Mathematical Sciences 813
Advanced Linear Programming

Course Syllabus (Spring, 1999)

Instructor: Matthew J. Saltzman
Office: Martin O-223   Phone: 656-3185
E-mail: `mjs@clemson.edu`
WWW: `http://www.math.clemson.edu/~mjs`, `http://webct.ces.clemson.edu:8900`

Office Hours: MWF 12:20pm–1:10pm or by appointment

Class Time/Room: MWF 1:25pm–2:15pm, Martin M-302

Description: This course will focus on the solution of large-scale linear optimization problems and systems of linear inequalities. Theoretical topics to be addressed include some fundamental results from convex analysis applied to linear programs, and basic ideas from complexity theory—especially the importance of polynomial-time algorithms. Algorithmic topics include extensions to the simplex method, the "primal-dual" simplex method, interior point algorithms, and decomposition and column- and row-generation methods. Numerical topics include anti-cycling, scaling, and sparse linear algebra techniques for simplex and interior point methods. Applications that give rise to special structures for which these methods are appropriate will be discussed. If there is time, we will consider extensions to the LP problem, such as quadratic and semi-definite programming.

Prerequisites: MthSc 810 or equivalent; knowledge of a computer programming language.

Textbooks:

- *Linear Programming: Foundations and Extensions*, by Robert Vanderbei, Kluwer, 1997 (strongly recommended).
- *Theory of Linear and Integer Programming*, by Alexander Schrijver, Wiley, 1986 (on reserve in the library).
- *Linear Programming*, by Vašek Chvátal, Freeman, 1983 (on reserve in the library).
- *AMPL: A Modeling Language for Mathematical Programming*, by Fourer, Gay, and Kernighan, Boyd and Fraser, 1993 (in the Sun lab or on reserve in the library).

Grading:

Homeworks	25%
Projects	25%
Midterm	20%
Final Exam (cumulative)	30%
Total	100%

Ground Rules:

Assignments: Homework will be assigned each class. Although I will not generally collect assignments, you are welcome to turn in any solutions on which you would like comments. I will give assignments to turn in on occasion. Assignments may involve use of computer packages such as AMPL and CPLEX.

Projects: There will be one programming project and one presentation. The subject of each project is up to the student, subject to the approval of the instructor. The programming project might involve modifying existing codes such as those in Vanderbei or building algorithms using MATLAB, AMPL or callable library routines such as CPLEX or LAPACK. The presentation is a 20-minute talk to the class describing a paper in the open literature.

Exams: Arrangements to make up a missed midterm *must* be made in advance. The *only* exception to this policy will be absence for verifiable medical reasons. There will be *no* makeup for the final exam.

Final Grades: No letter grades will be given during the term. Your final grade will be based on your performance relative to your classmates (i.e., on a curve).

A note on outside sources: It is in the nature of an introductory course that everything you will be asked to do for homework, exams or projects has been done before. The original papers and other textbooks that contain solutions may be available in the library or from other students or faculty. Computer codes may be available in these places or via computer network. In the interest of your own education and in fairness to other students, it seems necessary to provide some ground rules for the use of outside sources.

- Before going to an outside source, you should make a good-faith effort to solve the problem on your own. This is the best way to learn the material, and to find out what you really know and don't know.
- If you do find the solution in an outside source, you should acknowledge the source. This is only fair to the original author, whether we're talking about a book, monograph or even a fellow student. Failure to disclose your sources is plagiarism.
- If you use an outside source, don't copy the result (proof, program, solution) verbatim. Rewrite it in your own words; improve the notation, construct a new example, reorganize the code, etc.. This will maximize the benefit to you of the experience of finding a solution in existing literature.
- Don't check out original sources (particularly journal articles and monographs) from the library during take-home exams. This is only fair to other students in the class who may be led to the same source.