Stanford University, Management Science & Engineering (and ICME)

MS&E 318 (CME 338) Large-Scale Numerical Optimization

Instructor: Michael Saunders Spring 2015

Notes 1: **Overview**

Course description

The main algorithms and software for constrained optimization, emphasizing the sparse-matrix methods needed for their implementation. Iterative methods for linear equations and least squares. The simplex method. Basis factorization and updates. Interior methods for convex optimization. The reduced-gradient method, augmented Lagrangian methods, and SQP methods.

Recommended:	MS&E	310, 311, 312, 314, or 315
	CME	108, 200, 302, 304, 334, or 335

3 units, Grading basis ABCD/NP, 4 or 5 homeworks (60%), 1 project (40%), no mid-term or final.

Prerequisites: Basic numerical linear algebra, including LU, QR, and SVD factorizations, and an interest in Matlab, sparse-matrix methods, and gradient-based algorithms for constrained optimization.

`http://stanford.edu/class/msande318/`

Syllabus

1. Overview (problem types, NEOS, Matlab, TOMLAB)
2. Iterative methods for symmetric $Ax = b$ (symmetric Lanczos process, CG, SYMMLQ, MINRES, MINRES-QLP)
3. Iterative methods for unsymmetric $Ax = b$ and least squares (Golub-Kahan process, CGLS, LSQR, LSMR, Craig, Arnoldi process, GMRES)
4. The primal simplex method (phase 1 in practice, basis factorization, updating, crash, scaling, degeneracy)
5. Basis updates (Product-Form, Bartels-Golub, Forrest-Tomlin, Block-LU)
6. LUSOL: A Basis Factorization Package (the engine for MINOS, SQOPT, SNOPT, MILES, PATH, lp_solve)
7. Primal-dual interior methods for LP (CPLEX, HOPDM, IPOPT, KNITRO, LOQO, MOSEK) and convex nonlinear objectives (PDCO), Basis Pursuit, BP Denoising (Lasso, LARS, Homotopy, BPdual)
8. The reduced-gradient method (MINOS part 1)
9. BCL methods (Augmented Lagrangians, LANCELOT)
10. LCL methods (MINOS part 2, Knossos)
11. SQP methods (NPSOL, SQOPT, SNOPT)

1 Optimization problems

We study optimization problems involving linear and nonlinear constraints:

$$\text{NP} \qquad \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \phi(x) \qquad \text{subject to} \quad \ell \le \begin{pmatrix} x \\ Ax \\ c(x) \end{pmatrix} \le u,$$

where $\phi(x)$ is a linear or nonlinear objective function, A is a sparse matrix, $c(x)$ is a vector of nonlinear constraint functions $c_i(x)$, and ℓ and u are vectors of lower and upper bounds. We assume the functions $\phi(x)$ and $c_i(x)$ are *smooth*: they are continuous and have continuous first derivatives (gradients). Sometimes gradients are not available (or too expensive) and we use finite difference approximations. Sometimes we need second derivatives.

We study algorithms that find a *local optimum* for problem NP. Some examples follow. If there are many local optima, the starting point is important.

LP Linear Programming $\qquad \min\ c^Tx$ subject to $\ell \le \begin{pmatrix} x \\ Ax \end{pmatrix} \le u$
MINOS, SNOPT, SQOPT
LSSOL, QPOPT, NPSOL (dense)
CPLEX, Gurobi, LOQO, HOPDM, MOSEK, XPRESS
CLP, lp_solve, SoPlex (open source solvers [7, 30, 48])

QP Quadratic Programming $\qquad \min\ c^Tx + \frac{1}{2}x^THx$ subject to $\ell \le \begin{pmatrix} x \\ Ax \end{pmatrix} \le u$
MINOS, SQOPT, SNOPT, QPBLUR
LSSOL ($H = B^TB$, least squares), QPOPT (H indefinite)
CLP, CPLEX, Gurobi, LANCELOT, LOQO, MOSEK

BC Bound Constraints $\qquad \min\ \phi(x)$ subject to $\ell \le x \le u$
MINOS, SNOPT
LANCELOT, L-BFGS-B

LC Linear Constraints $\qquad \min\ \phi(x)$ subject to $\ell \le \begin{pmatrix} x \\ Ax \end{pmatrix} \le u$
MINOS, SNOPT, NPSOL

NC Nonlinear Constraints $\qquad \min\ \phi(x)$ subject to $\ell \le \begin{pmatrix} x \\ Ax \\ c(x) \end{pmatrix} \le u$
MINOS, SNOPT, NPSOL
CONOPT, LANCELOT
Filter, KNITRO, LOQO (second derivatives)
IPOPT (open source solver [26])

Algorithms for finding local optima are used to construct algorithms for more complex optimization problems: *stochastic*, *nonsmooth*, *global*, *mixed integer*. Excellent examples are SCIP [45] for **MILP** and BARON [3] for **MINLP**.

2 AMPL, GAMS, NEOS

A fuller picture emerges from the list of problem types and solvers handled by the AMPL [2] and GAMS [20] modeling systems and the NEOS server [37]. NEOS is a free service developed originally at Argonne National Laboratory and now hosted by the University of Wisconsin – Madison. It allows us to submit optimization problems in various formats (AMPL, GAMS, CPLEX, MPS, C, Fortran, . . .) to be solved remotely on geographically distributed solver stations.

Recent NEOS usage (up to Dec 31, 2014)**:**

Year	Total jobs	Top solvers	Top inputs
2002	80,000	XpressMP, MINLP, MINOS, SNOPT, SBB	AMPL, GAMS, Fortran
2003	136,000	XpressMP, SNOPT, FortMP, MINOS, MINLP	AMPL, GAMS, Fortran
2004	148,000	MINLP, FortMP, XpressMP, PENNON, MINOS	AMPL, GAMS, Mosel
2005	174,000	Filter, MINLP, XpressMP, MINOS, KNITRO	AMPL, GAMS, Fortran
2006	229,000	Filter, MINOS, SNOPT, XpressMP, MOSEK	AMPL, GAMS, MPS
2007	551,000	SNOPT, MINLP, KNITRO, LOQO, MOSEK	AMPL, GAMS, MPS
2008	322,000	MINOS, Bonmin, KNITRO, SNOPT, IPOPT	AMPL, GAMS, CPLEX
2009	235,000	KNITRO, BPMPD, MINTO, MINOS, SNOPT	AMPL, GAMS, CPLEX
2010	236,000	KNITRO, Concorde, SNOPT, SBB, BARON	AMPL, GAMS, TSP
2011	75,000	SBB, Gurobi, filter, MINLP, XpressMP, MINTO, KNITRO, PATH, MINOS, MOSEK, LOQO, sdpt3	GAMS, AMPL, Matlab_Binary, Fortran, Sparse_SDPA, MPS
2012	353,000	Gurobi, MINOS, CONOPT, SBB, KNITRO, XpressMP, MINTO, SNOPT, Bonmin, Ipopt	AMPL, GAMS, Sparse_SDPA, MPS, Fortran, MOSEL, TSP, CPLEX, C
2013	1,865,000	MINOS, MINLP, KNITRO, Gurobi, Ipopt, SNOPT, csdp, DICOPT, Cbc, XpressMP, MINTO, BARON	AMPL, GAMS, Sparse_SDPA, MPS, TSP, C, CPLEX, Fortran, MOSEL
2014	1,139,000	MINLP, Gurobi, filterMPEC, KNITRO, BARON, MINOS, Cbc, scip, concorde, LOQO, MOSEK	AMPL, GAMS, TSP, Sparse_SDPA, MPS, C, MOSEL, FORTRAN, CPLEX

NEOS problem categories:

BCO Bound Constrained Optimization
BLMVM, L-BFGS-B, TRON

COIP Combinatorial Optimization and Integer Programming
BiqMac, concorde

CP Complementarity Problems
filterMPEC, KNITRO, MILES, NLPEC, PATH

GO Global Optimization
ASA, BARON, Couenne, icos, LINDOGlobal, PGAPack, PSwarm, scip

LP Linear Programming
BDMLP, bpmpd, Clp, Gurobi, MOSEK, OOQP, SoPlex80bit, XpressMP

MILP Mixed Integer Linear Programming
Cbc, feaspump, Gurobi, MINTO, MOSEK, proxy, qsopt_ex, scip, SYMPHONY, XpressMP

MINCO Mixed Integer Nonlinearly Constrained Optimization
AlphaECP, BARON, Bonmin, Couenne, DICOPT, FilMINT, KNITRO, LINDOGlobal, MINLP, SBB, scip

MIOCP Mixed Integer Optimal Control Problems
MUSCOD-II

NCO Nonlinearly Constrained Optimization
CONOPT, filter, Ipopt, KNITRO, LANCELOT, LOQO, LRAMBO, MINOS, MOSEK, PATHNLP, SNOPT

NDO Nondifferentiable Optimization
condor

SDP Semidefinite Programming
csdp, DSDP, penbmi, pensdp, SDPA, sdplr, sdpt3, sedumi

SIO Semi-infinite Optimization
nsips

SLP Stochastic Linear Programming
bnbs, ddsip, sd

SOCP Second Order Conic Programming
MOSEK

UCO Unconstrained Optimization
NMTR

3 Interactive optimization systems

Several systems provide a *graphical user interface* (GUI) or *integrated development environment* (IDE) for mathematical optimization.

MATLAB [33] has an Optimization Toolbox with a selection of dense and sparse solvers (none of the above!, except `ktrlink` uses KNITRO [27]):
`fminbnd, fmincon, fminsearch, fminunc, fseminf, ktrlink`
`fgoalattain, fminimax`
`lsqlin, lsqnonneg, lsqcurvefit, lsqnonlin`
`fzero, fsolve`
`bintprog, linprog, quadprog`

TOMLAB [51] provides a complete optimization environment for MATLAB users. There are many problem types and solvers (CGO, CONOPT, CPLEX, GENO, GP, Gurobi, KNITRO, LGO, LSSOL, MINLP, MINOS, NLPQL, NLSSOL, NPSOL, OQNLP, PENBMI, PENSDP, PROPT, QPOPT, SNOPT, SQOPT, Xpress), a unified input format, automatic differentiation of M-files with MAD, an interface to AMPL, a GUI for selecting parameters and plotting output, and TomSym: a modeling language with complete source transformation.
Note: TOMLAB is available on the Stanford Linux cluster.
See `http://stanford.edu/group/SOL/download.html`

CVX [11] is a MATLAB-based modeling system for convex optimization problems (and for geometric programs). It allows objectives and constraints to be specified using MATLAB syntax.

AIMMS [1] includes solvers for CP, GO, LP, MILP, MINLP, NLP, QCP, QP. Its modeling language has historical connections to GAMS.

COMSOL Optimization Lab [10] provides the LP and NLP capabilities of SNOPT to users of COMSOL Multiphysics [9]. Currently, most problem classes are handled by SNOPT. A Nelder-Mead "simplex algorithm" is included for unconstrained optimization without derivatives.

Frontline Systems [19] provides Excel spreadsheet and Visual Basic access to optimizers for many problem classes: LP, QP, SOCP, MILP, NLP, GO, NDO.

GAMS IDE [20] provides an IDE for GAMS Windows installations.

IBM ILOG CPLEX Optimization Studio [25] provides an IDE for LP, MILP, and constraint-programming applications, along with a high-level modeling language (OPL).

4 More optimization systems

We cite a few systems in order to include them in the references: COIN-OR [8], CPLEX [24], Gurobi [22], Lindo [29], MOSEK [35], PENOPT [42], SeDuMi [46], TFOCS [50].

5 Sparse linear systems

Underlying almost all of the optimization algorithms is the need to solve a sequence of linear systems $Ax = b$ (where "x" is likely to be a *search direction*). We will study some of the linear system software below. Most codes are implemented in Fortran 77. MA57 includes an F90 interface, and newer HSL packages [23] are in F90. UMFPACK is written in C.

When A is a sparse matrix, MATLAB uses MA57 for `ldl(A)` and UMFPACK for $A\backslash b$ and `lu(A)` (both direct methods).

If A is a sparse matrix (or a linear operator defined by a function handle), MATLAB has the following iterative methods for solving $Ax = b$ or $\min \|Ax - b\|_2$: `bicg`, `bicgstab`, `bicgstabl`, `cgs`, `gmres`, `lsqr`, `minres`, `pcg`, `qmr`, `symmlq`, `tfqmr`.

Some of the iterative solvers are available in F77, F90, and MATLAB from SOL [47]. PETSc [43] provides many direct and iterative solvers for truly large problems.

Direct methods factorize sparse A into a product of triangular matrices that should be sparse and well defined even if A is singular or ill-conditioned.

LUSOL [21, 31, 32] Square or rectangular $Ax = b$, $A = LU$, plus updating

MA48 [16] Square or rectangular $Ax = b$, $A = LU$

MA57 [15] Symmetric $Ax = b$, $A = LDL^T$ or LBL^T (MATLAB's `[L,D,P] = ldl(A)`)

MUMPS [36] Square $Ax = b$, $A = LU$, LDL^T or LBL^T (massively parallel)

PARDISO [41] Square $Ax = b$ (shared memory)

SuperLU [14, 28, 49] Square $Ax = b$ (uniprocessor or shared or distributed memory)

SuiteSparseQR [13] Rectangular sparse QR (MATLAB's `[Q,R,P] = qr(A)`)

UMFPACK [52, 12] Square $Ax = b$ (MATLAB's `[L,U,P,Q] = lu(A)`)

Iterative methods regard A as a black box (a *linear operator*) for computing matrix-vector products Ax and sometimes A^Ty for given x and y.

CG, PCG [33, 43] Symmetric positive-definite $Ax = b$

SYMMLQ [38, 47, 43] Symmetric nonsingular $Ax = b$ (may be indefinite)

MINRES, MINRES-QLP [38, 47, 43, 4, 5, 18, 6] Symmetric $Ax = b$ (may be indefinite or singular)

GMRES [44, 43] Unsymmetric $Ax = b$

CGLS, LSQR, LSMR, LSRN [39, 40, 47, 43, 17, 34] $Ax = b,\ \min \|Ax-b\|_2^2,\ \min \left\| \begin{pmatrix} A \\ \delta I \end{pmatrix} x - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2^2$

References

[1] AIMMS modeling environment. `http://www.aimms.com/aimms`.

[2] AMPL modeling system. `http://www.ampl.com`.

[3] BARON global optimization system. `http://archimedes.scs.uiuc.edu/baron/baron.html`.

[4] S.-C. Choi. *Iterative Methods for Singular Linear Equations and Least-Squares Problems.* PhD thesis, ICME, Stanford University, Dec 2006.

[5] S.-C. Choi, C. C. Paige, and M. A. Saunders. MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems. *SIAM J. Sci. Comput.*, 33(4):1810–1836, 2011. `http://stanford.edu/group/SOL/software.html`.

[6] S.-C. Choi and M. A. Saunders. Algorithm 937: MINRES-QLP for symmetric and Hermitian linear equations and least-squares problems. *ACM Trans. Math. Softw.*, 40(2):Article 16, 12 pp., 2014. `http://stanford.edu/group/SOL/software.html`.

[7] CLP open source LP, QP, and MILP solver. `http://www.coin-or.org/projects/Clp.xml`.

[8] COIN-OR: Computational Infrastructure for Operations Research. `http://www.coin-or.org`.

[9] COMSOL AB. `http://www.comsol.com`.

[10] COMSOL Optimization Lab. `http://www.comsol.com/products/optlab`.

[11] CVX: MATLAB software for Disciplined Convex Programming. `http://cvxr.com`.

[12] T. A. Davis. *Direct Methods for Sparse Linear Systems.* Fundamentals of Algorithms. SIAM, Philadelphia, 2006.

[13] T. A. Davis. Algorithm 915, SuiteSparseQR: Multifrontal multithreaded rank-revealing sparse QR factorization. *ACM Trans. Math. Softw.*, 38(1):8:1–8:22, 2011.

[14] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Anal. Appl.*, 20(3):720–755, 1999.

[15] I. S. Duff. MA57: a Fortran code for the solution of sparse symmetric definite and indefinite systems. *ACM Trans. Math. Software*, 30(2):118–144, 2004. See `ldl` in MATLAB.

[16] I. S. Duff and J. K. Reid. The design of MA48: a code for the direct solution of sparse unsymmetric linear systems of equations. *ACM Trans. Math. Software*, 22(2):187–226, 1996.

[17] D. C.-L. Fong and M. A. Saunders. LSMR: An iterative algorithm for least-squares problems. *SIAM J. Sci. Comput.*, 33(5):2950–2971, 2011. `http://stanford.edu/group/SOL/software.html`.

[18] D. C.-L. Fong and M. A. Saunders. CG versus MINRES: An empirical comparison. *SQU Journal for Science*, 17(1):44–62, 2012. `http://stanford.edu/group/SOL/reports/SOL-2011-2R.pdf`.

[19] Frontline Systems, Inc. spreadsheet modeling system. `http://www.solver.com`.

[20] GAMS modeling system. `http://www.gams.com`.

[21] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Maintaining LU factors of a general sparse matrix. *Linear Algebra and its Applications*, 88/89:239–270, 1987.

[22] Gurobi optimization system for linear and integer programming. `http://www.gurobi.com`.

[23] The HSL Mathematical Software Library. `http://www.hsl.rl.ac.uk/`.

[24] IBM ILOG CPLEX optimizer. `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`.

[25] IBM ILOG CPLEX Optimization Studio. `https://www.ibm.com/developerworks/downloads/ws/ilogcplex/`.

[26] IPOPT open source NLP solver. `http://www.coin-or.org/projects/Ipopt.xml`.

[27] KNITRO optimization software. `http://www.ziena.com`.

[28] X. S. Li and J. W. Demmel. SuperLU-DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Trans. Math. Software*, 29(2):110–140, 2003.

[29] Lindo Systems optimization software. `http://www.lindo.com`.

[30] lp_solve open source LP and MILP solver. `http://groups.yahoo.com/group/lp_solve/`.

[31] LUSOL sparse matrix package. `http://stanford.edu/group/SOL/software.html`.

[32] LUSOL mex interface (Nick Henderson, ICME, Stanford University). `https://github.com/nwh/lusol_mex`, 2011.

[33] MATLAB matrix laboratory. `http://www.mathworks.com`.

[34] X. Meng, M. A. Saunders, and M. W. Mahoney. LSRN: a parallel iterative solver for strongly over- or underdetermined systems. *SIAM J. Sci. Comput.*, 36(2):C95–C118, 2014.

[35] MOSEK Optimization Software. `http://www.mosek.com/`.

[36] MUMPS: a multifrontal massively parallel sparse direct solver. `http://mumps.enseeiht.fr/`.

[37] NEOS server for optimization. `http://www.neos-server.org/neos/`.

[38] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12:617–629, 1975.

[39] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8(1):43–71, 1982.

[40] C. C. Paige and M. A. Saunders. Algorithm 583; LSQR: Sparse linear equations and least-squares problems. *ACM Trans. Math. Software*, 8(2):195–209, 1982.

[41] PARDISO parallel sparse solver. `http://www.pardiso-project.org`.

[42] PENOPT optimization systems for nonlinear programming, bilinear matrix inequalities, and linear semidefinite programming. `http://www.penopt.com`.

[43] PETSc toolkit for scientific computation. `http://www.mcs.anl.gov/petsc`.

[44] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, second edition, 2003.

[45] SCIP mixed integer programming solver. `http://scip.zib.de/`.

[46] SeDuMi optimization system for linear programming, second-order cone programming, and semidefinite programming. `http://sedumi.ie.lehigh.edu`.

[47] SOL downloadable software. `http://stanford.edu/group/SOL/software.html`.

[48] SoPlex linear programming solver. `http://soplex.zib.de/`.

[49] SuperLU software for sparse unsymmetric systems. `http://crd.lbl.gov/~xiaoye/SuperLU/`.

[50] TFOCS: Templates for First-Order Conic Solvers. `http://cvxr.com`.

[51] TOMLAB optimization environment for MATLAB. `http://tomopt.com`.

[52] UMFPACK solver for sparse $Ax = b$. `http://www.cise.ufl.edu/research/sparse/umfpack`.