Modern Computational Approaches to Early Universe Modeling

Jong-Hyun Yoon
Chungnam National University

AI+HEP in East Asia
Institute for Basic Science
27 Feb 2025

Table of Contents

Part 1

Minimal Cosmological Models in Inflationary Universe

- Inflationary epoch (< 10^-32 s)

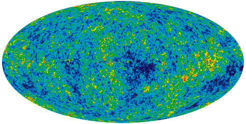
- Solution to Horizon & Flatness problem

- Inflaton?

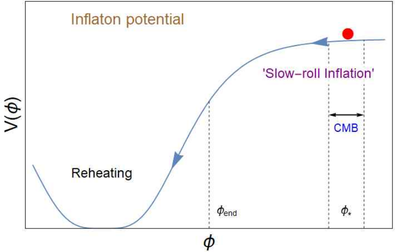
- Real scalar field
- Homogeneity & Inhomogeneity

→ Inflationary Cosmology

(1970~1980s, Alexei Starobinsky, Alan Guth, Paul Steinhardt, and Andrei Linde)



Nine-year Wilkinson Microwave Anisotropy Probe heat map of temperature fluctuations in the CMB

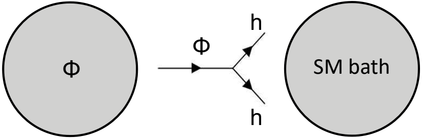


Part 1

Minimal Cosmological Models in Inflationary Universe

- Reheating (Inflaton → SM bath)

(little known: a few MeV < T_R < 10^13 GeV)



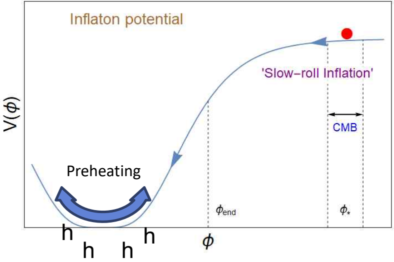
- Simplest reheating model

Inflaton quanta → Higgs

- However, the inflaton field oscillates around the minimum of the potential with large field values

→ Turbulent/non-pert. effects

→ Preheating



Part 1

Minimal Cosmological Models in Inflationary Universe

- While Inflaton → SM (reheating the universe) in the long run,

- DM is produced during preheating:

Inflaton=DM
Inflaton-DM scattering
Inflaton F.O., decay to DM
Inflaton-DM non-renormalizable couplings
Inflaton-DM via gravity

Part 1

Minimal Cosmological Models in Inflationary Universe

- While Inflaton → SM (reheating the universe) in the long run,

- DM is produced during preheating:

Inflaton=DM

Inflaton-DM scattering

Inflaton F.O., decay to DM

Inflaton-DM non-renormalizable couplings

$\boxed{\text{Inflaton-DM via gravity}}$

O. Lebedev, T. Solomko, and **J.-H. Yoon, "Dark matter production via a non-minimal coupling to gravity," JCAP, vol. 02, p. 035, 2023.**

Part 1

Minimal Cosmological Models in Inflationary Universe

- Inflaton-DM via gravity

ξ: coefficient
R: Ricci scalar
Φ: Inflaton field
s: scalar DM

- Non-minimal coupling to gravity

$$\mathcal{S} = \int d^4x\sqrt{-g}\left(\frac{1}{2}M_{\mathrm{Pl}}^2 R \boxed{- \frac{1}{2}\xi R\, s^2} - \frac{1}{2}g^{\mu\nu}\partial_\mu s\,\partial_\nu s - \frac{1}{2}g^{\mu\nu}\partial_\mu\phi\,\partial_\nu\phi - V\right)$$

- R is effectively dominated by Φ, so DM can interact with Φ via

$$R = -\frac{1}{M_{\mathrm{Pl}}^2}T^\mu_\mu$$

O. Lebedev, T. Solomko, and **J.-H. Yoon, "Dark matter production via a non-minimal coupling to gravity," JCAP, vol. 02, p. 035, 2023.**

Part 1

Minimal Cosmological Models in Inflationary Universe

- Equation of motion in momentum space

$$\ddot{Y}_k + \left(k^2 \boxed{+\,\xi R a^2} - \frac{\ddot{a}}{a}\right) Y_k = 0$$

$$Y_k \equiv a\, s_k \qquad dt = a\, d\tau$$

a: scale factor

k: comoving momentum

- Analytic Methods

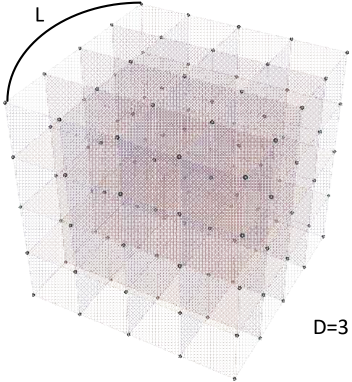
- Boundary Matching, Stokes Phenomenon, etc.

- Resonance Structures (Parametric, Tachyonic, etc.)

- For large ξ, we treat the system semi-classically and solve it numerically to take into account non-perturbative effects

Part 2

Numerical Approaches with High-Performance Computing



$$k_{min} = \frac{2\pi}{L} \qquad k_{max} = k_{min} \times \frac{\sqrt{D}}{2} N$$

- Equations of Motion for Particle Production

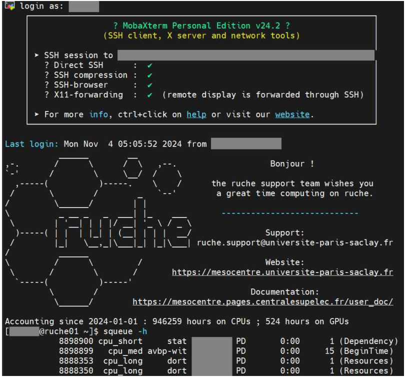
$$\boxed{\begin{aligned} \ddot{f} + 3\frac{\dot{a}}{a}\dot{f} - \frac{1}{a^2}\nabla^2 f + \frac{\partial V}{\partial f} &= 0 \\ \ddot{a} &= -\frac{4\pi a}{3}(\rho + 3p) \\ \left(\frac{\dot{a}}{a}\right)^2 &= \frac{8\pi}{3}\rho \end{aligned}}$$

$$\rho = T + G + V \ ; \ p = T - \frac{1}{3}G - V$$

$$T = \frac{1}{2}\dot{f}^2 \ ; \ G = \frac{1}{2a^2}|\nabla f|^2 \ .$$

Part 2

Numerical Approaches with High-Performance Computing

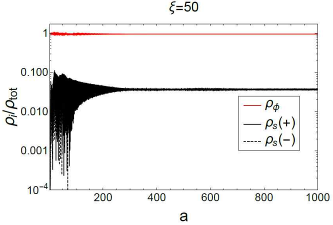


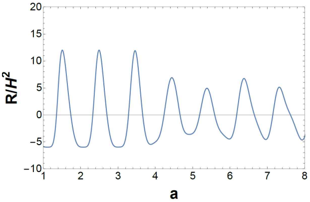
Hardware (ruche cluster at Paris-Saclay University)

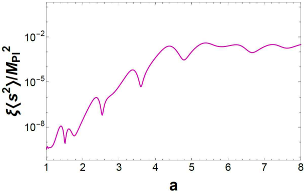
Software (CosmoLattice)

Numerical Approaches with High-Performance Computing

- CosmoLattice customized for NMC







- Energy distribution, R breakdown, resonant production, etc.

- Simulations provide intuitive insights into events in the early universe

Part 2

Numerical Approaches with High-Performance Computing

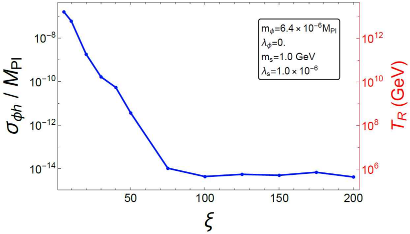
- DM relic abundance (conserved since reheating)

$$Y = \frac{n}{s_{\rm SM}} \quad , \quad s_{\rm SM} = \frac{2\pi^2}{45} g_{*s} T^3$$

$$Y_\infty = 4.4 \times 10^{-10} \left(\frac{\rm GeV}{m_s}\right)$$

- Reheating via inflaton decay into Higgs

$$H_R \simeq \Gamma_{\phi\to hh} \ , \quad \Gamma_{\phi\to hh} = \frac{\sigma^2_{\phi h}}{8\pi m_\phi} \qquad H_R = \sqrt{\frac{\pi^2 g_*}{90}} \frac{T_R^2}{M_{\rm Pl}}$$



T_R: Reheating temperature

- Early DM production can explain the relic abundance today

Part 2

Numerical Approaches with High-Performance Computing

- Intense Particle Production → GWB production
- BSM in the early universe: Phase transition, Sterile neutrino, Axion Inflation, Dark energy, Quantum gravity, etc.
- Thermalization

Part 2

Numerical Approaches with High-Performance Computing

- Simulating the hot universe is very challenging

- Timescale: preheating << reheating

- Limited momentum window $k_{min} = \frac{2\pi}{L}$ $k_{max} = k_{min} \times \frac{\sqrt{D}}{2} N$

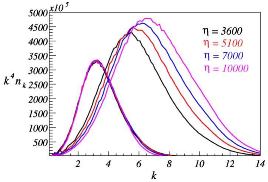
Part 3

Beyond Lattice Simulations: Integrating Deep Learning

- Late-time preheating dynamics exhibits a universal form:

Self-similar evolution of self- or gauge interacting field

→ Implies patterns and trends, which are what DL is all about



Distribution of Φ field in Φ^4 model

R. Micha and I. Tkachev, "Turbulent thermalization," arXiv:hep-ph/0403101

Part 3

Beyond Lattice Simulations: Integrating Deep Learning

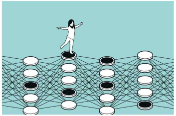
The Nobel Prize in Physics 2024

They used physics to find patterns in information

Machine learning has long been important for research, including the sorting and analysis of vast amounts of data. John Hopfield and Geoffrey Hinton used tools from physics to construct methods that helped lay the foundation for today's powerful machine learning. Machine learning based on artificial neural networks is currently revolutionising science, engineering and daily life.

Related articles

Press release
Popular information: They used physics to find patterns in information
Scientific background: "for foundational discoveries and inventions that enable machine learning with artificial neural networks"



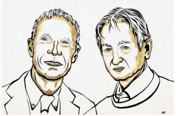
© Johan Jarnestad/The Royal Swedish Academy of Sciences

Nobel Prize in Physics

The 2024 physics laureates

The Nobel Prize in Physics 2024 was awarded to John J. Hopfield and Geoffrey E. Hinton "for foundational discoveries and inventions that enable machine learning with artificial neural networks."

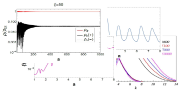
Hopfield created a structure that can store and reconstruct information. Hinton invented a method that can independently discover properties in data and which has become important for the large neural networks now in use.



Ill. Niklas Elmehed © Nobel Prize Outreach

https://www.nobelprize.org/prizes/physics/

- Simulations generate data that can be analyzed by Deep Learning



→ ?

J.-H. Yoon, S.Clery, M.Gross, Y.Mambrini, **"Preheating with deep learning,"** *JCAP*, vol. 08, p. 031, 2024. [arXiv:hep-ph/2405.08901]

- LatticeQCD, CMB, LHC, DM Exp., etc. wherever we have data

Part 3

Beyond Lattice Simulations: Integrating Deep Learning

- CNN (Convolutional Neural Network)

- Efficient at capturing ‘spatial’ hierarchies in data

- LSTM (Long Short-Term Memory)

- Effective at capturing ‘temporal’ dependencies

→ CNN-LSTM time series analysis

(input/output=particle distribution function, f[k, t])

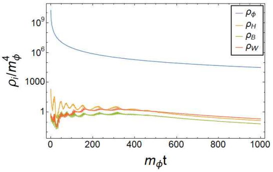
Part 3

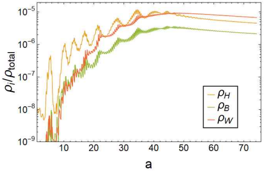
Beyond Lattice Simulations: Integrating Deep Learning

- Implementation
- Preheating model involving Higgs

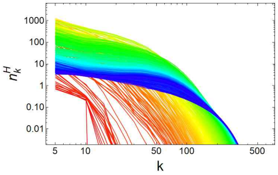
→ Minimal reheating scenario + self- and gauge interaction

$$\Delta V = \frac{1}{2} m_\phi^2 \phi^2 + \frac{1}{4}\lambda_\phi \phi^4 + \frac{1}{2}\lambda_{\phi h} \phi^2 H^\dagger H + \sigma_{\phi h} \phi H^\dagger H - m_h^2 H^\dagger H + \lambda_h (H^\dagger H)^2$$





Energy distributions over time/scale factor



Occupation number of Higgs ~ distribution function (red to blue over time)

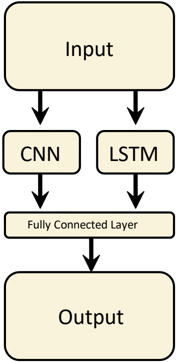
Part 3

Beyond Lattice Simulations: Integrating Deep Learning

```
import torch
import torch.nn as nn
import numpy as np
import matplotlib.pyplot as plt
```

Import PyTorch library for deep learnning

```
class LSTMModel(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, num_classes, seq_length, dropout, dropout2, dropout3):
        super(LSTMModel, self).__init__()
        self.cnn = nn.Sequential(
            nn.Conv1d(in_channels=num_classes, out_channels=15, kernel_size=5, stride=2, padding=1),
            nn.ReLU(),
            nn.Dropout(dropout2),
            nn.MaxPool1d(kernel_size=15, stride=5),
            nn.Flatten(),
            nn.Linear(in_features=30, out_features=20),
            nn.ReLU(),
            nn.Dropout(dropout3)
        )
        self.lstm = nn.LSTM(input_size=input_size, hidden_size=hidden_size, num_layers=num_layers, batch_first=True, bidirectional=False)
        self.fc_lstm = nn.Linear(hidden_size, 15)
        self.fc = nn.Linear(20+15, num_classes)
        self.dropout = nn.Dropout(dropout)
        self.dropoutcnn = nn.Dropout(dropout2)
        self.NLinear = nn.Linear(50, 50)

    def forward(self, x):
        x_cnn = x.permute(0, 2, 1)
        out_cnn = self.cnn(x_cnn)

        out_lstm, _ = self.lstm(x)
        out_lstm = self.dropout(out_lstm)
        out_lstm = self.fc_lstm(out_lstm[:, -1, :])
        out = torch.cat([out_cnn, out_lstm], dim=1)
        out = self.fc(out)
        return out
```



Different hierarchies can be assigned to the CNN and LSTM. This model features a parallel structure.

Part 3

Beyond Lattice Simulations: Integrating Deep Learning

```
for epoch in range(num_epochs):
    model.train()
    optimizer.zero_grad()
    outputs = model(x_train_tensor)
    loss = criterion(outputs, y_train_tensor)
    loss.backward()
    optimizer.step()
    train_losses.append(loss.item())


    val_loss = evaluate_model(model, criterion, x_test_tensor2, y_test_tensor2)
    val_losses.append(val_loss)
    loss_ratio = val_loss/train_losses[-1]
    if (epoch+1) % 1000 == 0:
        print(f'Epoch [{epoch+1}/{num_epochs}], Train Loss: {train_losses[-1]:.3e}, Validation Loss: {val_loss:.3e}', flush=True)


    if val_loss < best_loss:
        best_loss = val_loss
        no_improvement_count = 0
        best_epoch = epoch
        if best_epoch != 0:
            torch.save(model.module.state_dict(), model_path)
    else:
        no_improvement_count += 1

    if no_improvement_count == patience and save_count ==0:
        print(f"Early stopping at Epoch {epoch+1}! No improvement in validation loss.")
        torch.save(model.module.state_dict(), model_path)
        save_count+=1
        print(' Model saved', flush=True)


        continue

    scheduler.step()
```

The model performs “Supervised Learning” using the training data.

At each epoch, the model makes predictions and compares them with test data

To prevent overfitting, we perform early stopping when the model achieves the best predictive power on test data

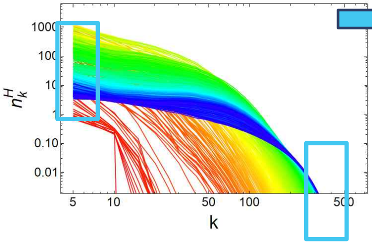
```
print('Forecasting...', flush=True)
model.eval()
with torch.no_grad():
    print(x_test_tensor.shape, flush=True)
    padtensor = x_test_tensor[-1].clone().unsqueeze(0)
    print(x_test_tensor[-seq_length][0][0], flush=True)
    for i in range(n_future):
        outputs_test = model(x_test_tensor)
        for j in range(seq_length):
            padtensor[0][j] = outputs_test[n_train- seq_length- seq_length- window_size+i+j]
        x_test_tensor = torch.cat((x_test_tensor, padtensor), dim=0)

    outputs_test = model(x_test_tensor)
print('Done', flush=True)

outputs_test2 = model(x_test_tensor2)
```

Predictions outside the data range, and the further predictions based on them, require extra caution.

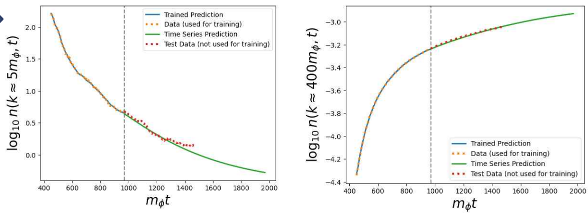
Part 3

Beyond Lattice Simulations: Integrating Deep Learning

- Training DL model



15 k-modes used for training the DL model
(only two of them are represented here)



CNN-LSTM time series analysis

Managed to extend simulation outcomes with DL

→ Once trained, the DL model's predictions are almost instantaneous

→ Successfully laid the groundwork for future developments

Conclusions

- We learned about (p)reheating and minimal cosmological models (e.g. DM via gravity)

→ Preheating effects are often unavoidable and require numerical approaches

- Simulating the early universe with HPC is interesting and useful

→ Intuitive insights, GW search, DM production, BSM physics, Reheating, etc.

- Deep learning can be applied to late-time self-similar systems in the early universe

→ It marks the first step toward simulating the entire history of thermal universe