## Sanjukta Bhowmick: Statement of Teaching

I believe that research and education complement each other. Teaching provides an opportunity to transfer, and in doing so re-evaluate our theories. On the other hand, research contributes to increasing our understanding of a subject, which is then incorporated into education. I enjoy both teaching and research, and therefore want a career as a faculty in a research-based university, where I can be involved in teaching, while simultaneously conducting multidisciplinary research in computational science and engineering.

I have had experience in teaching different levels of courses. My students include both computer science and non-computer science majors. Here are some of my teaching experiences:

**Instructor:** Currently, I am an Assistant Professor (fixed-term) in the Department of Computer Science and Engineering, Pennsylvania State University. My duties chiefly include teaching senior level scientific computing courses. In Fall 2007, I taught "Introduction to Parallel Programming". This was my first solo venture, where I was completely responsible from the selection of text-books and materials covered to the design of lectures, assignments and evaluation criteria. The course consisted two lectures of 75 minutes each week.

The aim of the course, as the name suggests, was to provide the students a window to the world of parallel programming. They were given a brief introduction to various parallel programming paradigms and architectures. The lectures emphasized on recognizing inherent parallelism within a given sequential algorithm, based on examples from numerical algebra (matrix multplication, PDEs) and combinatorial problems (branch and bound problems, game trees). The students learnt to code in MPI and implement the parallel algorithms that they learnt. As their final project, they had to design parallel versions of algorithms like Gaussian Elimination. The course website at *http://www.cse.psu.edu/∼bhowmick/cse457.html* provides the details of this course.

This was a course within my core area and I had complete freedom in designing it. Naturally, this was my most enjoyable teaching experience. The official student evaluations are not yet out, some of my students have told me that they liked the course. An average 80% attendance for a late-afternoon class, even on snowy days, seems a good indication that the students felt the class was interesting. There was one incident which made me feel that I had succeeded in arousing the students curiosity beyond course requirements. Here is the story.

I had taught Canon's algorithm as a representative of parallel matrix-matrix multiplication. One of the students told me that she liked the algorithm so much that she had dreamt of Canon. She opted to implement the algorithm for her final project, even though it was one of the more difficult options. Though her earlier programming assignments were about average, she did an excellent job on this project and also presented an impressive amount of research on Canon's algorithm for extra credit.

In the Spring semester I will be teaching two different levels of courses: Introduction to Numerical Analysis for senior undergraduates majoring in Computer Science or Mathematics, and Introduction to Matlab for freshmen and sophomores in engineering. I think they will be great fun too.

**Co-Mentoring:** Supervising individual projects is a study of contrasts. Projects have to be simple enough to be within the students ability yet interesting enough so as not to become tedious. Students must focus on a definite problem, that can be solved within a realistic time-frame, yet these should not be stand alone projects They should be relevant in the overall context of the big picture targeted by our research. Finally the level of expertise varies from student to student. The things I enjoy most

about mentoring are these challenges, balancing conflicting goals and one to one interaction with the students.

I am currently co-mentoring a couple of students, who are working on their masters degree in Computer Science at Penn State. One project is on observing linear system characteristics with respect to solver performance. The results of this project would provide important initial input for research on autotuning linear solvers. The second project is on combinatorics and deals with variations of graph embedding techniques. Many graph embedding techniques consider the vertices and edges as a spring-mass model and calculate their placement based on the forces on the strings. One problem that my student is considering is having variable force constants at the edges. These investigations would lead to intersting models for structures such as DNA chromatin fibers where the links are not always homogenous.

I have also had the opportunity to mentor students individually. During Summer 2006 and Fall 2006 semesters at Columbia University, I co-mentored an undergraduate student from the Mechanical Engineering Department on a project involving application of machine learning in selecting linear solvers. My student's task for the summer was to get familiar with Weka, a machine learning software, and then determine the optimal number of fields in a feature vector, that described the linear system, required to obtain an accurate prediction of "good" solvers. His project during the Fall semester was to create a graphical user interface in Java to facilitate user interaction with the solver selection software.

**Substitute Lecturer:** I did not have any official teaching duties during my postdoctoral tenure. However, I volunteered a substitute lecturer for some of the courses taught at the Applied Physics and Applied Mathematics Department at Columbia University. I taught a couple of classes on ordinary differential equations (75 minutes each), in Spring 2005 and Spring 2006, as well a weekly recitation section (1 hour per week). The recitation was designed to help the students understand the material, and involved discussion of homework and other problems beyond the basic lecture.

I believe I did a good job, because in Fall 2006, my postdoctoral advisor Dr. David Keyes, recommended me to another professor in the department as a substitute lecturer. I had the opportunity to teach a few classes for an introductory course on linear algebra. I taught the section on Gaussian Elimination and the section on Singular Value Decomposition.

The courses were designed for freshmen and sophomores in engineering and applied mathematics departments. The lectures dealt with explaining some of the theory of numerical algebra, but mostly their application to engineering problems, and in this context, introducing mathematical software such as MATLAB and Mathematica.

Apart from the experience gained by the substitute lectures, the recitations helped me to hone my teaching skills, in an informal small group setting. Since the recitations were on the day after the lecture, the students had time to think over the material and therefore their questions were more insightful. Answering such question also improved my understanding of the subject, and helped me anticipate areas that students find difficult. This knowledge would be very useful when I teach a similar course.

**Teaching Assistant:** During my graduate studies, I was a teaching assistant in the Department of Computer Science and Engineering, at the Pennsylvania State University, during Fall 2000 and Spring 2001 semesters. I was assigned to teach an undergraduate course titled, " Principles of Programming with Business Applications". This course was oriented to impart necessary computer skills to business majors. The syllabus included the use of database tools like Excel and programming skills

in Visual Basic. My duties involved giving two lectures to three sections (4.5 hours per week) and designing the homework and exams.

The students were from the spectrum of people whose goal is to use computer technologies rather than study computer science. I had to explain the concepts from a "non-computer science" point of view. For example, while introducing a software, I stressed on the tools and facilities that it provided rather than describe the technical details like CPU speed or amount of memory used. Similarly abstract concepts like infinite loops, that computer scientists take for granted, had to be explained with very specific examples. This was my first formal teaching assignment. Though I was developing my teaching skills as I taught, the student evaluations contained positive comments such as my willingness to help them.

With the benefit of hindsight, I feel that this has been a very valuable experience. This course taught me how to convey information from my area of expertise to people from other disciplines, how to sell ideas that are taken for granted in my field to a potentially skeptical audience. These are very necessary skills for a computational scientist like myself, who will spend most of her career in interdisciplinary collaborative research.

**My Ideas on Teaching Scientific Computing:** The area of scientific computing is still in its nascent stage and therefore presents many opportunities for design and evolution of the courses. I think that computational science courses should reflect their multidisciplinary aspects even in introductory stages. I would like to design a course which emphasizes on learning through doing. For example; students would be introduced to an application problem, understand the mathematical model for that application, learn how to discretize, and finally implement it. Depending on the complexity of the application this can be either an undergraduate or graduate course. I would also like to teach graduate level courses on more specialized topics like combinatorial scientific computing and automatic differentiation. In addition to scientific computing courses, I am also qualified to teach introductory courses on traditional topics such as those on Operating Systems, Algorithm Analysis, Introduction to Data structures, Programming Languages, etc.

I believe that a good teacher is not merely adept in explaining a topic, but can also inspire the students to go beyond what was taught, and investigate the subject by themselves. The opportunity to develop the curriculum of a newly emerging area, and train the future generation of scientists, is very exciting and challenging. My experience in teaching enables me to convey the concepts of computer science, not only to students from this field, but also from other disciplines. I enjoy teaching, and feel a certain satisfaction at the thought that what I teach will be applied to aid new contributions to science.