

# Teaching Model for Computational Science and Engineering Programme\*

Hayden Stainsby, Ronal Muresano, Leonardo Fialho, Juan Carlos González,  
Dolores Rexachs, and Emilio Luque

Universitat Autònoma de Barcelona  
Computer Architecture and Operating System Department (CAOS)  
Barcelona, Spain  
{hstainsby,rmuresano,lfialho,jgonzalez}@caos.uab.es  
{dolores.rexachs,emilio.luque}@uab.es

**Abstract.** Computational Science and Engineering is an inherently multidisciplinary field, the increasingly important partner of theory and experimentation in the development of knowledge. The Computer Architecture and Operating Systems department of the Universitat Autònoma de Barcelona has created a new innovative masters degree programme with the aim of introducing students to core concepts in this field such as large scale simulation and high performance computing. An innovative course model allows students without a computational science background to enter this arena. Students from different fields have already completed the first edition of the new course and positive feedback has been received from students and professors alike. The second edition is in development.

## 1 Introduction

Computational Science and Engineering (CSE) is the increasingly important partner of theory and experimentation in the development of knowledge. CSE requires multidisciplinary work, which allows the undertaking of complex scientific and engineering problems, under the unifying concepts of computing and mathematics.

Different industrial sectors such as medicine, life sciences, mechanics, economy, social sciences and management together with engineering use key CSE techniques, like modelling and simulation, to aid in solving their problems [1].

Due to this requirement the Computer Architecture and Operating Systems (CAOS) department of the Universitat Autònoma de Barcelona (UAB) initiated an innovative masters programme in Computational Science and Engineering. The members of the CAOS department have previous experience in these areas, for some years they have been performing research in the fields of advanced parallel and distributed simulation techniques, such as individual orientated simulation in biology and forest fire propagation, using High Performance Computing.

---

\* Supported by the MEC-Spain under contract TIN2007-64974.

The results of this experience have been a number of masters and doctoral (PhD) theses as well as research papers and participation in different research projects funded by the European Union.

The programme has been designed according to the European Higher Education Area (EHEA), the so-called Bologna process, the current European standardised university degree system [2].

The masters programme in CSE is geared towards accepting students of various backgrounds across different sectors. This includes students from all areas that make use of mathematical modelling and scientific computing technologies.

Further to the core subjects, depending on the student's background, a different set of subjects will be offered to complement their knowledge according to the skills expected at the conclusion of the programme.

Upon concluding the programme, students will have the capability to identify computational needs in the areas in which they are working. The degree gives students the option of following either an academic or professional career path.

In this programme students will be given opportunities to join and perform within a research group as well as the possibility of working with high performance computers. Students learn investigation techniques, experience participating in multidisciplinary groups, and working with other people with expertise in different areas. These skills are applied in the development of the masters thesis for students following a research specialisation, or project report for the students focussing on professional skills.

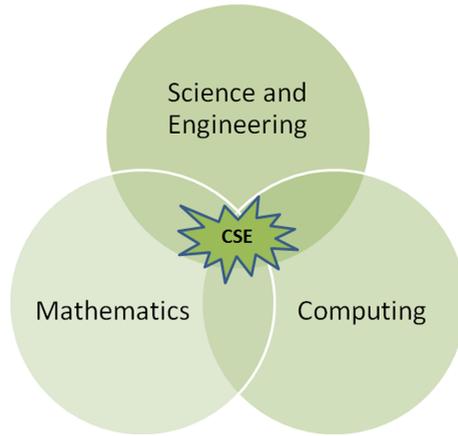
This paper is written as follows. Section 2 discusses the course model, both the process of knowledge standardisation and the core and optional subjects. Section 3 presents the experiences obtained and observations made after the first year of running the CSE masters programme, shown using examples of work from masters theses. Finally, the conclusions are stated in section 4.

## 2 Overview of the Proposed Course Model

The classical concept of three pillars of knowledge, upon which computational science is based, has driven the model for the masters degree in Computational Science and Engineering. These three pillars are Mathematics, Computation, and Science and Engineering as shown in Figure 1. The relationship between these three pillars and the content of the CSE masters programme is discussed in section 2.2.

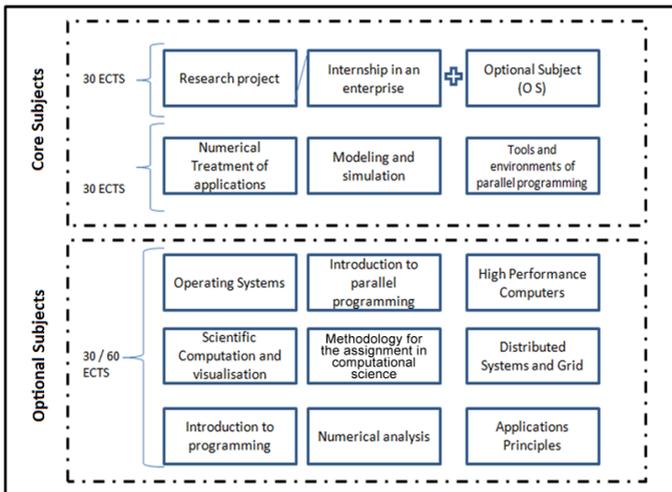
According to the Bologna Process to achieve a masters degree a student must have completed a total of 300 European Credit Transfer System (ECTS) credits including the bachelor degree (180 ECTS) or graduate (240 ECTS) credits. This model accepts students who have already completed more than 180 ECTS credits [3][4].

The core subjects of this masters programme are worth a total of 60 ECTS credits. The programme is divided in three compulsory subjects and one optional subject each worth 10 ECTS credits and a project which is worth 20 ECTS credits. There are nine additional subjects, each also worth 10 ECTS credits. The complete course structure is shown in Figure 2.



**Fig. 1.** Three pillars of Computational Science and Engineering

Students who have already completed 240 ECTS credits and have the requisite computational science background are able to begin the masters course with the core subjects, the second year of the programme. Other students may need to complete all or part of the first year. There are two possible reasons for this, the first is that the student has only completed 180 ECTS credits, usually analogous to three years of study under the ECTS. The second reason is that the student requires additional knowledge to follow the core of the course, this is discussed further in Section 2.1.



**Fig. 2.** CSE Course Structure

## 2.1 Knowledge Standardisation

This Computational Science and Engineering masters programme is designed to accept students from various academic fields. A challenge of this model is that students may begin the course without sufficient background in one of the fundamental areas of the course, such as computational science, mathematics, or parallel applications. In order to solve this problem, the masters programme has proposed a criteria to help standardise the students' knowledge in computational science.

The proposed criteria is based on two phases. The first phase starts with the analysis of each candidate's previous academic experience, sent during their application. In the second phase, once the candidate has been accepted, they attend an interview with the course director in order to define what additional or complimentary knowledge they will require. The optional subjects of the masters programme will provide the opportunity for the student to gain this knowledge.

A multidisciplinary committee will analyse each candidate's academic experience in order to define which subjects from the student's previous studies could be credited towards the masters course model. Taking into account the number of ECTS credits which the candidate must achieve to complete a masters degree, a different number of subjects may be required. This analysis takes into account the areas in which the candidate has previous academic experience, e.g. a candidate who already has some programming experience will most likely be better equipped to understand the course core than another candidate with no equivalent experience. There are a number of different aspects that are taken into consideration when choosing the optional subjects which will make up the student's knowledge standardisation. Some examples are listed in Table 1.

The possibility of choosing these optional subjects permits the student to compose a more personalised programme. The completion of the optional subjects also allows the student to follow the course core.

## 2.2 Core Subjects

The core of the course is comprised of two parts. The first part involves three core subjects and forms the basis of the skills students will learn in the masters programme. The second part allows students to orientate themselves towards research or professional skills, by choosing either the masters thesis or a professional internship.

While each of the core subjects features aspects of one of the three pillars more than the others, just like Computational Science and Engineering itself they are composed of all three key areas. Tools and Environments for Parallel Programming draws mainly from Computing , Modelling and Simulation is seated mostly within Science and Engineering, and Numerical Treatment of Applications is based in Mathematics.

**Tools and Environments for Parallel Programming.** Focusses on introducing students to parallel programming and the variety of tools available to

**Table 1.** Important aspects of a student's previous experience and examples

Academic experience	The specific topics the student has studied in previous academic courses. This also takes into account additional certificates and accreditations in relevant fields, e.g. an economist who has completed a course in financial mathematics may not need to take a numerical analysis subject.
Professional experience	Particular aspects of the student's professional career which are relevant to the CSE course, e.g. an electronic engineer who has worked with micro-controller programming will most likely not need to take the introductory programming subject.
Masters orientation	According to the student's motivation for starting the masters programme, whether they are considering a professional specialisation or to begin a research career, e.g. a student wishing to follow a research path would find subjects such as Methodology for the assignment in computational science or Scientific computation and visualisation of most interest.
Area of interest	Corresponding to the field in which the student wishes to apply the knowledge gained in this masters course, different subjects should be considered, e.g. a biologist who wishes to work in the management of a research centre which makes use of large scale calculations would attend subjects oriented towards computer architecture such as High performance computers or Distributed systems and GRID.

assist in all stages of the design, construction, and analysis of parallel applications. In this subject the students must make use of knowledge from all areas of CSE.

As part of this subject, the student will choose a problem to solve through the use of parallel computing. The subject is structured around providing the student with the skills and knowledge necessary to complete this project. The course begins with introductions to various methods of parallel programming, such as message passing and shared memory. During this time the student will select a topic and present it, along with how they propose to create a parallel version, to the class. In this presentation feedback is given, both by the class instructor and fellow students. Using this knowledge, the student then goes away to complete the parallelisation and performance analysis of their project. At the same time the class topics begin to cover a range of skills that will assist the student in this task, skills such as performance modelling and analysis, and scalability.

While the student is working on their personal project they may encounter difficulties, whereupon they are encouraged to seek assistance from their class instructor. This interaction provides a more personalised approach to teaching,

**Table 2.** Course content for Tools and Environments for Parallel Programming

Parallel programming	Message passing, shared memory, and skeletons
Parallel paradigms	Master-worker, pipeline, and single programme multiple data
Performance analysis	Performance models, dynamic instrumentation, and automatic tuning tools

where the instructor can give advice on tools and methods relating to the student's specific problem field.

At the end of the subject, the student will give a second presentation to the class detailing the process they went through in creating a parallel version of their problem and the performance that it has achieved.

Throughout this entire process the student will learn far more than how to write parallel code. They will understand the processes involved in analysing a serial algorithm and making efficient use of resources in order to parallelise it.

Table 2 shows the subject contents.

**Modelling and Simulation.** Aims to develop a model which represents a problem in a specific research field. This permits the student to apply methodologies, following scientific criteria, in order to define and extract conclusions and evaluate performance on the proposed problem.

The student first learns the basics of modelling, the considerations that need to be made when a real world system is modelled and how the expected outcome of a model effects decisions about which aspects of a system are important in the model.

This knowledge is used when the student goes on to learn about the various methods of simulation, especially the uses of system dynamics and discrete event simulation techniques. This leads into the importance of the analysis of simulation data and design practices.

Throughout a number of classes in this subject specific simulation areas are presented by experts in their respective fields. The student will understand how modelling and simulation is applied in real world situations, for example classes are given by a sociologist with computational experience. The student is shown a specific social sciences model and the method used to solve it using simulation software. This hands-on experience teaches the student the techniques and tools used to solve many kinds of problems. Other use cases give the student the appreciation of specific aspects of simulation.

During this subject the student will understand not only the basic concepts of how to develop models of real world systems and simulate them using computer software, but ways in which this has been applied to specific problem areas.

Table 3 shows the subject contents.

**Numerical Treatment of Applications.** Teaches students to apply the numerical techniques of computational modelling for applications in specific fields

**Table 3.** Course content for Modelling and Simulation

Modelling	Types of models (Heuristic and Empirical), application field (Conceptual and mathematical), representation field (Qualitative and Numerical) and verification and validation
Simulation	Physical systems and simulation, language and tools, performance metrics and distributed high performance simulation
Application use cases	Biological, economic and social systems, and propagation of forest fires

of science and engineering. This subject is a link between Modelling and Simulation and Tools and Environments for Parallel Programming just as mathematics is the path between a problem and the problem solving process.

In the first part of the subject the student learns mathematical techniques and how they are applied in a computing system. This covers such topics as floating-point errors, data mining, and efficient methods for storing data. This is important when explaining the solving of linear equation systems, where the student will use different solving methods and compare them using mathematical software.

The student will choose a topic in the area of statistical modelling which they will present to the class. This builds upon what the students have already been taught in class and allows the student to investigate a topic of interest to them in more depth. Another presentation is given by the student on different techniques for solving differential and integral calculus. These techniques are analysed and their potential for parallelisation is investigated.

This method of constant student feedback in the form of projects and presentations given to their peers allows a degree of interaction between the students and the instructor that gives an insight into these topics with the objective of identifying and reinforcing the key points.

The subject contents are shown in Table 4.

**Table 4.** Course content for Numerical Treatment of Applications

Mathematics	Application to problem modelling, linear equation systems, statistics, and integral and differential calculus
Application	Efficient methods of storing information, optimisation and search methods

**Selection of the Master Orientation.** Once the student has completed these three core subjects they will choose the focus of their masters degree. As mentioned previously, there are two possible profiles, researcher or professional skills.

Students who choose the research focus for their masters will join one of the existing research groups and choose a topic in that area. During the first part

of the course the student will research their chosen area and attend the research group meetings. This culminates in the final part of the course where they will write the masters thesis, and must defend it in front of a tribunal.

The students that follow the professional skills course will spend time working on a computational science project during an internship in a firm. They will then produce a report about the project and present it to a panel composed professors from the CAOS department.

Throughout this model students undergo a significant learning process. The three core subjects, and subsequent project give each student a well rounded knowledge of all aspects of the three Computational Science and Engineering.

### 2.3 Optional Subjects

The optional subjects exist to permit students from different areas to standardise their knowledge so that they may better follow the core subjects. Students with degrees not related to computational science or with bachelor degrees must choose a set of subjects in order to perform between 10 and 60 additional ECTS credits, depending on their previous knowledge. These are also the subjects that students choose from to fill the optional subject space in the course core.

There are nine optional subjects which the students may choose from, each one worth 10 ECTS credits. These subjects can be split into two groups. The first five subjects teach knowledge complimentary to the course core, and are as follows.

- **Methodology for the assignment in computational science** discusses the standards and formats used in project financing, writing computational science projects, and the presentation of results. Students learn the processes involved in applying for financing for projects and present a mock application to the other students. This is the subject where students will gain most of the knowledge they require to present their masters thesis.
- **High performance computers** describes the computer components used to create high performance machines, interconnection networks, computing clusters, fault tolerance techniques, and input/output mechanisms. As part of this subject students research an existing supercomputer and perform an analysis of it's components. Students also complete practical work either using a network simulator or parallel performance simulation software. Upon completing this subject the student will be able to evaluate different architectures to choose the most appropriate according to given criteria.
- **Distributed systems and GRID** introduces students to parallel and distributed systems across different administrative domains, including programming, performance optimisation, and simulation. During this subject the students will read a number of seminal papers in the field of distributed system and take turns presenting the contents to the other students, followed by discussions on these topics between the students and the instructor.
- **Application principles** teaches students to recognise different applications that need high performance computing and how to identify their requirements. In this subject the main goal is to show students the many uses of

high performance applications in industry. Students hear lectures from individuals from many different fields that use parallel tools, and can see the diverse range of areas that make use of them. Students will also choose an application and design some improvements to it, the results of which are presented to the class.

The remaining four subjects provide introductory knowledge, which enables students to follow the course core, they are listed below.

- **Operating Systems** assures that students have a standard level of knowledge of the efficient use of operating systems in scientific computation fields.
- **Scientific Computation and visualisation** teaches the basic techniques of representing computational results in different scientific fields.
- **Introduction to programming** gives students a knowledge of programming and debugging in scientific languages as well as the basic concepts of software engineering.
- **Introduction to parallel programming** establishes methodologies and strategies to parallelise applications using message passing and shared memory libraries.
- **Numerical analysis** teaches students to resolve differential equation systems and computational optimisation techniques.

Alongside the course work, there are conferences and seminars organised within the department, at which presentations are given by relevant people from a range of academic and professional areas, complementing the students' studies.

### 3 Experience Obtained

The Computational Science and Engineering masters programme has already begun, thus some experiences can be presented. Masters theses have been completed in different fields, including biology and forest fire propagation. There are also ongoing projects in other fields, such as social sciences and economics. These projects have been built upon areas in which the CAOS department already has experience<sup>1</sup>. Three examples of masters theses, which were presented and defended in July 2008, will be described here: forest fire propagation prediction[5], modelling and simulation of fish school movement[6], and parallelisation of the kriging interpolation method[7]. These topics provide good examples of fields which make use of the three pillars of computational science and engineering in order to achieve their objectives.

The prediction of forest fire propagation is possible through the use of computational science and engineering methods. In this case, science and engineering provides the moisture burn model, associated meteorological variables and other input data in order to create the propagation model. Through mathematical techniques this model can be enhanced with genetic and heuristic. Finally, computation gives the calculation power necessary to produce a set of scenarios based on these variables.

---

<sup>1</sup> [http://caos.uab.es/research\\_projects.php](http://caos.uab.es/research_projects.php)

The modelling of fish school behaviour is already established and well known in the field of biological science. However, in this case the numerical model does not accurately represent reality. Using fuzzy logic it may be possible to achieve better results. Computation provides the resources required to simulate this model and the mechanisms to visualise the results in a manner that permits the biological specialists to refine their original model.

Kriging is a method for interpolating unknown values from data at known locations, while slower than other methods of interpolation it can provide more accurate results. By implementing a parallel version of kriging interpolation using message passing on a distributed memory machine it was possible to significantly reduce the execution times of the entire process.

These examples show how this course model helps specialists use computational science and engineering in order to solve their problems.

To appraise the students' degree of satisfaction the CAOS department prepared a survey where students evaluated the masters programme in general as well as each specific subject. These results are being used in the development of the second edition of this course.

## 4 Conclusion

This paper has presented an innovative model for teaching computational science and engineering. The focus is providing the opportunity for students with and without computational science backgrounds to be introduced to this field. Some positive results have been shown through masters theses, which demonstrate the effective application of this model.

In the second year of this course, the number of students in the programme has increased considerably, which indicates the acceptance of this model.

## References

1. Guha, R., Hartman, J.: Teaching parallel processing: Where Architecture and Language Meet. In: Proceedings of Twenty-Second Annual conference on Frontiers in Education, 1992, pp. 475–479 (1992)
2. Wachter, B.: The bologna process: developments and prospects. *European Journal of Education* 39(3), 265–273 (2004)
3. Kargidis, T., Kefalas, P., Stamatis, D., Tsadiras, A.: Towards a European Credit Transfer System for Networked Learning (ECTS-NL) (2003)
4. European Union: ECTS Users' Guide. The European Credit Transfer System. European Commission Publication 31(03) (1998)
5. Wendt, K.: Efficient knowledge retrieval to calibrate input variables in forest fire prediction. Master's thesis, Universitat Autònoma de Barcelona (2008)
6. Gonzalez, J.: Individual oriented model applying fuzzy logic. Master's thesis, Universitat Autònoma de Barcelona (2008)
7. Pesque, L.: Solució paral.lelitzada d'interpolació kriging amb ajust automatitzat del variograma. Master's thesis, Universitat Autònoma de Barcelona (2008)