MATH/AMATH 516 $\rightarrow \infty \nabla \sum$ Numerical Optimization $\sum \nabla \infty \leftarrow$ SPRING 2009

WHERE & WHEN? BLD 286 at 9:30-10:20 a.m. on M, W, F.

WHO? Paul Tseng (Office: Padelford C-344; Tel: 543-1177;
Email: tseng@math.washington.edu;
Web: http://www.math.washington.edu/~tseng/516.html)

OFFICE HOURS: Tues 2:00–3:30 (or when I am free, usually M, W, F afternoons).

PREREQUISITE: Linear algebra (matrices, vectors, eigenvalues, etc.), multivariable calculus (partial derivatives, tangent plane, linear approximation, vectors, scalar product, max/min), real analysis (sequences, limits, continuity and differentiability of functions, Taylor polynomial). Prior knowledge of optimization is helpful, but not necessary.

TEXTBOOK: Dimitri P. Bertsekas, *Nonlinear Programming*, 2nd edition, Athena Scientific, 1999.
(see http://world.std.com/~athenasc/index.html for ordering info) Chapters 1, 2, 4; possibly first section of Chapter 5; parts of Chapter 6 on nondifferentiable optimization. Additional topics will be included as time permits. (Other references: *Practical Optimization*, 1981, by Gill, Murray, Wright; *Nonlinear Programming*, 1993, by Bazaraa, Sherali, Shetty; *Numerical Optimization*, 2006, by Nocedal and Wright; *Nonlinear Optimization*, 2006, by Ruszczyński)

GRADING	POINTS (%)	DATE
5 homeworks	80	bi-weekly
project	20	TBA

WHAT? Numerical optimization permeates everything from machine learning to data classification to portfolio selection to robotic control to predicting the 3D structure of proteins. This course gives an overview of numerical optimization, i.e., numerical algorithms for solving optimization problems. Algorithms include those for unconstrained smooth optimization (steepest descent, Newton, conjugate gradient, quasi-Newton), for smooth optimization with convex constraints (Frank-Wolfe, gradient projection, manifold minimization), for smooth optimization with nonconvex constraints (penalty, multiplier, sequential quadratic programming), interior point method, problem decomposition by Lagrangian relaxation. Issues such as convergence, convergence rate, complexity will be studied. Some programming is involved.

NOTE 1: Some of the homeworks will ask you to write a computer program implementing an optimization algorithm of some sort. You can write your program either using Matlab or using Fortran or C/C++. My experience suggests that Matlab is easier to work with. Matlab also has an optimization toolbox which can be used to check the correctness of your program.

NOTE 2: Software for solving optimization problems (some of which are free) are described at
http://plato.la.asu.edu/guide.html
http://www-fp.mcs.anl.gov/otc/Guide/SoftwareGuide/index.html

NOTE 3: For some personal perspectives on the history of numerical optimization, see the book *History of Mathematical Programming: A Collection of Personal Reminiscences*, edited by J. K. Lenstra et al., North-Holland, New York, 1991.

If I had inherited a fortune I should probably not have cast my lot with mathematics.
– Joseph-Louis Lagrange.