#include <stdlib.h>

#include <string.h>

#include <math.h>

int block0np0;

double Delta0block0;

double eps;

double TENO\_CT;

int niter;

double dt;

double gama;

double gamma\_m1;

double rc0;

double rc1;

double rc2;

double rc3;

double rc4;

double rc5;

double rc6;

double rc7;

double rc8;

double rc9;

double rc10;

double rc11;

double rcinv12;

double rcinv13;

#define OPS\_1D

#include "ops\_seq.h"

#include "opensbliblock00\_kernels.h"

int main(int argc, char \*\*argv)

{

block0np0 = 200;

Delta0block0 = 1.0/(block0np0-1);

eps = 1.0e-16;

TENO\_CT = 1.0e-5;

niter = ceil(0.2/0.0002);

double rkB[] = {1.0/3.0, 15.0/16.0, 8.0/15.0};

double rkA[] = {0, -5.0/9.0, -153.0/128.0};

dt = 0.0002;

gama = 1.4;

gamma\_m1 = gama - 1;

rc0 = 1.0/2.0;

rc1 = 1.0/4.0;

rc2 = 13.0/12.0;

rc3 = 3.0/2.0;

rc4 = 11.0/20.0;

rc5 = 2.0/5.0;

rc6 = 1.0/20.0;

rc7 = 1.0/12.0;

rc8 = 5.0/12.0;

rc9 = 1.0/6.0;

rc10 = 7.0/12.0;

rc11 = 11.0/12.0;

rcinv12 = 1.0/gamma\_m1;

rcinv13 = 1.0/Delta0block0;

// Initializing OPS

ops\_init(argc,argv,1);

ops\_decl\_const("block0np0" , 1, "int", &block0np0);

ops\_decl\_const("Delta0block0" , 1, "double", &Delta0block0);

ops\_decl\_const("eps" , 1, "double", &eps);

ops\_decl\_const("TENO\_CT" , 1, "double", &TENO\_CT);

ops\_decl\_const("niter" , 1, "int", &niter);

ops\_decl\_const("dt" , 1, "double", &dt);

ops\_decl\_const("gama" , 1, "double", &gama);

ops\_decl\_const("gamma\_m1" , 1, "double", &gamma\_m1);

ops\_decl\_const("rc0" , 1, "double", &rc0);

ops\_decl\_const("rc1" , 1, "double", &rc1);

ops\_decl\_const("rc2" , 1, "double", &rc2);

ops\_decl\_const("rc3" , 1, "double", &rc3);

ops\_decl\_const("rc4" , 1, "double", &rc4);

ops\_decl\_const("rc5" , 1, "double", &rc5);

ops\_decl\_const("rc6" , 1, "double", &rc6);

ops\_decl\_const("rc7" , 1, "double", &rc7);

ops\_decl\_const("rc8" , 1, "double", &rc8);

ops\_decl\_const("rc9" , 1, "double", &rc9);

ops\_decl\_const("rc10" , 1, "double", &rc10);

ops\_decl\_const("rc11" , 1, "double", &rc11);

ops\_decl\_const("rcinv12" , 1, "double", &rcinv12);

ops\_decl\_const("rcinv13" , 1, "double", &rcinv13);

// Define and Declare OPS Block

ops\_block opensbliblock00 = ops\_decl\_block(1, "opensbliblock00");

#include "defdec\_data\_set.h"

// Define and declare stencils

int stencil\_0\_00temp[] = {0};

ops\_stencil stencil\_0\_00 = ops\_decl\_stencil(1,1,stencil\_0\_00temp,"stencil\_0\_00temp");

int stencil\_0\_01temp[] = {-2, -1, 0, 1, 2, 3};

ops\_stencil stencil\_0\_01 = ops\_decl\_stencil(1,6,stencil\_0\_01temp,"stencil\_0\_01temp");

int stencil\_0\_02temp[] = {-1, 0};

ops\_stencil stencil\_0\_02 = ops\_decl\_stencil(1,2,stencil\_0\_02temp,"stencil\_0\_02temp");

// Init OPS partition

ops\_partition("");

int iteration\_range\_10\_block0[] = {-5, block0np0 + 5};

ops\_par\_loop(opensbliblock00Kernel010, "Grid\_based\_initialisation0", opensbliblock00, 1, iteration\_range\_10\_block0,

ops\_arg\_dat(rhou0\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rhoE\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(x0\_B0, 1, stencil\_0\_00, "double", OPS\_RW),

ops\_arg\_idx());

double cpu\_start0, elapsed\_start0;

ops\_timers(&cpu\_start0, &elapsed\_start0);

for(int iter=0; iter<=niter - 1; iter++)

{

int iteration\_range\_8\_block0[] = {-3, 1};

ops\_par\_loop(opensbliblock00Kernel008, "Dirichlet boundary dir0 side0", opensbliblock00, 1, iteration\_range\_8\_block0,

ops\_arg\_dat(rhou0\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rhoE\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE));

int iteration\_range\_9\_block0[] = {block0np0 - 1, block0np0 + 4};

ops\_par\_loop(opensbliblock00Kernel009, "Dirichlet boundary dir0 side1", opensbliblock00, 1, iteration\_range\_9\_block0,

ops\_arg\_dat(rhou0\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rhoE\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE));

for(int stage=0; stage<=2; stage++)

{

int iteration\_range\_3\_block0[] = {-3, block0np0 + 4};

ops\_par\_loop(opensbliblock00Kernel003, "CRu0", opensbliblock00, 1, iteration\_range\_3\_block0,

ops\_arg\_dat(rhou0\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(u0\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE));

int iteration\_range\_4\_block0[] = {-3, block0np0 + 4};

ops\_par\_loop(opensbliblock00Kernel004, "CRp", opensbliblock00, 1, iteration\_range\_4\_block0,

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(u0\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(rhoE\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(p\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE));

int iteration\_range\_2\_block0[] = {-3, block0np0 + 4};

ops\_par\_loop(opensbliblock00Kernel002, "CRa", opensbliblock00, 1, iteration\_range\_2\_block0,

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(p\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(a\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE));

int iteration\_range\_0\_block0[] = {-1, block0np0 + 1};

ops\_par\_loop(opensbliblock00Kernel000, "LLFTeno\_reconstruction\_0\_direction", opensbliblock00, 1, iteration\_range\_0\_block0,

ops\_arg\_dat(rhou0\_B0, 1, stencil\_0\_01, "double", OPS\_READ),

ops\_arg\_dat(p\_B0, 1, stencil\_0\_01, "double", OPS\_READ),

ops\_arg\_dat(a\_B0, 1, stencil\_0\_01, "double", OPS\_READ),

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_01, "double", OPS\_READ),

ops\_arg\_dat(u0\_B0, 1, stencil\_0\_01, "double", OPS\_READ),

ops\_arg\_dat(rhoE\_B0, 1, stencil\_0\_01, "double", OPS\_READ),

ops\_arg\_dat(wk2\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(wk0\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(wk1\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE));

int iteration\_range\_6\_block0[] = {0, block0np0};

ops\_par\_loop(opensbliblock00Kernel006, "LLFTeno Residual", opensbliblock00, 1, iteration\_range\_6\_block0,

ops\_arg\_dat(wk2\_B0, 1, stencil\_0\_02, "double", OPS\_READ),

ops\_arg\_dat(wk0\_B0, 1, stencil\_0\_02, "double", OPS\_READ),

ops\_arg\_dat(wk1\_B0, 1, stencil\_0\_02, "double", OPS\_READ),

ops\_arg\_dat(Residual1\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(Residual0\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(Residual2\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE));

int iteration\_range\_11\_block0[] = {0, block0np0};

ops\_par\_loop(opensbliblock00Kernel011, "Temporal solution advancement", opensbliblock00, 1, iteration\_range\_11\_block0,

ops\_arg\_dat(Residual2\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(Residual0\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(Residual1\_B0, 1, stencil\_0\_00, "double", OPS\_READ),

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_00, "double", OPS\_RW),

ops\_arg\_dat(tempRK\_rhou0\_B0, 1, stencil\_0\_00, "double", OPS\_RW),

ops\_arg\_dat(rhou0\_B0, 1, stencil\_0\_00, "double", OPS\_RW),

ops\_arg\_dat(tempRK\_rhoE\_B0, 1, stencil\_0\_00, "double", OPS\_RW),

ops\_arg\_dat(tempRK\_rho\_B0, 1, stencil\_0\_00, "double", OPS\_RW),

ops\_arg\_dat(rhoE\_B0, 1, stencil\_0\_00, "double", OPS\_RW),

ops\_arg\_gbl(&rkB[stage], 1, "double", OPS\_READ),

ops\_arg\_gbl(&rkA[stage], 1, "double", OPS\_READ));

int iteration\_range\_8\_block0[] = {-3, 1};

ops\_par\_loop(opensbliblock00Kernel008, "Dirichlet boundary dir0 side0", opensbliblock00, 1, iteration\_range\_8\_block0,

ops\_arg\_dat(rhou0\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rhoE\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE));

int iteration\_range\_9\_block0[] = {block0np0 - 1, block0np0 + 4};

ops\_par\_loop(opensbliblock00Kernel009, "Dirichlet boundary dir0 side1", opensbliblock00, 1, iteration\_range\_9\_block0,

ops\_arg\_dat(rhou0\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rho\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE),

ops\_arg\_dat(rhoE\_B0, 1, stencil\_0\_00, "double", OPS\_WRITE));

}

}

double cpu\_end0, elapsed\_end0;

ops\_timers(&cpu\_end0, &elapsed\_end0);

ops\_printf("\nTimings are:\n");

ops\_printf("-----------------------------------------\n");

ops\_printf("Total Wall time %lf\n",elapsed\_end0-elapsed\_start0);

char name0[80];

sprintf(name0, "opensbli\_output.h5");

ops\_fetch\_block\_hdf5\_file(opensbliblock00, name0);

ops\_fetch\_dat\_hdf5\_file(rho\_B0, name0);

ops\_fetch\_dat\_hdf5\_file(rhou0\_B0, name0);

ops\_fetch\_dat\_hdf5\_file(rhoE\_B0, name0);

ops\_fetch\_dat\_hdf5\_file(x0\_B0, name0);

ops\_exit();

//Main program end

}