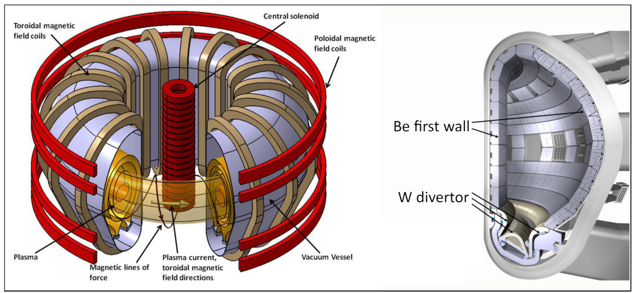
An Adjoint Method for the Nonlinear Boltzmann Equation

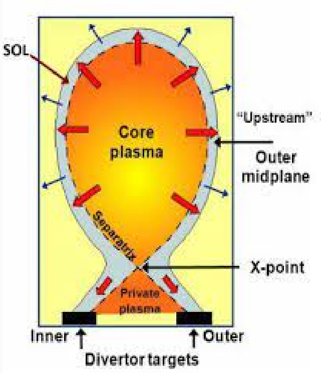
Russ Caflisch[1]

[1]Director, Courant Institute, NYU

This is joint work with Yunan Yang (ETH Zürich) and Denis Silantyev (U Colorado, Colorado Springs).

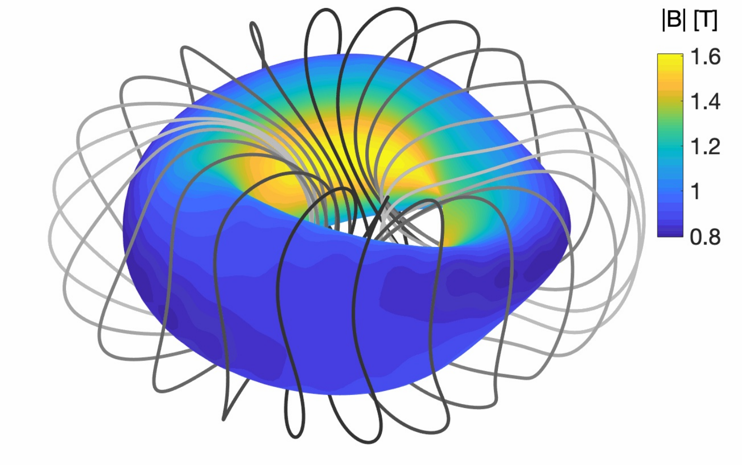
Optimization Example 1: Tokamak Divertor


Toroidal magnetic field coils
Central solenoid
Poloidal magnetic field coils
Plasma
Magnetic lines of force
Plasma current, toroidal magnetic field directions
Vacuum Vessel
Be first wall
W divertor



Optimization of divertor shape based on an adjoint method for MHD equations, W. Dekeyser, et al.
Collisional effects may be significant near the divertor.

Optimization Example 2: Stellarator Electromagnetic Coils



Optimization of coil geometry based on an adjoint method for MHD equations, E. Paul, M. Landreman, et al.
Collisional effects may be significant near the plasma edge.

Main ideas of this talk

- **Obstacle**: Shape design requires optimization over many parameters ξ.
    - Difficult for direct optimization methods.
- **Solution**: A major advantage of the adjoint method is that it can simultaneously optimize over many parameters.

- **Obstacle**: The adjoint of the Boltzmann equation for velocity density f is comparable to the linearized Boltzmann equation.
    - Computationally intractable.
- **Solution**: Discretize Boltzmann using DSMC, then compute adjoint relative to the DSMC velocities v
    - Computationally efficient.

- **Obstacle**: DSMC may require rejection sampling, which is discontinuous in v
    - The adjoint equations require differentiation.
- **Solution**: Differentiate the expectation (of rejection sampling), then sample
    - This leads to a new term: the score function

Boltzmann Equation

Inclusion of collisional effects is through the Boltzmann equation for rarefied gases (or the Landau-Fokker-Plank equation for plasmas)

$$\begin{cases} \frac{\partial f}{\partial t} + v \cdot \nabla_x f = Q\left(f, f\right) \\ f(0, x, v) = f_0 \text{ on } \Omega \\ f = f^{eq}(\rho_b, \boldsymbol{u}_b, T_b) \text{ on } \partial\Omega \end{cases}$$

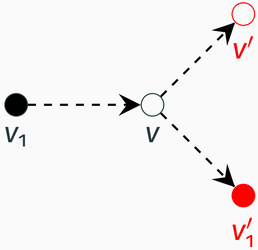
where

$$Q(f, f) = \int_{\mathbb{R}^3} \int_{\mathcal{S}^2} q(v - v_1, \sigma) \left(f(v_1')f(v') - f(v_1)f(v)\right) d\sigma dv_1,$$

v' and v_1' are the velocities after collision between v and v_1.

For simplicity, we will consider the spatially homogeneous Boltzmann equation, without the term $v \cdot \nabla_x f$.

Binary Collisions



● v_1 ○ v pre-collision velocities

● v_1' ○ v' post-collision velocities

Elastic collision preserves mass, momentum and energy. Assume particles have equal mass, then

$$\begin{aligned} v_1 + v &= v_1' + v' \\ |v_1|^2 + |v|^2 &= |v_1'|^2 + |v'|^2 \end{aligned}$$

Elastic Binary Collision Formulas

The σ-representation:

$$\begin{cases} v' & = 1/2(v+v_1) + 1/2|v-v_1|\sigma \\ v'_1 & = 1/2(v+v_1) - 1/2|v-v_1|\sigma \end{cases}, \quad \sigma = \frac{v'-v'_1}{|v'-v'_1|}. \tag{1}$$

$\sigma \in \mathcal{S}^2$ is the vector of collision angles. Also note that $|v-v_1|=|v'-v'_1|$ and define $\alpha = \frac{v-v'}{|v-v'|}$.

Collision Formulas in Matrix Form

We can then write the collision formulas as

$$\begin{pmatrix} v' \\ v_1' \end{pmatrix} = A(\sigma, \alpha) \begin{pmatrix} v \\ v_1 \end{pmatrix}, \quad \begin{pmatrix} v \\ v_1 \end{pmatrix} = B(\sigma, \alpha) \begin{pmatrix} v' \\ v_1' \end{pmatrix}, \quad \text{where}$$

$$A(\sigma, \alpha) = \frac{1}{2} \begin{pmatrix} I + \sigma\alpha^T & I - \sigma\alpha^T \\ I - \sigma\alpha^T & I + \sigma\alpha^T \end{pmatrix},$$

where I is the identity matrix in $\mathbb{R}^3$ and $B = A^\top = A^{-1}$.

The Collision Kernel

Different types of rarefied gas have different collision kernels:

$$q(v - v_1, \sigma) = \tilde{q}(|v - v_1|, \theta), \quad \text{where } \cos\theta = \sigma \cdot \alpha.$$

θ is the scattering angle. For the special case $q = C_\beta(\theta)|v - v_1|^\beta$

- $\beta = 0$: Maxwellian collision model. We also take C_β to be a constant.
- $\beta > 0$: Hard Potential model
- $\beta < 0$: Soft Potential model
- $\beta = -3$: Coulomb Potential model

Optimization Problem for Parameter ξ in Initial Data

$$\begin{cases} \dfrac{\partial f}{\partial t} = Q\left(f, f\right), \\ f(0, v) = f_0(v; \xi). \end{cases}$$

$$Q(f, f) = \iint q(v - v_1, \sigma)(f(v_1')f(v') - f(v_1)f(v))d\sigma dv_1.$$

We aim to find ξ that optimizes an objective function, e.g.,

$$J_1(\xi) = \int r(v)f(v, T)dv.$$

This is a Boltzmann equation **constrained** optimization.

Caflisch, R., Silantyev, D. and Yang, Y., 2021. Adjoint DSMC for nonlinear Boltzmann equation constrained optimization. Journal of Computational Physics, 439, p.110404. `https://arxiv.org/pdf/2009.01363.pdf`.

Optimize then Discretize (OTD)

Lagrangian Multipliers

$$J = \underbrace{\int r(v) f(v,T) dv}_{J_1} + \underbrace{\int \kappa(v)(f(v,0) - f_0(v;\xi))dv}_{J_2}$$
$$+ \underbrace{\int_0^T \int \gamma(v,t)(\partial_t f(v,t) - Q(f,f))dvdt}_{J_3}$$

where $\kappa(v)$ in J_2 is a Lagrange multiplier that enforces the initial condition for any $v \in \mathbb{R}^3$, and $\gamma(v,t)$ in J_3 is a Lagrange multiplier that enforces the Boltzmann equation (Eulerian verision of the forward problem).

First-order necessary condition for optimality (KKT)

$$\frac{\delta J}{\delta f(v,t)} = -\partial_t \gamma - \iint (\gamma_1' + \gamma' - \gamma_1 - \gamma) f(v_1) q \, d\sigma \, dv_1 = 0,$$
$$\frac{\delta J}{\delta f(v,T)} = \gamma(v,T) + r(v) = 0,$$
$$\frac{\delta J}{\delta f(v,0)} = -\gamma(v,0) + \kappa(v) = 0.$$

where $\gamma, \gamma_1, \gamma'$ and γ_1' represent $\gamma(v,t), \gamma(v_1,t), \gamma(v',t)$ and $\gamma(v_1',t)$.

The gradient of the objective function w.r.t. the unknown ξ is:

$$\frac{\partial J}{\partial \xi} = -\int \kappa(v) \partial_\xi f_0(v;\xi) dv$$

First-order necessary condition for optimality (KKT)

The continuous adjoint eqn., (appears in Cercignani book)

$$\boxed{\begin{cases} -\partial_t \gamma & = \iint \left(\gamma(v_1') + \gamma(v') - \gamma(v_1) - \gamma(v)\right) f(v_1) q d\sigma dv_1, \\ \gamma(v, T) & = -r(v). \end{cases}}$$

$$\boxed{\partial_\xi J = -\int \gamma(v, 0) \partial_\xi f_0(v; \xi) dv}$$

We have devised a particle method for solving for γ, but it is not systematically derived and it is very inefficient.

Discretize then Optimize (DTO)

Recap: The Optimization Problem

$$\begin{cases} \frac{\partial f}{\partial t} = Q\left(f, f\right), \\ f(0, v) = f_0(v; \xi). \end{cases}$$

where we assume the Maxwellian collision kernel $q(\sigma) = 1$ does not have angle dependence, and thus

$$Q(f,f) = \iint (f(v_1')f(v') - f(v_1)f(v))d\sigma dv_1.$$

We aim to find ξ that optimizes an objective function, e.g.,

$$J_1(\xi) = \int r(v)f(v,T)dv.$$

Discretization in v and t by DSMC (for Maxwellian gas with q =constant)

1. Initial Data. Sample $v_{0,i}$ (for $i = 1, \dots, N$) from $f_0(v; \alpha)$
2. For $M = T/(\Delta t)$ time steps, at timestep k

   2.1 Perform collisions.

   Select $N_c = q\Delta t N/2$ particle pairs $(v_{k,i}, v_{k,i_1})$ for collisions and set

   $$v_{k+1,i} = v'_{k,i} = C(v_{k,i}\, v_{k,i_1})$$
   $$v_{k+1,i_1} = v'_{k,i_1} = C(v_{k,i_1}\, v_{k,i}).$$

   2.2 Update the particles that do not collide in this time step.

   $$v_{k+1,i} = v_{k,i}$$

3. At the final time $t_M = T$, calculate the objective function $\mathcal{J}_1 = \frac{1}{N}\sum_{i=1}^{N} r(v_{M,i})$.

Optimize $\mathcal{J}_1$ Using the "Discretize Then Optimize" (DTO) Approach

Lagrangian multipliers are applied to enforce the DSMC equations

$$\mathcal{J} = \underbrace{\frac{1}{N}\sum_{i=1}^{N} r(v_{M,i})}_{\mathcal{J}_1} + \underbrace{\frac{1}{N}\sum_{i=1}^{N} {\color{blue}\gamma_{0,i}} \cdot (v_{0,i} - v_{0,i}(\alpha))}_{\mathcal{J}_2}$$

$$+ \underbrace{\frac{1}{N}\sum_{k=1}^{M}\sum_{i=1}^{N} {\color{blue}\gamma_{k,i}} \cdot (v_{k,i} - v'_{k-1,i})}_{\mathcal{J}_3}\,.$$

where ${\color{blue}\gamma_{I,i}}$ in $\mathcal{J}_2$ enforces the *discrete* initial condition for all N particles, and ${\color{blue}\gamma_{k,i}}$ in $\mathcal{J}_3$ enforces the particle collision rule.

$v'_{k-1,i}$ represents the post-collision velocity of $v_{k-1,i}$ if it participates in the collision at the k-th time interval. Otherwise, $v'_{k-1,i} = v_{k-1,i}$.

Solve the Discrete Adjoint Equations: Adjoint DSMC Method

Represent the adjoint variables as

$$\Gamma_k = \{\gamma_1, \ldots, \gamma_i, \ldots, \gamma_N\}(t_k),$$

denoting the *i*-th adjoint variables in Γ_k as $\gamma_{k,i} \in \mathbb{R}^3$.

Solve the Discrete Adjoint Equations: Adjoint DSMC Method

Represent the adjoint variables as

$$\Gamma_k = \{\gamma_1, \dots, \gamma_i, \dots, \gamma_N\}(t_k),$$

denoting the *i*-th adjoint variables in Γ_k as $\gamma_{k,i} \in \mathbb{R}^3$.

Starting with final-time condition $\gamma_F = -\partial_v r(v_F)$,

Solve the Discrete Adjoint Equations: Adjoint DSMC Method

Represent the adjoint variables as

$$\Gamma_k = \{\gamma_1, \ldots, \gamma_i, \ldots, \gamma_N\}(t_k),$$

denoting the *i*-th adjoint variables in Γ_k as $\gamma_{k,i} \in \mathbb{R}^3$.

Starting with final-time condition $\gamma_F = -\partial_v r(v_F)$, solve backwards for $k = M-1, \ldots, 2, 1$

$$\begin{pmatrix} \gamma_{k,i} \\ \gamma_{k,i_1} \end{pmatrix} = D \begin{pmatrix} \gamma_{k+1,i} \\ \gamma_{k+1,i_1} \end{pmatrix},$$

in which

$$D = \frac{\partial(v'_{k,i}, v'_{k,i_1})}{\partial(v_{k,i}, v_{k,i_1})} = \begin{cases} B(\sigma_k, \alpha_k) & \text{for } q = \text{constant} \\ B(\sigma_k, \alpha_k) + \text{additional angular terms} & \text{otherwise.} \end{cases}$$

Computation of the Gradient using the Adjoint Variables

The adjoint variable is the velocity gradient of the original objective function:

$$\gamma_{k,i} = \partial_{v_{k,i}} \mathcal{J}_1$$

in which $\mathcal{J}_1 = \frac{1}{N} \sum_{i=1}^{N} r(v_{F,i})$.

Any gradient of $\mathcal{J}_1$ can now be computed (e.g., for the initial value problem) as

$$\partial_\xi \mathcal{J}_1 = -\frac{1}{N} \sum_{i=1}^{N} \gamma_{0,i} \cdot \partial_\xi v_{0,i}(\xi).$$

This computes the gradient for use in optimization.

Numerical Comparison (Memory, Error, Speed)

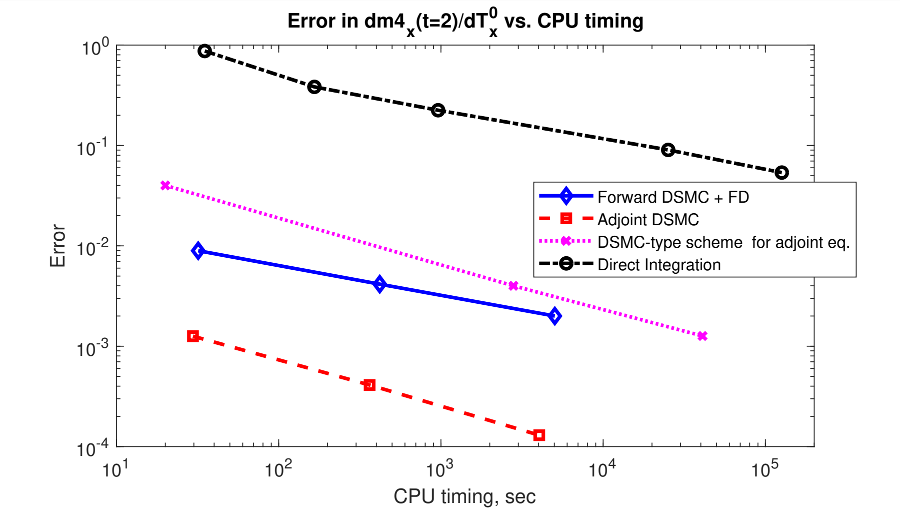
We compute the gradient numerically after the forward DSMC simulations solving the Boltzmann equation ($N = 10^6$):

1. finite difference method; (0.38s *for one parameter**)
2. adjoint DSMC method; (0.22 s)
3. particle method for the continuous adjoint eqn; (280 s)
4. direct discretization of the continuous adjoint eqn. (overnight)**

*Computational costs for #2, #3, and #4 are independent of the size of the unknowns but untrue for #1 — The beauty of the adjoint-state method.

**This is a result of backward Euler scheme in time and Riemann sum for the RHS integral.

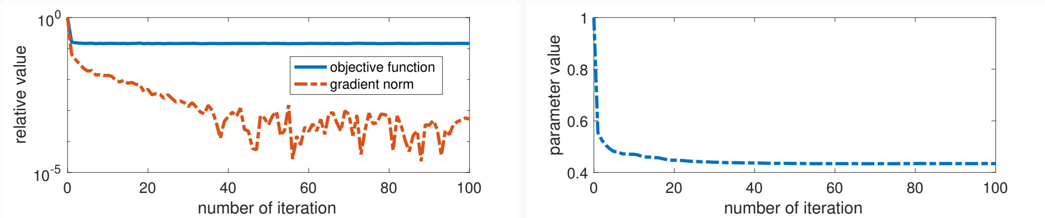
Error vs. CPU Timing in Gradient Computation



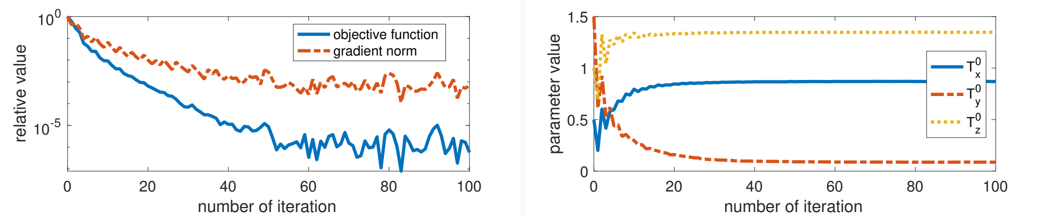
Caflisch, R., Silantyev, D. and Yang, Y., JCP 2021. `https://arxiv.org/pdf/2009.01363.pdf`.

Optimization Examples

Test 1: minimize the difference between 2nd and 4th moments at final time



Test 2: inverse data matching problem



Caflisch, R., Silantyev, D. and Yang, Y., JCP 2021. `https://arxiv.org/pdf/2009.01363.pdf`.

DSMC for General Collision Kernel $q = q(|v - v_1|, \theta)$

- Direct use of DSMC is intractable
  - Collision probability in time step Δt is $q\Delta t$ for each of $N/2$ particle pairs.
  - Direct use of DSMC requires random decision for each pair
    - since $q = q(|v_i - v_{i_1}, \theta)$ is different for each pair
    - Work per time step = $O(N)$
  - Intractable since total work (over $1/\Delta t$ times steps) is $O(N/\Delta t)$!

DSMC for General Collision Kernel $q = q(|v - v_1|, \theta)$

- Direct use of DSMC is intractable
    - Collision probability in time step Δt is $q\Delta t$ for each of $N/2$ particle pairs.
    - Direct use of DSMC requires random decision for each pair
        - since $q = q(|v_i - v_{i_1}, \theta)$ is different for each pair
        - Work per time step = $O(N)$
    - Intractable since total work (over $1/\Delta t$ times steps) is $O(N/\Delta t)$!
- Solution: Apply rejection sampling.
    - Set $\Sigma = \max(q(v_i - v_j, \theta))$= “virtual collision” rate.
    - For each particle pair:
        - No virtual collision with probability $h_1 = 1 - \Sigma\Delta t$.
        - Virtual collision that is a real collision, with probability $h_2 = q\Delta t$
        - Virtual collision that is not real (rejection) with probability $h_3 = (\Sigma - q)\Delta t$
    - No computation is required for particles that do not have virtual collisions.
        - Just select $(h_2 + h_3)N = \Sigma\Delta tN$ particles for virtual collisions
        - Work per time step $= O(\Sigma\Delta tN)$.
    - Tractable since total work is $O(\Sigma N)$.

Adjoint Equations for General Collision Kernel

For DSMC with rejection sampling, the adjoint equations become

$$\begin{pmatrix} \gamma_{k,i} \\ \gamma_{k,i_1} \end{pmatrix} = B(\sigma_k, \alpha_k) \begin{pmatrix} \gamma_{k+1,i} \\ \gamma_{k+1,i_1} \end{pmatrix} + (\partial_{v_k} \log h_j)(r(v_M) + r(\tilde{v}_M)) \begin{pmatrix} 1 \\ 1 \end{pmatrix} .$$

The extra term $\partial_{v_k} \log h_j$ is the “score function”.
It comes from the commutator

$$\partial_v E_r[\cdot] - E_r[\partial_v \cdot] = E_r[(\partial_{v_k} \log h_j) \cdot].$$

E_r is expectation over the rejection-sampling (with probabilities h_1, h_2, h_3).

Derivation of the Score Function

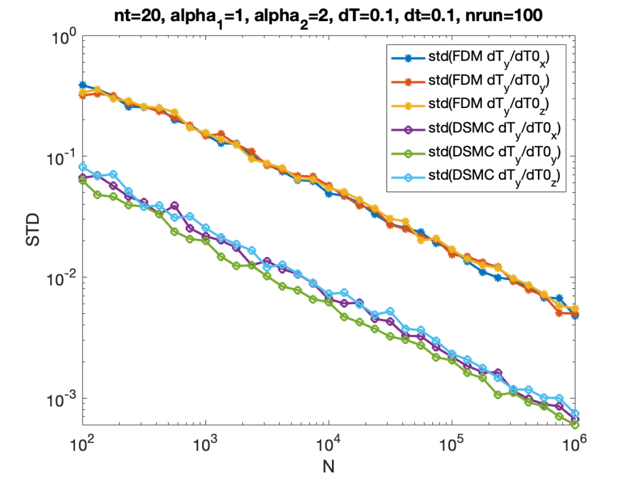
Avoiding the obstacle that rejection sampling is discontinuous:

- Partially discretize
  - Discretize the velocities: N velocities v_i for $i = 1, \ldots, N$
  - Keep expectations, rather than sampling the random variables that determine acceptance or rejection
  - Then differentiate wrt the v_i's, as a step in optimizing
  - Rewrite the derivative as an expectation - this leads to the score function

$$\begin{aligned}\partial_v E_r[\phi] &= \partial_v \Sigma_i h_i \phi = \Sigma_i \left(h_i \partial_v \phi + \phi \partial_v h_i\right) = \Sigma_i h_i \left(\partial_v \phi + h_i^{-1}(\partial_v h_i)\phi\right) \\ &= E_r[\partial_v \phi + (\partial_v log h_i)\phi].\end{aligned}$$

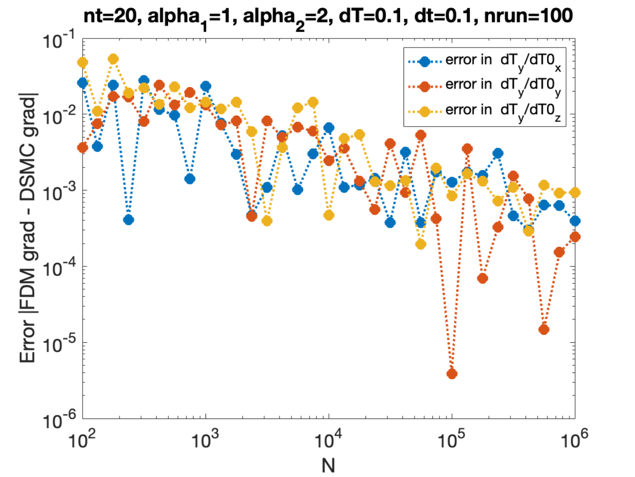
- Then sample the random variables that determine acceptance or rejection, as well as the choice of collision pairs and collision parameters
  - No discontinuities
- This again yields a single realization of DSMC

Accuracy of the Adjoint Method for Nonconstant q



Statistical error

Accuracy of the Adjoint Method for Nonconstant q



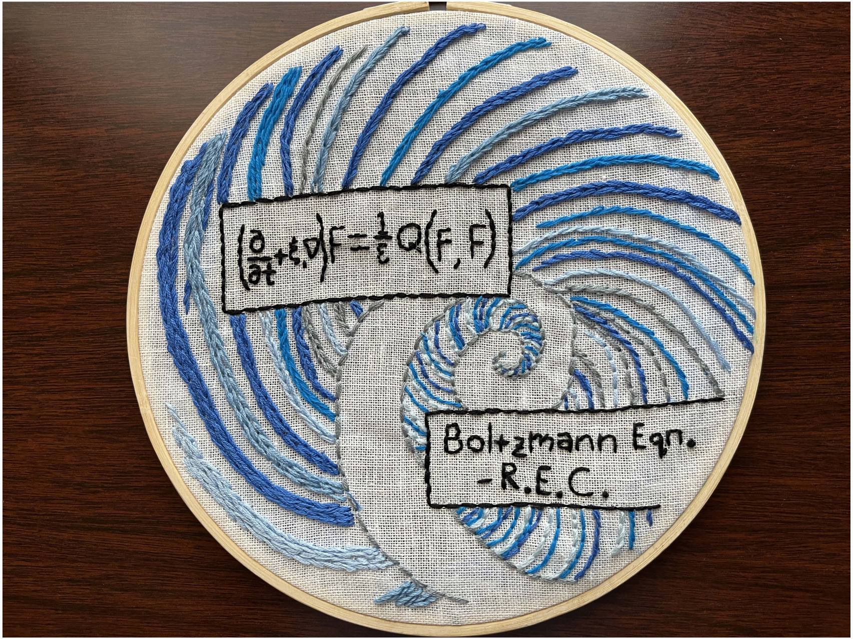
Average error

Conclusions

- **Adjoint for Boltzmann**
  - Discretize (i.e., sample) velocities and use DSMC
  - Formulate augmented Lagrangian with
    - objective function, constraints and adjoint variables γ
  - Differentiate in *v* to get adjoint equations for γ
  - Solve DSMC forward in time; then adjoint equations backward in time,
  - Obtain derivatives of objective function
- **Rejection Sampling**
  - Needed for general collisions
  - Differentiate the expectation (of rejection sampling), then sample
  - Leads to new term – score function – in adjoint equations
- **Computationally efficient and accurate**

Acknowledgement

This work is partially supported by NSF DMS-1913129, and U.S. Department of Energy under Award Number DE-FG02-86ER53223.


$\left(\frac{\partial}{\partial t}+\xi\cdot\nabla\right)F=\frac{1}{\varepsilon}Q(F,F)$
Boltzmann Eqn.
-R.E.C.