The University of Texas at Austin

Solving the Boltzmann equation for electron kinetics using Petrov-Galerkin approach

Milinda Fernando, Daniil Bochkov, Todd Oliver, George Biros, ...

All in hands meeting · December, 2021 · Austin, TX

ODEN INSTITUTE

PECOS Predictive Engineering & Computational Science

https://pecos.oden.utexas.edu

PSAAP III

Boltzmann equation

- Evolution of species distribution function $f = f(\boldsymbol{x}, \boldsymbol{v}, t)$ obeys the **Boltzmann equation**

$$\partial_t f + \boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f - \boldsymbol{E} \cdot \nabla_{\boldsymbol{v}} f = \sum_a C_a(f) + \sum_b C_b(f, f)$$

where, C_a denotes the collision term, for example, in case of $a =$ elastic collisions, with $\delta(0)$ for background distribution function, we can write,

$$C_a(f) = n_0 \int_{S^2} |v| \underbrace{\sigma_a(v, \omega)}_{\text{scat. cross sec.}} \left(|J| f(v^{\text{pre}}(v, \omega)) - f(v)\right) \mathrm{d}\omega$$

- **Challenges**: 6+1 dimensions, numerical issues, HPC, recombination term has additional integration in velocity space.

- Discretization: Petrov-Galerkin in x and v + time-marching (explicit for the time being)

Velocity-space discretization : Petrov-Galerkin approach

- Weak formulation (currently $\mathbf{E} = \mathbf{0}$):

$$\partial_t f = \sum_a C_a(f) \quad \rightarrow \quad \partial_t \int_{R^3} f\phi(\boldsymbol{v})\,\mathrm{d}\boldsymbol{v} = \sum_a \int_{R^3} C_a(f)\phi(\boldsymbol{v})\,\mathrm{d}\boldsymbol{v}, \forall\phi(\boldsymbol{v})$$

- Solution as a perturbed Maxwellian:

$$f(\boldsymbol{v}) = M(v)h(\boldsymbol{v},t), \qquad M(v) = n_e \left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} e^{-\frac{mv^2}{2kT}}$$

- Expansion in basis functions:

$$h(\boldsymbol{v},t) = \sum_{k,l,m} h_{k,l,m}(t)\Phi_k(v)\underbrace{Y_{lm}(v_\theta,v_\phi)}_{\text{sph. harm.}}, \qquad \phi(\boldsymbol{v}) = \underbrace{\Phi_p(v)}_{\text{Maxwell polynomials,BSplines}} \underbrace{Y_{qs}(v_\theta,v_\phi)}_{\text{sph. harm.}}$$

- Resulting system of ODEs:

$$\sum_{k,l,m} M_{p,q,s}^{k,l,m}\partial_t h_{k,l,m}(t) = \sum_{k,l,m} L_{p,q,s}^{k,l,m} h_{k,l,m}(t)$$

Collision operator (weak form)

$$\int_{R^3} C\phi(\boldsymbol{v}_e)d\boldsymbol{v}_e = n_0 \int_{R^3}\int_{S^2} \sigma_a(v,\omega)v f_e(\boldsymbol{v}_e)\left(\phi\left(\boldsymbol{v}_e^{\text{post}}(\boldsymbol{v}_e,\boldsymbol{\omega})\right) - \phi(\boldsymbol{v}_e)\right) d\boldsymbol{\omega} d\boldsymbol{v}_e$$

$$L_{k,l,m}^{p,q,s} = n_0 \int_{v_r}\int_{S^2}\int_{S^2} v^2 M(v_r) P_k\left(\frac{v_r}{v_{\text{th}}}\right) Y^{lm}(v_\theta, v_\phi) v_r \sigma(|v_r|,\chi) \times \left(P_p\left(\frac{v_r'}{v_{\text{th}}}\right) Y_{qs}(v_\theta', v_\phi') - P_p\left(\frac{v_r}{v_{\text{th}}}\right) Y_{qs}(v_\theta, v_\phi)\right) d\omega d\omega_v dv$$

where $\boldsymbol{v}' = v^{post}(\boldsymbol{v},\omega)$.

- Need to evaluate 5d integral, for each matrix element.
- Can be evaluated using precomputed tensors (need recomputation when quadrature grid changes)

$$L_{k,l,m}^{p,q,s} = P_k^{r\theta\phi} Y_{lm}^{r\theta\phi}\left(M^{r\theta\phi\chi\gamma}\left(P_p^{r\theta\phi\chi\gamma}(v_r')Y_{qs}^{r\theta\phi\chi\gamma}(v_\theta', v_\phi') - P_p^{r\theta\phi\chi\gamma}(v_r)Y_{qs}^{r\theta\phi\chi\gamma}(v_\theta, v_\phi)\right) W_\chi W_\gamma\right) W_\phi W_\theta W_r$$

Problem setup

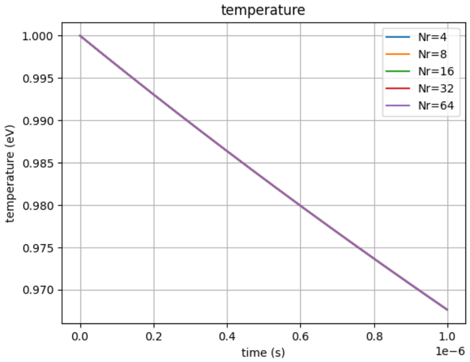
- Set $\boldsymbol{E} = 0$
- Use `LXCAT` cross-section data
- Experiment with Maxwell vs. BSpline polynomials in radial direction.
- Initial condition $f(v, t = 0) = M(v)$ with $T_e(0) = 1ev$
- Collisions considered,
  - G0 : $e + Ar \rightarrow e + Ar$
  - G2 : $e + Ar \rightarrow e + Ar^{+} + e$ with G0
- Use RK45 with adaptive time step size control with specified tolerance, $\mathcal{O}(\Delta t^4)$
- Convergence of $f(v) = M(v)h(v)$, $\rightarrow \|f_m - f_n\| \leq \|M(v)\| \, \|h_m - h_n\|$
- Temperature $T \sim \int_{R^3} v^2 f(v) dv$
- EEDF, for given energy ϵ, $\int_{S^2} \frac{1}{n_e} f(\sqrt{\frac{2\epsilon}{m}}, \theta, \phi) d\omega$

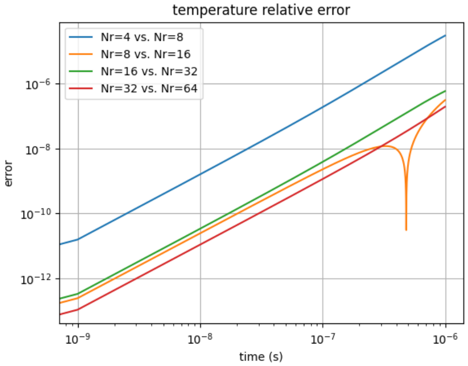
G0 : $e + Ar \rightarrow e + Ar$ (Maxwell polynomials)

- Uses `LXCAT` cross-section data.

G0 : $e + Ar \rightarrow e + Ar$ (Maxwell polynomials)

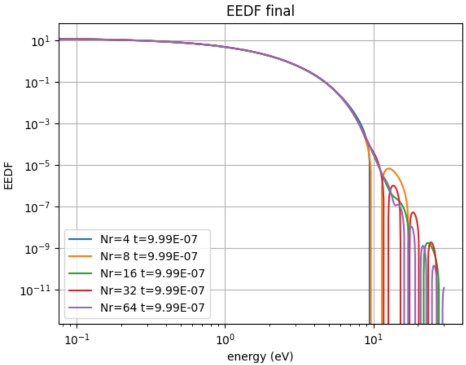
- Uses `LXCAT` cross-section data.

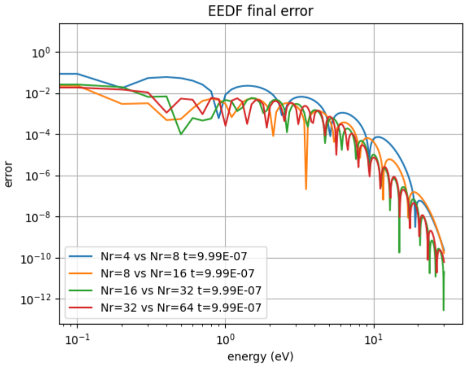




G0 : $e + Ar \rightarrow e + Ar$ (Maxwell polynomials)

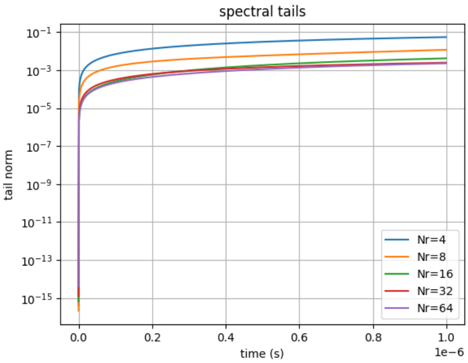
- Uses `LXCAT` cross-section data.





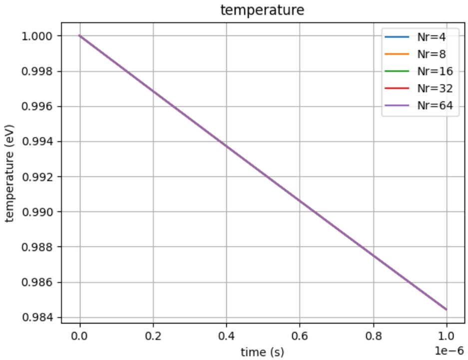
Spectral convergence (Maxwell polynomials)

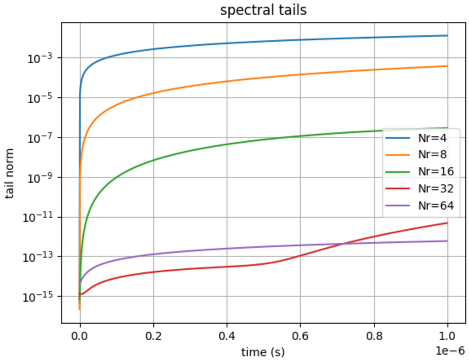
- Convergence of $f(v) = M(v)h(v)$, $\rightarrow$ $\|f_m - f_n\| \leq \|M(v)\| \, \|h_m - h_n\|$
- Tails of the correction term should decay with increasing N_r polynomials.
- Reason ? We don't know yet.
  - May be we need more N_r (Computation of Maxwell poly becomes unstable)
  - Experimental cross-section data is not smooth.
  - Quadrature issues.
- Quantities of interests temperature, mass, seem to converge nicely.



G0 convergence with constant cross-section

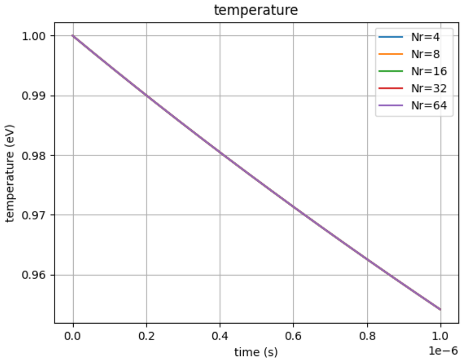
- $\sigma(x) = 2 \times 10^{-20}\mathrm{m}^2$

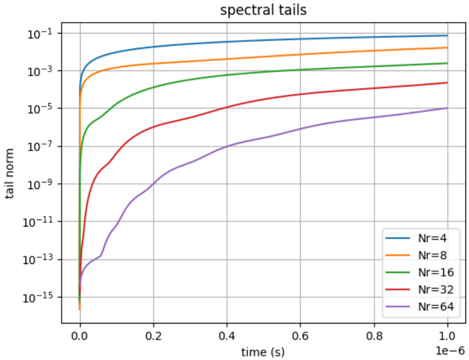




G0 convergence with linear cross-section

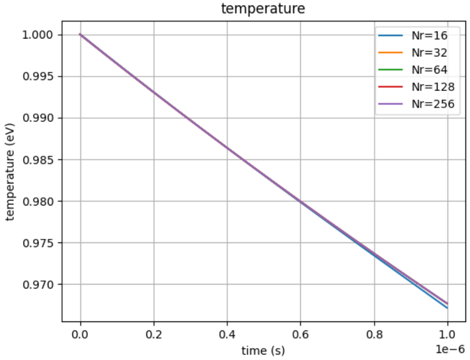
- $\sigma(x) = 2.116 \times 10^{-20} x + 1.24 \times 10^{-21}$

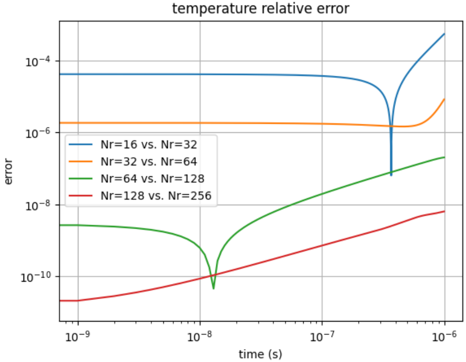




G0 : $e + Ar \rightarrow e + Ar$ (BSplines)

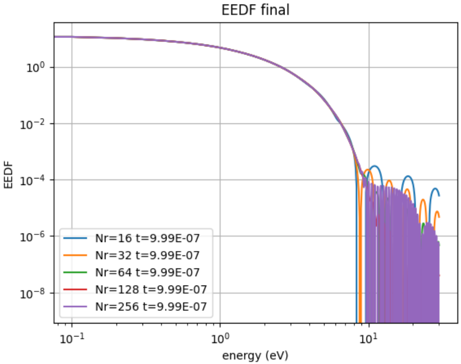
- Uses `LXCAT` cross-section data.

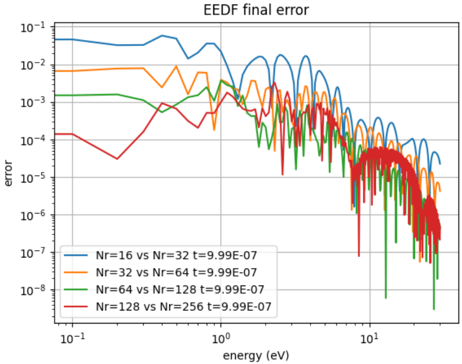




G0 : $e + Ar \rightarrow e + Ar$ (BSplines)

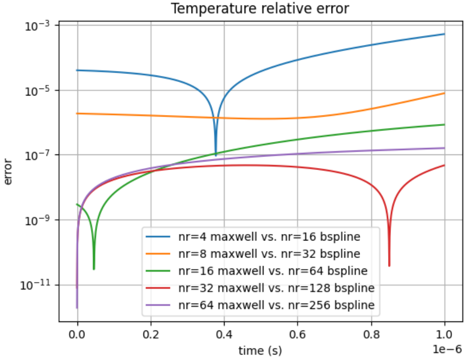
- Uses LXCAT cross-section data.

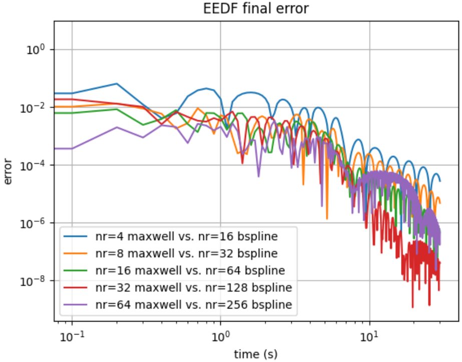




Maxwell Vs. Bsplines : Elastic collisions (G0)

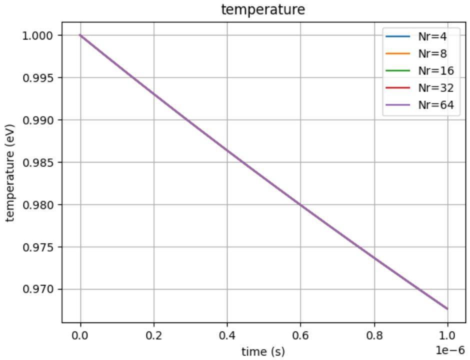
- Uses `LXCAT` cross-section data.

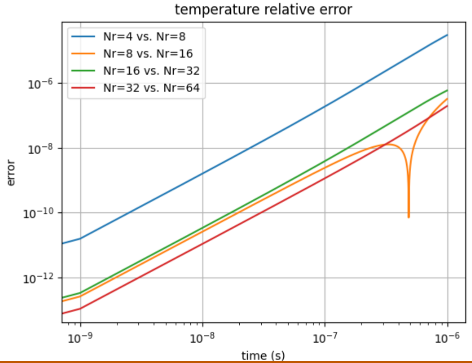




Elastic with ionization (G0+G2)(Maxwell + LXCAT cross-section)

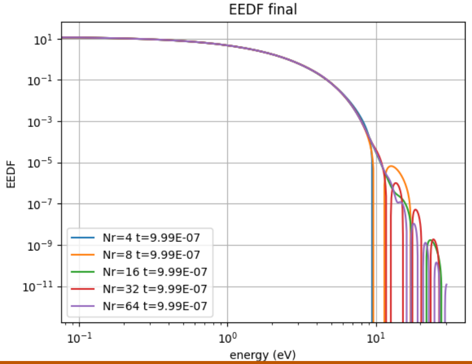
- Quasi-neutrality, $n_i = n_e$, $\partial_t f = n_0 C_0(f) + n_i C_2(f)$

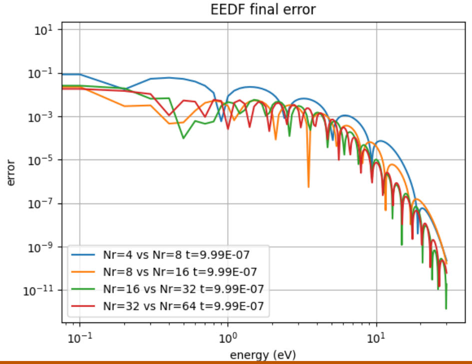




Elastic with ionization (G0+G2)(Maxwell + LXCAT cross-section)

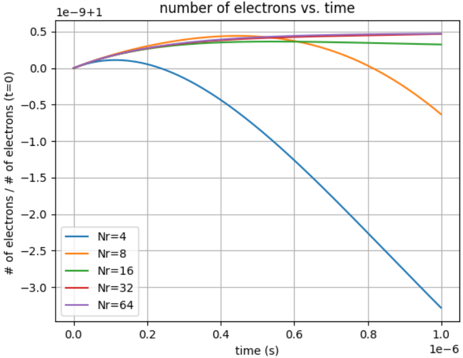
- Quasi-neutrality, $n_i = n_e$, $\partial_t f = n_0 C_0(f) + n_i C_2(f)$





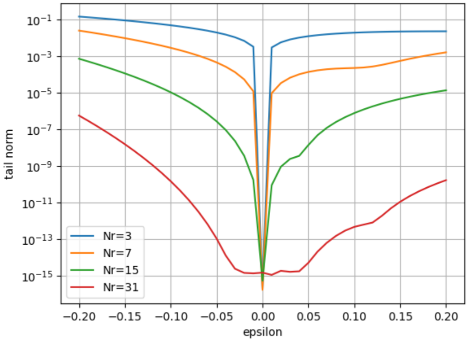
Elastic with ionization (G0+G2)(Maxwell + LXCAT cross-section)

- Quasi-neutrality, $n_i = n_e$, $\partial_t f = n_0 C_0(f) + n_i C_2(f)$



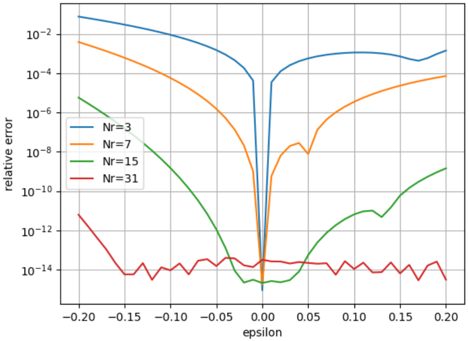
Projection operator

- All experiments are performed with a fixed Maxwellian, for specified temperature T.
- Expansion of a $f_{T'}$ for a given temperature $T \neq T' \rightarrow$ Large N_r.
- One approach : Re-project and recompute the collision operators, when temperature changes.
  - computationally expensive.
  - introduce noise in the tails of the correction term.
- $M_{ij} = \frac{n}{\sqrt{\pi^3}} \int_{R^3} \exp(-v_\alpha^2) P_i(v_\alpha(1+\frac{\epsilon}{\beta})) P_j(v_\alpha) \frac{dv}{\alpha}$



Projection operator

- All experiments are performed with a fixed Maxwellian, for specified temperature T.
- Expansion of a $f_{T'}$ for a given temperature $T \neq T' \rightarrow$ Large N_r.
- One approach : Re-project and recompute the collision operators, when temperature changes.
  - computationally expensive.
  - introduce noise in the tails of the correction term.
- $M_{ij} = \frac{n}{\sqrt{\pi^3}} \int_{R^3} \exp(-v_\alpha^2) P_i(v_\alpha(1+\frac{\epsilon}{\beta})) P_j(v_\alpha) \frac{dv}{\alpha}$



Current progress

- Understand the Boltzmann collision operator with Jacobian for the collisions and how that relate to the weak form.
- Derivation and numerically computation of Maxwell polynomials, associated quadrature.
- `Python` based code for solving collission operator.
- Integration with `LXCAT` cross-section data, for elastic, ionization and excitation reactions.
- Supports Maxwell and BSplines for radial direction, other radial polynomials can be added if needed.
- Tensoized version of the collision operator, (enables efficient GPU computations i.e., `numpy` $\rightarrow$ `cupy`)
- Re-projection with efficient recomputation of the collision operator.

Conclusions and future work

- Maxwell polynomials with experimental cross-section data, low convergence rate.
- Dealing with smoothness issues, discontinuities in experimental cross section data.
- Develop error metrics, when to perform re-projections, (only needed when operate in large temperature range.) , and numerical issues associated.
- Move on to single GPU implementation.

Conclusions and future work

- Maxwell polynomials with experimental cross-section data, low convergence rate.
- Dealing with smoothness issues, discontinuities in experimental cross section data.
- Develop error metrics, when to perform re-projections, (only needed when operate in large temperature range.) , and numerical issues associated.
- Move on to single GPU implementation.

Questions ?

Thank You.

Computing collision operator

- $S^{r\theta\phi\chi\gamma}_{r'\theta'\phi'}$: Scattering velocity tensor, for each $v = (r, \theta, \phi)$ and scattering solid angle (χ, γ) computes (r', θ', ϕ') scattered or newly created particle velocity
- $P^{r\theta\phi\chi\gamma}_{p}$ - radial polynomial evaluated at differed velocity for given incident particle $(r, \theta, \phi, \chi, \gamma)$
- $Y^{r\theta\phi\chi\gamma}_{qs}$ - qs spherical harmonic mode evaluated differed particle direction for a given incident particle $(r, \theta, \phi, \chi, \gamma)$
- $M^{r\theta\phi\chi\gamma}$ - Maxwellian times v_r evaluated for the differed particle for a given incident particle $(r, \theta, \phi, \chi, \gamma)$
- $\sigma^{r\theta\phi\chi\gamma}$ - differential cross section broadcasted on scattering cross section angles.
- $P^{r\theta\phi}_{k}$ - radial polynomials evaluated at radial quadrature points.
- $Y^{r\theta\phi}_{lm}$ - spherical harmonics evaluated angular quadrature points.

Re-computations are only needed if the quadrature grid is changed. Then we can write,

$$L^{p,q,s}_{k,l,m} = P^{r\theta\phi}_{k} Y^{r\theta\phi}_{lm} \left(M^{r\theta\phi\chi\gamma} \left(P^{r\theta\phi\chi\gamma}_{p}(v'_r) Y^{r\theta\phi\chi\gamma}_{qs}(v'_\theta, v'_\phi) - P^{r\theta\phi\chi\gamma}_{p}(v_r) Y^{r\theta\phi\chi\gamma}_{qs}(v_\theta, v_\phi) \right) W_\chi W_\gamma \right) W_\phi W_\theta W_r$$