Project Proposal for Google Summer of Code 2021

Google Summer of Code

Machine Learning for Turbulent Fluid Dynamics

Coherent States in Pipe Flows

ML 4 SCI

Mentors
Brad Marston, Brown University
Steve Tobias, University of Leeds
Chenyu Zhang, Brown University

Machine Learning for Science

Table of Contents

Contact Information

Name: Amey Varhade

University: Indian Institute of Technology (IIT), Guwahati

Email: ameyvarhade@gmail.com | varhade@iitg.ac.in

Github: yemaedahrav

GitLab: yemaedahrav

Medium: ameyvarhade

LinkedIn : amey-varhade

CV: AmeyVarhadeApril2021

Phone : (+91) 8108836731

Timezone: Indian Standard Time (UTC +05:30)

Location: Mumbai, India

Project Synopsis and Problem Description

The work on modeling Turbulent flow of an idealized barotropic fluid by the dimensionality reduction of Direct statistical simulation (DSS) has been done in place of accumulating statistics by Direct numerical simulation (DNS)for idealized models of barotropic fluid [3].

Turbulence is the chaotic behavior seen in many fluids due to changes in pressure and velocity. The transition from laminar to turbulent flows is dependent on Reynold's number.

Obtaining numerical solutions for turbulent flow is very challenging. Solving turbulent flows using laminar solvers doesn't give steady solutions. Hence, to work with these issues, Reynolds-averaged Navier-Stokes equations are used in appropriate turbulent models. Finding stable solutions or direct numerical simulation require computationally infeasible time.

Here, we aim at developing a statistical theory of transition of fluid to turbulence similar to the above-mentioned one. The discussion and work will be limited to Newtonian and Incompressible fluids.

Problem Approach and Implementation

Preprocessing

From the data available, first, we need to find the solutions of equations[Navier Stokes and Reynolds' averaged Navier Stokes equations(RANS)] through various numerical methods, the Dedalus package uses spectral and pseudospectral methods to solve Partial Differential Equations. Other numerical methods such as the Finite Element Method can be used through different packages such as MATLAB to get the results.

In the case of the dearth of data, techniques for generating dummy data and data augmentation can be used initially for testing purposes.

Casting the data in the appropriate form is needed, as mentioned earlier the two-point statistic of averaged velocity can be used. Various other measures as mentioned in [15] can be explored.

Dimensionality Reduction

The next step after representing the data in an appropriate form is to apply dimensionality reduction techniques. This is particularly needed as in higher dimensional data, space tends to become sparse and hence further treatment and analysis don't yield sound results as *dissimilarity* increases between close enough data points. Also, a lot of redundant computations can be pruned.

The techniques that can be used sequentially are

- Proper Orthogonal Decomposition(POD)[7]
- Principal Component Analysis(PCA)[10], kernel PCA, sparse PCA.
- Reconstruction using Singular Value Decomposition(SVD)[8] *[generalization of eigendecomposition for square matrices.]*
- Autoencoder Neural Networks, using autoencoders is particularly helpful over PCA since it can learn non-linear feature combinations well which PCA fails to do so.

  Various architectures such as Folded Neural Networks [19], Concrete Autoencoders [20], etc can be explored.

Conclusion

Comparing the results through those obtained by DNS or other methods as the base, we can verify the efficiency of the dimensionality reduction techniques. The entire work can be consolidated in as a specific mathematical describing the transition of fluid from laminar to turbulent flow.

The entire work will be summarised and appropriately documented.

Project Deliverables

The following will be the project deliverables that I am aiming to complete during my GSoC

- Overview of the existing literature and understanding of the equations and the issues in solving them, existing techniques, methods, and models.
- Generating the 4-dimensional data, including the vector velocity and scalar pressure $\vec{V}(\vec{r},t)$ $P(\vec{r},t)$ from the simulations of the Navier Stokes equations on Dedalus.
- Computation of the two-point statistic $\langle v_i(\vec{r}_1, t)\ v_j(\vec{r}_2, t)\rangle$ and other plausible data formats from the simulated data generated in the previous step.
- Generating the dimension reduced data from the data format obtained in the previous step by using various dimensionality reduction techniques.
- Analyzing the results, comparison with previous methods. Summarizing the entire work in a mathematical model describing the transition of flow to turbulent.

Evaluation Task

I had completed the evaluation tasks mentioned in the stipulated deadline. The Jupyter Notebook has been commented adequately.

As mentioned, I have generated the symmetric random matrices using properties of matrix transposes. I have reconstructed using the varying numbers of eigenvalues and eigenvectors.

The visualizations clearly show how reconstruction is lossy but still close enough to the original matrix.

The error can also be measured by using various loss functions such as Mean Square Error (MSE), and Kullback Leibler divergence among others.

All the work can be found here [Turbulent Fluid Dynamics evaluation Task]

Detailed Timeline

Phase/Week	Dates	Work Description
Community Bonding		
Community Bonding Period	May17-May23	Discussing an analysis of problems and final goals, familiarizing with the community and organization standards
	May24-May31	Define the project's outcomes more clearly and validate them with mentors. Discussing an analysis of problems and final goals.
	June1-June7	Breaking the goals of my project to several small trackable issues for better analysis of progress and milestones. Finishing the initial setup of the working environment
Phase 1		
Week 1	June7-June13	Reviewing existing literature, specifically the DSS[3] publication. (This would be more or less already done until this time in community bonding and pre GSoC application period)
Week 2	June14-June20	Developing a thorough understanding of the Navier-Stokes equation, the simulation methods. Gaining working proficiency in using the Dedalus package for the simulations to be performed.

Week 3	June21-June27	Generating the data from simulations of Navier Stokes equation using various methods. Generating the two-point statistic and other appropriate data forms and completing the data cleaning and preprocessing.
Week 4	June28-July4	Applying Proper Orthogonal Decomposition(POD), Principal Component Analysis(PCA), Singular Value Decomposition(SVD), and reducing the dimensionality of the data.
Week 5	July5-July11	Implementing various Autoencoder neural network architectures to reduce the dimensionality. Comparison of their results and choosing the appropriate model.
Phase 1 Evaluations July12-July16		
Phase 2		
Week 6	July17-July25	Completing any remaining testing work for dimensionality reductions. Comparing the various techniques used and results obtained in previous weeks
Week 7	July26-Augt1	Tweaking the algorithms in Week 4 and Week 5 to get the optimal results. Finalizing the dimension reduction technique and testing.
Week 8	Aug2-Augt8	Summarizing the entire work done, tabulating and plotting the results obtained.
Week 9	Aug9-Aug15	Completing the documentation, code cleanup, integrating all code.
Week 10	Aug16-Aug22	Buffer Week, Further Work.
Final Evaluations Aug 23 -Aug 30		

About Me

Overview

I am a Pre-Final Year undergraduate pursuing a B.Tech in Computer Science and Engineering from the Indian Institute of Technology, Guwahati (IITG), India.

I have solved over 300 problems on LeetCode[Interviewbit] with C++ as my primary language, majorly using Standard Template Library(STL). Python has been my secondary language. I have a good working experience of Git and GitHub.

Tools, technologies, languages used

C++(STL, Armadillo, mlpack, Boost), Python(Numpy, Pandas, Matplotlib, Scikit-learn), MATLAB, Bash, SQL, Tensorflow(elementary proficiency)

Operating System: Linux(Ubuntu)
Version Control System: Git

I have completed the following relevant coursework as part of my curriculum

1. Data Structures (Theory and Laboratory)
2. Design and Analysis of Algorithms
3. Discrete Mathematics(Graph Theory, Combinatorics)
4. Abstract Algebra and Number Theory
5. Linear Algebra
6. Ordinary Differential Equations
7. Multivariable Calculus and Analysis
8. Optimization
9. Probability and Stochastic Processes
10. Data Mining
11. Machine Learning
12. Deep Learning

I have also completed the following MOOCs from Coursera

1. Machine Learning
2. Neural Networks and Deep Learning
3. Improving Deep Neural Networks
4. Structuring Machine Learning Projects

Experience

Data Mining [Incremental Clustering Algorithms]

I have worked with Prof. Amit Awekar as a part of a group project on developing incremental data mining algorithms. We have proposed a novel technique to the paper on Co-clustering Words and Documents by Prof. Inderjit Dhillon. We have proposed an algorithm that can take new data in and incrementally apply our novel technique combined with the static version and achieve a 3x speedup over the original static algorithm and also displays better results than the static algorithm.

[Work under progress, almost done, might get published]

Code, Datasets, Reports

Mobile Systems [Side Channel attacks through sensor data]

I have also worked with Shirish Singh, Columbia University on inferring privacy threats through side-channel attacks on smartphone devices. We worked on inferring user activity such as walking, running, along with the user location through magnetometer sensor data.

We also explored the possibility of discovering malware on android devices by working with repackaged apps and comparing and magnetic fingerprints.

Side-Channel Attacks

Data Collection Android App

Hackathon

I have previously participated in Smart India Hackathon (SIH) 2020 conducted by the Government of India, and was part of the team selected for the National Stage among 5 teams for the problem statement RK-312. WE made an entire Call Data Visualizer and Internet Protocol Data Visualizer integrating location-based analysis. This was entirely built-in Python (Plotly-Dash, Numpy, Pandas, NetworkX).

SIH 2020, RK-312 Team Konigsberg

Questionnaire

Answers to the questions listed in application guidelines

Have you participated in Google Summer of Code(GSoC) before?

No.

Have you participated in Google Code-in before?

No.

Have you participated in Google Season od Docs(GSoD) before?

No.

Why this project and organization?

I am particularly interested in Algorithms, Optimization, Computational Linear Algebra. I have worked with C++ as my primary programming language for about 3 years now, and Python and MATLAB for about 2 years. I have completed a few Machine Learning, Deep Learning courses, and projects. Hence, I have chosen this project as it aligns with my interests.

What are the potential hurdles you might encounter, and how can you resolve them?

Although I have used Python, MATLAB, and many libraries such as matplotlib, scikit-learn, pandas, numpy, I haven't worked with equation solvers. I have studied ordinary differential equations before. However, using Partial Differential Equations solvers is something I have not worked on previously and hence I would devote additional time in studying it in the application and community bonding period.

Are there any stretch goals you can make if the main project goes smoothly?

I would like to work on theoretically developing other models, and working on solving the Navier Stokes equations in other settings. I would also be interested in working on finding general solutions and explanations of the same, although this would be extremely challenging and advanced work.

How have you decided to communicate with your mentors and how often?

I would be active on my emails, discussing the work with mentors twice a week would be good enough. I am comfortable with Slack, Gitter, Telegram, Meet, Zoom, or any other convenient platform.

I will maintain a shared document where I will update my work on an almost daily basis to communicate the progress to the mentors.

What other time commitments, such as summer courses, other jobs, planned vacations, etc., will you have over the summer?

My university's pre-final year would end by 25 April. I would be able to devote 20-25 hours/week. My university final year commences in August first week. However, there would be no initial work for about a month or so which implies that I would still be able to work for the stipulated time. I have adjusted my timeline accordingly.

Post GSoC/Future Work

Being a part of such a flourishing organization is a great opportunity for one and I have been planning to contribute to the project even after the GSoC Period is over particularly in surface mesh sections and Principal Component Analysis.

However, I might not be able to devote much time from September to December due to my Bachelors' Thesis where I would be working on designing hardware accelerators for machine learning algorithms.

Community Outreach

Under the guidance of mentors, I look forward to contributing to the project. I will also be providing Daily Scrum, Weekly Progress, Blog Posts, and Articles through the communication channels and Blogs.

I will continue the tradition of sharing my GSoC and Open-Source experience in general in my University student clubs primarily and through social media in general.

References

1. ML4SCI Project Ideas
2. Theoretical perspective on turbulent flow
3. Dimensionality Reduction of DSS
4. Dedalus Project
5. Flow of wet water, Feynman
6. Navier Stokes equation
7. Proper Orthogonal Decomposition
8. Singular Value Decomposition
9. Eigen Decomposition
10. PCA-Tutorial
11. Dimension reduction in fluid dynamics equations.
12. Machine Learning accelerated CFD
13. Dedalus, Numerical simulation through spectral methods
14. Numerical methods for Navier Stokes
15. Two-point statistics
16. Deep Learning Fluid Dynamics
17. DiscretizationNet
18. POD in Turbulence
19. Folded Neural Network
20. Concrete Autoencoders