Syllabus

1. Information regarding the programme

1.1 Higher education institution	**Babes-Bolyai University**
1.2 Faculty	**Mathematics and Computer Science**
1.3 Department	**Computer Science**
1.4 Field of study	**Computer Science**
1.5 Study cycle	**Masters**
1.6 Study programme / Qualification	**Applied Computational Intelligence**

2. Information regarding the discipline

2.1 Name of the discipline	**Simulation Methods**						
2.2 Course coordinator	**Andras Libal**						
2.3 Seminar coordinator	**Andras Libal**						
2.4. Year of study	**1**	2.5 Semester	**2**	2.6. Type of evaluation	**W**	2.7 Type of discipline	**Mandatory**

3. Total estimated time (hours/semester of didactic activities)

3.1 Hours per week	3	Of which: 3.2 course	2	3.3 seminar/laboratory	1
3.4 Total hours in the curriculum	42	Of which: 3.5 course	28	3.6 seminar/laboratory	14
Time allotment:	hours				
Learning using manual, course support, bibliography, course notes	28				
Additional documentation (in libraries, on electronic platforms, field documentation)	14				
Preparation for seminars/labs, homework, papers, portfolios and essays	14				
Tutorship	7				
Evaluations	8				
Other activities:					
3.7 Total individual study hours	71				
3.8 Total hours per semester	113				
3.9 Number of ECTS credits	7				

4. Prerequisites (if necessary)

4.1. curriculum	• None
4.2. competencies	• C/C++ programming skills

5. Conditions (if necessary)

5.1. for the course	• Projector, Wireless Internet connection
5.2. for the seminar /lab activities	• Projector/Wireless Internet connection

6. Specific competencies acquired

Professional competencies	Knowledge about the main simulation methods used in scientific computing (Molecular Dynamics, Monte Carlo, Cellular Automaton, FEM, CFD) Capability of writing a simple simulation code that can be later extended to incorporate more sophisticated models (Molecular Dynamics, Monte Carlo, Cellular Automaton, FEM, CFD) Capability of using a previously written simulation code and adapting it to the needs of the given research project (Lammps, OOMMF, GleaM, FoldIt, Ansys, Abaqus) Understanding the importance of efficient code writing, cache misses, introduction to high performance computing, the use of parallel programming, threads (OpenMP, pthreads), vectorization and the basics of writing algorithms for clusters (MPI)
Transversal competencies	Development of a scientific problem-solving mindset in describing and solving projects Development of flexibility in problem solving by encountering a variety of scientific problems and solutions to these problems, both different from the usual problems that computer science majors encounter

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	Familiarity with the major simulation methods used in science and engineering, capability of starting to write and develop his/her own simulation, capability of rewriting, modifying and adapting previously written simulations to a project's specific needs.
7.2 Specific objective of the discipline	Teaching the basics of Molecular Dynamics, Monte Carlo, Cellular Automaton, Complex Networks, Epidemiology, Biological Physics Problems, Finite Element, Computational Fluid Dynamics and presenting many interesting and actual research problems from different scientific fields.

8. Content

8.1 Course	Teaching methods	Remarks
1. 1. Molecular Dynamics Simulation - Introduction to Molecular Dynamics	presentation, individual study and evaluation	
1. 2. Molecular Dynamics Simulation - Optimization of a computer simulation	presentation, individual study and evaluation	
1. 3. Monte Carlo Simulation, Monte Carlo Metropolis Algorithm	presentation, individual study and evaluation	

1. 4. Cellular Automatons	presentation, individual study and evaluation	
1. 5. Soft Condensed Matter Simulations - Colloids, Granular Materials and Active Matter	presentation, individual study and evaluation	Introducing students to my current research topics
1. 6 .Hard Condensed Matter Simulations - Micromagnetic simulation (OOMMF)	presentation, individual study and evaluation	Introducing students to my previous research topics
1. 7. Complex Networks	presentation, individual study and evaluation	Course material available for further study from the Barabasi Lab
1. 8. Epidemiologic simulations	presentation, individual study and evaluation	In connection with the research done at Indiana University in the Vespigniani group
1. 9. Biological Physics - protein folding simulations, viral capside assembly simulations	presentation, individual study and evaluation	Research topics from Notre Dame, Harvard, RPI and UC Riverside
1. 10. Computational Neuroscience - on the boundary of Physics, Biology and Computer Science	presentation, individual study and evaluation	TED talks on current advances in the subject
1. 11. Engineering Simulations - Finite Element Modeling	presentation, individual study and evaluation	
1. 12. Engineering Simulations - Computational Fluid Dynamics	presentation, individual study and evaluation	
1. 13. Introduction to High Performance Computing - caching optimisation, threading (openmp, pthreads), vectorization	presentation, individual study and evaluation	course material from Jeff Amelang, Caltech
1. 14. Introduction to High Performance Computing - Parallelization (MPI), Load Balancing, MPMD simulations	presentation, individual study and evaluation	course material from Jeff Amelang, Caltech
Bibliography		
8.2 Seminar / laboratory	Teaching methods	Remarks
1. 1. Writing a molecular dynamics code for a brownian dynamics simulation	teaching by example, individual project	developing own code
1. 2. Optimising that molecular dynamics code with Verlet lookup lists, tabulated lookup tables and cache-friendly memory structures (structure of arrays/space filling curve sorting on the indices)		developing own code
1. 3. Granular simulation for random sequential adsorption with Monte Carlo method	teaching by example, individual project	developing own code
1. 4. Programming the Vichniac algorithm for Cellular Automatons	teaching by example, individual project	developing own code
1. 5. Writing a brownian dynamics code for active matter with the Vicsek model, showing the phase tradition towards ordered phase in the motion	teaching by example, individual project	developing own code

1. 6. Running a hysteresis simulation, post-processing the simulation data	teaching by example, individual project	using existing code
1. 7. Videos on complex networks	teaching by example, individual project	videos
1. 8. Videos on epidemiological simulations, setting up and running a GleaM simulation on a custom made infectious disease	teaching by example, individual project	videos, using existing code
1. 9. Writing a simple robotic arm simulation for protein folding. Playing with the FoldIt simulation through all introductory levels	teaching by example, individual project	developing own code
1. 10. Watching a series of TED videos on Computational Neuroscience	teaching by example, individual project	videos
1. 11. Setting up a tutorial simulation in Abaqus	teaching by example, individual project	using existing code
1. 12. Setting up a tutorial simulation in Ansys	teaching by example, individual project	using existing code
1. 13. Using papi for cache miss counting; vectorization optimisation on one of our previous codes	teaching by example, individual project	developing own code/learning from example code
1. 14. OpenMp and pthreads examples, parallelizing one of our previous codes	teaching by example, individual project	developing own code/learning from example code

Bibliography

The bibliography for this course consists of several books, shorter lecture notes, references to complete courses on the subject, each for the specific area of computer simulation we covered.

1. Computer simulation of liquids (MP Allen etc.)
2. The Art of Molecular Dynamics Simulation (DC Rapaport)
3. Lecture notes on Monte Carlo by Zoltan Neda
4. Introduction to MC Algorithms MC Krauth
5. An Introduction to CA and their applications, Sam Northshield et al
6. CA and LBA techniques: an approach to model and simulate complex systems, Bastien Chopard et al.
7. OOMMF user guide (NIST)
8. Barabasi Lab, lecture notes on complex networks by Albert-Laszlo Barabasi
9. Gleamwiz, Comptartmental models by Bruno Gonclaves
10. FoldIt
11. Abaqus/Ansys tutorials
12. Introduction to high performance computing, Jeff Amelang, Caltech

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

The knowledge acquired in this course allows computer science majors to get acquainted with the world of scientific computer simulation and modelling, thus enabling them to transfer their programming knowledge to fields that traditionally employ more scientists than programmers. Computer simulation and modelling has become a mainstay in many industries (from aerospace industry to car manufacturing, down to molecular modelling in medical and pharmaceutical industries). Today there is virtually no high-tech field that does not employ computer simulation at some point., and this course is meant to be a bridge for the students to enter that world and bring their computing expertise to fields that need that expertise.

10. Evaluation

Type of activity	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Share in the grade (%)
10.4 Course	Quizz on every course		20
	Final Exam		40
10.5 Seminar/lab activities	Individual Projects		40
10.6 Minimum performance standards			
• 50% (5.0) grade on the combined Quizz+Individual Projects Score • 50% (5.0) grade on the Final Exam score			

Date	Signature of course coordinator	Signature of seminar coordinator
15.01.2013	Andras Libal	....

Date of approval	Signature of the head of department
....	...........………........................