0; bb[0] = 0.0; cc[0] = 1.0; ff[0] = 0.0;
    ier = prog_right(nc,aa,bb,cc,ff,al,y1); if (ier!=0) mpierr("Bad solution 1",1);
    ff[0] = 1.0; for (i=1; i<=ncm; i++) ff[i] = 0.0;
    ier = prog_right(nc,aa,bb,cc,ff,al,y3); if (ier!=0) mpierr("Bad solution 3",3);
  }

  for (i=0; i<4*ncp; i++) dd[i] = 0; for (i=0; i<4*ncp; i++) ee[i] = 0;

  if (mp==0) {
    c1 = c1 - a1 * y2[ncm-1]; f1 = f1 + a1 * y1[ncm-1]; a1 = 0.0;
    dd[0] = a1; dd[1] = b1; dd[2] = c1; dd[3] = f1;
  }
  else if (mp<np-1) {
    c0 = c0 - b0 * y3[1]; f0 = f0 + b0 * y1[1]; b0 = b0 * y2[1];
    c1 = c1 - a1 * y2[ncm-1]; f1 = f1 + a1 * y1[ncm-1]; a1 = a1 * y3[ncm-1];
    i = mp * 8 - 4;
    dd[i]   = a0; dd[i+1] = b0; dd[i+2] = c0; dd[i+3] = f0;
    dd[i+4] = a1; dd[i+5] = b1; dd[i+6] = c1; dd[i+7] = f1;
  }
  else {
    c0 = c0 - b0 * y3[1]; f0 = f0 + b0 * y1[1]; b0 = 0.0;
    i = mp * 8 - 4;
    dd[i]   = a0; dd[i+1] = b0; dd[i+2] = c0; dd[i+3] = f0;
  }

  t2 = MPI_Wtime();
  MPI_Allreduce(dd,ee,4*ncp,MPI_DOUBLE,MPI_SUM,MPI_COMM_WORLD);
  t2 = MPI_Wtime() - t2;

  for (i=0; i<ncp; i++) {
    j = 4*i;
    aa[i] = ee[j]; bb[i] = ee[j+1]; cc[i] = ee[j+2]; ff[i] = ee[j+3];
  }

  ier = prog_right(ncp,aa,bb,cc,ff,al,y4); if (ier!=0) mpierr("Bad solution 4",4);
```

```
    if (mp==0){
      b1 = y4[0]; for (i=0; i<nc; i++) y1[i] = y1[i] + b1 * y2[i];
    }
    else if (mp<np-1) {
      a1 = y4[2*mp-1]; b1 = y4[2*mp];
      for (i=0; i<nc; i++) y1[i] = y1[i] + a1 * y3[i] + b1 * y2[i];
    }
    else {
      a1 = y4[2*mp-1]; for (i=0; i<nc; i++) y1[i] = y1[i] + a1 * y3[i];
    }
  }

  t1 = MPI_Wtime() - t1;

  s0 = 0.0;
  for (i=0; i<nc; i++) {
    s1 = u(xx[i]); s2 = dabs(s1-y1[i]); s0 = dmax(s0,s2);
  }

  if (np>1) {
    s1 = s0;
    MPI_Allreduce(&s1, &s0, 1, MPI_DOUBLE, MPI_MAX, MPI_COMM_WORLD);
  }

  if (mp==0)
    fprintf(stderr,"nx=%d t1=%le t2=%le dmax=%le\n",nx,t1,t2,s0);

  fprintf(Fo,"t1=%le t2=%le dmax=%le\n",t1,t2,s0);

  ier = fclose_m(&Fo);
  MPI_Finalize();

  return 0;
}
```

Трансляция:

```
>mpicc -o ex11a.px -O2 ex11a.c mycom.c mynet.c myprog.c -lm
```

Расчеты на сходимость по сетке:

```
>mpirun -np 1 -nolocal -machinefile hosts ex11a.px 10-10000000 (в классе)
>mpirun -np 1 ex11a.px 10-10000000                             (на сервере)

nx=      10 t1=7.000000e-06 t2=0.000000e+00 dmax=1.260130e-03
nx=     100 t1=3.500000e-05 t2=0.000000e+00 dmax=1.258753e-05
nx=    1000 t1=3.300000e-04 t2=0.000000e+00 dmax=1.258734e-07
nx=   10000 t1=3.146000e-03 t2=0.000000e+00 dmax=1.236859e-09
nx=  100000 t1=3.346400e-02 t2=0.000000e+00 dmax=9.804504e-10
nx= 1000000 t1=3.342550e-01 t2=0.000000e+00 dmax=3.189187e-08
nx=10000000 t1=3.009283e+00 t2=0.000000e+00 dmax=1.672966e-05
```

Расчеты на эффективность:

```
>mpirun -np 1-16 -nolocal -machinefile hosts ex11a.px 10000000
np= 1 nx=10000000 t1=3.227453e+00 t2=0.000000e+00 dmax=1.672966e-05
np= 2 nx=10000000 t1=2.164483e+00 t2=2.252420e-01 dmax=1.659745e-05
np= 3 nx=10000000 t1=1.583892e+00 t2=2.862640e-01 dmax=1.655211e-05
np= 4 nx=10000000 t1=1.214389e+00 t2=2.375560e-01 dmax=1.650617e-05
np= 5 nx=10000000 t1=9.821130e-01 t2=1.846140e-01 dmax=1.640877e-05
np= 6 nx=10000000 t1=8.235920e-01 t2=1.575800e-01 dmax=1.660942e-05
np= 7 nx=10000000 t1=7.175770e-01 t2=1.457190e-01 dmax=1.657224e-05
np= 8 nx=10000000 t1=6.279780e-01 t2=1.293170e-01 dmax=1.647258e-05
np= 9 nx=10000000 t1=5.709290e-01 t2=1.182260e-01 dmax=1.653773e-05
np=10 nx=10000000 t1=5.288050e-01 t2=1.222760e-01 dmax=1.645452e-05
np=11 nx=10000000 t1=4.882020e-01 t2=1.132150e-01 dmax=1.648861e-05
np=12 nx=10000000 t1=4.424140e-01 t2=1.010360e-01 dmax=1.657097e-05
np=13 nx=10000000 t1=4.208710e-01 t2=1.006940e-01 dmax=1.651073e-05
np=14 nx=10000000 t1=3.901090e-01 t2=9.523800e-02 dmax=1.663803e-05
np=15 nx=10000000 t1=3.699740e-01 t2=9.468500e-02 dmax=1.645407e-05
np=16 nx=10000000 t1=3.586580e-01 t2=1.007230e-01 dmax=1.648750e-05
```

Пример 2. Решение линейной второй краевой задачи методом правой прогонки (ex11b.c).

Отличия:

```
double u(double x); double u(double x){
                    return (ua*sin(px*(x-xa)) - ub*cos(px*(x-xa)))/px;}
  if (mp==0) {
    s0 = k(xx[0]); s2 = k(xx[1]);
    aa[0] = 0.0; bb[0] = 0.5 * (s0 + s2);
    cc[0] = 0.5 * hx2 * q(xx[0]) + bb[0]; ff[0] = 0.5 * hx2 * f(xx[0]) - hx * ua * s0;
  }
  else {
    s0 = k(xx[0]); s1 = k(xx[0]-hx); s2 = k(xx[0]+hx);
    aa[0] = 0.5 * (s0 + s1); bb[0] = 0.5 * (s0 + s2);
    cc[0] = hx2 * q(xx[0]) + aa[0] + bb[0]; ff[0] = hx2 * f(xx[0]);
  }
  for (i=1; i<ncm; i++) {
    s0 = k(xx[i]); s1 = k(xx[i-1]); s2 = k(xx[i+1]);
    aa[i] = 0.5 * (s0 + s1); bb[i] = 0.5 * (s0 + s2);
    cc[i] = hx2 * q(xx[i]) + aa[i] + bb[i]; ff[i] = hx2 * f(xx[i]);
  }
  if (mp==np-1) {
    s0 = k(xx[ncm]); s1 = k(xx[ncm-1]);
    aa[ncm] = 0.5 * (s0 + s1); bb[ncm] = 0.0;
    cc[ncm] = 0.5*hx2 * q(xx[ncm]) + aa[ncm]; ff[ncm] = 0.5*hx2 * f(xx[ncm]) + hx*ub*s0;
  }
  else {
    s0 = k(xx[ncm]); s1 = k(xx[ncm]-hx); s2 = k(xx[ncm]+hx);
    aa[ncm] = 0.5 * (s0 + s1);
    bb[ncm] = 0.5 * (s0 + s2);
    cc[ncm] = hx2 * q(xx[ncm]) + aa[ncm] + bb[ncm];
    ff[ncm] = hx2 * f(xx[ncm]);
  }
```

Трансляция:

```
>mpicc -o ex11b.px -O2 ex11b.c mycom.c mynet.c myprog.c -lm
```

Расчеты на эффективность:

```
>mpirun -np 1-16 -nolocal -machinefile hosts ex11b.px 10000000
np= 1 nx=10000000 t1=3.253029e+00 t2=0.000000e+00 dmax=2.813989e-04
np= 2 nx=10000000 t1=2.299431e+00 t2=1.907540e-01 dmax=3.204477e-04
np= 4 nx=10000000 t1=1.282975e+00 t2=2.143770e-01 dmax=3.189007e-04
np= 8 nx=10000000 t1=6.725080e-01 t2=1.207250e-01 dmax=3.189379e-04
np=16 nx=10000000 t1=3.714060e-01 t2=8.903900e-02 dmax=3.184931e-04
```

Пример 3. Решение линейной третьей краевой задачи методом правой прогонки (ex11c.c).

...

Пример 4. Решение линейной периодической задачи методом правой циклической прогонки (ex11d.c).

...

Пример 5. Решение линейной задачи с интегральными граничными условиями методом интегральной прогонки (ex11e.c).

...