The DQQ procedure for multiscale optimization

Ding Ma and Michael Saunders
MS&E and ICME, Stanford University

The Fourth International Conference on Numerical Analysis and Optimization

Muscat, Sultanate of Oman, Jan 2–5, 2017

Abstract

Constrained optimization solvers typically scale the constraints, solve the scaled problem, then unscale. With multiscale problems and a conventional double-precision solver, the unscaled solution may not satisfy the constraints well. Our DQQ procedure continues with a quad-precision version of the solver to obtain accurate solutions efficiently.

A prime application is to metabolic networks in systems biology. Keywords are flux balance analysis (FBA) and genome-scale modeling of Metabolism and macromolecular Expression (ME models). DQQ typically achieves at least 20 digits of precision.

For smooth functions with few variables, quad-precision improves the performance of quasi-Newton optimization with finite-difference gradients. We illustrate with IMSPE-optimal design of computer experiments.

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

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

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

ICCOPT Tokyo, Aug 2016

ICCOPT Tokyo, Aug 2016

ICCOPT Tokyo, Aug 2016

Bus tour to Mt Fuji

Mt Fuji

Mt Fuji

Motivation

Multiscale LPs in systems biology

$$\max_{v} \; c^T v \quad \text{s.t.} \quad Sv = 0, \quad l \le v \le u$$

Normal for LP (simplex or interior)

- Scale to reduce large S_{ij}
- Solve with Feasibility/Optimality tols 1e−6
- Unscale

Multiscale LPs in systems biology

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

Normal for LP (simplex or interior)

- Scale to reduce large S_{ij}
- Solve with Feasibility/Optimality tols 1e−6
- Unscale

Difficulty

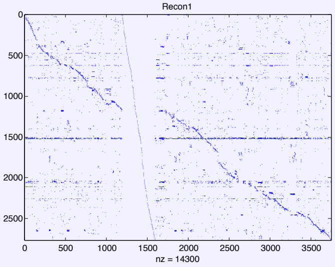
- Unscaling magnifies residuals
- Solution may be far from feasible or optimal

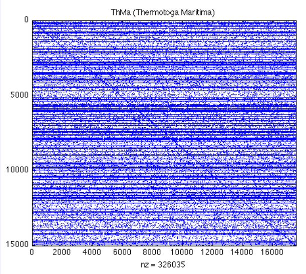
Stoichiometric matrices S

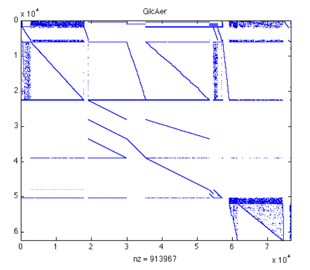
$$2H_2 + O_2 \rightarrow 2H_2O$$

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

chemicals $\times$ reactions







2800×3700 15000×18000 62000×77000

ME models (Metabolism and macromolecular Expression)

Coupling constraints

$$c_{\min} \le \frac{v_i}{v_j} \le c_{\max} \quad \equiv \quad c_{\min} v_j \le v_i, \quad v_i \le c_{\max} v_j$$

ME models (Metabolism and macromolecular Expression)

Coupling constraints

$$c_{\min} \le \frac{v_i}{v_j} \le c_{\max} \quad \equiv \quad c_{\min} v_j \le v_i, \quad v_i \le c_{\max} v_j$$

Lifting

$$v_1 \le 10000 v_2 \quad \equiv \quad v_1 \le 100 s_1, \quad s_1 \le 100 v_2$$

ME models (Metabolism and macromolecular Expression)

Coupling constraints

$$c_{\min} \le \frac{v_i}{v_j} \le c_{\max} \quad \equiv \quad c_{\min} v_j \le v_i, \quad v_i \le c_{\max} v_j$$

Lifting

$$v_1 \le 10000 v_2 \quad \equiv \quad v_1 \le 100 s_1, \qquad s_1 \le 100 v_2$$

Iterative refinement for LP problems

Ambros M. Gleixner (2015).
Exact and Fast Algorithms for Mixed-Integer Nonlinear Programming,
PhD thesis, ZIB, Technical University of Berlin.

Compute residuals in multiple precision, solve LP problem for correction, repeat.

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*

Double MINOS and Quad MINOS

`real(8)`	`real(16)`
`eps = 2.22e-16`	`eps = 1.93e-34`
Hardware	Software

We use this humane approach to Quad implementation

snopt9 = Double or Quad SQOPT, SNOPT

snPrecision.f90

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

snopt9 = Double or Quad SQOPT, SNOPT

snPrecision.f90

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

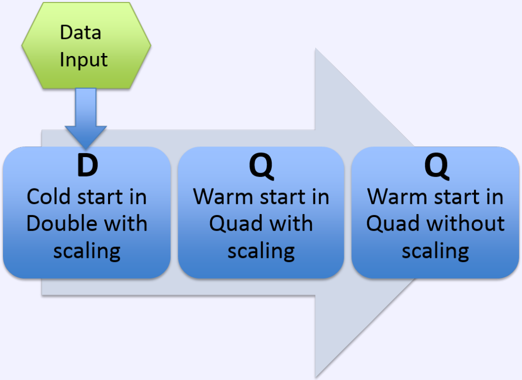
module sn50lp

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

DQQ procedure

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

DQQ procedure



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
Expand frequency	10000	100000	100000	100000
LU Factor tol	100.0	1.9	10.0	5.0
LU Update tol	10.0	1.9	10.0	5.0

pilot models from Netlib
Mészáros *problematic* LPs
ME biochemical network models

	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

Itns and runtimes in secs for DQQ Steps 1,2,3 (Double MINOS and Quad MINOS).
Pinf and Dinf = $\log_{10}$ max primal/dual infeasibilities.

	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**

Step 3 Pinf/ $\|x^*\|_\infty$ and Dinf/ $\|y^*\|_\infty$ are $\mathbf{O(10^{-30})}$

	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**

	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**

Step 3 Pinf/ $\|x^*\|_\infty$ and Dinf/ $\|y^*\|_\infty$ are $\mathbf{O(10^{-30})}$
except iprob is infeasible

DRR procedure

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

Plausible alternative to DQQ

DRR procedure

- We need Quad residuals for solving $Bx_B = b - Nx_n$ after LU and for solving $Bp = a$, $B^Ty = c_B$ each iteration

DRR procedure

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

DRR procedure

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

LPnetlib test problems

Compare DRR and DQQ

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

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

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

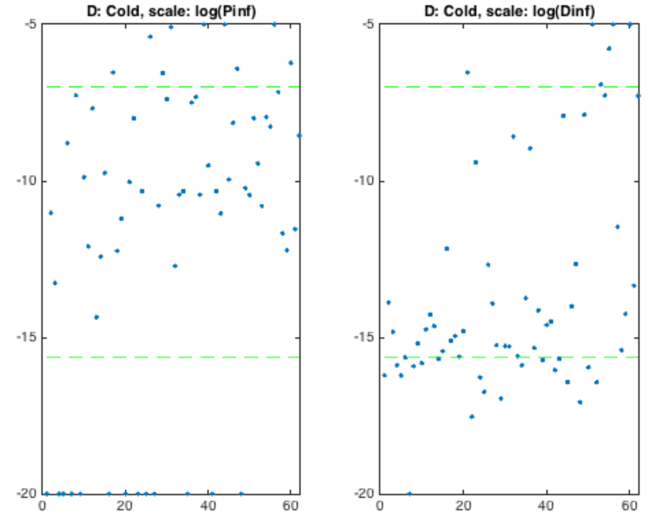
Default 1e−6
Default 1e−6

Plot $\log_{10}(\text{Pinf})$ and $\log_{10}(\text{Dinf})$ 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
```

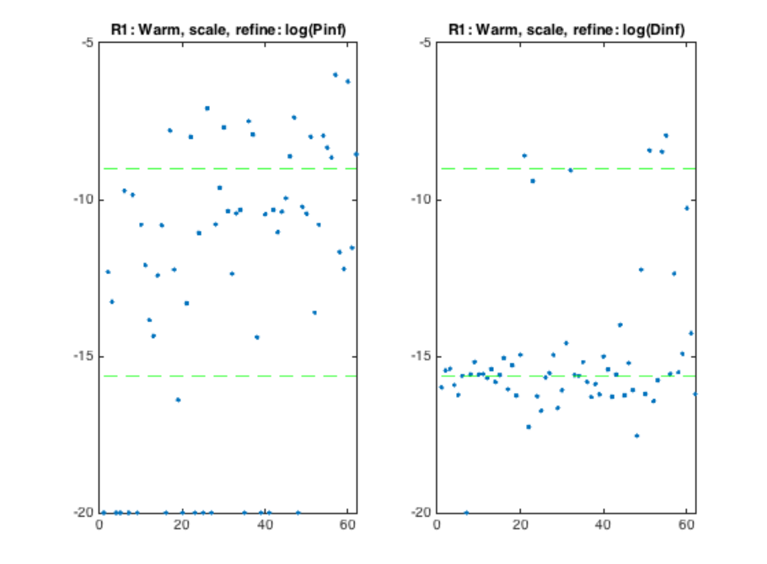
$\epsilon = 2.2\text{e-}16$



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
```

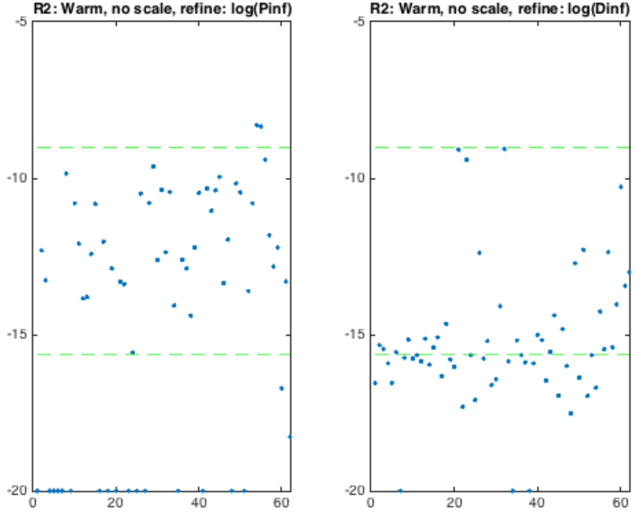
$\epsilon = 2.2\text{e-}16$



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

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

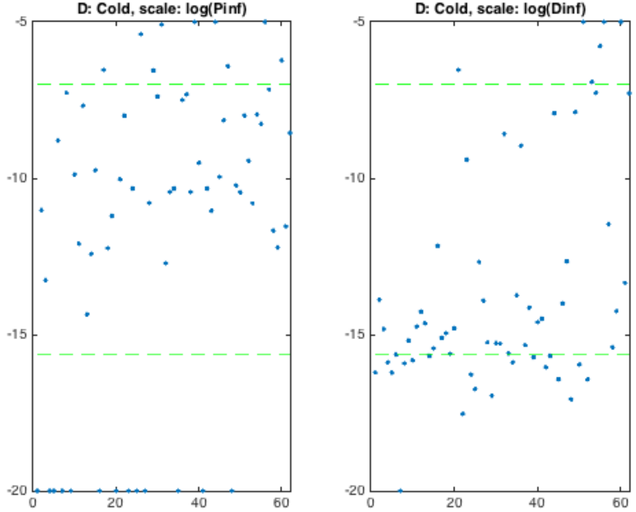
MINOS stops when $\begin{aligned} \text{Pinf} &\le \text{Feasibility tol} \\ \text{Dinf} &\le \text{Optimality tol} \end{aligned}$

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
```

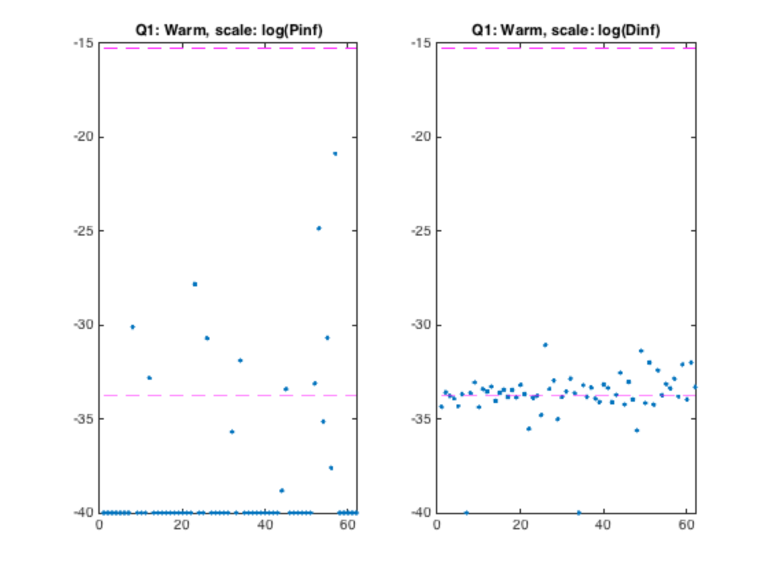
$\epsilon = 2.2\text{e-}16$



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
```

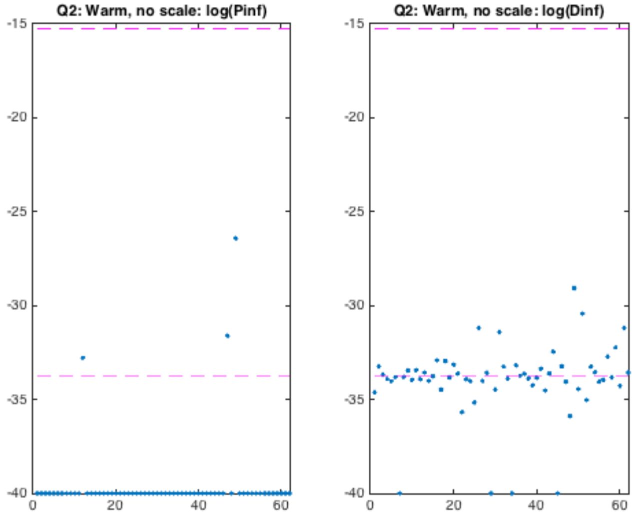
$\epsilon = 1.9\text{e-}35$



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
```

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



Soplex $\equiv$ SRR procedure

Step S: SoPlex 80-bit LP solver
Step R1: Refinement
Step R2: Refinement
⋮ ⋮

Refinement $\equiv$ Step S computing $(\Delta x, \Delta y)$

SoPlex = SRR procedure

- Ambros Gleixner PhD thesis (2015)

SoPlex = SRR procedure

- Ambros Gleixner PhD thesis (2015)
- Step S (SoPlex 80-bit LP solver) is slightly more accurate than Double Minos

SoPlex = SRR procedure

- Ambros Gleixner PhD thesis (2015)
- Step S (SoPlex 80-bit LP solver) is slightly more accurate than Double Minos
- Step R1, R2, ... (Refinement) need rational arithmetic to compute residuals $b - Ax$ and $c - A^T y$ for LP defining $(\Delta x, \Delta y)$.

SoPlex = SRR procedure

- Ambros Gleixner PhD thesis (2015)
- Step S (SoPlex 80-bit LP solver) is slightly more accurate than Double Minos
- Step R1, R2, . . . (Refinement) need rational arithmetic to compute residuals $b - Ax$ and $c - A^T y$ for LP defining $(\Delta x, \Delta y)$.
- SoPlex 80-bit solver computes $(\Delta x, \Delta y)$.

SoPlex = SRR procedure

- Ambros Gleixner PhD thesis (2015)
- Step S (SoPlex 80-bit LP solver) is slightly more accurate than Double Minos
- Step R1, R2, ... (Refinement) need rational arithmetic to compute residuals $b - Ax$ and $c - A^T y$ for LP defining $(\Delta x, \Delta y)$.
- SoPlex 80-bit solver computes $(\Delta x, \Delta y)$.
- SoPlex does very well on the systems biology ME models!

SoPlex = SRR procedure

- Ambros Gleixner PhD thesis (2015)
- Step S (SoPlex 80-bit LP solver) is slightly more accurate than Double Minos
- Step R1, R2, ... (Refinement) need rational arithmetic to compute residuals $b - Ax$ and $c - A^T y$ for LP defining $(\Delta x, \Delta y)$.
- SoPlex 80-bit solver computes $(\Delta x, \Delta y)$.
- SoPlex does very well on the systems biology ME models!
- Difficulty on some of the Mészáros models

Multiscale NLPs

Systems biology FBA problems with variable μ
Palsson Lab, UC San Diego, 2014–2016

ME models with nonlinear constraints

Coupling constraints can be functions of the organism's growth rate μ.
Lerman et al. (UCSD) consider growth-rate optimization nonlinearly with a single μ as the objective. Nonlinear constraints

$$\frac{v_i}{v_j} \leq \mu \qquad \equiv \qquad v_i \leq \mu v_j$$

are linear if μ is fixed. Can use a binary search to find the largest $\mu \in [\mu_{\mathrm{min}}, \mu_{\mathrm{max}}]$ that keeps the associated LP feasible.

Requires reliable solution of a sequence of related LPs.

tinyME (2500 × 2800 nonlinear ME model)

Nonlinear FBA formulation, Laurence Yang, UCSD, Dec 2014

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

- $\mu = x_1$ First columns of A, B are empty
- Binary search
  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                               13.5 secs
```

solveME (11000 × 19000 nonlinear ME model)

2600 nonlinear constraints, 16000 nonlinear variables
Laurence Yang, UCSD, Sep 2015

```
Itn        32 -- linear constraints satisfied.
Calling funcon.  mu =  0.832815729997476367249118875820191994

Major minor  step      objective  Feasible Optimal   nsb   ncon penalty BSwap
    1    32T 0.0E+00  8.32816E-01 4.3E-13 1.0E+03     0      4 1.0E+02     0

   19    40T 1.0E+00  8.32816E-01 2.5E-16 1.0E-03     0    743 1.0E+02     0
   20    40T 1.0E+00  8.32816E-01 1.0E-21 9.3E-04     0    784 1.0E+02     0

   23    40T 1.0E+00  8.55337E-01 3.4E-07 5.7E-05     0    907 1.0E+02     0
   24    40T 1.0E+00  8.55664E-01 2.1E-08 6.6E-07     0    948 1.0E+02     0
   25    11  1.0E+00  8.55664E-01 7.0E-17 1.8E-11     0    961 1.0E+02     0
   26     0  1.0E+00  8.55664E-01 9.3E-19 8.0E-29     0    962 1.0E+01     0

EXIT -- optimal solution found
```

ME 2.0 (large FBA and FVA problems)

$71,000 \times 80,000$ LPs
Laurence Yang, UCSD, Sep 2015

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
translation_b0025	RibF	0.212807	0.211712	0.210161
translation_b0071	LeuD	0.303304	0.765585	0.303634
translation_b0072	LeuC	0.303304	0.681146	0.303634

quadQP

Laurence Yang, UCSD, Jul 2016

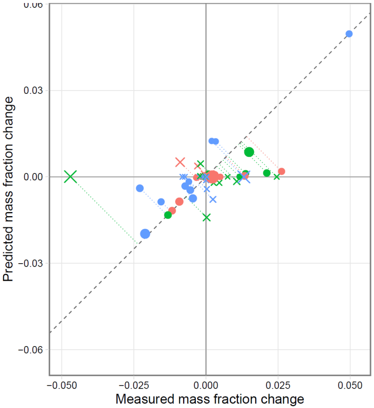
min $c^T x + \frac{1}{2} x^T H x$ st $Ax + s = b$ and bounds,

$$H \quad 4146 \times 4146$$
$$A \quad 12667 \times 21126$$

```
 Itn       rg      ninf   sinf        objective
   1 -1.1E+11     159 7.715E+00  0.00000000E+00
 500 -1.9E+10     159 7.715E+00  0.00000000E+00
 900 -1.3E+14     159 7.715E+00  0.00000000E+00

3000 -1.5E+05      11 9.784E-04  0.00000000E+00
3400 -5.1E-02       4 6.616E-05  0.00000000E+00
Itn     3554 -- feasible solution.  Objective =   7.634331271E-03

 Itn       rg      ninf   sinf        objective       nobj    nsb
4000 -4.8E-02       0 0.000E+00  3.08300094E-03
5000 -1.9E-04       0 0.000E+00  1.03117165E-03   1406      1
5200 -1.9E-05       0 0.000E+00  1.02662098E-03   1436      2
5300 -4.6E-09       0 0.000E+00  1.02660849E-03   1466      4
```

quadQP Laurence Yang, UCSD, Jul 2016 246 secs

Refining ME-computed proteomes



Proteome: the set of all proteins that is or can be expressed by an organism

Mass fraction: relative abundance of a given protein relative to all proteins (i.e., the proteome)

Includes solving quad-precision QP

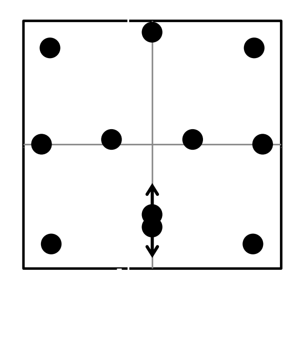
```
   Itn      rg     ninf   sinf       objective      nobj    nsb
  4700 -2.9E-03       0 0.000E+00  1.35572488E-03
  4800 -1.2E-03       0 0.000E+00  1.03964380E-03
  4900 -4.1E-04       0 0.000E+00  1.03667479E-03

   Itn      rg     ninf   sinf       objective      nobj    nsb
  5000 -1.9E-04       0 0.000E+00  1.03117165E-03   1406      1
  5100 -6.6E-05       0 0.000E+00  1.02674422E-03   1423      1
  5200 -1.9E-05       0 0.000E+00  1.02662098E-03   1436      2
  5300 -4.6E-09       0 0.000E+00  1.02660849E-03   1466      4
Itn     5318 --      25  nonbasics set on bound, basics recomputed
Itn     5318 -- feasible solution.  Objective =   1.026608483E-03

EXIT -- optimal solution found

No. of iterations                5318   Objective value      1.0266084830E-03
No. of major iterations             1   Linear objective     0.0000000000E+00
Penalty parameter          100.000000   Nonlinear objective  1.0266084830E-03
No. of calls to funobj           1489   No. of calls to funcon                0
No. of superbasics                  2   No. of basic nonlinears           12614
No. of degenerate steps          3686   Percentage                        69.31
Max x       (scaled)    12400 3.3E-01   Max pi      (scaled)        662 1.5E+04
Max x                   10269 1.9E+01   Max pi                     6083 3.4E+02
Max Prim inf(scaled)     7860 1.1E-25   Max Dual inf(scaled)       9618 1.5E-21
Max Primal infeas        8602 1.7E-22   Max Dual infeas            9618 4.8E-25
```

Quasi-Newton optimization with finite-difference gradients

Design of computer experiments

Selden Crary, indie-physicist, 2015

$n = 11$ points (x_i, y_i) on $[-1, 1]$ square (one twin-point)



[d,n,p,theta1,theta2]=[2,11,2,0.128,0.069]

Design of computer experiments

Selden Crary, indie-physicist, 2015

IMSPE-optimal designs (integrated mean-squared prediction error)

$$\min\ 1 - \text{trace}(L^{-1}R)$$

L and R: symmetric matrices of order $n+1$
with L increasingly ill-conditioned if points approach each other

Design of computer experiments

Selden Crary, indie-physicist, 2015

IMSPE-optimal designs (integrated mean-squared prediction error)

$$\min\ 1 - \mathrm{trace}(L^{-1}R)$$

L and R: symmetric matrices of order $n+1$
with L increasingly ill-conditioned if points approach each other

2D, Gaussian covariance parameters σ, θ_1, θ_2

$$L = \begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & V \end{bmatrix}, \qquad R \propto \int_{-1}^{1}\int_{-1}^{1} \begin{bmatrix} 1 \\ v \end{bmatrix} \begin{bmatrix} 1 & v^T \end{bmatrix} dx\, dy$$

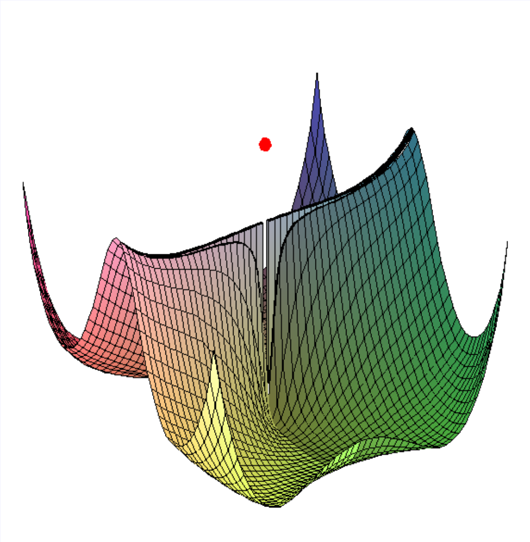
$$V_{ij} = \sigma^2 e^{-\theta_1(x_i - x_j)^2 - \theta_2(y_i - y_j)^2}, \qquad v_i \text{ functions of } \exp(\cdot) \text{ and } \mathrm{erf}(\cdot)$$

Pagoda plot Selden Crary, 2015

C^∞ a.e.

"Post", not pole

Need multistarts



With Maple, Selden has found twin-points, triple-points, . . . and a new class of rational functions (the Nu class)

Smooth sailing through the black-hole/white-hole event horizon

No string theory

No complex numbers in quantum mechanics

. . .

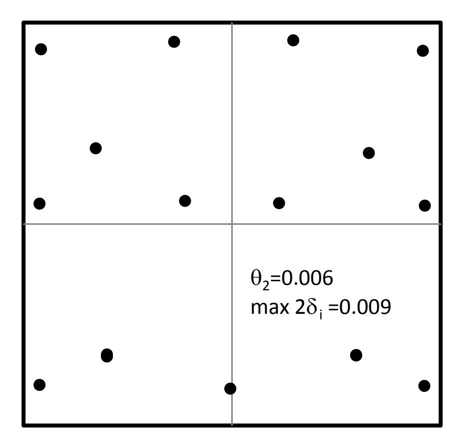
IMSPE, 2D, $n = 11$, $\theta_1 = 0.128$, $\theta_2 = 0.069$ Selden Crary

Quad MINOS unconstrained optimization $\in \mathbb{R}^{22}$ without gradients 6 secs

```
 Itn ph pp      rg      step      objective      nobj  nsb  cond(H)
   1  4  0  3.9E-05  1.0E+03  2.47305090E-05    57   22  4.5E+01
  50  4  1  6.4E-07  6.2E+00  6.01181966E-06  1384   22  2.1E+04
 100  3  1 -7.6E-08  1.1E+00  5.65611811E-06  2726   22  1.4E+05
 150  4  1  8.5E-07  6.0E+00  5.11053080E-06  4102   22  8.2E+03
 200  4  1  2.6E-08  1.1E-01  5.02762155E-06  5464   22  1.0E+07
 239  4  1  1.1E-07  1.5E-06  5.02762154E-06  7478   22  1.0E+10
alfmax = 3.1E+04   pnorm = 1.5E-04   gnorm = 1.1E-07   g'p = -2.6E-15   numf = 15
Search exit  7 -- too many functions.

EXIT -- optimal solution found

No. of iterations                  239   Objective value      5.0276215358E-06
No. of calls to funobj            7538   Calls with mode=2 (f, known g)    244
Calls for forward differencing    4466   Calls for central differencing   1716
Max Primal infeas            0 0.0E+00   Max Dual infeas                2 1.1E-07
                                                             (usually 1.0E-15)
```

17 points on $[-1, 1]^2$
(two twin-points)

Quad MINOS design
refined by Selden
via MAPLE



After further downhill searching, both sets of twins are now close, confirming the conjecture of a true double–twin-point design. 20161106

IMSPE linear algebra question

L, R real, symmetric, indefinite, ill-conditioned $\quad$ $\min\ 1 - \mathrm{trace}(L^{-1}R)$

cf. GEV problem $Rv = \lambda Lv$ $\quad$ $\mathrm{trace}(L^{-1}R) = \sum \lambda_i$

IMSPE linear algebra question

L, R real, symmetric, indefinite, ill-conditioned $\qquad$ $\min\ 1 - \text{trace}(L^{-1}R)$

cf. GEV problem $Rv = \lambda Lv$ $\qquad$ $\text{trace}(L^{-1}R) = \sum \lambda_i$

- QZ algorithm ignores symmetry but avoids ill-conditioned L^{-1}
- Will QZ compute real λ_i?

IMSPE linear algebra question

L, R real, symmetric, indefinite, ill-conditioned $\qquad$ $\min \ 1 - \mathrm{trace}(L^{-1}R)$

cf. GEV problem $Rv = \lambda Lv$ $\qquad$ $\mathrm{trace}(L^{-1}R) = \sum \lambda_i$

- QZ algorithm ignores symmetry but avoids ill-conditioned L^{-1}
- Will QZ compute real λ_i?

Yuji Nakatsukasa was in Tokyo; developing `qdwhgep.m` for $Ax = \lambda Bx$ (real, symmetric)

- Congruence transformations are real
- Eigenvalues can be complex conjugate pairs
- $\mathrm{trace}(B^{-1}A) = \sum \lambda_i$ will be real

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.

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.

- Exact simplex with rational arithmetic (e.g. Qsopt_ex) can be warm-started, but is still too expensive on large problems.

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.

- Exact simplex with rational arithmetic (e.g. Qsopt_ex) can be warm-started, but is still too expensive on large problems.
- Iterative refinement of $Bp = a$, $B^T y = c$ is a sparing use of Quad, but doesn't help $B = LU$ (with basis repair). Quad everywhere is the humane approach!

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.

- Exact simplex with rational arithmetic (e.g. Qsopt_ex) can be warm-started, but is still too expensive on large problems.
- Iterative refinement of $Bp = a$, $B^T y = c$ is a sparing use of Quad, but doesn't help $B = LU$ (with basis repair).
  Quad everywhere is the humane approach!
- Same difficulty with SoPlex
  Quad SoPlex is in the making!
  C++ allows switch from Double to Quad at runtime.

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
- Ambros M. Gleixner (2015). Exact and Fast Algorithms for Mixed-Integer Nonlinear Programming, PhD thesis, ZIB, Technical University of Berlin
- Selden Crary and Jan Stormann (2016). Four-point, 2D, free-ranging, IMSPE-optimal, twin-point designs, `arxiv.org/abs/1510.01685`
- Laurence Yang, Ding Ma, Ali Ebrahim, Colton Loyd, and Michael Saunders (2016). solveME: fast and reliable solution of nonlinear ME models, *BMC Bioinformatics*, 17(1):391
- Ding Ma, Laurence Yang, Ronan Fleming, Ines Thiele, Bernhard Palsson, and Michael Saunders (2017). Reliable and efficient solution of genome-scale models of Metabolism and macromolecular Expression, *Scientific Reports* 7, 40863; `doi:10.1038/srep40863`

Special thanks

- George Dantzig, born 102 years ago (8 Nov 1914)
- William Kahan (Berkeley), IEEE standard (including Quad), Boulder.pdf (2011)
- GCC, gfortran (GNU)
- Ronan Fleming, Ines Thiele (Luxembourg)
- Bernhard Palsson, Josh Lerman, Teddy O'Brien, Laurence Yang (UCSD)
- Philip Gill, Elizabeth Wong (UCSD)
- Ed Klotz (IBM CPLEX), Yuekai Sun, Jon Dattorro (Stanford)
- Selden Crary (Palo Alto), Richard Martinez (Stanford)
- Ambros Gleixner (ZIB), Yuji Nakatsukasa (Oxford)
- Mehi and colleagues (Oman)
- Sultan Qaboos bin Said Al Said

Imperial Palace, Tokyo

Tokyo subway

Welcome home

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? Yes for LPs
- How about AMPL? No, but should be feasible
- Is there a quadSNOPT? Yes, in f90 SNOPT9 we 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 wish, but Kahan is pessimistic