


Model Order Reduction of Rarefied Gases Using Neural Networks

Zachary Schellin | Institut für Numerische Fluiddynamik

Outline

Introduction

The BGK-Model

Sod’s shock tube

Model Order Reduction

Proper Orthogonal Decomposotion (POD)

Neural Networks

Results

Discussion

Table of Contents

Table of Contents

Governing equations

- The Boltzmann equation approximated by ***Q*** the BGK operator as a source term with

$$\partial_t f + v\partial_x f = \overbrace{\frac{1}{\tau}(M_f - f)}^{Q} \tag{1}$$

- The equilibrium solution is a Maxwellian distribution $\boldsymbol{M_f}$ with

$$M_f = \frac{\rho(x,t)}{(2\pi RT(x,t))^{\frac{3}{2}}} \exp\left(-\frac{(v - u(x,t))^2}{2RT(x,t)}\right) \tag{2}$$

- The duration to evolve into equilibrium is given by the relaxation time $\boldsymbol{\tau}$ with

$$\tau^{-1} = \frac{\rho(x,t)T^{1-\nu}(x,t)}{Kn} \tag{3}$$

- The rarefaction level is defined over the Knudsen number ***Kn*** with

$$Kn = \frac{\lambda}{l} \tag{4}$$

1

[1] **PhysRev.94.511**.

Knudsen number

– Solution is $f(x, v, t)$ in 1D and $f(x, y, v_x, v_y, t)$ in 2D and $f(x, y, z, v_x, v_y, v_z, t)$ in 3D

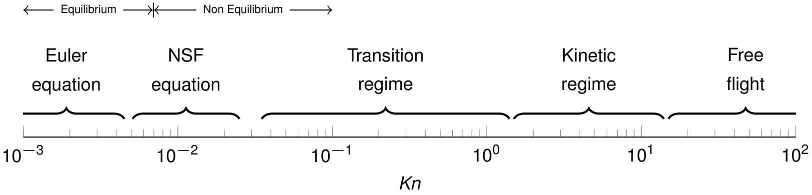


Figure: Partitioning of Kn, the Knudsen number, into levels of rarefaction.

[2]

[2] **NumaKUL**.

Discretization in space and velocity space in 1D

- Space and time discretization considering a uniform grid i.e. $x_j = j\Delta x$ and $j \in \mathbb{Z}$, $v_k = k\Delta v$ and $k \in \mathbb{Z}$, $t^i = i\Delta t$ and $t \in \mathbb{N}$,
- is leading from the full PDE to a set of ODE's in time

$$\partial_t f_{j,k} = -(v_k)_1 D_x f\big|_{j,k}(t) + \frac{1}{\tau}(M_{f\,j,k}(t) - f_{j,k}(t)). \tag{5}$$

- KJ first-order differential equations need to be evaluated when K and J are the number of gridpoints in space and velocity space.
- In 3D there are K^3J^3 first-order differential equations.
- The discretization in velocity space requires the computation of the moments of f.

Moments/ Expected values of f

- Collision invariants $\Phi(v) = [1, v, \frac{1}{2}v^2]$
- The first moment/ Density is

$$\rho(x,t) = \int f \, \mathrm{d}v \,, \tag{6}$$

- the second moment/ Momentum is

$$\rho(x,t)u(x,t) = \int vf \, \mathrm{d}v \,, \tag{7}$$

- the third moment/ kinetic Energy is

$$E(x,t) = \int \frac{1}{2}v^2 f \, \mathrm{d}v \,. \tag{8}$$

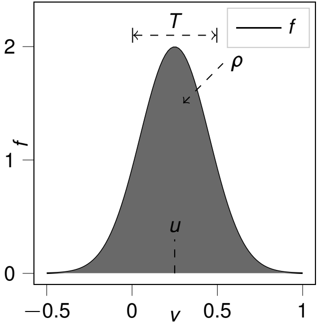


Figure: Illustration of the linkage between the macroscopic quantities of the gas flow and the distribution function f.

Table of Contents

Sod's shock tube

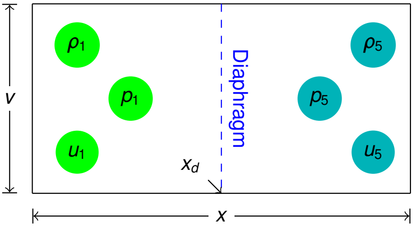


Figure: Problem setup of Sod's shock tube for the BGK model in 1D at $t = 0s$.

- Test case for numerical schemes solving
- **non-linear hyperbolic conservation laws** in gas dynamics (Gary A. Sod in 1978)
- **Idea**:
  - Solve problem analytically (Rankine-Hugoniot jump conditions)
  - Solve problem numerically
  - Compare results expecially **resolution of discontinuities**

Sod's shock tube

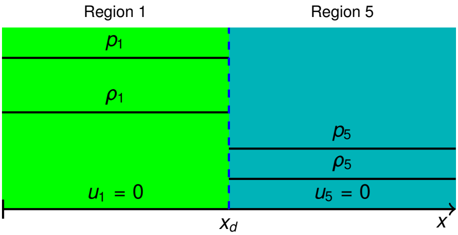


Figure: Problem setup of Sod's shock tube for the BGK model in 1D at $t = 0s$.

- Test case for numerical schemes solving
- **non-linear hyperbolic conservation laws** in gas dynamics (Gary A. Sod in 1978)
- **Idea**:
  - Solve problem analytically (Rankine-Hugoniot jump conditions)
  - Solve problem numerically
  - Compare results expecially **resolution of discontinuities**

Sod's shock tube

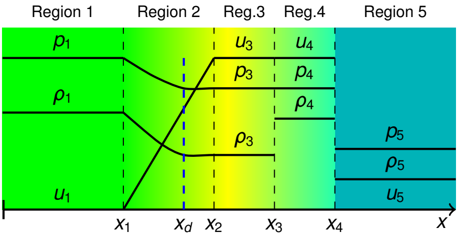


Figure: Problem setup of Sod's shock tube for the BGK model in 1D at $t > 0s$.

- Test case for numerical schemes solving
- **non-linear hyperbolic conservation laws** in gas dynamics (Gary A. Sod in 1978)
- **Idea**:
  - Solve problem analytically (Rankine-Hugoniot jump conditions)
  - Solve problem numerically
  - Compare results expecially **resolution of discontinuities**
    - » $\mathbf{x_1}$ head of rarefaction wave
    - » $\mathbf{x_2}$ tail of rarefaction wave
    - » $\mathbf{x_3}$ contact discontinuity
    - » $\mathbf{x_4}$ position of shockwave

Two Case Studies

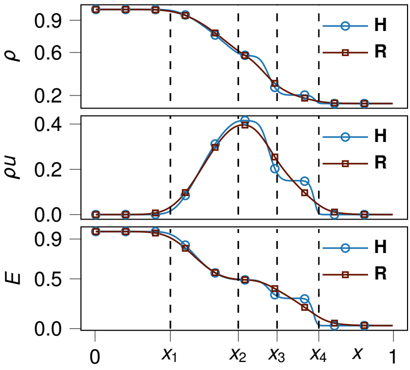


Figure: Moments of **H** and **R** at $t = 0.12s$ and $v = v_0$ in Sod's shock tube.

– Two solutions of the BGK model in Sod's shock tube

– Two levels of rarefaction

- **H**, $Kn = 0.00001$, "Continuum Flow"
- **R**, $Kn = 0.001$, "Slip flow"

– Present discontinuities

H	**R**
• x_1	• x_1
• x_2	• x_2
• x_3	• x_3
• x_4	• x_4

Two Case Studies

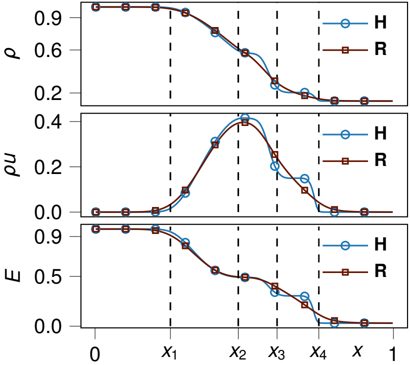


Figure: Moments of **H** and **R** at $t = 0.12s$ and $v = v_0$ in Sod's shock tube.

- Two solutions of the BGK model in Sod's shock tube
- Two levels of rarefaction
  - **H**, $Kn = 0.00001$, "Continuum Flow"
  - **R**, $Kn = 0.001$, "Slip flow"
- Present discontinuities

H	**R**
• x_1	• x_1
• x_2	• x_2
• x_3	• x_3
• x_4	• x_4

Two Case Studies

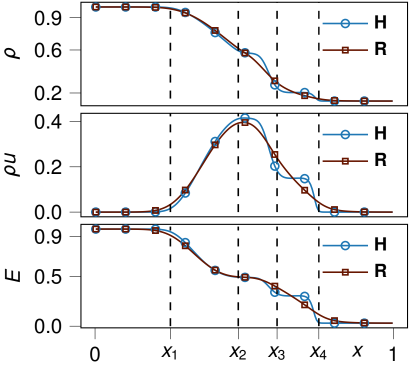


Figure: Moments of **H** and **R** at $t = 0.12s$ and $v = v_0$ in Sod's shock tube.

- Two solutions of the BGK model in Sod's shock tube
- Two levels of rarefaction
  - **H**, $Kn = 0.00001$, "Continuum Flow"
  - **R**, $Kn = 0.001$, "Slip flow"
- Present discontinuities

H	**R**
• x_1	• x_1
• x_2	• x_2
• x_3	• x_3
• x_4	• x_4

Two Case Studies

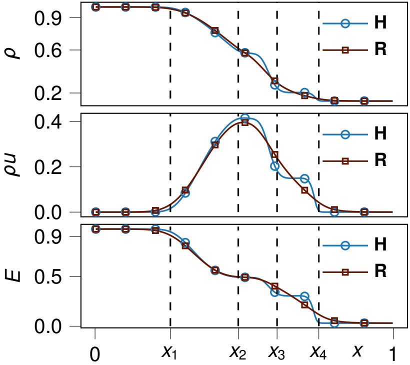


Figure: Moments of **H** and **R** at $t = 0.12s$ and $v = v_0$ in Sod's shock tube.

- Two solutions of the BGK model in Sod's shock tube
- Two levels of rarefaction
  - **H**, $Kn = 0.00001$, "Continuum Flow"
  - **R**, $Kn = 0.001$, "Slip flow"
- Present discontinuities

H	**R**
x_1	x_1
x_2	x_2
x_3	x_3
x_4	x_4

Two Case Studies

- Spatial resolution $J = 200$
- Temporal resolution $I = 25$
- Velocious resolution $K = 40$

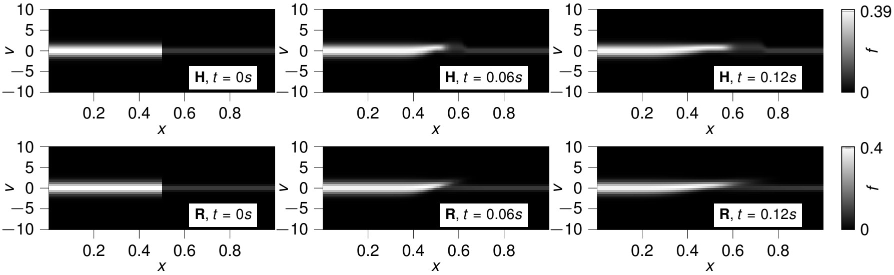
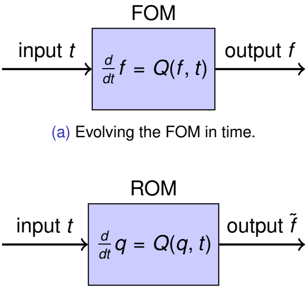


Figure: Solution f for **H** and **R** in x and v.

Table of Contents

Model Order Reduction



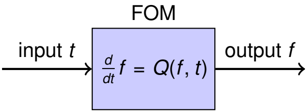
(a) Evolving the FOM in time.

(b) Evolving the ROM in time.

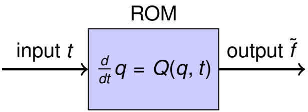
Figure: In the online phase the operator Q is different for the FOM and the ROM.

– **Goal**: Reduce computational cost

Model Order Reduction



(a) Evolving the FOM in time.

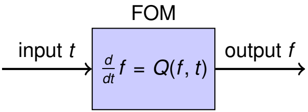


(b) Evolving the ROM in time.

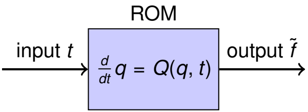
Figure: In the online phase the operator Q is different for the FOM and the ROM.

- **Goal**: Reduce computational cost
  - $f(x, v, t)$ with KJ ODE's in time for 1D

Model Order Reduction



(a) Evolving the FOM in time.

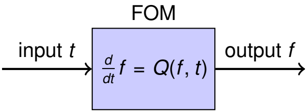


(b) Evolving the ROM in time.

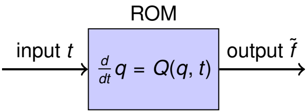
Figure: In the online phase the operator Q is different for the FOM and the ROM.

- **Goal**: Reduce computational cost
  - $f(x, v, t)$ with KJ ODE's in time for 1D
- **Require**: Reduction algorithm

Model Order Reduction



(a) Evolving the FOM in time.

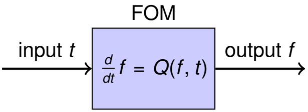


(b) Evolving the ROM in time.

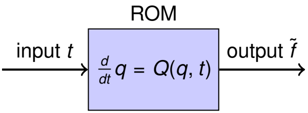
Figure: In the online phase the operator Q is different for the FOM and the ROM.

– **Goal**: Reduce computational cost
  - $f(x, v, t)$ with KJ ODE's in time for 1D
– **Require**: Reduction algorithm
  - Proper Orthogonal Decomposition **(POD)**
  - Neural Networks **(NN)**

Model Order Reduction



(a) Evolving the FOM in time.

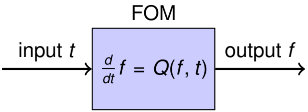


(b) Evolving the ROM in time.

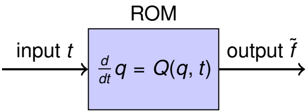
Figure: In the online phase the operator Q is different for the FOM and the ROM.

- **Goal**: Reduce computational cost
  - $f(x, v, t)$ with KJ ODE's in time for 1D
- **Require**: Reduction algorithm
  - Proper Orthogonal Decomposition **(POD)**
  - Neural Networks **(NN)**
- **Require**: Solution of f (only few timesteps)

Model Order Reduction



(a) Evolving the FOM in time.

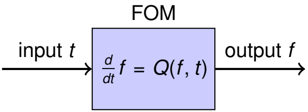


(b) Evolving the ROM in time.

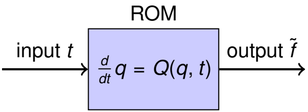
Figure: In the online phase the operator Q is different for the FOM and the ROM.

- **Goal**: Reduce computational cost
  - $f(x, v, t)$ with KJ ODE's in time for 1D
- **Require**: Reduction algorithm
  - Proper Orthogonal Decomposition **(POD)**
  - Neural Networks **(NN)**
- **Require**: Solution of f (only few timesteps)
- **Reduce**: $POD(f(x, v, t)) = q(x, n, t)$

Model Order Reduction



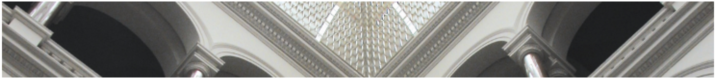
(a) Evolving the FOM in time.



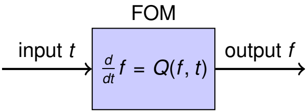
(b) Evolving the ROM in time.

Figure: In the online phase the operator Q is different for the FOM and the ROM.

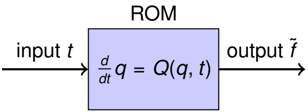
– **Goal**: Reduce computational cost
  - $f(x, v, t)$ with KJ ODE's in time for 1D
– **Require**: Reduction algorithm
  - Proper Orthogonal Decomposition **(POD)**
  - Neural Networks **(NN)**
– **Require**: Solution of f (only few timesteps)
– **Reduce**: $POD(f(x, v, t)) = q(x, n, t)$
  - P is # n with $P << K$



Model Order Reduction



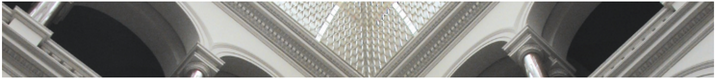
(a) Evolving the FOM in time.



(b) Evolving the ROM in time.

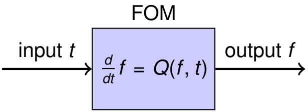
Figure: In the online phase the operator Q is different for the FOM and the ROM.

- **Goal**: Reduce computational cost
  - $f(x, v, t)$ with KJ ODE's in time for 1D
- **Require**: Reduction algorithm
  - Proper Orthogonal Decomposition **(POD)**
  - Neural Networks **(NN)**
- **Require**: Solution of f (only few timesteps)
- **Reduce**: $POD(f(x, v, t)) = q(x, n, t)$
  - P is # n with $P << K$
  - KJ ODE's vs. PJ ODE's

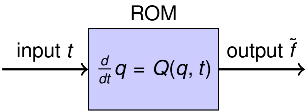


Technische Universität Berlin

Model Order Reduction



(a) Evolving the FOM in time.

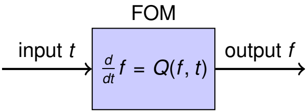


(b) Evolving the ROM in time.

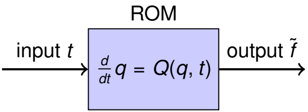
Figure: In the online phase the operator Q is different for the FOM and the ROM.

– **Goal**: Reduce computational cost
  - $f(x, v, t)$ with KJ ODE's in time for 1D
– **Require**: Reduction algorithm
  - Proper Orthogonal Decomposition **(POD)**
  - Neural Networks **(NN)**
– **Require**: Solution of f (only few timesteps)
– **Reduce**: $POD(f(x, v, t)) = q(x, n, t)$
  - P is # n with $P << K$
  - KJ ODE's vs. PJ ODE's
– **Evolve in time** $\rightarrow Q(q, t) = \tilde{f}$

Model Order Reduction



(a) Evolving the FOM in time.



(b) Evolving the ROM in time.

Figure: In the online phase the operator Q is different for the FOM and the ROM.

- **Goal**: Reduce computational cost
  - $f(x, v, t)$ with KJ ODE's in time for 1D
- **Require**: Reduction algorithm
  - Proper Orthogonal Decomposition **(POD)**
  - Neural Networks **(NN)**
- **Require**: Solution of f (only few timesteps)
- **Reduce**: $POD(f(x, v, t)) = q(x, n, t)$
  - P is # n with $P << K$
  - KJ ODE's vs. PJ ODE's
- **Evolve in time** $\rightarrow Q(q, t) = \tilde{f}$
- $f - \tilde{f} < \epsilon$

Table of Contents

Proper Orthogonal Decomposition

– Solution of a PDE is $f(x, v, t)$ can be obtained

Proper Orthogonal Decomposition

- Solution of a PDE is $f(x, v, t)$ can be obtained
  - Discretization into a system of ODE’s

Proper Orthogonal Decomposition

- Solution of a PDE is $f(x, v, t)$ can be obtained
  - Discretization into a system of ODE's
  - Separation of variables ansatz

Proper Orthogonal Decomposition

- Solution of a PDE is $f(x, v, t)$ can be obtained
  - Discretization into a system of ODE's
  - Separation of variables ansatz
    » $f(t, v, x) = \sum_{i=1}^{n} \boxed{a_i(t)} \boxed{\Phi_i(x, v)}$ (9)

Proper Orthogonal Decomposition

- Solution of a PDE is $f(x, v, t)$ can be obtained
  - Discretization into a system of ODE's
  - Separation of variables ansatz
    - » $f(t, v, x) = \sum_{i=1}^{n} \boxed{a_i(t)} \boxed{\Phi_i(x, v)}$ (9)
- How to get $\boxed{\Phi_i}$?

Proper Orthogonal Decomposition

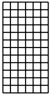
- Solution of a PDE is $f(x, v, t)$ can be obtained
  - Discretization into a system of ODE's
  - Separation of variables ansatz
    - » $f(t, v, x) = \sum_{i=1}^{n} \boxed{a_i(t)} \boxed{\Phi_i(x, v)}$ (9)
- How to get $\boxed{\Phi_i}$?
  - **Preprocessing**: Seperating the spatial and temporal axis of the solution $f(x, v, t)$
    - » $P =$ 

Table of Contents

Table of Contents

Table of Contents

ToDo

- **ToDo** schreiben
- **ToDo** abarbeiten