Machine Learning for Memory Reduction in the Implicit Monte Carlo Simulations of the Thermal Radiative Transfer [1]

Anna Matsekh
Luis Chacon, HyeongKae Park, Guangye Chen

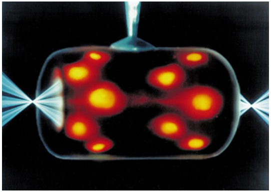
Theoretical Division
Los Alamos National Laboratory

[1] LA-UR-18-26613

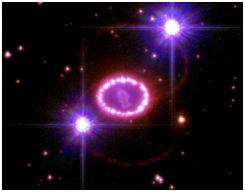
Thermal Radiative Transfer

- *Thermal Radiative Transfer (TRT) equations*
  - describe propagation and interaction of photons with the surrounding material
  - are challenging to solve due to the stiff non-linearity and high-dimensionality of the problem

- *TRT applications at LANL include simulations of*
  - Inertial Confinement Fusion experiments
  - astrophysical events



(a) ICF



(b) supernova

Figure: TRT applications

Implicit Monte Carlo Simulations

- *Advantages, compared to the deterministic case*
  - easier to extend to complex geometries and higher dimensions
  - easier to parallelize
- *Disadvantages*
  - Monte Carlo solutions to IMC equations exhibit statistical variance and IMC convergence rate is estimated to be

$$O(1/\sqrt{N_p})$$

    where N_p is the number of simulation particles
  - Even when advanced variance reduction techniques employed, Monte Carlo simulations
    - exhibit slow convergence
    - prone to statistical errors
    - require a very large number of simulation particles
  - Implicit Monte Carlo codes are typically very large, long running codes with large memory requirements at checkpointing & restarting

Machine Learning for IMC

Project Goal: use parametric Machine Learning methods in order to reduce memory requirements at checkpointing & restarting in the IMC simulations of Thermal Radiative Transfer using

- *Expectation Maximization and Weighted Gaussian Mixture Model-based approach for 'particle-data compression', introduced in Plasma Physics to model Maxwellian particle distributions by Luis Chacon and Guangye Chen*
- *Expectation Maximization with Weighted Hyper-Erlang Model in order to compress isotropic IMC particle data in the frequency domain*
- *Expectation Maximization and von Mises Mixture Models for compression of anisotropic directional IMC data on a circle (work-in-progress)*

TRT Equations

Consider 1-d Transport Equation without scattering and without external sources in the Local Thermodynamic Equilibrium (LTE):

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mu\frac{\partial I_\nu}{\partial x} + \sigma_\nu\, I_\nu = \frac{1}{2}\,\sigma_\nu\, B_\nu \tag{1}$$

coupled to the Material Energy Equation

$$c_v\,\frac{\partial T}{\partial t} = \iint \sigma_\nu\, I_\nu\, \mathrm{d}\nu\, \mathrm{d}\mu - \int \sigma_\nu\, B_\nu \mathrm{d}\nu \tag{2}$$

where the emission term

$$B_\nu(T) = \frac{2\,h\,\nu^3}{c^2}\,\frac{1}{e^{\frac{h\,\nu}{k\,T}} - 1} \tag{3}$$

is the Planckian (Blackbody) distribution and

- $I_\nu = I(x, \mu, t, \nu)$ - radiation intensity
- ν - frequency, T - temperature
- σ_ν - opacity, c_v - material heat capacity
- k - Boltzmann constant, h - Planck's constant, c - speed of light

Implicit Monte Carlo Method of Fleck and Cummings

Transport Equation

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mu \frac{\partial I_\nu}{\partial x} + (\sigma_{\nu a} + \sigma_{\nu s})\, I_\nu = \frac{1}{2}\sigma_{\nu a}\, c\, u_r^n + \frac{1}{2}\sigma_{\nu s}\,(b_\nu/\sigma_p) \iint \sigma_{\nu'}\, I_{\nu'}\, \mathrm{d}\nu'\, \mathrm{d}\mu \tag{4}$$

Material Temperature Equation $(T^n = T(t_n) \approx T(t),\ t_n \le t \le t_{n+1})$

$$c_\nu\, T^{n+1} = c_\nu\, T^n - f\, \sigma_p\, c\, \Delta t\, u_r^n + f \int_{t_n}^{t_{n+1}} \mathrm{d}t \iint \sigma_{\nu'}\, I_{\nu'}\, \mathrm{d}\nu'\, \mathrm{d}\mu \tag{5}$$

- $f = 1/(1 + \alpha\, \beta\, c\, \Delta t\, \sigma_p)$ - *Fleck factor*
- $\sigma_{\nu a} = f\, \sigma_\nu$ - *effective absorption opacity*
- $\sigma_{\nu s} = (1 - f)\, \sigma_\nu$ - *effective scattering opacity*
- u_r - *radiation energy density*, $b_\nu(T)$ - *normalized Planckian*
- $\sigma_p = \int \sigma_\nu\, b_\nu\, d\nu$ - *Planck opacity*
- $\alpha \in [0, 1]$ $s.t.$ $u_r \approx \alpha\, u_r^{n+1} + (1 - \alpha)\, u_r^n; \beta = \partial u_r / \partial u_m$

Monte Carlo Implementation of the IMC method

On each simulation time step

- *Particle Sourcing* – calculating total energy in the system from different sources due to the
  - boundary conditions and initial conditions
  - external sources and emission sources
- *Particle Tracking* – tracking distance to an event in time:
  - distance to the spacial cell boundary
  - distance to the end of the time-step
  - distance to the next collision
- *Tallying* - computing sample means of such quantities as
  - energy deposited due to all effective absorptions
  - volume-averaged energy density
  - fluxes
- *Calculate next time-step temperature approximation* T^{n+1}

The concept of an IMC particle

- An IMC particle is a simulation abstraction representing a 'radiation energy bundle' characterized by
  - the energy-weight (*relative number of photons represented by an IMC particle*)
  - spacial location
  - angle of flight
  - frequency group it belongs to
- On each simulation time step $t_n \leq t \leq t_{n+1}$ an IMC particle can undergo the following events:
  - escape through the boundaries
  - get absorbed by the material
  - scattering / re-emission
  - survive (particle goes to census at t_{n+1})
- Surviving particles are called *census particles* and have to be stored in memory to be reused on the next time-step

Can we 'learn' the probability distribution function describing census particles at the end of each time-step and store in memory only this distribution?

Expectation Maximization Method

- *Expectation Maximization (EM) is an iterative method for estimating parameters from probabilistic models. It is typically applied to the Finite Mixture Models*

$$\mathcal{P}(x_j, \theta) = \sum_{i=1}^{k} p_i \, \mathcal{F}(x_j, \theta_i) \tag{6}$$

  - $\mathcal{F}(x_j, \theta_i)$ - *pdf* with the parameter vector θ_i
  - x_j - data points from the sample $X^n = (x_1, \, x_2, \, \ldots, \, x_n)$
  - p_i - probability of $\mathcal{F}(x_j, \theta_i)$ in the mixture $\sum_{i=1}^{n} p_i = 1$

- *EM algorithm alternates between the Expectation and Maximization steps:*
  - *Expectation (E) step - computing priors (probabilities)*
  - *Maximization (M) step - updating model parameters θ_i that maximize expected Likelihood function*

$$\mathcal{L}(X^n) = \sum_{j=1}^{n} \ln \sum_{i=1}^{k} p_i \, \mathcal{F}(x_j, \theta_i) \tag{7}$$

EM and Hyper-Erlang Model

- *A Hyper-Erlang Model* $\mathcal{H}(\nu, \alpha, \beta)$ is a mixture of Erlang distributions $\mathcal{E}(\nu, \alpha, \beta)$:

$$\sum_{k=1}^{m} p_k \, \mathcal{E}(\nu_i, \alpha_k, \beta_k) = \sum_{k=1}^{m} p_k \, \frac{1}{(\alpha_k - 1)!} \, \nu_i^{\alpha_k - 1} \, \beta_k^{-\alpha_k} \, e^{(-\nu_i/\beta_k)}$$

  where
  - $\{\nu_1, \dots, \nu_n\}$ is an *iid* data sample
  - $\alpha_k > 0$ - integer *shape parameter*, $\beta_k > 0$ - real *scale parameter*
- *Expectation Maximization priors*

$$\gamma_{ik} = \frac{p_k \, \mathcal{E}(\nu_i, \alpha_k, \beta_k)}{\sum_{k=1}^{m} p_k \, \mathcal{E}(\nu_i, \alpha_k, \beta_k)}, \quad i = 1, \dots, n, \;\; k = 1, \dots m \qquad (8)$$

- *Maximum Likelihood parameter estimates*

$$\beta_k = \frac{\sum_{i=1}^{n} \gamma_{ik} \, \nu_i}{\alpha_k \, \sum_{i=1}^{n} \gamma_{ik}}, \quad p_k = \frac{\sum_{i=1}^{n} \gamma_{ik}}{n}, \;\; k = 1, \dots, m \qquad (9)$$

Planckian and Erlang Distributions

- *In the LTE emission term in the Transport Equation is given by the Planckian B_ν*
- *Frequency-normalized Planckian density function*

$$b_\nu(T) = \frac{B_\nu(T)}{\int_0^\infty B_\nu(T) d\nu} = \frac{15}{\pi^4} \frac{\nu^3}{T^4 \left(e^{\nu/T} - 1\right)} \tag{10}$$

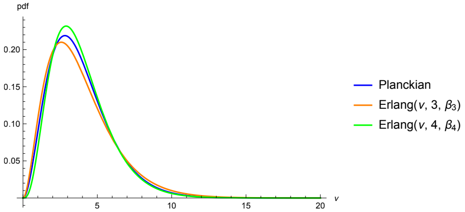
- *There are no closed-form EM estimates of T for $b_\nu(T)$ mixtures*
- *Can we model Planckian-like distributions with Erlang mixtures?*
    - Erlang distribution

$$\mathcal{E}(\nu, \alpha, T) = \frac{1}{(\alpha - 1)!} \frac{\nu^{\alpha-1}}{T^\alpha e^{\nu/T}} \tag{11}$$

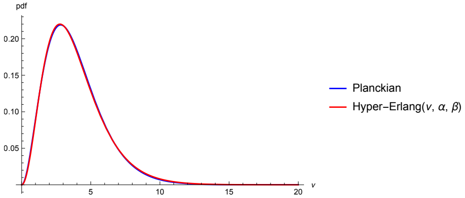
    - Consider Erlang distributions with shapes $\alpha = \{3, 4\}$

$$\mathcal{E}(\nu, 3, T) = \frac{1}{2} \frac{\nu^2}{T^3 e^{\nu/T}}, \quad \mathcal{E}(\nu, 4, T) = \frac{1}{6} \frac{\nu^3}{T^4 e^{\nu/T}} \tag{12}$$

Planckian and Erlang Distributions



(a) Planckian and Erlang pdf



(b) Hyper-Erlang pdf ($\alpha = \{3, 4\}$-mixture)

Figure: Planckian frequency data sample size: $200,000$

Hyper-Erlang Model for IMC

- IMC particles in group g are characterized by the same average frequency, i.e. have the same likelihood to be drawn from g
- Cumulative energy-weight $\omega^g = \sum_{j=1}^{np_g} \omega_j^g$ of np_g particles in g is the relative number of photons represented by all particles in g
- *Weighted Log-Likelihood of the Hyper-Erlang IMC Model*

$$\ln \prod_{g=1}^{n} \left(\sum_{k=1}^{m} p_k \, \mathcal{E}(\nu_g, \alpha_k, \beta_k) \right)^{\omega^g} = \sum_{g=1}^{n} \omega^g \ln \sum_{k=1}^{m} p_k \, \mathcal{E}(\nu_g, \alpha_k, \beta_k)$$

- *Weighted Maximum Likelihood / EM parameter estimates*

$$\gamma_{gk} = \frac{p_k \, \mathcal{E}(\nu_g, \alpha_k, \beta_k)}{\sum_{k=1}^{m} p_k \, \mathcal{E}(\nu_g, \alpha_k, \beta_k)}, \quad g = 1, \ldots, n, \;\; k = 1, \ldots m$$

$$\beta_k = \frac{\sum_{g=1}^{n} \omega^g \, \gamma_{gk} \, \nu_g}{\alpha_k \sum_{g=1}^{n} \omega^g \, \gamma_{gk}}, \quad p_k = \frac{\sum_{g=1}^{n} \gamma_{gk}}{\sum_{g=1}^{n} \omega^g}$$

Weighted Hyper-Erlang Model for compressing IMC particles

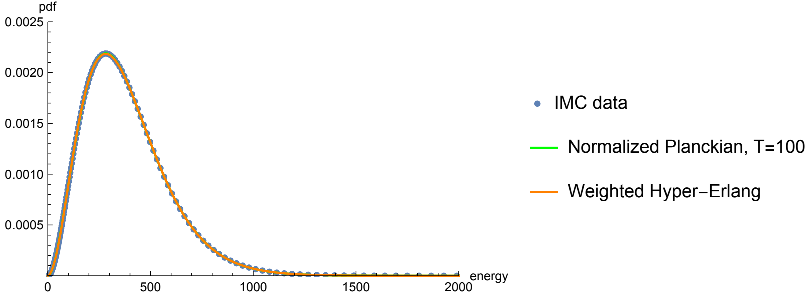


Figure: Planckian IMC data sample from the time step $t_0 = 0$ (initial temperature $T_0 = 100$, 500 groups), Normalized Planckian $b_\nu(T)$ at $T = 100$ and the Hyper-Erlang Model $\mathcal{H}(\nu, \alpha, \beta)$ of the sample with 20 mixture elements.

Note: for $\mathbf{x} \sim b_\nu(T)$ and $\mathbf{y} \sim \mathcal{H}(\nu, \alpha, \beta)$: $\|\mathbf{x} - \mathbf{y}\|_\infty = 6.2 \times 10^{-6}$

Weighted Hyper-Erlang Model

Table: Parameters of the Hyper-Erlang Model of the Planckian IMC data sample with initial temperature $T_0 = 100$ from the time step $t_0 = 0$. Shape parameters α_k are fixed, scale parameters β_k model radiation temperature and $k = 1, 2, \ldots, 20$.

p_k	α_k	β_k
0.	1	358.246
0.0248371	2	182.592
0.2026990	3	97.1897
0.2191770	4	71.5419
0.1673600	5	73.3836
0.1283270	6	69.3727
0.0927585	7	64.5944
0.0622957	8	61.4264
0.0394477	9	60.4135
0.0244054	10	60.7461
0.0151412	11	61.0587
0.0094205	12	60.8283
0.0058112	13	59.8523
0.0035262	14	58.2133
0.0020989	15	56.3901
0.0012253	16	54.7773
0.0007035	17	53.6442
0.0004003	18	53.2266
0.0002299	19	53.6196
0.0001359	20	54.2246

Summary and Future Work

- *Memory usage reduction during the code checkpointing and restarting steps is of great importance in the IMC simulations*
- *Innovation: storing only parameters of the probability distributions of census particles in place of the data structures describing them in order to reduce IMC storage requirements*
- *Innovation: Expectation Maximization and Weighted Hyper-Erlang Models can accurately model Planckian and therefore are appropriate for compression of isotropic IMC census data in the frequency domain in LTE*
- *Work-in-Progress: We are currently researching applicability of the Expectation Maximization and von Mises Mixture Models for compression of anisotropic directional IMC census data on a circle*