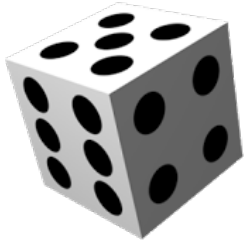
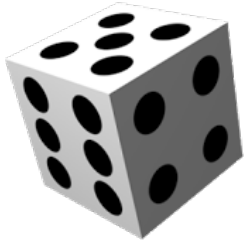


Challenges in DSMC



Cyril Galitzine and Iain D. Boyd

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI

Workshop on Moment Methods in Kinetic Theory II
October 14-17, 2014

Fields Institute, Toronto, ON

Outline

- A Brief Review of DSMC
- Developing hybrid CFD/DSMC methods
- Increasing the computational efficiency of DSMC
- Quantifying the convergence/error of DSMC
- Conclusions

Introduction

- DSMC = Direct Simulation Monte Carlo
  - 1963 Invented by G.A. Bird using physical/heuristic arguments[1]
  - '70s-'80s Improved collision schemes
  - '80s Chemically reacting flow
  - 1992 Convergence proof[2]
  - '90s Low speed/micro flows
  - '00s Hybrid methods and wider usage (research/open source codes)
- "Solves" the Boltzmann equation and easily extendable to multiple species/ phases, chemically reacting, ionized flows flows
- By far the most dominant methods for rarefied gas flows in the transition regime
- Often used to obtain a "reference" solution for moment methods
- Widely used in Aerospace/Materials processing

[1]Bird, G.A., *Phys. Fluids*, 6, 1518-1519 (1963).
[2]Wagner, W. *J. Stat Phys.*, 66, 1011-1044 (1992)

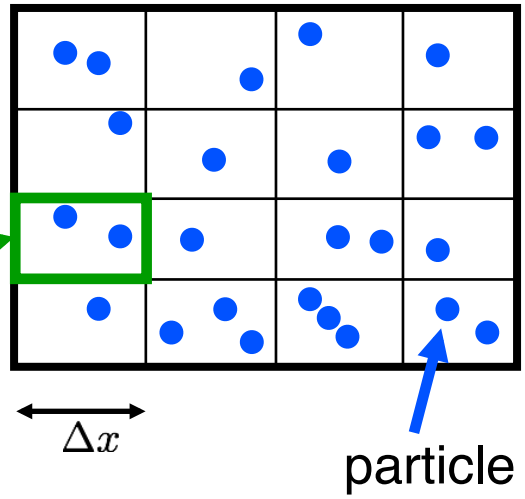
Formulation

- Discretize space using computational cells (mesh)
- Use N computational particles that represent $W_p \sim 10^5$ -10^{10} physical particles

number density

$$n = \frac{W_p \ N}{V}$$

# of particles



- Split the Boltzmann equation

(1) Streaming $$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} = 0$$

(2) Collision $$\frac{\partial f}{\partial t} = \int_{\mathbb{R}^3} d\vec{v}_1 \int_{4\pi} d\Omega \ \sigma \ v_r \left[f(\vec{x}, \vec{v}_1', t) \ f(\vec{x}, \vec{v}', t) - f(\vec{x}, \vec{v}_1, t) \ f(\vec{x}, \vec{v}, t) \right]$$

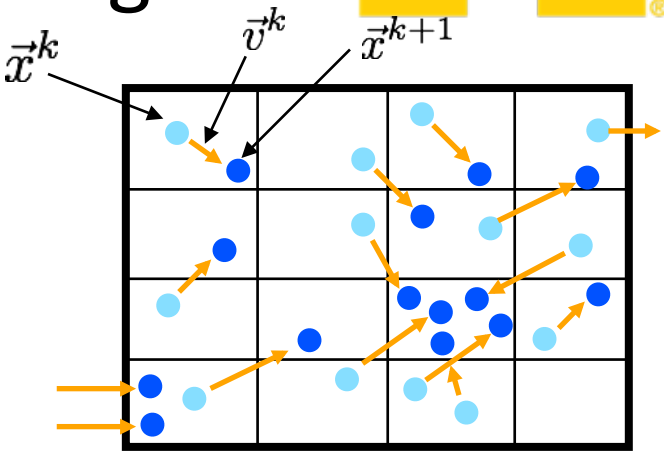
Kac master equation in reality when $N < \infty$ …

Sequentially "solve" Eqs. **(1)** and **(2)** on the mesh at each time step using computational particles.

[1]Bird, G.A., Molecular Gas Dynamics and the Direct Simulation of Gas Flows, OUP (1994)
[2]Rjasanow, S. and Wagner, W., Stochastic Numerics for the Boltzmann Equation, Springer (2005)

Formulation: Streaming

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} = 0$$ (Vlasov Equation)



$$f(\vec{x}, \vec{v}, t) \approx W_p \sum_{j=1}^{N} \delta(\vec{x} - \vec{x}_j(t)) \ \delta(\vec{v} - \vec{v}_j(t))$$

1) Generate particles at inflows/Remove particles at outflows
2) Move particles

$$\begin{cases} \frac{d\vec{x}}{dt} & = \vec{v} \\ \frac{d\vec{v}}{dt} & = \vec{0} \end{cases} \Rightarrow \begin{cases} \vec{x}^{k+1} & = \vec{x}^k + \Delta t \times \vec{v}^k \\ \vec{v}^{k+1} & = \vec{v}^k \end{cases}$$

$$n_i(t) = \frac{1}{\text{Vol}_i} \int_{\text{Cell i}} \int_{\mathbb{R}^3} f(\vec{x}, \vec{v}, t) \, d\vec{v} \, dV_i = \frac{W_p}{\text{Vol}_i} \sum_{j=1}^{N} \mathbb{1}_i[\vec{x}_j(t)]$$

$$\vec{v}_i(t) = \frac{1}{n_i} \int_{\mathbb{R}^3} \vec{v} \, f(\vec{x}, \vec{v}, t) \, d\vec{v} = \frac{1}{N_i} \sum_{j=1}^{N} \mathbb{1}_i[\vec{x}_j(t)] \, \vec{v}_j(t)$$

DSMC

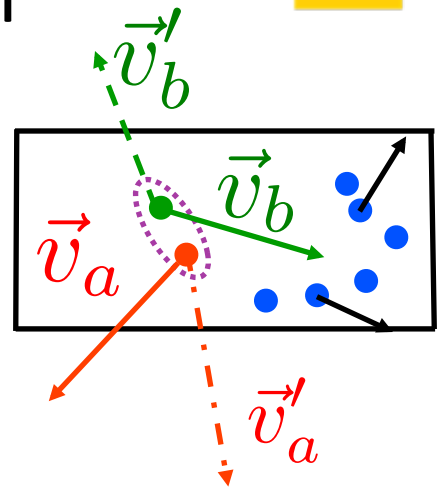
Formulation: Collision

State vector:

$$\vec{V} = (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_{N-1}, \vec{v}_N)$$

Joint PDF of all the N particles in the cell:

$$dP = F^N\left(\vec{V}\right) d\vec{v}_1 \; d\vec{v}_2 \; \dots \; d\vec{v}_{N-1} \; d\vec{v}_N$$



$$\vec{V} = (\vec{v}_1, \vec{v}_2, \dots, \vec{v}_a, \dots, \vec{v}_b, \dots, \vec{v}_{N-1}, \vec{v}_N)$$ Before collision

$$\vec{V}'_{ab} = (\vec{v}_1, \vec{v}_2, \dots, \vec{v}'_a, \dots, \vec{v}'_b, \dots, \vec{v}_{N-1}, \vec{v}_N)$$ After particles “a” and “b” have collided

Master equation

Before collision

$$\begin{cases} \frac{\partial F^N(\vec{V},t)}{\partial t} = \frac{W_p}{\mathrm{Vol}} \sum_{1 \le a < b \le N} \int_{\mathbb{S}^2} \sigma_{ab} \|\vec{v}_a - \vec{v}_b\| \left[F^N\left(\vec{V}'_{ab}, t\right) - F^N\left(\vec{V}, t\right) \right] d\vec{e} \\ F^N\left(\vec{V}, 0\right) = \frac{\mathrm{Vol}}{W_p \; N} \prod_{j=1}^{N} \hat{f}\left(0, \vec{v}_j\right) \end{cases}$$

After collision

Formulation: Collision

$$\begin{cases} \frac{\partial F^N\left(\vec{V},t\right)}{\partial t} = \frac{W_p}{\mathrm{Vol}} \sum_{1\le a<b\le N} \int_{\mathbb{S}^2} \sigma_{ab}\|\vec{v}_a - \vec{v}_b\| \left[F^N\left(\vec{V}'_{ab},t\right) - F^N\left(\vec{V},t\right)\right] \, d\vec{e} \\ F^N\left(\vec{V},0\right) = \frac{\mathrm{Vol}}{W_p \ N} \prod_{j=1}^{N} \hat{f}\left(0,\vec{v}_j\right) \end{cases}$$

- Linear in F
- RHS Identical to Boltzmann collision operator in the limit of N →∞ (propagation of chaos)

$$\nu_{\mathrm{coll}}\left(\vec{V}\right) = \frac{W_p}{\mathrm{Vol}} \sum_{1\le a<b\le N} \int_{\mathbb{S}^2} \sigma_{ab}\|\vec{v}_a - \vec{v}_b\| d\vec{e}$$

$$\frac{\partial F^N}{\partial t}\left(\vec{V},t\right) + \nu_{\mathrm{coll}} \ F^N\left(\vec{V},t\right) = \frac{W_p}{\mathrm{Vol}} \sum_{1\le a<b\le N} \int_{\mathbb{S}^2} \sigma_{ab}\|\vec{v}_a - \vec{v}_b\| \ F^N\left(\vec{V}'_{ab},t\right) \, d\vec{e} \quad (**)$$

- DSMC = Create a Jump Markov process to solve (**)
- Collisions in a cell follow a Poisson process
  - Time between collisions follows exponential distribution
- Collision Mechanics
  - Preserve collision invariants
  - Reproduce transport properties in equilibrium

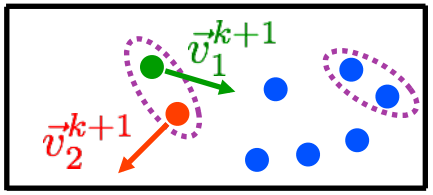
Formulation: Collision

- Various collision schemes developed over the years
- One of the most important contributor to error in DSMC simulations but accuracy difficult to gauge a priori
- Most widely used is the No Time Counter (NTC) scheme:
  - Compromise between accuracy and computational cost
  - Does not give accurate results when N < 20

1) Determine # of potential collisions during Δt

$$N_{\text{coll}} = \frac{n\Delta t}{2}(N-1)\left(\sigma v_r\right)_{\max}$$

$$N_{\text{coll}} = 2$$



2) Determine whether collision occurs

$$P_{\text{coll}}(1,2) = \frac{\sigma_{12}\ \| \vec{v}_2^{k+1} - \vec{v}_1^{k+1} \|}{\left(\sigma v_r\right)_{\max}}$$

Yes:

$$\vec{v}_1^{k+1} = C_{\text{coll}}\left(\vec{v}_1^{k+1}, \vec{v}_2^{k+1}\right)$$
$$\vec{v}_2^{k+1} = C_{\text{coll}}\left(\vec{v}_2^{k+1}, \vec{v}_1^{k+1}\right)$$

No: Velocities unchanged

DSMC

Formulation: Algorithm

For all time steps $k=0,\ldots k_{end}$

1. Move particles during Δt
2. In each cell, perform collisions between particles
3. Generate/Discard particles at boundaries
4. If in steady state sample cell properties

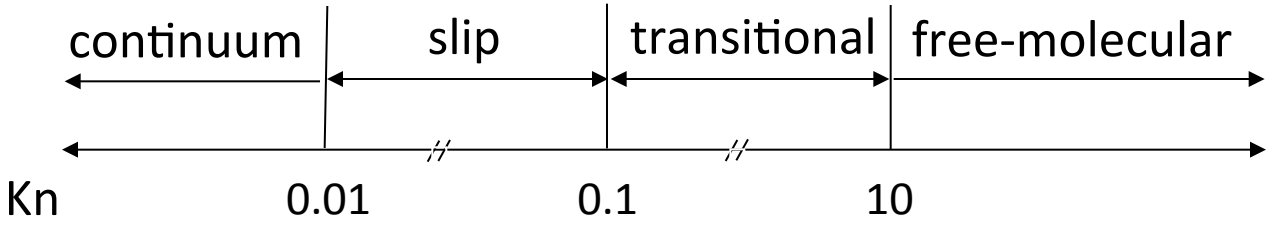
EndFor

Constraints to satisfy

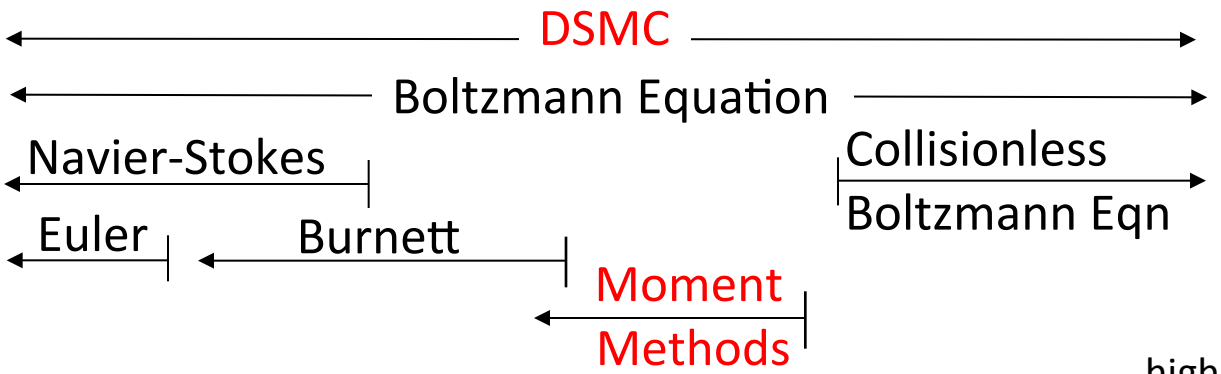
- Δt < mean collision time
- Particles cross less than 1 cell/timestep
- Δx < mean free path

Motivation

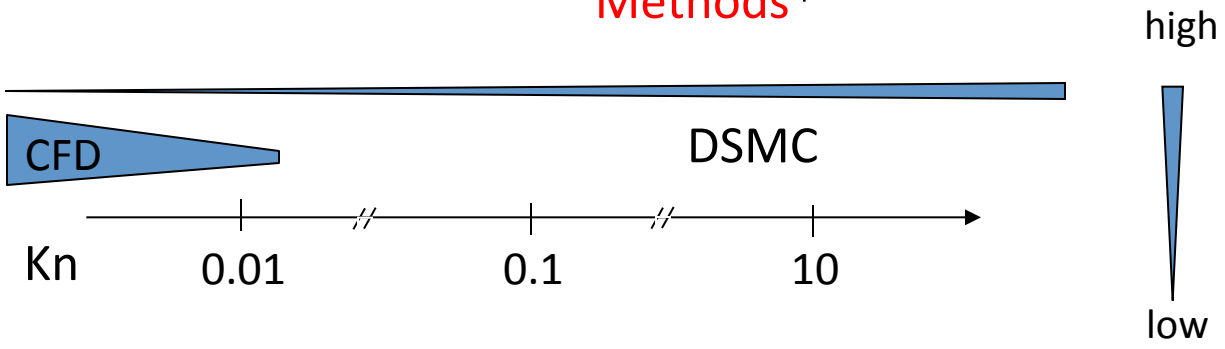
Flow Regimes:



Model Accuracy:



Numerical Performance:



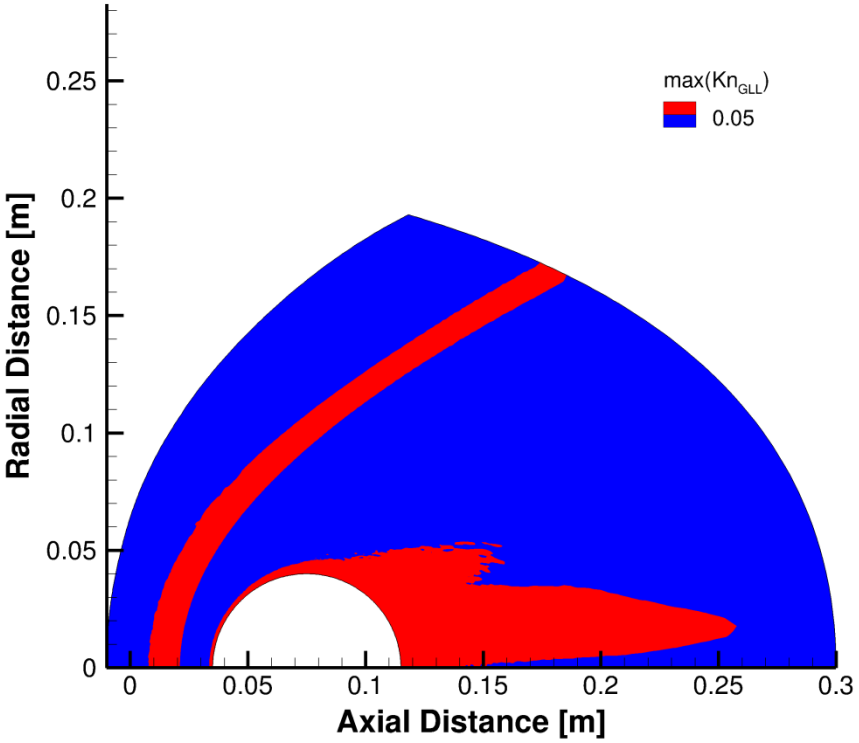
Accuracy + Performance } ⇨ hybrid approach

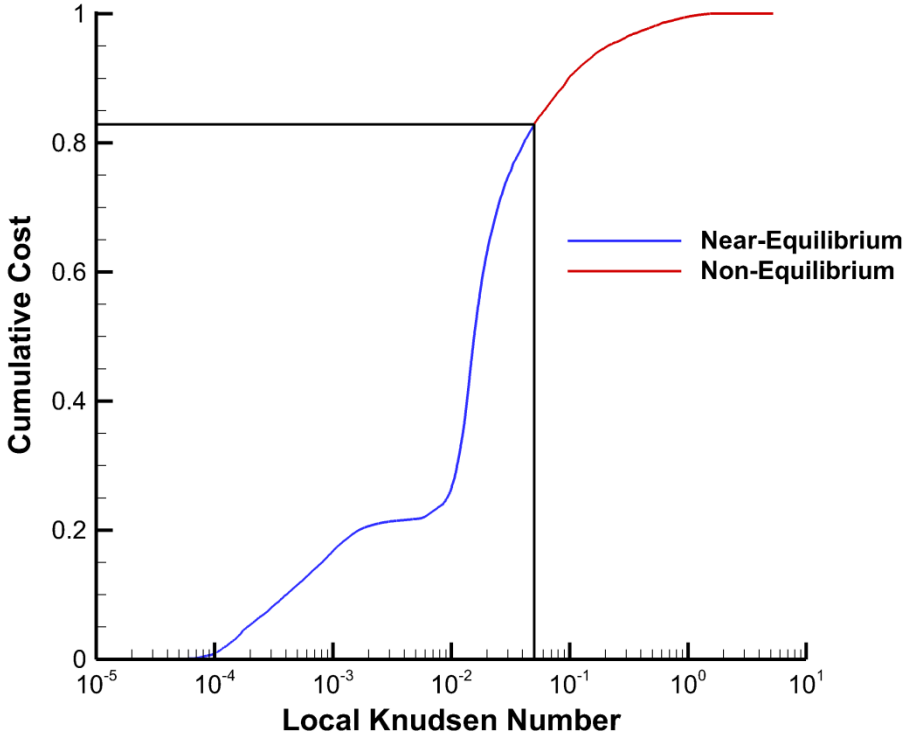
Motivation

- Kinetic theory-based simulation techniques, e. g. Direct Simulation Monte Carlo (DSMC) method, are required for nonequilibrium regions
  - Applicable for both continuum and rarefied flows, but computationally expensive for continuum flows
- Computational Fluid Dynamics (CFD) methods are numerically efficient, but physically inaccurate where continuum assumptions are invalid
- Hybrid solvers are an attractive alternative for transitional hypersonic flows
  - Continuum breakdown parameter used to divide the flow field into domains analyzed using two different, yet coupled, solution techniques

Motivation

M_∞=12, Kn_∞=0.01, N_2

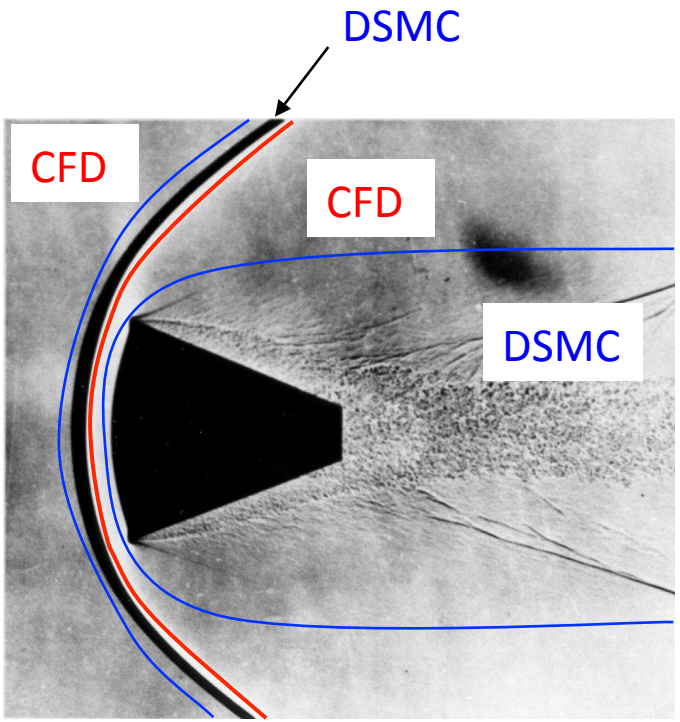




>80% of the cost of this DSMC simulation is spent solving near-equilibrium regions of the flow field!

Practical applications

- Flows in a mixed continuum/transition regime are important for some Aerospace applications
  - Capsule Aerothermodynamics at high altitude (z = 70 km)
  - Nozzle expansion flows



Formulation

- Hybrid framework that loosely couples a DSMC code (MONACO[1]) with a CFD solver (LeMANS[2])
  - MONACO
    - General, cell-based implementation of the DSMC method
  - LeMANS
    - Second-order accurate, finite volume CFD solver
- Current capabilities
  - Physical accuracy and numerical efficiency demonstrated for axisymmetric and 2D flow domains
- Requires a significant programming effort!

[1]Dietrich, S. and Boyd, I.D. Scalar and parallel implementation of the DSMC method, JCP (1994)
[2]Martin, A., Scalabrin, L.C. and Boyd, I.D., High performance modeling of reentry vehicles , JPCS (2012)

Formulation[1]

1. Obtain converged CFD solution
2. Calculate and apply continuum breakdown parameters
   – Gradient-length local Knudsen number[2] Kn_{GLL}.

$$\mathrm{Kn}_{\mathrm{GLL\text{-}Q}} = \frac{\lambda}{Q}\left|\nabla Q\right|$$

$\mathrm{Kn}_{\mathrm{GLL\text{-}Q}} < 0.05$ → Continuum regime (CFD)

$\mathrm{Kn}_{\mathrm{GLL\text{-}Q}} > 0.05$ → Transition regime (DSMC)

3. Employ *MONACO* to obtain DSMC solution in particle regions

[1]Schwartzentruber, T. E., Scalabrin, L. C., and Boyd, I. D., *JCP* (2007).
[2]Boyd, et al., *Phys. Fluids*, 7 (1995).

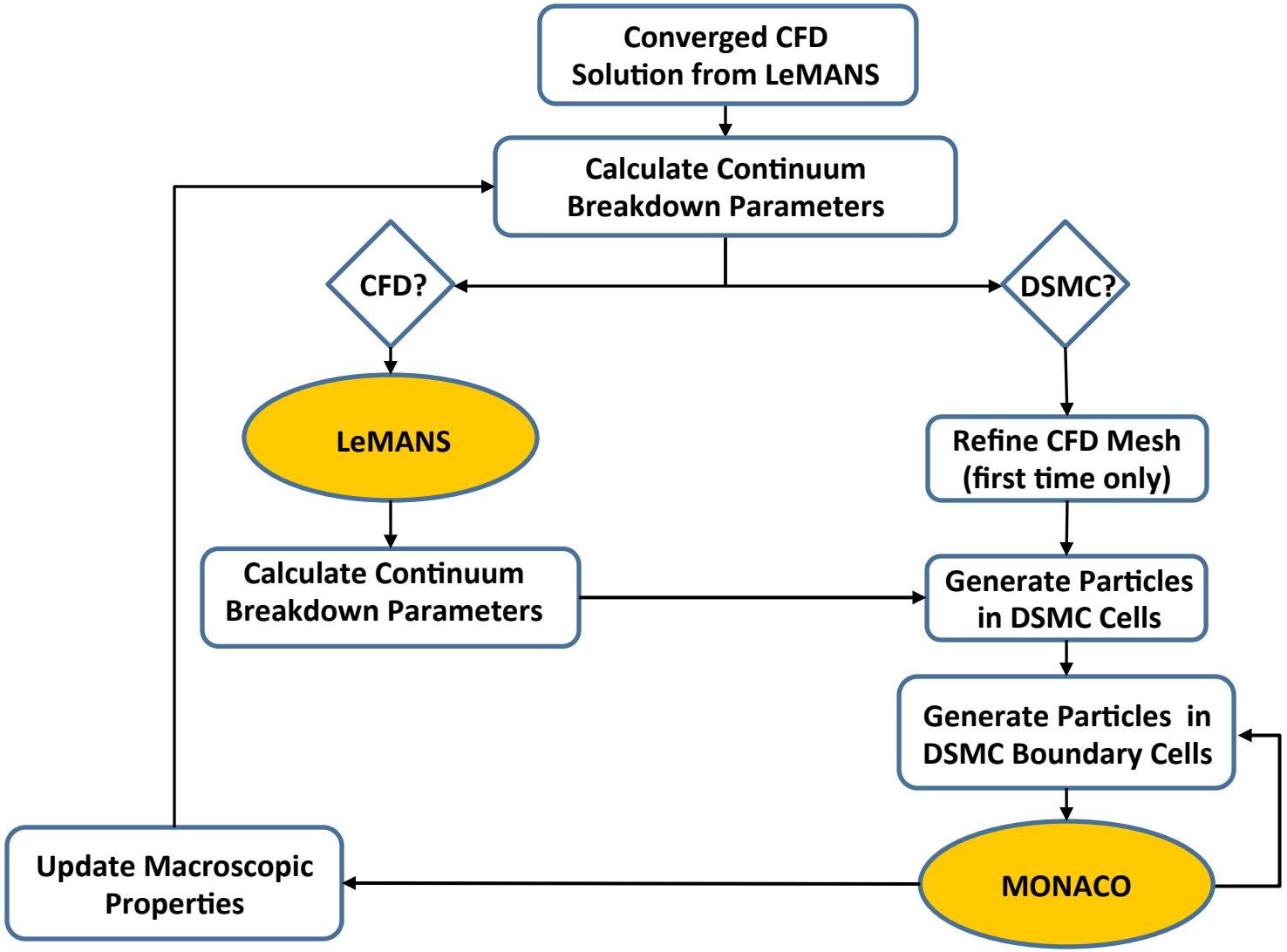
Formulation

4. After hybrid interfaces cease to move, transfer information to CFD via boundary conditions
   - Sub-relaxation average of Sun and Boyd used to mitigate statistical scatter at hybrid interfaces

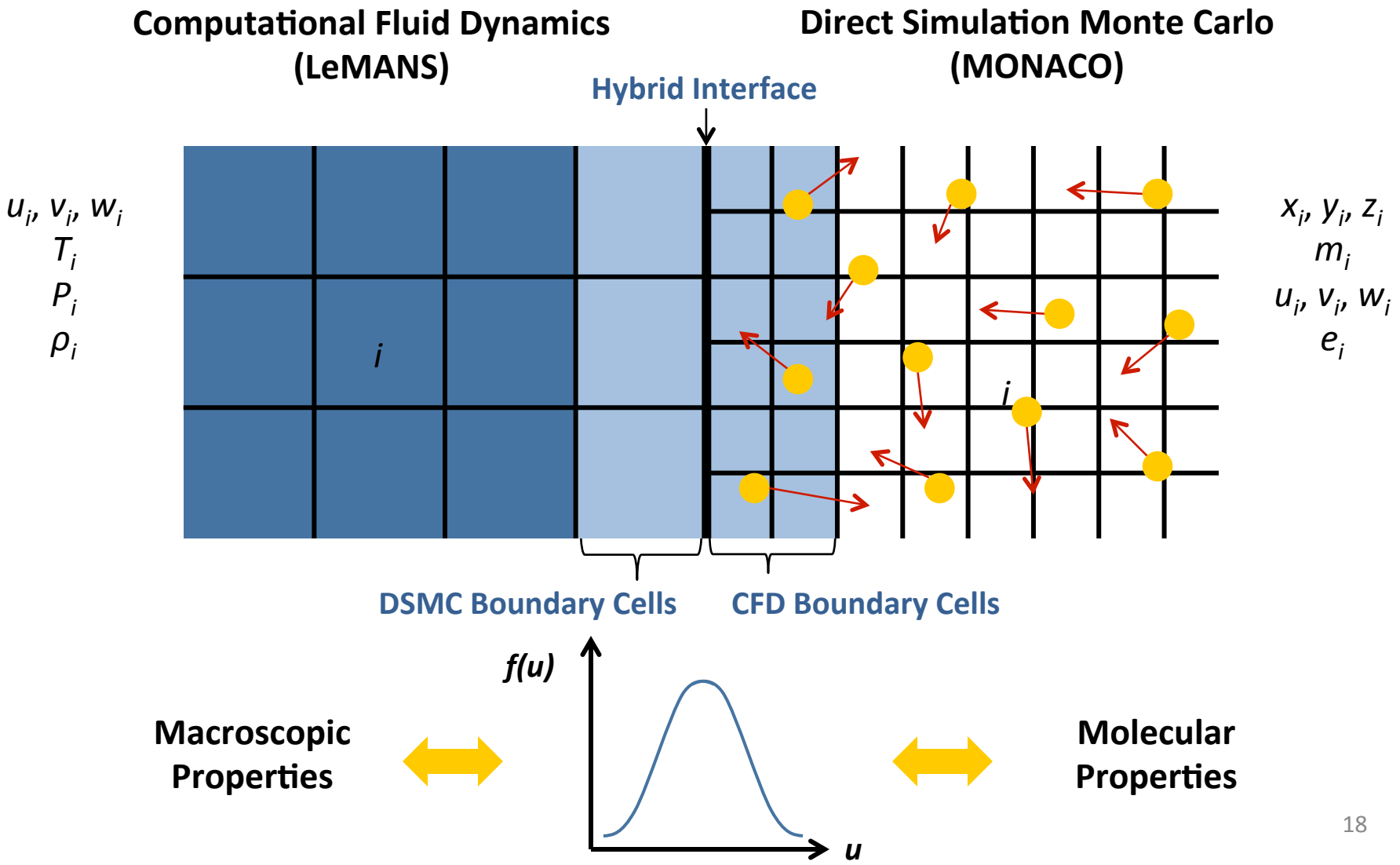
$$\overline{Q_j} = (1-\theta)\cdot\overline{Q_{j-1}} + \theta\cdot Q_j$$

5. Progress CFD solution a specified number of iterations
6. Transfer information back to DSMC through generation of simulator particles at hybrid interfaces
7. Continue hybrid cycling until interfaces cease to move, after which proceed with standard DSMC and CFD procedures

Formulation



CFD/DSMC Interface

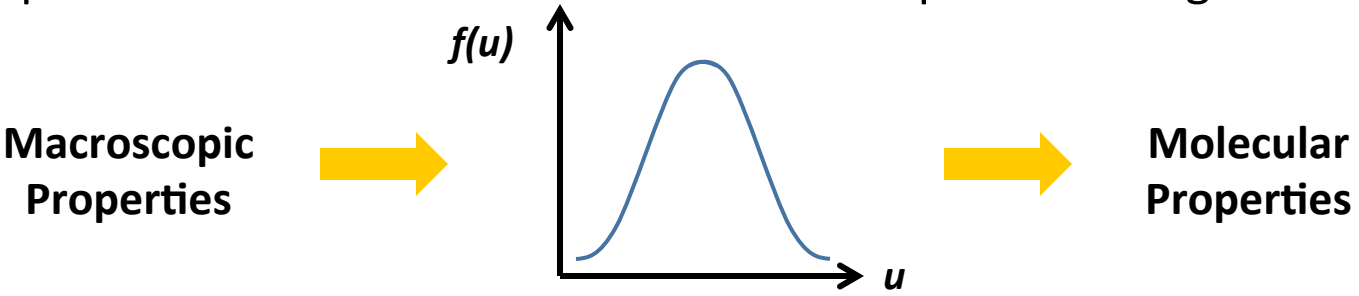


Coupling: CFD -> DSMC

- On a macroscopic level, the solution obtained using CFD can be used to completely define a Chapman-Enskog VDF
  - Chapman-Enskog VDF represents a first-order approximate solution to the Boltzmann Equation
  - Perturbation term incorporates mass diffusivity, viscosity, and thermal conductivity

$$\underbrace{f^{[1]}\left(\vec{x},\vec{v},t\right)}_{\text{Chapman-Enskog VDF}} = \phi\left(\vec{x},\vec{v},t\right)\cdot \underbrace{f^{[0]}\left(\vec{x},\vec{v},t\right)}_{\text{Maxwellian VDF}}$$

- An acceptance-rejection method proposed by Garcia, et al.[1] is used to sample random thermal velocities from a Chapman-Enskog VDF



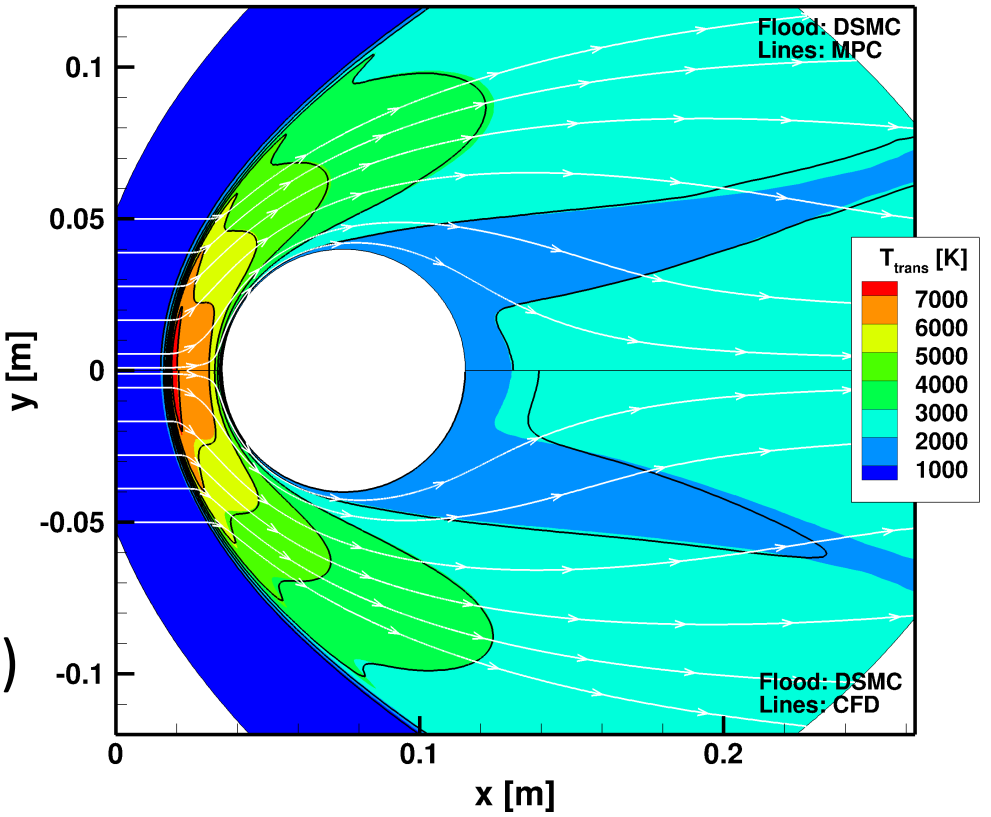
[1]Garcia, et al., *J. Comp. Phys.*, 140 (1998).

Results*

- Mach 12
- Kn_∞ 0.01
- N_2
- Rotational nonequilibrium

Error = |Hybrid – Full DSMC|

- Error <5% (most of the domain)
- Error <10% (all of the domain)



Test Case	Speedup	Memory usage
Kn = 0.002	28.1	75%
Kn = 0.01	2.94	28%

*Deschenes, T. R., Holman, T. D. and Boyd, I. D. JTHT (2011).

Challenges

- Which regions should be simulated using DSMC and which regions should be solved using CFD?
  - Requires a measure of continuum breakdown
  - Goal is to achieve a balance between physical accuracy and numerical efficiency
    - This usually means that a CFD solution is used to guess where the hybrid interfaces should be placed, but CFD does not give a completely accurate picture
  - Most measures of continuum breakdown are heuristic, either in their formulation or in their choice of cutoff value
    - No guarantee of applicability to every flow field
- Difficult to implement
  - Need a DSMC and a CFD code
  - Internal energy and chemical reaction adds complexity

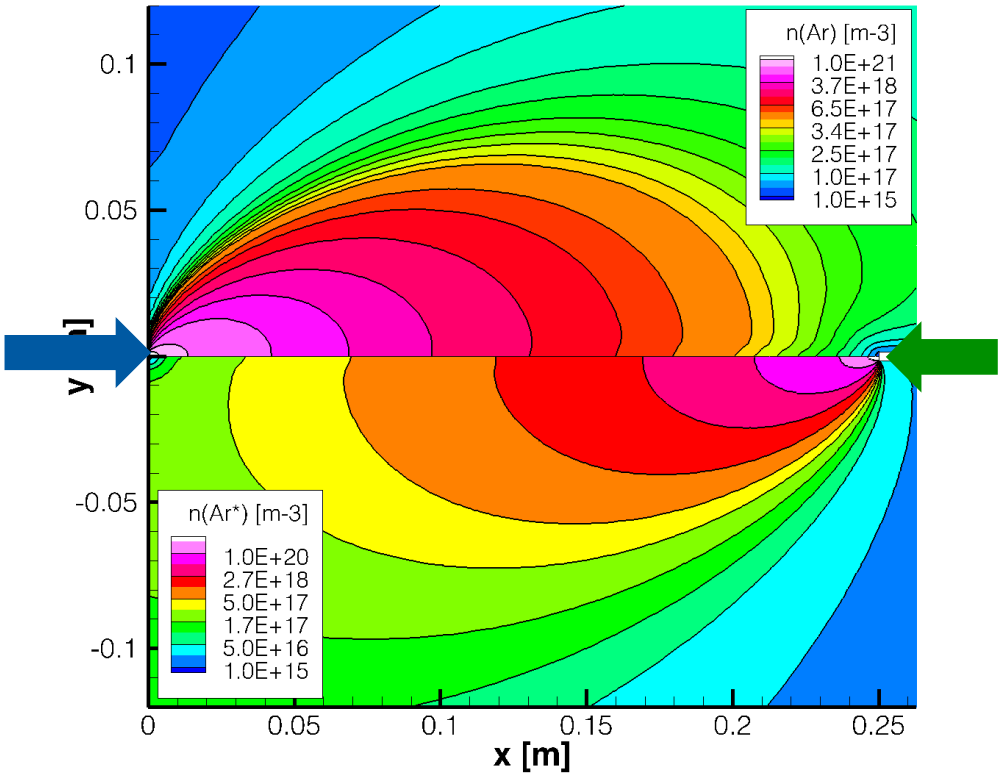
Increasing the computational efficiency of DSMC

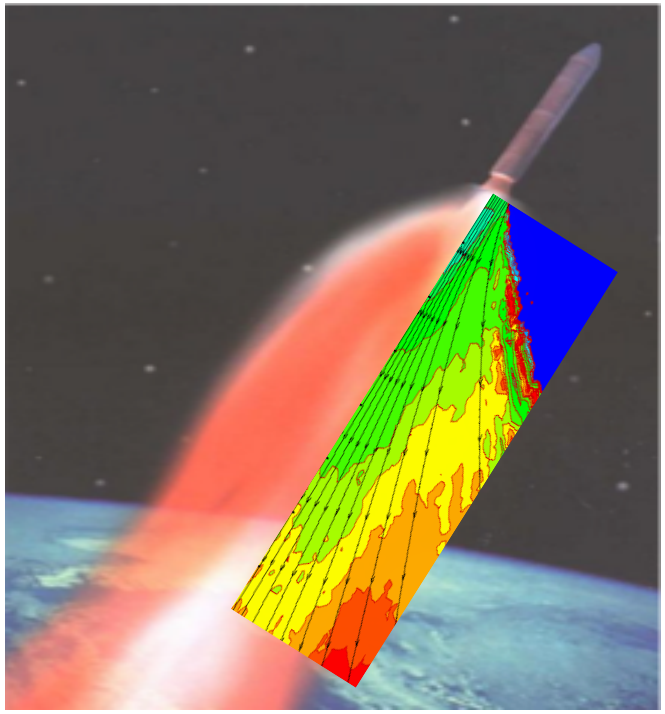
*An Adaptive procedure for the time step and weights**

*C. Galitzine and I. D. Boyd, *J. Comp. Phys.,* Accepted (2014).

Motivation/Applications

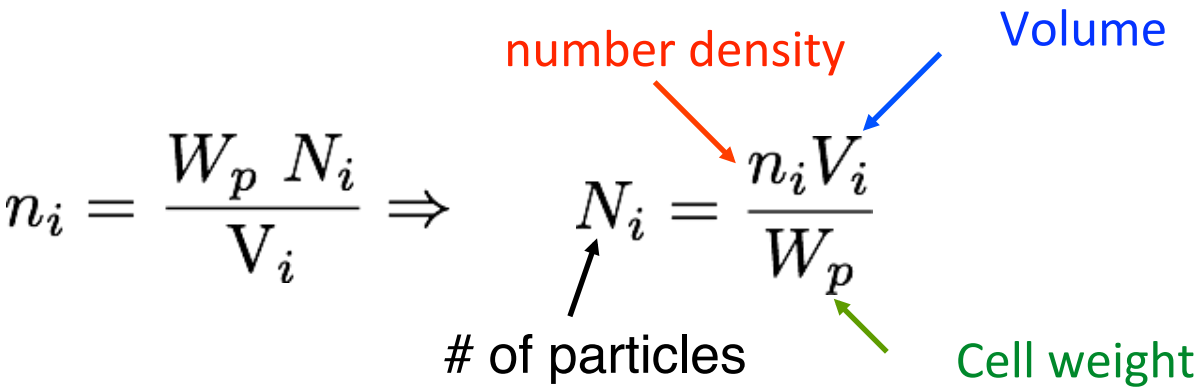
- Flows with large scale disparities





- Trace species in chemically reacting/radiating flows

Motivation: Cell/Species weights



- n(x) or V(x) widely vary for many flows (esp. Axisymmetric flows)
- Different species have different number densities and distributions

→ N is either very large or very small

Standard DSMC algorithm very inefficient for multispecies flows with large density variations

Motivation: Time step

$$\begin{cases} \Delta t < \text{Mean collision time} \sim n^{-1} \times T^{-1/2} \\ \Delta t < 0.1 \dfrac{\Delta x}{\langle v \rangle} \Leftrightarrow \text{No more than 1 cell crossed / time step} \end{cases}$$

Cell length

Average particle velocity

- n(x) , <v>, Δx all vary widely for many flows
- Δt only dictated by a small portion of the flow
- Low Δt produces large time correlation of samples

DSMC algorithm with uniform Δt very inefficient for flows with large density, velocity or cell size variations

Spatially varying weights & time step

1) Vary cell weights in space

$$N_i = \frac{n_i V_i}{W_{p,i}}$$

$n_i V_i$ ← Volume

$W_{p,i}$ ← Weight of cell i

2) Introduce species relative weights

# of particles of species j in cell i

$$N_{j,i} = \frac{n_{j,i}\ V_i}{W_{p,i}\ W_{\mathrm{rel},j,i}}$$

$n_{j,i}$ ← number density of species j in cell i

$W_{\mathrm{rel},j,i}$ ← Relative weight of species

$W_{rel,j,i} < 1$

3) Vary time step in space

$$\Delta t \Rightarrow \Delta t_i$$

- Spatially varying weights and time step are widely used
- Uniform species relative weights are widely used

} Set before the simulation

Goals

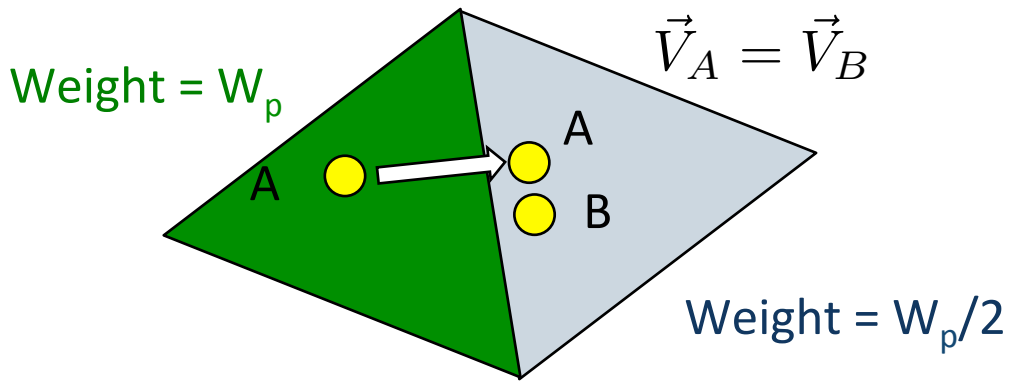
1. Determine $\{W_{p,i}\}_i$ and $\{W_{rel,j,i}\}_{j,i}$ to have $N_{p,want}$ particles for all "j" species in all "i" cells.

2. Maximize Δt_i in all "i" cells

- $\{W_{p,i}\}_i$, $\{W_{rel,j,i}\}_{j,i}$ and $\{\Delta t_i\}_i$ have to be tailored to each individual simulation
- Have to be determined during the simulation: "adaptive"
- Have to modify DSMC collision/move algorithm

Spatially varying weights & time step

- Need to change particle movement procedure when W_p, W_{rel} or Δt varies in space to conserve fluxes between cells



- Cloning procedure is a source of error
  - Particles with identical properties cause samples to be correlated
  - Collision rate is inaccurate $P_{coll} \sim (\sigma g) \sim (\sigma \|\vec{v}_A - \vec{v}_B\|)$
  - Difficult to quantify
- Error important at high Kn with few collisions

Integration in a DSMC code

For time step k=0,….,N_{step}

Every 1,000 steps

UpdateΔt() and UpdateW_p()

or

UpdateW_{rel}()

Adaptive procedure

For all cells i=1,….,N_{cells}

1. PerformCollisions()
2. MoveParticles()
3. ApplyBC()

"Standard" DSMC procedure

- "Independent" of DSMC algorithm
- Fairly easy to implement in existing DSMC codes

Formulation: Weights: UpdateW_p()

$\{W_{p,i}\}_i$ is periodically updated during the convergence to steady state

For all i cells

1. Update weights

$$W_{p,i}^{k+1} = \frac{\overline{N}}{N_{p,\text{want}} \times N_{\text{spec}}} W_{p,i}^{k}$$

2. Smooth weight

Laplacian smoothing $\nabla^2 W_p^{k+1} = 0$

3. Limit variations and bound values

4. Update flow

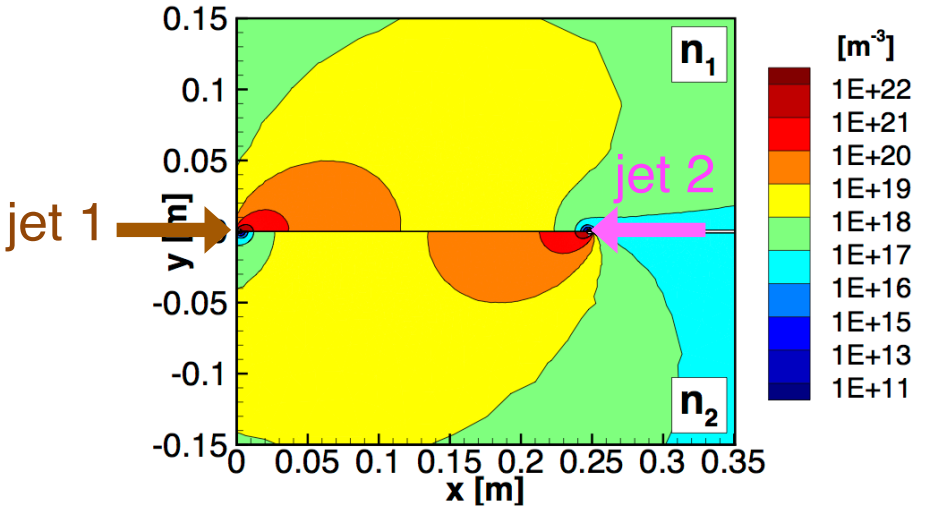
$$\frac{1}{2} < \frac{W_p^{k+1}}{W_p^k} < 2 \ , \ W_{p,\min} < W_p^{k+1} < W_{p,\max}$$

Create/destroy particles ~ $\mathcal{U}\left(\frac{W_p^{k+1}}{W_p^k} - 1\right)$ $\quad N_{j,i}^{k+1} = N_{j,i}^{k} \frac{W_{p,i}^{k}}{W_{p,i}^{k+1}}$

EndFor

Test Case

- Axisymmetric geometry
- 3 species
- Knudsen number = 0.01
- Mach number = 1



Species #1

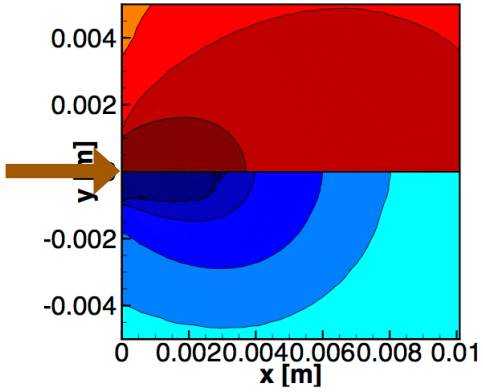
1.25 10^{23} m^{-3}

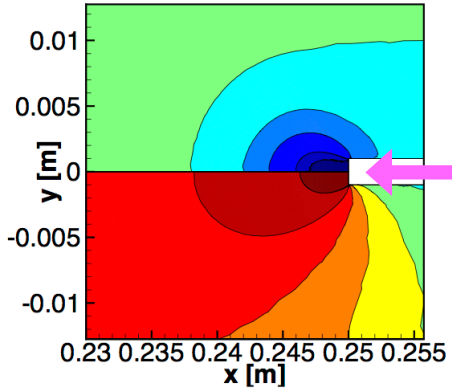
jet 1

Species #3

1.25 10^{21} m^{-3}

("Trace" species)



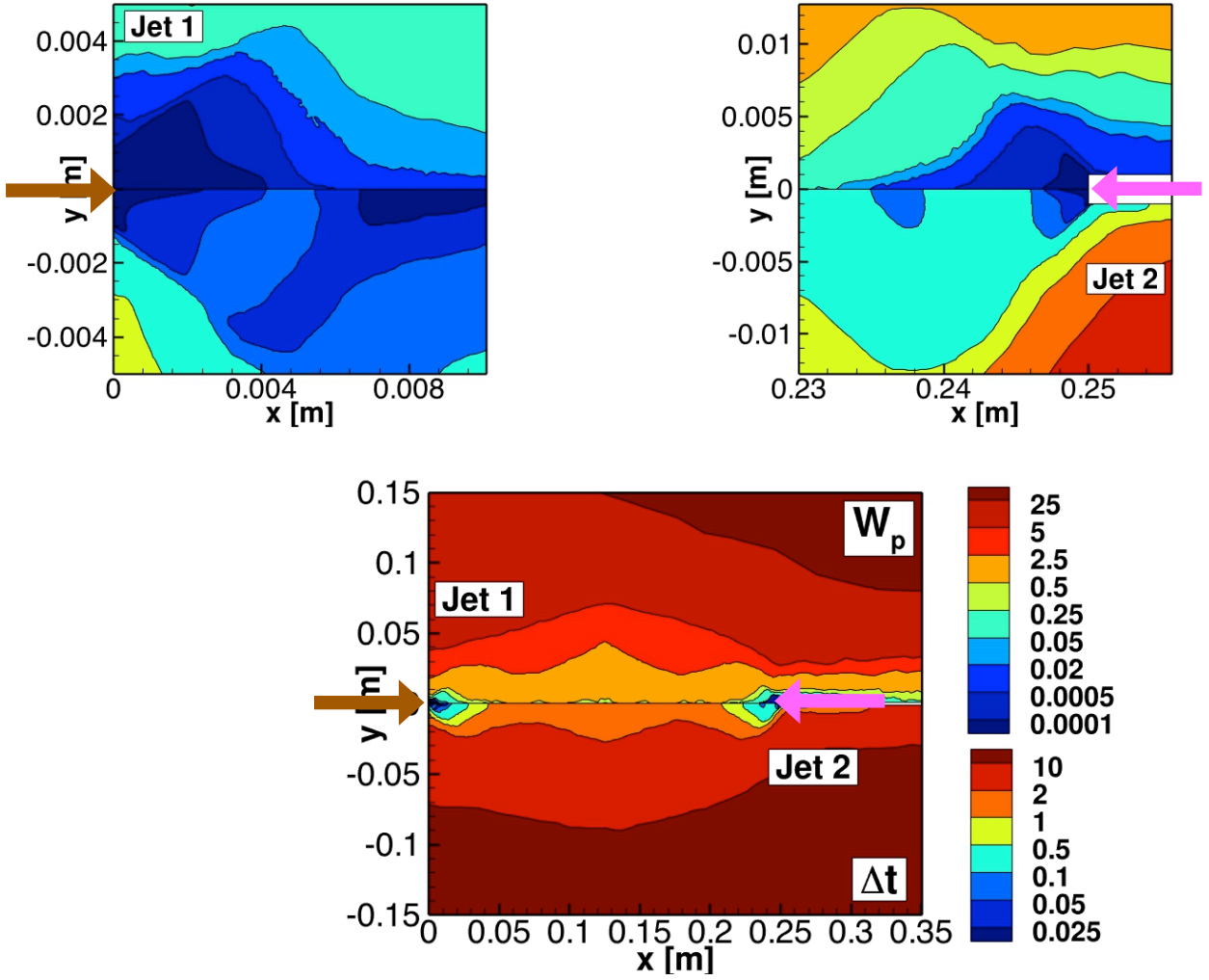


jet 2

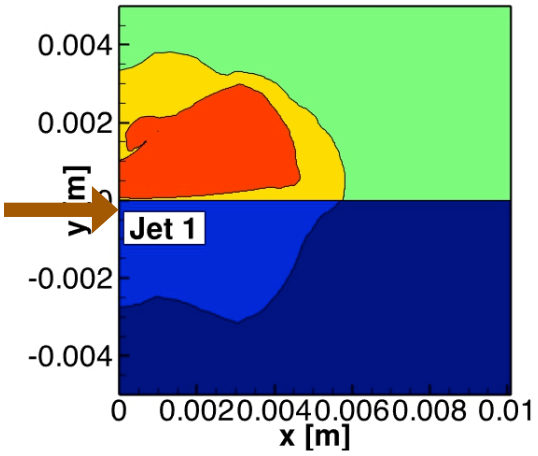
Species #2

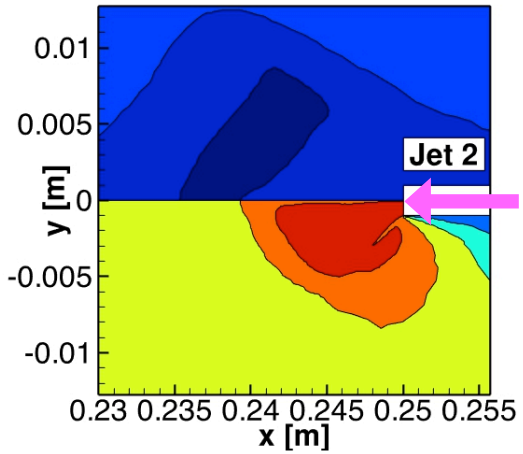
1.25 10^{23} m^{-3}

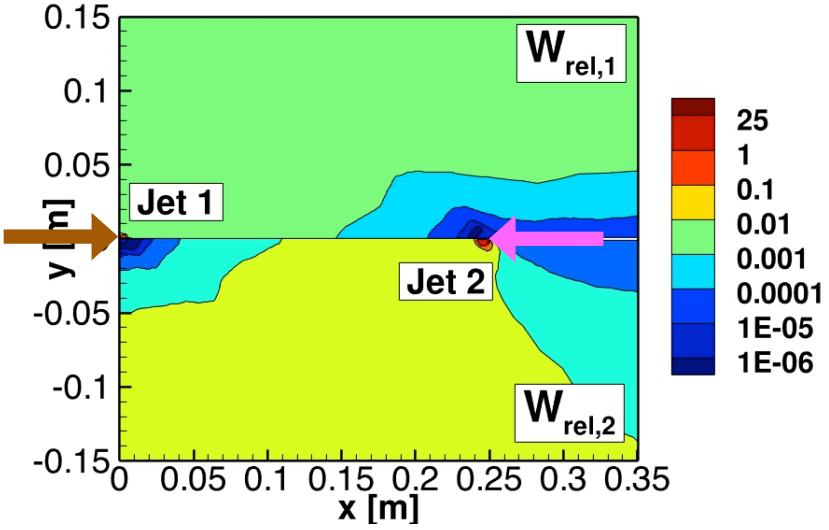
Results- Wp, Δt



Results-W_{rel}





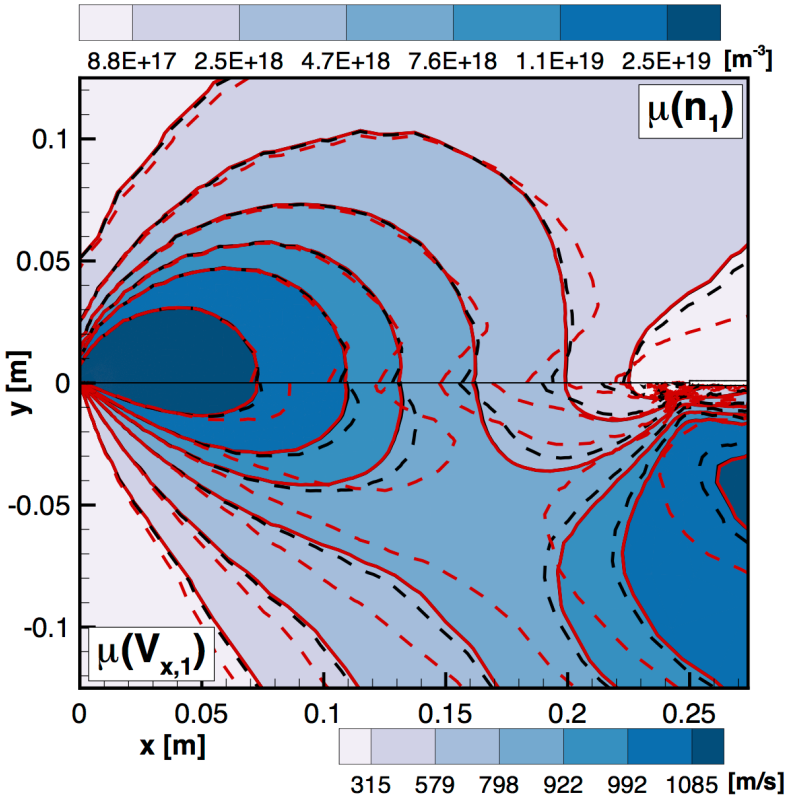


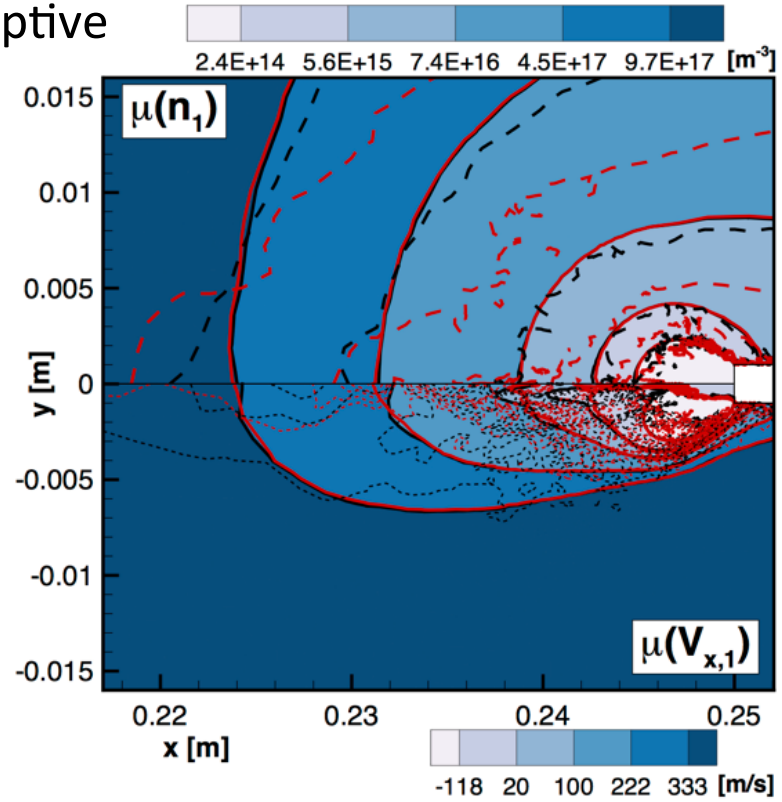
Comparison with standard DSMC

- "Standard" = {weight + time step}
- Adaptive = {weight + time step + relative weights}

Same # of particles

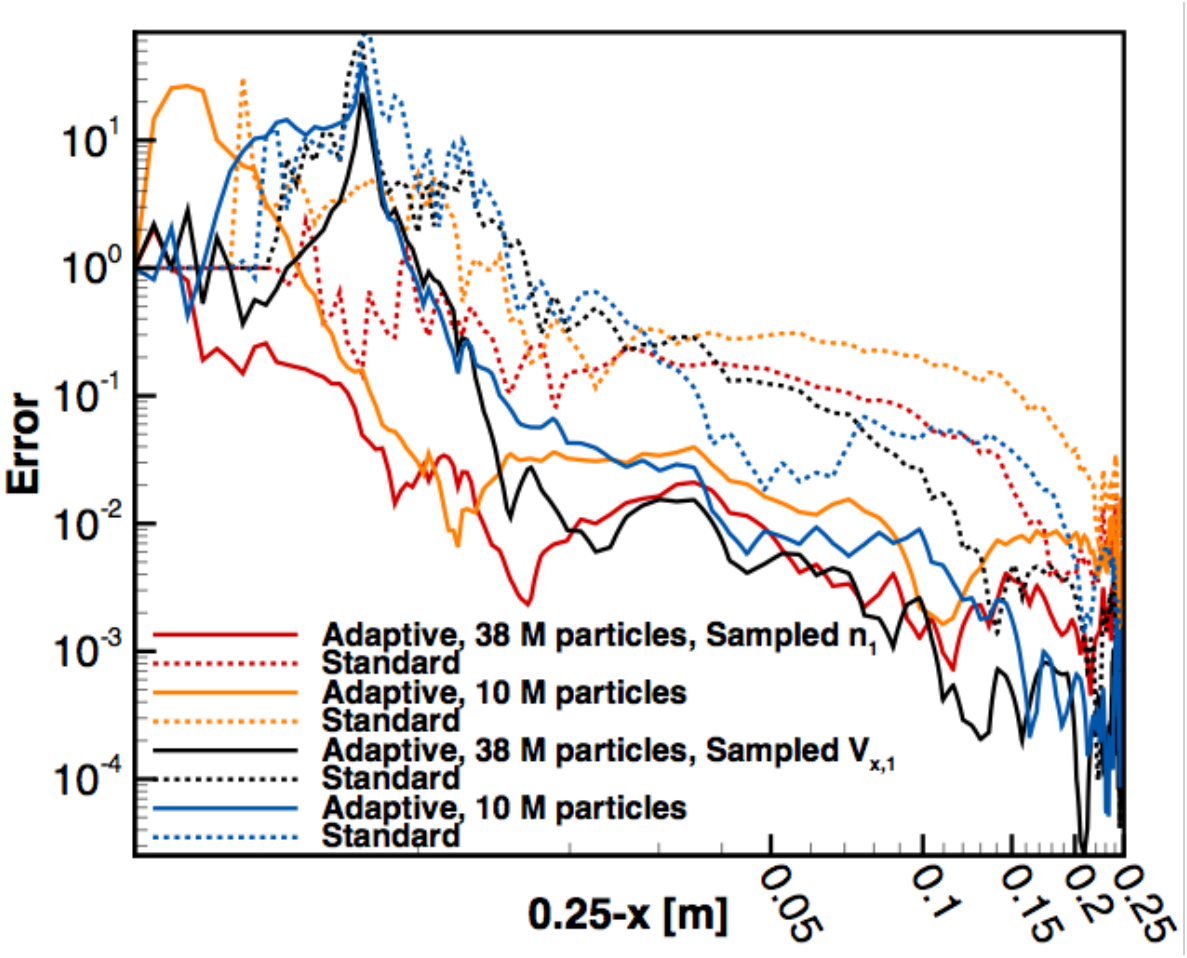
Standard
Adaptive





Comparison with "standard" DSMC

- Large error reduction observed for the same total number of particles



Conclusions

- "Standard" DSMC very inefficient for certain flows
  - Multispecies flows with large density gradients → Adaptive method
  - Low Mach number flows (not discussed here)
- Have to modify procedure to improve efficiency
  - Multiple formulations possible
  - Introduces new problems
- Could Moment Methods be a viable alternative?
  - For low Kn, YES
  - For high Kn, NO → Direct Boltzmann solver

Quantifying the error in DSMC simulations

*A Framework for error analysis in DSMC**

* C. Galitzine and I. D. Boyd, Submitted to *JCP* (2014).

Error Framework

- Sources of error considered:

Numerical Error

1. Not enough samples

Convergence Error

2. Not enough particles
3. Too large time step

- Other sources of "error" are ignored:

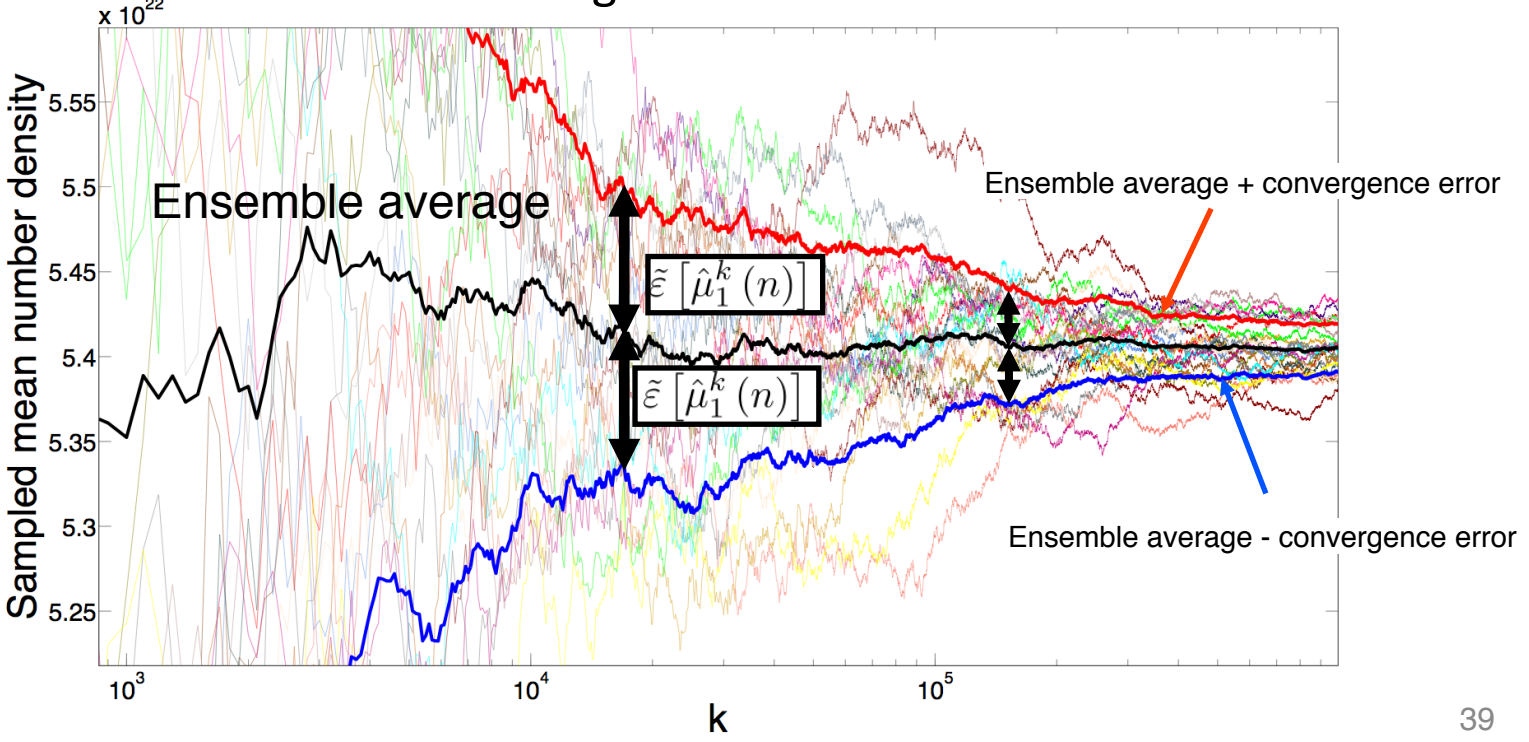
1. Effect of mesh
2. Non-convergence to Boltzmann equation
3. Inaccuracies of physical models / actual gas

Error Framework: Convergence error

Def: convergence error = Ensemble standard deviation of estimator

$$\tilde{\varepsilon}\left[\hat{\mu}_1^k\right] = \left(\left\langle \hat{\mu}_2^k \right\rangle - \left\langle \hat{\mu}_1^k \right\rangle^2\right)^{1/2}$$

- Obtained by running multiple simulations and calculating variance of estimator
- Monotonically decreases when k $\rightarrow \infty$
- No Residual to evaluate convergence in DSMC simulations



A central limit theorem to predict convergence error

$$\hat{\mu}_1^k(n) = \frac{1}{k}\left(n^0 + n^1 + \ldots + n^{k-1} + n^k\right) \qquad \lim_{k\to\infty} \hat{\mu}_1^k(n) = \mu_1(n)$$

If samples were statistically independent:

$$\mathrm{Var}\left(\hat{\mu}_1^k\right) = \frac{\mathrm{Var}(n)}{k}$$

and

Normal distribution

$$\hat{\mu}_1^k(n) \sim \mathcal{N}\left(\mu_1(n), \frac{\mathrm{Var}(n)}{k}\right)$$

Central limit theorem

BUT samples are NOT statistically independent:

$$\mathrm{Var}\left[\hat{\mu}_1^k(n)\right] \approx \frac{\mathrm{Var}(n)}{k} + \frac{2}{k}\sum_{k'=1}^{K}\gamma_n(k') \equiv \frac{\tilde{\sigma}^2(n)}{k}$$

≡ Convergence error

$$\gamma_n(k') \triangleq \mathbb{E}\left[n^{k''} \times n^{k''+k'}\right] - \mathbb{E}\left[n^{k''}\right] \times \mathbb{E}\left[n^{k''+k'}\right]$$

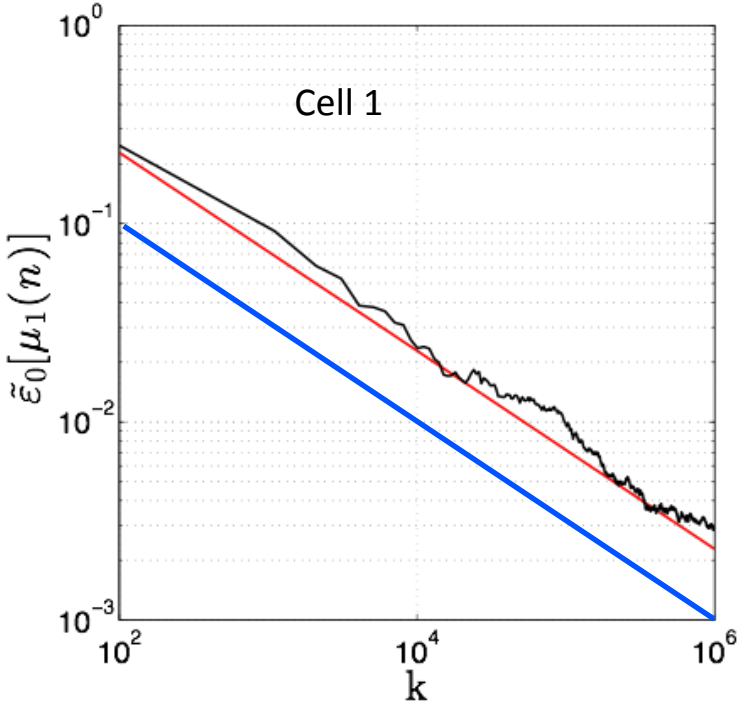
$$\hat{\mu}_1^k(n) \sim \mathcal{N}\left(\mu_1(n), \frac{\tilde{\sigma}^2(n)}{k}\right)$$

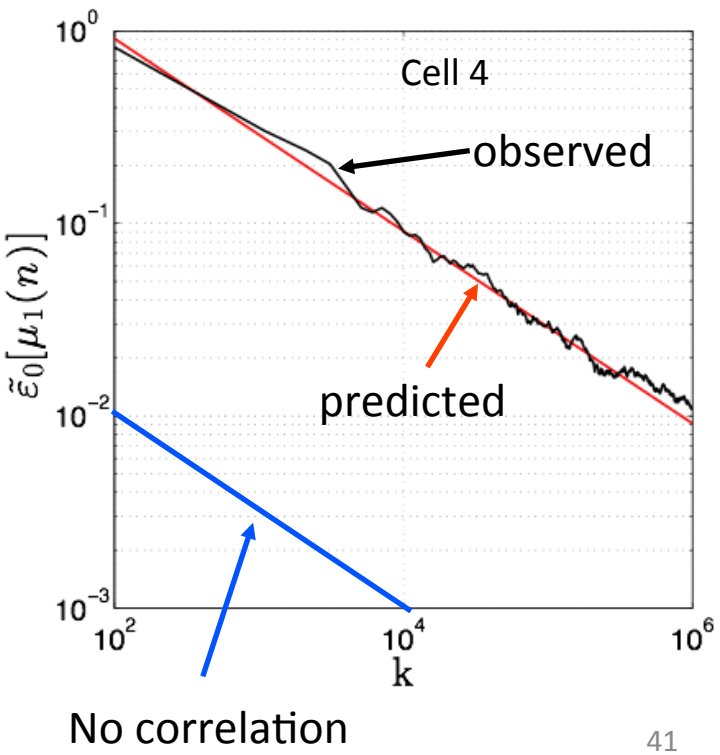
Extended central limit theorem

Allows the calculation of the convergence error from the autocorrelation spectrum

Evolution with number of samples

- Extended central limit theorem predicts evolution very well
- Can be used to predict convergence error (and the number of steps required to achieve a specified convergence level)
- Correlation results in large increases in observed standard deviation





Error Framework: Numerical error

DSMC samples depend on numerical parameters:

$$\mu_1\left(n, W_p, \Delta t\right) = \lim_{k\to\infty} \hat{\mu}_1^k\left(n\right)$$

Number of particles
(Each computational particle represents Wp physical particles)

Time step

Def: numerical error = Error due to $W_p \neq 1$ and $\Delta t \neq 0$

$$\bar{\varepsilon}\left[\mu_1\left(n, W_p, \Delta t\right)\right] = \left|\mu_1\left(n, W_p, \Delta t\right) - \mu_1\left(n, W_{p0}, \Delta t_0\right)\right|$$

Numerical error

Exact solution

- The "Exact solution" is obtained by varying $W_p \to 1$ and $\Delta t \to 0$ up until no variation is obtained in the solution (similar to grid convergence study for CFD)

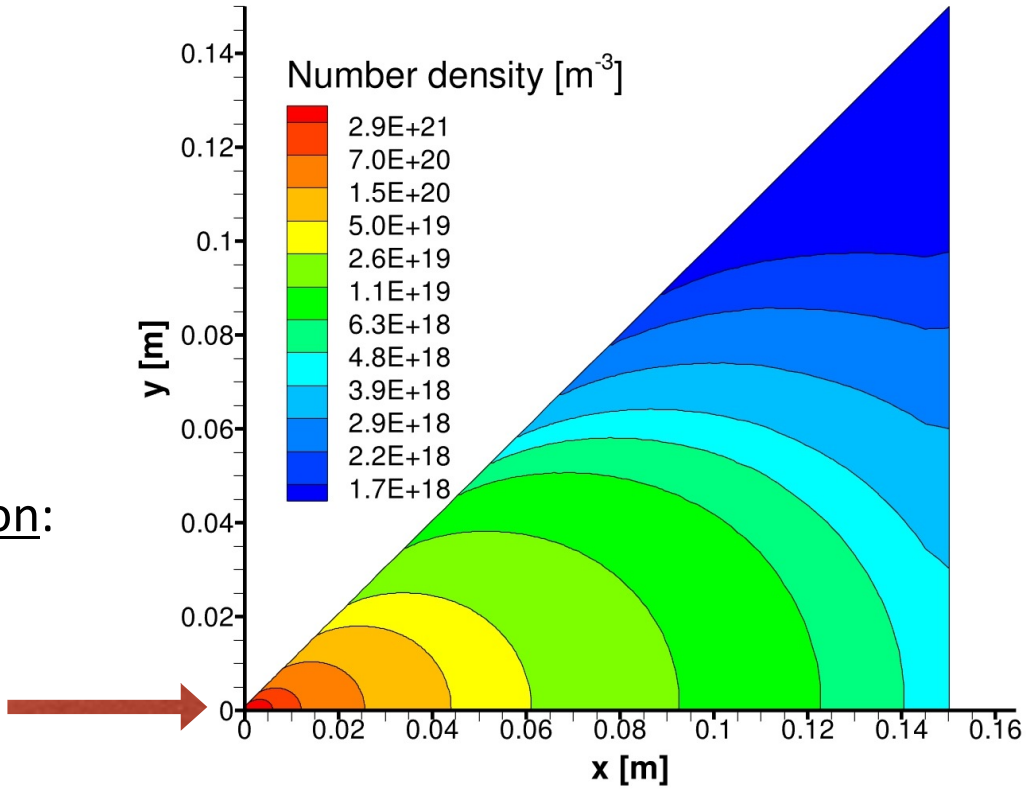
Test Case

- Simple Argon gas, axisymmetric geometry, fixed quad mesh $(\Delta x/\lambda)_{max} = 0.7$
- Spatially constant particle weight W_p and time step Δt
- "Complex" test case, not channel flow

<u>Flow Condition</u>:

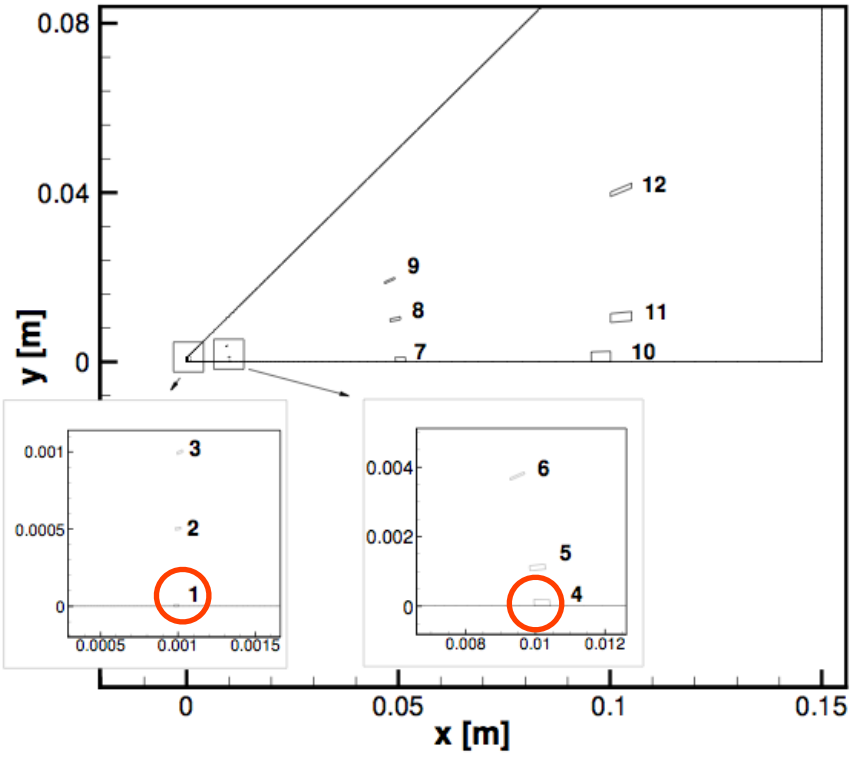
M = 1.0

Kn = 0.01



Test Case

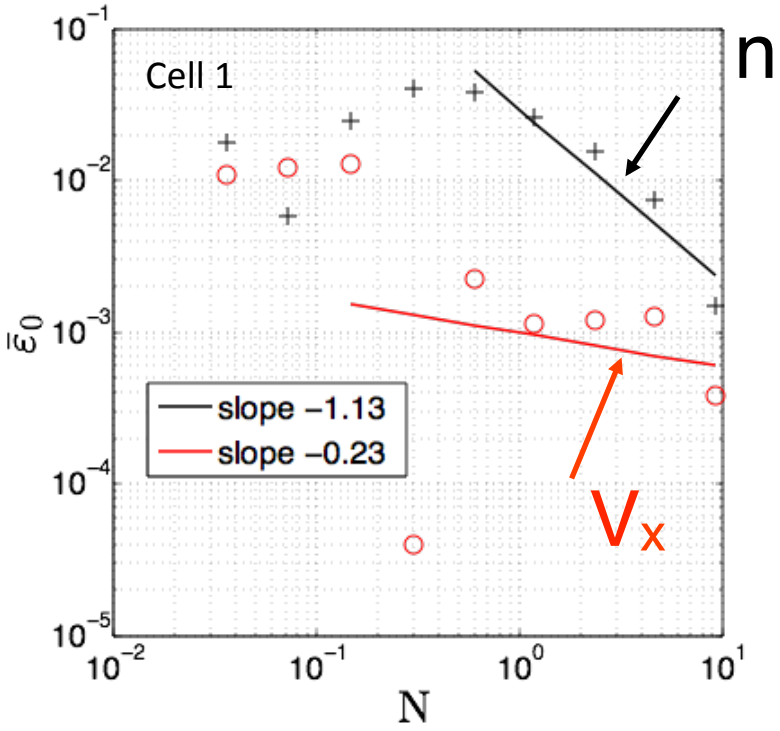
- Study focuses on cellwise error (≠ global error)
- Cellwise error of greatest interest to practitioners
- Error examined in discrete cells of the simulation

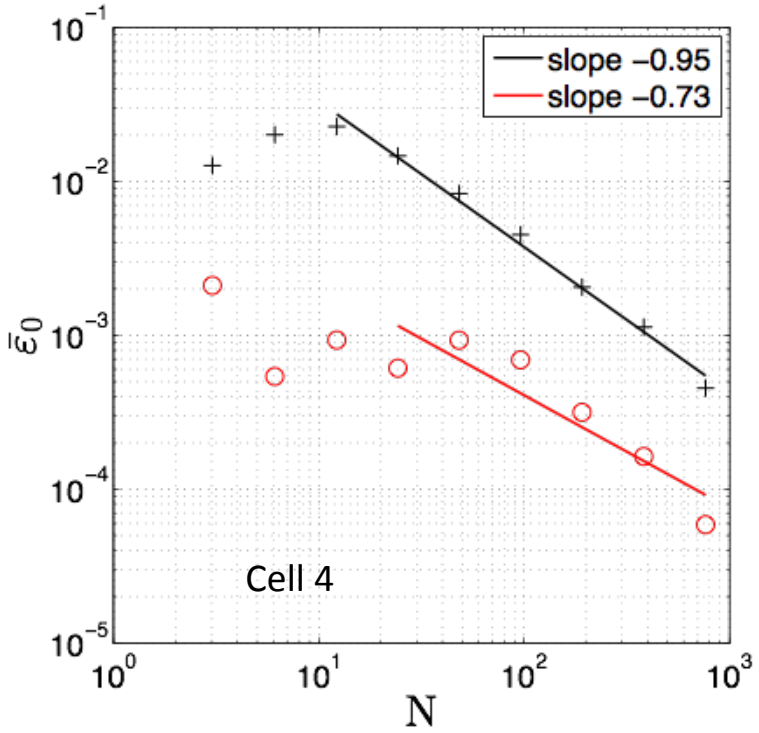


Influence of the number of particles

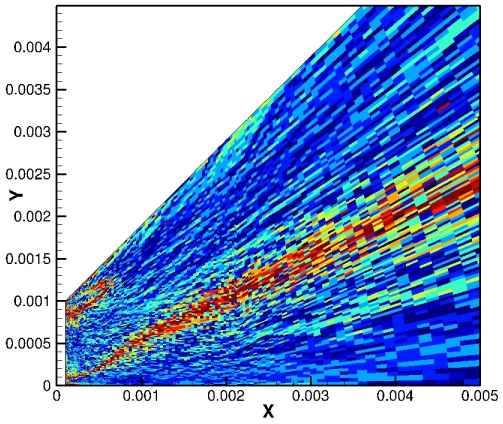
- No relation between error and number of particles
- Power law observed for the numerical error for n and V
- Large spatial variations observed for C and α

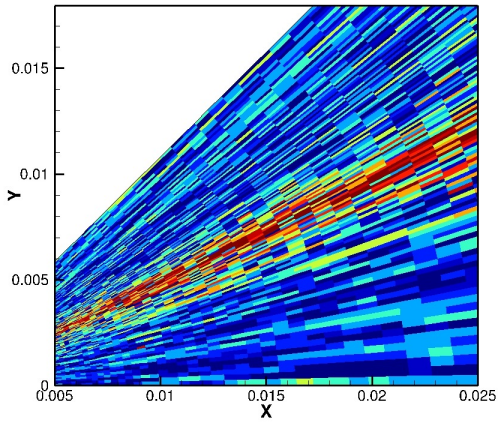
$$\bar{\varepsilon} = C \times N^{-\alpha}$$

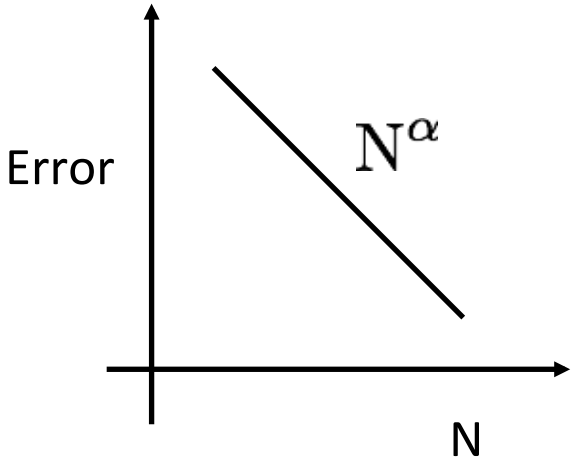


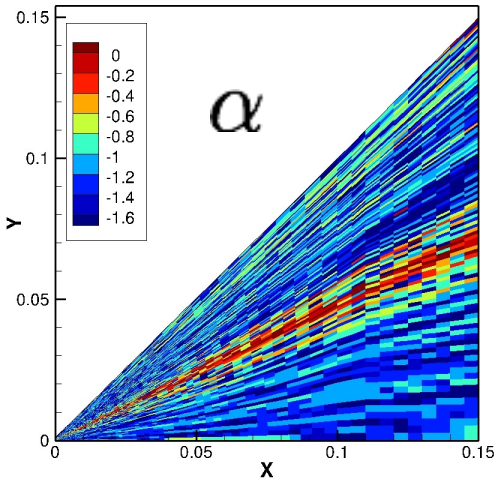


Spatial Variation







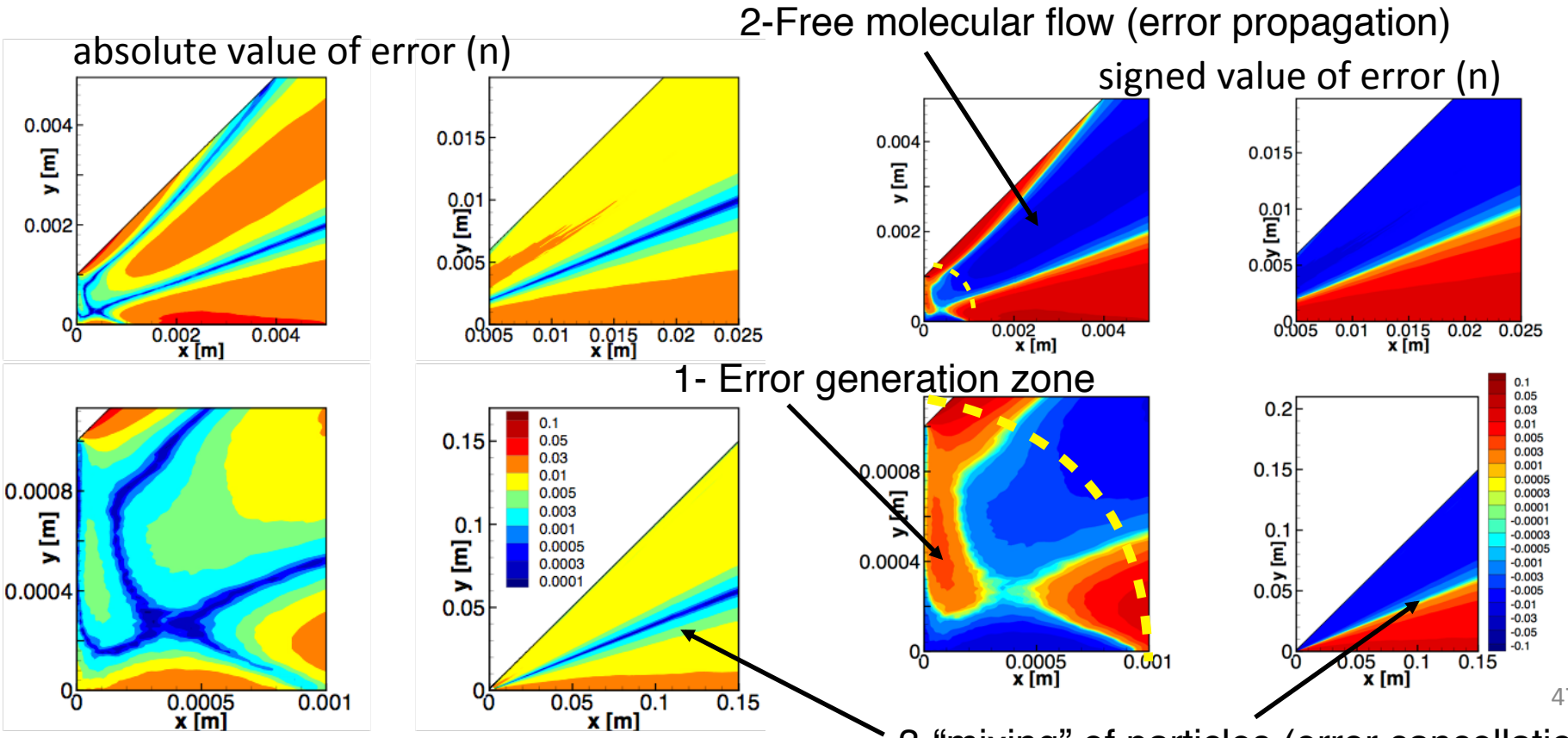


- Large spatial variation of scaling exponent
- Coherent patterns observed

Spatial Variation

- Numerical error follows coherent patterns dictated by trajectories of particles

1. Error generated in highly collisional zone
2. Few collisions → error propagates
3. Particle mixing → error cancellation



Conclusions

Can predict a priori the convergence error

- Significant for high speed flows without many collisions
- Overlooked/Not considered by many

Not being able to quantify the numerical error a-priori is problematic

- Cannot use PDE error analysis framework
- Have to consider the modeling of collisions

Very little research on this topic

Summary

3 Challenges for DSMC

1. Simulation of low Kn flow
   Hybrid DSMC/CFD
   - Breakdown criterion not general enough?
   - Need to quantify accuracy
   - Onerous to implement

2. Numerical efficiency of DSMC
   Adaptive technique
   - Non standard DSMC necessary for many flows
   - Not very rigorous/introduces additional error

3. Error quantification
   Error analysis framework
   - Numerical error difficult to predict/quantify
   - Numerical method and physical model are intertwined

Thank you!

Formulation: Time step: UpdateΔt()

$\{\Delta t_i\}_i$ is periodically updated during the convergence to steady state

For all i cells

1. Update time step

$$0.05\ \frac{\Delta x_i}{\langle v \rangle_i} \leq \Delta t_i \leq 0.1\ \frac{\Delta x_i}{\langle v \rangle_i}$$
$$\Delta t_i \leq 0.2\ \tau_{\mathrm{mct},i}$$

2. Smooth time step

Laplacian smoothing

$$\nabla^2 \Delta t^{k+1} = 0$$

3. Limit variations and bound values

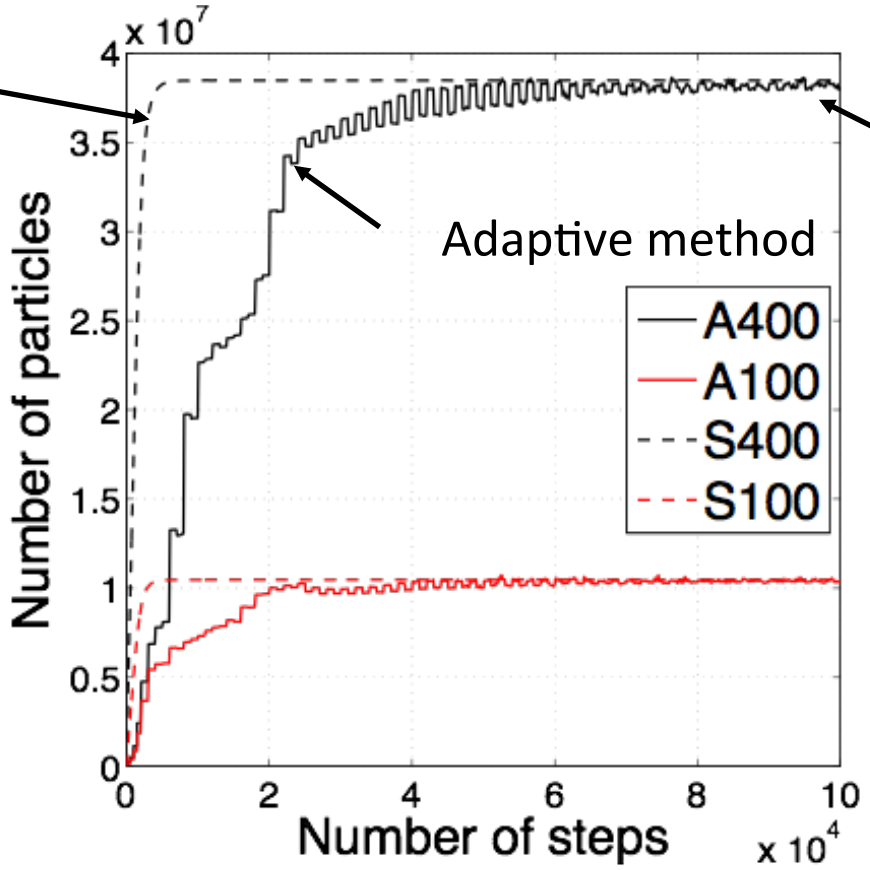
$$\frac{1}{2} < \frac{\Delta t_i^{k+1}}{\Delta t_i^k} < 2\ ,\ \Delta t_{\min} < \Delta t_i^{k+1} < \Delta t_{\max}$$

4. Update flow

EndFor

Integration in a DSMC code

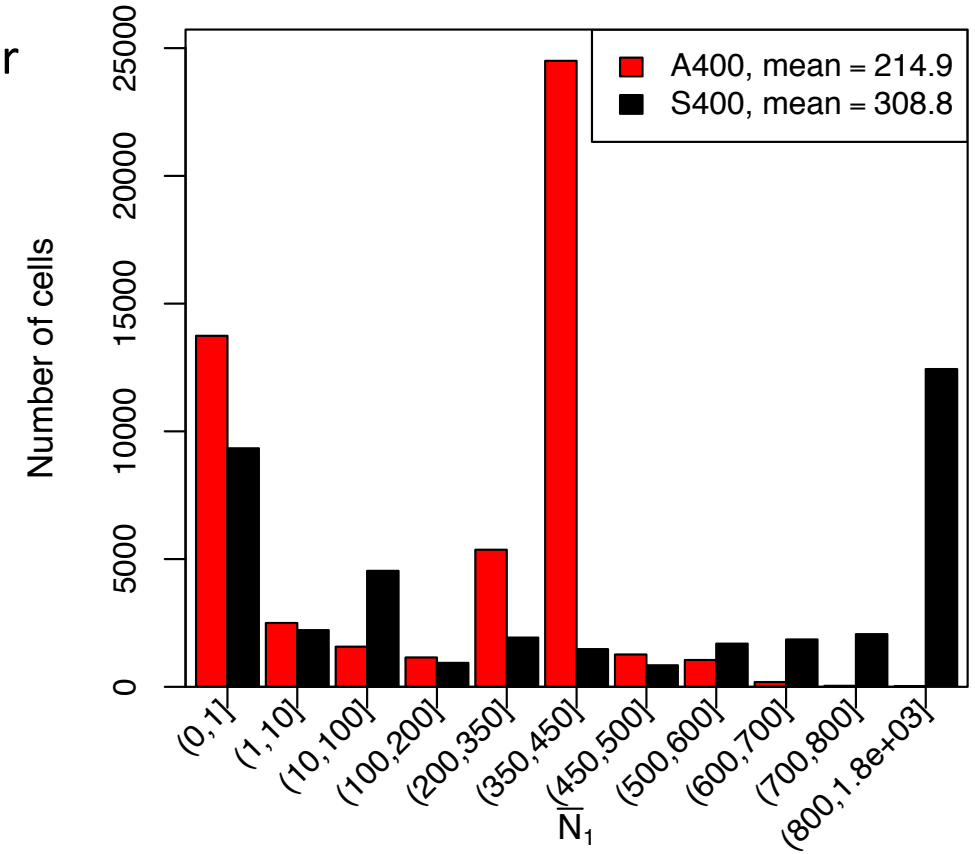
Weight & Δt set at k = 0



Steady state

Distribution of number of particles

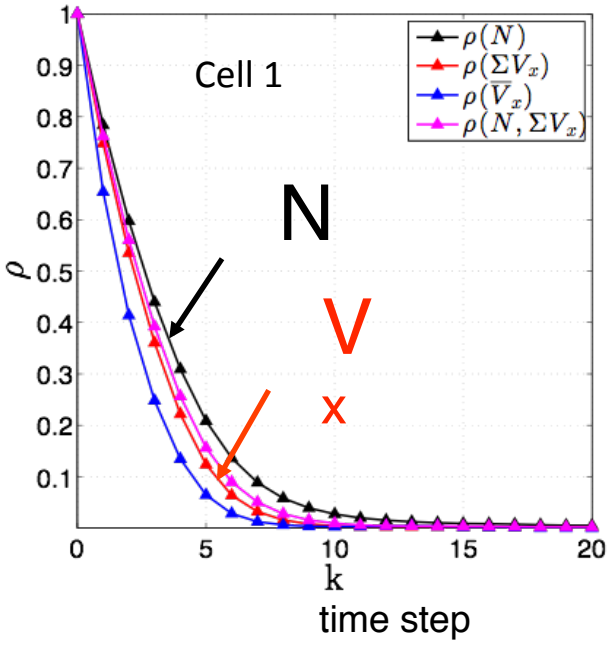
- "Standard" DSMC very inefficient for test case
- Greater efficiency achieved by better distribution of particles
- Greatest benefit for flows with U shaped distributions for N



Time correlation of samples

- Correlation of samples characterized by autocorrelation function
- Samples are very correlated not statistically independent
- Identical correlation function observed for n and V

$$\mathrm{Cov}\left(Y^k, Y^{k'+k}\right) \triangleq \mathbb{E}\left[Y^{k+k'} \times Y^k\right] - \mathbb{E}\left[Y^{k+k'}\right] \times \mathbb{E}\left[Y^k\right]$$



$$\rho(k) = \frac{\mathrm{Cov}\left(Y^{k'}, Y^{k'+k}\right)}{\mathrm{Var}\left(Y\right)}$$

$$\mathrm{Var}\left(Y\right) = \mathrm{Cov}\left(Y^{k'}, Y^{k'}\right)$$