AC 2008-2901: EXPERIENCES IN TEACHING AND MENTORING INTERDISCIPLINARY GRADUATE STUDENTS OF DIVERSIFIED BACKGROUNDS

Ram Mohan, North Carolina A&T State University

Dr. Ram Mohan is an associate professor with the computational science and engineering graduate program at North Carolina A&T State University.

Vinaya Kelkar, North Carolina A&T State University

Dr. Vinaya Kelkar is a statistician and assistant research professor in the Department of Biology at North Carolina A&T State University.

Ajit Kelkar, North Carolina A&T State University

Dr. Kelkar is Director of Computational Science and Engineering graduate program at North Carolina A&T State University.

Experiences in Teaching and Mentoring Interdisciplinary Graduate Students of Diversified Backgrounds

Abstract

The interdisciplinary master's degree program in Computational Science and Engineering (CSE) at North Carolina A&T State University, Greensboro, NC is now more than 3 years old, and provides graduate education in several computational areas and the associated primary field disciplines. The CSE program since its inception has presently graduated more than 12 students who are currently placed in several major industries. Our CSE graduate program offers an interdisciplinary curriculum combining computational core areas along with various domain areas. The students enrolled in the program begin with diversified backgrounds (prior undergraduate studies in various fields such as engineering, physical sciences, life sciences, mathematics, business, etc), and are required to take four core courses relevant to CSE for their graduation in the areas of applied probability and statistics, comprehensive numerical analysis, Data structures and parallel programming, computational and scientific visualization irrespective of their prior background. The preparation level for the diversified group of students in these courses depends on their undergraduate major. This poses significant challenges to graduate faculty teaching these courses and mentoring these students with diversified backgrounds.

Our experiences and observations with the course content and structure, teaching methods, evaluation and student performances in these courses with diversified graduate students and their mentoring for the past 3 years are presented. The performances of the students in these core courses are correlated to their background and analyzed. Our experiences indicate students with a lesser preparation level seem to get geared quickly via peer guidance from the stronger students. In several cases, the performances were not related to an individual's prior background, but on their motivation and willingness to strive and succeed. The experiences indicate that the students can benefit well with additional short training courses at the beginning of their graduate study. Though challenging, the positive outcome of such interdisciplinary education is that the graduates are able to technically understand and communicate effectively across disciplines in complex problem areas where such interdisciplinary interactions are not only critical, but are required in the current market place and global economy. This is reflected in the career placement of graduates in areas that generally would not have been possible based solely on their undergraduate field.

Introduction

The new paradigm in graduate studies is interdisciplinary programs that meet the technical needs of the current practices in the field and industry. Modeling and simulation built upon computational science and engineering has now become the third key solution methodology in not only engineering and physical sciences but also in other areas such as biology and economics that are generally considered to be non-computational fields. The interdisciplinary master's degree program in Computational Science and Engineering (CSE) at North Carolina A&T State University is now more than 3 years old, and provides graduate education in several computational areas and the associated primary field disciplines. The CSE program since its inception has presently graduated more than 12 students who are currently placed in several major industries.

Our CSE graduate program offers an interdisciplinary curriculum combining applied mathematics, high performance parallel and scalable computing, scientific modeling and simulation, data visualization, and domain areas such as physical sciences and engineering, life sciences, agricultural and environmental sciences, technology and business. The students enrolled in the program begin with diversified backgrounds (prior undergraduate studies in engineering, physical sciences, life sciences, mathematics, business, etc). However, all students in the program are required to take four core courses relevant to CSE. These are: comprehensive numerical analysis; scientific visualization, applied probability and statistics, parallel programming and data structures. The preparation level for the diversified group of students in these courses depends on their undergraduate major. This poses significant challenges to graduate faculty teaching these courses and mentoring these students with diversified backgrounds.

This paper focuses on our experiences and observations with the course content and structure, teaching methods, evaluation and student performances in these courses with diversified graduate students and their mentoring for the past 3 years. The performances of the students in these core courses based on the evaluations through tests, projects, etc., using the data collected for the past 3 years are correlated to their background and analyzed. Our experiences and observations of the technical and interdisciplinary maturity from the time of the admission and the graduation of the students are highlighted.

This paper is organized as follows. The interdisciplinary graduate master's program and its structure at our institution are discussed first. This is followed by discussions of our experiences and observations in teaching and mentoring the diversified group of students with varying incoming preparation levels and undergraduate degree background in two of the core courses. The positive outcomes of the interdisciplinary education that we had observed and based on the feedback from the graduates who are placed in varying industries are presented next. Our experiences indicate that the interdisciplinary CSE graduate education had enabled the career placement of graduates in areas that generally would not have been possible based solely on their undergraduate field.

Computational Science and Engineering Program at Our University – Overview

The computational science and engineering graduate program at North Carolina A&T State University is now more than 3 years old and had graduated more than 12 students who are currently placed in several major industries. This program is highly interdisciplinary, drawing expertise and resources from various disciplines across the University, and operating outside a department. It offers an interdisciplinary curriculum combining applied mathematics, high performance parallel and scalable computing, scientific modeling and simulation, data visualization, and domain areas such as physical science and engineering, life sciences, agricultural and environmental sciences, technology and business and will help in learning across the boundaries between engineering, science, technology and business.

Our CSE master's program has three tracks with a focus on computational science, but distinguishes across the domain areas of specialization. The three tracks has a common curriculum in their core courses will account for the variations in computational science field requirements across the several domains, and is required to be taken by all the CSE students. Thus the students in these core courses are a diversified group with a different preparation level depending on their background and undergraduate major. The three major tracks with a common core curriculum required for all the students are briefly presented next.

Computational Science and Engineering

This track is designed primarily for students with undergraduate degrees in engineering, physics, mathematics, and computer science who will are trained to develop problem-solving methodologies and computational tools as well as interdisciplinary technical expertise in CSE for solving challenging problems in physical science, engineering, applied mathematics or computer science. This includes domains that are both in the College of Engineering, and the College of Arts and Sciences. The curriculum emphasizes computational sciences and engineering along with training in the domain areas. The goal of this track is to produce scientists, and engineers with focus, training and application in computational sciences, scalable computing, physics-based modeling and simulations, and with expertise in the application of computational techniques and principles in their primary domain areas. Based on their undergraduate degrees, the students in this track have had an increased level of prior training, courses and exposure to mathematics including areas such as numerical analysis and in some cases to a high level programming languages.

Computational Sciences

This track is designed primarily for students with undergraduate degrees in chemistry, biology, business, and agricultural sciences who are trained to apply or extend computational tools and methods as well as data acquisition, processing and visualization techniques to study computationally intensive problems in life sciences, agricultural and environmental sciences, and business and economics. This track primarily includes domain areas with lesser training in mathematics including numerical analysis, programming languages.

Computational Technology

This track is designed primarily for students with undergraduate degrees in technology disciplines with focus on computational science and engineering. These technology disciplines currently include computation technology, computer numerical control machining, remote sensing, GIS/GPS data analysis, and nanotechnology with additional potential disciplines in the future. The goal of this track is to produce technologists with a focus and training in computational sciences, and in their primary technology domain area.

Educational Objectives of our CSE Graduate Program

The educational objectives of our CSE graduate program are as follows:

- Educate and graduate students with a mastery of high performance computer programming tools as well as processing, data acquisition, analysis techniques.

- Acquire, educate and train in computational modeling, simulation and visualization.
- Learning across the boundaries between engineering, science, technology and business.
- Relate acquired computational science and engineering knowledge and skills to specific application fields of engineering, science, technology and business.
- Learn to develop novel and robust computational methods and tools to solve scientific, engineering, and technological and business problems.
- Produce highly versatile computational scientists, engineers, technologists, or business executives with a good understanding of the connections among various disciplines and capable of interacting and collaborating effectively with scientists, engineers, and professional in other fields.
- Increase the number of graduate professionals available to work in computational science and engineering.

Course Curriculum for CSE

CSE has emerged as a powerful and indispensable method to analyze a variety of problems in research, production and process development, design, manufacturing as well as in domains such as computational chemistry, biology, genomics, business forecasting, economic modeling, etc. Computational modeling and simulation is being accepted as a third methodology in scientific discovery processing and engineering design, complementing the traditional approaches of theory and experiment. Many experiments and investigations that have traditionally been performed in a laboratory or the field are being augmented or replaced by computational modeling and simulation. Examples include weather and climate modeling [1,] fossil fuel combustion simulation [2], engine and vehicle design [3], materials development [4], aircraft design [5], electronic design automation [5], and drug design and development [6]. Scientific visualization is another primary element of CSE, and has become an essential tool for the preprocessing of data sets and the investigation of massive amounts of computational results, as increasingly evident in bioinformatics, finance, and the mining of huge data sets [7]. Computational modeling, simulation, and visualization are immensely useful for studying things that are otherwise too big, too small, too expensive, too scarce, or too inaccessible to study.

Even though CSE makes use of the techniques of applied mathematics and computer science for the development of numerical algorithms and computing tools to the study of scientific and engineering problems, as well as other physical, biological and non-scientific areas, it is by no means a subfield or extension of applied mathematics or computer science, nor is it a discipline where a scientist or engineer or domain specialist simply uses a canned code to simulate data and visualize results. "CSE is a legitimate and important academic enterprise," as noted in a comprehensive, report [5] published by the SIAM Workgroup on CSE on Graduate Education in CSE. "Although it includes elements from computer science, applied mathematics, engineering and science, CSE focuses on the integration of knowledge and methodologies from all these disciplines, and as such is a subject which is distinct from any of them." The following figure, which has been widely accepted in the CSE community, reflects the view that besides connecting the sciences, engineering, mathematics, and computer science, CSE also has its own

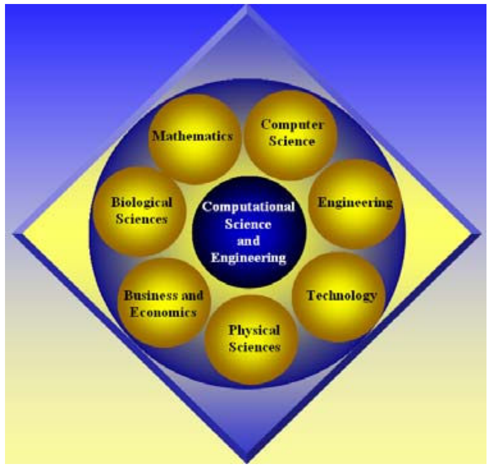


Figure 1: Interdisciplinary Nature of the CSE Program

core of elements that draws together and bridges all these disciplines. Such a CSE core is made up of a collection of computationally intensive problem-solving methodologies and robust tools, which constitute the building blocks for the study of scientific and engineering problems of ever increasing complexity and realism.

Based on the above core requirements of CSE education and objectives, our CSE program requires all the CSE students to take four CSE core courses irrespective of the undergraduate background and CSE tracks. These courses provide CSE the foundations of the enabling technologies of the computational sciences for all students irrespective of their background. These core courses are:

CSE 701: Applied Probability and Statistics
CSE 702: Comprehensive Numerical Analysis
CSE 703: Parallel Programming and Data Structures
CSE 704: Computational and Scientific Visualization

These four core courses are taken by all the students in their first two years of their graduate study. Thus the students taking these classes have a varied preparation level, background and exposure to some of the pre-requisite concepts that are generally needed in these core courses. The preparation level for the diversified group of students in these courses depends on their undergraduate major. This poses significant challenges to graduate faculty teaching these courses and mentoring these students with diversified backgrounds. Our experiences and observations with the course content and structure, teaching methods, evaluation and student performances in two of these courses with diversified graduate students and their mentoring are discussed next.

Experiences and Observations in Teaching CSE Core Courses

This section discusses our experiences and observations in teaching and mentoring the diversified graduate students in the CSE courses. Our experiences and observations from CSE 701 which is also cross listed as a course in Biology are presented next.

Experiences and Observations from CSE 701: Applied Probability and Statistics

Most students taking CSE 701 – Applied Probability and Statistics are new Master's students in their first year of graduate school. (Occasionally there is an anxious doctoral student at the end of his/her dissertation looking to reinforce their analytical skills). It is always wonderful to teach new incoming graduate students who come in focused and on a mission. Several have experienced the real world outside their alma mater and are looking forward to acquiring new skills and degrees that will make them professionally more competitive. However, most students have entered graduate school directly from their undergraduate semesters, armed with Bachelors degrees, but unhappy with the job market and their career prospects.

Just as their motives for being in this class are diverse so are their academic backgrounds for the CSE graduate students. Many students have undergraduate degrees in mathematics or engineering, whereas some have majored in Biology, Business, Physics, agricultural sciences. Their level of knowledge of statistics is also varied, although, most have had at least some elementary statistical theory. Teaching statistics to students from different backgrounds is not difficult, statistics is, in essence, an applied science and lends itself well to most disciplines than can be quantified. However, all the different backgrounds and skill levels at the same time can sometimes be a challenge.

The content of this core course in applied probability and statistics is traditional, in the sense that it covers both descriptive and inferential statistics for various types of data that will be useful for different disciplines and a necessary tool for computational scientists. Data transformation techniques often required for statistical modeling of real life data are discussed, along with non-parametric methods for when parametric assumptions are not satisfied. Fundamentals of statistical and research design methods are also covered to train and educate the students about to embark on their graduate thesis and projects. Statistics software such as SAS and/or SPSS are used and tailored according to student's skill levels and disciplines. Along with the conventional examples and problems (often based on agricultural research data), analysis of cutting edge research data from several disciplines are discussed. These include: Statistical modeling or designing experiments in composite material research to experimental design and analytical issues involved in the analysis of gene expression data from Microarrays in computational biology. The objective of these different examples and problems is to provide relevance of the statistical modeling to their own background and provides the increased motivational interest of the students. These multi-disciplinary examples also provide an understanding of the cross-disciplinary applications of the principles of statistics and their relevance in several fields. This being the case, this course as taught as an CSE core course is not designed to provide any specific entree to satisfy every appetite, but a smorgasbord of statistical concepts, tools and techniques, that are available and can be made relevant to most types of research studies and subsequent data analysis. This approach was found to positively impact the student understanding as well as potential subsequent use of the applicable techniques during their graduate studies, while providing a platform for their continued interactions after the successful completion of this course.

Our observations and experiences indicate that one unique aspect of working with or teaching an interdisciplinary course such as this is that the teaching or mentoring rarely ended with the end of the semester or even with an ‘A’ grade. The statistical concepts introduced to the students have made them realize that in a research study the time to see a statistician is not when the data has been collected but when planning for the data collection. This was found to be the case in few case study observations. Few examples of such experiences are:

- “John Smith” came to see me this summer when he started planning his study to discuss his research design and its validity and of course the type of data he plans to collect and the appropriate methods to analyze.

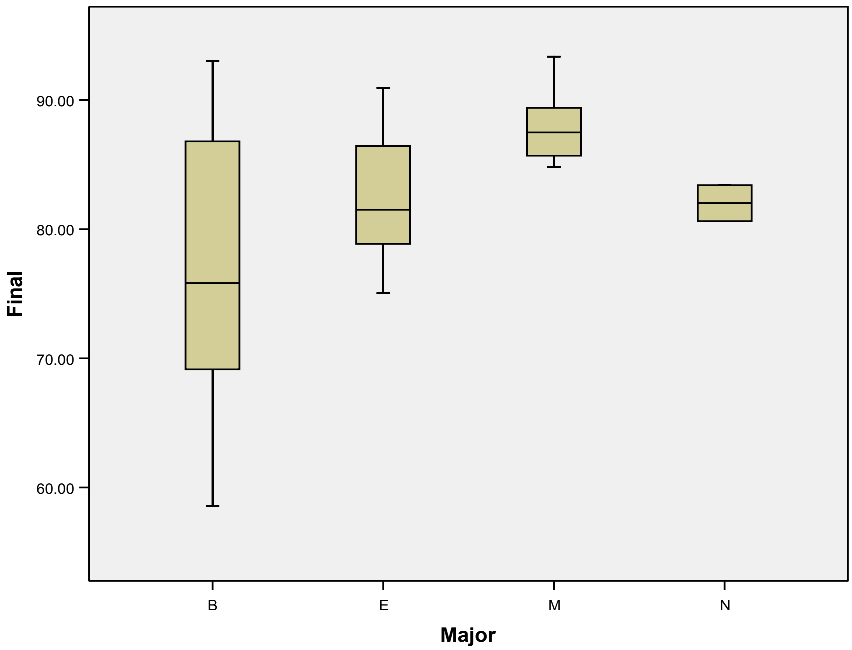
- “Jane Jones” knew she would need to use regression models for her analysis, but we had only covered linear regression in class, “and came to discuss if some other model be more appropriate?” - She was willing to research and study some advanced models on her own.

- “James R” was working on an agri-business project for his thesis, it involved a large data set with missing values that required some imputation; He knew SAS was good at handling these types of datasets needed help with debugging some do-loops.

Teaching interdisciplinary students is an invigorating experience for faculty that challenges one to think outside the box of your own field or discipline, and in finding effective ways of teaching these students. The effect of the diversified background can be seen in the performance of students in this class. A comparison of the performance and grade distributions based on the student background is presented next.

The Applied Probability and Statistics (CSE 701) is also cross listed as Statistical Methods course in the biology department and is taken by graduate and advanced undergraduate students in Biology. The box plot below shows the grade distribution of the students’ final grade scores classified by their undergraduate and domain majors. The data from recent offering of this course is presented here. The data from other prior terms were statistically similar. For the analysis purposes, the distribution of scores and grades based on the four major undergraduate and domain background grouping are considered. They are: 1. Mathematics, 2. Engineering, 3. Biology, 4. All others. The mathematics majors appear to have the numbers advantage; they scored the highest grades, clearly well above the rest of the class. The engineering and other majors had the next highest median scores with almost 75% of the groups scoring a grade of B or better (80+ points). The scores for biology majors appear to be more varied. The top 25^{th} percentile scores for biology and engineering majors are similar (at about 87 points), however, the lower 25^{th} percentile of biology scorers appear to be the lowest scores in the class. It should however be noted that the biology group not only included some undergraduate students but is also traditionally a non quantitative major.

The distribution of final grade scores classified according to student-majors for course BIOL681/CSE 703 is shown in figure 2.



Majors Code:
B: Biology E: Engineering M: Mathematics N: Others

Figure 2: Grade and Score Distribution in CSE 701

<u>Experiences and Observations from CSE 702: Comprehensive Numerical Analysis</u>

CSE involves the use of computational hardware architecture and associated software to develop numerical algorithms or methods to study scientific, engineering and other application domain problems. Most mathematical models in physical sciences, engineering as well as in other non-scientific disciplines involve the solution of systems of linear or non-linear equations, differential equations, integral equations and other mathematical such descriptions. As such, all graduates in CSE should be educated and have knowledge of the numerical algorithms for the solution of such mathematical models, their computational implementation, the computational complexity and cost of such algorithms, and the computational errors and accuracy of such computational algorithms. The core course of comprehensive numerical analysis was developed to provide a foundational background of these techniques to the diversified student background of the incoming graduate students.

Traditional courses in numerical analysis cover in depth the algorithmic methodologies, their formal mathematical development, errors and accuracy if the course is taught as a formal mathematics course. A similar course in computer science generally focuses on the efficient computational implementation of the step by step numerical algorithms, computing errors, computational complexity of the algorithms and the computational cost. Computational scientists of any domain discipline should have a good understanding of the underlying computational algorithms, limitation of such algorithms, computational implementation and computational cost of the algorithms employed for the solution of the mathematical model equations. The comprehensive numerical analysis course (CSE 702) was developed and formulated to include this understanding of the computational algorithms (with a reduced emphasis on the formal mathematical development of the algorithms) and its computer code implementation, computational cost comparisons of different algorithms that are possible for the solution of the model equations, and the computational caveats (things to look out for that can lead to incorrect results) in such computational algorithms. As such, the course content provided a wider foundational coverage that is required for all the CSE incoming graduates with a diversified background. The course content included illustrative applications of the mathematical models in various disciplines to provide the students the motivational and relevance of the computational techniques discussed in their own specific domain. For example, the solution of linear system of equations is required in several areas: Mechanical and civil engineers need these techniques for the solution of computational grid equations formed from the physical model equations via the finite element and finite difference types of approaches; the same linear system of equations are also applicable in business and economic models linking different variables, network and electrical circuit modeling, etc.

The CSE 702 course content and evaluations thus included the following major components:

- Exercises and problems to formally evaluate the understanding of the computational algorithms and the computational cost with simple system of model equations.

- Computational code implementation of the algorithms in a high level programming language to understand the practical aspects of such implementation and understand the caveats of computational algorithms via a larger set of representative problems involved in the numerical solution. Students have the option of computational implementation in any high language they are familiar. These implementations are necessary as an integral part of CSE education as the CSE graduate education is to go beyond the general use of canned analysis software, and understand their development and the associated computational issues of computational algorithms and their implementation.

These two major components required students to develop proficiency not only in the basic analysis that involved the understanding of the algorithm logic and mathematical principles, but also the programming and code development background. The student background and preparation level for these two core activities varied widely in the diversified student composition in this course. Experiences of the typical student background in the class composition for CSE 702, their strengths and weaknesses are presented next

Student Background in CSE 702

The student in CSE 702 had a diversified background with undergraduate and graduate domain in various disciplines. The major grouping of the background and the associated strengths and weaknesses observed based on our experiences are presented next.

Physical Sciences: This included students with an undergraduate degree in one of the physical science areas of physics and chemistry. Though most of them had at least two courses in calculus, a course in differential equation, some introduction to linear algebra, many did not have a formal course in numerical analysis. Surprisingly, they also appear to have very limited experience in any high level computer language at our University.

Engineering: This included students with different domains of engineering. The students with the bachelor degree background in industrial engineering were found to have a lesser mathematical background (calculus, basic linear algebra) compared to the other domains of engineering (mechanical, chemical, etc.). The CSE incoming class students also were found lacking in the area of formal programming languages, with many having limited experience in code implementation, compiling and debugging and had only prior experience in canned codes such as MATLAB functions.

Computer Science: The students with an undergraduate background in computer science had a prior background in some formal programming language and an algorithmic mindset. However, they lacked the physical connection associated with the problems.

Mathematics: The students with an undergraduate profile in mathematical sciences had a broader base in advanced calculus, primary numerical analysis but surprisingly lacked the code implementation and high level programming language skills.

Technology: The students with undergraduate background in technology lacked formal mathematical background beyond the basic courses in mathematics, with several having negligible experiences in code development and implementation.

Business & Economics: The students with this undergraduate background generally lacked the mathematical background (other than preliminary calculus courses) and formal programming background.

This diversity in student background led to the following experiences and observations and required modifications during the teaching of the course based on the class background. Each batch of students were unique but with the following commonalities observed.

- The general lack of prior computational code implementation, development and implementation by most of the students. This required additional class handout material to cover these topic areas. Generally the students used either the UNIX based computing systems for their code implementations or the Visual C, Visual C++, Visual Fortran compilers on Windows based machines. Many were not familiar with this aspect of code development, compilation and code execution that are not generally associated with the use of canned computing modules such as MATLAB. Additional class materials and tutorial sessions were required to bring the lagging students up to speed. Interestingly, most students were able to understand the algorithms, but had several issues in the computer code implementation syntax due to the lack of prior code implementation background. Their experience helped to reinforce that the internal working of canned codes are also built upon these methodologies and emphasize the intellectual knowledge, time and effort involved in such code developments. The practical code development and implementation however was found to be beneficial providing them with the much needed skills in this area that benefited the students in the other two core courses CSE 703 and CSE 704 that they took in the following semester.

- Generally, the students with a computer science background had a faster turn around time in the completion of the computer implementations due to their formal training in the data structures and computer code languages and syntaxes. These stronger students were also able to provide peer guidance to the weaker students and geared them up quickly. Several students also required extra time for the completion of the computer implementation assignments during the initial assignments with the improving performance as their experience and confidence level increased.

- Students who are weaker in mathematical aspects required some additional assistance, but were able to gear up quickly with focused, individual guidance.

- The motivation level and the willingness to strive and succeed were observed to play a major role in the computer implementation irrespective of their prior background.

The average raw scores (non-curved) based on the evaluations through tests, computer assignments and projects correlated to their major undergraduate background and domain are as shown in Table 1. Due to the diversity of the student background, their evolution and relative improvement compared during the course of the semester were also factored in the final grading. Teaching interdisciplinary students is an invigorating experience for faculty that challenges one to think outside the box of your own field or discipline, and in finding effective ways of teaching these students.

Student Undergraduate/Domain Background	Overall Raw Performance Score Range (Max = 100)
Physical Sciences	65 – 75
Computer Science	90 – 95
Mathematics	55 – 93
Technology	70 – 80
Business & Economics	60 – 70

Table 1: Correlation of Raw Performance Score to the Student Domain Background

All the observations are based on our observations and experiences at our university. The performance data suggests that the performance is not related not only to an individual's prior background but also on their willingness and motivation to strive and succeed. The data based on student population at our university suggests that though it is generally expected for the mathematics trained students to perform well in numerical analysis courses, it was not the case in some instances as it also depends on their effort level. Some of the non-science and engineering undergraduate majors out performed relatively the math majors. Our mental note of their observation indicated a higher motivation and willingness to strive and succeed in such cases.

Experiences and Observations in CSE 703 and CSE 704

Both these core courses require a computational (computer science oriented) background with a focus on data structures, parallel programming and data manipulation that is required for visualization data. Similar to the numerical analysis, students in these classes required additional training in the programming and code syntax depending on their background. The core course on visualization also involved team projects with the teams that combined engineers and computer scientists performing well. These project efforts required code implementation for data manipulation of physical data. The success depended on the students with computer science background providing the data manipulation support and interpretations and analysis and inferences by the engineers in the team. This reflects the real life situations in computational science and engineering developments where a team combination of engineers and computer scientists providing for success with the engineers focusing on the computational algorithms and physical aspects of the problem (physical problem solution methodologies and computational methods) while the computer scientists focused on the optimal code performance, improving the load balancing via better data structures and memory management.

Concluding Remarks

The graduate teaching and mentoring of students with diversified background is definitely challenging and required some out of the box approaches for effective communication and teaching of materials along with additional training in areas of programming, code development, etc. This area was found to be weaker in our experience at our university. It is clear that the incoming students need additional training in code development, other computer operating systems, and high level programming languages. The training in these areas can be achieved through additional short training workshops in the first two weeks of the first semester of the incoming CSE graduate student. Our experiences and observations indicate such short training courses for the students should include the following areas that would benefit the CSE graduate education and training.

- High Performance Computing Operating Systems (Introduction to Unix/Linux)
- Practical Programming in C/C++/Fortran

Our CSE graduate program provided the students with an interdisciplinary education with the core courses contributing to this interdisciplinary education. Generally students have a perception that disciplines and disciplinary methods are distinct and disciplines do not a commonality. The core courses of our CSE program is providing the students with an effective interdisciplinary thought process and bringing out the commonality of applicable techniques across the disciplines. This has resulted in the career placement of graduates in areas that generally would not have been possible based solely on their undergraduate field. Some representative cases in our experience are:

- “Jane Doe” with an undergraduate in mathematics, and minimal prior programming experience obtaining a “Computer Scientist” position after graduation with CSE degree.

- "John Smith", an undergraduate in computer science, with minimal engineering experience developing expertise in finite element modeling and large scale, parallel codes in mechanical engineering. These cross-disciplinary understanding is a positive outcome of the interdisciplinary graduate program and mentoring.

- "Mary John" a mathematics undergraduate major with minimal science background working in the area of computational material science.

In summary, despite the challenges involved, the positive outcome of this interdisciplinary education is that our graduates are able to technically understand across disciplines in complex problem areas where such interdisciplinary interactions are not only critical, but are required in the current market place and global economy.

Bibliography

1. Computing in Science and Engineering, Special Issue on Climate Modeling, **4** (Sept./Oct. 2002).
2. National Workshop on Advanced Scientific Computing, National Academy of Sciences, J.S. Langer, Chair, 1998.
3. Yasar, O., "A Scalable Model for Complex Flows," J. Computers and Mathematics, **33**, 117 (1998).
4. Computing in Science and Engineering, Special Issue on Materials Science, **3** (Nov./Dec. 2001).
5. SIAM Workgroup on CSE Education, "Graduate Education in Computational Science and Engineering," SIAM Review, **43** (1) (2001), pp. 163-177.
6. Computing in Science and Engineering, Special Issue on Data Mining, **4** (July/Aug. 2002).