Course Syllabus

MECH 603 – SPECIAL TOPICS: PARALLEL COMPUTING METHODS IN COMPUTATIONAL MECHANICS

Marek Behr

Fall 1999

General Information

Class hours:	Tuesdays and Thursdays, 2:30 – 3:45 pm, ME Building 123
Office hours:	Tuesdays and Wednesdays, 1:30 – 2:30 pm, ME Building 234 and by appointment
Phone:	(713) 737-5657
Email:	`behr@rice.edu`
Web:	`http://www.ruf.rice.edu/~behr/mech603.html`

Course Description

This course will introduce analysis techniques that exploit modern computing engines to solve computational mechanics problems, with emphasis on fluid flows and finite element methods. Sections of the course will concentrate on discretization methods, scalar optimization, parallel computing, and practical aspects of parallel finite element implementations. A prior knowledge of FORTRAN or C programming, and familiarity with the UNIX operating system are prerequisite. Course requirements include class participation and completion of hands-on programming projects.

Course Objectives

Graduate students in the course will:

- Become acquainted with commonly used discretization techniques, and their role in the analysis of continuum mechanics problems
- Be exposed to currently available high performance computing platforms
- Learn to improve the efficiency of their analysis codes by using programming techniques that take into account the memory hierarchy of the processing unit
- Begin to apply parallelization techniques in order to dramatically reduce time required to complete the numerical simulation
- Explore in detail the internal workings of a parallel finite element implementation for fluid flow analysis

Grading

Participants will be graded on the quality and effort put into class participation (20%), homework assignments (40%), and the take-home programming projects (40%).

Attendance

A reasonable attendance is expected, and is essential in order to fulfill the class participation requirement. Every attempt will be made to enable processes of active learning, hopefully alleviating the following:

> Problem with lectures is that information passes from the notes of the instructor to the notes of the student without passing through the minds of either one.

Lecture notes will be posted on the course web page, but will not include notes from the associated class discussion.

Recommended Reading

Although the course will not follow one particular textbook, the following list of books and online articles will help further the understanding of the topics under discussion:

- R.H. Landau and P.J. Fink, "A Scientist's and Engineer's Guide to Workstations and Supercomputers: Coping with Unix, RISC, Vectors, and Programming", John Wiley & Sons, (1992)
- T.J.R. Hughes, "The Finte Element Method. Linear Static and Dynamic Finite Element Analysis", Prentice Hall, (1987)
- additional references given in class

Tentative Course Calendar (updated 1999.10.22)

Aug 31 Course overview. Engineering apps: flow around helicopter.

Sep 2 Equations of fluid dynamics. Engineering apps: flow in a spillway. Discretization: finite differences, volumes, elements, idealization. Components of finite element code. Mesh generation intro.

Sep 7 Mesh generation discussion.

Sep 9 Initial conditions. Time stepping. Linearization. Solving linear equation systems. Postprocessing.

Sep 14 *no class*

Sep 16 Increasing resolution: accuracy, cost. Application requirements. Computer evolution: scalar, vector, parallel.

Sep 21 Memory hierarchy. Data caches. Commercially available CPUs.

Sep 23 Scalar optimization. Memory access patterns. Loop restructuring.

Sep 28 Instruction cache. Automatic optimization. Sun and SGI compilers. Scalar optimization exercises.

Sep 30 Motivation for parallelism. History of parallel computing.

Oct 5 Finding parallelism. Speed-up. Amdahl's law. Taxonomy of parallel computers.

Oct 7 Interconnects. Commercial parallel computers. Programs that support supercomputing: ASCI, HPCMO.

Oct 12 Parallelization via compiler directives. SGI directives. OpenMP.

Oct 14 HP Exemplar access. HP directives.

Oct 21 Scheduling. Parallelization via compiler directives exercises.

Oct 26 Parallelization via language extensions. High Performance Fortran.

Oct 28 Message-passing concepts.

Nov 2 Parallel Virtual Machine.

Nov 4 Message Passing Interface. MPI exercises.

Nov 9 One-sided communication: SHMEM, BSP. Parallel libraries.

Nov 11 Portability issues. Default data types. Mixing FORTRAN and C. Memory allocation. Makefiles.

Nov 16 Analysis tools: Paragraph, VAMPIR, TotalView. Review of finite element program. FE blocks: driver, input/output.

Nov 18 FE blocks: integration, element-level matrices and vectors.

Nov 23 FE blocks: stabilization, turbulence modeling, refinement.

Nov 30 FE blocks: multi-level gather/scatter, assembly.

Dec 2 FE blocks: GMRES solver, matrix-vector product.

Dec 7 Recent developments: VLIW, TERA MTA, Fortran 95, MPI-IO, MPI-2.

An up-to-date version of the schedule will be posted on the course web page as changes occur.

Students with Disabilities

Any student with a disability requiring accommodations in this course is encouraged to contact me after class or during office hours. Additionally, students should contact Disabled Student Services in the Ley Student Center.