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
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
1 Motivation
2 System and Methods
3 DQQ procedure
4 DRR procedure
5 62 LPnetlib test problems
6 Multiscale NLPs
7 Philosophy
8 Conclusions
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
- Unsacle

**Difficulty**
- Unscaling magnifies residuals
- Solution may be far from feasible or optimal
Stoichiometric matrices $S$

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

<table>
<thead>
<tr>
<th></th>
<th>$H_2$</th>
<th>$O_2$</th>
<th>$H_2O$</th>
<th>$W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_2$</td>
<td></td>
<td></td>
<td></td>
<td>$-2$</td>
</tr>
<tr>
<td>$O_2$</td>
<td></td>
<td></td>
<td></td>
<td>$-1$</td>
</tr>
<tr>
<td>$H_2O$</td>
<td></td>
<td></td>
<td></td>
<td>$2$</td>
</tr>
</tbody>
</table>

chemicals $\times$ reactions

$2800 \times 3700$

$15000 \times 18000$

$62000 \times 77000$
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:

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

where $x(t) \in \mathbb{R}^m$ is a vector of time-dependent concentrations and $v(t) \in \mathbb{R}^n$ is a vector of reaction fluxes. The objective of maximizing growth rate at steady state leads to an LP:

$$\max_v c^T v \quad (2a)$$

s.t. $Sv = 0, \quad (2b)$

$$l \leq v \leq u, \quad (2c)$$

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

\[ c_{\text{min}} \leq \frac{v_i}{v_j} \leq c_{\text{max}} \]  

become linear constraints

\[ c_{\text{min}} v_j \leq v_i, \quad v_i \leq c_{\text{max}} v_j \]  

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

\[ 0.0001 \leq \frac{v_1}{v_2} \leq 10000. \]  \hspace{1cm} (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

\[ v_1 \leq 100s_1, \quad s_1 \leq 100v_2. \]  \hspace{1cm} (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

\[ v_2 \leq 100s_2, \quad s_2 \leq 100v_1. \]
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 \( q = \text{qdotdd}(v,w) \) needs several double functions

\text{twosum, split, twoproduct, sum2, dot2}

to compute \( \text{double } x, y \)

and hence quad result \( q = \text{quad}(x) + \text{quad}(y) \)

Double-double datatype (\( \approx 32 \) digits)

\text{QD: http://crd-legacy.lbl.gov/~dhbailey/mpdist/}

\text{C++ with interfaces to C++ and F90}

\text{DDFUN90: entirely F90}

\text{Minor changes to source code}

Quad datatype (\( \approx 34 \) digits)

Some \text{f90} compilers such as \text{gfortran}

Again minor changes to source code

We use this \text{humane approach to Quad implementation}
System and Methods
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 \texttt{real(8)} changed to \texttt{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:

```fortran
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:

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

Later:

```fortran
module sn50lp
    use snModulePrecision, only : ip, rp
    subroutine s5solveLP ( x, y )
        real(rp), intent(inout) :: x(nb), y(nb)
    end subroutine s5solveLP
end module sn50lp
```
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 \text{ with } \begin{align*}
\text{real}(8) & \rightarrow \text{real}(16) \\
\text{eps} = 2.22e-16 & \rightarrow \text{eps} = 1.93e-34
\end{align*}
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

<table>
<thead>
<tr>
<th></th>
<th>Default</th>
<th>Step D</th>
<th>Step Q1</th>
<th>Step Q2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale option</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Feasibility tol</td>
<td>1e-6</td>
<td>1e-7</td>
<td>1e-15</td>
<td>1e-15</td>
</tr>
<tr>
<td>Optimality tol</td>
<td>1e-6</td>
<td>1e-7</td>
<td>1e-15</td>
<td>1e-15</td>
</tr>
<tr>
<td>LU Factor tol</td>
<td>100.0</td>
<td>10.0</td>
<td>10.0</td>
<td>5.0</td>
</tr>
<tr>
<td>LU Update tol</td>
<td>10.0</td>
<td>10.0</td>
<td>10.0</td>
<td>5.0</td>
</tr>
</tbody>
</table>
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 $|A_{ij}|$ | $\|x^*\|_\infty$ | $\|y^*\|_\infty$ |
|----------|------|------|----------|-------------|----------------|----------------|
| 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. Pinf/ $\|x^*\|_{\infty}$ and 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.

<table>
<thead>
<tr>
<th>model</th>
<th>Itns</th>
<th>Times</th>
<th>Final objective</th>
<th>Pinf</th>
<th>Dinf</th>
</tr>
</thead>
<tbody>
<tr>
<td>pilot4</td>
<td>1571</td>
<td>0.1</td>
<td>$-2.5811392602e+03$</td>
<td>-05</td>
<td>-13</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>0.0</td>
<td>$-2.5811392589e+03$</td>
<td>-39</td>
<td>-31</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.0</td>
<td>$-2.5811392589e+03$</td>
<td>-</td>
<td>-30</td>
</tr>
<tr>
<td>pilot</td>
<td>16060</td>
<td>5.7</td>
<td>$-5.5739887685e+02$</td>
<td>-06</td>
<td>-03</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>0.7</td>
<td>$-5.5748972928e+02$</td>
<td>-</td>
<td>-27</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.2</td>
<td>$-5.5748972928e+02$</td>
<td>-</td>
<td>-32</td>
</tr>
<tr>
<td>pilot87</td>
<td>19340</td>
<td>15.1</td>
<td>$3.017103489e+02$</td>
<td>-09</td>
<td>-06</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>2.2</td>
<td>$3.0171034733e+02$</td>
<td>-</td>
<td>-33</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1.2</td>
<td>$3.0171034733e+02$</td>
<td>-</td>
<td>-33</td>
</tr>
<tr>
<td>model</td>
<td>Itns</td>
<td>Times</td>
<td>Final objective</td>
<td>Pinf</td>
<td>Dinf</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
<td>-------</td>
<td>-----------------------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>de063155</td>
<td>921</td>
<td>0.0</td>
<td>1.8968704286e+10</td>
<td>−13</td>
<td>+03</td>
</tr>
<tr>
<td></td>
<td>78</td>
<td>0.1</td>
<td>9.8830944565e+09</td>
<td>−</td>
<td>−17</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.0</td>
<td>9.8830944565e+09</td>
<td>−</td>
<td>−24</td>
</tr>
<tr>
<td>de063157</td>
<td>488</td>
<td>0.0</td>
<td>1.4561118445e+11</td>
<td>+20</td>
<td>+18</td>
</tr>
<tr>
<td></td>
<td>476</td>
<td>0.5</td>
<td>2.1528501109e+07</td>
<td>−27</td>
<td>−12</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.0</td>
<td>2.1528501109e+07</td>
<td>−</td>
<td>−12</td>
</tr>
<tr>
<td>de080285</td>
<td>418</td>
<td>0.0</td>
<td>1.4495817688e+01</td>
<td>−09</td>
<td>−02</td>
</tr>
<tr>
<td></td>
<td>132</td>
<td>0.1</td>
<td>1.3924732864e+01</td>
<td>−35</td>
<td>−32</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.0</td>
<td>1.3924732864e+01</td>
<td>−</td>
<td>−32</td>
</tr>
<tr>
<td>gen1</td>
<td>369502</td>
<td>205.3</td>
<td>−1.6903658594e−08</td>
<td>−06</td>
<td>−12</td>
</tr>
<tr>
<td></td>
<td>246428</td>
<td>9331.3</td>
<td>1.2935699163e−06</td>
<td>−12</td>
<td>−31</td>
</tr>
<tr>
<td></td>
<td>2394</td>
<td>81.6</td>
<td>1.2953925804e−06</td>
<td>−45</td>
<td>−30</td>
</tr>
<tr>
<td>gen2</td>
<td>44073</td>
<td>60.0</td>
<td>3.2927907828e+00</td>
<td>−04</td>
<td>−11</td>
</tr>
<tr>
<td></td>
<td>1599</td>
<td>359.9</td>
<td>3.2927907840e+00</td>
<td>−</td>
<td>−29</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>10.4</td>
<td>3.2927907840e+00</td>
<td>−</td>
<td>−32</td>
</tr>
<tr>
<td>gen4</td>
<td>45369</td>
<td>212.4</td>
<td>1.5793970394e−07</td>
<td>−06</td>
<td>−10</td>
</tr>
<tr>
<td></td>
<td>53849</td>
<td>14812.5</td>
<td>2.8932268196e−06</td>
<td>−12</td>
<td>−30</td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>10.4</td>
<td>2.8933064888e−06</td>
<td>−54</td>
<td>−30</td>
</tr>
<tr>
<td>model</td>
<td>Itns</td>
<td>Times</td>
<td>Final objective</td>
<td>Pinf</td>
<td>Dinf</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>--------------------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>l30</td>
<td>1229326</td>
<td>876.7</td>
<td>9.5266141574e-01</td>
<td>-10</td>
<td>-9</td>
</tr>
<tr>
<td></td>
<td>275287</td>
<td>7507.1</td>
<td>-7.5190273434e-26</td>
<td>-25</td>
<td>-32</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.2</td>
<td>-4.2586876849e-24</td>
<td>-24</td>
<td>-33</td>
</tr>
<tr>
<td>iprob</td>
<td>1087</td>
<td>0.2</td>
<td>2.6891551285e+03</td>
<td>+02</td>
<td>-11</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.0</td>
<td>2.6891551285e+03</td>
<td>+02</td>
<td>-31</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0.0</td>
<td>2.6891551285e+03</td>
<td>+02</td>
<td>-28</td>
</tr>
<tr>
<td>TMA_ME</td>
<td>12225</td>
<td>37.1</td>
<td>8.0051076669e-07</td>
<td>-06</td>
<td>-05</td>
</tr>
<tr>
<td></td>
<td>685</td>
<td>61.5</td>
<td>8.7036315385e-07</td>
<td>-24</td>
<td>-30</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>6.7</td>
<td>8.7036315385e-07</td>
<td>-1</td>
<td>-31</td>
</tr>
<tr>
<td>GlcAerWT</td>
<td>62856</td>
<td>9707.3</td>
<td>-2.4489880182e+04</td>
<td>+04</td>
<td>-05</td>
</tr>
<tr>
<td></td>
<td>5580</td>
<td>3995.6</td>
<td>-7.0382449681e+05</td>
<td>-07</td>
<td>-26</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>60.1</td>
<td>-7.0382449681e+05</td>
<td>-19</td>
<td>-21</td>
</tr>
<tr>
<td>GlcAlift</td>
<td>134693</td>
<td>14552.8</td>
<td>-5.1613878666e+05</td>
<td>-03</td>
<td>-01</td>
</tr>
<tr>
<td></td>
<td>3258</td>
<td>1067.1</td>
<td>-7.0434008750e+05</td>
<td>-09</td>
<td>-26</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>48.1</td>
<td>-7.0434008750e+05</td>
<td>-20</td>
<td>-22</td>
</tr>
</tbody>
</table>
**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 $r ← r - y_k B_k$ (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 $By = a$, $B^Ty = c_B$ each iteration

- Quad $r = a - By$ needs $r \leftarrow r - y_kB_k$ (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
DQR 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 $By = a$, $B^Ty = c_B$ each iteration
- Quad $r = a - By$ needs $r \leftarrow r - y_kB_k$ (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
- James Ho (1975) SRR procedure?
LPnetlib test problems

Unexpectedly high accuracy in Quad
### 62 classic LP problems (ordered by file size)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Problem</th>
<th>Problem</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>afiro</td>
<td>scfxml</td>
<td>ship04s</td>
<td>pilotja</td>
</tr>
<tr>
<td>stocfor1</td>
<td>bandm</td>
<td>seba</td>
<td>ship08l</td>
</tr>
<tr>
<td>adlittle</td>
<td>e226</td>
<td>grow15</td>
<td>nesm</td>
</tr>
<tr>
<td>scagr7</td>
<td>grow7</td>
<td>fffff800</td>
<td>ship12l</td>
</tr>
<tr>
<td>sc205</td>
<td>etamacro</td>
<td>scfxml3</td>
<td>cycle</td>
</tr>
<tr>
<td>share2b</td>
<td>agg</td>
<td>ship04l</td>
<td>greenbea</td>
</tr>
<tr>
<td>recipe</td>
<td>scsd1</td>
<td>ganges</td>
<td>greenbeb</td>
</tr>
<tr>
<td>vtpbase</td>
<td>standata</td>
<td>sctap2</td>
<td>80bau3b</td>
</tr>
<tr>
<td>share1b</td>
<td>beaconfd</td>
<td>grow22</td>
<td>d2q06c</td>
</tr>
<tr>
<td>bore3d</td>
<td>gfrdpnc</td>
<td>ship08s</td>
<td>woodw</td>
</tr>
<tr>
<td>scorpion</td>
<td>stair</td>
<td>stocfor2</td>
<td>d6cube</td>
</tr>
<tr>
<td>capri</td>
<td>scrs8</td>
<td>pilotwe</td>
<td>pilot</td>
</tr>
<tr>
<td>brandy</td>
<td>shell</td>
<td>ship12s</td>
<td>wood1p</td>
</tr>
<tr>
<td>scagr25</td>
<td>scfxml2</td>
<td>25fv47</td>
<td>pilot87</td>
</tr>
<tr>
<td>sctap1</td>
<td>pilot4</td>
<td>sierra</td>
<td></td>
</tr>
<tr>
<td>israel</td>
<td>scsd6</td>
<td>czprob</td>
<td></td>
</tr>
</tbody>
</table>
DRR procedure on LPnetlib problems

\[ P_{\text{inf}} = \max \text{ Primal infeasibility} \]
\[ D_{\text{inf}} = \max \text{ Dual infeasibility}/(1 + \| y^* \|_{\infty}) \]

MINOS stops when

\[ P_{\text{inf}} \leq \text{ Feasibility tol} \quad \text{Default } 1e^{-6} \]
\[ D_{\text{inf}} \leq \text{ Optimality tol} \quad \text{Default } 1e^{-6} \]

Plot $\log_{10}(P_{\text{inf}})$ and $\log_{10}(D_{\text{inf}})$ 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

$\epsilon = 2.2e^{-16}$
Primal/dual infeasibilities:

Step R1: Double MINOS, warm start, scale, refine

Scale option 2
Feasibility tol $1\times10^{-9}$
Optimality tol $1\times10^{-9}$

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

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

\[ \epsilon = 2.2e-16 \]
DQQ procedure on LPnetlib problems

\[ \text{Pinf} = \max \text{ Primal infeasibility} \]
\[ \text{Dinf} = \max \frac{\text{Dual infeasibility}}{1 + \|y^*\|_{\infty}} \]

MINOS stops when
\[ \text{Pinf} \leq \text{Feasibility tol} \]
\[ \text{Dinf} \leq \text{Optimality tol} \]

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

\[ \epsilon = 2.2e^{-16} \]
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

$\epsilon = 1.9e-35$
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

\( \epsilon = 1.9e-35 \)
Multiscale NLPs

Systems biology FBA problems with variable $\mu$
(Palsson Lab, UC San Diego, 2014)
ME models with nonlinear constraints

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

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

represented as

$$v_i \leq \mu v_j$$

are added to (2b), where $v_i, v_j, \mu$ are all variables. Constraints (8) are linear if $\mu$ is fixed at a specific value $\mu_k$. Lerman et al. employ a binary search to find the largest $\mu_k \in [\mu_{\text{min}}, \mu_{\text{max}}]$ 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

\[
\begin{align*}
\text{max} & \quad \mu \\
\text{st} & \quad \mu Ax + Bx = 0 \quad \text{界限} \\
& \quad Sx = b \\
\text{bounds on } x
\end{align*}
\]

\[
\begin{align*}
\text{max} & \quad \mu \\
\text{st} & \quad \mu Ax + w = 0 \\
& \quad Bx - w = 0 \\
& \quad Sx = b \\
\text{bounds on } x, \text{ no bounds on } w
\end{align*}
\]

- 小例子: $\approx 2500 \times 3000$
- $\mu = x_1$ 且 $A$, $B$ 的第一列是空的
- 约束是线性的如果 $\mu$ 是固定的，建议二分搜索在一系列的 LP 上
  - 25 个 LP 子问题将给出 8 位精度（实际上需要 quad Simplex）
- 而且，应用 quad MINOS LCL 方法 = 线性约束拉格朗日
  - 6 个 NLP 子问题（线性化约束）给出 20 位精度
Quadratic convergence of major iterations (Robinson 1972)

quadMINOS 5.6 (Nov 2014)

Begin tinyME-NLP cold start NLP with mu = mu0
Ittn 304 -- linear constraints satisfied.
Calling funcon. mu = 0.800000000000000000000000000000004
nnCon, nnJac, neJac 1073 1755 2681
funcon sets 2681 out of 2681 constraint gradients.
funobj sets 1 out of 1 objective gradients.

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

EXIT -- optimal solution found
EXIT -- optimal solution found

<table>
<thead>
<tr>
<th>Problem name</th>
<th>tinyME</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of iterations</td>
<td>912</td>
</tr>
<tr>
<td>No. of major iterations</td>
<td>6</td>
</tr>
<tr>
<td>Penalty parameter</td>
<td>1.000000</td>
</tr>
<tr>
<td>No. of calls to funobj</td>
<td>98</td>
</tr>
<tr>
<td>No. of superbasics</td>
<td>0</td>
</tr>
<tr>
<td>No. of degenerate steps</td>
<td>0</td>
</tr>
<tr>
<td>Max x (scaled)</td>
<td>12 5.6E-01</td>
</tr>
<tr>
<td>Max x</td>
<td>1020 6.1E+01</td>
</tr>
<tr>
<td>Max Prim inf(scaled)</td>
<td>0 0.0E+00</td>
</tr>
<tr>
<td>Max Primal infeas</td>
<td>0 0.0E+00</td>
</tr>
<tr>
<td>Nonlinear constraint violn</td>
<td>1.9E-20</td>
</tr>
<tr>
<td>Objective value</td>
<td>8.4694810579E-01</td>
</tr>
<tr>
<td>Linear objective</td>
<td>0.0000000000E+00</td>
</tr>
<tr>
<td>Nonlinear objective</td>
<td>8.4694810579E-01</td>
</tr>
<tr>
<td>No. of basic nonlinears</td>
<td>786</td>
</tr>
<tr>
<td>No. of calls to funcon</td>
<td>98</td>
</tr>
<tr>
<td>Percentage</td>
<td>0.00</td>
</tr>
<tr>
<td>Max x</td>
<td>1020 6.1E+01</td>
</tr>
<tr>
<td>Max pi (scaled)</td>
<td>103 8.3E+05</td>
</tr>
<tr>
<td>Max pi</td>
<td>103 9.7E+03</td>
</tr>
<tr>
<td>Max Dual inf(scaled)</td>
<td>9 2.9E-14</td>
</tr>
<tr>
<td>Max Dual infeas</td>
<td>9 1.3E-18</td>
</tr>
<tr>
<td>Funcon called with nstate =</td>
<td>2</td>
</tr>
<tr>
<td>Final value of mu</td>
<td>0.84694810578563166175146802332321527</td>
</tr>
</tbody>
</table>

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 \(v_j\)

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Protein</th>
<th>Double CPLEX</th>
<th>Quad MINOS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(v_{\min})</td>
<td>(v_{\max})</td>
</tr>
<tr>
<td>translation_b0169</td>
<td>RpsB</td>
<td>30.715011</td>
<td>30.712581</td>
</tr>
<tr>
<td>translation_b0025</td>
<td>RibF</td>
<td>0.212807</td>
<td>0.211712</td>
</tr>
<tr>
<td>translation_b0071</td>
<td>LeuD</td>
<td>0.303304</td>
<td>0.765585</td>
</tr>
<tr>
<td>translation_b0072</td>
<td>LeuC</td>
<td>0.303304</td>
<td>0.681146</td>
</tr>
</tbody>
</table>
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 (?)

We will cover some variations which may be useful.

- which [X]
- , which [✓]
- that [✓]

We will cover some variations which 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.

- Can humour (not satire) be the antidote to extremism?
  It would be great to think so.

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

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?

– Dalai Lama  

– 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

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


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