DPD Simulation of a Membrane

Description: This project will work toward the simulation of a cell membrane using a type of molecular dynamics known as Dissipative Particle Dynamics (DPD). In the first week, we will implement a basic molecular dynamics simulator in C++, probably in 2D, to learn about the inner workings of particle-based simulators. In the second week, we will learn about the DPD method and how to implement it in LAMMPS, a DOE-funded software package for molecular dynamics. We will use LAMMPS to simulate a cell membrane and get some nice 3D visuals. Time permitting, we can explore some more involved things like how the membrane moves when subject to a force from polymerization of a rigid polymer network, or how it flexes if the membrane is pinned at certain points. You can see what these sorts of simulations look like in this YouTube video.

Prerequisites: While we will make our basic MD simulator in C++, basic programming background in any language and the ability to Google things are probably sufficient. LAMMPS uses its own scripting language which is very easy to learn and no prior background with LAMMPS is assumed. We will not be focusing on the high performance computing aspects like how to optimize and parallelize code (LAMMPS is already highly optimized), so this project is suited to beginner or intermediate programmers who are more interested in learning how to approach the simulation of some biophysical problems.
Density Functional Theory Calculations of Molecular Orbital Evolution During Chemical Reactions

Description: In this project we will focus on the calculation and visualization of the transformation of molecular orbitals following chemical reaction paths. A brief introduction will be given on background knowledge in chemistry, followed by hands-on tutorials on using the Atomic Simulation Environment (ASE) Python package to perform the calculations. Methods that use ensemble parallelization will then be explored to acceleration the turnaround time and throughput of the calculation.

Prerequisites: There are no formal prerequisites. Python proficiency is a plus but not required.
Uncovering the Patterns Behind Patterns

**Project description:** In this project we will explore localized patterns on the plane. Such patterns appear in a wide variety of physical contexts, which include— but are not limited to!— fluid flows, crime hot spots, buckling problems, vegetation growth and optical systems. Bifurcation diagrams for widely disparate systems have proved to be remarkably similar; see Figure 1.

Various types of patterns which are periodic in one direction and localized in the other, including those termed “rolls”, “spots and stripes”, and “squares,” have been investigated (see, for example, [1], and refer to Figure 2 for visualizations of particular patterns). We
Figure 2: Examples of patterns observed in the Swift–Hohenberg equation on an infinite cylinder. See [1] and [5].

would like to use this time and computing resources to better understand the relationships between these patterns, and, in particular, their connections in parameter space.

As time and interest allow, we may also explore the formation of large localized hexagon patches. Computing resources were specifically identified as a limiting factor in a 2008 study [4]. However, since this time the underlying package capabilities have progressed substantially, and revisiting this problem with new computing resources may enable substantial progress.

**Prerequisites:** While there are no formal prerequisites for participation in this project, some familiarity with differential equations and dynamical systems theory would be helpful. In particular, a basic understanding of bifurcation theory underlies most of the work. I would be happy to review with any interested students lacking this background; Strogatz [9] also provides an accessible and useful introduction. Programming for this project will be conducted in C/C++. Current software is written in MATLAB, so our first task will be converting this software. Depending on our progress we may also use AUTO07p, a Fortran based program for continuation, but no knowledge of Fortran is expected.
References


Computer Simulations on Rocket-Plasma Interactions

- Research Background
  Near-Earth space is filled with a plasma, consisting of a huge number of free moving electrons and ions. The plasma interacts with spacecraft and rockets in space and sometimes causes anomalies on their systems. The interaction of spacecraft or rocket with the plasma environment is a very crucial issue for future space development. The computer simulation based on the Particle-In-Cell method is a very powerful tool to solve the problem.

- Project
  The project focuses on a sounding rocket which is launched to investigate the Earth’s ionospheric region. As a result of interactions of the rocket with the ionospheric plasma, a number of interesting phenomena will take place around the rocket such as rocket charging, plasma sheath/wake formation, and non-uniform potential structures. You may use/improve a pre-existing plasma particle simulator called EMSES to simulate these phenomena. After determining physical and numerical parameters for the EMSES simulations, you will run the program on the FX10 supercomputer owned by Kobe University. Post-processing is also an important aspect of this project; the output data should be processed and visualized with ParaView and a virtual reality system for better understanding of physical phenomena reproduced by the simulations.

- Programming and Computational Skills:
  Basic knowledge about Fortran90, parallelization with Message Passing Interface (MPI), and visualization with ParaView will be helpful, but the project is open also for those who are not familiar with the subjects. Some materials and short courses will be given before/during the school.

- Project Leader: Yohei Miyake (Kobe University, y-miyake@eagle.kobe-u.ac.jp)
Simulations of Geophysical Fluids and Planetary Atmospheres

Brief Introduction:
Atmospheric circulation is one of the important applications of computer simulation. In fact, atmospheric models are used for daily weather prediction and climate prediction of the Earth’s atmosphere. In addition, atmospheric circulation models are used for research of atmospheres of planets, such as Mars.

Project:
The project will work on the simulations of atmospheres of the Earth or Mars by use of a pre-existing codes, DCPAM*, or some geophysical fluids in simple systems by making codes. Some lectures and tutorials will be given during a first week. Followings are plausible topics which participants can select based on one’s experiences and interests.

- Perform simulations of Earth’s or Mars’ atmospheres by use of the DCPAM,
- Make a code for geophysical fluids, such as a two-dimensional turbulence, a shallow water system, and a three-dimensional fluid, and perform simulations,
- Make a tracer transport codes which calculate advection by a meteorological fields, and perform simulations.

Programming language:
Fortran: Experiences on coding Fortran90 program are helpful, but those who are not familiar to Fortran90 are welcome.

Project Leader:
Yoshiyuki O. Takahashi (Department of Planetology, Kobe University, Japan)

* DCPAM: Planetary atmosphere general circulation model developed by members of GFD** Dennou Club. See http://www.gfd-dennou.org/library/dcpam/index.htm.en for more information of the model.
** GFD: Geophysical Fluid Dynamics

Figure. An example of water vapor distribution in the Earth’s atmosphere simulated by the DCPAM