SPARTA: Improving Computational Efficiency

Michael A. Gallis

Engineering Sciences Center
Sandia National Laboratories
Albuquerque, New Mexico, USA

magalli@sandia.gov

Short Course

Direct Simulation Monte Carlo 2023:

Theory, Methods, and Applications

September 24-27, 2015; Santa Fe, NM, USA

U.S. DEPARTMENT OF ENERGY

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

NNSA National Nuclear Security Administration

DSMC Procedures

- Since the original inception of the DSMC algorithm (1963), there have been no major changes to it.
- DSMC has been shown to provide solutions to the Boltzmann equation more accurately than any other numerical method.
- The DSMC algorithm has been criticized as being computationally inefficient.
- 15 years ago a series of changes to the DSMC algorithm (“*sophisticated DSMC*”) were proposed by Bird that improve efficiency while maintaining the accuracy of the method.
- *What are the tools SPARTA offers to improve computational efficiency?*

DSMC and the Boltzmann Equation

“Given the physical basis of the DSMC method, the existence, uniqueness and convergence issues that are important in the traditional mathematical analysis of equations are largely irrelevant.”

Molecular Dynamics and the Direct Simulation of Gas Flows, Chapter 10.

DSMC Numerical Error

Four parameters control DSMC numerical error:

- Sample size per cell (M_c) → statistical error
- Simulators per cell (N_c)
- Cell size (Δx)
- Time step (Δt)

→ discretization error

Early DSMC users followed rule-of-thumb guidelines:

- Sample enough to drive statistical error down
- Keep time step smaller than ~1/4 mean collision time
- Keep cell size smaller than ~1/3 mean free path
- Use a minimum of ~20 simulators per cell

Cell-Size Error

Error related to cell width, Δx

- **Collision partners selected from anywhere in same cell**
- Some potential partners move into adjacent cells, some invalid partners move into the same cell

Green-Kubo theoretical analysis (Alexander *et al*., 1998)

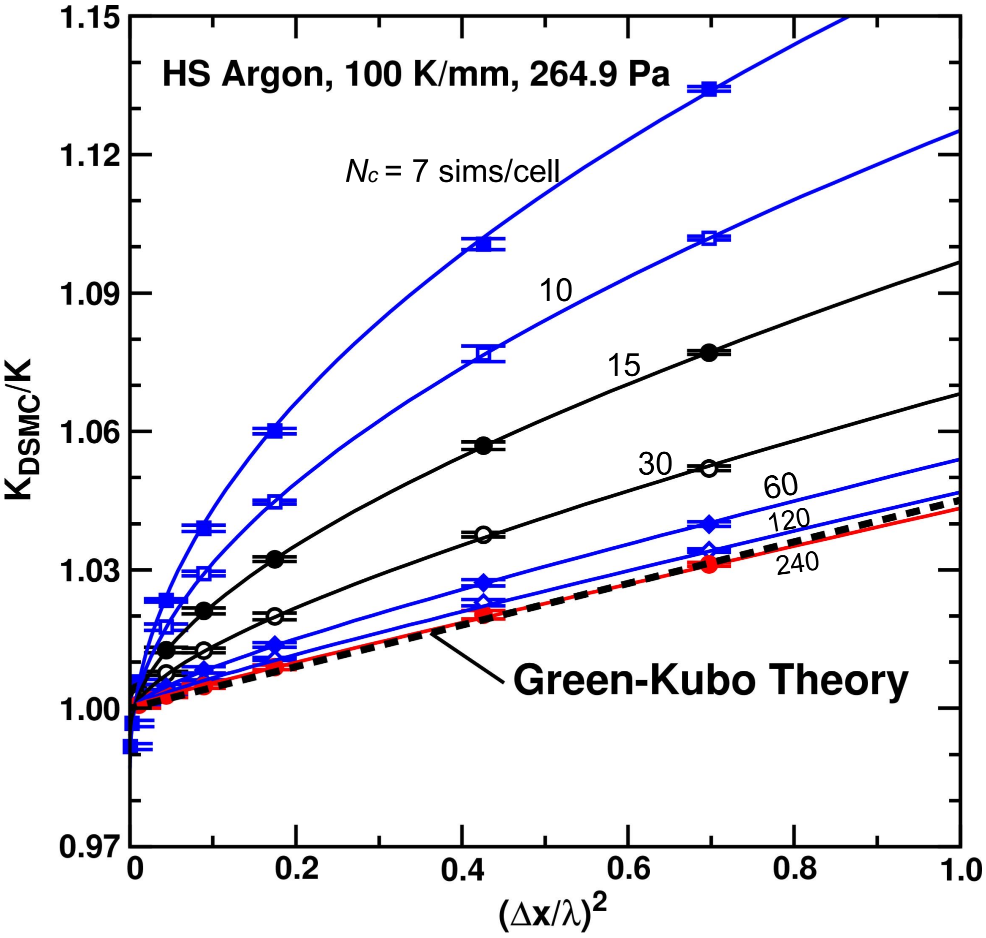
- Thermal conductivity for hard-sphere gas ($\Delta t \to 0$, $N_c \to \infty$)

$$\boxed{\frac{K_{DSMC}}{K} = 1 + \frac{32}{225\pi}\left(\frac{\Delta x}{\lambda}\right)^2 = 1 + 0.04527\left(\frac{\Delta x}{\lambda}\right)^2}$$

- Where the hard-sphere mean free path is $\lambda = \dfrac{1}{\sqrt{2}\pi d_{ref}^2 n}$

Cell-Size Convergence

Δt = 7 ns, $\Delta t/t_o$ = 0.1



- Complicated behavior of error with cell size
- **Quadratic dependence on cell size observed only for large N_c**
- Quadratic coefficient (0.041) agrees with G-K theory (0.0453) in the limit $N_c \to \infty$
- Curves through data from least squares fitting

Time-Step Error

Error related to time step, Δt

- **Collisions occur at the end of time step**
- Collisions should be uniformly distributed over time step

Green-Kubo theoretical analysis (Hadjiconstantinou, 2000)

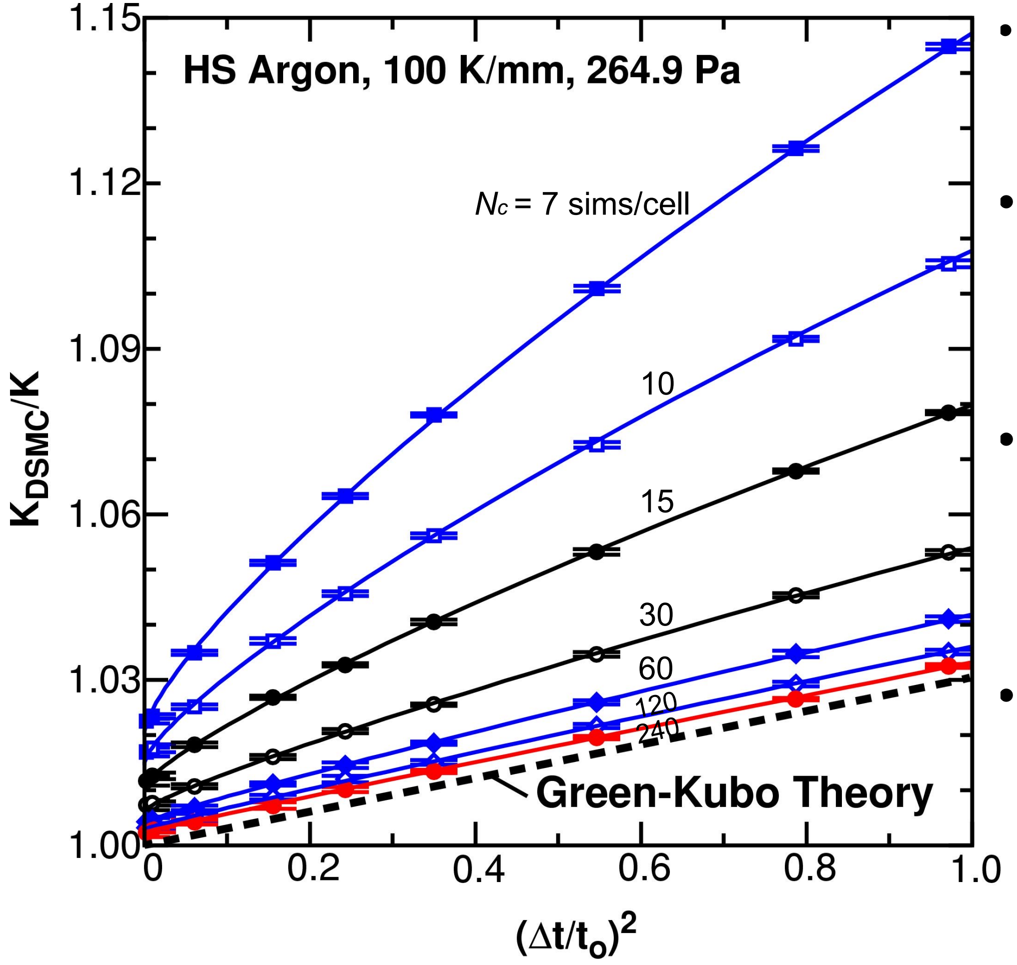
- Thermal conductivity for hard-sphere gas ($\Delta x \rightarrow 0$, $N_c \rightarrow \infty$)

$$\boxed{\frac{K_{DSMC}}{K} = 1 + \frac{64}{675\pi}\left(\frac{\Delta t}{t_o}\right)^2 = 1 + 0.03018\left(\frac{\Delta t}{t_o}\right)^2}$$

- With the hard-sphere mean collision time $t_o = \frac{\lambda}{c_o} = \frac{\lambda}{\sqrt{2k_B T / m}}$
- Supporting DSMC calculations (Garcia & Wagner, 2000)
- *More data needed for $\Delta t/t_o$ < 1*

Time-Step Convergence

200 cells, $\Delta x/\lambda = 0.209$



- Complicated behavior of error with time step
- **Quadratic dependence on time step observed only for large N_c**
- Quadratic coefficient (0.029) agrees with G-K theory (0.0302) in the limit $N_c \rightarrow \infty$
- Curves through data from least squares fitting

Functional Form of Error

Functional form that represents DSMC data

- Taylor series expansion in Dx, Dt, and 1/N_c
- Perform least-squares fitting of entire data set
- Retain statistically significant terms:

$$\frac{K_{DSMC}}{K} = 1.0001 + 0.0286\left(\frac{\Delta t}{t_o}\right)^2 + 0.0411\left(\frac{\Delta x}{\lambda}\right)^2$$

$$-0.01\left(\frac{\Delta t}{t_o}\right)^2\left(\frac{\Delta x}{\lambda}\right)^2 - 0.147\frac{1}{N_c} + \frac{1}{N_c}F\left[\frac{\Delta t}{t_o},\frac{\Delta x}{\lambda},\left(\frac{\Delta t}{t_o}\right)^2\right]$$

Key results:

- DSMC reproduces CE conductivity to within fitting uncertainty
- Quadratic terms $(\Delta x)^2$ and $(\Delta\ t)^2$ agree with Green-Kubo theory
- Other terms have not been reported previously

Summary

- DSMC limiting convergence behavior matches theory
  - Quadratic convergence in time step ($\Delta x/\lambda \to 0$, $N_c \to \infty$)
  - Quadratic convergence in cell size ($\Delta t/t_o \to 0$, $N_c \to \infty$)
  - Linear convergence in $1/N_c$ for $N_c \geq 30$ simulators/cell
  - *Coefficients* in good agreement with available theory
- For finite values of parameters, convergence behavior is a complicated function of higher-order cross terms

DSMC Collision Partner Selection

- DSMC randomly selects collision partners from a cell
- Average distance between two randomly selected points in an n-D cell:
  - 1-D = 0.333…
  - 2-D = 0.521…
  - 3-D = 0.662…
- Independent of number of points per cell
  - Only depends on cell size
- Increasing the number of simulators per cell (N_c) does not reduce the mean separation between randomly selected molecules.

VSC and TASC

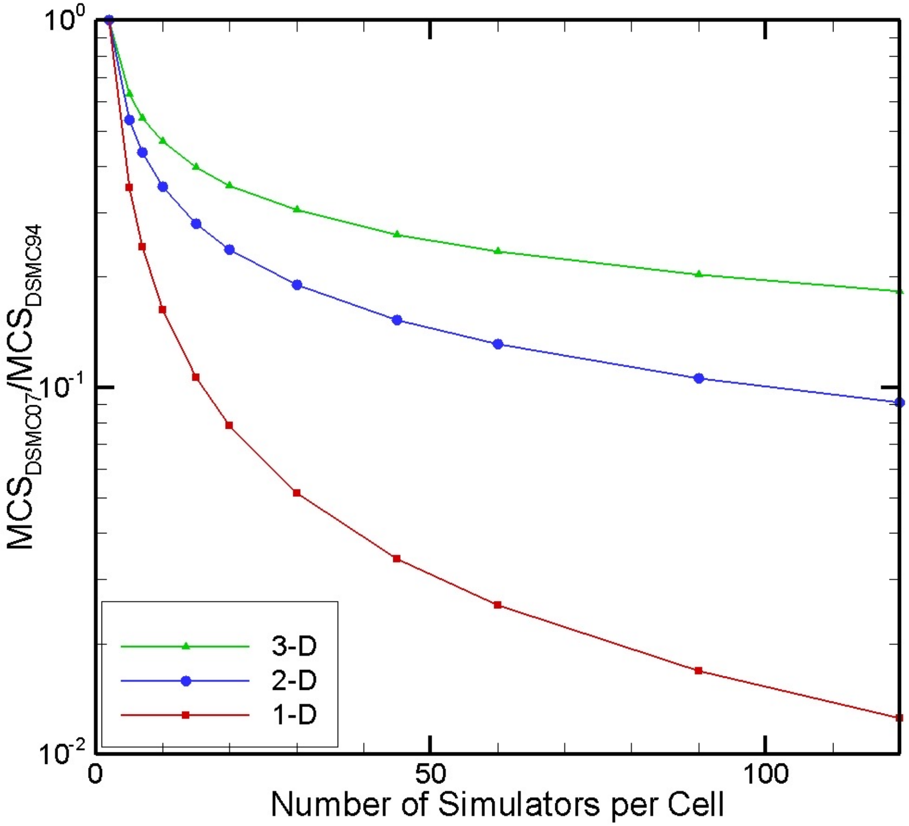
- Transient Adaptive Sub-Cells (Bird 2000)
  - Stochastically determines the nearest neighbor
  - Creates Cartesian sub-cell structure for each collision phase
  - Number of simulators in each sub-cell is ~ 1-2
  - Relatively inexpensive for large N: O(N)

- Virtual Sub-Cells (LeBeau 2003)
  - Deterministically determines the nearest neighbor
  - More accurate
  - More expensive than TASC for large N: $O(N^2)$

VSC, TASC Spatial Discretization

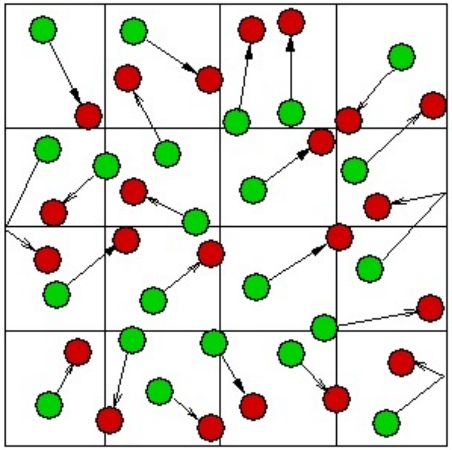
- With VSC and TASC, the average distance between nearest-neighbor points in a cell is a function of:
  - Dimensionality
  - Cell Size
  - Number of Simulators

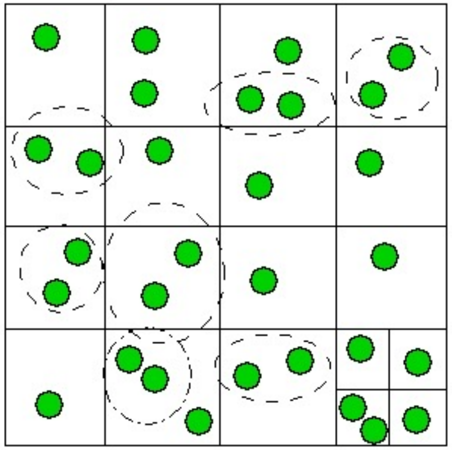
- Effect plateaus after about
  - 50 sims/cell for 3-D
  - 100 sims/cell for 2-D
  - 250 sims/cell for 1-D



Sophisticated DSMC (DSMC07)

- Basic features of DSMC algorithm retained
  - Move-collide separation, molecular models, collision frequency calculation
- Changes in collide
  - Virtual collision cells: nearest-neighbor (*NN*) collisions, N^2 search operation
  - Adaptive transient collision cells based on a background grid *(N>30)*
  - Exclusion of latest collision partner: physically realistic requirement for *NN* schemes





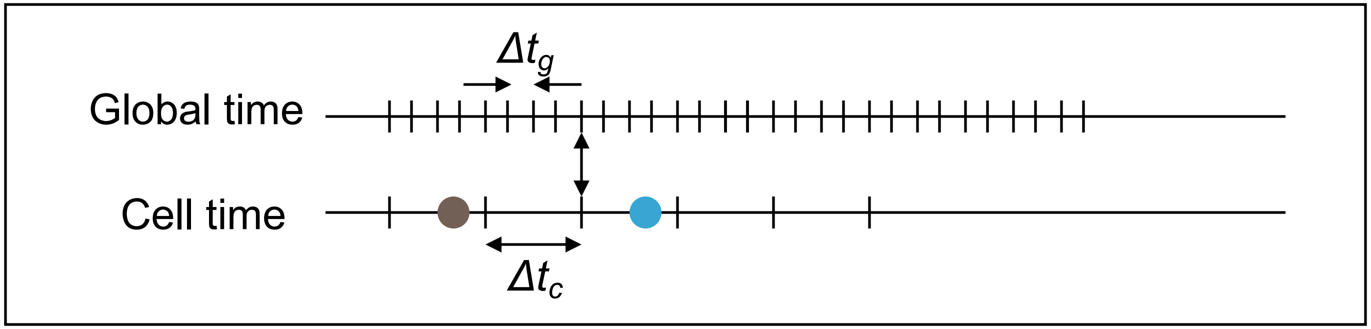
- Changes in temporal advection
  - Global time step
  - Cell-based time step

DSMC07 Temporal Advection

- Global time advances with small global time steps (Δt_g)
  Uniform throughout the domain, similar to DSMC94 time step

- Cell time step (Δt_c) is local (cell-based) and is a fraction of the local mean collision time (*MCT*) and the mean cell transient time (*MTT*)

- *With this (cell-based) time step:*
  - Cells perform collisions only when their "time" (cell time) falls more than $\Delta t_c/2$ behind global time
  - Molecules move only when their "time" (molecule time) falls more than $\Delta t_c/2$ behind global time



DSMC07 Time-Step Properties

- Local: Cell-based time step (Δt_c): function of the local collision frequency and average molecular speed
  - *Aims at improving efficiency, effective load balancing.*
- Adaptive: Global time step (Δt_g): function of the smallest Δt_c (adjusted during run)
  - *Aims at improving numerical accuracy*
- Dynamic: Move and collision phases are not synchronized for the whole domain (large low-collisionality regions)
- Collisions are considered for a cell
  Move is considered for a particular molecule
- Molecules cannot travel across a cell in one move without considering collisions
- Same pair cannot have sequential collisions
  - *Aims at improving physical accuracy*

Reduced Set Selection VSC

- Not all simulators need to be examined to determine an approximate nearest neighbor.

- Significant performance improvement with smaller computational cost can be achieved when a random sub-set of simulators is used.

- A limit of 30 should be adequate for most applications, as suggested by Bird.

DSMC07
Functional Form of Error

Best-fit correlation function for sophisticated DSMC with $\Delta t = \Delta t_o$

$$\frac{K_{DSMC}}{K} = 0.9953 + 0.07773\left(\frac{2\cdot\Delta t}{t_o}\right) + 0.02795\left(\frac{2\cdot\Delta t}{t_o}\right)^2 \text{-}0.0066\left(\frac{\Delta x}{\lambda}\right)^2$$
$$-0.0234\left(\frac{2\cdot\Delta t}{t_o}\right)^2\left(\frac{\Delta x}{\lambda}\right)^2 + 0.60375\frac{1}{N_c} + \frac{1}{N_c}F\left[\frac{2\cdot\Delta t}{t_o},\frac{\Delta x}{\lambda},\left(\frac{2\cdot\Delta t}{t_o}\right)^2\right]$$

DSMC limiting convergence differs from GK behavior

- Linear convergence in time step ($\Delta x/\lambda \to 0$, $N_c \to \infty$)
- Weak quadratic convergence in cell size ($\Delta t/t_o \to 0$, $N_c \to \infty$)
- Linear convergence in $1/N_c$ for $N_c \geq 30$ simulators/cell
- Cross terms needed for finite discretization (but in general smaller than for DSMC94)

Move-Collide: A Closer Coupling

- Molecules should not travel across a sampling cell in one move without considering collisions.
- DSMC07 is more sensitive to time step.
- It is physically inconsistent to allow molecules to ignore collision partners during long advection phases.
- Allowing nearest-neighbor collisions when MCS~MFP introduces an error that is linear in time step.
- Similar to DSMC94 sub-cells, time step should conform to sub-cell structure.

Trajectory-Based Selection

- To minimize the time step error in the VSC scheme, a near-neighbor collision partner is selected from within a sphere centered on the simulator with a radius proportional to the distance traveled by the simulator during the latest advection phase.
  - *Simulators with high velocities have more simulators available as collision partners, whereas simulators with low velocities have to collide with their nearest neighbor.*
- Two options investigated:
  - Distance traveled = selection sphere diameter (DSMC07-D)
  - Distance traveled = selection sphere radius (DSMC07-R)

Comparison of Convergence Rates

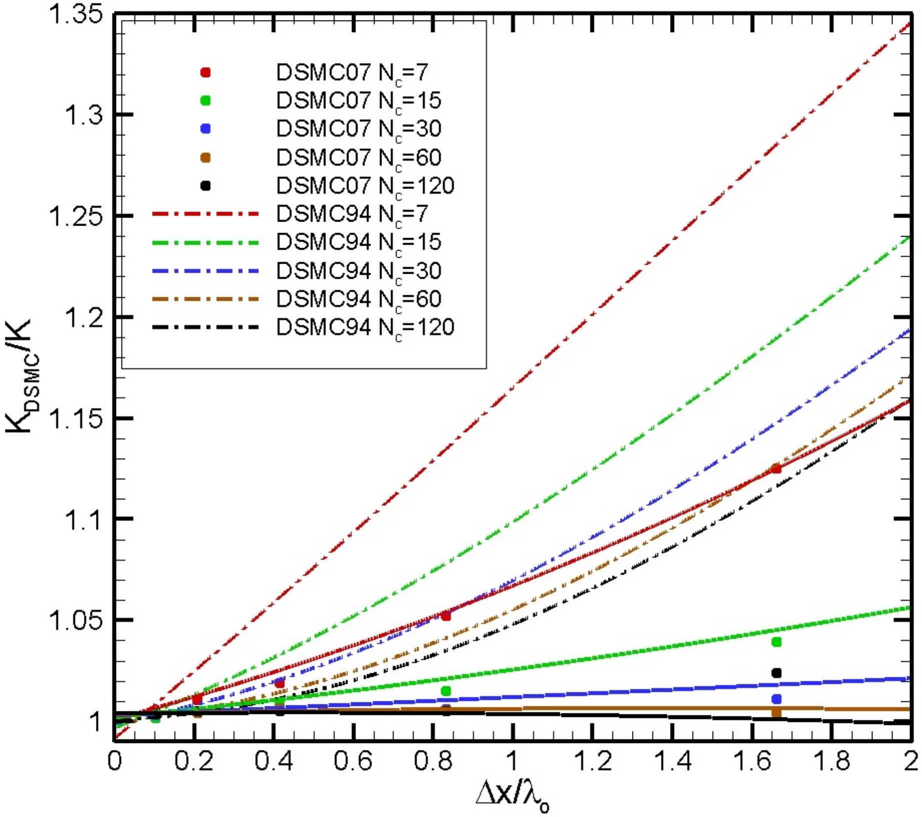
$$\frac{\bar{K}_{\mathrm{DSMC94}}}{K} = 1.0001 + 0.0287\Delta\tilde{t}^{2} + 0.0405\Delta\tilde{x}^{2} - \frac{0.083}{N_c} + F\left(\Delta\tilde{t}, \Delta\tilde{x}, 1/N_c\right)$$

$$\begin{aligned}\frac{K_{DSMC}}{K} &= 0.9953 + 0.07773\left(\frac{2\cdot\Delta t}{t_o}\right) + 0.02795\left(\frac{2\cdot\Delta t}{t_o}\right)^{2} \text{-}0.0066\left(\frac{\Delta x}{\lambda}\right)^{2} \\ &-0.0234\left(\frac{2\cdot\Delta t}{t_o}\right)^{2}\left(\frac{\Delta x}{\lambda}\right)^{2} + 0.60375\frac{1}{N_c} + \frac{1}{N_c}F\left[\frac{2\cdot\Delta t}{t_o}, \frac{\Delta x}{\lambda}, \left(\frac{2\cdot\Delta t}{t_o}\right)^{2}\right]\end{aligned}$$

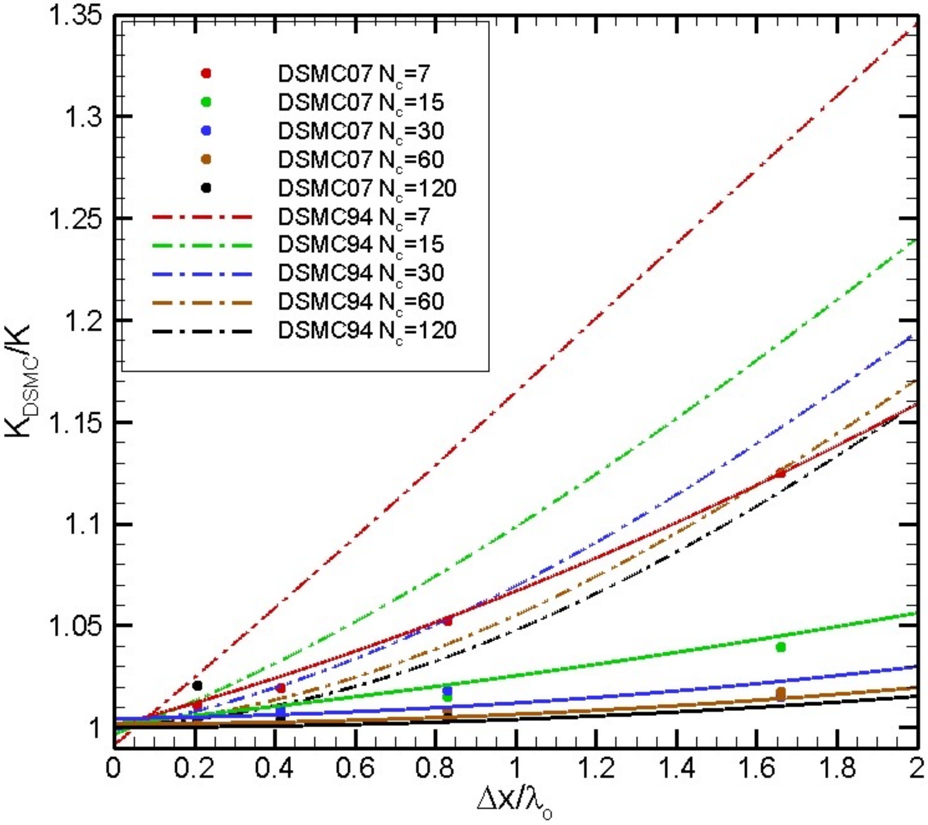
$$\begin{aligned}\frac{\bar{K}_{\mathrm{DSMC07,VSC}}}{K} &= 0.9951 + 0.0629\Delta\tilde{t} + 0.0478\Delta\tilde{t}^{2} + 0.0027\Delta\tilde{x}^{2} \\ &- 0.0034\Delta\tilde{t}^{4} + 0.0019\Delta\tilde{t}^{2}\Delta\tilde{x}^{2} \\ &- 0.1062\big/N_c^{2} + 0.0922/N_c + F_{\mathrm{VSC}}\left(\Delta\tilde{x}, \Delta\tilde{t}, 1/N_c\right)\end{aligned}$$

DSMC07 Convergence ($\Delta t/t_o$=10)

VSC



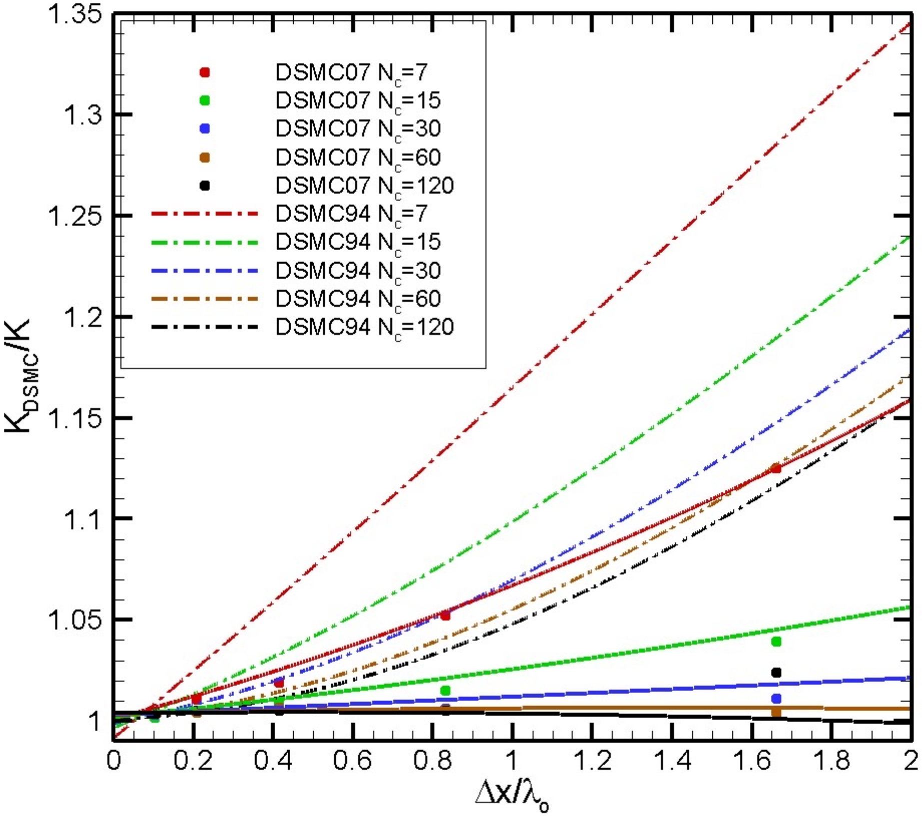
TASC



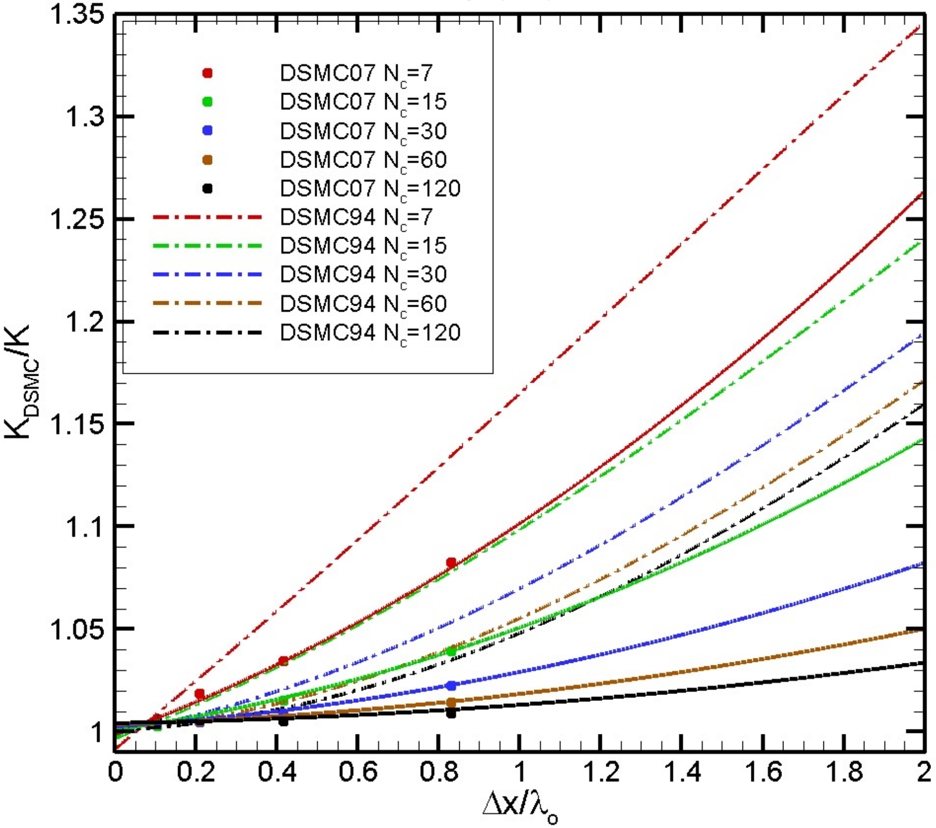
- VSC and TASC have similar performance for small Nc
- VSC is more accurate for determining the nearest neighbor for large Nc

DSMC07-3D Convergence ($\Delta t/t_o$=10)

VSC-1D

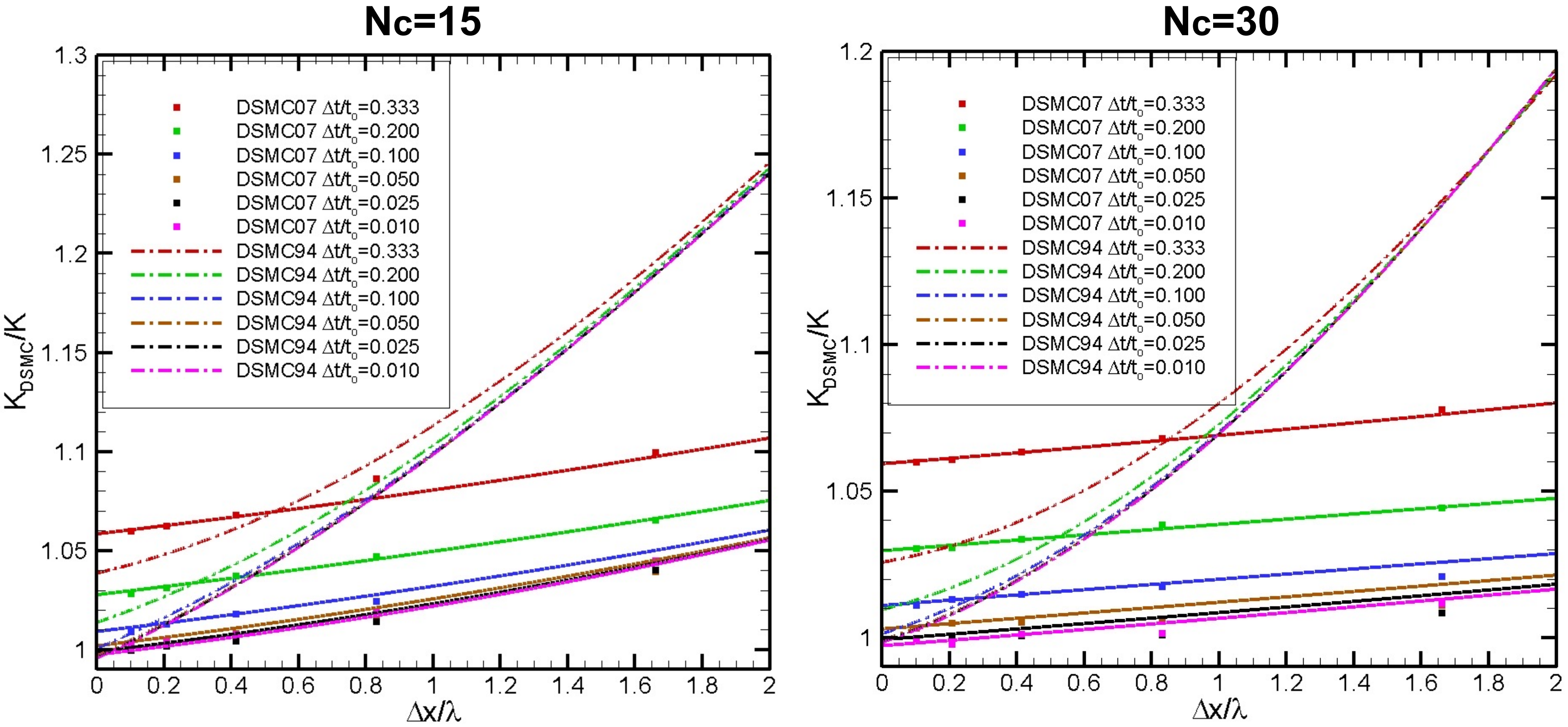


VSC-3D



- The 1D domain is replaced with a 1D series of 3D cells.
- Nearest neighbors determined by $ds = \sqrt{dx^2 + dy^2 + dz^2}$
- Increased dimensionality reduces the effectiveness of the scheme.

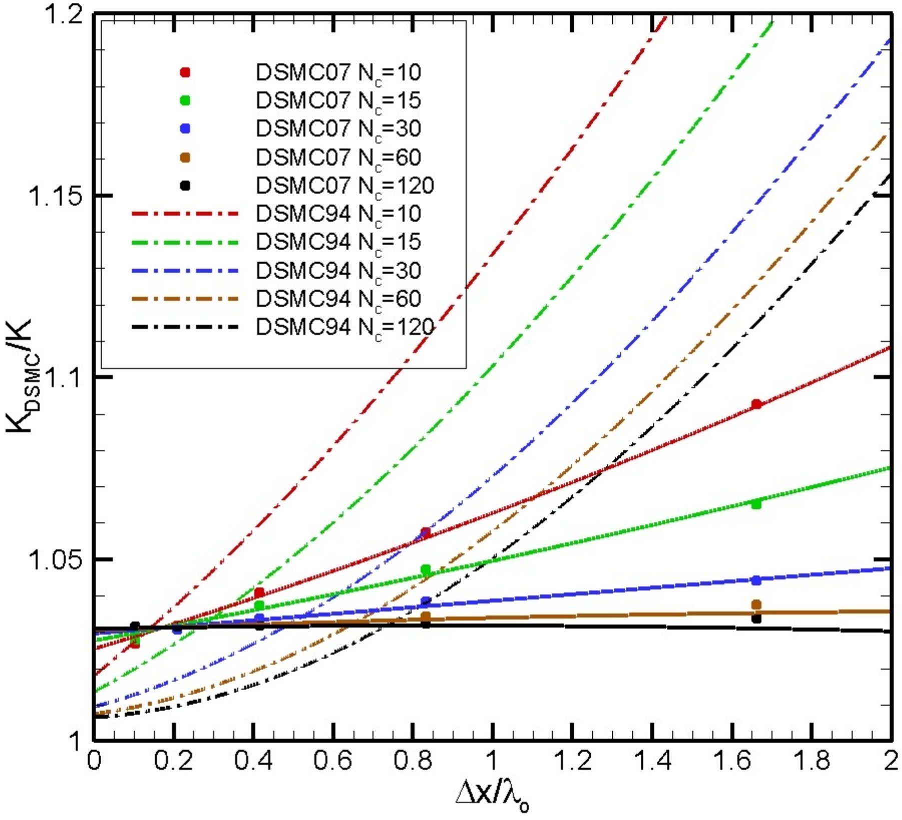
Convergence Behavior for N_c = 15,30
Effect of Cell Size



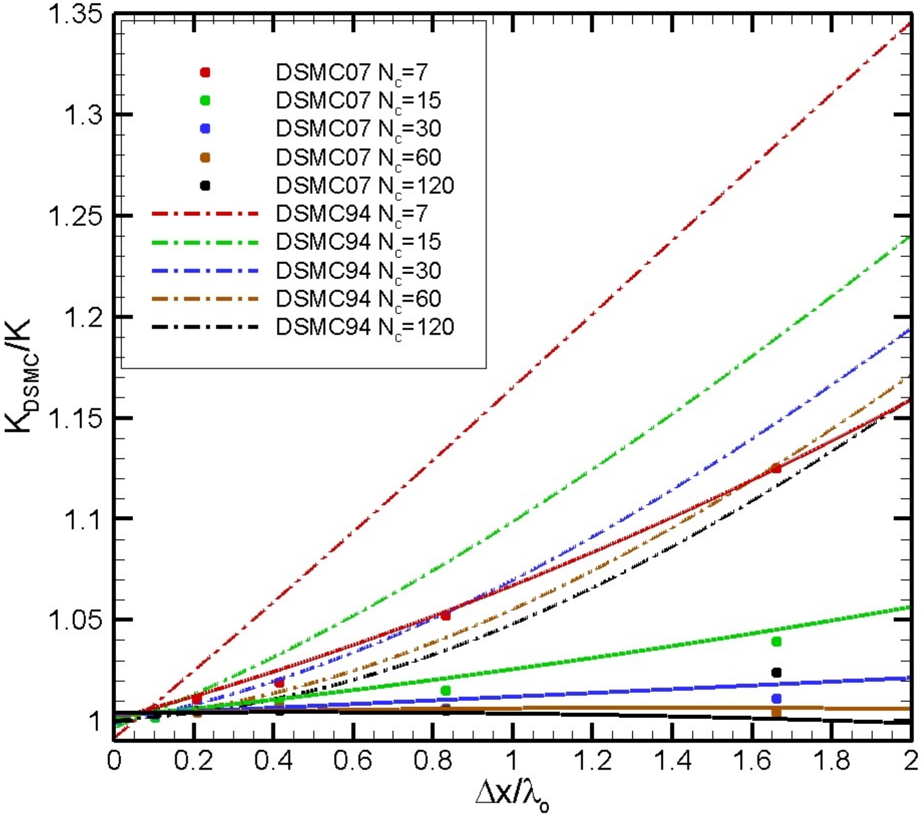
- For finite number of simulators, the algorithm is
  - Insensitive to spatial resolution.
  - DSMC07 error is lower than DSMC94 when
  - *$\Delta t/t_o$* ≤ 0.1 (Nc=15), *$\Delta t/t_o$* ≤ 0.05 (Nc=30).

Effect of Time-Step Selection

$\Delta t/t_o$=5

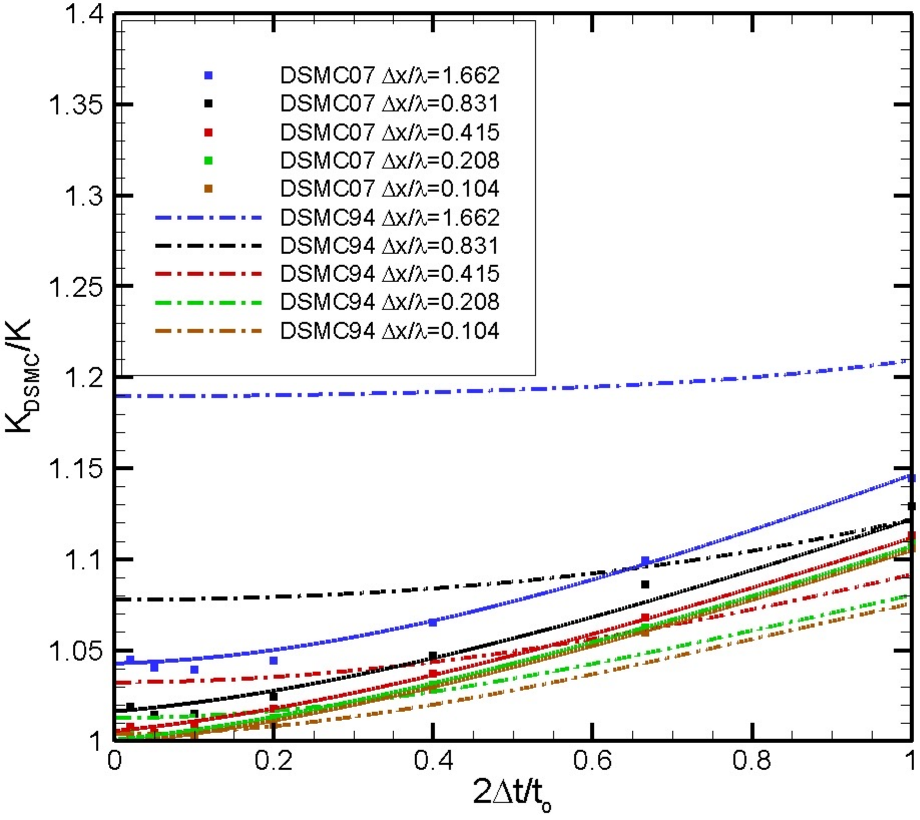


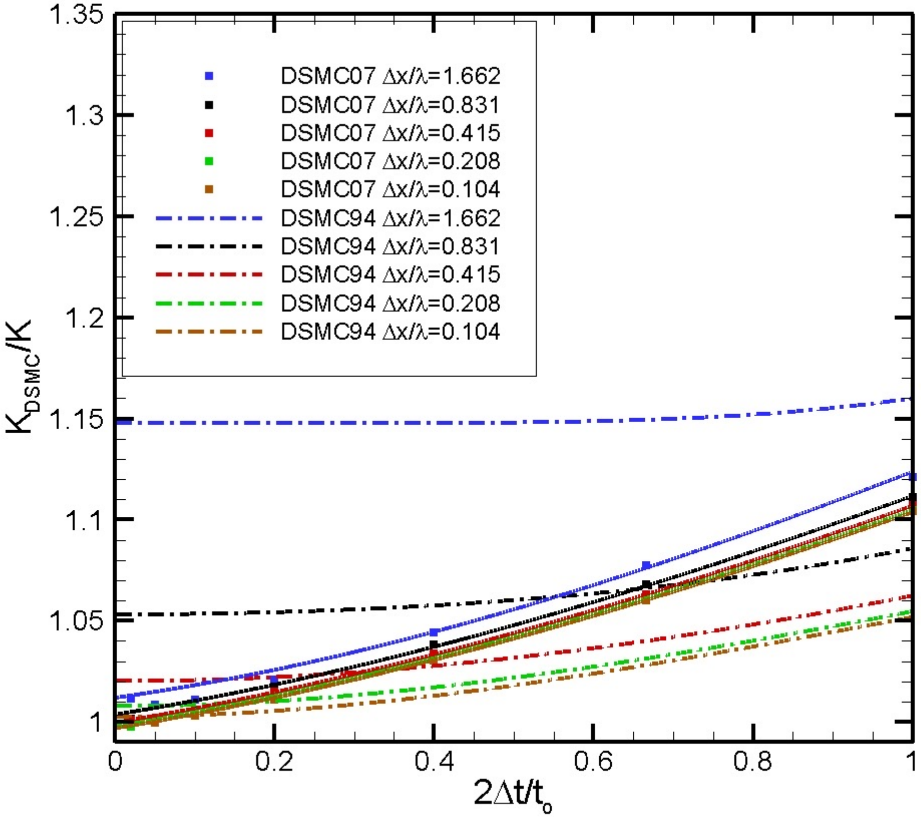
$\Delta t/t_o$=10



- Time-step selection must be based on both : local collision time and local transit time

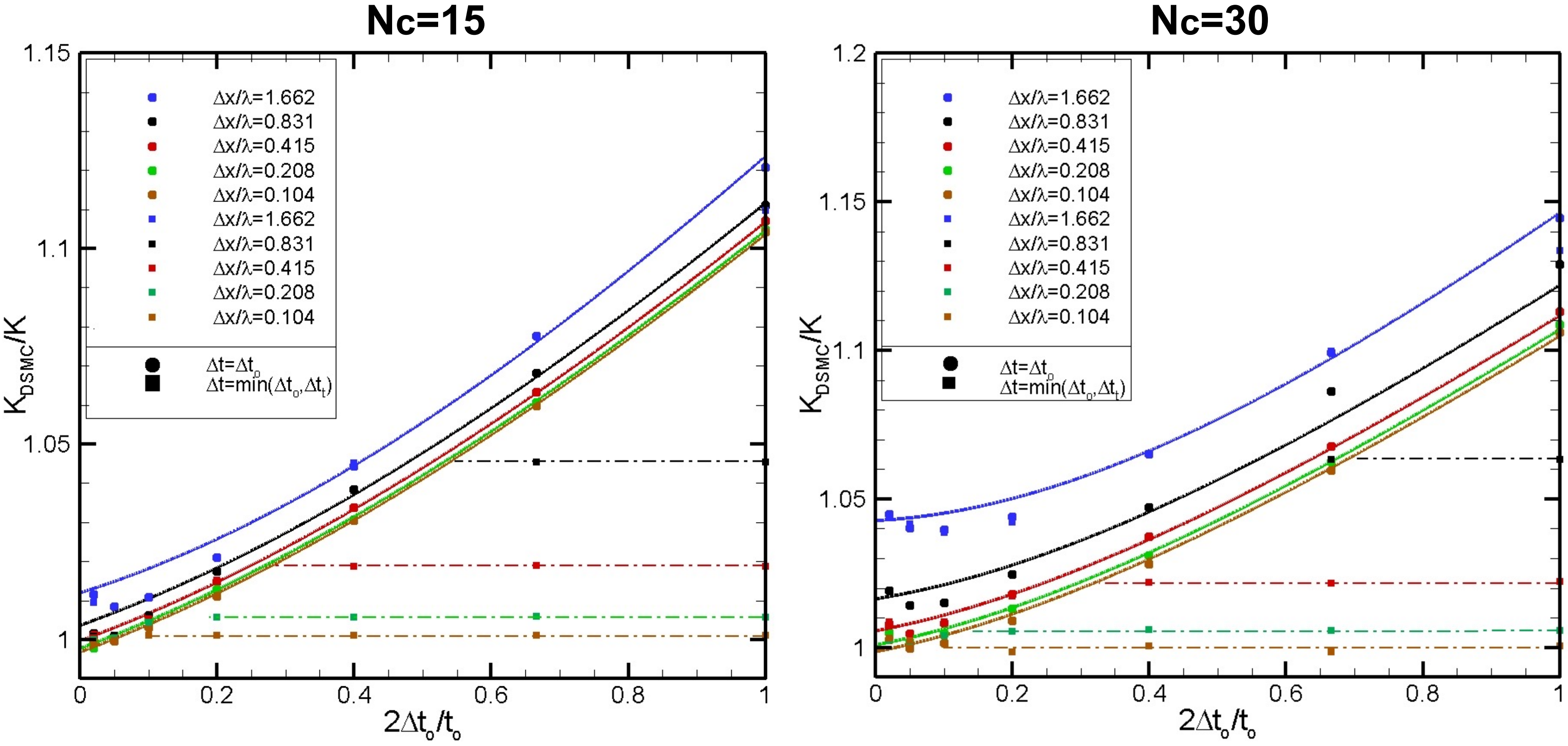
Convergence Behavior for N_c = 15,30
Effect of Time Step


Nc=15


Nc=30

- The algorithm exhibits a near-linear dependence on time step.
- For nearest-neighbor schemes, time step should conform to the effective spatial resolution.

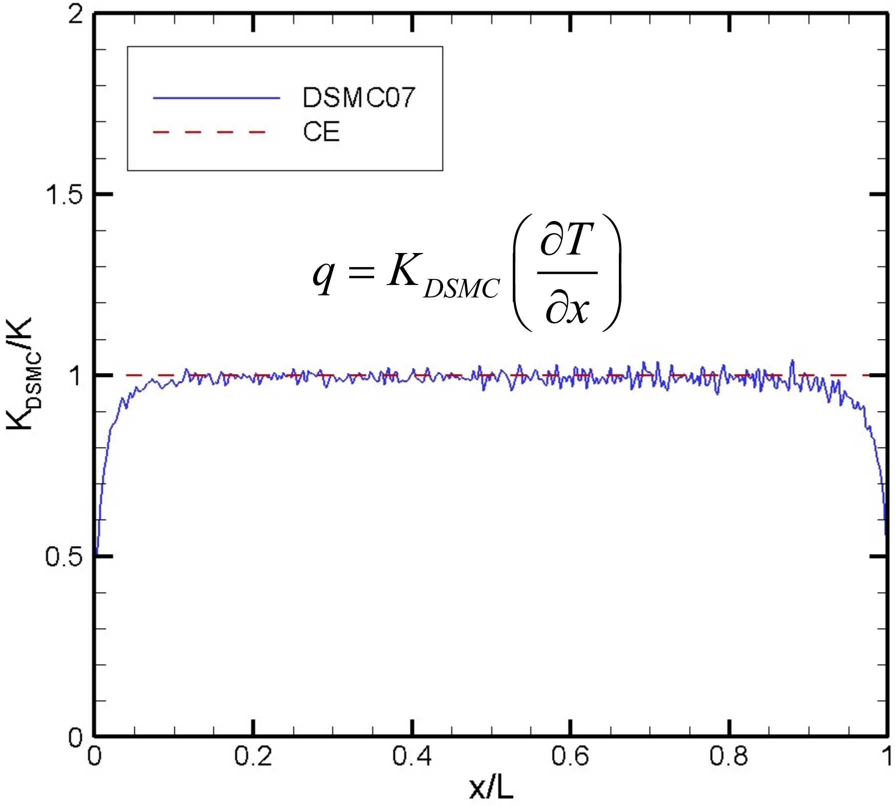
Convergence Behavior for N_c = 15,30
Effect of Time Step

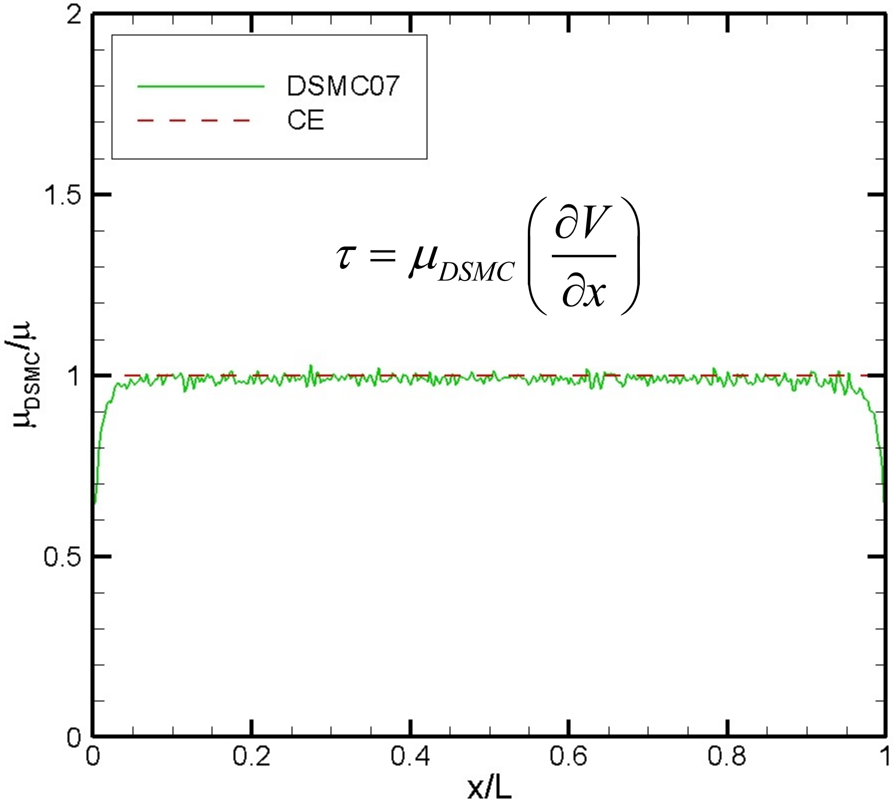


The DSMC07 time step Δt is based on both

- mean collision time t_o, via $\Delta t_o \sim t_o$
- mean cell-transit time t_t, via $\Delta t_t \sim t_t$

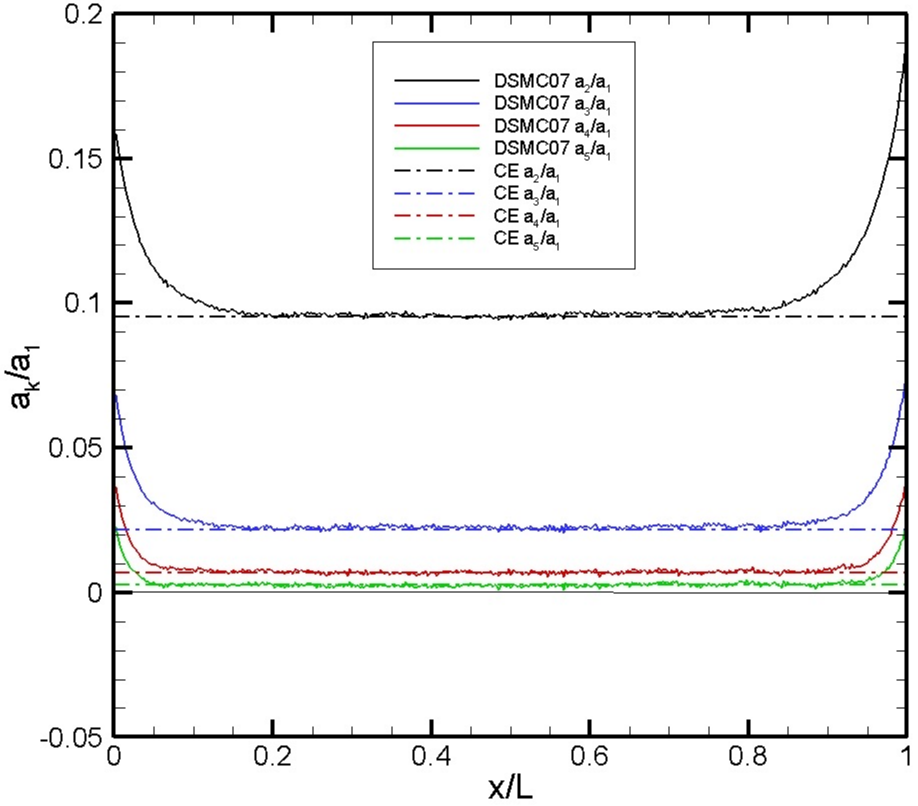
Hard-Sphere
Transport Coefficient Profiles

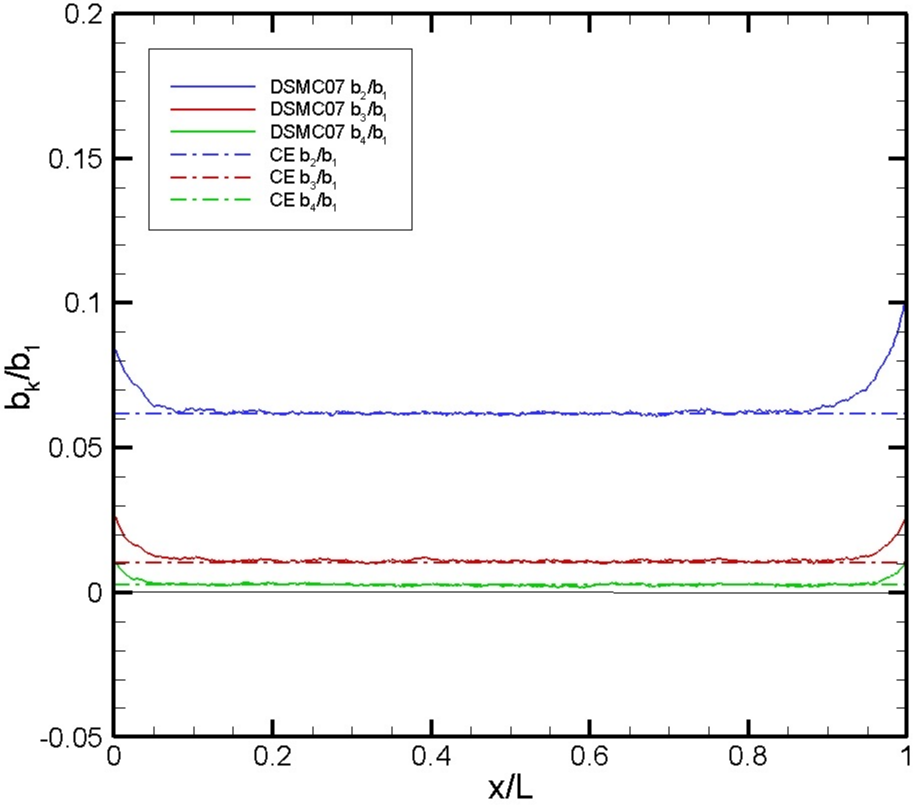




- DSMC07 and CE thermal conductivity and viscosity
  - Low heat flux, low shear stress: $Kn_q = 0.006$, $Kn_\tau = 0.003$
  - Agreement in central region: normal solution

Hard-Sphere
Sonine-Coefficient Profiles





- DSMC07 and CE Hard-Sphere Sonine polynomial coefficients a_k/a_1 and b_k/b_1
  - Low heat flux, low shear stress: $Kn_q = 0.006$, $Kn_\tau = 0.003$
  - Good agreement in central region: normal solution
  - Demonstrates accuracy of molecular velocity distribution

Grid Adaptation

- User specified
  - Multiple levels (up to 16 in 64 bit architecture)

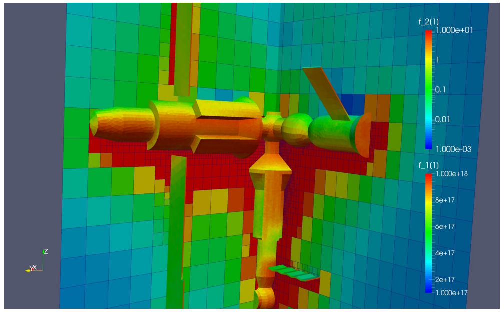
```
create_grid 100 100 100 level 2 25*75 25*75 25*75 2 2 2
create_grid 5 5 1 level 2 * * * 5 5 1 level 3 * * * 2 2 1
```

- Automated grid adaptation
  - Static

```
adapt_grid refine coarsen surf 0.15 iterate 1 dir 1 0 0
```

  - Dynamic

```
fix 10 adapt 10 refine coarsen value c_1[1] 4500 5500 &
fix 10 adapt 100 refine coarsen value c_1b[2] 1.00 2.00 &
combine min thresh less more maxlevel 5 cells 2 2 1
```



Time Step Adaptation

Time step can be calculated based on mean collision time and mean transit time
The time step can be calculated based on averaged or instantaneous cell data.

```
compute           1t grid all all nrho temp usq vsq wsq
fix               1t ave/grid all 1 1000 1000 c_1t[*]

compute           lambda lambda/grid c_1t[1] c_1t[2] N2
fix               lambda ave/grid all 1 1000 1000 c_lambda

compute           tstep dt/grid all 0.25 0.25 &
                  c_lambda f_1t[2] f_1t[3] f_1t[4] f_1t[5]

fix               DT dt/reset 1000 c_tstep 0.1 1
```

Steady State in SPARTA

- Traditional DSMC codes define “steady state” at some time during the run.

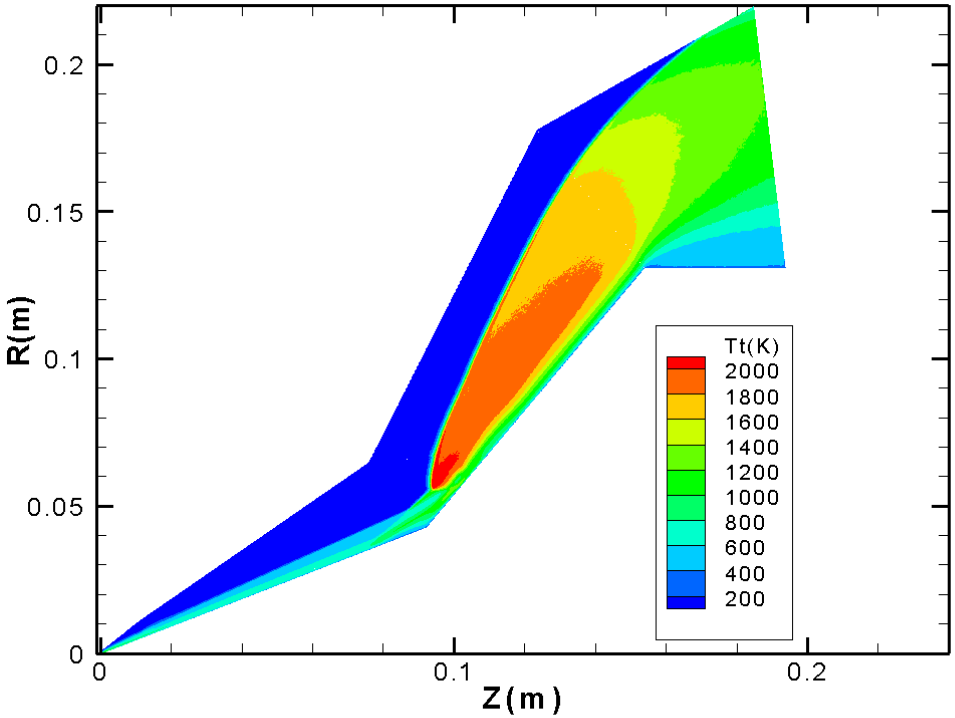
- Sparta defines a series of “runs”.
- Steady state is defined as a run where the maximum collision probability is not reset.
- Moment accumulators are reset between runs.
- Moments are accumulated as instantaneous or running averages during runs.

Near-Neighbor Selection

Near-neighbors with selection limit.

```
collide_modify      nearcp yes 30
```

Simulation Optimization



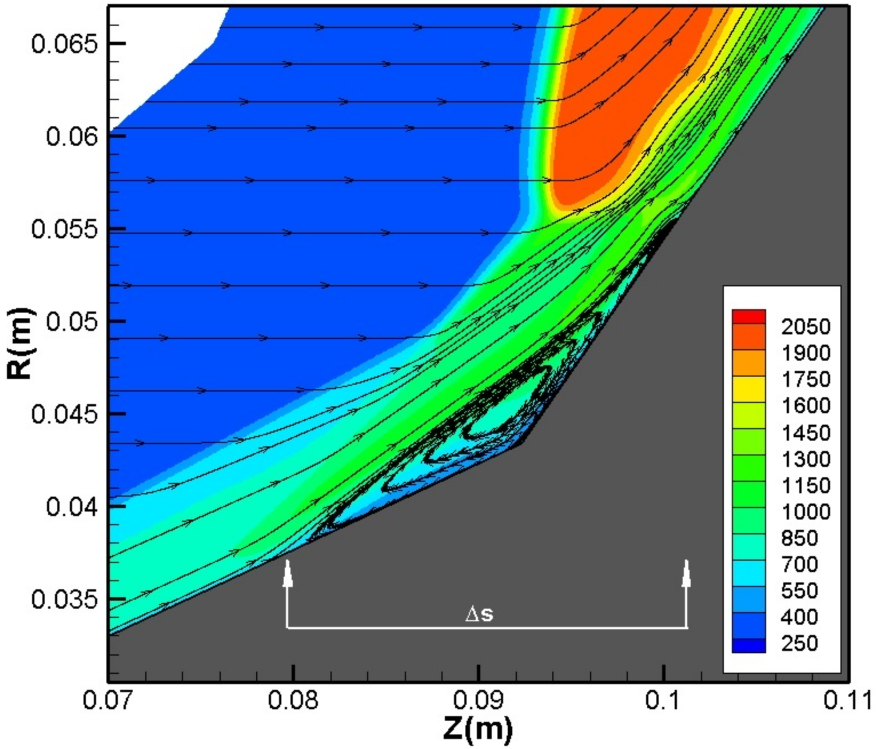
Goal: optimize simulation at the lowest computational cost

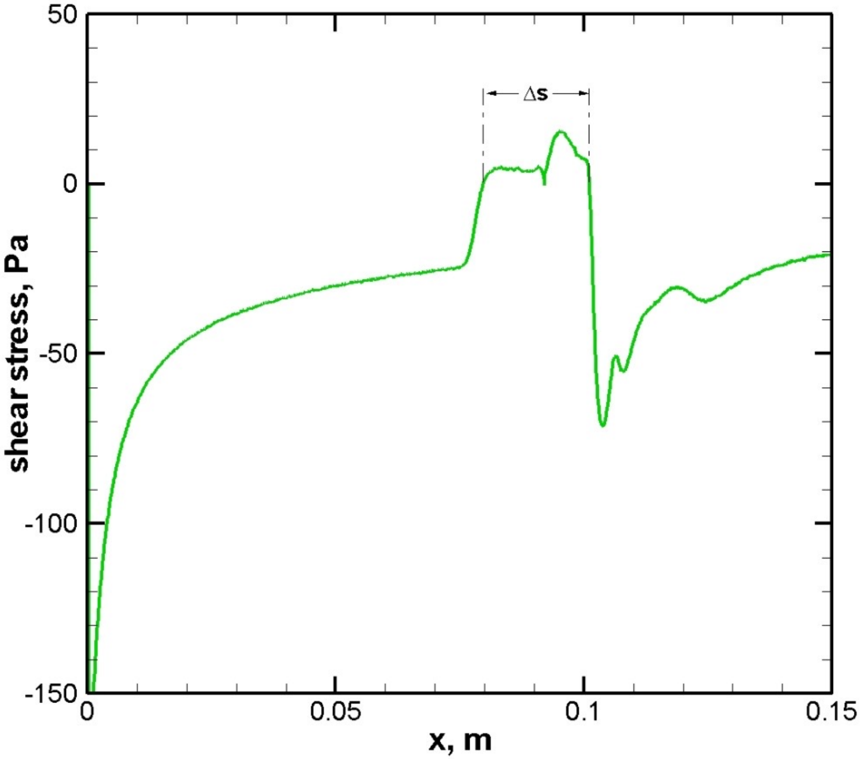
The simulation was refined using the number of cells as a parameter.

Simulation Parameters

Sims per cell:	**10,30**
Time step:	**2.5 ns (minimum)**
Cells:	**10^6, $0.25\text{x}10^6$**

Recirculation Zone

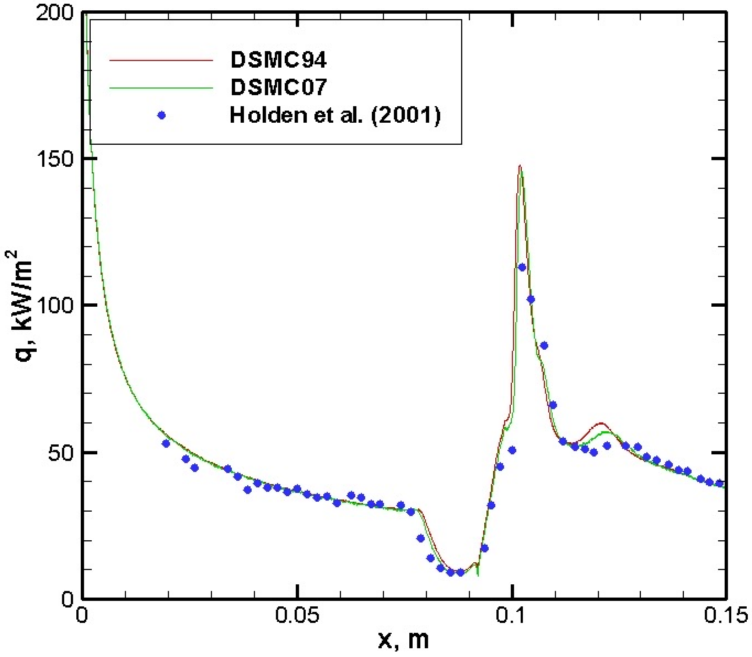


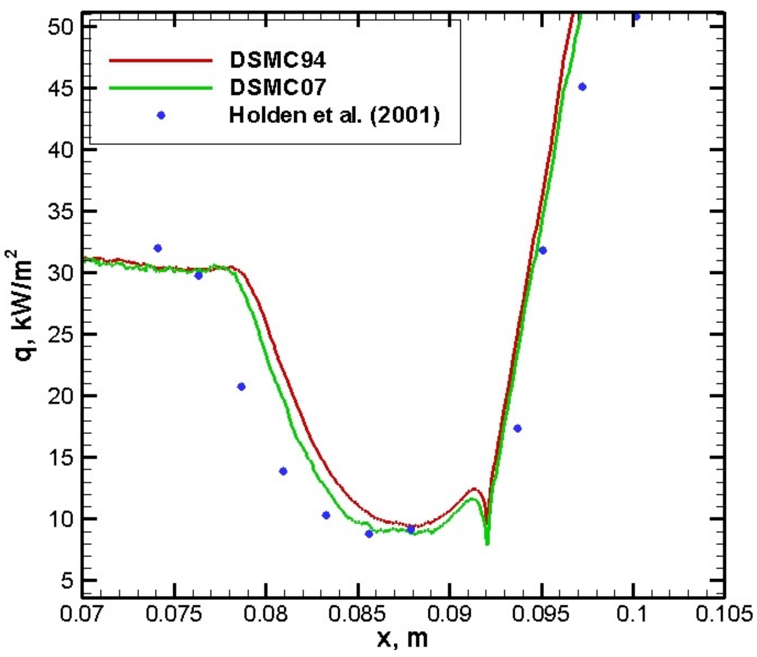


Extensive studies (Moss, Markelov) indicate that the length of the recirculation zone is Δs_0 = 21.5 mm.

The recirculation zone length Δs is sensitive to discretization parameters and is used as a convergence metric.

DSMC94 and DSMC07 Simulation Speeds When Achieving Equal Accuracy





DSMC94 MCS = DSMC07 MCS							
DSMC94				DSMC07			
# Cells	Sims/cell	Δs Error (%)	ns/proc-min	# Cells	Sims/cell	Δs Error (%)	ns/proc-min
250,000	10	24.19	8.72	250,000	10	5.58	5.61
1,000,000	10	5.58	1.48	1,000,000	10	0.00	1.01

Relative Performance of Selection Schemes

Algorithm	Search Limit	Normalized Simulation Rate	Δs (mm)	Error $(\Delta s_0 - \Delta s)/\Delta s_0$
DSMC94	1	1.48	20.3	6.0 %
DSMC07	∞	1.00	21.4	0.5 %
DSMC07-10	10	1.32	21.3	0.9 %
DSMC07-30	30	1.15	21.5	0.0 %
DSMC07-R	∞	1.14	21.6	-0.5 %
DSMC07-D	∞	1.11	21.5	0.0 %

Sparta Tools to improve computational efficiency

- Grid adaptation (increased memory)
- Timestep adaptation
- Near-neighbors
- Load balancing (increased memory)

Careful use of these functions can give you a speedup factor of 3-10.