ÖAW AUSTRIAN ACADEMY OF SCIENCES

IWF SPACE RESEARCH INSTITUTE

This PDF file

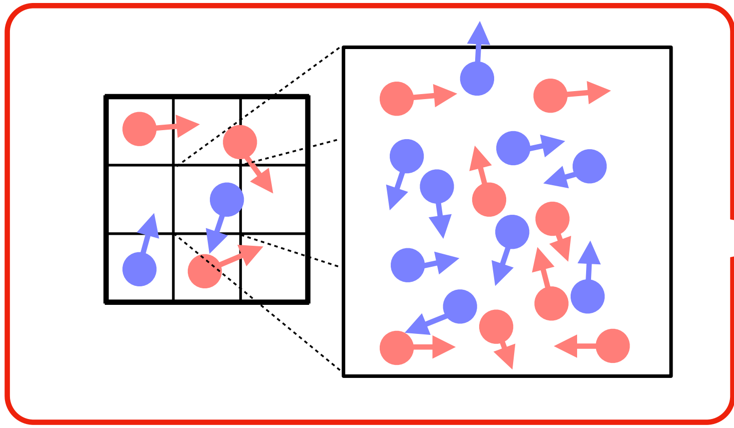
Monte Carlo Generation of Kappa Distributions in Kinetic Plasma Simulations

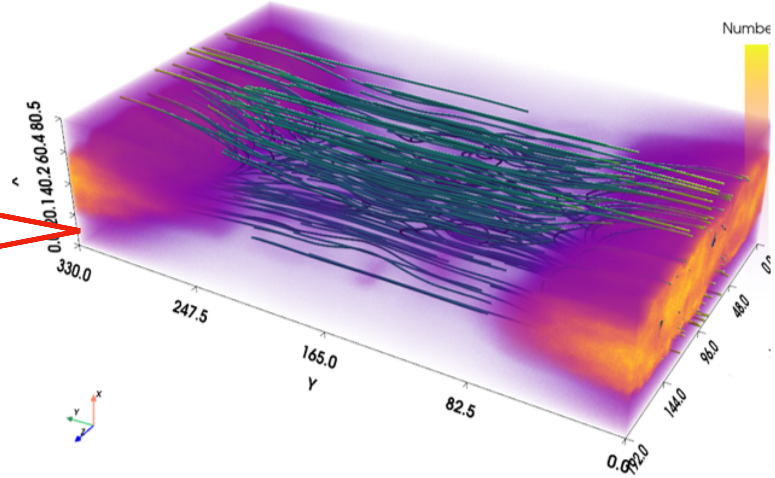
Seiji ZENITANI

S. Usami, S. Matsukiyo, T. Umeda

- 1. Standard t-generator
- 2. Pareto method
- 3. Approximate generator
- 4. Subtracted Kappa generator

Particle-in-cell (PIC) simulation





1. Particle loader

2. Particle integrator

$$\boldsymbol{x}_n \rightarrow \boldsymbol{x}_{n+1}$$
$$\boldsymbol{v}_{n+\frac{1}{2}} \rightarrow \boldsymbol{v}_{n+\frac{3}{2}}$$

3. Charge deposition

$$x_n \rightarrow \overline{\rho}_{c,n}$$
$$\boldsymbol{v}_{n+\frac{1}{2}} \rightarrow \overline{\boldsymbol{j}}_{n+\frac{1}{2}}$$

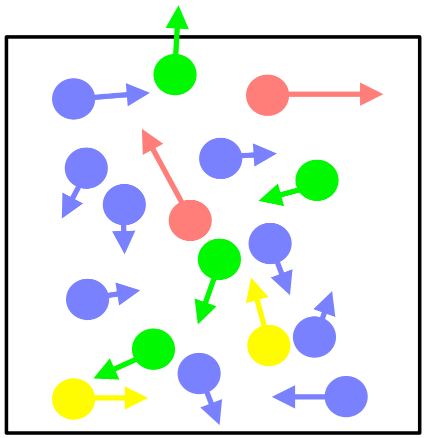
4. Field solver

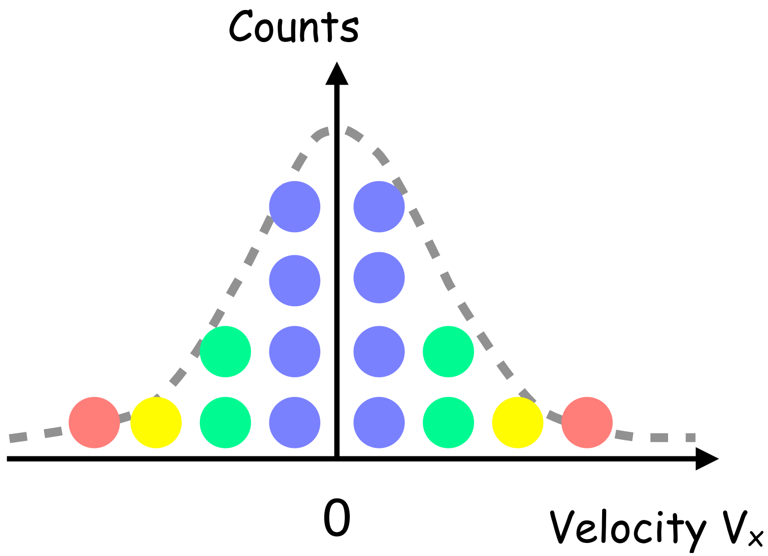
$$\boldsymbol{E}_n \rightarrow \boldsymbol{E}_{n+1}$$
$$\boldsymbol{B}_n \rightarrow \boldsymbol{B}_{n+1}$$

5. Field interpolation

$$\boldsymbol{E}'_n$$
$$\boldsymbol{B}'_n$$

Velocity distribution in PIC simulations

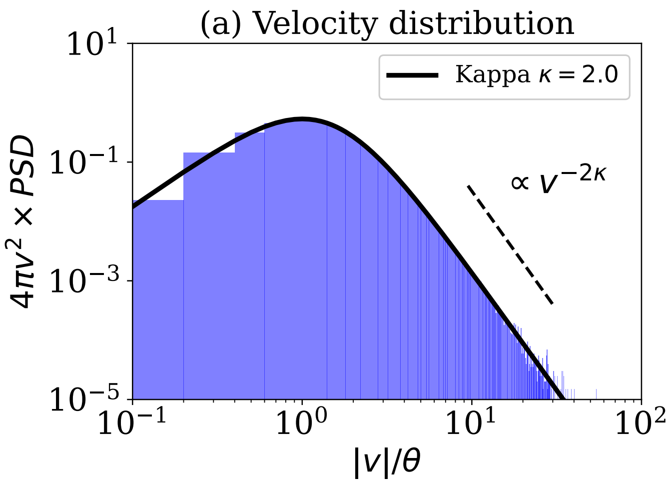




- Maxwell = Normal distribution
  - randn() routine, Box-Muller (1958) method

How to generate Kappa-type (Kappa, kappa-loss-cone, relativistic...) distributions in PIC simulation?

Kappa distribution



- Popular form

$$f(\boldsymbol{v})d^3v = \frac{N_\kappa}{(\pi\kappa\theta^2)^{3/2}}\frac{\Gamma(\kappa+1)}{\Gamma(\kappa-1/2)}\left(1+\frac{\boldsymbol{v}^2}{\kappa\theta^2}\right)^{-(\kappa+1)}d^3v$$

  - Maxwellian-like core + power-law tail

- Random number generation methods
  - Standard method (t-generator)
  - Pareto method
  - Bailey method
  - Galton board method
  - Approximate Kappa generator for GPUs

Rejection sampling (not GPU-friendly)

1D/2D only

Standard method: t-generator

Kappa $$f(\boldsymbol{v})d^3v = \frac{N}{(\pi\kappa\theta^2)^{3/2}}\frac{\Gamma(\kappa+1)}{\Gamma(\kappa-1/2)}\left(1+\frac{v^2}{\kappa\theta^2}\right)^{-(\kappa+1)}d^3v$$

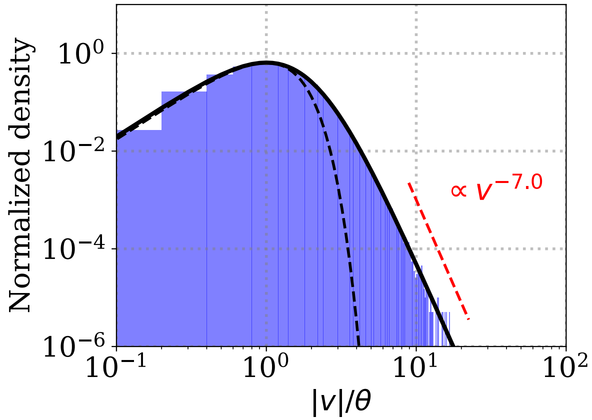
$$\longleftrightarrow \quad \frac{N}{\nu^{p/2}\pi^{p/2}\sigma^p}\frac{\Gamma[(\nu+p)/2]}{\Gamma(\nu/2)}\left(1+\frac{1}{\nu}\frac{\boldsymbol{t}^2}{\sigma^2}\right)^{-\frac{(\nu+p)}{2}}$$ multivariate t-distribution

- Numerical procedure

$$n_1, n_2, n_3 \sim \mathcal{N}(0,1)$$ Normal random number

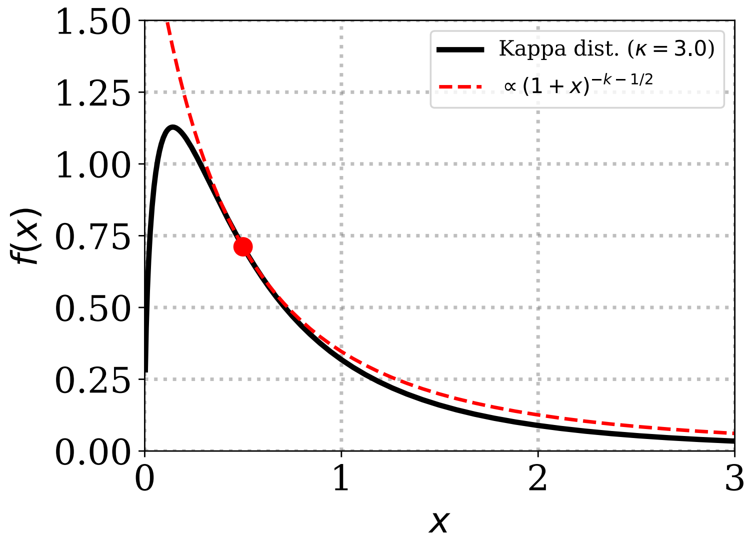
$$x \sim \mathrm{Ga}(\kappa-1/2, 2)$$ Gamma random number

$$v_x \leftarrow \sqrt{\kappa\theta^2}\,\frac{n_1}{\sqrt{x}}$$ Division by gamma



See also Abdul & Mace 2015, SZ & Nakano 2022

Pareto method: Simple & portable



$\mathcal{D} \leftarrow \sqrt{(\kappa-1)^{\kappa-1}/\kappa^{\kappa}}$

repeat

  generate $U_1, U_2 \sim U(0,1)$

  $W \leftarrow \sqrt{(1-U_1)^{-2/\kappa} - 1}$

until $W(1-U_1) > \mathcal{D}U_2$

generate $U_3, U_4 \sim U(0,1)$

$V \leftarrow \sqrt{\kappa\theta^2}W$

$v_x \leftarrow V(2U_3 - 1)$

$v_y \leftarrow 2V\sqrt{U_3(1-U_3)}\cos(2\pi U_4)$

$v_z \leftarrow 2V\sqrt{U_3(1-U_3)}\sin(2\pi U_4)$

return v_x, v_y, v_z

- Kappa, rescaled by energy

$$f(x)\,dx = \frac{1}{\mathrm{B}\left(\frac{3}{2}, \kappa - \frac{1}{2}\right)} x^{1/2}(1+x)^{-\kappa-1}\,dx$$

- Type II Pareto distribution --> Rejection sampling (73-80%)

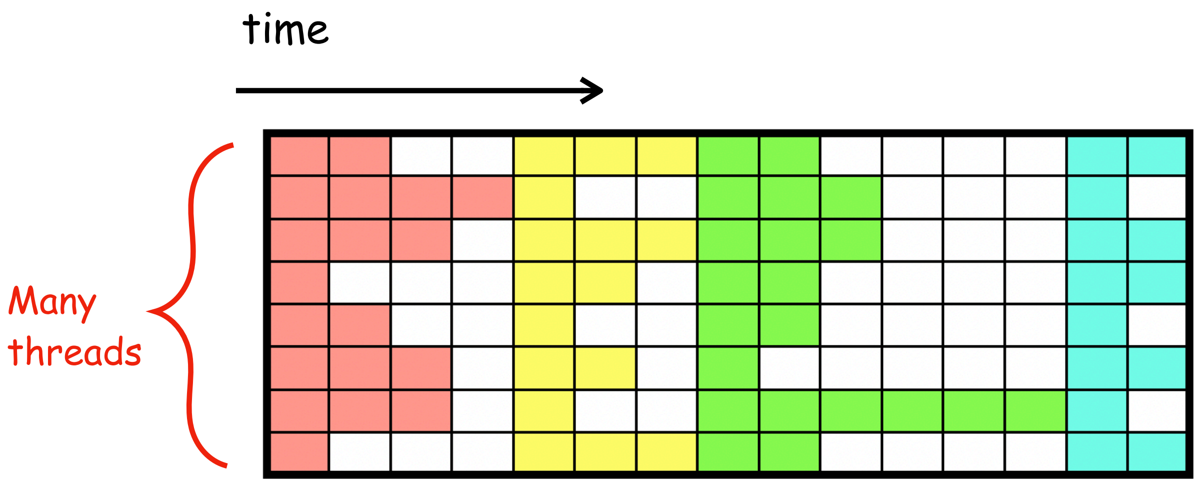
$$g(x) \equiv n(1+x)^{-(n+1)}$$

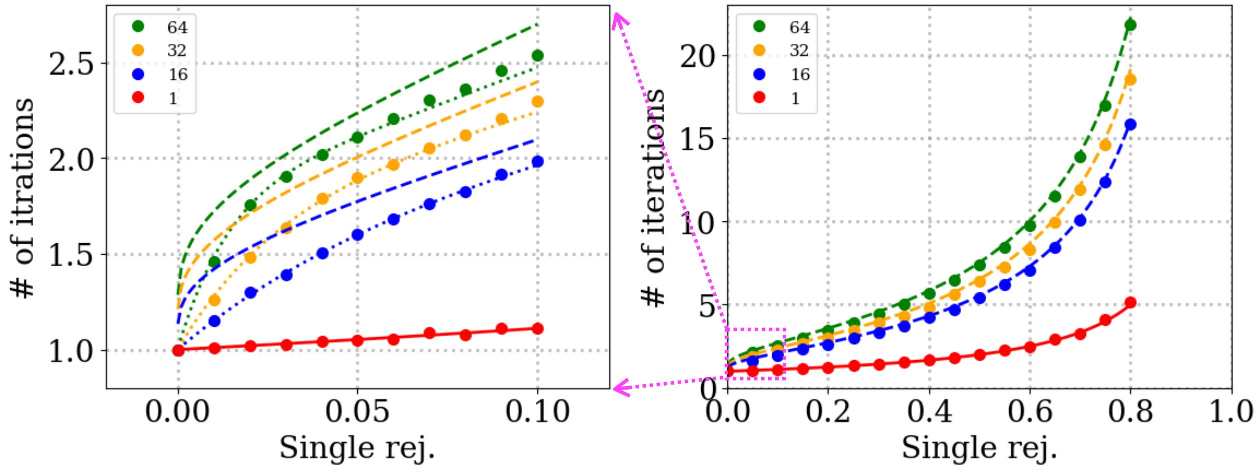
- No need for a Gamma random number

SZ 2025 RNAAS

A problem of a GPU execution model

- A single instruction controls 32 or 64 threads simultaneously (SIMT model)
- Significant slowdown for independent loops





Ridley & Forget 2021
Chakraborty & Gupta 2015

Approximate Kappa generator (1/4)

- Kappa distribution

$$f(\boldsymbol{v})d^3v = \frac{N_\kappa}{(\pi\kappa\theta^2)^{3/2}} \frac{\Gamma(\kappa+1)}{\Gamma(\kappa-1/2)} \left(1 + \frac{\boldsymbol{v}^2}{\kappa\theta^2}\right)^{-(\kappa+1)} d^3v$$

- Cumulative distribution function (CDF)

$$F(x) = I_{\frac{x}{x+\kappa}}\left(\frac{3}{2}, \kappa - \frac{1}{2}\right)$$

$I_x()$: Regularized incomplete beta function

$$\approx G(x) \equiv \left\{1 - \exp_{q*}\left(-\frac{ax+bx^2}{1+cx}\right)\right\}^{3/2}$$

- q-exponential function

$$\exp_q(x) \equiv \Big(1 + (1-q)x\Big)^{\frac{1}{1-q}} \quad \Longleftrightarrow \quad \ln_q(x) \equiv \frac{x^{1-q}-1}{1-q}$$

Approximate Kappa generator (2/4)

- Kappa distribution

$$f(\boldsymbol{v})d^3v = \frac{N_\kappa}{(\pi\kappa\theta^2)^{3/2}}\frac{\Gamma(\kappa+1)}{\Gamma(\kappa-1/2)}\left(1+\frac{\boldsymbol{v}^2}{\kappa\theta^2}\right)^{-(\kappa+1)}d^3v$$

- Approx. Kappa distribution

$$g(\boldsymbol{v})d^3v = \frac{3N_\kappa(a\theta^4+2bv^2\theta^2+bcv^4)}{4\pi v\theta^2\left(\theta^2+cv^2\right)^2} \times \sqrt{1-\left(1+\frac{(a\theta^2+bv^2)v^2}{\kappa^*(\theta^2+cv^2)\theta^2}\right)^{-\kappa^*}} \times \left(1+\frac{(a\theta^2+bv^2)v^2}{\kappa^*(\theta^2+cv^2)\theta^2}\right)^{-(\kappa^*+1)} d^3v$$

Inverse-transform procedure
(No iteration loop)

$\kappa^* \leftarrow \kappa - 1/2$

$a \leftarrow \frac{1}{\kappa}\left(\frac{2}{3B(3/2,\kappa^*)}\right)^{2/3}$

$c \leftarrow \dfrac{0.123\kappa^2 - 1.12\,\kappa + 2.56}{\kappa^2 - 7.89\,\kappa + 15.6}$

$b \leftarrow \left(\kappa^* \frac{3}{2} B(3/2,\kappa^*)\right)^{1/\kappa^*} \frac{\kappa^*}{\kappa}\, c$

generate $U_1, U_2, U_3 \sim \mathrm{Uniform}(0,1)$

$L \leftarrow -\kappa^* \left\{\left(1-U_1^{2/3}\right)^{-1/\kappa^*} - 1\right\}$

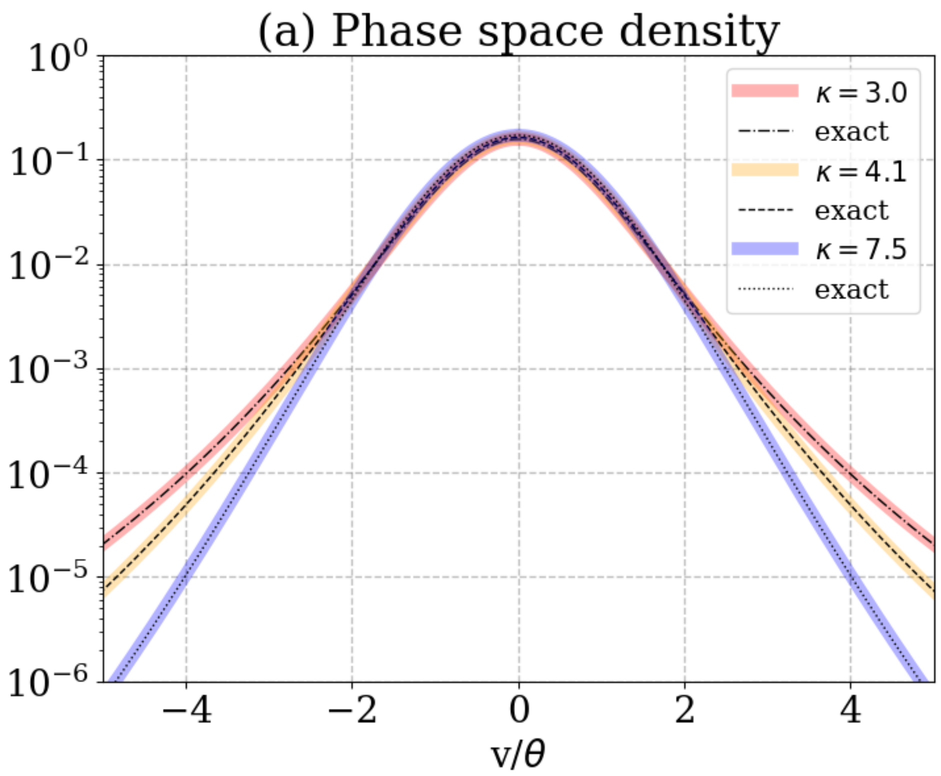
$V \leftarrow \theta\sqrt{\dfrac{-2L}{(a+cL)+\sqrt{(a+cL)^2-4bL}}}$

$v_x \leftarrow V(2U_2 - 1)$

$v_y \leftarrow 2V\sqrt{U_2(1-U_2)}\cos(2\pi U_3)$

$v_z \leftarrow 2V\sqrt{U_2(1-U_2)}\sin(2\pi U_3)$

Approx. Kappa (3/4): Comparison with exact Kappa



- δ(PSD) < PIC noise

$$N_p \lesssim 10^7$$

- Total number

  same

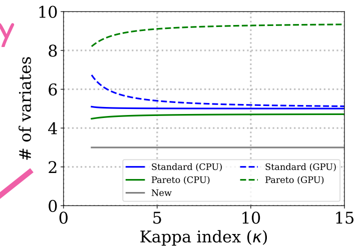
- Total energy

$$\frac{\Delta\mathcal{E}}{\mathcal{E}} \lesssim 10^{-2.5} \quad (\kappa \approx 4.1)$$

$$\frac{\Delta\mathcal{E}}{\mathcal{E}} \lesssim 10^{-3.5} \quad (\kappa \neq 4.1)$$

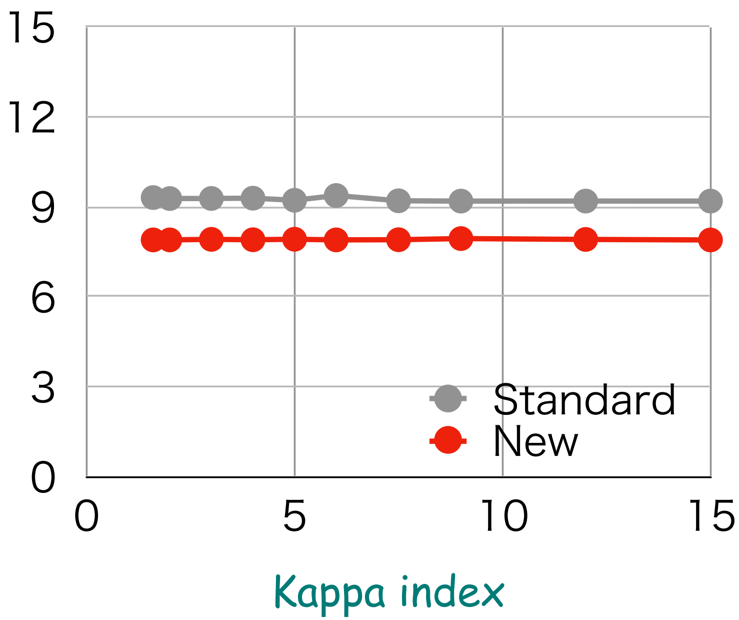
Approx. Kappa (4/4): benchmark

Theory

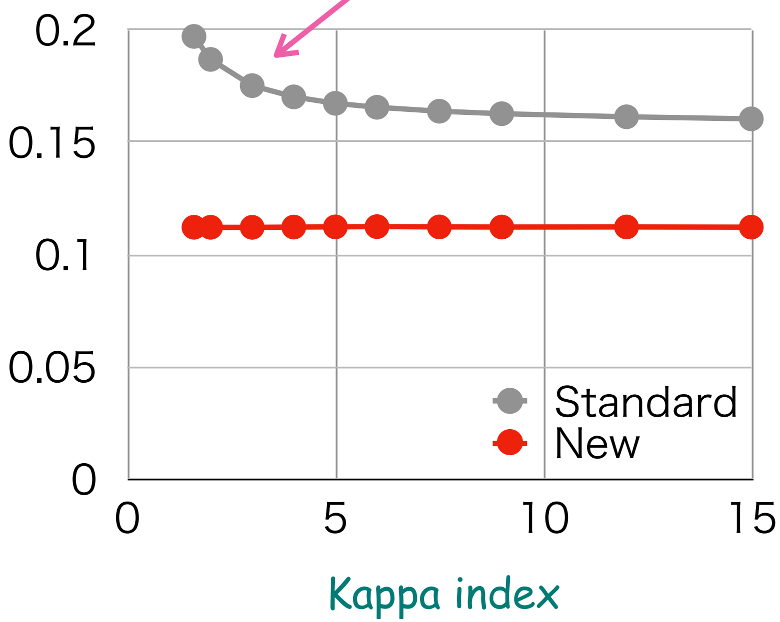


- CPU

Fast

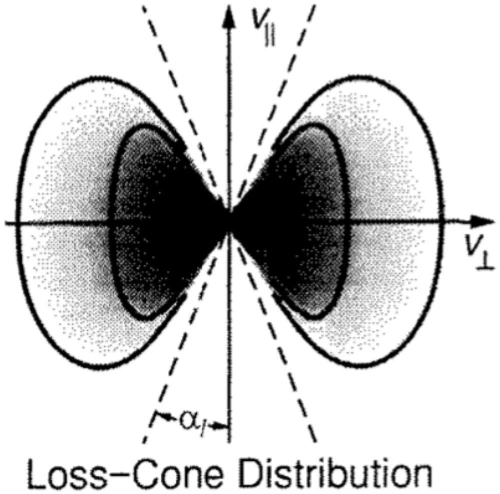


- Nvidia GPU



- Although the Standard method suffers from the SIMT problem, New method runs constantly fast
- SIMT-aware algorithms are useful in plasma simulations

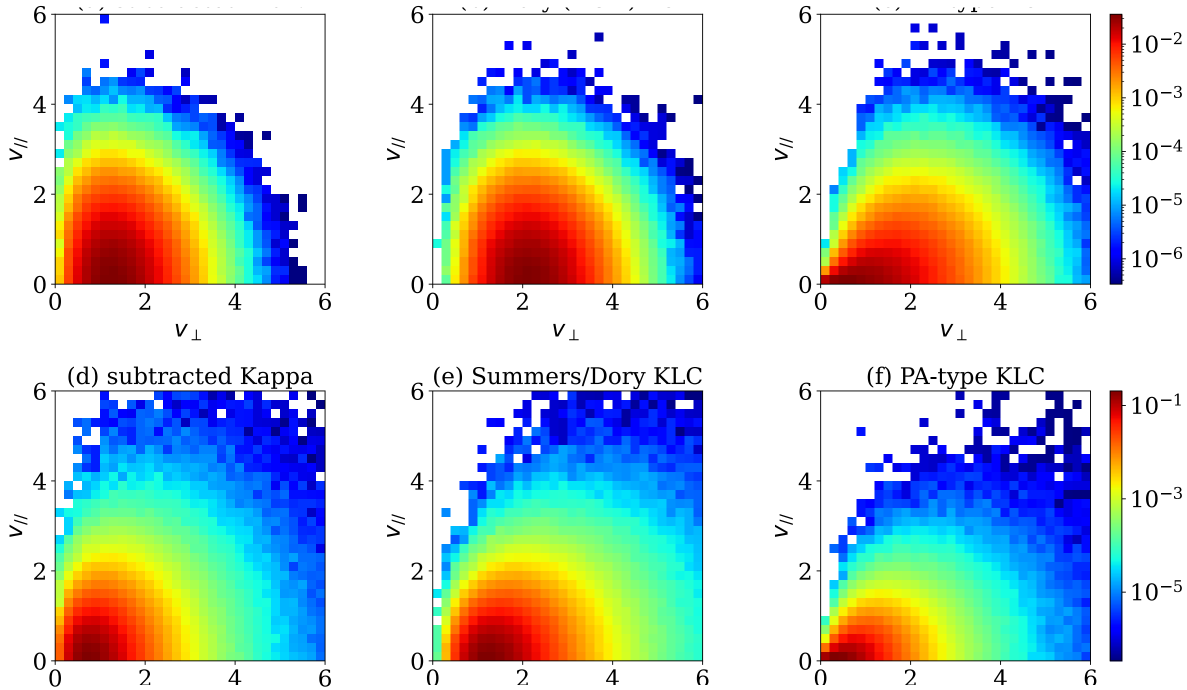
Loss-cone distributions



Subtracted Maxwellian

Dory-type loss-cone

Pitch-angle type loss-cone



Subtracted Kappa

Summers/Dory -type KLC

Pitch-angle type KLC

Subtracted Maxwellian

[Ashour-Abdalla & Kennel, 1978]

$$f(v_\parallel, v_\perp) = \frac{N_0}{\pi^{1/2}\theta_\parallel} \exp\left(-\frac{v_\parallel^2}{\theta_\parallel^2}\right) \times \frac{1}{\pi\theta_\perp^2(1-\beta)} \left\{ \exp\left(-\frac{v_\perp^2}{\theta_\perp^2}\right) - \exp\left(-\frac{v_\perp^2}{\beta\theta_\perp^2}\right) \right\}$$

$$x \equiv v_\perp^2/\theta_\perp^2 \qquad f_X(x) = \frac{1}{1-\beta}\left(\exp(-x) - \exp\left(-\frac{x}{\beta}\right)\right)$$

- Convolution of two exponential distributions

$$x \leftarrow -\log U_1 - \beta \log U_2$$

$$s \sim G_s(s) = \exp(-s), \quad s \geq 0$$

$$t \sim G_t(t) = \frac{1}{\beta}\exp\left(-\frac{t}{\beta}\right), \quad t \geq 0$$

$$G(x) = \int_0^x G_s(s) \cdot G_t(x-s)\, ds = \frac{1}{\beta}\int_0^x \exp\left(-\frac{\beta-1}{\beta}s - \frac{x}{\beta}\right) ds$$
$$= \frac{1}{1-\beta}\left\{\exp(-x) - \exp\left(-\frac{x}{\beta}\right)\right\}$$

- Recipe

Algorithm for the Subtracted Maxwellian

Algorithm 2

generate $U_1, U_2, U_3 \sim U(0,1)$

generate $N \sim \mathcal{N}(0,1)$

$x \leftarrow -\log U_1 - \beta \log U_2$

$v_{\perp 1} \leftarrow \theta_\perp \sqrt{x}\, \cos(2\pi U_3)$

$v_{\perp 2} \leftarrow \theta_\perp \sqrt{x}\, \sin(2\pi U_3)$

$v_\parallel \leftarrow \theta_\parallel \sqrt{1/2}\; N$

return $v_{\perp 1}, v_{\perp 2}, v_\parallel$

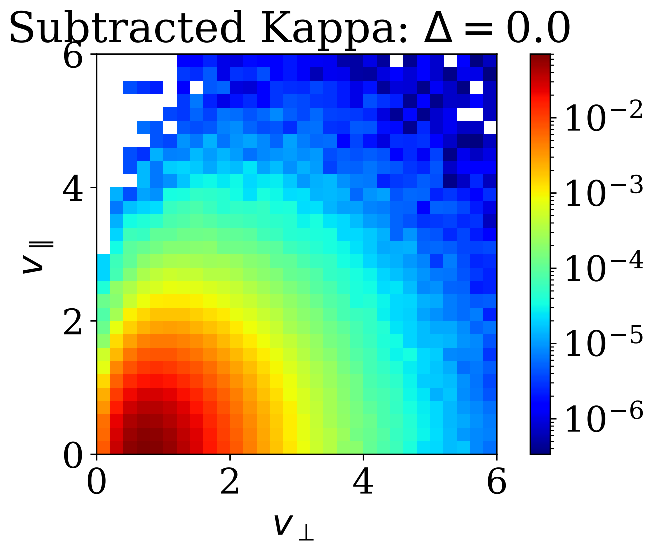
SZ & Nakano 2023 JGR

Subtracted Kappa distribution

[Summers & Stone, 2025]

$$f_{\mathrm{sk}}(v_\parallel, v_\perp)\, d^3v = N \cdot C_{\mathrm{sk}} \left\{ \frac{1-\Delta\beta}{1-\beta} \left(1 + \frac{{v_\parallel}^2}{\kappa\theta_\parallel^2} + \frac{v_\perp^2}{\kappa\theta_\perp^2} \right)^{-(\kappa+1)} - \frac{1-\Delta}{1-\beta} \left(1 + \frac{{v_\parallel}^2}{\kappa\theta_\parallel^2} + \frac{v_\perp^2}{\beta\kappa\theta_\perp^2} \right)^{-(\kappa+1)} \right\} d^3v$$

- Inspired by the t-generator, we divide sub-Max by Gamma



Algorithm for the Subtracted Kappa Distribution

Algorithm 5: subtracted Kappa ($\kappa > 3/2$)

generate $U_1, U_2, U_3 \sim U(0,1)$

generate $N \sim \mathcal{N}(0,1)$

generate $Y \sim \mathrm{Ga}(\kappa - 1/2, 2)$

$x \leftarrow -\log U_1 - \beta \log(\min(\frac{U_2}{1-\Delta}, 1))$

$v_{\perp 1} \leftarrow \theta_\perp \sqrt{2\kappa x} \cos(2\pi U_3)/\sqrt{Y}$

$v_{\perp 2} \leftarrow \theta_\perp \sqrt{2\kappa x} \sin(2\pi U_3)/\sqrt{Y}$

$v_\parallel \leftarrow \theta_\parallel \sqrt{\kappa} N/\sqrt{Y}$

return $v_{\perp 1}, v_{\perp 2}, v_\parallel$

SZ, Usami, & Matsukiyo 2026 JGR

Summary

Based on three elements:

1. Uniform
2. Normal
3. Gamma

- 1. Standard t-generator [1]
  - Well known?
- 2. Pareto method [2]
  - Simple & portable
- 3. Approximate generator [3]
  - GPU-friendly
- 4. Subtracted Kappa generator [4,5]
  - Useful for Kappa Loss-Cone (KLC) problems

- References

- [1] Zenitani & Nakano, *Phys. Plasmas*, 2022, **Link**
- [2] Zenitani, *RNAAS*, 2025, **Link**
- [3] Zenitani & Umeda, *EPS*, 2026, **Link**
- [4] Zenitani & Nakano, *JGR Space Phys.*, 2023, **Link**
- [5] Zenitani et al., *JGR Space Phys.*, 2026, **Link**