This document describes the details about how to use the DSMC-SG code.

0. Request to Users

Any publication by using the ***DSMC-SG*** code or modified version, please cite the following paper for acknowledgment and also e-mail a copy of the paper or DOI address to Prof. J.-S. Wu at chongsin@faculty.nctu.edu.tw:

> C.-C. Su, M.R. Smith, F.-A. Kuo, J.-S. Wu, C.-W. Hsieh, K.-C. Tseng, "Large-scale Simulations on Multiple Graphics Processing Units (GPUs) for the Direct Simulation Monte Carlo Method," Journal of Computational Physics, Vol. 231, pp. 7932-7958, 2012. http://dx.doi.org/10.1016/j.jcp.2012.07.038

1. Introduction

DSMC-SG is a two-dimensional direct simulation Monte Carlo (DSMC) [1] code on single graphics processing unit (GPU) which was developed by Prof. Wu's group (APPL: http://www.me.nctu.edu.tw/appl/). The ***DSMC-SG*** is a computational tool for simulating gas flow dynamics in rarefied gas regime using Cartesian structured grids with single species (only single atom gas). In this code, the nVIDIA's CUDA [2] is used to accelerate all components of the DSMC method, including particle movement, indexing, collision, and sampling. About the details of the algorithm, please refer to our recent JCP paper [3]. Instructions on how to use the ***DSMC-SG*** are described next.

2. How to use the DSMC-SG code

For the use of the ***DSMC-SG*** code, one needs to set up the following three steps: simulation/initial conditions (*input.txt*), boundary conditions (*dsmcsg_init.cpp*), and species conditions (*parameter.h*). The unit of all parameters is SI system. First, this code is just used to simulate the flow field with single species and monatomic gas without surface sampling. Two examples in supersonic flow are demonstrated in the following for the users to

understand the setup procedures. These two include supersonic flow over a block and Mach reflection in a parallel channel, and are described in the following in turn.

2.1 Supersonic flow over a block

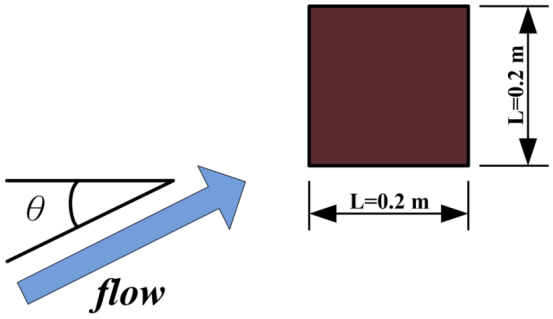


Fig. 1. Supersonic flow over a block.

The first test case is a two-dimensional supersonic flow over a rectangular block as shown in Fig. 1. Free-stream conditions include: argon gas, 300 K, M=4.0 with a 30 degree of angle of attack over a diffusely reflecting block with a fixed temperature of 1,200 K. The block is square with sides of length L=0.2 m. The corresponding free-stream Knudsen number is 0.02 based on the free-stream mean free path (λ_∞=0.004 m) and the length (L=0.2 m). The properties setup of argon gas is in the *parameter.h* file as shown in Fig. 2.

```
9   // Argon gas properties.
10  #define DIAMETER              4.17e-10
11  #define REF_TEMP              273.
12  #define VIS_TEMP_INDEX        0.81
13  #define MASS                  6.63e-26
```

Fig. 2. The setup of species in the file "*parameter.h*"

The detailed setup of simulation/initial conditions in *DSMC-SG* code is described in the file "*input.txt*". Fig. 3 shows the setup of this case in the *input.txt* file. In this case, the setup is that the range of simulation domain is from 0 to 0.8 meter (see X_LOWER and X_HIGHER) in the x-direction, and from 0 to 1 meter (Y_LOWER and Y_HIGHER) in the y-direction with 400 and 500 cell numbers (X_CELL_NUMBER and Y_CELL_NUMBER) in the x-/y-direction, respectively. The initial number of particles per cell and maximum number of

simulated particles are set as 15 and 5,000,000 (PARTICLE_PER_CELL and MAX_PARTICLE_NUMBER). The maximal number of total time steps is 60,000 (NUMBER_OF_TIMESTEP), and the period of sampling is from 20,000 to 60,000 time steps (i.e. sampling time begin at timesteps of 20,000, SAMPLING_TIME). Next, we introduce how to set up boundary conditions with a solid block inside the computational domain.

```
#=============================================================================
#
# This file is a input file which is used to set up simulation, initial condiaitons.
#
#=============================================================================


# SAMPLING_TIME:        start to sampled timestep number.
# NUMBER_OF_TIMESTEP:   total executed number of timesteps.
# PARTICLE_PER_CELL:    initial number of particle per cell.
# MAX_PARTICLE_NUMBER:  maximum number of simulation particles.
#=============================================================
SAMPLING_TIME                 = 20000
NUMBER_OF_TIMESTEP            = 60000
PARTICLE_PER_CELL             = 15.
MAX_PARTICLE_NUMBER           = 5000000


# Setup of simulation domain region (The unit is meter).
#=============================================================
# X_LOWER:              domain region in x-direction.
# X_HIGHER:             domain region in x-direction.
# Y_LOWER:              domain region in y-direction.
# Y_HIGHER:             domain region in y-direction.
# X_CELL_NUMBER:        number of cells in x-direction.
# Y_CELL_NUMBER:        number of cells in y-direction.
#=============================================================
X_LOWER                       = 0.
X_HIGHER                      = 0.8
Y_LOWER                       = 0.
Y_HIGHER                      = 1.
X_CELL_NUMBER                 = 400
Y_CELL_NUMBER                 = 500


# Setup of initial flow condition.
#=============================================================
# X_VELOCITY:           initial velocity in x-direction.
# Y_VELOCITY            initial velocity in y-direction.
# Z_VELOCITY            initial velocity in z-direction.
# NUMBER_DENSITY        initial number density.
# TEMPERATURE           initial temperature.
#=============================================================
X_VELOCITY                    = 1117.14
Y_VELOCITY                    = 644.98
Z_VELOCITY                    = 0.
NUMBER_DENSITY                = 3.24E+20
TEMPERATURE                   = 300.


# Setup of number of boundary/internal blocks, and inlet faces.
#=============================================================
# BOUNDARY_BLOCK_NUMBER:  number of blocks.
# INTERNAL_BLOCK_NUMBER:  number of internal blocks.
# INLET_CELL_FACE_NUMBER: number of inlet faces.
#=============================================================
BOUNDARY_BLOCK_NUMBER         = 4
INTERNAL_BLOCK_NUMBER         = 1
INLET_CELL_FACE_NUMBER        = 1800
```

Fig. 3. The setup for supersonic flow over a block problem in the file "*input.txt*".

The setup of boundary conditions and internal block in the *DSMC-SG* code is described in the function "*BoundaryCondition()*" in the file *dsmcsg_init.cpp*. Further, for the setup of properties of each boundary condition and internal block in this case, one needs to modify the C++ class *h_Block[]* in the function "*BoundaryCondition()*". Fig. 4 shows the step of

boundary conditions and internal block by modifying the *h_Block[]* in this case. The *h_block[].Type* is used to define the type of boundary or internal block (Types of BCs include -3: inlet, -4: vacuum, -21: fully-specular reflection, -22: fully-diffusive reflection). Here, setup of number of domain boundaries and internal blocks (i.e. the size of array *h_Block[]*) is described in the file *input.txt* (BOUNDARY_BLOCK_NUMBER and INTERNAL_BLOCK_NUMBER). In addition, setup of number of the inlet faces (INLET_CELL_FACE_NUMBER) is described in *input.txt*. Note: coordinates of all nodes of internal blocks must be assigned in the counterclockwise direction (see setup of *h_Block[4].Node[0-3]*), and the geometry of internal block can be triangular (*h_Block[].NodeNum = 3*) or quadrilateral (*h_Block[].NodeNum = 4*). Next, we will introduce how to compile and execute the *DSMC-SG* code.

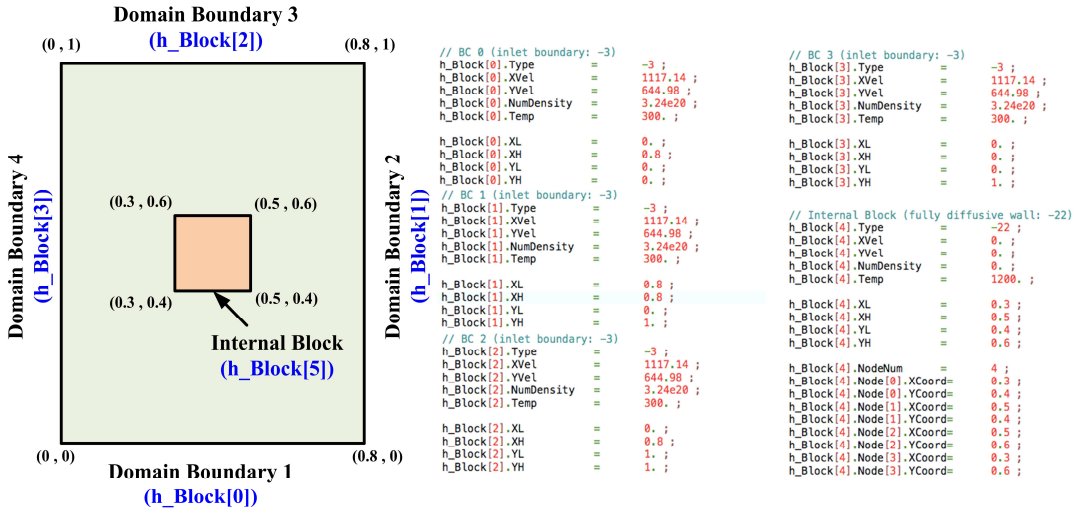


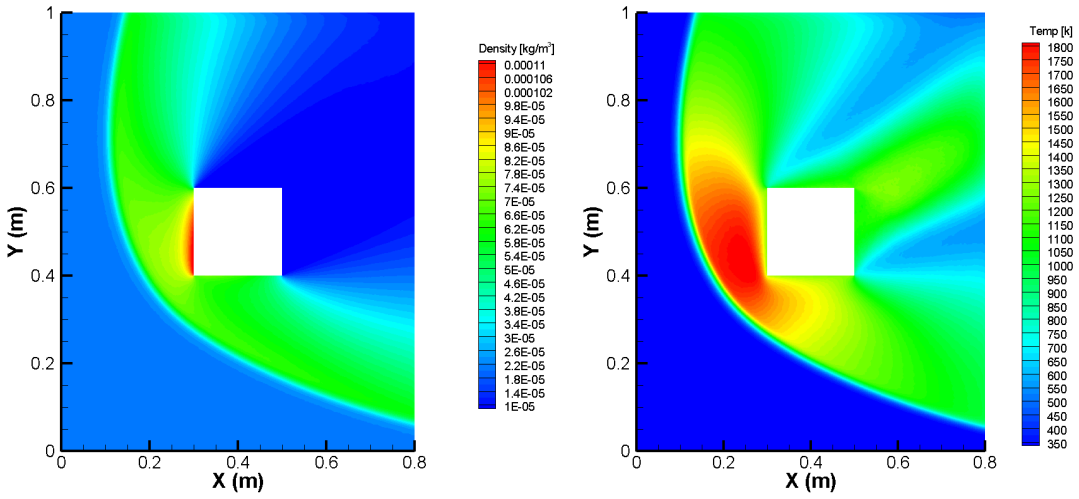
Fig. 4. The setup of properties of domain boundary and internal block in the file "*dsmcsg_init.cpp*" for supersonic flow over a block problem.

To compile the *DSMC-SG* code, one needs to use "*make*" to activate "*Makefile*". You can modify this file based on your environment of computer to compile the *DSMC-SG* code. After compilation, two executable files are generated, which include ***dsmcsg*** and ***dsmcsg_postprocessor***. The ***dsmcsg*** is a DSMC executable file that can be run directly on a single GPU, and the ***dsmcsg_postprocessor*** is a post-processor of *DSMC-SG* code, which is

used to convert DSMC simulation results (Result.dat) to format of Tecplot software (Result-tec.dat) for visualization. Fig.6 shows the procedures of compilation and execution of the program. Fig. 7 shows the contours of density and temperature simulated in this case. The simulation is executed on a CPU of AMD Phenom II X6 1090T and a GPU of nVIDIA GeForce GTX 590 with CUDA 4.0. The execution time is approximately 42 minutes (SimTime.dat).

```
ccs@appl-gpu-1:~/Code/develop/DSMC-SG-online$ make
g++ -O3 -c -c dsmcsg_main.cpp -I/usr/local/cuda/include
g++ -O3 -c -c dsmcsg_class.cpp
g++ -O3 -c -c dsmcsg_init.cpp
g++ -O3 -c -c dsmcsg_output.cpp
g++ -O3 -c -c dsmcsg_readfile.cpp
nvcc -DUNIX -O3 -c --use_fast_math -arch sm_20 -c dsmcsg_cudafunction.cu -I/usr/local/cuda/include
nvcc -DUNIX -O3 -c --use_fast_math -arch sm_20 -c scanLargeArray_kernel.cu -I/usr/local/cuda/include
g++ -fPIC -o dsmcsg dsmcsg_main.o dsmcsg_class.o dsmcsg_init.o dsmcsg_output.o dsmcsg_readfile.o dsmcsg_cudafunction.o scanLargeArray_kernel.o -L/usr/local/cu
da/lib64 -lcudart
g++ -O3 -c -c dsmcsg_postprocessor.cpp
g++ -fPIC -o dsmcsg_postprocessor dsmcsg_postprocessor.o
ccs@appl-gpu-1:~/Code/develop/DSMC-SG-online$ ls
dsmcsg            dsmcsg_cudafunction.cu  dsmcsg_init.h    dsmcsg_output.cpp     dsmcsg_postprocessor.cpp  dsmcsg_readfile.o  scanLargeArray_kernel.cu
dsmcsg_class.cpp  dsmcsg_cudafunction.h   dsmcsg_init.o    dsmcsg_output.h       dsmcsg_postprocessor.o    input.txt          scanLargeArray_kernel.o
dsmcsg_class.h    dsmcsg_cudafunction.o   dsmcsg_main.cpp  dsmcsg_output.o       dsmcsg_readfile.cpp       Makefile
dsmcsg_class.o    dsmcsg_init.cpp         dsmcsg_main.o    dsmcsg_postprocessor  dsmcsg_readfile.h         parameter.h
ccs@appl-gpu-1:~/Code/develop/DSMC-SG-online$ ./dsmcsg > out.dat
ccs@appl-gpu-1:~/Code/develop/DSMC-SG-online$ mv Result-60000.dat Result.dat
ccs@appl-gpu-1:~/Code/develop/DSMC-SG-online$ ./dsmcsg_postprocessor
ccs@appl-gpu-1:~/Code/develop/DSMC-SG-online$ ls
cell.dat              dsmcsg_cudafunction.h  dsmcsg_main.o            dsmcsg_postprocessor.o  neighbor.dat     Result-32000.dat  scanLargeArray_kernel.cu
dsmcsg                dsmcsg_cudafunction.o  dsmcsg_output.cpp        dsmcsg_readfile.cpp     node.dat         Result-40000.dat  scanLargeArray_kernel.o
dsmcsg_class.cpp      dsmcsg_init.cpp        dsmcsg_output.h          dsmcsg_readfile.h       out.dat          Result-48000.dat  SimTime.dat
dsmcsg_class.h        dsmcsg_init.h          dsmcsg_output.o          dsmcsg_readfile.o       parameter.h      Result-56000.dat
dsmcsg_class.o        dsmcsg_init.o          dsmcsg_postprocessor     input.txt               particles.dat    Result.dat
dsmcsg_cudafunction.cu  dsmcsg_main.cpp      dsmcsg_postprocessor.cpp  Makefile               Result-24000.dat  Result-tec.dat
ccs@appl-gpu-1:~/Code/develop/DSMC-SG-online$
```

Fig. 6. The step of compilation and execution of DSMC-SG code.



(a) (b)

Fig. 7. Contours of properties of supersonic flow over a block: (a) density and (b) temperature.

2.2 Mach reflection problem

The second test case is a Mach reflection problem which involves the high-speed flow (M = 5.0) between two symmetrical wedges with angle θ (=30 degree) and L/W (=1.5) as shown in Fig. 8. Since the problem is symmetric, we only simulate the upper half of the problem. The initial conditions are argon gas at temperature 300K with a uniform flow speed of 1612.45 m/s moving from the left. The initial number density is 1.294×10^{20} particles/m^3 and the wall temperature is fixed at 600 K – these conditions correspond to a free-stream Knudsen number of 0.01 based on the free-stream mean free path (λ_∞=0.01 m) and the length of the inclined wall (W=1 m). Fig. 8 shows the corresponding setup of boundary conditions and internal block in the file *dsmcsg_init.cpp*. Fig. 9 shows the contours of density and temperature obtained in this simulation.

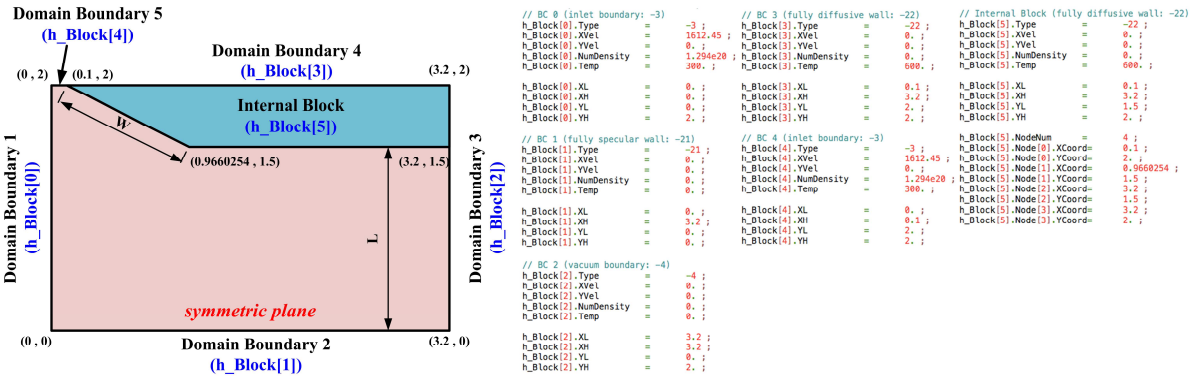
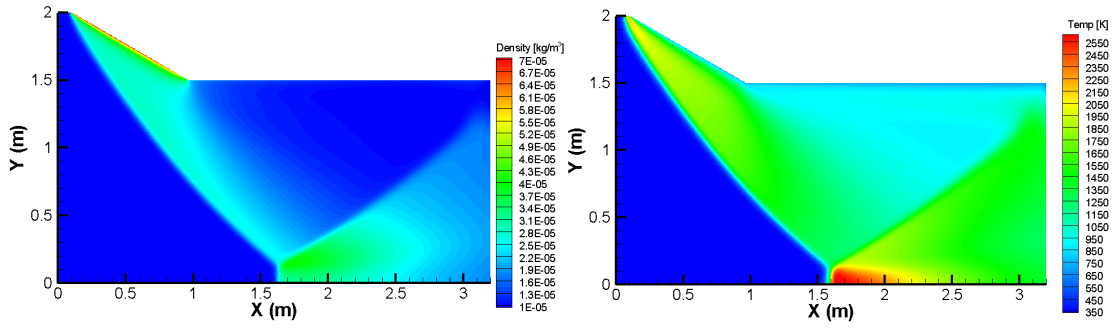


Fig. 8. The setup of properties of domain boundary and internal block in the file "*dsmcsg_init.cpp*" for Mach reflection problem.



(a) (b)

Fig. 9. Contours of properties of Mach reflection problem: (a) density and (b) temperature.

References

[1] G.A. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Clarendon Press, Oxford, 1994.

[2] NVIDIA Crop. NVIDIA CUDA C Programming Guide Version 4.0, 2011.

[3] C.-C. Su, M.R. Smith, F.-A. Kuo, J.-S. Wu, C.-W. Hsieh, K.-C. Tseng, Large-scale simulations on multiple Graphics Processing Units (GPUs) for the direct simulation Monte Carlo method, Journal of Computational Physics, Vol. 231, pp. 7932-7958, 2012.