Thermal Conductivity via Iterative Methods

Jiayi Hu
APC final Project

Linearized Boltzmann Transport Equations

Starting with: $$\frac{df}{dt} = \frac{\partial f}{\partial t}\bigg|_{\text{force}} + \frac{\partial f}{\partial t}\bigg|_{\text{diff}} + \frac{\partial f}{\partial t}\bigg|_{\text{sctt}}$$

Relaxation Time Approximation (RTA) $$-\vec{v}_\lambda \cdot \vec{\nabla} T \frac{\partial f_\lambda^0}{\partial T} = \frac{f_\lambda - f_\lambda^0}{\tau_\lambda^0}$$

Linearized around equilibrium:

$$\Big[-\vec{v}_\lambda \frac{\partial f_\lambda^0}{\partial T} + \frac{1}{N}\sum_{\lambda'\lambda''}\Big\{\Gamma^+_{\lambda\lambda'\lambda''}(\vec{F}_\lambda + \frac{\omega_{\lambda'}}{\omega_\lambda}\vec{F}_{\lambda'} - \frac{\omega_{\lambda''}}{\omega_\lambda}\vec{F}_{\lambda''})$$
$$+ \frac{1}{2}\Gamma^-_{\lambda\lambda'\lambda''}(\vec{F}_\lambda - \frac{\omega_{\lambda'}}{\omega_\lambda}\vec{F}_{\lambda'} - \frac{\omega_{\lambda''}}{\omega_\lambda}\vec{F}_{\lambda''}) + \Gamma_{\lambda\lambda'}\frac{\omega_{\lambda'}}{\omega_\lambda}\vec{F}_{\lambda'}\Big\}\Big] \cdot \vec{\nabla} T = 0$$

$$\boxed{\kappa^{\alpha\beta} = \frac{1}{k_B T N V_{uc}} \sum_\lambda f_\lambda^0 (f_\lambda^0 + 1)(\hbar\omega_\lambda)^2 v_\lambda^\alpha F_\lambda^\beta}$$

Iterative Methods

solving the set of linear equations:

$$d\vec{F}_\lambda = \frac{1}{N}\sum_{\lambda'\lambda''}^{+}\Gamma^{+}_{\lambda\lambda'\lambda''}\left(\frac{\omega_{\lambda'}}{\omega_\lambda}\vec{F}_{\lambda'} - \frac{\omega_{\lambda''}}{\omega_\lambda}\vec{F}_{\lambda''}\right) + \frac{1}{N}\sum_{\lambda'\lambda''}^{-}\frac{1}{2}\Gamma^{-}_{\lambda\lambda'\lambda''}\left(\frac{\omega_{\lambda'}}{\omega_\lambda}\vec{F}_{\lambda'} + \frac{\omega_{\lambda''}}{\omega_\lambda}\vec{F}_{\lambda''}\right) + \frac{1}{N}\sum_{\lambda'}\Gamma_{\lambda\lambda'}\frac{\omega_{\lambda'}}{\omega_\lambda}\vec{F}_{\lambda'}$$

with…

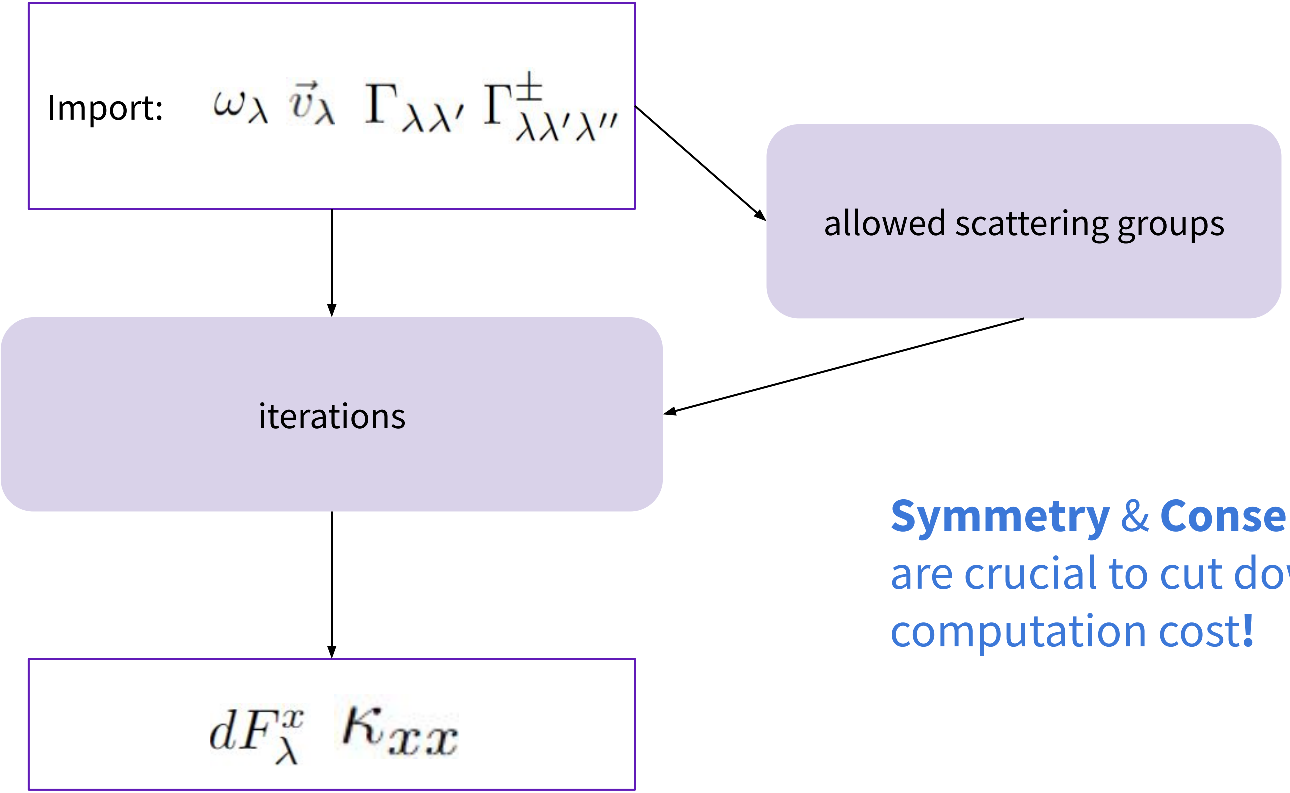
$$\left\{\begin{array}{l}
(\vec{F}_\lambda)_i^{n+1} = (\vec{F})_i^0 + \sum_{j=1, j\neq i}^{N} M_{i,j}\,(d\vec{F}_\lambda)_j^n \\[2ex]
(\vec{F}_\lambda)_i^{n+1} = (\vec{F})_i^0 + \sum_{j=1}^{i-1} M_{i,j}\,(d\vec{F}_\lambda)_j^{n+1} + \sum_{j=i+1}^{N} M_{i,j}\,(d\vec{F}_\lambda)_j^n \\[2ex]
(\vec{F}_\lambda)_i^{n+1} = (\vec{F})_i^0 + \omega\left(\sum_{j=1}^{i-1} M_{i,j}\,(d\vec{F}_\lambda)_j^{n+1} + \sum_{j=i+1}^{N} M_{i,j}\,(d\vec{F}_\lambda)_j^n\right)
\end{array}\right.$$

Jacobian

Gauss-Seidel

SOR

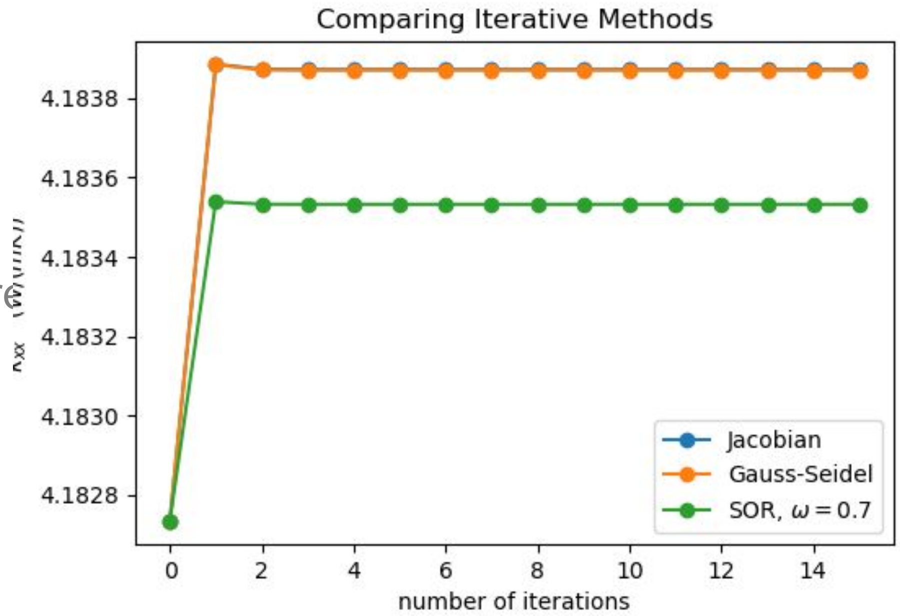
Coding Setup



Symmetry & **Conservation Laws** are crucial to cut down computation cost**!**

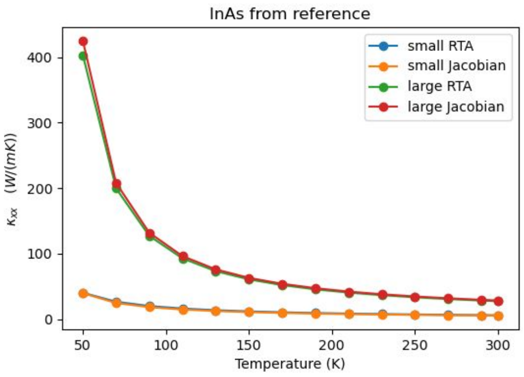
Result

- all methods converge within 2~3 steps
- Jacobian & Gauss-Seidel change more violently
- SOR converge to a different value



InAs on **4*4*4** grid

data here calculated at 50K



ref. data run through external package