Large-scale Linear and Nonlinear Optimization in Quad Precision

Ding Ma and Michael Saunders

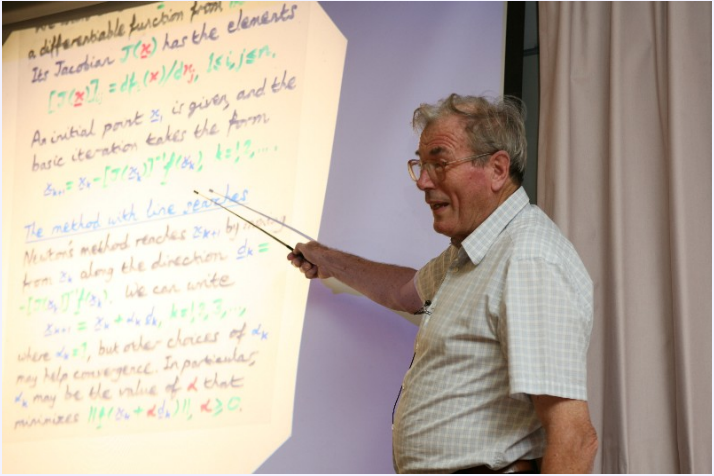
MS&E and ICME, Stanford University

US-Mexico Workshop on Optimization and its Applications
Mérida, Yucatán, Mexico, Jan 4–8, 2016

From INFORMS SFO Nov 9–12, 2014
1st Fletcher-Powell Lecture, Strathclyde, Glasgow, June 23-26, 2015

Coauthor Ding Ma at INFORMS 2014

Roger Fletcher FRS and Mike Powell FRS



Abstract

For challenging numerical problems, William Kahan has said that "default evaluation in Quad is the humane option" for avoiding severe unexpected error in floating-point computations. The IEEE 754-2008 standard includes Quad precision (about 34 significant digits) and is provided by some compilers as a software library. For example, gfortran provides a real(16) datatype. This is the humane option for producing Quad-precision software.

We describe experiments on multiscale linear and nonlinear optimization problems using Double and Quad implementations of MINOS. On a range of examples we find that Quad MINOS achieves exceptionally small primal and dual infeasibilities (of order 1e-30) when "only" 1e-15 is requested. The motivation has been large multiscale LP and NLP problems arising in systems biology (flux balance analysis models of metabolic networks).
Standard solvers are not sufficiently accurate, and exact simplex solvers are extremely slow.
Quad precision offers a reliable compromise.

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

Motivation

Multiscale LPs in systems biology

Normal approach for LP solvers (simplex or interior)

- Scale (to reduce large matrix values)
- Solve with Feasibility/Optimality tols = 1e−6 say
- Unscale

Difficulty

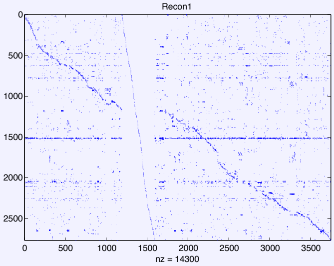
- Unscaling magnifies residuals
- Solution may be far from feasible or optimal

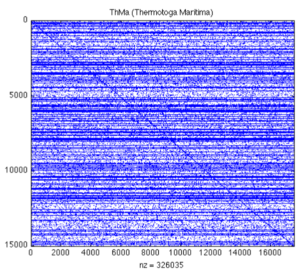
Stoichiometric matrices S

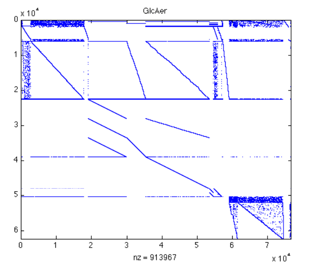
$$2H_2 + O_2 \rightarrow 2H_2O$$

	.	.	.	W	.	.	.
H_2	.	.	.	-2	.	.	.
O_2	.	.	.	-1	.	.	.
H_2O	.	.	.	2	.	.	.

chemicals $\times$ reactions







2800×3700 15000×18000 62000×77000

Constraint Based Reconstruction and Analysis (COBRA)

A biochemical network (inherently multiscale) is represented by a stoichiometric matrix S with m rows corresponding to metabolites (chemicals) and n columns representing reactions. S is part of the ODE that governs the time-evolution of concentrations:

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

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. The objective of maximizing growth rate at steady state leads to an LP:

$$\max_v \quad c^T v \tag{2a}$$
$$\text{s.t.} \quad Sv = 0, \tag{2b}$$
$$l \le v \le u, \tag{2c}$$

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 Ines Thiele (2012) for the first integrated stoichiometric multiscale model of Metabolism and macromolecular Expression (ME) for *Escherichia coli* K12 MG1655. Added coupling constraints

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

become linear constraints

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

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.

Coupling constraints

Two fluxes could be related by

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

Lifting approach (Yuekai Sun, ICME, 2012)
Transform into sequences of constraints involving auxiliary variables with reasonable coefficients. The second inequality in (5) becomes $v_1 \leq 10000v_2$, which is equivalent to

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

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

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

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 (now version 5.2.0)

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
```

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

quadMINOS $\equiv$ MINOS with `real(8)` $\rightarrow$ `real(16)`

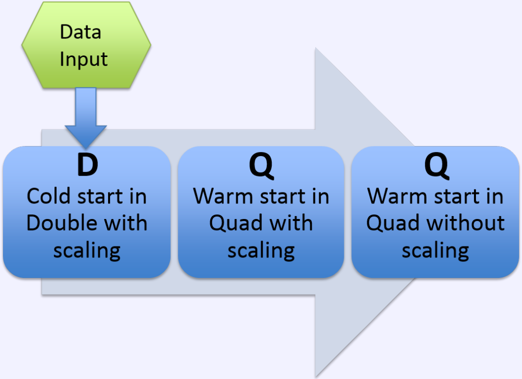
`eps = 2.22e-16` $\rightarrow$ `eps = 1.93e-34`

DQQ procedure

Step D: Double MINOS
Step Q1: Quad MINOS
Step Q2: Quad MINOS with no scaling

DQQ procedure

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



MINOS runtime options for DQQ procedure

	Default	Step D	Step Q1	Step Q2
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. $\text{Pinf}/\|x^*\|_\infty$ and $\text{Dinf}/\|y^*\|_\infty$ 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**

DRR procedure

Step D: Double MINOS
Step R1: Refinement
Step R2: Refinement with no scaling

Plausible alternative to DQQ

DRR procedure

1. Cold start Double MINOS with scaling and somewhat strict settings
2. Warm start with scaling and Iterative Refinement and tighter Featol/Opttol
3. Warm start with no scaling but Iterative Refinement and tighter LU tols

- We need Quad residuals for $Bx_B = b - Nx_n$ after LU
  and for $By = a$, $B^T y = c_B$ each iteration

DRR procedure

1. Cold start Double MINOS with scaling and somewhat strict settings
2. Warm start with scaling and Iterative Refinement and tighter Featol/Opttol
3. Warm start with no scaling but Iterative Refinement and tighter LU tols

- We need Quad residuals for $Bx_B = b - Nx_n$ after LU
  and for $By = a$, $B^T y = c_B$ each iteration
- Quad $r = a - By$ needs $r \leftarrow r - y_k B_k$ (qaxpy)
  Compiler converts B to Quad every iteration

DRR procedure

1. Cold start Double MINOS with scaling and somewhat strict settings
2. Warm start with scaling and Iterative Refinement and tighter Featol/Opttol
3. Warm start with no scaling but Iterative Refinement and tighter LU tols

- We need Quad residuals for $Bx_B = b - Nx_n$ after LU
  and for $By = a$, $B^T y = c_B$ each iteration
- Quad $r = a - By$ needs $r \leftarrow r - y_k B_k$ (qaxpy)
  Compiler converts B to Quad every iteration
- Quad $r = c_B - B^T y$ needs Quad dotproducts (qdot)
  Again, compiler converts B to Quad every iteration

DRR procedure

1. Cold start Double MINOS with scaling and somewhat strict settings
2. Warm start with scaling and Iterative Refinement and tighter Featol/Opttol
3. Warm start with no scaling but Iterative Refinement and tighter LU tols

- We need Quad residuals for $Bx_B = b - Nx_n$ after LU
  and for $By = a$, $B^T y = c_B$ each iteration
- Quad $r = a - By$ needs $r \leftarrow r - y_k B_k$ (qaxpy)
  Compiler converts B to Quad every iteration
- Quad $r = c_B - B^T y$ needs Quad dotproducts (qdot)
  Again, compiler converts B to Quad every iteration
- James Ho (1975) SRR procedure?

LPnetlib test problems

Unexpectedly high accuracy in 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	

DRR procedure on LPnetlib problems

$$\begin{aligned} \text{Pinf} &= \text{max Primal infeasibility} \\ \text{Dinf} &= \text{max Dual infeasibility}/(1 + \|y^*\|_\infty) \end{aligned}$$

MINOS stops when

$$\begin{aligned} \text{Pinf} &\leq \text{Feasibility tol} \\ \text{Dinf} &\leq \text{Optimality tol} \end{aligned}$$

Default 1e−6

Default 1e−6

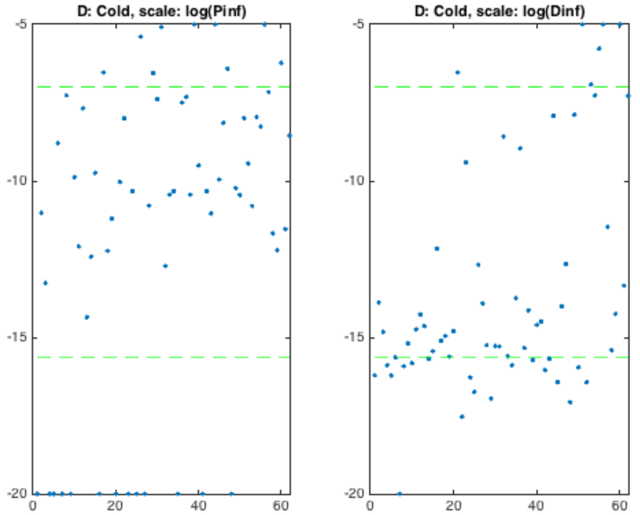
Plot $\log_{10}$(Pinf) and $\log_{10}$(Dinf) for steps D, R1, R2

Primal/dual infeasibilities: Step D: Double MINOS, cold start, scale

```
Scale option 2
Feasibility tol 1e-7
Optimality  tol 1e-7

LU Partial Pivoting
LU Factor tol    10.0
LU Update tol    10.0
Quad refinement     0
```

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

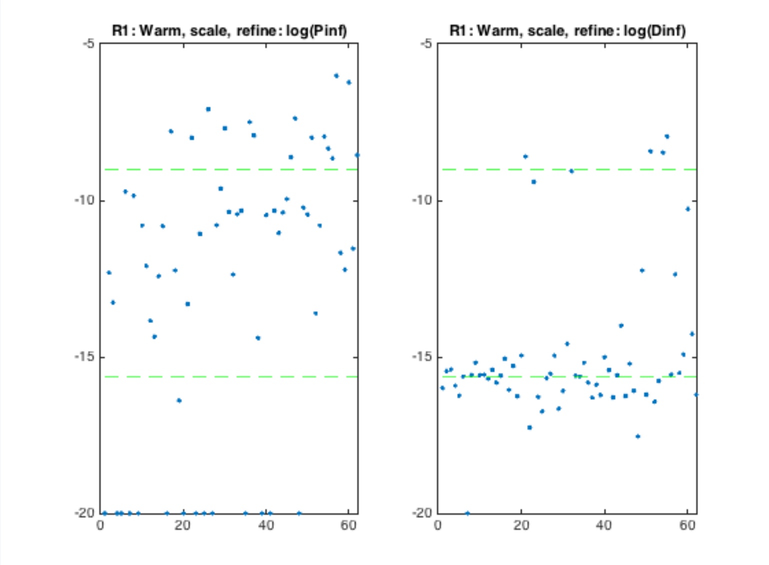


Primal/dual infeasibilities: Step R1: Double MINOS, warm start, scale, refine

```
Scale option 2
Feasibility tol 1e-9
Optimality  tol 1e-9

LU Partial Pivoting
LU Factor tol     1.9
LU Update tol     1.9
Quad refinement     1
```

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

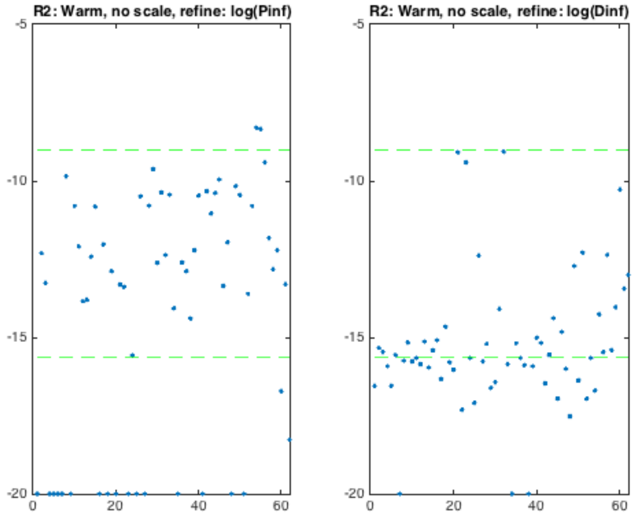


Primal/dual infeasibilities: Step R2: Double MINOS, warm start, no scale, refine

```
Scale option 0
Feasibility tol 1e-9
Optimality  tol 1e-9

LU Partial Pivoting
LU Factor tol     1.9
LU Update tol     1.9
Quad refinement     1
```

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



DQQ procedure on LPnetlib problems

$$\begin{aligned}\text{Pinf} &= \text{max Primal infeasibility}\\ \text{Dinf} &= \text{max Dual infeasibility}/(1+\|y^*\|_\infty)\end{aligned}$$

MINOS stops when $\begin{aligned}\text{Pinf} &\le \text{Feasibility tol}\\ \text{Dinf} &\le \text{Optimality tol}\end{aligned}$

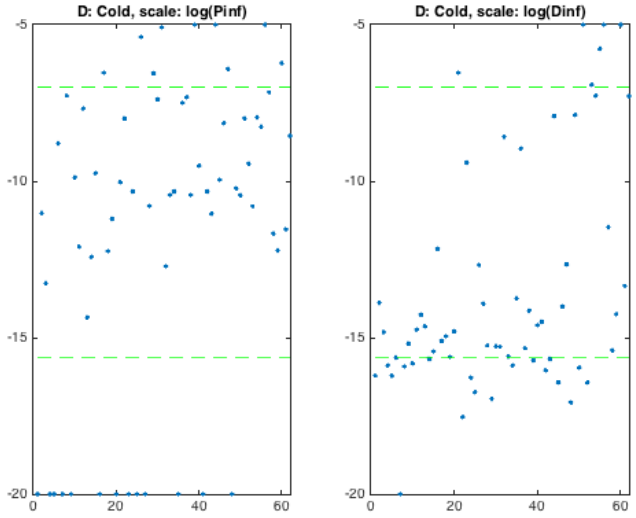
Plot $\log_{10}(\text{Pinf})$ and $\log_{10}(\text{Dinf})$ for steps D, Q1, Q2

Primal/dual infeasibilities: Step D: Double MINOS, cold start, scale (repeat)

```
Scale option 2
Feasibility tol 1e-7
Optimality  tol 1e-7

LU Partial Pivoting
LU Factor tol    10.0
LU Update tol    10.0
Expand freq    100000
```

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

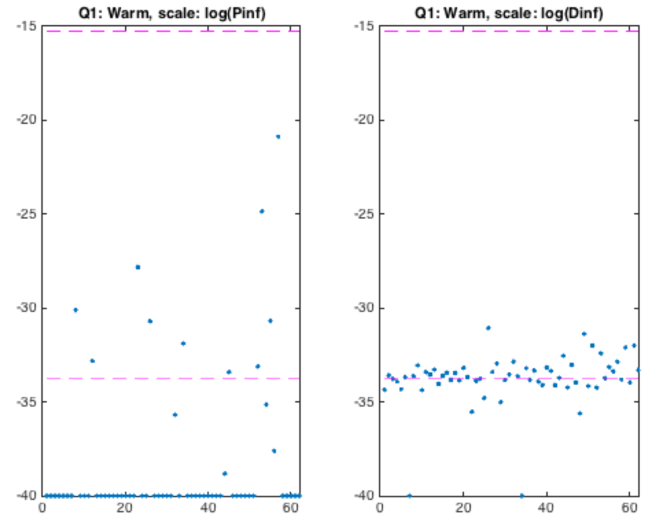


Primal/dual infeasibilities: Step Q1: Quad MINOS, warm start, scale

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

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

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

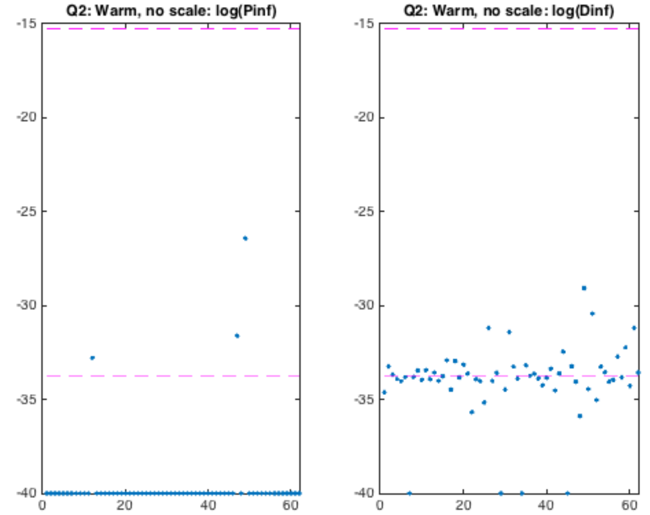


Primal/dual infeasibilities: Step Q2: Quad MINOS, warm start, no scale

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

LU Partial Pivoting
LU Factor tol     5.0
LU Update tol     5.0
```

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



Multiscale NLPs

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

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

- 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 =  0.8000000000000000000000000000000004
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

Reaction	Protein	Double CPLEX $v_{\min}$	Double CPLEX $v_{\max}$	Quad MINOS $v_{\min}$	Quad MINOS $v_{\max}$
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

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.
- ✗ Jan 5
  ✓ Tues, Jan 5

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.

References

- Y. Sun, R. M. T. Fleming, I. Thiele, and M. A. Saunders. Robust flux balance analysis of multiscale biochemical reaction networks, *BMC Bioinformatics* 14:240, 2013, 6 pp.
- Ding Ma and Michael Saunders (2015). Solving multiscale linear programs using the simplex method in quadruple precision, Proceedings of NAO-III, Muscat, Oman, Jan 2014, Springer.

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.

References

- Y. Sun, R. M. T. Fleming, I. Thiele, and M. A. Saunders. Robust flux balance analysis of multiscale biochemical reaction networks, *BMC Bioinformatics* 14:240, 2013, 6 pp.
- Ding Ma and Michael Saunders (2015). Solving multiscale linear programs using the simplex method in quadruple precision, Proceedings of NAO-III, Muscat, Oman, Jan 2014, Springer.

Special thanks

- George Dantzig, born 101 years ago (8 Nov 1914)
- William Kahan, IEEE floating-point standard, including Quad, 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)
- Frank Curtis, Jose Luis Morales, Katya Scheinberg, Andreas Wächter

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 but Kahan is pessimistic