

```

#ifdef OPENSBLIBLOCK00_KERNEL_H
#define OPENSBLIBLOCK00_KERNEL_H
void opensblliblock00Kernel024(double *rho_B0, double *rhou0_B0, double *rhoE_B0, double *x0_B0, const int *idx)
{
    double p = 0.0;
    double d = 0.0;
    double u0 = 0.0;
    x0_B0[OPS_ACC3(0)] = Delta0block0*idx[0];

    u0 = 0;

    p = ((x0_B0[OPS_ACC3(0)] < 0.5) ? (
    1.0
    )
    : (
    0.1
    ));

    d = ((x0_B0[OPS_ACC3(0)] < 0.5) ? (
    1.0
    )
    : (
    0.125
    ));

    rho_B0[OPS_ACC0(0)] = d;

    rhou0_B0[OPS_ACC1(0)] = d*u0;

    rhoE_B0[OPS_ACC2(0)] = 0.5*d*pow(u0, 2.0) + p/(gama - 1.0);
}

void opensblliblock00Kernel025(double *detJ_B0)
{
    detJ_B0[OPS_ACC0(-1)] = detJ_B0[OPS_ACC0(1)];

    detJ_B0[OPS_ACC0(-2)] = detJ_B0[OPS_ACC0(2)];

    detJ_B0[OPS_ACC0(-3)] = detJ_B0[OPS_ACC0(3)];
}

void opensblliblock00Kernel026(double *detJ_B0)
{
    detJ_B0[OPS_ACC0(1)] = detJ_B0[OPS_ACC0(-1)];

    detJ_B0[OPS_ACC0(2)] = detJ_B0[OPS_ACC0(-2)];

    detJ_B0[OPS_ACC0(3)] = detJ_B0[OPS_ACC0(-3)];

    detJ_B0[OPS_ACC0(4)] = detJ_B0[OPS_ACC0(-4)];
}

void opensblliblock00Kernel022(double *rho_B0, double *rhou0_B0, double *rhoE_B0)
{
    double p = 0.0;
    double d = 0.0;
    double u0 = 0.0;
    d = 1.0000000000000000;

    u0 = 0.0;

    p = 1.0000000000000000;

    rho_B0[OPS_ACC0(0)] = d;

    rhou0_B0[OPS_ACC1(0)] = d*u0;
}

```

```

rhoE_B0[OPS_ACC2(0)] = 0.5*d*pow(u0, 2.0) + p/(gama - 1.0);
}

void opensblliblock00Kernel023(double *rho_B0, double *rho_u0_B0, double *rhoE_B0)
{
double p = 0.0;
double d = 0.0;
double u0 = 0.0;
d = 0.1250000000000000;

u0 = 0.0;

p = 0.1000000000000000;

rho_B0[OPS_ACC0(0)] = d;

rho_u0_B0[OPS_ACC1(0)] = d*u0;

rhoE_B0[OPS_ACC2(0)] = 0.5*d*pow(u0, 2.0) + p/(gama - 1.0);
}

void opensblliblock00Kernel003(const double *rho_B0, const double *rho_u0_B0, double *u0_B0)
{
u0_B0[OPS_ACC2(0)] = rho_u0_B0[OPS_ACC1(0)]/rho_B0[OPS_ACC0(0)];
}

void opensblliblock00Kernel013(const double *rho_B0, const double *u0_B0, const double *rhoE_B0, double *p_B0)
{
p_B0[OPS_ACC3(0)] = (gama - 1)*(rhoE_B0[OPS_ACC2(0)] - rc1*rho_B0[OPS_ACC0(0)]*pow(u0_B0[OPS_ACC1(0)], 2));
}

void opensblliblock00Kernel002(const double *rhoE_B0, const double *p_B0, const double *rho_B0, const double *rho_u0_B0,
const double *u0_B0, double *wk16_B0, double *wk14_B0, double *wk17_B0, double *wk15_B0)
{
wk14_B0[OPS_ACC6(0)] = rhoE_B0[OPS_ACC0(0)]*u0_B0[OPS_ACC4(0)];

wk15_B0[OPS_ACC8(0)] = rho_B0[OPS_ACC2(0)]*u0_B0[OPS_ACC4(0)];

wk16_B0[OPS_ACC5(0)] = rho_u0_B0[OPS_ACC3(0)]*u0_B0[OPS_ACC4(0)];

wk17_B0[OPS_ACC7(0)] = p_B0[OPS_ACC1(0)]*u0_B0[OPS_ACC4(0)];
}

void opensblliblock00Kernel005(const double *wk14_B0, double *wk3_B0, const int *idx)
{
wk3_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
-8.34657956545823e-15*wk14_B0[OPS_ACC0(4)] -
1.83333333333334*wk14_B0[OPS_ACC0(0)] + 1.06910315192207e-15*wk14_B0[OPS_ACC0(5)] +
3.00000000000002*wk14_B0[OPS_ACC0(1)] - 1.50000000000003*wk14_B0[OPS_ACC0(2)] +
0.33333333333356*wk14_B0[OPS_ACC0(3)]
)
: ((idx[0] == 1) ? (
0.00571369039775442*wk14_B0[OPS_ACC0(4)] -
0.322484932882161*wk14_B0[OPS_ACC0(0)] - 0.376283677513354*wk14_B0[OPS_ACC0(-1)] +
0.719443173328855*wk14_B0[OPS_ACC0(1)] + 0.0394168524399447*wk14_B0[OPS_ACC0(2)] -
0.0658051057710389*wk14_B0[OPS_ACC0(3)]
)
: ((idx[0] == 2) ? (
0.197184333887745*wk14_B0[OPS_ACC0(0)] +
0.113446470384241*wk14_B0[OPS_ACC0(-2)] - 0.791245592765872*wk14_B0[OPS_ACC0(-1)] +
0.521455851089587*wk14_B0[OPS_ACC0(1)] - 0.0367146847001261*wk14_B0[OPS_ACC0(2)] -
0.00412637789557492*wk14_B0[OPS_ACC0(3)]
)
: ((idx[0] == 3) ? (

```

```

-0.00932597985049999*wk14_B0[OPS_ACC0(-3)] +
0.0451033223343881*wk14_B0[OPS_ACC0(0)] + 0.121937153224065*wk14_B0[OPS_ACC0(-2)] -
0.727822147724592*wk14_B0[OPS_ACC0(-1)] + 0.652141084861241*wk14_B0[OPS_ACC0(1)] -
0.082033432844602*wk14_B0[OPS_ACC0(2)]
)
: ((idx[0] == block0np0 - 1) ? (
-0.333333333333356*wk14_B0[OPS_ACC0(-3)] + 1.83333333333334*wk14_B0[OPS_ACC0(0)] +
1.500000000000003*wk14_B0[OPS_ACC0(-2)] - 3.000000000000002*wk14_B0[OPS_ACC0(-1)] -
1.06910315192207e-15*wk14_B0[OPS_ACC0(-5)] + 8.34657956545823e-15*wk14_B0[OPS_ACC0(-4)]
)
: ((idx[0] == block0np0
- 2) ? (
0.0658051057710389*wk14_B0[OPS_ACC0(-3)] + 0.322484932882161*wk14_B0[OPS_ACC0(0)] -
0.0394168524399447*wk14_B0[OPS_ACC0(-2)] - 0.719443173328855*wk14_B0[OPS_ACC0(-1)] +
0.376283677513354*wk14_B0[OPS_ACC0(1)] - 0.00571369039775442*wk14_B0[OPS_ACC0(-4)]
)
: ((idx[0] == block0np0 - 3)
? (
0.00412637789557492*wk14_B0[OPS_ACC0(-3)] - 0.197184333887745*wk14_B0[OPS_ACC0(0)] +
0.0367146847001261*wk14_B0[OPS_ACC0(-2)] - 0.521455851089587*wk14_B0[OPS_ACC0(-1)] +
0.791245592765872*wk14_B0[OPS_ACC0(1)] - 0.113446470384241*wk14_B0[OPS_ACC0(2)]
)
: ((idx[0] == block0np0 - 4) ?
(
-0.0451033223343881*wk14_B0[OPS_ACC0(0)] + 0.082033432844602*wk14_B0[OPS_ACC0(-2)] -
0.652141084861241*wk14_B0[OPS_ACC0(-1)] + 0.727822147724592*wk14_B0[OPS_ACC0(1)] -
0.121937153224065*wk14_B0[OPS_ACC0(2)] + 0.00932597985049999*wk14_B0[OPS_ACC0(3)]
)
: (
-rc2*wk14_B0[OPS_ACC0(2)] + (rc3)*wk14_B0[OPS_ACC0(1)] - rc3*wk14_B0[OPS_ACC0(-1)] +
(rc2)*wk14_B0[OPS_ACC0(-2)]
))))))));
}

```

```

void opensblblock00Kernel007(const double *q0_B0, double *wk4_B0, const int *idx)
{
wk4_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
0.333333333333356*q0_B0[OPS_ACC0(3)] -
8.34657956545823e-15*q0_B0[OPS_ACC0(4)] - 1.83333333333334*q0_B0[OPS_ACC0(0)] +
1.06910315192207e-15*q0_B0[OPS_ACC0(5)] + 3.000000000000002*q0_B0[OPS_ACC0(1)] -
1.500000000000003*q0_B0[OPS_ACC0(2)]
)
: ((idx[0] == 1) ? (
-0.0658051057710389*q0_B0[OPS_ACC0(3)] +
0.00571369039775442*q0_B0[OPS_ACC0(4)] - 0.322484932882161*q0_B0[OPS_ACC0(0)] -
0.376283677513354*q0_B0[OPS_ACC0(-1)] + 0.719443173328855*q0_B0[OPS_ACC0(1)] +
0.0394168524399447*q0_B0[OPS_ACC0(2)]
)
: ((idx[0] == 2) ? (
-0.00412637789557492*q0_B0[OPS_ACC0(3)] +
0.197184333887745*q0_B0[OPS_ACC0(0)] + 0.113446470384241*q0_B0[OPS_ACC0(-2)] +
0.521455851089587*q0_B0[OPS_ACC0(1)] - 0.791245592765872*q0_B0[OPS_ACC0(-1)] -
0.0367146847001261*q0_B0[OPS_ACC0(2)]
)
: ((idx[0] == 3) ? (
-0.00932597985049999*q0_B0[OPS_ACC0(-3)] +
0.0451033223343881*q0_B0[OPS_ACC0(0)] - 0.727822147724592*q0_B0[OPS_ACC0(-1)] +
0.121937153224065*q0_B0[OPS_ACC0(-2)] + 0.652141084861241*q0_B0[OPS_ACC0(1)] -
0.082033432844602*q0_B0[OPS_ACC0(2)]
)
: ((idx[0] == block0np0 - 1) ? (
-0.333333333333356*q0_B0[OPS_ACC0(-3)]
+ 1.83333333333334*q0_B0[OPS_ACC0(0)] + 1.500000000000003*q0_B0[OPS_ACC0(-2)] -
3.000000000000002*q0_B0[OPS_ACC0(-1)] - 1.06910315192207e-15*q0_B0[OPS_ACC0(-5)] +
8.34657956545823e-15*q0_B0[OPS_ACC0(-4)]
)
)
}

```

```

: ((idx[0] == block0np0 - 2) ? (
0.0658051057710389*q0_B0[OPS_ACC0(-3)] + 0.322484932882161*q0_B0[OPS_ACC0(0)] -
0.0394168524399447*q0_B0[OPS_ACC0(-2)] + 0.376283677513354*q0_B0[OPS_ACC0(1)] -
0.719443173328855*q0_B0[OPS_ACC0(-1)] - 0.00571369039775442*q0_B0[OPS_ACC0(-4)]
)
: ((idx[0] == block0np0 - 3) ?
(
0.00412637789557492*q0_B0[OPS_ACC0(-3)] - 0.197184333887745*q0_B0[OPS_ACC0(0)] -
0.521455851089587*q0_B0[OPS_ACC0(-1)] + 0.0367146847001261*q0_B0[OPS_ACC0(-2)] +
0.791245592765872*q0_B0[OPS_ACC0(1)] - 0.113446470384241*q0_B0[OPS_ACC0(2)]
)
: ((idx[0] == block0np0 - 4) ? (
0.00932597985049999*q0_B0[OPS_ACC0(3)] - 0.0451033223343881*q0_B0[OPS_ACC0(0)] -
0.652141084861241*q0_B0[OPS_ACC0(-1)] + 0.082033432844602*q0_B0[OPS_ACC0(-2)] +
0.727822147724592*q0_B0[OPS_ACC0(1)] - 0.121937153224065*q0_B0[OPS_ACC0(2)]
)
: (
(rc2)*q0_B0[OPS_ACC0(-2)] -
rc2*q0_B0[OPS_ACC0(2)] - rc3*q0_B0[OPS_ACC0(-1)] + (rc3)*q0_B0[OPS_ACC0(1)]
))))))));
}

void opensbliblock00Kernel009(const double *wk15_B0, double *wk5_B0, const int *idx)
{
wk5_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
0.333333333333356*wk15_B0[OPS_ACC0(3)] -
1.83333333333334*wk15_B0[OPS_ACC0(0)] - 8.34657956545823e-15*wk15_B0[OPS_ACC0(4)] +
3.00000000000002*wk15_B0[OPS_ACC0(1)] + 1.06910315192207e-15*wk15_B0[OPS_ACC0(5)] -
1.50000000000003*wk15_B0[OPS_ACC0(2)]
)
: ((idx[0] == 1) ? (
-0.0658051057710389*wk15_B0[OPS_ACC0(3)] -
0.376283677513354*wk15_B0[OPS_ACC0(-1)] - 0.322484932882161*wk15_B0[OPS_ACC0(0)] +
0.00571369039775442*wk15_B0[OPS_ACC0(4)] + 0.719443173328855*wk15_B0[OPS_ACC0(1)] +
0.0394168524399447*wk15_B0[OPS_ACC0(2)]
)
: ((idx[0] == 2) ? (
-0.00412637789557492*wk15_B0[OPS_ACC0(3)] +
0.113446470384241*wk15_B0[OPS_ACC0(-2)] - 0.791245592765872*wk15_B0[OPS_ACC0(-1)] +
0.197184333887745*wk15_B0[OPS_ACC0(0)] + 0.521455851089587*wk15_B0[OPS_ACC0(1)] -
0.0367146847001261*wk15_B0[OPS_ACC0(2)]
)
: ((idx[0] == 3) ? (
0.121937153224065*wk15_B0[OPS_ACC0(-2)] -
0.727822147724592*wk15_B0[OPS_ACC0(-1)] + 0.0451033223343881*wk15_B0[OPS_ACC0(0)] +
0.652141084861241*wk15_B0[OPS_ACC0(1)] - 0.082033432844602*wk15_B0[OPS_ACC0(2)] -
0.00932597985049999*wk15_B0[OPS_ACC0(-3)]
)
: ((idx[0] == block0np0 - 1) ? (
1.50000000000003*wk15_B0[OPS_ACC0(-2)] - 3.00000000000002*wk15_B0[OPS_ACC0(-1)] +
1.83333333333334*wk15_B0[OPS_ACC0(0)] - 1.06910315192207e-15*wk15_B0[OPS_ACC0(-5)] +
8.34657956545823e-15*wk15_B0[OPS_ACC0(-4)] - 0.333333333333356*wk15_B0[OPS_ACC0(-3)]
)
: ((idx[0] == block0np0 -
2) ? (
-0.0394168524399447*wk15_B0[OPS_ACC0(-2)] - 0.719443173328855*wk15_B0[OPS_ACC0(-1)] +
0.322484932882161*wk15_B0[OPS_ACC0(0)] + 0.376283677513354*wk15_B0[OPS_ACC0(1)] -
0.00571369039775442*wk15_B0[OPS_ACC0(-4)] + 0.0658051057710389*wk15_B0[OPS_ACC0(-3)]
)
: ((idx[0] == block0np0 -
3) ? (
0.0367146847001261*wk15_B0[OPS_ACC0(-2)] - 0.521455851089587*wk15_B0[OPS_ACC0(-1)] -
0.197184333887745*wk15_B0[OPS_ACC0(0)] + 0.791245592765872*wk15_B0[OPS_ACC0(1)] -
0.113446470384241*wk15_B0[OPS_ACC0(2)] + 0.00412637789557492*wk15_B0[OPS_ACC0(-3)]
)
: ((idx[0] == block0np0 - 4)

```

```

? (
0.00932597985049999*wk15_B0[OPS_ACC0(3)] + 0.082033432844602*wk15_B0[OPS_ACC0(-2)] -
0.652141084861241*wk15_B0[OPS_ACC0(-1)] - 0.0451033223343881*wk15_B0[OPS_ACC0(0)] +
0.727822147724592*wk15_B0[OPS_ACC0(1)] - 0.121937153224065*wk15_B0[OPS_ACC0(2)]
)
: (
-rc3*wk15_B0[OPS_ACC0(-1)] + (rc3)*wk15_B0[OPS_ACC0(1)] - rc2*wk15_B0[OPS_ACC0(2)] +
(rc2)*wk15_B0[OPS_ACC0(-2)]
))))))));

}

void opensbliblock00Kernel011(const double *wk16_B0, double *wk6_B0, const int *idx)
{
wk6_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
3.000000000000002*wk16_B0[OPS_ACC0(1)] -
8.34657956545823e-15*wk16_B0[OPS_ACC0(4)] - 1.833333333333334*wk16_B0[OPS_ACC0(0)] +
0.3333333333333356*wk16_B0[OPS_ACC0(3)] - 1.500000000000003*wk16_B0[OPS_ACC0(2)] +
1.06910315192207e-15*wk16_B0[OPS_ACC0(5)]
)
: ((idx[0] == 1) ? (
0.719443173328855*wk16_B0[OPS_ACC0(1)] +
0.00571369039775442*wk16_B0[OPS_ACC0(4)] - 0.322484932882161*wk16_B0[OPS_ACC0(0)] -
0.0658051057710389*wk16_B0[OPS_ACC0(3)] + 0.0394168524399447*wk16_B0[OPS_ACC0(2)] -
0.376283677513354*wk16_B0[OPS_ACC0(-1)]
)
: ((idx[0] == 2) ? (
0.521455851089587*wk16_B0[OPS_ACC0(1)] +
0.197184333887745*wk16_B0[OPS_ACC0(0)] - 0.00412637789557492*wk16_B0[OPS_ACC0(3)] -
0.0367146847001261*wk16_B0[OPS_ACC0(2)] - 0.791245592765872*wk16_B0[OPS_ACC0(-1)] +
0.113446470384241*wk16_B0[OPS_ACC0(-2)]
)
: ((idx[0] == 3) ? (
0.652141084861241*wk16_B0[OPS_ACC0(1)] -
0.00932597985049999*wk16_B0[OPS_ACC0(-3)] + 0.0451033223343881*wk16_B0[OPS_ACC0(0)] -
0.082033432844602*wk16_B0[OPS_ACC0(2)] + 0.121937153224065*wk16_B0[OPS_ACC0(-2)] -
0.727822147724592*wk16_B0[OPS_ACC0(-1)]
)
: ((idx[0] == block0np0 - 1) ? (
-0.3333333333333356*wk16_B0[OPS_ACC0(-3)] + 1.833333333333334*wk16_B0[OPS_ACC0(0)] +
8.34657956545823e-15*wk16_B0[OPS_ACC0(-4)] - 1.06910315192207e-15*wk16_B0[OPS_ACC0(-5)] -
3.000000000000002*wk16_B0[OPS_ACC0(-1)] + 1.500000000000003*wk16_B0[OPS_ACC0(-2)]
)
: ((idx[0] == block0np0 - 2) ?
(
0.376283677513354*wk16_B0[OPS_ACC0(1)] + 0.0658051057710389*wk16_B0[OPS_ACC0(-3)] +
0.322484932882161*wk16_B0[OPS_ACC0(0)] - 0.00571369039775442*wk16_B0[OPS_ACC0(-4)] -
0.719443173328855*wk16_B0[OPS_ACC0(-1)] - 0.0394168524399447*wk16_B0[OPS_ACC0(-2)]
)
: ((idx[0] == block0np0 - 3)
? (
0.791245592765872*wk16_B0[OPS_ACC0(1)] + 0.00412637789557492*wk16_B0[OPS_ACC0(-3)] -
0.197184333887745*wk16_B0[OPS_ACC0(0)] - 0.113446470384241*wk16_B0[OPS_ACC0(2)] +
0.0367146847001261*wk16_B0[OPS_ACC0(-2)] - 0.521455851089587*wk16_B0[OPS_ACC0(-1)]
)
: ((idx[0] == block0np0 - 4)
? (
0.727822147724592*wk16_B0[OPS_ACC0(1)] - 0.0451033223343881*wk16_B0[OPS_ACC0(0)] +
0.00932597985049999*wk16_B0[OPS_ACC0(3)] - 0.121937153224065*wk16_B0[OPS_ACC0(2)] -
0.652141084861241*wk16_B0[OPS_ACC0(-1)] + 0.082033432844602*wk16_B0[OPS_ACC0(-2)]
)
: (
(rc3)*wk16_B0[OPS_ACC0(1)] + (rc2)*wk16_B0[OPS_ACC0(-2)] - rc3*wk16_B0[OPS_ACC0(-1)] -
rc2*wk16_B0[OPS_ACC0(2)]
))))))));

}

```

```

void opensbliblock00Kernel012(const double *rhoE_B0, double *wk7_B0, const int *idx)
{
    wk7_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
    1.06910315192207e-15*rhoE_B0[OPS_ACC0(5)] -
    1.50000000000003*rhoE_B0[OPS_ACC0(2)] + 0.333333333333356*rhoE_B0[OPS_ACC0(3)] -
    1.83333333333334*rhoE_B0[OPS_ACC0(0)] - 8.34657956545823e-15*rhoE_B0[OPS_ACC0(4)] +
    3.00000000000002*rhoE_B0[OPS_ACC0(1)]
    )
    : ((idx[0] == 1) ? (
    -0.376283677513354*rhoE_B0[OPS_ACC0(-1)] +
    0.0394168524399447*rhoE_B0[OPS_ACC0(2)] - 0.0658051057710389*rhoE_B0[OPS_ACC0(3)] -
    0.322484932882161*rhoE_B0[OPS_ACC0(0)] + 0.00571369039775442*rhoE_B0[OPS_ACC0(4)] +
    0.719443173328855*rhoE_B0[OPS_ACC0(1)]
    )
    : ((idx[0] == 2) ? (
    0.113446470384241*rhoE_B0[OPS_ACC0(-2)] -
    0.791245592765872*rhoE_B0[OPS_ACC0(-1)] - 0.0367146847001261*rhoE_B0[OPS_ACC0(2)] -
    0.00412637789557492*rhoE_B0[OPS_ACC0(3)] + 0.197184333887745*rhoE_B0[OPS_ACC0(0)] +
    0.521455851089587*rhoE_B0[OPS_ACC0(1)]
    )
    : ((idx[0] == 3) ? (
    0.121937153224065*rhoE_B0[OPS_ACC0(-2)] -
    0.727822147724592*rhoE_B0[OPS_ACC0(-1)] - 0.082033432844602*rhoE_B0[OPS_ACC0(2)] +
    0.0451033223343881*rhoE_B0[OPS_ACC0(0)] - 0.00932597985049999*rhoE_B0[OPS_ACC0(-3)] +
    0.652141084861241*rhoE_B0[OPS_ACC0(1)]
    )
    : ((idx[0] == block0np0 - 1) ? (
    1.50000000000003*rhoE_B0[OPS_ACC0(-2)] - 3.00000000000002*rhoE_B0[OPS_ACC0(-1)] -
    1.06910315192207e-15*rhoE_B0[OPS_ACC0(-5)] + 8.34657956545823e-15*rhoE_B0[OPS_ACC0(-4)] +
    1.83333333333334*rhoE_B0[OPS_ACC0(0)] - 0.333333333333356*rhoE_B0[OPS_ACC0(-3)]
    )
    : ((idx[0] == block0np0 - 2) ?
    (
    -0.0394168524399447*rhoE_B0[OPS_ACC0(-2)] - 0.719443173328855*rhoE_B0[OPS_ACC0(-1)] -
    0.00571369039775442*rhoE_B0[OPS_ACC0(-4)] + 0.322484932882161*rhoE_B0[OPS_ACC0(0)] +
    0.0658051057710389*rhoE_B0[OPS_ACC0(-3)] + 0.376283677513354*rhoE_B0[OPS_ACC0(1)]
    )
    : ((idx[0] == block0np0 - 3) ?
    (
    0.0367146847001261*rhoE_B0[OPS_ACC0(-2)] - 0.521455851089587*rhoE_B0[OPS_ACC0(-1)] -
    0.113446470384241*rhoE_B0[OPS_ACC0(2)] - 0.197184333887745*rhoE_B0[OPS_ACC0(0)] +
    0.00412637789557492*rhoE_B0[OPS_ACC0(-3)] + 0.791245592765872*rhoE_B0[OPS_ACC0(1)]
    )
    : ((idx[0] == block0np0 - 4)
    ? (
    0.082033432844602*rhoE_B0[OPS_ACC0(-2)] - 0.652141084861241*rhoE_B0[OPS_ACC0(-1)] -
    0.121937153224065*rhoE_B0[OPS_ACC0(2)] + 0.00932597985049999*rhoE_B0[OPS_ACC0(3)] -
    0.0451033223343881*rhoE_B0[OPS_ACC0(0)] + 0.727822147724592*rhoE_B0[OPS_ACC0(1)]
    )
    : (
    -rc3*rhoE_B0[OPS_ACC0(-1)] + (rc2)*rhoE_B0[OPS_ACC0(-2)] - rc2*rhoE_B0[OPS_ACC0(2)] +
    (rc3)*rhoE_B0[OPS_ACC0(1)]
    ))))));
}

void opensbliblock00Kernel014(const double *wk17_B0, double *wk8_B0, const int *idx)
{
    wk8_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
    3.00000000000002*wk17_B0[OPS_ACC0(1)] +
    1.06910315192207e-15*wk17_B0[OPS_ACC0(5)] - 1.83333333333334*wk17_B0[OPS_ACC0(0)] -
    8.34657956545823e-15*wk17_B0[OPS_ACC0(4)] + 0.333333333333356*wk17_B0[OPS_ACC0(3)] -
    1.50000000000003*wk17_B0[OPS_ACC0(2)]
    )
    : ((idx[0] == 1) ? (
    0.719443173328855*wk17_B0[OPS_ACC0(1)] -

```

```

0.322484932882161*wk17_B0[OPS_ACC0(0)] + 0.00571369039775442*wk17_B0[OPS_ACC0(4)] -
0.0658051057710389*wk17_B0[OPS_ACC0(3)] - 0.376283677513354*wk17_B0[OPS_ACC0(-1)] +
0.0394168524399447*wk17_B0[OPS_ACC0(2)]
)
: ((idx[0] == 2) ? (
0.521455851089587*wk17_B0[OPS_ACC0(1)] +
0.197184333887745*wk17_B0[OPS_ACC0(0)] - 0.791245592765872*wk17_B0[OPS_ACC0(-1)] -
0.00412637789557492*wk17_B0[OPS_ACC0(3)] + 0.113446470384241*wk17_B0[OPS_ACC0(-2)] -
0.0367146847001261*wk17_B0[OPS_ACC0(2)]
)
: ((idx[0] == 3) ? (
-0.00932597985049999*wk17_B0[OPS_ACC0(-3)] +
0.652141084861241*wk17_B0[OPS_ACC0(1)] + 0.0451033223343881*wk17_B0[OPS_ACC0(0)] -
0.727822147724592*wk17_B0[OPS_ACC0(-1)] + 0.121937153224065*wk17_B0[OPS_ACC0(-2)] -
0.082033432844602*wk17_B0[OPS_ACC0(2)]
)
: ((idx[0] == block0np0 - 1) ? (
-0.333333333333356*wk17_B0[OPS_ACC0(-3)] + 8.34657956545823e-15*wk17_B0[OPS_ACC0(-4)] +
1.83333333333334*wk17_B0[OPS_ACC0(0)] - 3.000000000000002*wk17_B0[OPS_ACC0(-1)] -
1.06910315192207e-15*wk17_B0[OPS_ACC0(-5)] + 1.500000000000003*wk17_B0[OPS_ACC0(-2)]
)
: ((idx[0] == block0np0 - 2)
? (
0.0658051057710389*wk17_B0[OPS_ACC0(-3)] + 0.376283677513354*wk17_B0[OPS_ACC0(1)] -
0.00571369039775442*wk17_B0[OPS_ACC0(-4)] + 0.322484932882161*wk17_B0[OPS_ACC0(0)] -
0.719443173328855*wk17_B0[OPS_ACC0(-1)] - 0.0394168524399447*wk17_B0[OPS_ACC0(-2)]
)
: ((idx[0] == block0np0 - 3)
? (
0.00412637789557492*wk17_B0[OPS_ACC0(-3)] + 0.791245592765872*wk17_B0[OPS_ACC0(1)] -
0.197184333887745*wk17_B0[OPS_ACC0(0)] - 0.521455851089587*wk17_B0[OPS_ACC0(-1)] +
0.0367146847001261*wk17_B0[OPS_ACC0(-2)] - 0.113446470384241*wk17_B0[OPS_ACC0(2)]
)
: ((idx[0] == block0np0 - 4) ?
(
0.727822147724592*wk17_B0[OPS_ACC0(1)] - 0.0451033223343881*wk17_B0[OPS_ACC0(0)] -
0.652141084861241*wk17_B0[OPS_ACC0(-1)] + 0.00932597985049999*wk17_B0[OPS_ACC0(3)] +
0.082033432844602*wk17_B0[OPS_ACC0(-2)] - 0.121937153224065*wk17_B0[OPS_ACC0(2)]
)
: (
-rc3*wk17_B0[OPS_ACC0(-1)] + (rc2)*wk17_B0[OPS_ACC0(-2)] - rc2*wk17_B0[OPS_ACC0(2)] +
(rc3)*wk17_B0[OPS_ACC0(1)]
))))))));
}

void opensblliblock00Kernel015(const double *p_B0, double *wk9_B0, const int *idx)
{
wk9_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
-1.833333333333334*p_B0[OPS_ACC0(0)] +
1.06910315192207e-15*p_B0[OPS_ACC0(5)] + 3.000000000000002*p_B0[OPS_ACC0(1)] - 1.500000000000003*p_B0[OPS_ACC0(2)] +
0.333333333333356*p_B0[OPS_ACC0(3)] - 8.34657956545823e-15*p_B0[OPS_ACC0(4)]
)
: ((idx[0] == 1) ? (
-0.322484932882161*p_B0[OPS_ACC0(0)] + 0.719443173328855*p_B0[OPS_ACC0(1)] + 0.0394168524399447*p_B0[OPS_ACC0(2)]
- 0.376283677513354*p_B0[OPS_ACC0(-1)] - 0.0658051057710389*p_B0[OPS_ACC0(3)] +
0.00571369039775442*p_B0[OPS_ACC0(4)]
)
: ((idx[0] == 2) ? (
0.197184333887745*p_B0[OPS_ACC0(0)] +
0.521455851089587*p_B0[OPS_ACC0(1)] - 0.0367146847001261*p_B0[OPS_ACC0(2)] + 0.113446470384241*p_B0[OPS_ACC0(-2)]
- 0.791245592765872*p_B0[OPS_ACC0(-1)] - 0.00412637789557492*p_B0[OPS_ACC0(3)]
)
: ((idx[0] == 3) ? (
0.0451033223343881*p_B0[OPS_ACC0(0)] + 0.652141084861241*p_B0[OPS_ACC0(1)] -
0.00932597985049999*p_B0[OPS_ACC0(-3)] - 0.082033432844602*p_B0[OPS_ACC0(2)] -
0.727822147724592*p_B0[OPS_ACC0(-1)] + 0.121937153224065*p_B0[OPS_ACC0(-2)]

```



```

)
: ((idx[0] == block0np0 - 1) ? (
1.8333333333333334*p_B0[OPS_ACC0(0)] - 1.06910315192207e-15*p_B0[OPS_ACC0(-5)] +
8.34657956545823e-15*p_B0[OPS_ACC0(-4)] - 0.333333333333356*p_B0[OPS_ACC0(-3)] +
1.500000000000003*p_B0[OPS_ACC0(-2)] - 3.000000000000002*p_B0[OPS_ACC0(-1)]
)
: ((idx[0] == block0np0 - 2) ? (
0.322484932882161*p_B0[OPS_ACC0(0)] - 0.00571369039775442*p_B0[OPS_ACC0(-4)] + 0.376283677513354*p_B0[OPS_ACC0(1)]
+ 0.0658051057710389*p_B0[OPS_ACC0(-3)] - 0.0394168524399447*p_B0[OPS_ACC0(-2)] -
0.719443173328855*p_B0[OPS_ACC0(-1)]
)
: ((idx[0] == block0np0 - 3) ? (
-0.197184333887745*p_B0[OPS_ACC0(0)] +
0.791245592765872*p_B0[OPS_ACC0(1)] + 0.00412637789557492*p_B0[OPS_ACC0(-3)] - 0.113446470384241*p_B0[OPS_ACC0(2)]
- 0.521455851089587*p_B0[OPS_ACC0(-1)] + 0.0367146847001261*p_B0[OPS_ACC0(-2)]
)
: ((idx[0] == block0np0 - 4) ? (
-0.0451033223343881*p_B0[OPS_ACC0(0)] + 0.727822147724592*p_B0[OPS_ACC0(1)] -
0.121937153224065*p_B0[OPS_ACC0(2)] - 0.652141084861241*p_B0[OPS_ACC0(-1)] + 0.082033432844602*p_B0[OPS_ACC0(-2)]
+ 0.00932597985049999*p_B0[OPS_ACC0(3)]
)
: (
-rc2*p_B0[OPS_ACC0(2)] + (rc3)*p_B0[OPS_ACC0(1)] -
rc3*p_B0[OPS_ACC0(-1)] + (rc2)*p_B0[OPS_ACC0(-2)]
))))))));
}

```

```

void opensbliblock00Kernel016(const double *rho_B0, double *wk10_B0, const int *idx)
{
wk10_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
0.333333333333356*rho_B0[OPS_ACC0(3)] -
1.500000000000003*rho_B0[OPS_ACC0(2)] + 1.06910315192207e-15*rho_B0[OPS_ACC0(5)] +
3.000000000000002*rho_B0[OPS_ACC0(1)] - 8.34657956545823e-15*rho_B0[OPS_ACC0(4)] -
1.833333333333334*rho_B0[OPS_ACC0(0)]
)
: ((idx[0] == 1) ? (
-0.0658051057710389*rho_B0[OPS_ACC0(3)] +
0.0394168524399447*rho_B0[OPS_ACC0(2)] - 0.376283677513354*rho_B0[OPS_ACC0(-1)] +
0.719443173328855*rho_B0[OPS_ACC0(1)] + 0.00571369039775442*rho_B0[OPS_ACC0(4)] -
0.322484932882161*rho_B0[OPS_ACC0(0)]
)
: ((idx[0] == 2) ? (
-0.00412637789557492*rho_B0[OPS_ACC0(3)] -
0.0367146847001261*rho_B0[OPS_ACC0(2)] + 0.113446470384241*rho_B0[OPS_ACC0(-2)] -
0.791245592765872*rho_B0[OPS_ACC0(-1)] + 0.521455851089587*rho_B0[OPS_ACC0(1)] +
0.197184333887745*rho_B0[OPS_ACC0(0)]
)
: ((idx[0] == 3) ? (
-0.082033432844602*rho_B0[OPS_ACC0(2)] +
0.121937153224065*rho_B0[OPS_ACC0(-2)] - 0.727822147724592*rho_B0[OPS_ACC0(-1)] +
0.652141084861241*rho_B0[OPS_ACC0(1)] - 0.00932597985049999*rho_B0[OPS_ACC0(-3)] +
0.0451033223343881*rho_B0[OPS_ACC0(0)]
)
: ((idx[0] == block0np0 - 1) ? (
-1.06910315192207e-15*rho_B0[OPS_ACC0(-5)] + 1.500000000000003*rho_B0[OPS_ACC0(-2)] -
3.000000000000002*rho_B0[OPS_ACC0(-1)] - 0.333333333333356*rho_B0[OPS_ACC0(-3)] +
1.833333333333334*rho_B0[OPS_ACC0(0)] + 8.34657956545823e-15*rho_B0[OPS_ACC0(-4)]
)
: ((idx[0] == block0np0 - 2) ? (
-0.0394168524399447*rho_B0[OPS_ACC0(-2)] - 0.719443173328855*rho_B0[OPS_ACC0(-1)] +
0.376283677513354*rho_B0[OPS_ACC0(1)] + 0.0658051057710389*rho_B0[OPS_ACC0(-3)] +
0.322484932882161*rho_B0[OPS_ACC0(0)] - 0.00571369039775442*rho_B0[OPS_ACC0(-4)]
)
: ((idx[0] == block0np0 - 3) ?

```



```

(
-0.113446470384241*rho_B0[OPS_ACC0(2)] + 0.0367146847001261*rho_B0[OPS_ACC0(-2)] -
0.521455851089587*rho_B0[OPS_ACC0(-1)] + 0.791245592765872*rho_B0[OPS_ACC0(1)] +
0.00412637789557492*rho_B0[OPS_ACC0(-3)] - 0.197184333887745*rho_B0[OPS_ACC0(0)]
)
: ((idx[0] == block0np0 - 4) ?
(
0.00932597985049999*rho_B0[OPS_ACC0(3)] - 0.121937153224065*rho_B0[OPS_ACC0(2)] +
0.082033432844602*rho_B0[OPS_ACC0(-2)] - 0.652141084861241*rho_B0[OPS_ACC0(-1)] +
0.727822147724592*rho_B0[OPS_ACC0(1)] - 0.0451033223343881*rho_B0[OPS_ACC0(0)]
)
: (
(rc3)*rho_B0[OPS_ACC0(1)]
- rc3*rho_B0[OPS_ACC0(-1)] + (rc2)*rho_B0[OPS_ACC0(-2)] - rc2*rho_B0[OPS_ACC0(2)]
))))))));
}

void opensbliblock00Kernel017(const double *rhou0_B0, double *wk11_B0, const int *idx)
{
wk11_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
-8.34657956545823e-15*rhou0_B0[OPS_ACC0(4)] +
3.00000000000002*rhou0_B0[OPS_ACC0(1)] + 1.06910315192207e-15*rhou0_B0[OPS_ACC0(5)] -
1.500000000000003*rhou0_B0[OPS_ACC0(2)] + 0.333333333333356*rhou0_B0[OPS_ACC0(3)] -
1.83333333333334*rhou0_B0[OPS_ACC0(0)]
)
: ((idx[0] == 1) ? (
0.00571369039775442*rhou0_B0[OPS_ACC0(4)] +
0.719443173328855*rhou0_B0[OPS_ACC0(1)] + 0.0394168524399447*rhou0_B0[OPS_ACC0(2)] -
0.0658051057710389*rhou0_B0[OPS_ACC0(3)] - 0.376283677513354*rhou0_B0[OPS_ACC0(-1)] -
0.322484932882161*rhou0_B0[OPS_ACC0(0)]
)
: ((idx[0] == 2) ? (
0.521455851089587*rhou0_B0[OPS_ACC0(1)] -
0.0367146847001261*rhou0_B0[OPS_ACC0(2)] - 0.791245592765872*rhou0_B0[OPS_ACC0(-1)] -
0.00412637789557492*rhou0_B0[OPS_ACC0(3)] + 0.113446470384241*rhou0_B0[OPS_ACC0(-2)] +
0.197184333887745*rhou0_B0[OPS_ACC0(0)]
)
: ((idx[0] == 3) ? (
0.652141084861241*rhou0_B0[OPS_ACC0(1)] -
0.082033432844602*rhou0_B0[OPS_ACC0(2)] - 0.00932597985049999*rhou0_B0[OPS_ACC0(-3)] +
0.121937153224065*rhou0_B0[OPS_ACC0(-2)] - 0.727822147724592*rhou0_B0[OPS_ACC0(-1)] +
0.0451033223343881*rhou0_B0[OPS_ACC0(0)]
)
: ((idx[0] == block0np0 - 1) ? (
8.34657956545823e-15*rhou0_B0[OPS_ACC0(-4)] - 0.333333333333356*rhou0_B0[OPS_ACC0(-3)] -
3.000000000000002*rhou0_B0[OPS_ACC0(-1)] + 1.500000000000003*rhou0_B0[OPS_ACC0(-2)] +
1.83333333333334*rhou0_B0[OPS_ACC0(0)] - 1.06910315192207e-15*rhou0_B0[OPS_ACC0(-5)]
)
: ((idx[0] == block0np0 -
2) ? (
0.376283677513354*rhou0_B0[OPS_ACC0(1)] - 0.00571369039775442*rhou0_B0[OPS_ACC0(-4)] +
0.0658051057710389*rhou0_B0[OPS_ACC0(-3)] - 0.719443173328855*rhou0_B0[OPS_ACC0(-1)] -
0.0394168524399447*rhou0_B0[OPS_ACC0(-2)] + 0.322484932882161*rhou0_B0[OPS_ACC0(0)]
)
: ((idx[0] == block0np0 - 3)
? (
0.791245592765872*rhou0_B0[OPS_ACC0(1)] - 0.113446470384241*rhou0_B0[OPS_ACC0(2)] +
0.00412637789557492*rhou0_B0[OPS_ACC0(-3)] + 0.0367146847001261*rhou0_B0[OPS_ACC0(-2)] -
0.521455851089587*rhou0_B0[OPS_ACC0(-1)] - 0.197184333887745*rhou0_B0[OPS_ACC0(0)]
)
: ((idx[0] == block0np0 - 4)
? (
0.727822147724592*rhou0_B0[OPS_ACC0(1)] - 0.121937153224065*rhou0_B0[OPS_ACC0(2)] +
0.082033432844602*rhou0_B0[OPS_ACC0(-2)] + 0.00932597985049999*rhou0_B0[OPS_ACC0(3)] -
0.652141084861241*rhou0_B0[OPS_ACC0(-1)] - 0.0451033223343881*rhou0_B0[OPS_ACC0(0)]
)
)
}

```

```

: (
(rc2)*rhou0_B0[OPS_ACC0(-2)] - rc3*rhou0_B0[OPS_ACC0(-1)] + (rc3)*rhou0_B0[OPS_ACC0(1)] -
rc2*rhou0_B0[OPS_ACC0(2)]
))))))));

}

void opensblibblock00Kernel018(const double *u0_B0, double *wk12_B0, const int *idx)
{
wk12_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
-1.500000000000003*u0_B0[OPS_ACC0(2)] +
3.000000000000002*u0_B0[OPS_ACC0(1)] + 1.06910315192207e-15*u0_B0[OPS_ACC0(5)] -
1.833333333333334*u0_B0[OPS_ACC0(0)] - 8.34657956545823e-15*u0_B0[OPS_ACC0(4)] +
0.333333333333356*u0_B0[OPS_ACC0(3)]
)
: ((idx[0] == 1) ? (
0.0394168524399447*u0_B0[OPS_ACC0(2)] +
0.719443173328855*u0_B0[OPS_ACC0(1)] - 0.376283677513354*u0_B0[OPS_ACC0(-1)] -
0.322484932882161*u0_B0[OPS_ACC0(0)] + 0.00571369039775442*u0_B0[OPS_ACC0(4)] -
0.0658051057710389*u0_B0[OPS_ACC0(3)]
)
: ((idx[0] == 2) ? (
-0.0367146847001261*u0_B0[OPS_ACC0(2)] +
0.521455851089587*u0_B0[OPS_ACC0(1)] + 0.113446470384241*u0_B0[OPS_ACC0(-2)] -
0.791245592765872*u0_B0[OPS_ACC0(-1)] + 0.197184333887745*u0_B0[OPS_ACC0(0)] -
0.00412637789557492*u0_B0[OPS_ACC0(3)]
)
: ((idx[0] == 3) ? (
-0.082033432844602*u0_B0[OPS_ACC0(2)] +
0.652141084861241*u0_B0[OPS_ACC0(1)] - 0.727822147724592*u0_B0[OPS_ACC0(-1)] +
0.121937153224065*u0_B0[OPS_ACC0(-2)] + 0.0451033223343881*u0_B0[OPS_ACC0(0)] -
0.00932597985049999*u0_B0[OPS_ACC0(-3)]
)
: ((idx[0] == block0np0 - 1) ? (
8.34657956545823e-15*u0_B0[OPS_ACC0(-4)] - 1.06910315192207e-15*u0_B0[OPS_ACC0(-5)] +
1.500000000000003*u0_B0[OPS_ACC0(-2)] - 3.000000000000002*u0_B0[OPS_ACC0(-1)] + 1.833333333333334*u0_B0[OPS_ACC0(0)] -
- 0.333333333333356*u0_B0[OPS_ACC0(-3)]
)
: ((idx[0] == block0np0 - 2) ? (
-0.00571369039775442*u0_B0[OPS_ACC0(-4)] + 0.376283677513354*u0_B0[OPS_ACC0(1)] -
0.0394168524399447*u0_B0[OPS_ACC0(-2)] - 0.719443173328855*u0_B0[OPS_ACC0(-1)] +
0.322484932882161*u0_B0[OPS_ACC0(0)] + 0.0658051057710389*u0_B0[OPS_ACC0(-3)]
)
: ((idx[0] == block0np0 - 3) ? (
-0.113446470384241*u0_B0[OPS_ACC0(2)] + 0.791245592765872*u0_B0[OPS_ACC0(1)] -
0.521455851089587*u0_B0[OPS_ACC0(-1)] + 0.0367146847001261*u0_B0[OPS_ACC0(-2)] -
0.197184333887745*u0_B0[OPS_ACC0(0)] + 0.00412637789557492*u0_B0[OPS_ACC0(-3)]
)
: ((idx[0] == block0np0 - 4) ? (
-0.121937153224065*u0_B0[OPS_ACC0(2)] + 0.727822147724592*u0_B0[OPS_ACC0(1)] -
0.652141084861241*u0_B0[OPS_ACC0(-1)] + 0.082033432844602*u0_B0[OPS_ACC0(-2)] -
0.0451033223343881*u0_B0[OPS_ACC0(0)] + 0.00932597985049999*u0_B0[OPS_ACC0(3)]
)
: (
(rc2)*u0_B0[OPS_ACC0(-2)]
- rc2*u0_B0[OPS_ACC0(2)] + (rc3)*u0_B0[OPS_ACC0(1)] - rc3*u0_B0[OPS_ACC0(-1)]
))))))));

}

void opensblibblock00Kernel020(const double *tau00_B0, double *wk13_B0, const int *idx)
{
wk13_B0[OPS_ACC1(0)] = inv_0*((idx[0] == 0) ? (
-1.500000000000003*tau00_B0[OPS_ACC0(2)] +
0.333333333333356*tau00_B0[OPS_ACC0(3)] - 8.34657956545823e-15*tau00_B0[OPS_ACC0(4)] -
1.833333333333334*tau00_B0[OPS_ACC0(0)] + 1.06910315192207e-15*tau00_B0[OPS_ACC0(5)] +

```

```

3.00000000000002*tau00_B0[OPS_ACC0(1)]
)
: ((idx[0] == 1) ? (
0.0394168524399447*tau00_B0[OPS_ACC0(2)] -
0.0658051057710389*tau00_B0[OPS_ACC0(3)] + 0.00571369039775442*tau00_B0[OPS_ACC0(4)] -
0.322484932882161*tau00_B0[OPS_ACC0(0)] - 0.376283677513354*tau00_B0[OPS_ACC0(-1)] +
0.719443173328855*tau00_B0[OPS_ACC0(1)]
)
: ((idx[0] == 2) ? (
-0.791245592765872*tau00_B0[OPS_ACC0(-1)] -
0.0367146847001261*tau00_B0[OPS_ACC0(2)] - 0.00412637789557492*tau00_B0[OPS_ACC0(3)] +
0.197184333887745*tau00_B0[OPS_ACC0(0)] + 0.113446470384241*tau00_B0[OPS_ACC0(-2)] +
0.521455851089587*tau00_B0[OPS_ACC0(1)]
)
: ((idx[0] == 3) ? (
0.121937153224065*tau00_B0[OPS_ACC0(-2)] -
0.082033432844602*tau00_B0[OPS_ACC0(2)] - 0.00932597985049999*tau00_B0[OPS_ACC0(-3)] +
0.0451033223343881*tau00_B0[OPS_ACC0(0)] - 0.727822147724592*tau00_B0[OPS_ACC0(-1)] +
0.652141084861241*tau00_B0[OPS_ACC0(1)]
)
: ((idx[0] == block0np0 - 1) ? (
-3.000000000000002*tau00_B0[OPS_ACC0(-1)] - 1.06910315192207e-15*tau00_B0[OPS_ACC0(-5)] +
8.34657956545823e-15*tau00_B0[OPS_ACC0(-4)] - 0.333333333333356*tau00_B0[OPS_ACC0(-3)] +
1.83333333333334*tau00_B0[OPS_ACC0(0)] + 1.500000000000003*tau00_B0[OPS_ACC0(-2)]
)
: ((idx[0] == block0np0 - 2) ?
(
-0.719443173328855*tau00_B0[OPS_ACC0(-1)] - 0.00571369039775442*tau00_B0[OPS_ACC0(-4)] +
0.0658051057710389*tau00_B0[OPS_ACC0(-3)] + 0.322484932882161*tau00_B0[OPS_ACC0(0)] -
0.0394168524399447*tau00_B0[OPS_ACC0(-2)] + 0.376283677513354*tau00_B0[OPS_ACC0(1)]
)
: ((idx[0] == block0np0 - 3)
? (
0.0367146847001261*tau00_B0[OPS_ACC0(-2)] - 0.113446470384241*tau00_B0[OPS_ACC0(2)] +
0.00412637789557492*tau00_B0[OPS_ACC0(-3)] - 0.197184333887745*tau00_B0[OPS_ACC0(0)] -
0.521455851089587*tau00_B0[OPS_ACC0(-1)] + 0.791245592765872*tau00_B0[OPS_ACC0(1)]
)
: ((idx[0] == block0np0 - 4)
? (
0.082033432844602*tau00_B0[OPS_ACC0(-2)] - 0.121937153224065*tau00_B0[OPS_ACC0(2)] +
0.00932597985049999*tau00_B0[OPS_ACC0(3)] - 0.0451033223343881*tau00_B0[OPS_ACC0(0)] -
0.652141084861241*tau00_B0[OPS_ACC0(-1)] + 0.727822147724592*tau00_B0[OPS_ACC0(1)]
)
: (
-rc2*tau00_B0[OPS_ACC0(2)] + (rc3)*tau00_B0[OPS_ACC0(1)] - rc3*tau00_B0[OPS_ACC0(-1)] +
(rc2)*tau00_B0[OPS_ACC0(-2)]
))))))));
}

```

```

void opensbliblock00Kernel021(const double *wk5_B0, const double *rho0_B0, const double *wk3_B0, const double
*rhoE_B0, const double *wk11_B0, const double *wk13_B0, const double *wk6_B0, const double *tau00_B0, const double
*rho_B0, const double *wk10_B0, const double *u0_B0, const double *wk12_B0, const double *wk7_B0, const double *wk4_B0,
const double *wk8_B0, const double *wk9_B0, double *Residual0_B0, double *Residual1_B0, double *Residual2_B0)
{
Residual0_B0[OPS_ACC16(0)] = -rc1*rho_B0[OPS_ACC8(0)]*wk12_B0[OPS_ACC11(0)] -
rc1*u0_B0[OPS_ACC10(0)]*wk10_B0[OPS_ACC9(0)] - rc1*wk5_B0[OPS_ACC0(0)];

Residual1_B0[OPS_ACC17(0)] = -rc1*rho0_B0[OPS_ACC1(0)]*wk12_B0[OPS_ACC11(0)] -
rc1*u0_B0[OPS_ACC10(0)]*wk11_B0[OPS_ACC4(0)] + wk13_B0[OPS_ACC5(0)] - rc1*wk6_B0[OPS_ACC6(0)] -
wk9_B0[OPS_ACC15(0)];

Residual2_B0[OPS_ACC18(0)] = -rc1*rhoE_B0[OPS_ACC3(0)]*wk12_B0[OPS_ACC11(0)] +
tau00_B0[OPS_ACC7(0)]*wk12_B0[OPS_ACC11(0)] + u0_B0[OPS_ACC10(0)]*wk13_B0[OPS_ACC5(0)] -
rc1*u0_B0[OPS_ACC10(0)]*wk7_B0[OPS_ACC12(0)] - rc1*wk3_B0[OPS_ACC2(0)] + wk4_B0[OPS_ACC13(0)] -
wk8_B0[OPS_ACC14(0)];

```

```

}

void opensbliblock00Kernel030(const double *Residual1_B0, const double *Residual2_B0, const double *Residual0_B0,
double *tempRK_rho_B0, double *tempRK_rhoE_B0, double *rho_B0, double *rho0_B0, double *tempRK_rho0_B0, double
*rhoE_B0, const double *rkB, const double *rkA)
{
tempRK_rho_B0[OPS_ACC3(0)] = dt*Residual0_B0[OPS_ACC2(0)] + rkA[0]*tempRK_rho_B0[OPS_ACC3(0)];

rho_B0[OPS_ACC5(0)] = rkB[0]*tempRK_rho_B0[OPS_ACC3(0)] + rho_B0[OPS_ACC5(0)];

tempRK_rho0_B0[OPS_ACC7(0)] = dt*Residual1_B0[OPS_ACC0(0)] + rkA[0]*tempRK_rho0_B0[OPS_ACC7(0)];

rho0_B0[OPS_ACC6(0)] = rkB[0]*tempRK_rho0_B0[OPS_ACC7(0)] + rho0_B0[OPS_ACC6(0)];

tempRK_rhoE_B0[OPS_ACC4(0)] = dt*Residual2_B0[OPS_ACC1(0)] + rkA[0]*tempRK_rhoE_B0[OPS_ACC4(0)];

rhoE_B0[OPS_ACC8(0)] = rkB[0]*tempRK_rhoE_B0[OPS_ACC4(0)] + rhoE_B0[OPS_ACC8(0)];

}

void opensbliblock00Kernel027(const double *rho_B0, const double *rho0_B0, const double *rhoE_B0, double
*Mach_sensor_B0, double *u0_B0, double *a_B0, double *p_B0)
{
double M = 0.0;
double p = 0.0;
double inv_rho = 0.0;
double inv_a = 0.0;
double u0 = 0.0;
inv_rho = 1.0/rho_B0[OPS_ACC0(0)];

u0_B0[OPS_ACC4(0)] = inv_rho*rho0_B0[OPS_ACC1(0)];

p_B0[OPS_ACC6(0)] = (gama - 1)*(-0.5*inv_rho*pow(rho0_B0[OPS_ACC1(0)], 2) + rhoE_B0[OPS_ACC2(0)]);

a_B0[OPS_ACC5(0)] = sqrt(gama*inv_rho*p_B0[OPS_ACC6(0)]);

inv_rho = 1.0/rho_B0[OPS_ACC0(0)];

u0 = inv_rho*rho0_B0[OPS_ACC1(0)];

p = (gama - 1)*(-0.5*inv_rho*pow(rho0_B0[OPS_ACC1(0)], 2) + rhoE_B0[OPS_ACC2(0)]);

inv_a = 1.0/sqrt(gama*inv_rho*p);

M = inv_a*sqrt(pow(u0, 2));

Mach_sensor_B0[OPS_ACC3(0)] = fmin(1, 0.5*pow(M, 2)*sqrt(pow(-pow(M, 2) + 1, 2) + 4)/(pow(M, 2) + 1));

}

void opensbliblock00Kernel028(const double *rhoE_B0, const double *p_B0, const double *rho_B0, const double *rho0_B0,
const double *u0_B0, const double *a_B0, double *wk0_B0, double *wk2_B0, double *wk1_B0)
{
double CF_14 = 0.0;
double CS_14 = 0.0;
double CF_21 = 0.0;
double AVG_0_0_LEV_10 = 0.0;
double CF_02 = 0.0;
double beta_2 = 0.0;
double alpha_0 = 0.0;
double CS_04 = 0.0;
double CS_23 = 0.0;
double AVG_0_rho = 0.0;
double CS_01 = 0.0;
double AVG_0_0_LEV_01 = 0.0;
double CF_00 = 0.0;
double rj1 = 0.0;
double AVG_0_0_LEV_21 = 0.0;
double AVG_0_a = 0.0;

```

```

double inv_AVG_rho = 0.0;
double CS_05 = 0.0;
double AVG_0_0_LEV_11 = 0.0;
double CF_03 = 0.0;
double AVG_0_0_LEV_20 = 0.0;
double CF_20 = 0.0;
double CS_00 = 0.0;
double CS_02 = 0.0;
double CF_25 = 0.0;
double Recon_0 = 0.0;
double omega_0 = 0.0;
double beta_1 = 0.0;
double AVG_0_0_LEV_12 = 0.0;
double beta_0 = 0.0;
double Recon_1 = 0.0;
double inv_AVG_a = 0.0;
double inv_alpha_sum = 0.0;
double CF_22 = 0.0;
double max_lambda_0_00 = 0.0;
double CS_25 = 0.0;
double rj0 = 0.0;
double CS_15 = 0.0;
double alpha_2 = 0.0;
double CS_22 = 0.0;
double AVG_0_0_LEV_02 = 0.0;
double omega_1 = 0.0;
double AVG_0_0_LEV_22 = 0.0;
double omega_2 = 0.0;
double AVG_0_u0 = 0.0;
double CS_13 = 0.0;
double CF_24 = 0.0;
double CS_12 = 0.0;
double CS_21 = 0.0;
double CF_23 = 0.0;
double CF_04 = 0.0;
double rj2 = 0.0;
double CF_13 = 0.0;
double CS_03 = 0.0;
double CS_11 = 0.0;
double Recon_2 = 0.0;
double CS_10 = 0.0;
double CF_15 = 0.0;
double CS_24 = 0.0;
double alpha_1 = 0.0;
double CF_05 = 0.0;
double CF_11 = 0.0;
double CF_01 = 0.0;
double max_lambda_0_11 = 0.0;
double CS_20 = 0.0;
double AVG_0_0_LEV_00 = 0.0;
double CF_12 = 0.0;
double max_lambda_0_22 = 0.0;
double CF_10 = 0.0;
AVG_0_a = (rc1)*(a_B0[OPS_ACC5(1)] + a_B0[OPS_ACC5(0)]);

AVG_0_u0 = (rc1)*(u0_B0[OPS_ACC4(0)] + u0_B0[OPS_ACC4(1)]);

inv_AVG_a = 1.0/AVG_0_a;

inv_AVG_rho = 1.0/AVG_0_rho;

AVG_0_0_LEV_00 = (rc4)*AVG_0_u0*inv_AVG_a*(gama*AVG_0_a*AVG_0_u0*pow(inv_AVG_a, 2) -
AVG_0_a*AVG_0_u0*pow(inv_AVG_a,
2) + 2);

AVG_0_0_LEV_01 = -rc1*inv_AVG_a*(gama*AVG_0_a*AVG_0_u0*pow(inv_AVG_a, 2) - AVG_0_a*AVG_0_u0*pow(inv_AVG_a, 2) + 1);

```

AVG_0_0_LEV_02 = (rc1)*gamma_m1*pow(inv_AVG_a, 2);

AVG_0_0_LEV_10 = -rc1*(gama*pow(AVG_0_u0, 2)*pow(inv_AVG_a, 2) - pow(AVG_0_u0, 2)*pow(inv_AVG_a, 2) - 2);

AVG_0_0_LEV_11 = gamma_m1*AVG_0_u0*pow(inv_AVG_a, 2);

AVG_0_0_LEV_12 = -gamma_m1*pow(inv_AVG_a, 2);

AVG_0_0_LEV_20 = (rc4)*AVG_0_u0*inv_AVG_a*(gama*AVG_0_a*AVG_0_u0*pow(inv_AVG_a, 2) -
AVG_0_a*AVG_0_u0*pow(inv_AVG_a,
2) - 2);

AVG_0_0_LEV_21 = -rc1*inv_AVG_a*(gama*AVG_0_a*AVG_0_u0*pow(inv_AVG_a, 2) - AVG_0_a*AVG_0_u0*pow(inv_AVG_a, 2) - 1);

AVG_0_0_LEV_22 = (rc1)*gamma_m1*pow(inv_AVG_a, 2);

CF_00 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(-2)]*u0_B0[OPS_ACC4(-2)] + AVG_0_0_LEV_01*p_B0[OPS_ACC1(-2)] +
AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(-2)]*u0_B0[OPS_ACC4(-2)] + AVG_0_0_LEV_02*p_B0[OPS_ACC1(-2)]*u0_B0[OPS_ACC4(-2)]
+ AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(-2)]*u0_B0[OPS_ACC4(-2)];

CF_10 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(-2)]*u0_B0[OPS_ACC4(-2)] + AVG_0_0_LEV_11*p_B0[OPS_ACC1(-2)] +
AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(-2)]*u0_B0[OPS_ACC4(-2)] + AVG_0_0_LEV_12*p_B0[OPS_ACC1(-2)]*u0_B0[OPS_ACC4(-2)]
+ AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(-2)]*u0_B0[OPS_ACC4(-2)];

CF_20 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(-2)]*u0_B0[OPS_ACC4(-2)] + AVG_0_0_LEV_21*p_B0[OPS_ACC1(-2)] +
AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(-2)]*u0_B0[OPS_ACC4(-2)] + AVG_0_0_LEV_22*p_B0[OPS_ACC1(-2)]*u0_B0[OPS_ACC4(-2)]
+ AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(-2)]*u0_B0[OPS_ACC4(-2)];

CF_01 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(-1)]*u0_B0[OPS_ACC4(-1)] + AVG_0_0_LEV_01*p_B0[OPS_ACC1(-1)] +
AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(-1)]*u0_B0[OPS_ACC4(-1)] + AVG_0_0_LEV_02*p_B0[OPS_ACC1(-1)]*u0_B0[OPS_ACC4(-1)]
+ AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(-1)]*u0_B0[OPS_ACC4(-1)];

CF_11 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(-1)]*u0_B0[OPS_ACC4(-1)] + AVG_0_0_LEV_11*p_B0[OPS_ACC1(-1)] +
AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(-1)]*u0_B0[OPS_ACC4(-1)] + AVG_0_0_LEV_12*p_B0[OPS_ACC1(-1)]*u0_B0[OPS_ACC4(-1)]
+ AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(-1)]*u0_B0[OPS_ACC4(-1)];

CF_21 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(-1)]*u0_B0[OPS_ACC4(-1)] + AVG_0_0_LEV_21*p_B0[OPS_ACC1(-1)] +
AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(-1)]*u0_B0[OPS_ACC4(-1)] + AVG_0_0_LEV_22*p_B0[OPS_ACC1(-1)]*u0_B0[OPS_ACC4(-1)]
+ AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(-1)]*u0_B0[OPS_ACC4(-1)];

CF_02 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(0)]*u0_B0[OPS_ACC4(0)] + AVG_0_0_LEV_01*p_B0[OPS_ACC1(0)] +
AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(0)]*u0_B0[OPS_ACC4(0)] + AVG_0_0_LEV_02*p_B0[OPS_ACC1(0)]*u0_B0[OPS_ACC4(0)] +
AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(0)]*u0_B0[OPS_ACC4(0)];

CF_12 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(0)]*u0_B0[OPS_ACC4(0)] + AVG_0_0_LEV_11*p_B0[OPS_ACC1(0)] +
AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(0)]*u0_B0[OPS_ACC4(0)] + AVG_0_0_LEV_12*p_B0[OPS_ACC1(0)]*u0_B0[OPS_ACC4(0)] +
AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(0)]*u0_B0[OPS_ACC4(0)];

CF_22 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(0)]*u0_B0[OPS_ACC4(0)] + AVG_0_0_LEV_21*p_B0[OPS_ACC1(0)] +
AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(0)]*u0_B0[OPS_ACC4(0)] + AVG_0_0_LEV_22*p_B0[OPS_ACC1(0)]*u0_B0[OPS_ACC4(0)] +
AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(0)]*u0_B0[OPS_ACC4(0)];

CF_03 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(1)]*u0_B0[OPS_ACC4(1)] + AVG_0_0_LEV_01*p_B0[OPS_ACC1(1)] +
AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(1)]*u0_B0[OPS_ACC4(1)] + AVG_0_0_LEV_02*p_B0[OPS_ACC1(1)]*u0_B0[OPS_ACC4(1)] +
AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(1)]*u0_B0[OPS_ACC4(1)];

CF_13 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(1)]*u0_B0[OPS_ACC4(1)] + AVG_0_0_LEV_11*p_B0[OPS_ACC1(1)] +
AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(1)]*u0_B0[OPS_ACC4(1)] + AVG_0_0_LEV_12*p_B0[OPS_ACC1(1)]*u0_B0[OPS_ACC4(1)] +
AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(1)]*u0_B0[OPS_ACC4(1)];

CF_23 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(1)]*u0_B0[OPS_ACC4(1)] + AVG_0_0_LEV_21*p_B0[OPS_ACC1(1)] +
AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(1)]*u0_B0[OPS_ACC4(1)] + AVG_0_0_LEV_22*p_B0[OPS_ACC1(1)]*u0_B0[OPS_ACC4(1)] +
AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(1)]*u0_B0[OPS_ACC4(1)];

CF_04 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(2)]*u0_B0[OPS_ACC4(2)] + AVG_0_0_LEV_01*p_B0[OPS_ACC1(2)] +
AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(2)]*u0_B0[OPS_ACC4(2)] + AVG_0_0_LEV_02*p_B0[OPS_ACC1(2)]*u0_B0[OPS_ACC4(2)] +
AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(2)]*u0_B0[OPS_ACC4(2)];

CF_14 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(2)]*u0_B0[OPS_ACC4(2)] + AVG_0_0_LEV_11*p_B0[OPS_ACC1(2)] +

AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(2)]*u0_B0[OPS_ACC4(2)] + AVG_0_0_LEV_12*p_B0[OPS_ACC1(2)]*u0_B0[OPS_ACC4(2)] + AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(2)]*u0_B0[OPS_ACC4(2)];

CF_24 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(2)]*u0_B0[OPS_ACC4(2)] + AVG_0_0_LEV_21*p_B0[OPS_ACC1(2)] + AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(2)]*u0_B0[OPS_ACC4(2)] + AVG_0_0_LEV_22*p_B0[OPS_ACC1(2)]*u0_B0[OPS_ACC4(2)] + AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(2)]*u0_B0[OPS_ACC4(2)];

CF_05 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(3)]*u0_B0[OPS_ACC4(3)] + AVG_0_0_LEV_01*p_B0[OPS_ACC1(3)] + AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(3)]*u0_B0[OPS_ACC4(3)] + AVG_0_0_LEV_02*p_B0[OPS_ACC1(3)]*u0_B0[OPS_ACC4(3)] + AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(3)]*u0_B0[OPS_ACC4(3)];

CF_15 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(3)]*u0_B0[OPS_ACC4(3)] + AVG_0_0_LEV_11*p_B0[OPS_ACC1(3)] + AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(3)]*u0_B0[OPS_ACC4(3)] + AVG_0_0_LEV_12*p_B0[OPS_ACC1(3)]*u0_B0[OPS_ACC4(3)] + AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(3)]*u0_B0[OPS_ACC4(3)];

CF_25 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(3)]*u0_B0[OPS_ACC4(3)] + AVG_0_0_LEV_21*p_B0[OPS_ACC1(3)] + AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(3)]*u0_B0[OPS_ACC4(3)] + AVG_0_0_LEV_22*p_B0[OPS_ACC1(3)]*u0_B0[OPS_ACC4(3)] + AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(3)]*u0_B0[OPS_ACC4(3)];

CS_00 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(-2)] + AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(-2)] + AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(-2)];

CS_01 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(-1)] + AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(-1)] + AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(-1)];

CS_02 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(0)] + AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(0)] + AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(0)];

CS_03 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(1)] + AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(1)] + AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(1)];

CS_04 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(2)] + AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(2)] + AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(2)];

CS_05 = AVG_0_0_LEV_00*rho_B0[OPS_ACC2(3)] + AVG_0_0_LEV_01*rhou0_B0[OPS_ACC3(3)] + AVG_0_0_LEV_02*rhoE_B0[OPS_ACC0(3)];

CS_10 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(-2)] + AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(-2)] + AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(-2)];

CS_11 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(-1)] + AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(-1)] + AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(-1)];

CS_12 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(0)] + AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(0)] + AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(0)];

CS_13 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(1)] + AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(1)] + AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(1)];

CS_14 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(2)] + AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(2)] + AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(2)];

CS_15 = AVG_0_0_LEV_10*rho_B0[OPS_ACC2(3)] + AVG_0_0_LEV_11*rhou0_B0[OPS_ACC3(3)] + AVG_0_0_LEV_12*rhoE_B0[OPS_ACC0(3)];

CS_20 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(-2)] + AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(-2)] + AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(-2)];

CS_21 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(-1)] + AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(-1)] + AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(-1)];

CS_22 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(0)] + AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(0)] + AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(0)];

CS_23 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(1)] + AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(1)] + AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(1)];

CS_24 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(2)] + AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(2)] + AVG_0_0_LEV_22*rhoE_B0[OPS_ACC0(2)];

CS_25 = AVG_0_0_LEV_20*rho_B0[OPS_ACC2(3)] + AVG_0_0_LEV_21*rhou0_B0[OPS_ACC3(3)] +
AVG_0_0_LEV_22*rhouE_B0[OPS_ACC0(3)];

max_lambda_0_00 = fmax(fabs(a_B0[OPS_ACC5(0)] - u0_B0[OPS_ACC4(0)]), fmax(fabs(a_B0[OPS_ACC5(-2)] -
u0_B0[OPS_ACC4(-2)]), fmax(fabs(a_B0[OPS_ACC5(3)] - u0_B0[OPS_ACC4(3)]), fmax(fabs(a_B0[OPS_ACC5(-1)] -
u0_B0[OPS_ACC4(-1)]), fmax(fabs(a_B0[OPS_ACC5(1)] - u0_B0[OPS_ACC4(1)]), fabs(a_B0[OPS_ACC5(2)] -
u0_B0[OPS_ACC4(2)]))));

max_lambda_0_11 = fmax(fabs(u0_B0[OPS_ACC4(-2)]), fmax(fabs(u0_B0[OPS_ACC4(-1)]), fmax(fabs(u0_B0[OPS_ACC4(3)]),
fmax(fabs(u0_B0[OPS_ACC4(2)]), fmax(fabs(u0_B0[OPS_ACC4(1)]), fabs(u0_B0[OPS_ACC4(0)]))));

max_lambda_0_22 = fmax(fabs(a_B0[OPS_ACC5(3)] + u0_B0[OPS_ACC4(3)]), fmax(fabs(a_B0[OPS_ACC5(0)] +
u0_B0[OPS_ACC4(0)]), fmax(fabs(a_B0[OPS_ACC5(-2)] + u0_B0[OPS_ACC4(-2)]), fmax(fabs(a_B0[OPS_ACC5(1)] +
u0_B0[OPS_ACC4(1)]), fmax(fabs(a_B0[OPS_ACC5(-1)] + u0_B0[OPS_ACC4(-1)]), fabs(a_B0[OPS_ACC5(2)] +
u0_B0[OPS_ACC4(2)]))));

beta_0 = (rc1)*(CF_02 + CS_02*max_lambda_0_00)*((rc5)*(CF_02 + CS_02*max_lambda_0_00) - rc6*(CF_03 +
CS_03*max_lambda_0_00) + (rc7)*(CF_04 + CS_04*max_lambda_0_00)) + (rc1)*(CF_03 +
CS_03*max_lambda_0_00)*((rc8)*(CF_03 + CS_03*max_lambda_0_00) - rc9*(CF_04 + CS_04*max_lambda_0_00)) +
(rc10)*pow(CF_04 + CS_04*max_lambda_0_00, 2);

beta_1 = (rc1)*(CF_01 + CS_01*max_lambda_0_00)*((rc3)*(CF_01 + CS_01*max_lambda_0_00) - rc11*(CF_02 +
CS_02*max_lambda_0_00) + (rc12)*(CF_03 + CS_03*max_lambda_0_00)) + (rc1)*(CF_02 +
CS_02*max_lambda_0_00)*((rc11)*(CF_02 + CS_02*max_lambda_0_00) - rc11*(CF_03 + CS_03*max_lambda_0_00)) +
(rc10)*pow(CF_03 + CS_03*max_lambda_0_00, 2);

beta_2 = (rc1)*(CF_00 + CS_00*max_lambda_0_00)*((rc3)*(CF_00 + CS_00*max_lambda_0_00) + (rc7)*(CF_02 +
CS_02*max_lambda_0_00)) + (rc1)*(CF_01 + CS_01*max_lambda_0_00)*(-rc9*(CF_00 + CS_00*max_lambda_0_00) +
(rc8)*(CF_01 + CS_01*max_lambda_0_00) - rc6*(CF_02 + CS_02*max_lambda_0_00)) + (rc12)*pow(CF_02 +
CS_02*max_lambda_0_00, 2);

alpha_0 = (rc13)/pow(beta_0 + 1.0e-6, 2);

alpha_1 = (rc14)/pow(beta_1 + 1.0e-6, 2);

alpha_2 = (rc15)/pow(beta_2 + 1.0e-6, 2);

inv_alpha_sum = 1.0/(alpha_0 + alpha_1 + alpha_2);

omega_0 = alpha_0*inv_alpha_sum;

omega_1 = alpha_1*inv_alpha_sum;

omega_2 = alpha_2*inv_alpha_sum;

Recon_0 = Recon_0 + omega_0*((rc16)*(CF_02 + CS_02*max_lambda_0_00) + (rc17)*(CF_03 + CS_03*max_lambda_0_00) -
rc2*(CF_04 + CS_04*max_lambda_0_00)) + omega_1*(-rc2*(CF_01 + CS_01*max_lambda_0_00) + (rc17)*(CF_02 +
CS_02*max_lambda_0_00) + (rc16)*(CF_03 + CS_03*max_lambda_0_00)) + omega_2*((rc16)*(CF_00 + CS_00*max_lambda_0_00)
- rc18*(CF_01 + CS_01*max_lambda_0_00) + (rc19)*(CF_02 + CS_02*max_lambda_0_00));

rj0 = (pow(fabs((rc20)*omega_0 - 1.0), sensor_theta) + pow(fabs((rc5)*omega_1 - 1.0), sensor_theta) +
pow(fabs(10*omega_2 - 1.0), sensor_theta))/(pow(9.0, sensor_theta) + 2);

beta_0 = (rc1)*(CF_03 - CS_03*max_lambda_0_00)*((rc5)*(CF_03 - CS_03*max_lambda_0_00) - rc6*(CF_04 -
CS_04*max_lambda_0_00) + (rc7)*(CF_05 - CS_05*max_lambda_0_00)) + (rc1)*(CF_04 -
CS_04*max_lambda_0_00)*((rc8)*(CF_04 - CS_04*max_lambda_0_00) - rc9*(CF_05 - CS_05*max_lambda_0_00)) +
(rc10)*pow(CF_05 - CS_05*max_lambda_0_00, 2);

beta_1 = (rc1)*(CF_02 - CS_02*max_lambda_0_00)*((rc3)*(CF_02 - CS_02*max_lambda_0_00) - rc11*(CF_03 -
CS_03*max_lambda_0_00) + (rc12)*(CF_04 - CS_04*max_lambda_0_00)) + (rc1)*(CF_03 -
CS_03*max_lambda_0_00)*((rc11)*(CF_03 - CS_03*max_lambda_0_00) - rc11*(CF_04 - CS_04*max_lambda_0_00)) +
(rc10)*pow(CF_04 - CS_04*max_lambda_0_00, 2);

beta_2 = (rc1)*(CF_01 - CS_01*max_lambda_0_00)*((rc3)*(CF_01 - CS_01*max_lambda_0_00) - rc9*(CF_02 -
CS_02*max_lambda_0_00) + (rc7)*(CF_03 - CS_03*max_lambda_0_00)) + (rc1)*(CF_02 -
CS_02*max_lambda_0_00)*((rc8)*(CF_02 - CS_02*max_lambda_0_00) - rc6*(CF_03 - CS_03*max_lambda_0_00)) +
(rc12)*pow(CF_03 - CS_03*max_lambda_0_00, 2);

alpha_0 = (rc15)/pow(beta_0 + 1.0e-6, 2);

```

alpha_1 = (rc14)/pow(beta_1 + 1.0e-6, 2);

alpha_2 = (rc13)/pow(beta_2 + 1.0e-6, 2);

inv_alpha_sum = 1.0/(alpha_0 + alpha_1 + alpha_2);

omega_0 = alpha_0*inv_alpha_sum;

omega_1 = alpha_1*inv_alpha_sum;

omega_2 = alpha_2*inv_alpha_sum;

Recon_0 = Recon_0 + omega_0*((rc19)*(CF_03 - CS_03*max_lambda_0_00) - rc18*(CF_04 - CS_04*max_lambda_0_00) +
(rc16)*(CF_05 - CS_05*max_lambda_0_00)) + omega_1*((rc16)*(CF_02 - CS_02*max_lambda_0_00) + (rc17)*(CF_03 -
CS_03*max_lambda_0_00) - rc2*(CF_04 - CS_04*max_lambda_0_00)) + omega_2*(-rc2*(CF_01 - CS_01*max_lambda_0_00) +
(rc17)*(CF_02 - CS_02*max_lambda_0_00) + (rc16)*(CF_03 - CS_03*max_lambda_0_00));

rj0 = fmax(rj0, (pow(fabs(10*omega_0 - 1.0), sensor_theta) + pow(fabs((rc5)*omega_1 - 1.0), sensor_theta) +
pow(fabs((rc20)*omega_2 - 1.0), sensor_theta))/(pow(9.0, sensor_theta) + 2));

beta_0 = (rc1)*(CF_12 + CS_12*max_lambda_0_11)*((rc5)*(CF_12 + CS_12*max_lambda_0_11) - rc6*(CF_13 +
CS_13*max_lambda_0_11) + (rc7)*(CF_14 + CS_14*max_lambda_0_11)) + (rc1)*(CF_13 +
CS_13*max_lambda_0_11)*((rc8)*(CF_13 + CS_13*max_lambda_0_11) - rc9*(CF_14 + CS_14*max_lambda_0_11)) +
(rc10)*pow(CF_14 + CS_14*max_lambda_0_11, 2);

beta_1 = (rc1)*(CF_11 + CS_11*max_lambda_0_11)*((rc3)*(CF_11 + CS_11*max_lambda_0_11) - rc11*(CF_12 +
CS_12*max_lambda_0_11) + (rc12)*(CF_13 + CS_13*max_lambda_0_11)) + (rc1)*(CF_12 +
CS_12*max_lambda_0_11)*((rc11)*(CF_12 + CS_12*max_lambda_0_11) - rc11*(CF_13 + CS_13*max_lambda_0_11)) +
(rc10)*pow(CF_13 + CS_13*max_lambda_0_11, 2);

beta_2 = (rc1)*(CF_10 + CS_10*max_lambda_0_11)*((rc3)*(CF_10 + CS_10*max_lambda_0_11) + (rc7)*(CF_12 +
CS_12*max_lambda_0_11)) + (rc1)*(CF_11 + CS_11*max_lambda_0_11)*(-rc9*(CF_10 + CS_10*max_lambda_0_11) +
(rc8)*(CF_11 + CS_11*max_lambda_0_11) - rc6*(CF_12 + CS_12*max_lambda_0_11)) + (rc12)*pow(CF_12 +
CS_12*max_lambda_0_11, 2);

alpha_0 = (rc13)/pow(beta_0 + 1.0e-6, 2);

alpha_1 = (rc14)/pow(beta_1 + 1.0e-6, 2);

alpha_2 = (rc15)/pow(beta_2 + 1.0e-6, 2);

inv_alpha_sum = 1.0/(alpha_0 + alpha_1 + alpha_2);

omega_0 = alpha_0*inv_alpha_sum;

omega_1 = alpha_1*inv_alpha_sum;

omega_2 = alpha_2*inv_alpha_sum;

Recon_1 = Recon_1 + omega_0*((rc16)*(CF_12 + CS_12*max_lambda_0_11) + (rc17)*(CF_13 + CS_13*max_lambda_0_11) -
rc2*(CF_14 + CS_14*max_lambda_0_11)) + omega_1*(-rc2*(CF_11 + CS_11*max_lambda_0_11) + (rc17)*(CF_12 +
CS_12*max_lambda_0_11) + (rc16)*(CF_13 + CS_13*max_lambda_0_11)) + omega_2*((rc16)*(CF_10 + CS_10*max_lambda_0_11)
- rc18*(CF_11 + CS_11*max_lambda_0_11) + (rc19)*(CF_12 + CS_12*max_lambda_0_11));

rj1 = (pow(fabs((rc20)*omega_0 - 1.0), sensor_theta) + pow(fabs((rc5)*omega_1 - 1.0), sensor_theta) +
pow(fabs(10*omega_2 - 1.0), sensor_theta))/(pow(9.0, sensor_theta) + 2);

beta_0 = (rc1)*(CF_13 - CS_13*max_lambda_0_11)*((rc5)*(CF_13 - CS_13*max_lambda_0_11) - rc6*(CF_14 -
CS_14*max_lambda_0_11) + (rc7)*(CF_15 - CS_15*max_lambda_0_11)) + (rc1)*(CF_14 -
CS_14*max_lambda_0_11)*((rc8)*(CF_14 - CS_14*max_lambda_0_11) - rc9*(CF_15 - CS_15*max_lambda_0_11)) +
(rc10)*pow(CF_15 - CS_15*max_lambda_0_11, 2);

beta_1 = (rc1)*(CF_12 - CS_12*max_lambda_0_11)*((rc3)*(CF_12 - CS_12*max_lambda_0_11) - rc11*(CF_13 -
CS_13*max_lambda_0_11) + (rc12)*(CF_14 - CS_14*max_lambda_0_11)) + (rc1)*(CF_13 -
CS_13*max_lambda_0_11)*((rc11)*(CF_13 - CS_13*max_lambda_0_11) - rc11*(CF_14 - CS_14*max_lambda_0_11)) +
(rc10)*pow(CF_14 - CS_14*max_lambda_0_11, 2);

beta_2 = (rc1)*(CF_11 - CS_11*max_lambda_0_11)*((rc3)*(CF_11 - CS_11*max_lambda_0_11) - rc9*(CF_12 -
CS_12*max_lambda_0_11) + (rc7)*(CF_13 - CS_13*max_lambda_0_11)) + (rc1)*(CF_12 -

```

```

CS_12*max_lambda_0_11)*(rc8)*(CF_12 - CS_12*max_lambda_0_11) - rc6*(CF_13 - CS_13*max_lambda_0_11)) +
(rc12)*pow(CF_13 - CS_13*max_lambda_0_11, 2);

alpha_0 = (rc15)/pow(beta_0 + 1.0e-6, 2);

alpha_1 = (rc14)/pow(beta_1 + 1.0e-6, 2);

alpha_2 = (rc13)/pow(beta_2 + 1.0e-6, 2);

inv_alpha_sum = 1.0/(alpha_0 + alpha_1 + alpha_2);

omega_0 = alpha_0*inv_alpha_sum;

omega_1 = alpha_1*inv_alpha_sum;

omega_2 = alpha_2*inv_alpha_sum;

Recon_1 = Recon_1 + omega_0*((rc19)*(CF_13 - CS_13*max_lambda_0_11) - rc18*(CF_14 - CS_14*max_lambda_0_11) +
(rc16)*(CF_15 - CS_15*max_lambda_0_11)) + omega_1*((rc16)*(CF_12 - CS_12*max_lambda_0_11) + (rc17)*(CF_13 -
CS_13*max_lambda_0_11) - rc2*(CF_14 - CS_14*max_lambda_0_11)) + omega_2*(-rc2*(CF_11 - CS_11*max_lambda_0_11) +
(rc17)*(CF_12 - CS_12*max_lambda_0_11) + (rc16)*(CF_13 - CS_13*max_lambda_0_11));

rj1 = fmax(rj1, (pow(fabs(10*omega_0 - 1.0), sensor_theta) + pow(fabs((rc5)*omega_1 - 1.0), sensor_theta) +
pow(fabs((rc20)*omega_2 - 1.0), sensor_theta))/(pow(9.0, sensor_theta) + 2));

beta_0 = (rc1)*(CF_22 + CS_22*max_lambda_0_22)*((rc5)*(CF_22 + CS_22*max_lambda_0_22) - rc6*(CF_23 +
CS_23*max_lambda_0_22) + (rc7)*(CF_24 + CS_24*max_lambda_0_22)) + (rc1)*(CF_23 +
CS_23*max_lambda_0_22)*((rc8)*(CF_23 + CS_23*max_lambda_0_22) - rc9*(CF_24 + CS_24*max_lambda_0_22)) +
(rc10)*pow(CF_24 + CS_24*max_lambda_0_22, 2);

beta_1 = (rc1)*(CF_21 + CS_21*max_lambda_0_22)*((rc3)*(CF_21 + CS_21*max_lambda_0_22) - rc11*(CF_22 +
CS_22*max_lambda_0_22) + (rc12)*(CF_23 + CS_23*max_lambda_0_22)) + (rc1)*(CF_22 +
CS_22*max_lambda_0_22)*((rc11)*(CF_22 + CS_22*max_lambda_0_22) - rc11*(CF_23 + CS_23*max_lambda_0_22)) +
(rc10)*pow(CF_23 + CS_23*max_lambda_0_22, 2);

beta_2 = (rc1)*(CF_20 + CS_20*max_lambda_0_22)*((rc3)*(CF_20 + CS_20*max_lambda_0_22) + (rc7)*(CF_22 +
CS_22*max_lambda_0_22)) + (rc1)*(CF_21 + CS_21*max_lambda_0_22)*(-rc9*(CF_20 + CS_20*max_lambda_0_22) +
(rc8)*(CF_21 + CS_21*max_lambda_0_22) - rc6*(CF_22 + CS_22*max_lambda_0_22)) + (rc12)*pow(CF_22 +
CS_22*max_lambda_0_22, 2);

alpha_0 = (rc13)/pow(beta_0 + 1.0e-6, 2);

alpha_1 = (rc14)/pow(beta_1 + 1.0e-6, 2);

alpha_2 = (rc15)/pow(beta_2 + 1.0e-6, 2);

inv_alpha_sum = 1.0/(alpha_0 + alpha_1 + alpha_2);

omega_0 = alpha_0*inv_alpha_sum;

omega_1 = alpha_1*inv_alpha_sum;

omega_2 = alpha_2*inv_alpha_sum;

Recon_2 = Recon_2 + omega_0*((rc16)*(CF_22 + CS_22*max_lambda_0_22) + (rc17)*(CF_23 + CS_23*max_lambda_0_22) -
rc2*(CF_24 + CS_24*max_lambda_0_22)) + omega_1*(-rc2*(CF_21 + CS_21*max_lambda_0_22) + (rc17)*(CF_22 +
CS_22*max_lambda_0_22) + (rc16)*(CF_23 + CS_23*max_lambda_0_22)) + omega_2*((rc16)*(CF_20 + CS_20*max_lambda_0_22)
- rc18*(CF_21 + CS_21*max_lambda_0_22) + (rc19)*(CF_22 + CS_22*max_lambda_0_22));

rj2 = (pow(fabs((rc20)*omega_0 - 1.0), sensor_theta) + pow(fabs((rc5)*omega_1 - 1.0), sensor_theta) +
pow(fabs(10*omega_2 - 1.0), sensor_theta))/(pow(9.0, sensor_theta) + 2);

beta_0 = (rc1)*(CF_23 - CS_23*max_lambda_0_22)*((rc5)*(CF_23 - CS_23*max_lambda_0_22) - rc6*(CF_24 -
CS_24*max_lambda_0_22) + (rc7)*(CF_25 - CS_25*max_lambda_0_22)) + (rc1)*(CF_24 -
CS_24*max_lambda_0_22)*((rc8)*(CF_24 - CS_24*max_lambda_0_22) - rc9*(CF_25 - CS_25*max_lambda_0_22)) +
(rc10)*pow(CF_25 - CS_25*max_lambda_0_22, 2);

beta_1 = (rc1)*(CF_22 - CS_22*max_lambda_0_22)*((rc3)*(CF_22 - CS_22*max_lambda_0_22) - rc11*(CF_23 -
CS_23*max_lambda_0_22) + (rc12)*(CF_24 - CS_24*max_lambda_0_22)) + (rc1)*(CF_23 -
CS_23*max_lambda_0_22)*((rc11)*(CF_23 - CS_23*max_lambda_0_22) - rc11*(CF_24 - CS_24*max_lambda_0_22)) +

```

```

(rc10)*pow(CF_24 - CS_24*max_lambda_0_22, 2);

beta_2 = (rc1)*(CF_21 - CS_21*max_lambda_0_22)*((rc3)*(CF_21 - CS_21*max_lambda_0_22) - rc9*(CF_22 -
CS_22*max_lambda_0_22) + (rc7)*(CF_23 - CS_23*max_lambda_0_22)) + (rc1)*(CF_22 -
CS_22*max_lambda_0_22)*((rc8)*(CF_22 - CS_22*max_lambda_0_22) - rc6*(CF_23 - CS_23*max_lambda_0_22)) +
(rc12)*pow(CF_23 - CS_23*max_lambda_0_22, 2);

alpha_0 = (rc15)/pow(beta_0 + 1.0e-6, 2);

alpha_1 = (rc14)/pow(beta_1 + 1.0e-6, 2);

alpha_2 = (rc13)/pow(beta_2 + 1.0e-6, 2);

inv_alpha_sum = 1.0/(alpha_0 + alpha_1 + alpha_2);

omega_0 = alpha_0*inv_alpha_sum;

omega_1 = alpha_1*inv_alpha_sum;

omega_2 = alpha_2*inv_alpha_sum;

Recon_2 = Recon_2 + omega_0*((rc19)*(CF_23 - CS_23*max_lambda_0_22) - rc18*(CF_24 - CS_24*max_lambda_0_22) +
(rc16)*(CF_25 - CS_25*max_lambda_0_22)) + omega_1*((rc16)*(CF_22 - CS_22*max_lambda_0_22) + (rc17)*(CF_23 -
CS_23*max_lambda_0_22) - rc2*(CF_24 - CS_24*max_lambda_0_22)) + omega_2*(-rc2*(CF_21 - CS_21*max_lambda_0_22) +
(rc17)*(CF_22 - CS_22*max_lambda_0_22) + (rc16)*(CF_23 - CS_23*max_lambda_0_22));

rj2 = fmax(rj2, (pow(fabs(10*omega_0 - 1.0), sensor_theta) + pow(fabs((rc5)*omega_1 - 1.0), sensor_theta) +
pow(fabs((rc20)*omega_2 - 1.0), sensor_theta))/(pow(9.0, sensor_theta) + 2));

Recon_0 = rj0*(Recon_0 - rc21*(CF_00 - 8*CF_01 + 37*CF_02 + 37*CF_03 - 8*CF_04 + CF_05));

Recon_1 = rj1*(Recon_1 - rc21*(CF_10 - 8*CF_11 + 37*CF_12 + 37*CF_13 - 8*CF_14 + CF_15));

Recon_2 = rj2*(Recon_2 - rc21*(CF_20 - 8*CF_21 + 37*CF_22 + 37*CF_23 - 8*CF_24 + CF_25));

wk0_B0[OPS_ACC6(0)] = Recon_0 + Recon_1 + Recon_2;

wk1_B0[OPS_ACC8(0)] = AVG_0_u0*Recon_1 + Recon_0*(-AVG_0_a + AVG_0_u0) + Recon_2*(AVG_0_a + AVG_0_u0);

wk2_B0[OPS_ACC7(0)] = (rc1)*pow(AVG_0_u0, 2)*Recon_1 + Recon_0*(rcinv22*pow(AVG_0_a, 2) - AVG_0_a*AVG_0_u0 +
(rc1)*pow(AVG_0_u0, 2)) + Recon_2*(rcinv22*pow(AVG_0_a, 2) + AVG_0_a*AVG_0_u0 + (rc1)*pow(AVG_0_u0, 2));

}

void openslibblock00Kernel029(const double *wk0_B0, const double *Mach_sensor_B0, const double *wk2_B0, const double
*wk1_B0, double *rho_B0, double *rho_u0_B0, double *rhoE_B0, const int *idx)
{
double Wall = 0.0;
double Grid_0 = 0.0;
Grid_0 = idx[0];

Wall = 1;

rho_B0[OPS_ACC4(0)] = -dt*inv_0*shock_filter_control*Wall*(wk0_B0[OPS_ACC0(0)] -
wk0_B0[OPS_ACC0(-1)])*Mach_sensor_B0[OPS_ACC1(0)] + rho_B0[OPS_ACC4(0)];

rho_u0_B0[OPS_ACC5(0)] = -dt*inv_0*shock_filter_control*Wall*(wk1_B0[OPS_ACC3(0)] -
wk1_B0[OPS_ACC3(-1)])*Mach_sensor_B0[OPS_ACC1(0)] + rho_u0_B0[OPS_ACC5(0)];

rhoE_B0[OPS_ACC6(0)] = -dt*inv_0*shock_filter_control*Wall*(-wk2_B0[OPS_ACC2(-1)] +
wk2_B0[OPS_ACC2(0)])*Mach_sensor_B0[OPS_ACC1(0)] + rhoE_B0[OPS_ACC6(0)];

}

#endif

```