

INSIDE THIS ISSUE

(3 Pages)

- Research Highlight: **How Coherent Structures Form in an Overdense Plasma?**
- HPC Article Series: **Python Package Management Using Conda — Part-4: Generating and Sharing Your Conda Environments (Continued to Page 3)**
- ANTYA Updates and News: **OMFIT module Available 2-Day GPU Bootcamp @IPR**
- ANTYA HPC Users' Statistics—**SEPTEMBER**
- Other Recent Work on HPC (Available in IPR Library)

GANANAM (गणनम्)

HIGH PERFORMANCE COMPUTING NEWSLETTER
INSTITUTE FOR PLASMA RESEARCH, INDIA



How Coherent Structures Form in an Overdense Plasma?

Devshree Mandal (PhD Student, Basic Theory and Simulation Division, IPR)
Email: devshree.mandal@ipr.res.in

The intensity of sunlight that we get on Earth is around 0.1413 W/cm^2 , which is enough to burn the paper when it is made to concentrate over a small area of paper. Now imagine, concentrated light of power of 10^{21} Watts in the same 1 cm^2 area. This mind boggling number can actually create havoc on any material it is incident upon but we need to look at their time scales too which are very short (10^{-15} s). We have lasers as concentrated light sources [1-2] with intensity greater than 10^{18} W/cm^2 and when they are incident upon any solid material like glass, they ionize the target instantaneously and turns them into plasma and we find ourselves dealing with solid-density plasma. This solid density plasma under the effect of extreme electromagnetic fields i.e. $E_{\text{laser}} \sim 10^{12-10^{13}} \text{ V/m}$ can behave non-linearly in a couple of femtoseconds. Due to such non-linear interaction, plasma with an intense laser can form coherent structures. Coherent structures are ordered and robust structures spanning several temporal time scales of the system. In this article, we discuss formation of such coherent structures and how HPC has been used to study this complex physical problem.

Main focus of our problem is to understand the dynamics of hot matter at 10^{21} particles/cc. Any numerical scheme will find it very challenging to solve equations of motion for such a huge number of particles self-consistently. These challenges have been partly mitigated by using a well-known technique named as Particle-In-Cell (PIC). In this technique, instead of real particles, pseudo particles are used where charge is smeared out over a finite size. These fat pseudo particles represent many real particles, hence the number of particles in the simulations can be reduced effectively with space being divided into grids. The fields are calculated on the grid points with particles evolution from the force equation using Maxwell's equation. PIC simplifies the problem at the cost of short range collisional effects but fortunately in laser plasma, the dynamics of charge species are of the order of speed of light hence Rutherford collisional cross section reduces considerably and collisional effects take backseat in these interactions. After all this, state of art PIC codes like OSIRIS [3], EPOCH still remain among the expensive simulation codes. **We are using OSIRIS code (OSIRIS 4.0 used under UCLA/IST-IPR OSIRIS Agreement, provided by Dr. Bhavesh G. Patel) to study the intense laser interaction with solid density plasma and typically we simulate our physical problems with 10-100 million particles. A single run on 1600 CPU cores took around 6 hours of walltime.**

"In laser plasma interactions, laser reflects back from critical density. In this study, we simulated how current vortices can travel into denser regions of plasma where powerful laser can not."

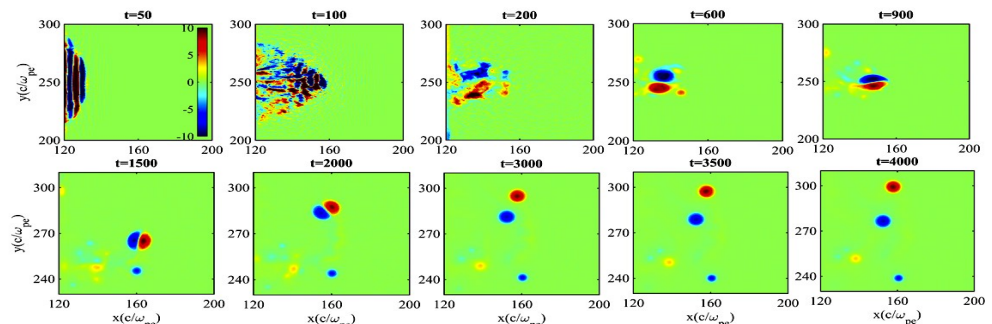


Figure 1: When spatially overlapping currents break into filaments due to various instabilities there's a generation of magnetic field in the perpendicular plane. Here we are presenting the result from PIC simulation, the evolution of B_z as current profiles were in X-Y plane. In this study we have used a p-polarized laser pulse of $1 \mu\text{m}$ wavelength and intensity 10^{21} W/cm^2 and it is made to interact with an overdense inhomogeneous plasma. These snapshots show the generation of B_z inside the plasma ($t=50$), filamentation of current sheets ($t=100$), organization of magnetic fields into a coherent structure ($t=200, 600$), propagation into denser plasma ($t=900$), turning its trajectory and separating into monopole coherent structures ($t=1500-4000$).

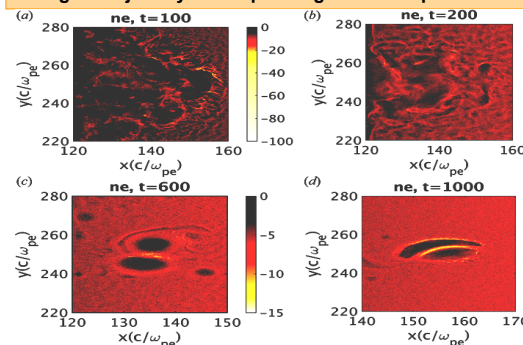


Figure 2: Electron charge density evolution when an intense laser ($I=10^{21} \text{ W/cm}^2$) interacts with an overdense plasma. Such an intense laser will violently disturb the plasma that can be observed in (a) where electrons have been pushed inside the plasma in the shape of laser Gaussian waveform. After most of laser has been reflected back electrons will start rearranging themselves according to self-generated fields. While doing so they can form these structures if local kinetic energy pressure of structure is equivalent to electromagnetic field pressure of that region (progress of this event is observed in (b, c, d)). 't' is normalized by plasma frequency 10^{15} Hz .

When the laser frequency (ω) is less than plasma frequency (i.e. Fundamental frequency of unmagnetized plasma, ω_{pe}), the laser reflects from the critical surface. But in a scenario where a parameter ($a_0 = eE_0/m\omega c$) is greater than unity then plasma dynamics are relativistic and electrons are pushed into the plasma in the manner of forward current while plasma responds by generating a return current inhibiting the propagation of this energetic current profiles. These spatially separated current profiles are unstable and they produce a magnetic field in the perpendicular plane.

Now due to various instabilities, these current profiles get filamented and magnetic field structures merge and form bigger structures as they are left to evolve (Figure 1). If the conditions are appropriate such that these structures can balance particle kinetic energy pressure with electromagnetic field pressure, one can observe these coherent structures to become stable and robust. Additionally, if magnetic field structures of opposite polarities come together within a skin depth, their dynamics will be coupled. The fascinating property that we have observed

about these structures is that they are not formed behind a wake field, formed by a laser but they are formed spontaneously in overdense plasma where the laser has been reflected (Figure 2). And these structures can propagate into those regions of plasma that are not accessible by powerful lasers. We have characterized these coherent structures and found their dependency on plasma inhomogeneity. We have found them to entrap 78% of the electromagnetic energy of the system and that makes them compact sources of energy which can be useful in electromagnetic energy transport in the overdense plasma. This work was carried in collaboration with Ayushi Vashistha at IPR and Prof. Amita Das at IIT Delhi. The published work can be accessed here [4].

References:

- K.A. Tanaka et al. "Current status and highlights of the ELI-NP research program", <https://doi.org/10.1063/1.5093535>.
- Danson, C. N., Hillier, D. I., Hopps, N. W., and Neely, D., High Power Laser Sci. Eng. 3, e3 (2015).
- R. G. Hemker, Particle-in-Cell Modeling of Plasma-Based Accelerators in Two and Three Dimensions, PhD Dissertation UCLA 1999 [arXiv:1503.00276].
- Mandal, D., et al. Spontaneous formation of coherent structures by an intense laser pulse interacting with overdense plasma. Journal of Plasma Physics, 86(6), 905860606. doi:10.1017/S0022377820001361 (2020).

Python Package Management Using Conda – Part-4 Generating and Sharing Your Conda Environments

Rather than installing packages from scratch whenever moving from one system to another system having different operating systems (Linux, Mac, Windows), wouldn't it be great if we can create an OS-agnostic environment file that can be used to manage packages across different platforms? We have already discussed how to create a Conda environment in part-2, issue 10). In this part-4 of the Conda series, we will cover how you can create or auto-generate an environment file for all the packages of a working project which is truly shareable across Linux, Windows, and Mac OS. This environment file can then be shared with your team members/ collaborators to reproduce the environment of the working project without worrying about the python packages and their versions.

What is a Conda Environment File?

A Conda environment file (named as environment.yml) contains the list of dependencies of a python project environment which can be used to re-create the same dependencies across Linux, Windows, and Mac OS. Conda environment files are written in a human-readable language, YAML Ain't Markup Language (a recursive acronym) that uses Python-style indentation.

Why to use Conda Environment File?

When working on a project with a team, the working project environment must be reproducible even if the team members are working on different operating systems. This file can be easily generated and shared with team members. Recreating the Conda environment from an environment file is a "Conda Best Practice".

How to Create Environment File?

There are two methods, first is to generate the environment file from the existing environment automatically, and the other is to manually write the environment file, mentioning the packages and the channels from where packages can be installed.

Example Project: Implementation in ANTYA

Suppose we need to create an environment file for recreating the machine-learning project environment as discussed in issue 10. The packages installed in the project are scipy-1.6.2, and combo-0.1.2 which were installed from conda-forge and pip channels respectively. We will create an environment file from this working project and recreate the environment again for another user using this environment file.

Method-1: Creating the environment file from the existing project environment viz. machine-learning (created in issue 10)

List the main packages installed by user1 in the environment name machine-learning.

```
(machine-learning) [user1@login1 ~]$ conda list | grep
-e combo -e scipy
combo          0.1.2                      pypi_0    pypi
scipy          1.6.2                      py39had2a1c9_1
```

The following command will create an environment.yml file which can be viewed to check about the main packages scipy and combo installed in machine-learning project along with other dependencies from different conda channels.

```
[user1@login1 ~]$ conda env export --name machine-learning --file environment.yml
```

Give this environment.yml file to user2 for replicating the user1 machine-learning environment. Then run the following command from user2 /home directory to create machine-learning environment.

```
[user2@login1 ~]$ conda env create --name machine-learning --file environment.yml
```

Activate the newly created environment and check the available packages in environment

```
[user2@login1 ~]$ conda activate machine-learning
(machine-learning) [user2@login1 ~]$ conda list | grep
-e combo -e scipy
combo          0.1.2                      pypi_0    pypi
scipy          1.6.2                      py39had2a1c9_1
```

For creating an OS agnostic environment file, the following command to create the environment.yml file could be used

```
[user1@login1 ~]$ conda env export --name machine-learning --from-history --file environment.yml
```

ANTYA UPDATES AND NEWS

1. New Packages/Applications Installed

Installed

⇒ **OMFIT module**

now available

module load omfit/unstable

⇒ **USER HOME Area**

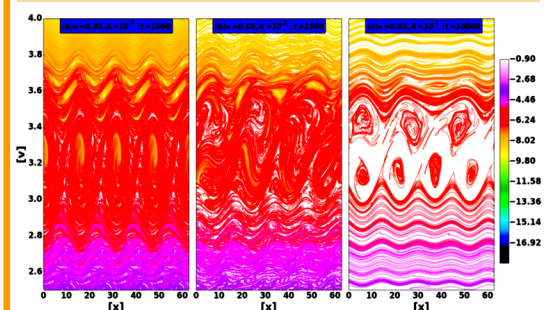
Backed-up

2-Day GPU Bootcamp@IPR

Dates: 20-21 October 2021

[Registration link](#)

Trapped Particle Instability (TPI) in an Inhomogeneous Vlasov Plasmas (HPC Picture of the Month)



Pic Credit: [Sanjeev Kumar Pandey](#)

The figure shows the phase space portrait of electron distribution function at different times. It demonstrates the instability due to trapped particles in a Vlasov-Poisson plasmas in the presence of spatially non-uniform background of ions. [Ref: [Sanjeev Kumar Pandey and Rajaraman Ganesh 2021 Phys. Scr. 96 125616](#)]

The figure is generated using in-house developed **VPPM-OMP1.0** with simulation run on ANTYA which took around 145 hours to complete on 40 CPU cores.



TO LOAD AN APPLICATION FROM THE AVAILABLE MODULES

```
$ module load module_name
```

HPC Article Continues...Generating and Sharing Your Conda Environments

Method-2: Creating the environment file manually for the two main packages required for the existing project environment viz. machine-learning (created in issue 10)

Create the environment.yml for user2 based on the main packages including the versions (scipy and combo) installed by user1 in the environment name machine-learning. The command shows the content of environment.yml file.

```
[user2@login1 ~]$ cat environment.yml
name: machine-learning
dependencies:
  - scipy=1.6.2
  - pip=21.1.2
  - pip:
    - combo==0.1.2
```

Run the following command from user2 /home directory to create machine-learning environment.

```
[user2@login1 ~]$ conda env create --name machine-learning --file environment.yml
```

Activate the newly created environment and check the available packages in environment

```
[user2@login1 ~]$ conda activate machine-learning
(machine-learning) [user2@login1 ~]$ conda list | grep -e combo -e scipy
combo                                0.1.2                                pypi_0    pypi
scipy                                1.6.2                                py39hee8e79c_0    conda-forge
```

If there is requirement to update or add some of the packages for the project, then all you need to do is update the contents of your environment.yml file accordingly and then run the following command

```
[user2@login1 ~]$ conda env update --name machine-learning --file environment.yml
```

The pip packages can also be installed through a separate requirements.txt file having the name of pip packages

```
[user2@login1 ~]$ cat environment.yml
- pip:
  - -r file:requirements.txt
```

The important thing to note about method-1 is that the exported environment file will produce the environment consistently across Linux platforms. However for creating an OS agnostic Conda environment file, Conda currently does not seem to have this feature to automatically write pip installed packages in the OS agnostic Conda environment file created with “—from-history” flag. So the user needs to note down the packages which have been installed using pip channel and write these packages manually (as shown in method-2) in the exported environment file.

Other Recent Work on HPC (Available in IPR Library)

Conceptual Design and Analysis of Prototype Center Stack for Spherical Tokamak based Technologies development

ADITYA KUMAR VERMA

Design and analysis of the primary chamber of plasma pyrolysis system to dispose biomedical waste

DEEPAK SHARMA

Acknowledgement

The HPC Team, Computer Division IPR, would like to thank all Contributors for the current issue of GANANAM. Thanks Mywish for helping in editing the research article.

ANTYA HPC USERS' STATISTICS—SEPTEMBER

◆ Total Successful Jobs — **3751**

◆ Top Users (Cumulative Resources):

⇒ CPU Cores — **Sanjeev Pandey**

⇒ GPU Cards — **Suruj Kalita**

⇒ Walltime — **Mywish Anand**

⇒ Jobs — **Mywish Anand**

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**

Disclaimer: “GANANAM” is IPR’s informal HPC Newsletter to disseminate technical HPC related work performed at IPR from time to time. Responsibility for the correctness of the Scientific Contents including the statements and cited resources lies solely with the Contributors.