Experiments with linear and nonlinear optimization using Quad precision

Michael Saunders and Ding Ma

MS&E and ICME, Stanford University

1st Fletcher-Powell Lecture

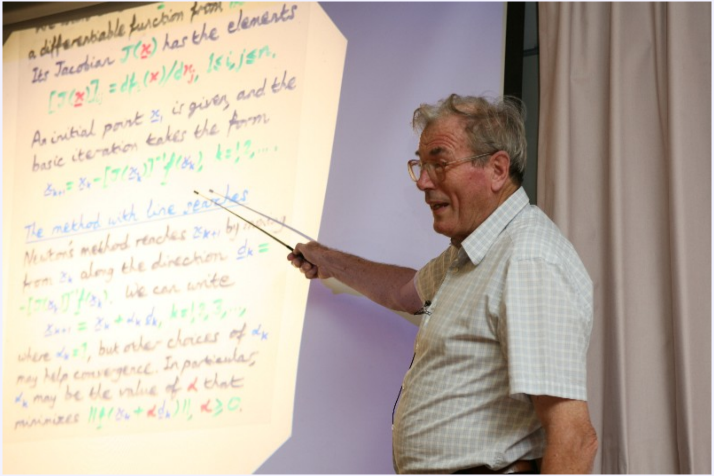
26th Biennial Numerical Analysis Conference

University of Strathclyde, Glasgow

June 23-26, 2015

Presented at CME 510, Stanford, Oct 15, 2015

Roger Fletcher FRS and Mike Powell FRS



Roger Fletcher — pioneer

Michael Friedlander's *first slides* from
IMA Conference on Numerical Analysis and Optimisation, Birmingham, Sep 2014

Nonlinearly constrained optimization

$$\min\ f(x) \quad \text{st} \quad c(x) = 0$$

R. Fletcher (1970) Smooth primal penalty function

$$\min\ f(x) - c(x)^T y(x) + \tfrac{1}{2}\sigma \|c(x)\|^2$$

Roger Fletcher — pioneer

Michael Friedlander's *last slides* from
IMA Conference on Numerical Analysis and Optimisation, Birmingham, Sep 2014

$$\min \tfrac{1}{2}\|Ay-g\|^2 + \sigma c^T y + \tfrac{1}{2}\delta^2\|y\|^2 \quad \Leftrightarrow \quad \begin{bmatrix} I & A \\ A^T & -\delta^2 I \end{bmatrix} \begin{bmatrix} r \\ y \end{bmatrix} = \begin{bmatrix} g \\ \sigma c \end{bmatrix} \quad (SQD)$$

R. Fletcher (1970) A class of methods for nonlinear programming with termination and convergence properties. *Integer and Nonlinear Programming* (Abadie, ed.)

R. Fletcher and S. A. Lill (1971) A class of methods for nonlinear programming. II. Computational experience. *Nonlinear Programming* (Rosen, Mangasarian, and Ritter, eds.)

R. Fletcher (1972) A class of methods for nonlinear programming III: rates of convergence. *Numerical Methods for Nonlinear Optimization* (Lootsma, ed.)

R. Fletcher (1973) An exact penalty function for nonlinear programming with inequalities. *Math. Prog. 5*

Mike Powell — individualist

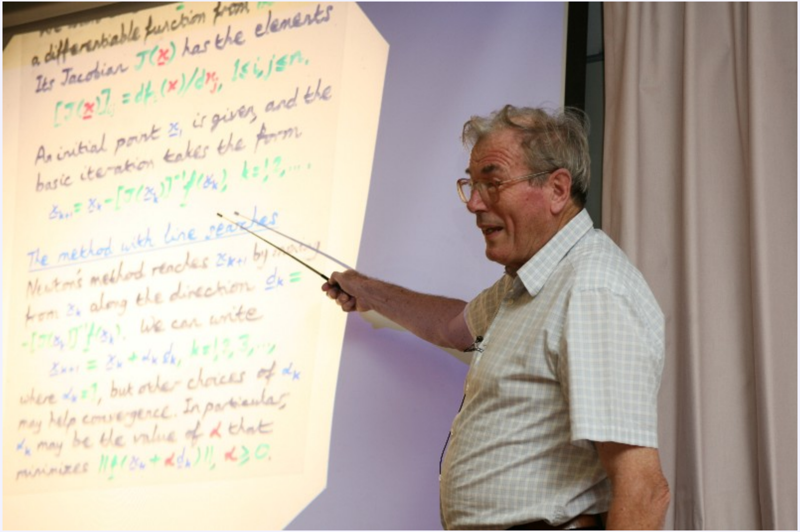
- Powell 1969 penalty function: $\min\ f(x) + \frac{1}{2}\sigma \left\| c(x) - \theta \right\|^2$
- Hestenes 1969 method of multipliers: $\min\ f(x) - c(x)^T y + \sigma \left\| c(x) \right\|^2$
- Rockafellar 1973 generalization for $c(x) \geq 0$:

$$\min\ f(x) + \frac{1}{2}\sigma \left\| c(x) - \theta \right\|_{-}^2$$

M. J. D. Powell (1974) Ch I. Introduction to constrained optimization.
Numerical Methods for Constrained Optimization (Gill and Murray, eds.)

R. Fletcher (1974) Ch VIII. Methods related to Lagrangian functions.
Same book. This chapter explains the above.

Mike Powell — individualist



Abstract

For challenging numerical problems, William Kahan has said that “default evaluation in Quad is the humane option” for reducing the risk of embarrassment due to rounding errors. Fortunately the gfortran compiler now has a `real(16)` datatype. This is the humane option for producing Quad-precision software. It has enabled us to build a Quad version of MINOS.

The motivating influence has been increasingly large LP and NLP problems arising in systems biology. Flux balance analysis (FBA) models of metabolic networks generate multiscale problems involving some large data values in the constraints (stoichiometric coefficients of order 10,000) and some very small values in the solution (chemical fluxes of order 10^{-10}). Standard solvers are not sufficiently accurate, and exact simplex solvers are extremely slow. Quad precision offers a reliable and practical compromise even via software. On a range of multiscale LP examples we find that 34-digit Quad floating-point achieves primal and dual infeasibilities of order 10^{-30} when “only” 10^{-15} is requested.

Partially supported by the
National Institute of General Medical Sciences
of the National Institutes of Health (NIH)
Award U01GM102098

Coauthor Ding Ma at INFORMS 2014

Coauthor Ding Ma at INFORMS 2014

Unexpected excitement in Zhenjiang, China (13 Dec 2014)

Bart De Moor met President Xi JinPing already (Oct 2009)

Vice-Rector for International Policy at KU Leuven, Belgium

William Kahan, LA/Opt seminar, Thursday Oct 13, 2011

Desperately Needed Remedies for the Undebuggability of Large Floating-Point Computations in Science and Engineering

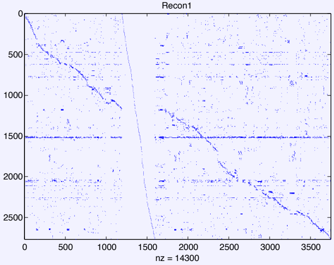
Motivation

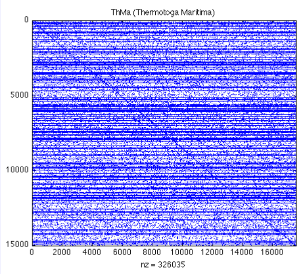
Stoichiometric matrices S

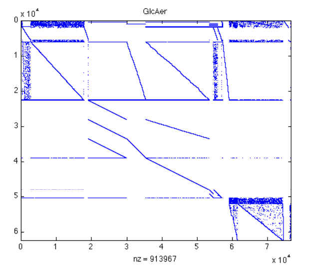
$$2H_2 + O_2 \rightarrow 2H_2O$$

	·	·	·	W	·	·	·
H_2	·	·	·	-2	·	·	·
O_2	·	·	·	-1	·	·	·
H_2O	·	·	·	2	·	·	·

chemicals $\times$ reactions







2800×3700 | 15000×18000 | 62000×77000

In Constraint Based Reconstruction and Analysis (COBRA), a biochemical network, which is inherently multiscale, is represented by a stoichiometric matrix S with m rows corresponding to metabolites (chemicals) and n columns representing reactions. Mathematically, S is part of the ODE that governs the time-evolution of concentrations in the network:

$$\frac{d}{dt}x(t) = Sv(t), \tag{1}$$

where $x(t) \in \mathbf{R}^m$ is a vector of time-dependent concentrations and $v(t) \in \mathbf{R}^n$ is a vector of reaction fluxes. With the objective of maximizing growth rate at steady state, the following LP is constructed:

$$\begin{aligned} \max_{v} \quad & c^T v && \text{(2a)} \\ \text{s.t.} \quad & Sv = 0, && \text{(2b)} \\ & l \leq v \leq u, && \text{(2c)} \end{aligned}$$

where growth is defined as the biosynthetic requirements of experimentally determined biomass composition, and biomass generation is a set of reaction fluxes linked in the appropriate ratios.

ME models (FBA with coupling constraints)

Flux Balance Analysis (FBA) has been used by Ines2012ME for the first integrated stoichiometric multiscale model of metabolism and macromolecular synthesis for *Escherichia coli* K12 MG1655. The model modifies (2) by adding constraints that couple enzyme synthesis and catalysis reactions to (2b). Coupling constraints of the form

$$c_{\min} \le \frac{v_i}{v_j} \le c_{\max} \tag{3}$$

become linear constraints

$$c_{\min} v_j \le v_i, \quad v_i \le c_{\max} v_j \tag{4}$$

for various pairs of fluxes v_i, v_j. They are linear approximations of nonlinear constraints and make S in (2b) even less well-scaled because of large variations in reaction rates. Quad precision is evidently more appealing in this case.

Coupling constraints

Two fluxes could be related by

$$0.0001 \leq \frac{v_1}{v_2} \leq 10000. \tag{5}$$

Lifting approach: due to Yuekai Sun, ICME

We can decompose these constraints into sequences of constraints involving auxiliary variables with reasonable coefficients. If the second inequality in (5) were presented to our implementation as $v_1 \leq 10000 v_2$, we would transform it to two constraints involving an auxiliary variable s_1:

$$v_1 \leq 100 s_1, \qquad s_1 \leq 100 v_2. \tag{6}$$

If the first inequality in (5) were presented as $v_1 \geq 0.0001 v_2$, we would leave it alone, but the equivalent inequality $10000 v_1 \geq v_2$ would be transformed to

$$v_2 \leq 100 s_2, \qquad s_2 \leq 100 v_1.$$

The desirability of Quad precision

"Carrying somewhat more precision in the arithmetic than twice the precision carried in the data and available for the result will vastly reduce embarrassment due to roundoff-induced anomalies."

"Default evaluation in Quad is the humane option."

— *William Kahan*

Methods for achieving Quad precision

Hand-code calls to auxiliary functions

Even `q = qdotdd(v,w)` needs several double functions

`twosum, split, twoproduct sum2, dot2`

to compute double x, y

and hence quad result `q = quad(x) + quad(y)`

Double-double datatype (≈ 32 digits)

QD: `http://crd-legacy.lbl.gov/~dhbailey/mpdist/`

C++ with interfaces to C++ and F90

DDFUN90: entirely F90

Minor changes to source code

Quad datatype (≈ 34 digits)

Some f90 compilers such as gfortran

Again minor changes to source code

We use this humane approach to quad implementation

System and Methods

quadMINOS

The GNU GCC compilers make Quad available via 128-bit data types. We have therefore been able to make a Quad version of the Fortran 77 linear and nonlinear optimization solver MINOS using the gfortran compiler[1] with `real(8)` changed to `real(16)` everywhere.

Double is implemented in hardware, while Quad is a software library.

Our aim is to explore combined use of the Double and Quad MINOS simplex solvers for the solution of large multiscale linear programs. We seek greater efficiency than is normally possible with exact simplex solvers.

[1] GNU Fortran (GCC) 4.6.2 20111019 on Mac OS X

quadSNOPT

In the f90 implementations of SQOPT and SNOPT, we select one of the modules

```
snPrecision32.f90
snPrecision64.f90
snPrecision128.f90
```

For example, `snPrecision128.f90`:

```
module snModulePrecision
  implicit none
  public
  integer(4), parameter :: ip = 8, rp = 16  ! quad precision
end module snModulePrecision
```

Later:

```
module sn50lp
  use  snModulePrecision, only : ip, rp
  subroutine s5solveLP ( x, y )
  real(rp),     intent(inout) :: x(nb), y(nb)
```

MINOS and quadMINOS

The primal simplex solver in MINOS includes

- geometric-mean scaling of the constraint matrix
- the EXPAND anti-degeneracy procedure
- partial pricing (but no steepest-edge pricing, which would generally reduce total iterations and time)
- Basis LU factorizations and updates via LUSOL

NEOS Statistics

NEOS

Free optimization solvers
via Argonne National Lab
(now Univ of Madison, Wisconsin)

NEOS Solver Statistics for 2 years 1 Jan 2012 -- 1 Jan 2014

Total Jobs 2218537

Solver	Submissions	Solver	Submissions	Solver	Submissions	Solver	Submissions
MINOS	774695	filter	8123	PATHNLP	1423	PGAPack	350
MINLP	514475	Couenne	7996	L-BFGS-B	1351	sd	124
KNITRO	276896	BDMLP	6691	ASA	1326	xpress	123
Gurobi	130334	PATH	6298	NLPEC	1281	Cplex	32
SNOPT	48281	bpmpd	6121	RELAX4	1265	DONLP2	3
Ipopt	46305	BLMVM	6005	condor	993	LGO	3
CONOPT	38331	NMTR	5248	SYMPHONY	871		
XpressMP	32688	AlphaECP	5201	sedumi	833		
MINTO	30367	OOQP	5147	icos	808		
csdp	28662	LANCELOT	5045	DSDP	805		
DICOPT	25524	MUSCOD-II	4973	Glpk	785		
BARON	25138	FilMINT	4523	PSwarm	784		
Cbc	23752	feaspump	3731	sdplr	741		
scip	21529	TRON	2237	Clp	735		
SBB	21466	MILES	1853	penbmi	573		
MOSEK	21192	LRAMBO	1774	bnbs	547		
Bonmin	19144	qsopt_ex	1718	nsips	516		
LOQO	16095	SDPA	1669	FortMP	492		
concorde	9652	sdpt3	1582	ddsip	489		
LINDOGlobal	8459	filterMPEC	1438	pensdp	447		

NEOS Solver Statistics for 2 years 1 Jan 2012 -- 1 Jan 2014

Total Jobs 2218537

Category	Submissions
nco	1170088
kestrel	533865
milp	190822
minco	117723
lp	81472
sdp	35312
go	29246
cp	23210
co	9676
bco	9585
uco	5248
miocp	4973
lno	4155
slp	1160
ndo	993
sio	516
socp	206

Input	Submissions
AMPL	1850882
GAMS	274585
SPARSE_SDPA	31266
MPS	15319
TSP	9652
Fortran	7811
CPLEX	7396
C	7375
MOSEL	4998
MATLAB_BINARY	2364
LP	1496
DIMACS	1148
ZIMPL	1078
SDPA	805
SMPS	671
MATLAB	402
SDPLR	332

Algorithm and Implementation

3-step procedure

1. Cold start Double MINOS with scaling and somewhat strict settings, save basis
2. Warm start Quad MINOS with scaling and tighter Feasibility and Optimality tols, save basis
3. Warm start Quad MINOS without scaling but tighter LU tols

MINOS runtime options for Steps 1–3

	Default Double	Step1 Double	Step2 Quad	Step3 Quad
Scale option	2	2	2	0
Feasibility tol	1e-6	1e-7	1e-15	1e-15
Optimality tol	1e-6	1e-7	1e-15	1e-15
LU Factor tol	100.0	10.0	10.0	5.0
LU Update tol	10.0	10.0	10.0	5.0

Table: Three pilot models from Netlib, eight Mészáros *problematic* LPs, and three ME biochemical network models. Dimensions of $m \times n$ constraint matrices A and size of the largest optimal primal and dual variables x^*, y^*.

model	m	n	nnz(A)	$\max \lvert A_{ij} \rvert$	$\lVert x^* \rVert_\infty$	$\lVert y^* \rVert_\infty$
pilot4	411	1000	5145	3e+04	1e+05	3e+02
pilot	1442	3652	43220	2e+02	4e+03	2e+02
pilot87	2031	4883	73804	1e+03	2e+04	1e+01
de063155	853	1488	5405	8e+11	3e+13	6e+04
de063157	937	1488	5551	2e+18	2e+17	6e+04
de080285	937	1488	5471	1e+03	1e+02	3e+01
gen1	770	2560	64621	1e+00	3e+00	1e+00
gen2	1122	3264	84095	1e+00	3e+00	1e+00
gen4	1538	4297	110174	1e+00	3e+00	1e+00
l30	2702	15380	64790	1e+00	1e+09	4e+00
iprob	3002	3001	12000	1e+04	3e+02	1e+00
TMA_ME	18210	17535	336302	2e+04	6e+00	1e+00
GlcAerWT	68300	76664	926357	8e+05	6e+07	2e+07
GlcAlift	69529	77893	928815	3e+05	6e+07	2e+07

Table: Itns and runtimes in secs for Step 1 (Double MINOS) and Steps 2–3 (Quad MINOS). Pinf and Dinf = $\log_{10}$ final maximum primal and dual infeasibilities. Problem iprob is infeasible. Bold figures show Pinf and Dinf at the end of Step 3. $\text{Pinf}/\left\|x^*\right\|_\infty$ and $\text{Dinf}/\left\|y^*\right\|_\infty$ are all $O(10^{-30})$ or smaller, even though only $O(10^{-15})$ was requested. This is an unexpectedly favorable empirical finding.

model	Itns	Times	Final objective	Pinf	Dinf
pilot4	1571	0.1	−2.5811392602e+03	−05	−13
	6	0.0	−2.5811392589e+03	−39	−31
	0	0.0	−2.5811392589e+03	**–**	**−30**
pilot	16060	5.7	−5.5739887685e+02	−06	−03
	29	0.7	−5.5748972928e+02	–	−27
	0	0.2	−5.5748972928e+02	**–**	**−32**
pilot87	19340	15.1	3.0171038489e+02	−09	−06
	32	2.2	3.0171034733e+02	–	−33
	0	1.2	3.0171034733e+02	**–**	**−33**

model	Itns	Times	Final objective	Pinf	Dinf
de063155	921	0.0	1.8968704286e+10	−13	+03
	78	0.1	9.8830944565e+09	–	−17
	0	0.0	9.8830944565e+09	–	**−24**
de063157	488	0.0	1.4561118445e+11	+20	+18
	476	0.5	2.1528501109e+07	−27	−12
	0	0.0	2.1528501109e+07	–	**−12**
de080285	418	0.0	1.4495817688e+01	−09	−02
	132	0.1	1.3924732864e+01	−35	−32
	0	0.0	1.3924732864e+01	–	**−32**
gen1	369502	205.3	−1.6903658594e−08	−06	−12
	246428	9331.3	1.2935699163e−06	−12	−31
	2394	81.6	1.2953925804e−06	**−45**	**−30**
gen2	44073	60.0	3.2927907828e+00	−04	−11
	1599	359.9	3.2927907840e+00	–	−29
	0	10.4	3.2927907840e+00	–	**−32**
gen4	45369	212.4	1.5793970394e−07	−06	−10
	53849	14812.5	2.8932268196e−06	−12	−30
	37	10.4	2.8933064888e−06	**−54**	**−30**

model	Itns	Times	Final objective	Pinf	Dinf
l30	1229326	876.7	9.5266141574e−01	−10	−09
	275287	7507.1	−7.5190273434e−26	−25	−32
	0	0.2	−4.2586876849e−24	**−24**	**−33**
iprob	1087	0.2	2.6891551285e+03	+02	−11
	0	0.0	2.6891551285e+03	+02	−31
	0	0.0	2.6891551285e+03	+02	**−28**
TMA_ME	12225	37.1	8.0051076669e−07	−06	−05
	685	61.5	8.7036315385e−07	−24	−30
	0	6.7	8.7036315385e−07	**–**	**−31**
GlcAerWT	62856	9707.3	−2.4489880182e+04	+04	−05
	5580	3995.6	−7.0382449681e+05	−07	−26
	4	60.1	−7.0382449681e+05	**−19**	**−21**
GlcAlift	134693	14552.8	−5.1613878666e+05	−03	−01
	3258	1067.1	−7.0434008750e+05	−09	−26
	2	48.1	−7.0434008750e+05	**−20**	**−22**

Multiscale NLPs

Systems biology FBA problems with variable μ
(Palsson Lab, UC San Diego, 2014)

Analog filter design for a personalized hearing aid
(Jon Dattorro, Stanford, 2014)

ME models with nonlinear constraints

As coupling constraints are often functions of the organism's growth rate μ, Lerman et al. (UCSD) consider growth-rate optimization nonlinearly with the single μ as the objective instead of via a linear biomass objective function. Nonlinear constraints of the form

$$\frac{v_i}{v_j} \leq \mu \tag{7}$$

represented as

$$v_i \leq \mu v_j \tag{8}$$

are added to (2b), where v_i, v_j, μ are all variables. Constraints (8) are linear if μ is fixed at a specific value μ_k. Lerman et al. employ a binary search to find the largest $\mu_k \in [\mu_{\min}, \mu_{\max}]$ that keeps the associated LP feasible. Thus, the procedure requires reliable solution of a sequence of related LPs.

tinyME

Nonlinear FBA formulation, Laurence Yang, UCSD, Dec 2014

$$
\begin{array}{rl}
\max & \mu \\
\text{st} & \mu Ax + Bx = 0 \\
& Sx \qquad = b \\
& \text{bounds on } x
\end{array}
\quad \equiv \quad
\begin{array}{rl}
\max & \mu \\
\text{st} & \mu Ax + w = 0 \\
& Bx - w = 0 \\
& Sx \qquad = b \\
& \text{bounds on } x\text{, no bounds on } w
\end{array}
$$

- Tiny example: $\approx 2500 \times 3000$
- $\mu = x_1$ and the first columns of A, B are empty
- Constraints are linear if μ is fixed   suggests binary search on sequence of LPs
  25 LP subproblems would give 8 digits   (really need quad Simplex)
- Instead, apply quad MINOS   LCL method = Linearly Constrained Lagrangian
  6 NLP subproblems (with linearized constraints) give 20 digits

Quadratic convergence of major iterations (Robinson 1972)

```
 quadMINOS  5.6      (Nov 2014)

Begin tinyME-NLP   cold start NLP with mu = mu0
Itn      304 -- linear constraints satisfied.
Calling funcon.  mu =  0.80000000000000000000000000000000004
nnCon, nnJac, neJac        1073        1755        2681
funcon  sets    2681   out of    2681   constraint gradients.
funobj  sets       1   out of       1   objective  gradients.

Major minor  step     objective  Feasible Optimal   nsb   ncon penalty
    1   304T 0.0E+00  8.00000E-01 6.1E-03 2.1E+03     0      4 1.0E+02
    2   561T 1.0E+00  8.00000E-01 2.6E-14 3.2E-04     0     46 1.0E+02
    3    40T 1.0E+00  8.28869E-01 5.4E-05 3.6E-05     0     87 1.0E+02
    4      7 1.0E+00  8.46923E-01 1.2E-05 2.9E-06     0     96 1.0E+02
    5      0 1.0E+00  8.46948E-01 4.2E-10 2.6E-10     0     97 1.0E+02
    6      0 1.0E+00  8.46948E-01 7.9E-23 1.2E-20     0     98 1.0E+01

EXIT -- optimal solution found
```

```
EXIT -- optimal solution found

Problem name                   tinyME
No. of iterations                 912   Objective value      8.4694810579E-01
No. of major iterations             6   Linear objective     0.0000000000E+00
Penalty parameter            1.000000   Nonlinear objective  8.4694810579E-01
No. of calls to funobj             98   No. of calls to funcon             98
No. of superbasics                  0   No. of basic nonlinears           786
No. of degenerate steps             0   Percentage                       0.00
Max x       (scaled)       12 5.6E-01   Max pi      (scaled)      103 8.3E+05
Max x                    1020 6.1E+01   Max pi                    103 9.7E+03
Max Prim inf(scaled)        0 0.0E+00   Max Dual inf(scaled)        9 2.9E-14
Max Primal infeas           0 0.0E+00   Max Dual infeas             9 1.3E-18
Nonlinear constraint violn    1.9E-20

funcon called with nstate =   2
Final value of   mu =  0.84694810578563166175146802332321527

Time for solving problem                   13.50 seconds
```

ME 2.0

Large FBA and FVA problems, Laurence Yang, UCSD, Sep 2015

FBA model iJL1678:	$71,000 \times 80,000$ LP
Quad MINOS cold start:	$\sim$ 3 hours
FVA problems:	min and max individual variables v_j

		Double CPLEX		Quad MINOS	
Reaction	Protein	$v_{\min}$	$v_{\max}$	$v_{\min}$	$v_{\max}$
translation_b0169	RpsB	30.715011	30.712581	30.719225	30.719225
translation_b0025	RibF	0.212807	0.211712	0.210161	0.210161
translation_b0071	LeuD	0.303304	0.765585	0.303634	0.303634
translation_b0072	LeuC	0.303304	0.681146	0.303634	0.303634

Analog filter design

Hearing aid design, Jon Dattorro, Stanford

Frequencies	$\omega = 2\pi[30,\ 45,\ \ldots,\ 12000,\ 16000]$
Data	$g = [1,\ 1.3,\ \ldots,\ 44.7,\ 79.4]$ (filter magnitudes)
Variables	$U,\ V,\ u,\ v \geq 0$

$$U_i(u) \equiv 1 + u_1\omega_i^2 + u_2\omega_i^4 + \cdots + u_\eta\omega_i^{2\eta} \qquad \eta = 2, 3, \ldots, 8$$

$$V_i(v) \equiv 1 + v_1\omega_i^2 + v_2\omega_i^4 + \cdots + v_\eta\omega_i^{2\eta}$$

Find u, v so that $\dfrac{V_i}{U_i} \approx g_i^2 \quad \Rightarrow \quad g_i^2\dfrac{U_i}{V_i} \approx 1$

Analog filter design

NLP1

$$\underset{\beta\geq 1,\, U,V\geq 0, u,v}{\text{minimize}} \quad \beta$$

$$\text{subject to} \quad \frac{1}{\beta} \leq g_i^2 \frac{U_i}{V_i} \leq \beta\,, \qquad \omega_i \in \Omega$$

where

$$U_i(u) \equiv 1 + u_1\omega_i^2 + u_2\omega_i^4 + \cdots$$

$$V_i(v) \equiv 1 + v_1\omega_i^2 + v_2\omega_i^4 + \cdots$$

19 frequencies ω_i (Hz):

$$\omega = 2\pi[\,30 \;\; 45 \;\; 60 \;\; 90 \;\; 125 \;\; 187 \;\; 250 \;\; 375 \;\; 500 \;\; 750 \;\; \ldots \;\; 1000 \;\; 1500 \;\; 2000 \;\; 3000 \;\; 4000 \;\; 6000 \;\; 8000 \;\; 12000 \;\; 16000\,]^{\mathrm{T}}$$

19 filter magnitudes:

$$g = [\,1. \;\; 1.2589 \;\; 2.2387 \;\; 2.5119 \;\; 2.8184 \;\; 5.0119 \;\; 5.0119 \;\; 7.9433 \;\; 10. \;\; 6.3096 \;\; \ldots \;\; 6.3096 \;\; 4.4668 \;\; 6.3096 \;\; 10. \;\; 7.9433 \;\; 14.125 \;\; 25.119 \;\; 44.668 \;\; 79.433\,]^{\mathrm{T}}$$

Analog filter design

NLP2

$$\begin{array}{lll} \underset{\beta\geq 1,\, U,V\geq 0,u,v}{\text{minimize}} & \beta & \\ \text{subject to} & \beta V_i - \gamma_i U_i \;\; \geq 0\,, & \omega_i \in \Omega \\ & \beta U_i - \gamma_i^{-1} V_i \geq 0 & \\ & U_i \; - \omega_i^2 u_1 - \omega_i^4 u_2 = 1 & \\ & V_i \; - \omega_i^2 v_1 \; - \omega_i^4 v_2 = 1 & \end{array}$$

$\gamma_i \equiv g_i^2$, β, U_i, V_i appear nonlinearly

$\beta \equiv \beta_0$ fixed $\Rightarrow$ the problem is an LP

$\Rightarrow$ can do binary search with LP solver (CVX, Gurobi)

Proof for bisection of a quasiconcave monotonic function: p210 Dattorro 2015, *Convex Optimization* † *Euclidean Distance Geometry* 2ϵ, Meboo `http://stanford.edu/group/SOL/Books/0976401304.pdf`

Filter design, $\eta = 2$

With $\beta \equiv \beta_0 = 5.0$ fixed, the problem is an LP

The LP and NLP2 solve as follows:

	major itns	minor itns	f/g evaluations	Pinf	Dinf
LP	3	9	7		
NLP2	13	33	79	—	-23

$$\beta = 2.7837077182, \quad u_1 = 1.333433 \times 10^{-6}, \quad u_2 = 0.0$$
$$v_1 = 4.853544 \times 10^{-5}, \quad v_2 = 2.942739 \times 10^{-13}$$

Improvement if the frequencies ω_i are measured in kHz instead of Hz:

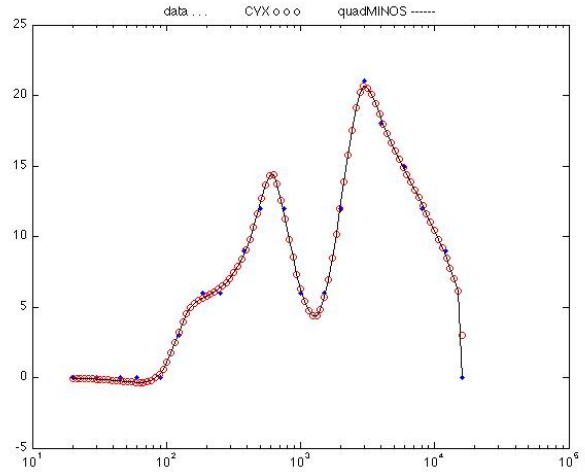
	major itns	minor itns	f/g evaluations	Pinf	Dinf
LP	2	8	5		
NLP2	12	19	39	—	-31

$$\beta = 2.7837077182, \quad u_1 = 1.333433 \times 10^{-0}, \quad u_2 = 0.0$$
$$v_1 = 4.853544 \times 10^{+1}, \quad v_2 = 2.942739 \times 10^{-1}$$

Filter design, $\eta = 2:8$

η	β_0	β^*	Pinf	Dinf
2	5.0	4.2368	−30	−32
3	5.0	2.5154	−32	−33
4	3.0	1.4227	−30	−34
5	1.4	1.3637	−23	−35
6	1.37	1.2625	−25	−34
7	1.2	1.1053	−07	−29
8	1.1	1.0809	−29	−34

Figure: $\eta = 8$
Blue dots = given data
Red circles = fit by CVX/gurobi
Black curve = fit by quadMINOS



Filter design, $\eta = 8$ (more quadratic convergence)

With $\beta \equiv \beta_0 = 1.1$ fixed, the problem is an LP

	Scale	major itns	minor itns	f/g evaluations	Pinf	Dinf
LP	Yes		44		-42	-01
NLP2	Yes	9	86	186	-03	-35
NLP2	No	6	12	41	-18	-16

NLP2 with scaling:

```
 Major minor   step      objective  Feasible Optimal    nsb    ncon penalty BSwap
     1     0T 0.0E+00  1.10000E+00 1.1E-42 1.0E-01      0       4 1.0E+02     0
     2      6 8.1E-01  1.06179E+00 2.7E-16 1.3E+02      0      12 1.0E+02     0
     3    41T 8.2E-03  1.12705E+00 1.8E-17 4.6E+02      2      96 1.0E+02     0
     4     14 1.0E+00  1.09696E+00 6.5E-51 2.4E+02      1     127 1.0E+02     2
     5     24 1.0E+00  1.08217E+00 2.4E-18 1.2E+01      1     173 1.0E+02     1
     6      1 1.0E+00  1.08079E+00 9.3E-19 1.2E-06      0     183 1.0E+02     1
     7      0 1.0E+00  1.08089E+00 1.4E-22 3.4E-08      0     184 1.0E+01     0
     8      0 1.0E+00  1.08089E+00 4.0E-30 2.4E-15      0     185 1.0E+00     0
     9      0 1.0E+00  1.08089E+00 4.0E-30 3.1E-36      0     186 1.0E-01     0
 EXIT -- optimal solution found
```

Filter design, $\eta = 8$

With $\beta \equiv \beta_0 = 1.1$ fixed, the problem is an LP

	Scale	major itns	minor itns	f/g evaluations	Pinf	Dinf
LP	Yes		44		-42	-01
NLP2	Yes	9	86	186	-03	-35
NLP2	No	6	12	41	-18	-16

NLP2 with no scaling:

```
Major minor  step      objective  Feasible Optimal   nsb   ncon penalty BSwap
    1     0T 0.0E+00  1.08089E+00 3.7E-20 8.2E-19     0      3 1.0E-02     0
    2      6 1.0E+00  1.08089E+00 1.6E-34 4.9E-17     4     27 1.0E-02     0
    3      2 1.0E+00  1.08089E+00 1.6E-34 7.0E-18     3     31 1.0E-03     3
    4      2 1.0E+00  1.08089E+00 1.6E-34 9.3E-17     2     35 1.0E-04     1
    5      1 1.0E+00  1.08089E+00 1.6E-34 9.3E-17     2     38 1.0E-05     1
    6      1 1.0E+00  1.08089E+00 1.6E-34 9.3E-17     2     41 1.0E-06     1
EXIT -- the current point cannot be improved upon
```

LPnetlib test problems

Unexpectedly high accuracy in Double and Quad

62 classic LP problems (ordered by file size)

afiro	scfxm1	ship04s	pilotja
stocfor1	bandm	seba	ship08l
adlittle	e226	grow15	nesm
scagr7	grow7	fffff800	ship12l
sc205	etamacro	scfxm3	cycle
share2b	agg	ship04l	greenbea
recipe	scsd1	ganges	greenbeb
vtpbase	standata	sctap2	80bau3b
share1b	beaconfd	grow22	d2q06c
bore3d	gfrdpnc	ship08s	woodw
scorpion	stair	stocfor2	d6cube
capri	scrs8	pilotwe	pilot
brandy	shell	ship12s	wood1p
scagr25	scfxm2	25fv47	pilot87
sctap1	pilot4	sierra	
israel	scsd6	czprob	

LP experiment

MINOS double precision `real(8)` $\epsilon =$ 2.2e-16

```
Feasibility tol = 1e-8
Optimality  tol = 1e-8
```

Compare with MINOS quad precision `real(16)` $\epsilon =$ 1.9e-35

```
Feasibility tol = 1e-15
Optimality  tol = 1e-15
```

In both cases:

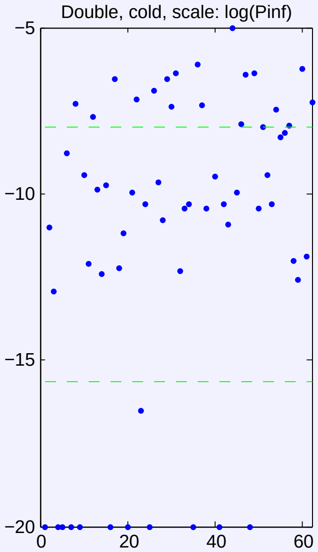
- Cold start with scaling and other defaults
- Warm start, no scaling, LU rook pivoting
- Plot $\log_{10}$ of Pinf and Dinf$/(1 + \|y^*\|_\infty)$

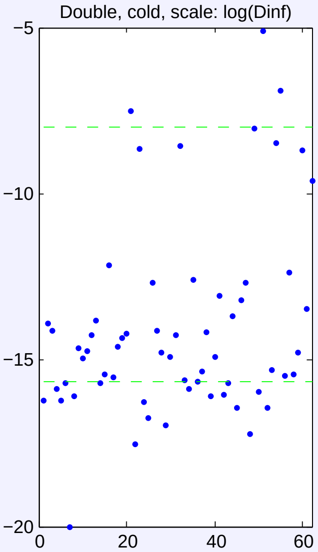
Max primal and dual infeasibilities: Double precision, cold start, scaling

```
Scale option 2
Feasibility tol 1e-8
Optimality  tol 1e-8

LU Partial Pivoting
LU Factor tol   100.0
LU Update tol    10.0
```

$\epsilon = 2.2\text{e-}16$



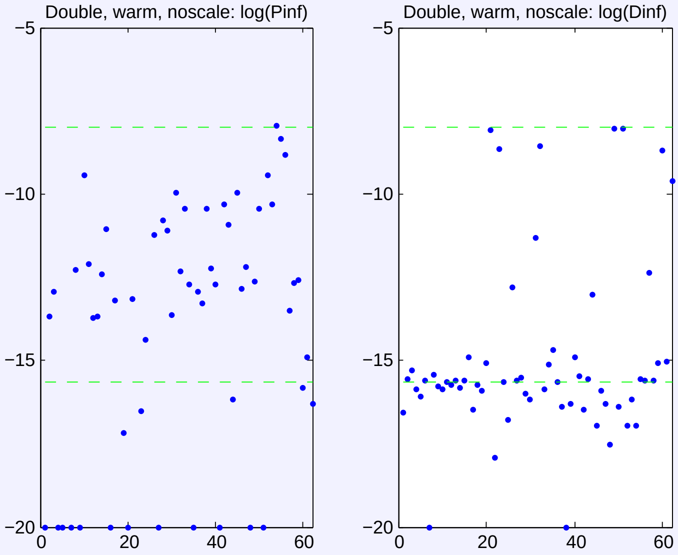


Max primal and dual infeasibilities: Double precision, warm start, no scaling

```
Scale option 0
Feasibility tol 1e-8
Optimality  tol 1e-8

LU Rook Pivoting
LU Factor tol     1.9
LU Update tol     1.9
```

$\epsilon = 2.2\text{e-}16$

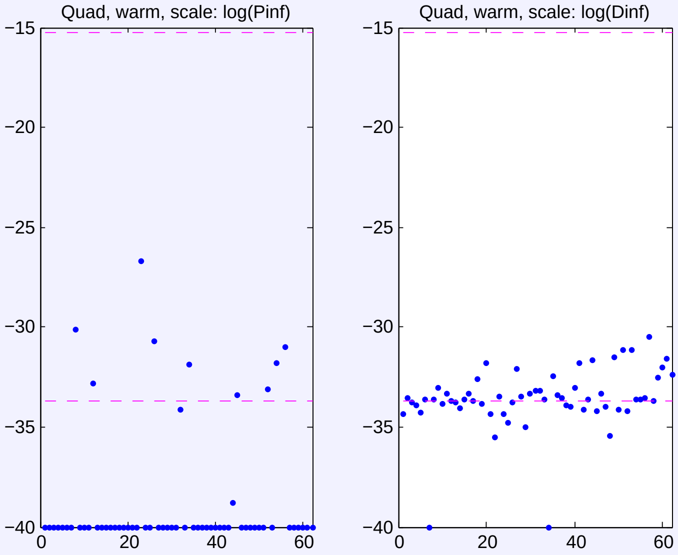


Max primal and dual infeasibilities: Quad precision, cold start, scaling

```
Scale option 2
Feasibility tol 1e-15
Optimality  tol 1e-15

LU Partial Pivoting
LU Factor tol  100.0
LU Update tol   10.0
```

$\epsilon =$ 1.9e-35

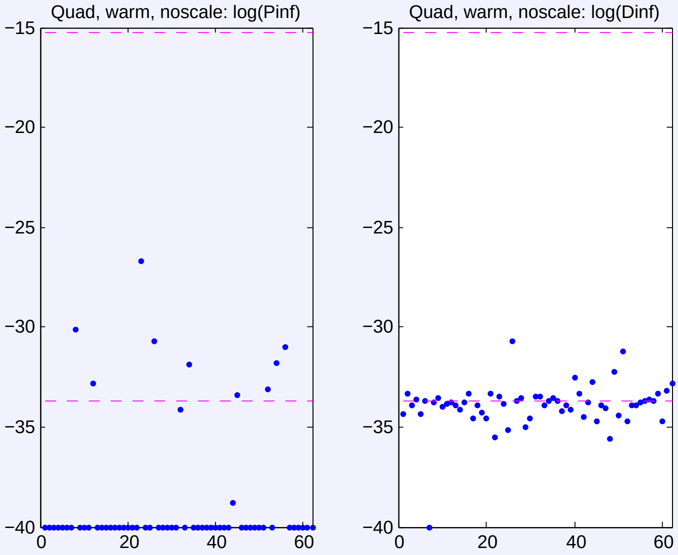


Max primal and dual infeasibilities: Quad precision, warm start, no scaling

```
Scale option 0
Feasibility tol 1e-15
Optimality  tol 1e-15

LU Rook Pivoting
LU Factor tol     1.9
LU Update tol     1.9
```

$\epsilon = 1.9\text{e-}35$



Philosophy

Philosophy

- Humor is mankind's greatest blessing. – Mark Twain

Philosophy

- Humor is mankind's greatest blessing. – Mark Twain
- There are three rules for writing a great English novel. Unfortunately noone knows what they are. – Somerset Maugham (?)

Philosophy

- Humor is mankind's greatest blessing. – Mark Twain
- There are three rules for writing a great English novel. Unfortunately noone knows what they are. – Somerset Maugham (?)
which	✗	We will cover some variations which may be useful.
, which	✓	We will cover some variations, which may be useful.
that	✓	We will cover some variations that may be useful.

Philosophy

- Humor is mankind's greatest blessing. – Mark Twain
- There are three rules for writing a great English novel.
  Unfortunately noone knows what they are. – Somerset Maugham (?)
- **which** ✗ We will cover some variations which may be useful.

  , which ✓ We will cover some variations, which may be useful.

  that ✓ We will cover some variations that may be useful.
- If the glove won't fit, you must acquit.

Philosophy

- Humor is mankind's greatest blessing. – Mark Twain
- There are three rules for writing a great English novel. Unfortunately noone knows what they are. – Somerset Maugham (?)
- **which** ✗ We will cover some variations which may be useful.

  , which ✓ We will cover some variations, which may be useful.

  that ✓ We will cover some variations that may be useful.
- If the glove won't fit, you must acquit.
- If the comma's omitted, the which is wicked.

Philosophy

- ✗ Thanks for the quick reply.

  ✓ Thanks for your quick reply.

  ✓✓ Peter, thanks for your quick reply.

Philosophy

- ✗ Thanks for the quick reply.
  ✓ Thanks for your quick reply.
  ✓✓ Peter, thanks for your quick reply.

- ✗ Oct 15
  ✓ Thurs, Oct 15

Philosophy

- The purpose of our lives is to be happy. – Dalai Lama

Philosophy

- The purpose of our lives is to be happy. – Dalai Lama
- Can humour (not satire) be the antidote to extremism?
  It would be great to think so.

Philosophy

- The purpose of our lives is to be happy. – Dalai Lama
- Can humour (not satire) be the antidote to extremism?
  It would be great to think so.
- You have to think anyway, so why not think big? – Donald Trump

Philosophy

- The purpose of our lives is to be happy. – Dalai Lama
- Can humour (not satire) be the antidote to extremism?
  It would be great to think so.
- You have to think anyway, so why not think big? – Donald Trump
- Metabolic networks will keep getting bigger (genome-scale up to whole human).

Philosophy

- The purpose of our lives is to be happy. – Dalai Lama
- Can humour (not satire) be the antidote to extremism?
  It would be great to think so.
- You have to think anyway, so why not think big? – Donald Trump
- Metabolic networks will keep getting bigger (genome-scale up to whole human).
- Urge chip-makers to implement hardware quad precision.

Conclusions

Conclusions

Just as double-precision floating-point hardware revolutionized scientific computing in the 1960s, the advent of quad-precision data types (even in software) brings us to a new era of greatly improved reliability in optimization solvers.

Reference

Ding Ma and Michael Saunders (2014). Solving multiscale linear programs using the simplex method in quadruple precision. http://stanford.edu/group/SOL/multiscale/papers/quadLP3.pdf

Conclusions

Just as double-precision floating-point hardware revolutionized scientific computing in the 1960s, the advent of quad-precision data types (even in software) brings us to a new era of greatly improved reliability in optimization solvers.

Reference

Ding Ma and Michael Saunders (2014). Solving multiscale linear programs using the simplex method in quadruple precision. http://stanford.edu/group/SOL/multiscale/papers/quadLP3.pdf

Special thanks

- George Dantzig, born 100 years ago (8 Nov 1914)
- William Kahan, IEEE floating-point standard, including Quad
- William Kahan, Boulder.pdf (2011)
- GNU gfortran
- Ronan Fleming, Ines Thiele (Luxembourg)
- Bernhard Palsson, Josh Lerman, Teddy O'Brien, Laurence Yang (UCSD)
- Ed Klotz (IBM CPLEX), Yuekai Sun, Jon Dattorro (Stanford)
- Alison Ramage, Iain Duff

FAQ

- Is quadMINOS available? Yes, in the openCOBRA toolbox
  `http://opencobra.github.io/cobratoolbox/`
- Can quadMINOS be called from Matlab or Tomlab? Yes via system call (not Mex)
- Is quadMINOS available in GAMS? Soon Yes
- How about AMPL? No, but should be feasible
- Is there a quadSNOPT? Yes, in f90 SNOPT9 we can change 1 line
- Can CPLEX / Gurobi / Mosek / . . . help? Yes, they can provide Presolve and Warm start, especially from GAMS
- Will Quad hardware eventually be standard? We hope so