機械学習を用いた
初代星形成過程における
化学進化のエミュレーション

京都大学　理学研究科　宇宙物理学専攻
M1 小野　壮洵（共同研究者：杉村　和幸（北海道大学））

Introduction 1

Accepted for publication in ApJ

arXiv > astro-ph > arXiv:2508.16114

Search... Help | Adva

Astrophysics > Astrophysics of Galaxies

[Submitted on 22 Aug 2025 (v1), last revised 13 Nov 2025 (this version, v2)]

Neural-Network Chemical Emulator for First-Star Formation: Robust Iterative Predictions over a Wide Density Range

Sojun Ono, Kazuyuki Sugimura

We present a neural-network emulator for the thermal and chemical evolution in Population III star formation. The emulator accurately reproduces the thermochemical evolution over a wide density range spanning 21 orders of magnitude (10^{-3}-10^{18} cm^{-3}), tracking six primordial species: H, H_2, e^-, H^+, H^-, and H_2^+. To handle the broad dynamic range, we partition the density range into five subregions and train separate deep operator networks (DeepONets) in each region. When applied to randomly sampled thermochemical states, the emulator achieves relative errors below 10% in over 90% of cases for both temperature and chemical abundances (except for the rare species H_2^+). The emulator is roughly ten times faster on a CPU and more than 1000 times faster for batched predictions on a GPU, compared with conventional numerical integration. Furthermore, to ensure robust predictions under many iterations, we introduce a novel timescale-based update method, where a short-timestep update of each variable is computed by rescaling the predicted change over a longer timestep equal to its characteristic variation timescale. In one-zone collapse calculations, the results from the timescale-based method agree well with traditional numerical integration even with many iterations at a timestep as short as 10^{-4} of the free-fall time. This proof-of-concept study suggests the potential for neural network-based chemical emulators to accelerate hydrodynamic simulations of star formation.

Comments: 19 pages, 7 figures, Accepted for publication in ApJ
Subjects: **Astrophysics of Galaxies (astro-ph.GA)**; Instrumentation and Methods for Astrophysics (astro-ph.IM); Solar and Stellar Astrophysics (astro-ph.SR); Machine Learning (cs.LG)
Cite as: arXiv:2508.16114 **[astro-ph.GA]**
(or arXiv:2508.16114v2 **[astro-ph.GA]** for this version)
https://doi.org/10.48550/arXiv.2508.16114

Submission history

From: Sojun Ono [view email]
[v1] Fri, 22 Aug 2025 06:11:03 UTC (467 KB)
[v2] Thu, 13 Nov 2025 01:58:01 UTC (894 KB)

https://arxiv.org/abs/2508.16114

Introduction

星の性質は、ガスの性質や周囲の環境に左右される

▶ 様々な星の性質を知るには様々な条件で熱的・化学的発展を計算することが必要

- 多変数でstiffな微分方程式をImplicitに何度も計算する必要がある。

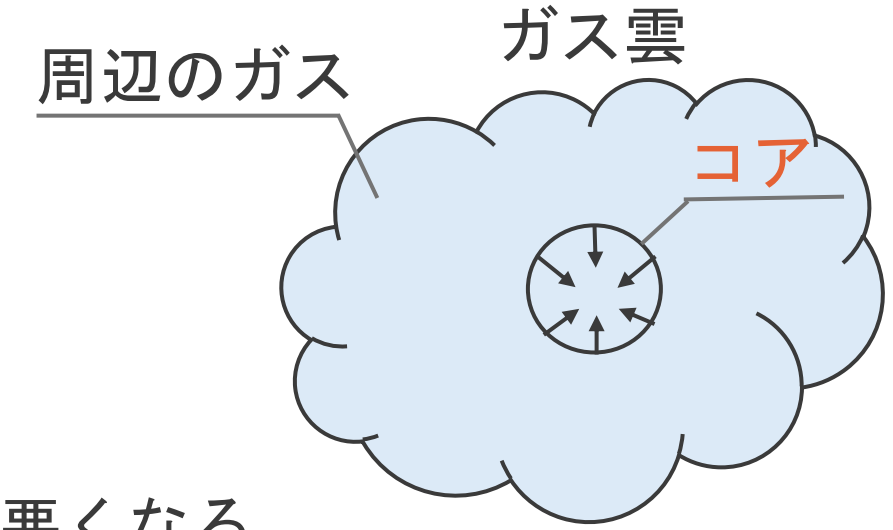
従来の計算法（数値積分法）

化学的進化

$$\dot{y}_i = \sum_j A_{ij}(n_{\mathrm{H}}, T, \boldsymbol{y})$$

熱的進化

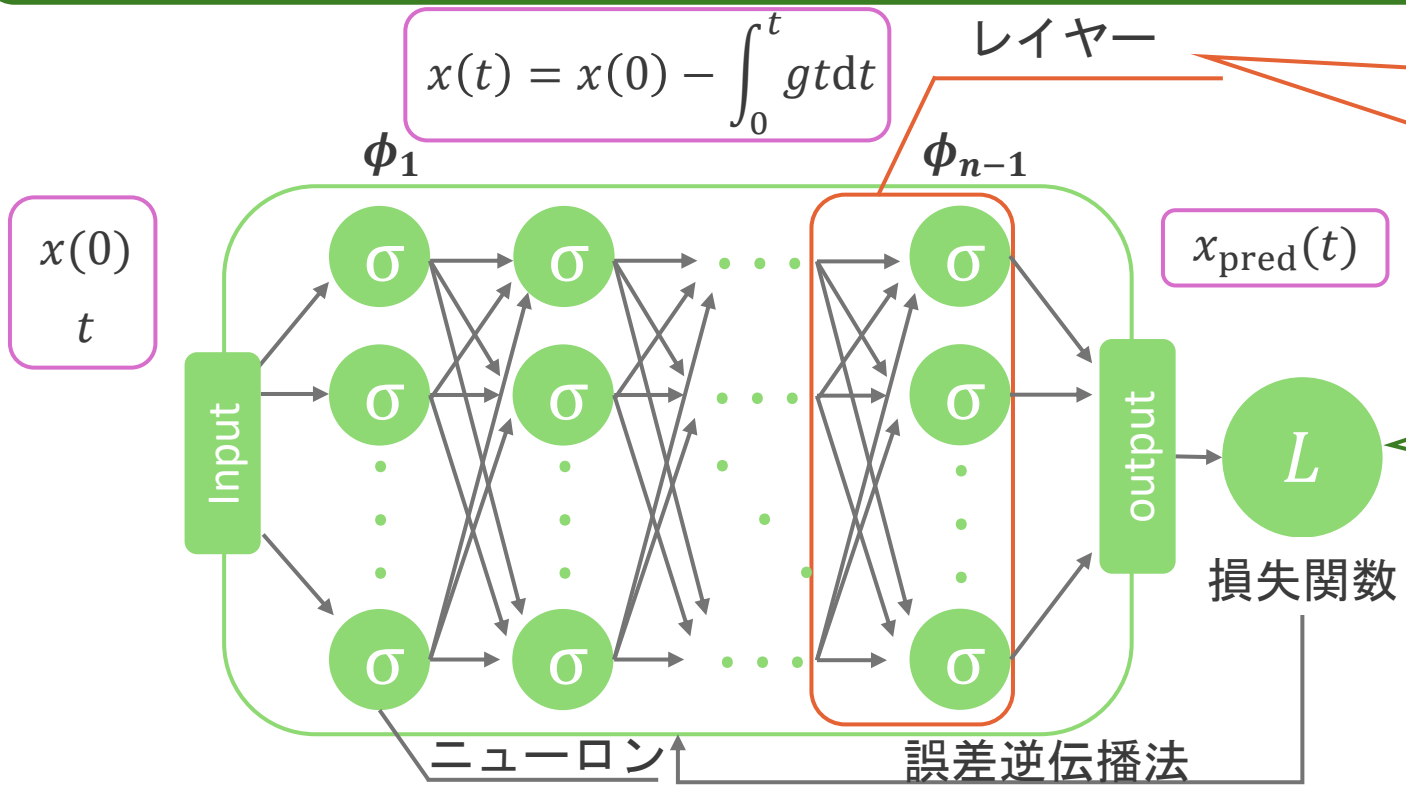
$$n_{\mathrm{H}}(\Sigma_i y_i + y_{\mathrm{He}}) c_v \dot{T} = \Gamma(n_{\mathrm{H}}, T, \boldsymbol{y}) - \Lambda\,(n_{\mathrm{H}}, T, \boldsymbol{y})$$



☹ 計算コストがかなり大きくなり、並列計算効率が悪くなる

▶ 数値積分を解く際の計算コストの削減手段として
ニューラルネットワーク（NN）が注目されている

Introduction



$$\boldsymbol{\phi_{i+1}} = \sigma(\boldsymbol{\phi_i W_i} + \boldsymbol{b_i})$$

$$\boldsymbol{y} = \boldsymbol{\phi_{n-1} W_{n-1}} + \boldsymbol{b_{n-1}}$$

σ は非線形関数 (e.g., sigmoid, ReLU)

$$L = \frac{1}{N_{\mathrm{out}}} \sum_j \left(x_{j,\mathrm{pred}} - x_{j,\mathrm{true}}\right)^2$$

Lが小さくなるように$\boldsymbol{W}, \boldsymbol{b}$を更新していく

Lが十分小さい ⇒NNはデータから十分正しいパターンや関係を学習

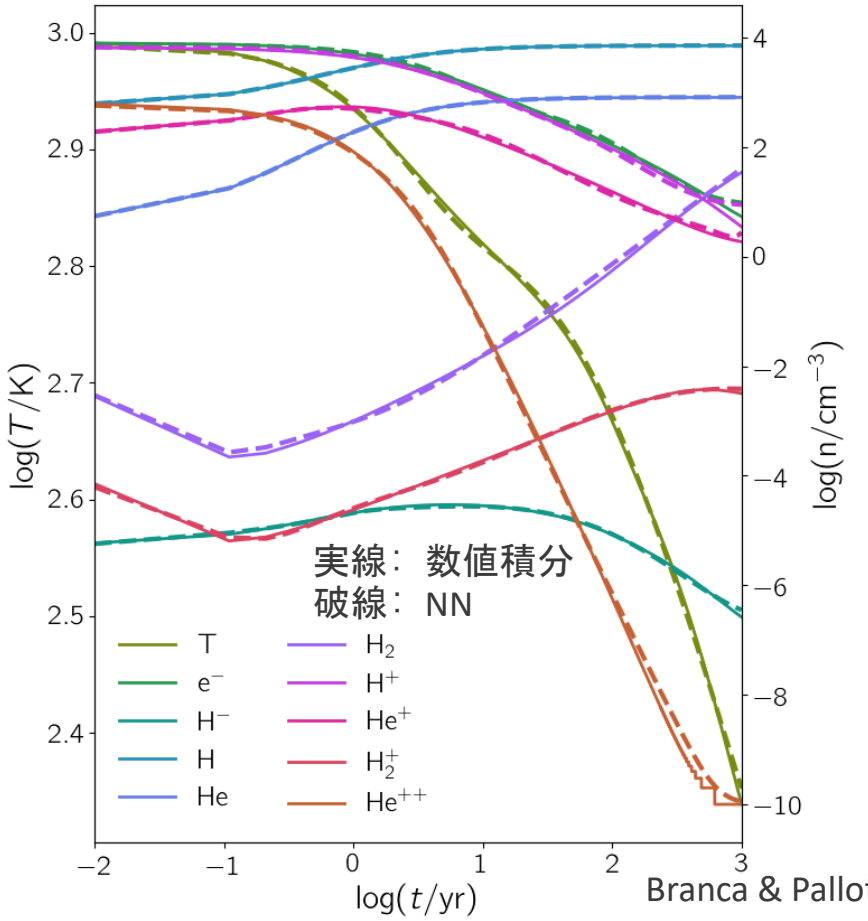
実際にNNが星形成過程の計算へ応用可能か検証の必要有り

本研究では、第一歩として金属や輻射のないシンプルな熱的化学的発展をもとに、星形成への機械学習の応用法を検討

先行研究

➢ Branca & Pallottini 2024

- 星間物質の発展をモデル化したエミュレータを設計
- DeepONet が主なNNの構造
- 低密度 $(n_{\mathrm{H}} \lesssim 10^4 \mathrm{cm}^{-3})$領域で、NNの有用性が実証



Branca & Pallottini 2024

本研究では同様の手法を用いる

星形成に用いるためには、高密度$(\sim 10^{22} \mathrm{cm}^{-3})$領域まで拡張可能か考える必要がある

方法 | 基礎方程式

化学的発展

化学種: H, H_2, e^-, H^+, H^-, H_2^+

各化学種は18種の反応を含む
以下の常微分方程式に従って発展する

$$\dot{y}_i = \sum_j A_{ij}(n_{\rm H}, T, \boldsymbol{y})$$

$y_i = n_i/n_{\rm H}$: i番目の化学種の存在量

n_i: i番目の化学種の数密度

$n_{\rm H}$: 水素原子核の数密度

e.g., $A_{{\rm H}j} = k_j(n_{\rm H}, T)y({\rm H}^+)y({\rm e}^-)n_{\rm H}$

$k_j(n_{\rm H}, T)$: j番目の反応係数

熱的発展

温度は以下の常微分方程式に従う

$$n_{\rm H}(\Sigma_i y_i + y_{\rm He})c_v\dot{T} = \Gamma - \Lambda$$

$c_v(T)$: 1粒子あたりの定積熱容量
e.g., $c_v = \frac{3}{2}k_B$ (単原子分子理想気体)

$\Gamma(n_{\rm H}, T, \boldsymbol{y})$: 加熱関数

$\Lambda(n_{\rm H}, T, \boldsymbol{y})$: 冷却関数

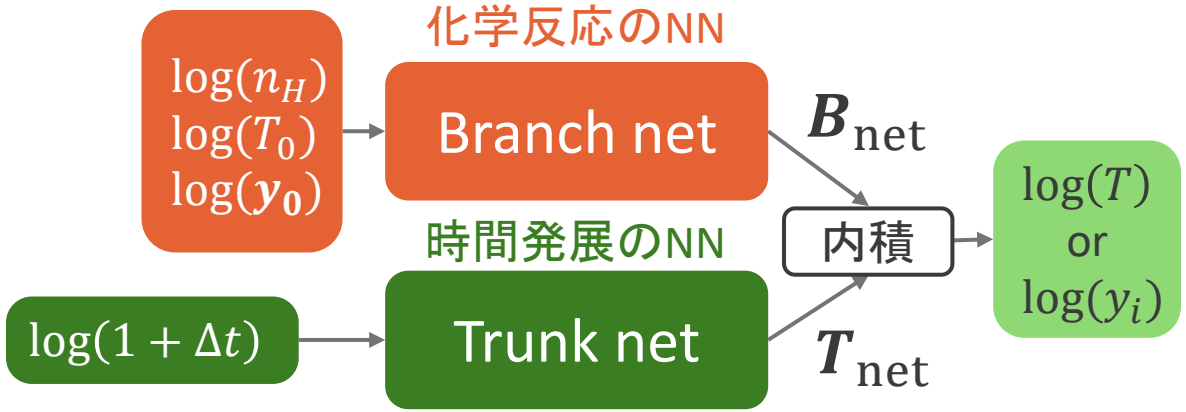
$y_{\rm He}$: (He 数密度) / $n_{\rm H}$ (=const.)

 Sugimura et al. 2020, 2023; Sadanari et al. 2021

方法 | 主なNN構造

DeepONet (Lu et al. 2019)

- DeepONetを主なNN構造として使う
- DeepONetは二つのネットワーク (branch net, trunk net)からなる



DeepONetの模式図

方法 | 幅広い密度を計算する方法

Branca & Pallottini 2024

本研究の調査範囲（高密度）

Saha方程式

Region 0	Region 1	Region 2	Region 3	Region 4	Region 5
$10^{-3}\sim10^{4}\mathrm{cm}^{-3}$	$10^{4}\sim10^{8}\mathrm{cm}^{-3}$	$10^{8}\sim10^{12}\mathrm{cm}^{-3}$	$10^{12}\sim10^{15}\mathrm{cm}^{-3}$	$10^{15}\sim10^{18}\mathrm{cm}^{-3}$	$10^{18}\sim10^{22}\mathrm{cm}^{-3}$
DeepONet	DeepONet	DeepONet	DeepONet	DeepONet	Table
1つの化学種	1つの化学種	1つの化学種	1つの化学種	1つの化学種	全ての化学種
10層	10層	10層	10層	10層	NNではない

低密度 ⟷ 高密度

データセット

各領域における初期条件のパラメータの範囲

	$\log(n_{\mathrm{H}}/\mathrm{cm}^{-3})$	$\log(T_0/\mathrm{K})$	$\log(y_0(\mathrm{H}))$	$\log(y_0(\mathrm{H}^2))$	$\log(y_0(\mathrm{e}^-))$
Region 0	$-3\sim4$	$\log(20)\sim4.5$	$-6\sim0$	$-8\sim\log(0.5)$	$-10\sim-4$
Region 1	$4\sim8$	$\log(20)\sim4.5$	$-6\sim0$	$-6\sim\log(0.5)$	$-10\sim-4$
Region 2	$8\sim12$	$\log(20)\sim4$	$-6\sim0$	$-6\sim\log(0.5)$	$-12\sim-6$
Region 3	$12\sim15$	$\log(20)\sim4$	$-6\sim0$	$-6\sim\log(0.5)$	$-14\sim-8$
Region 4	$15\sim18$	$\log(20)\sim4$	$-6\sim0$	$-6\sim\log(0.5)$	$-15\sim-9$

	$\log(y_0(\mathrm{H}^+))$	$\log(y_0(\mathrm{H}^-))$	$\log(y_0(\mathrm{H}_2^+))$	$\Delta t/\mathrm{s}$
Region 0	$-10\sim-4$	$-19\sim-13$	$-19\sim-13$	$0\sim10^{0.5}t_{\mathrm{ff}}\ (=\Delta t_{\mathrm{max}})$
Region 1	$-10\sim-4$	$-20\sim-14$	$-20\sim-14$	
Region 2	$-12\sim-6$	$-21\sim-15$	$-21\sim-15$	
Region 3	$-14\sim-8$	$-21\sim-15$	$-21\sim-15$	
Region 4	$-15\sim-9$	$-22\sim-16$	$-22\sim-16$	

DeepONetが学習できているのかの検証

初期値

$$n_{\mathrm{H}} = 10^{12}\ \mathrm{cm}^{-3}$$

$$T = 23\ \mathrm{K}$$

$$y(\mathrm{H2}) = 5.7 \times 10^{-5}$$

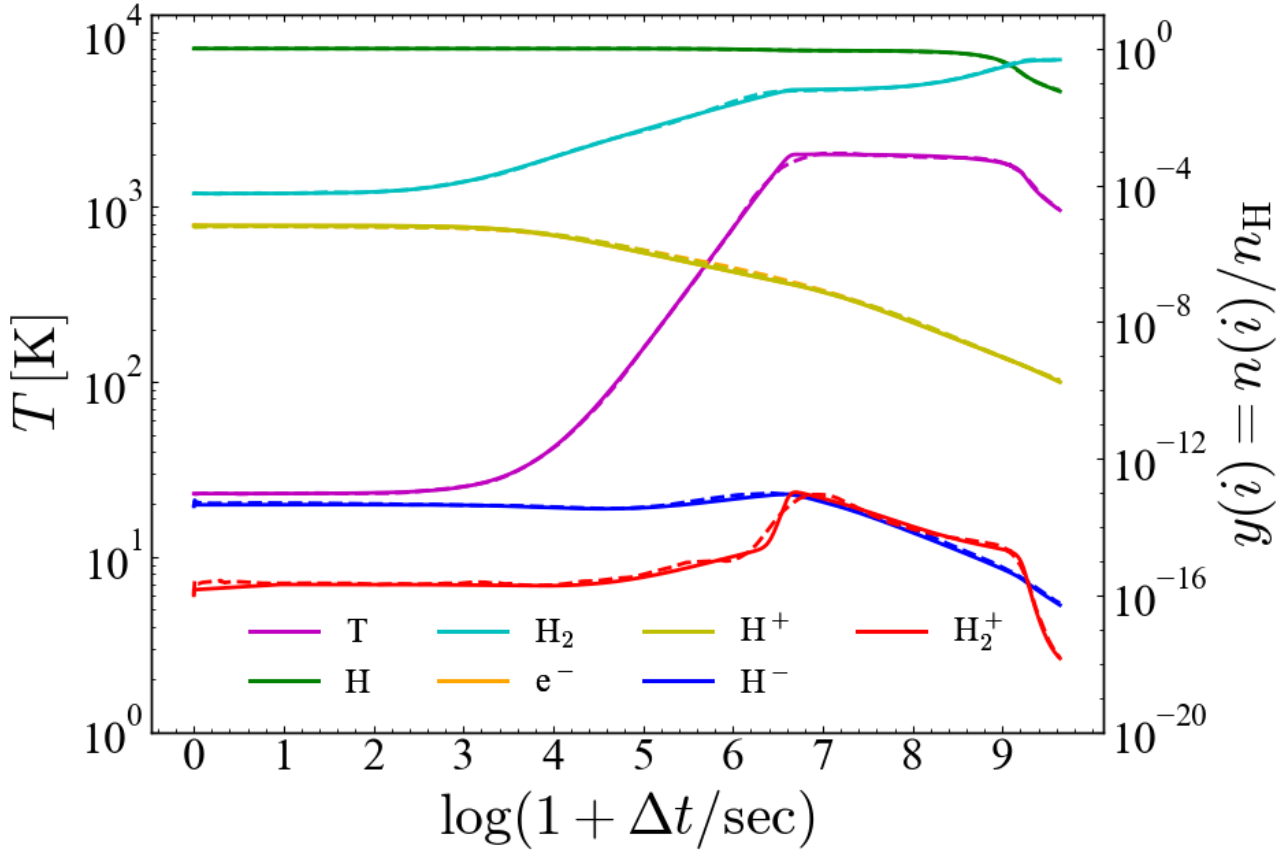
$$y(\mathrm{H}^+) = 6.6 \times 10^{-6}$$

$$y(\mathrm{H}^-) = 4.0 \times 10^{-14}$$

$$y(\mathrm{H}_2^+) = 1.0 \times 10^{-16}$$

実線: 数値積分、破線: DeepONet

$$\log(1+\Delta t) = i/1000 \times \log(1+\Delta t_{\max})$$



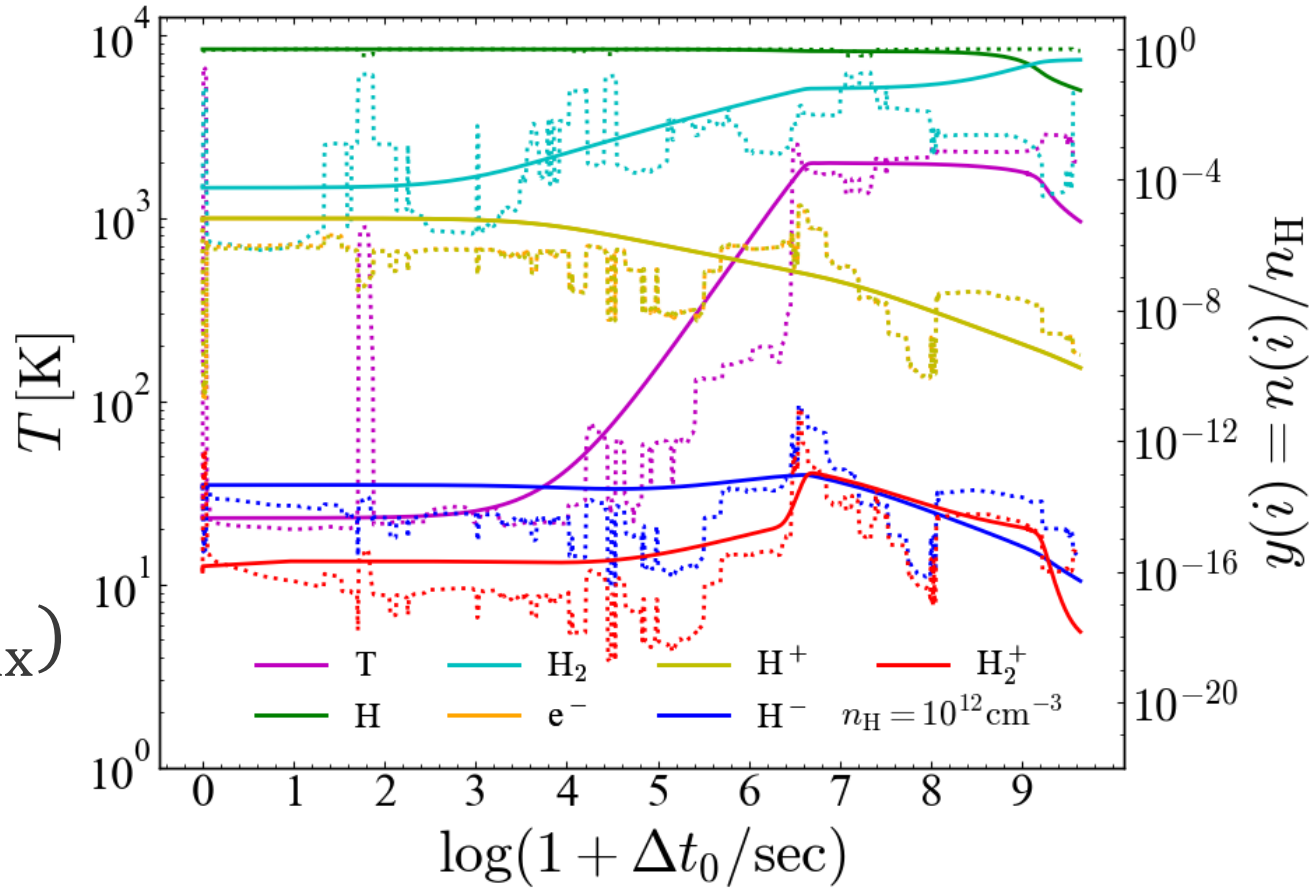
結果 | 単純にiterateした場合

- 星形成の流体シミュレーションに組み込むことを考えたい
- Δtを短く取ったり、長いタイムスケールを計算したりする必要がありiterationが多くなる
  - NNを何度も用いる必要がある

$$\log(1+\Delta t_0) = i/1000 \log(1+\Delta t_{\max})$$

$$\Delta t = \Delta t_0/1000$$

実線: 数値積分、点線: 単純なiterate



誤差を抑制する方法

星形成の流体シミュレーションに組み込む際はΔtを短く取ることがある
NNを何度も用いる必要があり、誤差が蓄積する

▶ Tやyの変化量を調整することで誤差を抑制する

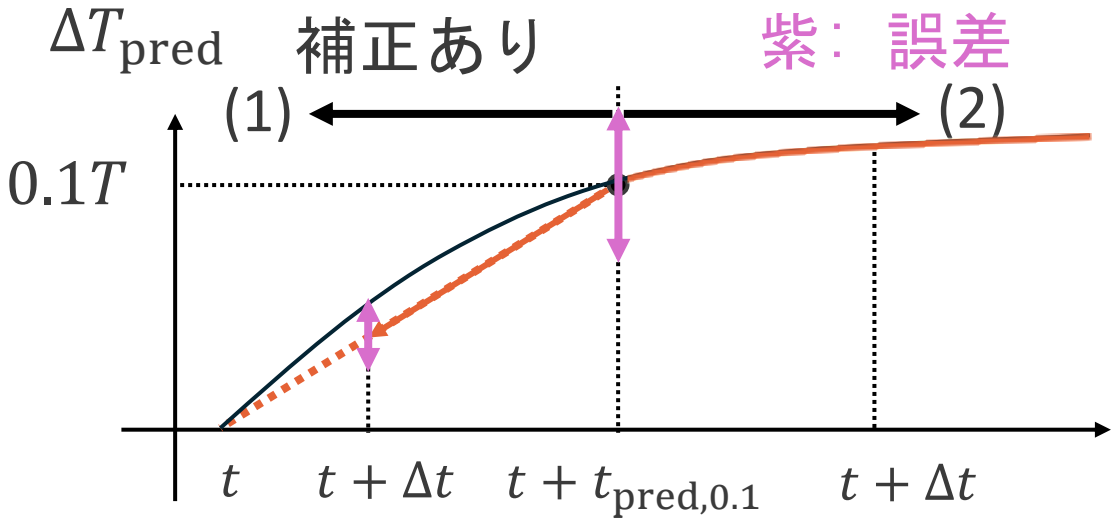
温度のNNの相対誤差が<<10%であるとき、　$t_{\mathrm{pred},0.1}$ = (T が10%変化する時間)

$$\Delta T(\Delta t) = \begin{cases} \dfrac{\Delta t}{t_{\mathrm{pred},0.1}} \Delta T_{\mathrm{pred}}\left(t_{\mathrm{pred},0.1}\right) & \text{for } \Delta t < t_{\mathrm{pred},0.1} \quad (1) \\ \Delta T_{\mathrm{pred}}(\Delta t) & \text{for } \Delta t \geq t_{\mathrm{pred},0.1} \quad (2) \end{cases}$$

とする

(1) 誤差が相対的に無視できない ▶ 補正

(2) 誤差が相対的に無視できる ▶ そのまま



結果 | 密度固定での時間発展

初期値

$$n_{\mathrm{H}} = 10^{12}\ \mathrm{cm}^{-3}$$

$$T = 23\ \mathrm{K}$$

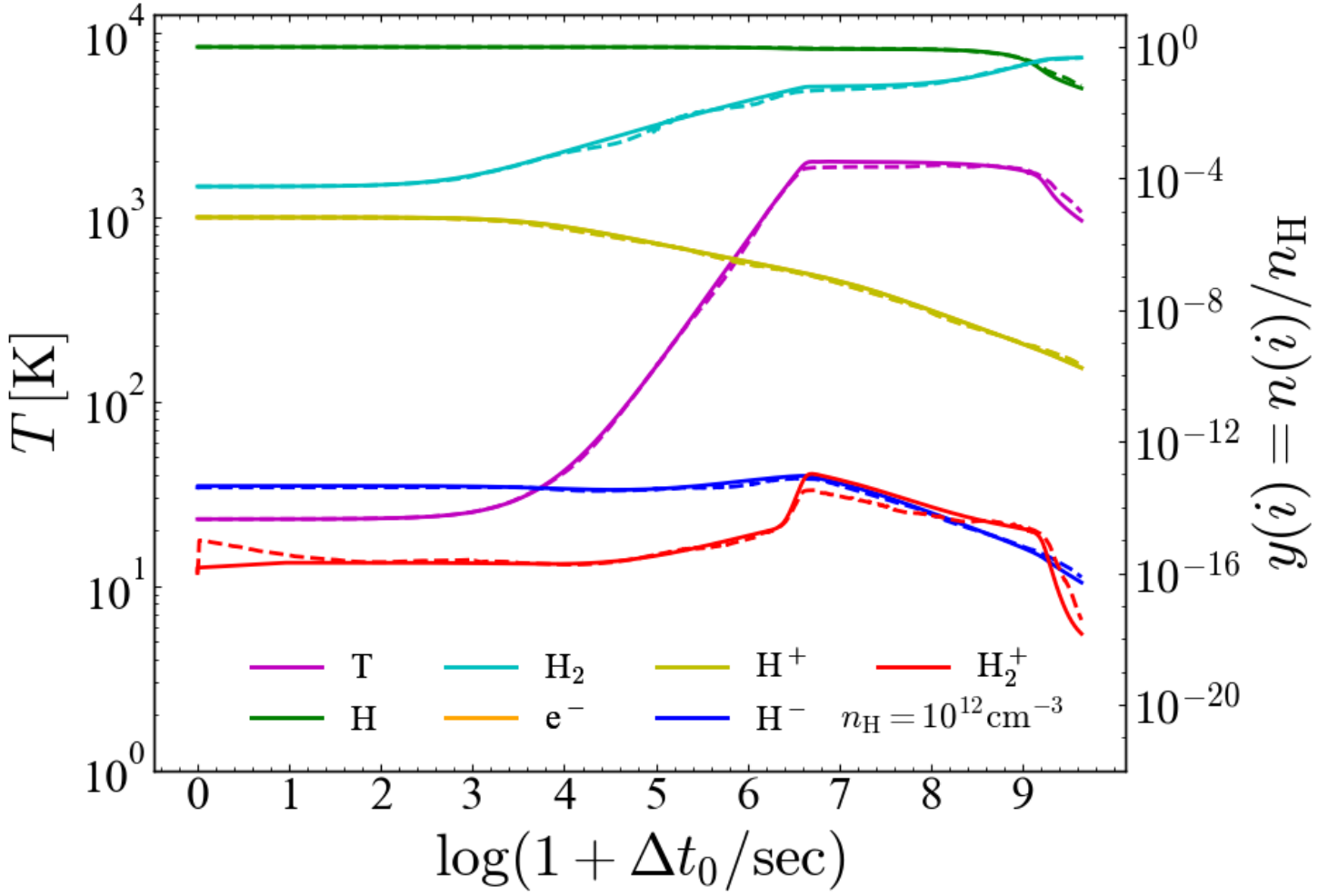
$$y(\mathrm{H2}) = 5.7 \times 10^{-5}$$

$$y(\mathrm{H}^+) = 6.6 \times 10^{-6}$$

$$y(\mathrm{H}^-) = 4.0 \times 10^{-14}$$

$$y(\mathrm{H}_2^+) = 1.0 \times 10^{-16}$$

実線: 数値積分、破線: 補正あり



結果 | 密度固定での時間発展

実線: 数値積分、破線: 補正あり、点線: 補正なし

初期値

$$n_{\mathrm{H}} = 10^{12}\ \mathrm{cm}^{-3}$$

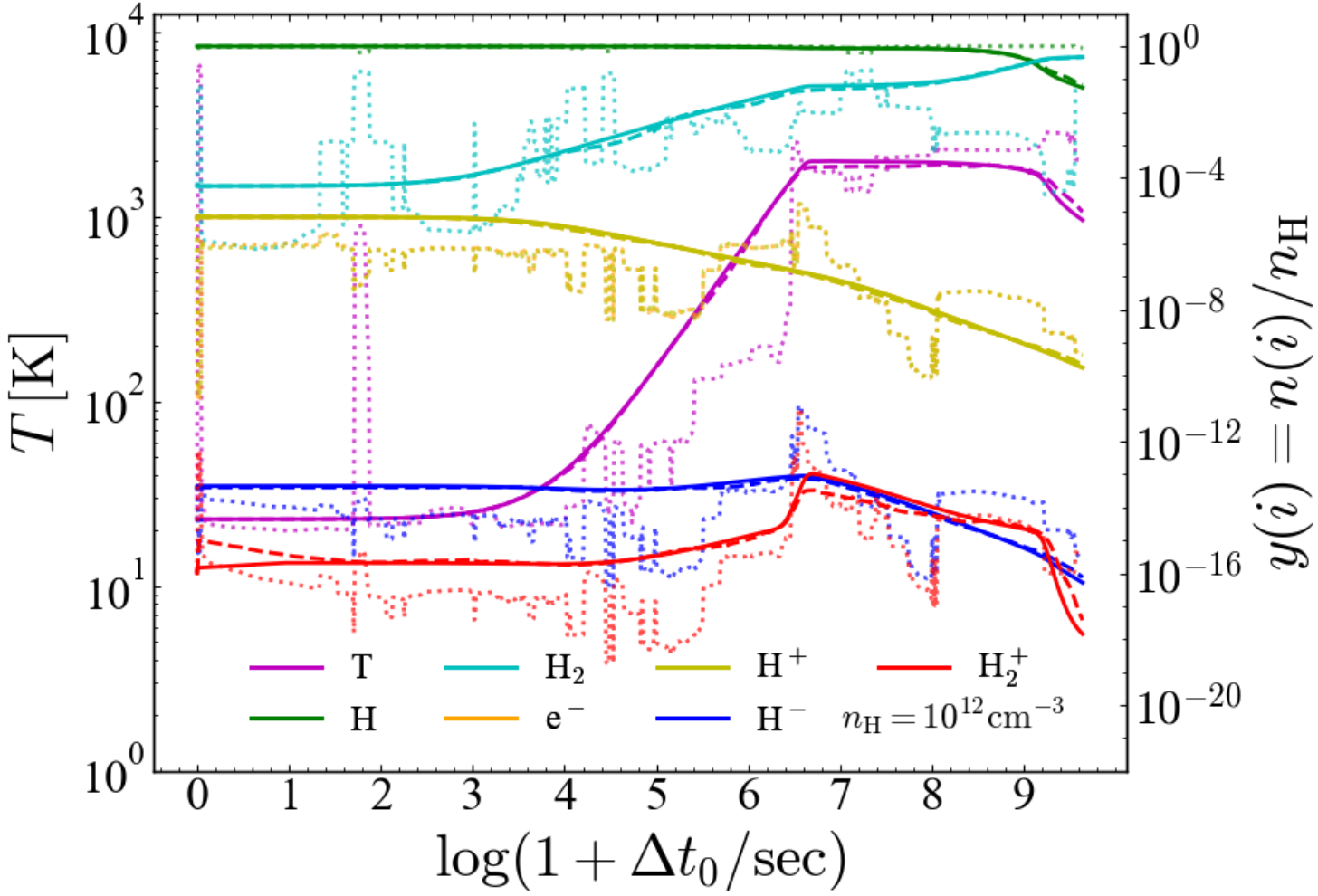
$$T = 23\ \mathrm{K}$$

$$y(\mathrm{H2}) = 5.7 \times 10^{-5}$$

$$y(\mathrm{H}^+) = 6.6 \times 10^{-6}$$

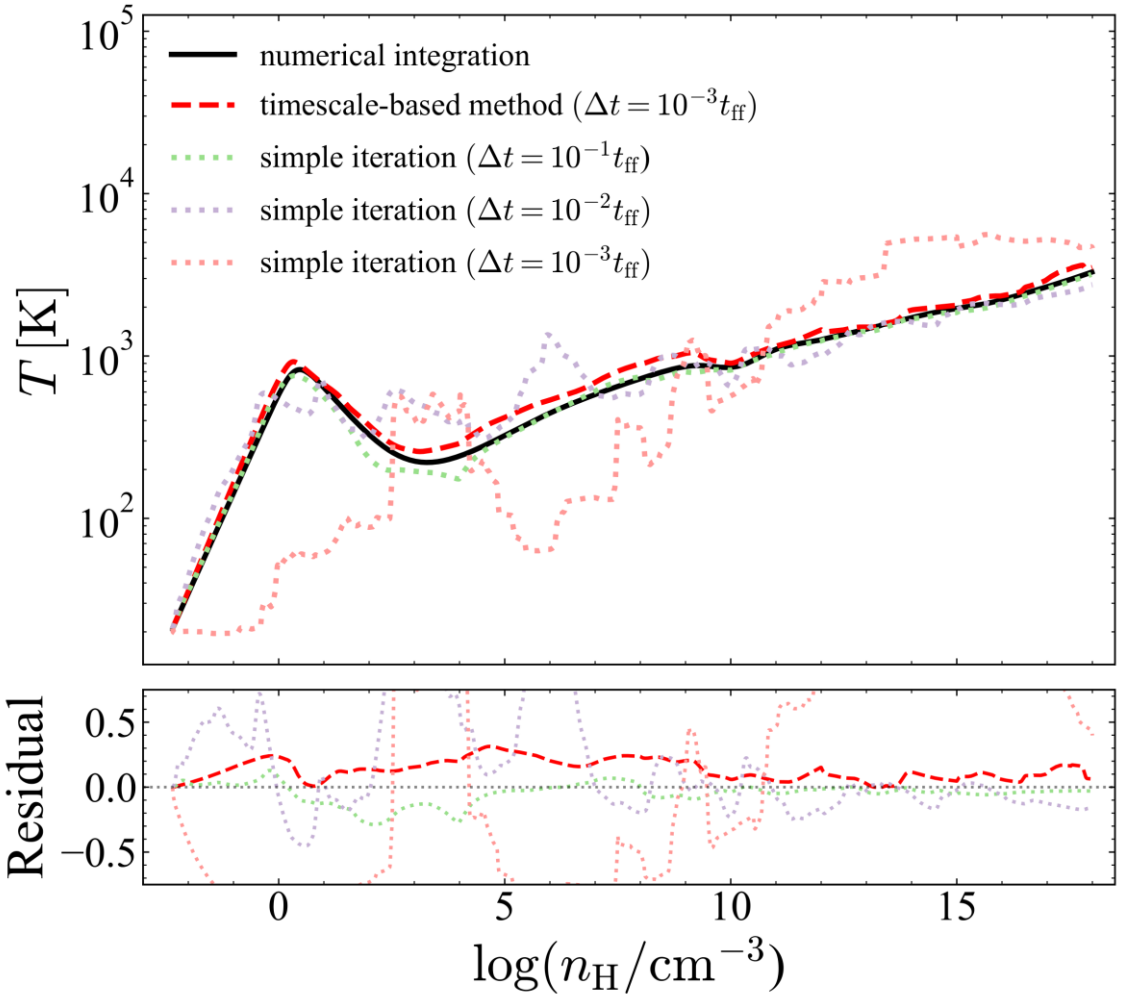
$$y(\mathrm{H}^-) = 4.0 \times 10^{-14}$$

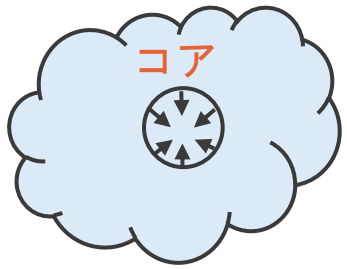
$$y(\mathrm{H}_2^+) = 1.0 \times 10^{-16}$$



結果 | one-zone計算

- 実線: 数値積分、破線: 補正あり、点線: 補正なし Omukai (2001)のような発展





時間発展の方向

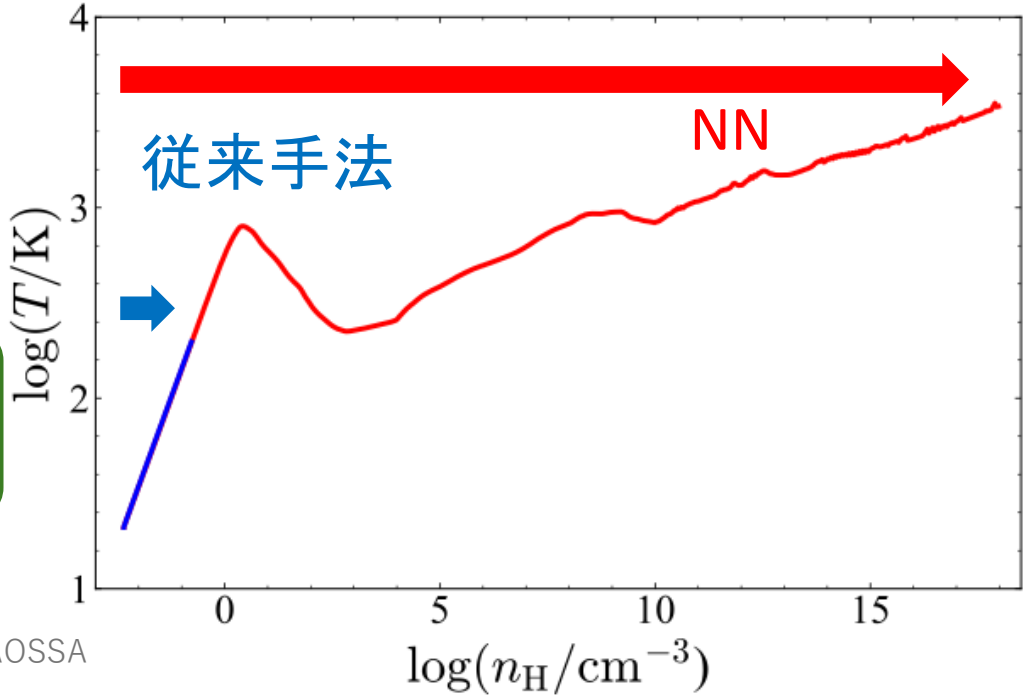
考察 | シミュレーションの計算時間

- 3次元シミュレーションで1コアあたり32^3個のセルをもつと仮定し、全てのセルの成分を一つの大きな行列としてNNに与える

$$\begin{pmatrix} n_{\mathrm{H},1} & T_1 & y(\mathrm{H})_1 & \cdots & y(\mathrm{H}_2^+)_1 \\ n_{\mathrm{H},2} & T_2 & y(\mathrm{H})_2 & \cdots & y(\mathrm{H}_2^+)_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n_{\mathrm{H,n}} & T_n & y(\mathrm{H})_n & \cdots & y(\mathrm{H}_2^+)_n \end{pmatrix}$$

- 機械学習を用いた計算ではGPUを使うことで非常に高速化することが可能である
- 今回は擬似的にone-zone計算を32^3個複製し、計算時間を計測する

同じ計算時間での比較



従来の計算に比べ、約35倍高速化

Conclusion & Future Works

- 初代星形成の熱的・化学的発展を計算するエミュレータを設計した
- 幅広い密度レンジでのエミュレートを可能にした
- タイムステップを細かくしても誤差が蓄積するのを防ぐ方法を開発した
- 作ったモデルにより、熱的・化学的発展を正確に計算することができた
- GPUを用いることで約35倍の高速化を実現した

Future works

- 水素だけでなく、金属を入れたり、輻射の影響を入れたNNを作る
- 3D流体シミュレーションなどさらに複雑な計算に組み込む