ISSUE 47 OCTOBER 2024

GAṆANAM (गणनम्)

HIGH PERFORMANCE *COMPUTING* NEWSLETTER
INSTITUTE FOR PLASMA RESEARCH, INDIA

INSIDE THIS ISSUE
(4 Pages)

A Simple Implementation of Deep Learning methods in Particle Simulation

Sagar Choudhary (Research Scholar, PDTSD, IPR)
Email: sagar.choudhary@ipr.res.in

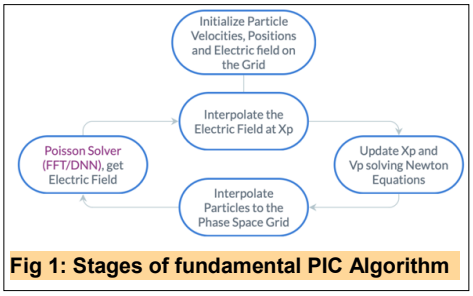


Fig 1: Stages of fundamental PIC Algorithm

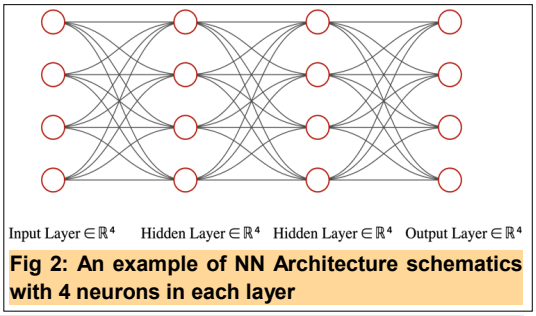


Fig 2: An example of NN Architecture schematics with 4 neurons in each layer

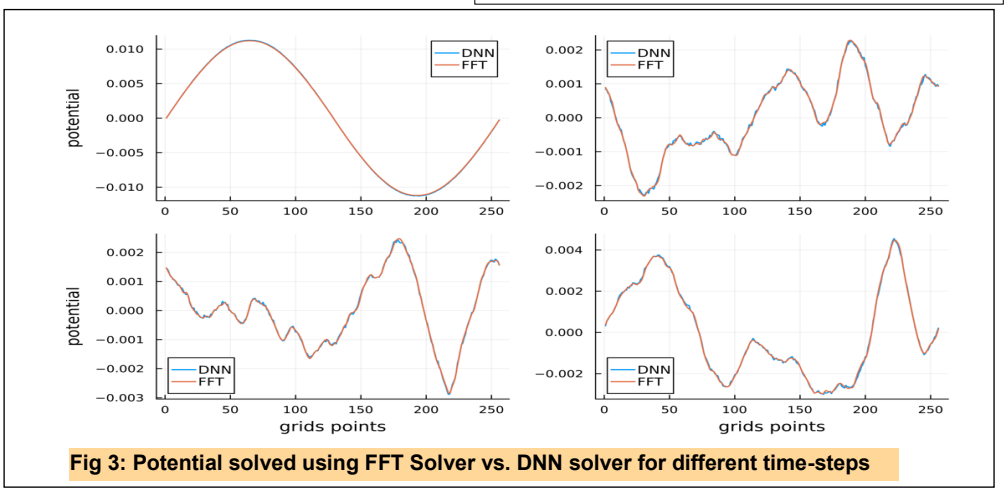


Fig 3: Potential solved using FFT Solver vs. DNN solver for different time-steps

The Particle-in-Cell (PIC) method has long served as a foundational tool in computational plasma physics, with various formulations being developed and refined over time to accommodate the specific requirements of different physical systems. Its versatility and robustness have made it indispensable for simulating plasma behavior in numerous plasma systems. The fundamental PIC algorithm is composed of an initialization phase, followed by a series of iterative computational steps, which are repeated cyclically. These key stages are illustrated in the figure 1, providing a schematic overview of the process.

Although Machine Learning (ML) techniques have been integrated into Particle-in-Cell (PIC) formulations in recent years, as demonstrated by prior works [1,2], the application of Deep Learning (DL) methods is relatively new and has only just begun to gain attention [3]. To date, DL-based approaches remain largely under-explored within this domain, leaving significant potential for further investigation. In the present study, we introduce a Deep Learning-based Poisson solver within the framework of a PIC simulation, aiming to explore the phenomenon of Landau damping. A comparative analysis is conducted between the performance of this DL-based solver and that of traditional simulation techniques, offering new insights into the capabilities and limitations of both approaches.

A Multilayer Perceptron (MLP) network, comprising two hidden layers, has been trained using data from a traditional Particle-in-Cell (PIC) simulation, specifically focusing on the density and potential fields over the course of 100,000 timesteps. Each hidden layer consists of 256 neurons, with the hyperbolic tangent (tanh) function employed as the activation mechanism, facilitating non-linear, bounded transformations within the network. The training process utilizes the Adam optimizer, which is recognized for its efficiency in handling sparse gradients, and a batch size of 100 is maintained to ensure stable learning. A learning rate of 5e-05 is chosen to balance convergence speed and stability. The mean squared error (MSE) is selected as the loss function to quantify the discrepancy between the predicted and actual values, guiding the model's performance improvements during training. Convergence between the training and validation loss history is obtained within 20 epochs of training pertaining to the simplistic nature of the data .

In order to carry out a thorough comparative study, the well-known Landau damping problem has been selected as the primary test case. The system is initialized at time $t = 0$ using a standard distribution function, which serves as the initial condition for the simulation. This setup allows for a direct evaluation of the performance and accuracy of the proposed Deep Learning-based approach in comparison to the traditional methods employed in Particle-in-Cell simulations.

$$f(x,v,t=0) = f_0(1 + \alpha sin(kx))$$

where f_0 is maxwellian distribution function.

The potentials computed using both the Fast Fourier Transform (FFT) solver and the Deep Neural Network (DNN) solver are compared first, with the results presented in Figure 3. The comparison shows that the potentials calculated by the DNN solver closely align with those obtained through the FFT solver, demonstrating a high degree of agreement. However, slight discrepancies are observed in the regime of high-frequency variations in the potential, where the DNN solver results deviates minorly from the FFT-based solution.
Also, energy conservation in the Deep Neural Network (DNN)-based Particle-in-Cell (PIC) simulation is not maintained as rigorously as in the traditional PIC approach, but it remains well within the acceptable range for the duration of the simulation. A consistent discrepancy in the form of a slight error is observed between the two methods throughout the entire runtime. It is important to note the nearly constant energy difference between the two implementations over time, indicating that noise accumulation is similar in both cases.

The damping of the mode, particularly within the linear regime of the Landau damping rate, is effectively captured by the Deep Neural Network (DNN) solver, demonstrating a performance nearly equivalent to that of the traditional Fast Fourier Transform (FFT)-based solver. This close agreement between the two methods highlights the capability of the DNN solver in accurately modeling the damping behavior. This comparison is illustrated in Figure 6, highlighting the DNN solver's capability in accurately simulating this critical phenomenon.

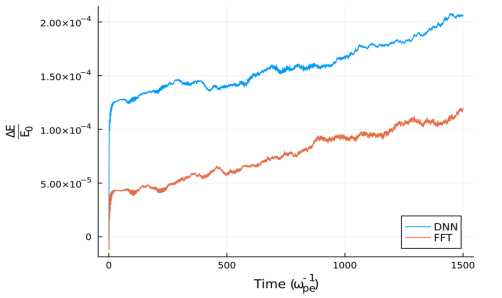


Fig 4: Energy conservation with respect to time

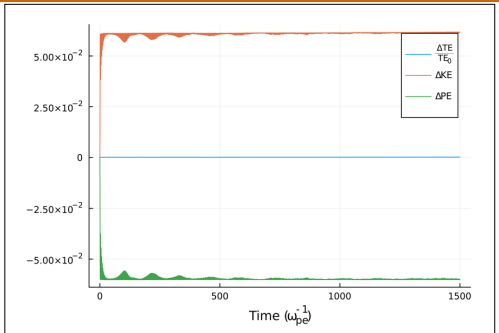


Fig 5: Change in Potential and Kinetic energy with respect to time

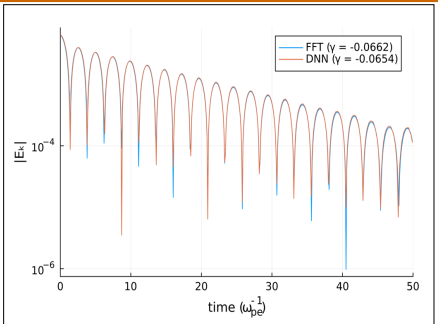


Fig 6: Electric field during the linear regime

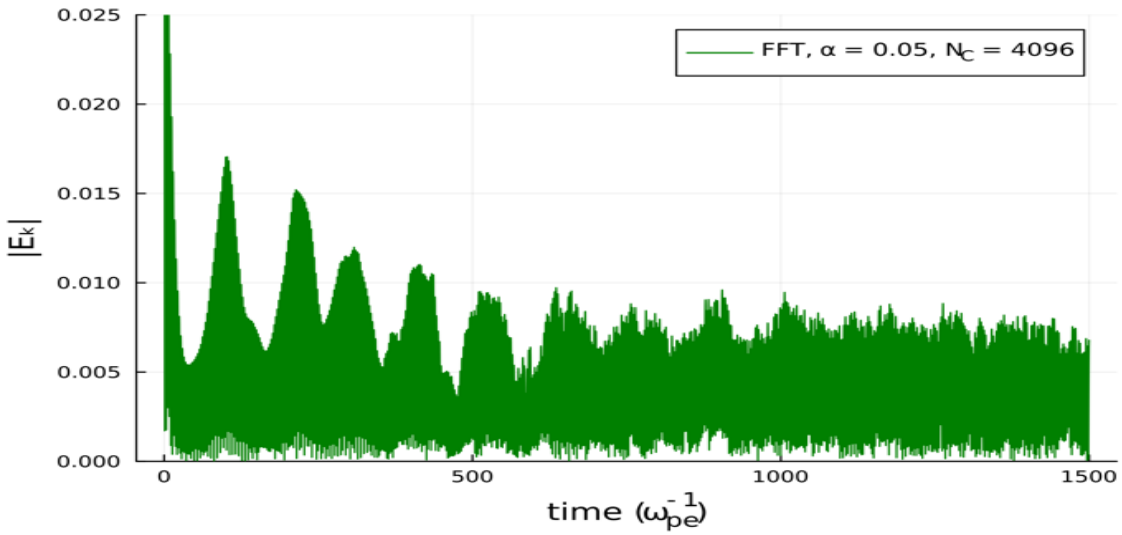


Fig 7: Non-linear evolution of the mode for particle/cell(N_c) = 4096 and perturbation parameter (α) = 0.05

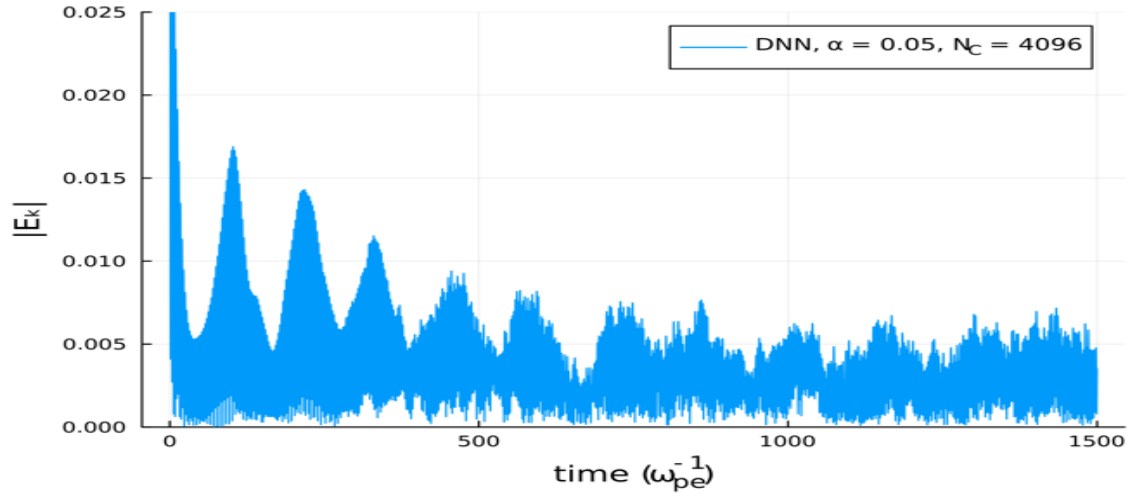


Fig 8: Non-linear evolution of the mode using DNN-based solver for particle/cell(N_c) = 4096 and perturbation parameter (α) = 0.05

The non-linear evolution of the mode is depicted in Figures 7 and 8 for the Fast Fourier Transform (FFT)-based solver and the Deep Neural Network (DNN)-based solver, respectively. Once again, the DNN-based solver successfully captures the non-linear dynamics of the system with notable accuracy. However, it is observed that the influence of numerical noise emerges earlier in the DNN solver compared to the FFT-based solver. Despite this, the overall non-linear evolution is modeled effectively by the DNN-based approach.

DNN based PIC is able to predict the linear damping rate and also captures the non-linear effects very closely to the traditional PIC simulation. A computational performance study is yet to be done. However, DNN solver is a simple prediction/inference step involving a series of matrix-vector multiplications, which can be offloaded to GPUs as well.

Considering that phase space and electric field values at a time-step is similar to the previous time-step, Residual networks (ResNet), might be a better fit to DL-based PIC methods than MLPs and are a part of future work.

References:

[1] R. Kube, et.al., Machine learning accelerated particle-in-cell plasma simulations, arXiv:2110.12444v1, 2021.
[2]C. Badiali, et.al., Machine-learning-based models in particle-in-cell codes for advanced physics extensions, Journal of Plasma Physics Volume 88 , Issue 6 , 2022 , 895880602
[3] A. Xavier, et.al., A Deep Learning-Based Particle-in-Cell Method for Plasma Simulations, arXiv:2107.02232v1, 2021

Combining Multiple Spack Environments for Hybrid Workflows

In high-performance computing (HPC), hybrid workflows often require running different parts of a workload on distinct hardware—such as CPUs for general processing and GPUs for intensive parallel computations. Using Spack, we can create separate environments for CPU and GPU-based dependencies and manage them efficiently. This approach ensures reproducibility and ease of switching between environments when needed.

This article covers example of data analysis and visualization using python with **Numpy (CPU)** for numerical computations and **PyTorch (GPU)** for accelerated data analysis or machine learning. This hybrid workflow is relevant in HPC environments for scientific research, where large datasets needs to be processed.

1) CPU Environment for NumPy

 # Create a spack environment
[user@login1]$ spack env create numpy-cpu
 # Activate the spack environment
[user@login1]$ spack env activate numpy-cpu
 # Add packages and concretize the environment
[user@login1]$ spack add py-numpy
[user@login1]$ spack concretize
 # Install the packages
[user@login1]$ spack install

2) GPU Environment for PyTorch

[*user@login1]$ spack env create pytorch-gpu*
[user@login1]$ spack env activate pytorch-gpu
[user@login1]$ spack add py-torch +cuda cuda_arch=80
[user@login1]$ spack concretize
[user@login1]$ spack install

3) Hybrid Workflow Example: Data Preprocessing and Analysis

The flow of practical guide is to process a large dataset on the CPU using NumPy, then apply a simple neural network model using PyTorch on the GPU to analyze the data, and finally visualize the results. All source and output files are accessible on GitHub here.

Step1: Data Preprocessing on CPU (NumPy)
NumPy is used to preprocess the dataset on the CPU, which is suitable for tasks like data normalization, shuffling, or splitting into training and testing sets.
[user@login1]$ spack env activate numpy-cpu
[user@login1]$ python preprocess.py
[user@login1]$ spack deactivate

Two Files will be created after successful execution of the program
1) labels.npy 2) data.npy

Step 2: Data Analysis on GPU (PyTorch)
PyTorch is used to perform analysis on GPU. Here, a simple neural network is used to classify the data.
[user@login1]$ spack env activate pytorch-gpu
[user@login1]$ python analysis.py
[user@login1]$ spack deactivate

A model will be trained and saved after successful execution of the code

ANTYA UPDATES AND NEWS

1. New Packages/ Applications Installed

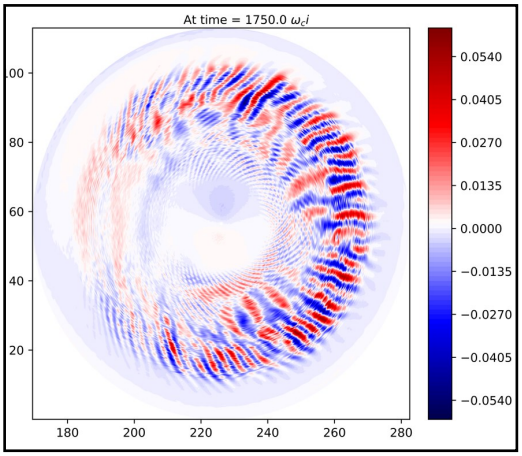
=> New modules have been installed in ANTYA

module load anaconda/2024

module load cmake/3.22

To check the list of available modules
$ module avail –l

HPC PICTURE OF THE MONTH



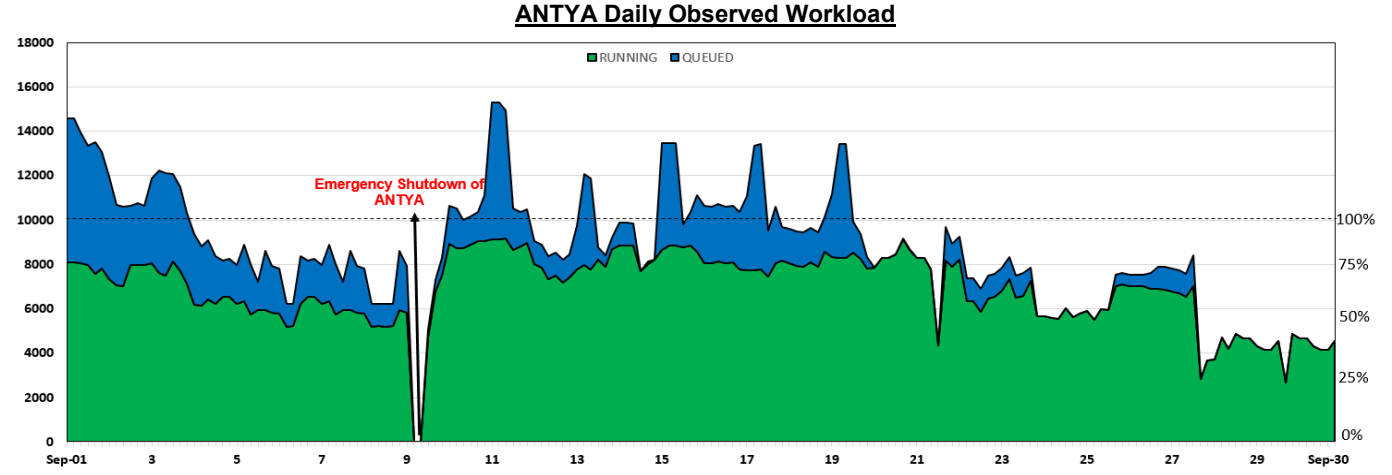
Pic Credit: Sagar Choudhary

Transition phase of the Ubiquitous mode structure from linear to Non-linear regime. This is captured during the non-linear evolution of a global gyro-kinetic electrostatic simulation using the Orb5 code corresponding to ad-hoc equilibria of a large-aspect ratio tokamak.

Benefits of Using Spack Environments

Separation of Dependencies, Ease of Switching and Optimized Resource Utilisation are essential components while executing large workflows. Spack is designed specifically for HPC and scientific software, allowing users to satisfy above mentioned criteria easily.

ANTYA Utilization: SEPTEMBER 2024



Other Recent Work on HPC

The effect of curvature on the dynamics of two interacting plasma blobs in the scrape-off layer region of a tokamak	Souvik Mondal
Magneto-convective fluctuation studies in molten metal MHD duct flow	Srikanta Sahu
Thermal and structural Simulation of HCSB Blanket Concept for Fusion Pilot Plants	Piyush Prajapati
Experimental Study of Double Vortex Structures in a Dusty Plasma Medium	Prasanta Amat
Design and Analysis of the Cryogenic Extruder of Solid Hydrogen	Vishal Gupta
Liquid Metal MHD Flow Analysis in a Square Duct Under Inclined Magnetic Field	Arpita Vipat
Numerical Simulation of Inertial Electrostatic Confinement Fusion Device	Alli Amardas
Current Status of IN-DA Diagnostic Systems for ITER	Sanjeev Kumar Varshney
Data Adequacy for Deep Learning Assisted Microwave Plasma Interaction based Plasma Density Diagnostic	Dr. Pratik Ghosh
DEM-CFD simulations for heat transfer analysis in mixed pebble bed concept of helium cooled Solid Breeder (HCSB) Blanket	Deepak Sharma
Predicting Fluid Turbulence Using Deep Neural Networks	Rajaraman Ganesh
A Global Gyrokinetic Study of Electromagnetic Turbulence in Aditya-U Tokamak	Gopal Krishna M
Runaway Electron Dynamics in a Tokamak Under the Influence of Local Magnetic Field Perturbation	Someswar Dutta
Investigation of Air Drag Reduction Using Dielectric Barrier Discharge Plasma	Jugal Chowdhury
Effects of Magnetic Shaping on Turbulence in ADITYA-U tokamak	Amit Kumar Singh
Laser-cluster interaction in an ambient magnetic field with circularly polarized laser light	Kalyani Swain
3D Computational Fluid Dynamics Analysis using ANSYS of Prototype Ion Extractor Grid for NBI System	Tejendrakumar Patel

ANTYA HPC USERS' STATISTICS–

SEPTEMBER 2024

Total Successful Jobs~ 1208

- CPU Cores — Amit Singh
- GPU Cards — Shishir Biswas
- Walltime — Amit Singh
- Jobs — Tulchhi Ram

Acknowledgement

The HPC Team, Computer Division IPR, would like to thank all Contributors for the current issue of *GANANAM*.

On Demand Online Tutorial Session on HPC Environment for New Users Available
Please send your request to ***hpcteam@ipr.res.in.***

Join the HPC Users Community
hpcusers@ipr.res.in
If you wish to contribute an article in GANANAM, please write to us.

Contact us
HPC Team
Computer Division, IPR
Email: ***hpcteam@ipr.res.in***