

Course Title: Analysis & Modeling of Bio. Structures

Instructor: Michela Taufer

Weeks: 1 – 7

IMPORTANT: This course has three key modules that are taught by three instructors during the semester. Please note that this document is the syllabus of the first part of this course.

Instructor: Dr. Michela Taufer (Assistant Professor in Computer Science)

Day, Time, and Location (room and building):

Class: Tuesday 9:00AM – 11:50PM (PSCI 314, room 5)

Seminar: Thursday 9:00AM – 10:20PM (PSCI 314, room 5)

Lab: Thursday 10:40AM – 11:50

Office Hours:

Dr. Michela Taufer TBD

Email Address:

Dr. Michela Taufer mtaufer@utep.edu

Campus Office Number:

Dr. Michela Taufer CS 207

Office Phone Number:

Dr. Michela Taufer 6957

Textbooks

Class notes and slides will be distributed through WebCT one week before the class takes place.

The material used in class is based on material from:

- Scientific Computing and Visualization (course given at USC)
http://cacs.usc.edu/education/cs596_f05.html
- High Performance Computing and Simulations (course given at USC)
http://cacs.usc.edu/education/cs653_s06.html
- Molecular dynamics simulation with Charmm:
http://www.ch.embnet.org/MD_tutorial/
- Introduction in C Programming (Universtiy of Leicester)
<http://www.le.ac.uk/cc/tutorials/c/index.html>

The material is also based on material in:

- The Art of Molecular Dynamics Simulation by D.C. Rapaport, Cambridge Ed.

Course Information and web-site

The material related to this course (e.g., syllabus, news, reading material, slides, etc.) are available through WebCT. In WebCT, a mailbox, a public forum for discussions, and the course calendar are also available. Every week, the instructor will update the information on the course in WebCT. Therefore you are encouraged to consult the information available on WebCT on a weekly base.

Prerequisites

Students must have prior programming experience using either C or Perl on Unix/Linux computing systems, plus mathematics coursework through calculus. Experience using and developing scientific codes and mathematics coursework through differential equations is helpful but not required.

Goals

This course is intended to provide student in Bioinformatics with an understanding of molecular simulations, their principals, and their sequential and parallel algorithms. The student will learn the fundamental aspects of coding, compilation and execution. Emphasis is placed on the algorithmic and code components of the simulation and visualization algorithms. The student is encouraged to analyze performance of these algorithms.

Course Description

This first part of the course, presented in this syllabus, provides students with advanced knowledge in modeling techniques that are commonly used to study molecular behaviors in high performance computer simulations. Molecular dynamics simulation algorithms for particles will be implemented for desktop machines and massively parallel, distributed computing platforms. Scientific visualization using OpenGL programming and grid computing for large-scale molecular simulations will be introduced.

Course Calender (tentative schedule)

Week 1: Introduction in C programming (3 hours)

Week 2: Molecular Dynamics (MD) simulations (3 hours)

- Trajectory, coordinates and acceleration
- Newton's equation
- Lennard-Jones Potential
- Discretization, Verlet Discretization
- Energy conservation

Week 3: Advanced Molecular Dynamics

- Charmm force fields
- Solvent representation
- Charmm code and simple examples

Week 4: Introduction in parallel computing (3 hours)

- Parallel architectures
- Introduction in Message Passing Interface (MPI) Programming

Week 5: Parallel Molecular Dynamics simulations (3 hours)

- Load balancing
- Scalability metrics for parallel molecular dynamics
- Divide-and-conquer parallelization, spatial vs. particle vs. force decomposition, data-driven parallelization

Week 6: Visualization tools for molecular systems

- OpenGL library

Visualizing Molecular Dynamics
Open source tools for molecular visualization: VMD

Week 7: Grid scientific computing (3 hours)

Definition of grid and global computing systems

Examples of global computing projects in biophysics: protein prediction, protein folding, and protein-ligand docking.

Grid computing systems and their applications: TeraGrid, Globus

Volunteer computing and desktop grid: BOINC

Guidelines: You are expected to do the reading of the class material **BEFORE** the specified class meeting date. Reading assignments will be announced in class and posted on WebCT. Not all assigned material will be covered in class. It is your responsibility to ask questions in class regarding assigned material that is not fully understood.

Seminar (tentative list of papers)

1. L. Greengard and V. Rokhlin (1987), "A fast algorithm for particle simulations", J. Comput. Phys. 73, 325.
2. M. Shirts and V. S. Pande (2000) "Screen savers of the world unite", Science 290, 1903.
3. J.C. Phillips, et al. (2002), "NAMD: biomolecular simulation on thousands of processors", in Proc. of IEEE/ACM Supercomputing 2002.
4. S. Plimpton (1995), "Fast parallel algorithms for short-range molecular dynamics", J. Comp. Phys. 117, 1.
5. A. Sharma, A. Nakano, R. K. Kalia, P. Vashishta, S. Kodiyalam, P. Miller, W. Zhao, X. Liu, T.J. Campbell, and A. Haas, (2003), "Immersive and interactive exploration of billion-atom systems. Presence: Teleoper." Virtual Environ. 12(1)85-95.
6. C. D. Snow, E. J. Sorin, Y. M. Rhee, and V. S. Pande (2005), "How well can simulation predict protein folding kinetics and thermodynamics?" Annu Rev Biophys Biomol Struct. 34:43-69.
7. D. C. Rapaport (1997), "An introduction to interactive molecular dynamics simulation." Computers in Physics 11:337-347.
8. S. Ogata, et al., (2003), "Scalable and portable implementation of the fast multipole method on parallel computers," Comput. Phys. Commun. 153, 445.
9. M. Tuckerman, B. J. Berne, and G. J. Martyna, J. (1992), "Reversible multiscale molecular dynamics," Chem. Phys. 97, 1990.
10. A. Nakano, R. K. Kalia, and P. Vashishta (1994), "Multiresolution molecular dynamics algorithm for realistic materials modeling on parallel computers," Comput. Phys. Commun. 83, 197.

Important Dates:

First Class Meeting: 01/17/07

Last Day to Drop with a W (*): 03/23/07

Holiday:

- 03/12 – 03/18 (Spring Break - No Classes)
- 03/31 (Cesar Chavez Day - No Classes)

Last Class Meeting of this course part: 03/02/07

Last Class Meeting of the whole semester: 05/03/07

Note: Other important dates will be communicated in class.

(*) Beginning the 1997 Fall semester, students or faculty members may initiate a drop with a

grade of W until the drop deadline (3/18/2005). After that date, students may be dropped only with a grade of F. After that date, grades of W may only be assigned in exceptional circumstances after a written petition from a student and with the approval of the faculty member and the academic dean.

Knowledge and Abilities Required Before the Students Enter the Course

1. Knowledge and Comprehension

- a. Have a basic knowledge of computer architectures, compilers, and operating systems.
- b. Have a basic understanding of mathematics coursework through calculus.

2. Application and Analysis

- a. Use with familiarity compiler/library technologies for sequential codes and UNIX/Linux operating system environments.
- b. Be able to program in C.

3. Synthesis and Evaluation

- a. Be able to understand and present technical work orally.
- b. Write technical documents that are grammatical correct and technically sound.
- c. Be able to summarize technical papers concisely and clearly

Learning Outcomes

1. Knowledge and Comprehension

- a. Develop the understanding of molecular modeling using Molecular Dynamics.
- b. Develop the understanding of basic parallel concepts, i.e., shared-memory and distributed memory architectures
- c. Develop the understanding of basic parallel metrics such as speed-up and efficiency.
- d. Develop the understanding of basic concepts in Message Passing Interface (MPI) for distributed-memory architectures such as point-to-point communication and collective communications, as well as basic and derived data.
- e. Be aware of other existing environments for parallel simulations, e.g., OpenMP, and their differences with MPI programming.
- f. Develop the understanding of visualization tools for molecular dynamics.

2. Application and Analysis

- a. Design and implement simple sequential algorithms for the simulation of molecular behaviors based on Molecular Dynamics.
- b. Design and implement simple parallel algorithms for the simulation of molecular behaviors based on Molecular Dynamics.
- c. Compile, run, and, analyze performance of parallel codes using MPI.
- d. Implement simple algorithms for molecular visualization and combine them to MD algorithms for the visualization of trajectories.

3. Synthesis and Evaluation

- a. Present in a concise and clear way the material learned in the course orally and in writing.
- b. Decide the most effective molecular models based on characteristics of the molecular system, type of information required and data available.
- c. Decide the most suitable techniques for parallelizing based on the computing resources available.
- d. Understand and improve more complex algorithms for the simulation and visualization of molecular systems.

Grade Standards and Criteria

Grading will be based on:

1. **Lab assignments:** 5 programming assignments.
2. **Seminar:** one presentation and several summaries of papers presented by other students in the seminar.
3. **Attendance and active participation to classes, seminars, and lab:** attendance to classes, seminars, and lab activities is mandatory. Signatures will be collected at the beginning of each meeting. Students will also get credits for their active participation to the class discussions.

No final exam is scheduled.

The final grade will be assigned as follows:

Seminars (35%):

- Seminar presentation (One presentation) 15%
- Paper summary (4 paper summary over the semesters) 20%

Lab projects (55%):

- Lab 1-4 (10% each) 40%
- Lab 5 (15% each) 15%

Attendance (10%):

- Attendance and active participation to the course activities 10%

Evaluation Criteria

• **Seminars (35%):**

Your grade for the seminars is so distributed: 15% for your presentation and 20% for your summaries (5% for each summary of the papers that you do not present, whose presentations are given by the other students in the course).

With reference to your presentations: you are required to give one presentation. The list of papers is listed in this syllabus and the papers are available in WebCT. You must consult the instructor for the choice of the papers. Each paper can be presented only once per semester and is assigned to students on a first-come, first-served base.

With reference to the summaries: when you do NOT present a paper, you have to summarize it. The summary is due the day before the presentation takes place and needs to be submitted per e-mail to the instructor. Each summary has to be no longer than three pages (US-Letter, 11pt, single line). Before to summarize the paper, read the paper toughly and make sure that you understand in detail what the authors have achieved and how they achieved it.

1. What is the nice of the research presented in the paper?
2. What is the main contribution(s) of this paper?
3. What is the methodology used in the paper to present the research contribution(s)?
Is the methodology chosen effective to assess the contribution(s)?
4. What are the main results presented in the paper?
5. If you are aware of past, related work against which this paper can be judged, do the authors cite it? Are the citations appropriate? Are they correct?
6. What are the implications of the work?
7. What are the limitations of the work?

Your grade will be reduced of 2% each time you will not be able to submit your summary on time. No credit will be given if plagiarism is detected.

- **Lab Assignments: (45%)**

Lab assignments are programming tasks. Solutions have to be returned on time per e-mail. Please do not send the executable but the source code and any file needed for the compilation/testing. To be graded, solutions should be compilable on the machines in the Bioinformatics lab. The code has to be well-written and commented. The solution should perform what required in the assignments.

- **Attendance: (10%)**

Attending the classes, seminars, and laboratory is mandatory. Students are required to take part to discussions. Your grade will be reduced of 1% each time you will not be able to attend a class/seminar/lab without a justification.

Facilities

Students will get class accounts at the bioinformatics computer laboratory to complete their programming assignments in the laboratory.

Withdrawal and Grade I

Grade I: The grade of I (incomplete) will be given ONLY if you are unable to complete the course due to documented appropriate circumstances beyond your control that develop after the last day to withdraw from the course. Appropriate circumstances include (1) illness and (2) death or crisis in your immediate family. In NO case will an I grade be assigned to avoid a grade of D or F in the course.

Course Withdrawals: If you decide to withdraw from the course, you are responsible for ensuring that all steps are taken to formally withdraw. Do not assume that you will be dropped automatically.

Disciplinary Infractions and Ethical Behavior

Mobile telephones MUST be switched off during the lesson. The use of laptops is also not allowed during the lesson. Discussions related to graded work (i.e., homeworks, midterm exams, quizzes) will take place in in-person meetings scheduled by appointment via e-mail or during office hours.

Students are expected to behave courteously and professionally according to the standards published at http://hoop.utep.edu/Student_Affairs_Chapter_One-HOP.htm. Disciplinary infractions, e.g., exams that represent collaborative effort, will be submitted to the Dean of Students.