Numerics in LP and MIP solvers

Ambros Gleixner[1,2]

[1]Zuse Institute Berlin and [2]HTW Berlin

gleixner@htw-berlin.de

CO@Work 2024 · September 19, 2024

Hochschule für Technik und Wirtschaft Berlin

University of Applied Sciences

ZUSE INSTITUTE BERLIN

MODAL FORSCHUNGSCAMPUS

SPONSORED BY THE

Federal Ministry of Education and Research

Numerical trouble and inaccuracies

```
SCIP> display solution

objective value:                     147726.165057267
x2826                                               1   (obj:13.8132)
x2840                                               1   (obj:17.3592)
x2842                                 1.000000000318   (obj:17.2398)
x2845                                               1   (obj:12)
x2850                                               1   (obj:12.1173)
x2851                                 0.999999999301   (obj:14.4105)
x2852                                 0.999999999987   (obj:12)
x2854                                               1   (obj:14.9862)
x2857                                               1   (obj:18.7107)
x2861                                               1   (obj:15.7479)
```

Numerical trouble and inaccuracies

```
  9.7s|     1 |     0 |  3507 |     - |   154M |   0 |2712 |  13k|  13k|  68 |  8 |  18 |   0 | 9.625279e+04 | 3.723994e+05 | 286.90%| unknown
 10.5s|     1 |     0 |  3645 |     - |   157M |   0 |2712 |  13k|  13k|  93 |  9 |  18 |   0 | 9.625416e+04 | 3.723994e+05 | 286.89%| unknown
 11.0s|     1 |     0 |  3718 |     - |   158M |   0 |2712 |  13k|  13k| 110 | 10 |  18 |   0 | 9.625435e+04 | 3.723994e+05 | 286.89%| unknown
 11.1s|     1 |     0 |  3782 |     - |   160M |   0 |2712 |  13k|  13k| 124 | 11 |  18 |   0 | 9.625474e+04 | 3.723994e+05 | 286.89%| unknown
 11.3s|     1 |     0 |  3893 |     - |   161M |   0 |2712 |  13k|  13k| 144 | 12 |  18 |   0 | 9.625534e+04 | 3.723994e+05 | 286.89%| unknown
 time | node  | left  |LP iter|LP it/n|mem/heur|mdpt |vars |cons |rows |cuts |sepa|confs|strbr|  dualbound   | primalbound  |  gap   | compl.
 24.3s|     1 |     2 | 35722 |     - |   163M |   0 |2712 |  13k|  13k| 144 | 12 |  44 |  19 | 9.626681e+04 | 3.723994e+05 | 286.84%| unknown
(node 5) numerical troubles in LP 104 -- unresolved
L 129s|    76 |    73 |127950 |1663.7 |    alns|  20 |2712 |  13k|  13k| 264 |  2 |  75 | 519 | 9.628491e+04 | 3.530913e+05 | 266.72%| unknown
  144s|   100 |    97 |148598 |1469.0 |   187M |  23 |2712 |  13k|  13k| 285 |  1 |  94 | 553 | 9.628491e+04 | 3.530913e+05 | 266.72%| unknown
```

Numerical trouble and inaccuracies

```
2800s| 93200 |   348 |  4749k|  49.7 |   253M |  55 | 289 |1925 |1764 | 207k|  0 |5691 |8539 | 3.027253e+03 | 3.336240e+03 |  10.21%|  98.40%

SCIP Status        : problem is solved [optimal solution found]
Solving Time (sec) : 2802.85
Solving Nodes      : 93263 (total of 95347 nodes in 2 runs)
Primal Bound       : +3.33623984845755e+03 (43 solutions)
Dual Bound         : +3.33623984845755e+03
Gap                : 0.00 %
[linear] <c13452>:  -0.002186551<x12>[C] (+38233.728) +<x47>[C] (+85.2158153) +1600000<x477>[B] (+0) +1600000<x622>[I] (-1.15885912e-11)
+1600000<x682>[I] (+0) >= 1.615819082;
                                         ;
violation: left hand side is violated by 1.85415070872441e-05
best solution is not feasible in original problem
```

1. Introduction

2. Floating-point arithmetic and MIP tolerances

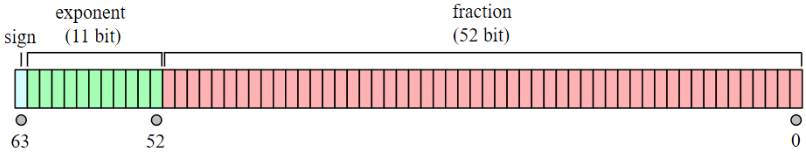
3. Some guidelines and tools

4. Iterative refinement for LP

5. Solving MIPs exactly

Floating-point arithmetic

Virtually all MIP solvers are built on double-precision floating-point arithmetic (IEEE754):



- Real numbers stored as $(-1)^{\text{sign}} \cdot 1.\text{fraction} \cdot 2^{\text{exponent}-1023}$
- enough to represent about 15 digits $\rightsquigarrow$ round-off errors afterwards, e.g.

$$\frac{1}{3} =_{fp} 0.333333333333333314829 6\ldots$$

$$3,000,000 \cdot \frac{1}{3} - 1,000,000 =_{fp} 0.00000000148296\ldots$$

$$3,000,000,000,000 \cdot \frac{1}{3} - 1,000,000,000,000 =_{fp} 0.00148296\ldots$$

Feasibility and optimality in floating-point solvers

MIP solvers use numerical tolerances, typically in the range 10^{-6} to 10^{-9}:

- Integrality tolerance ϵ_{int} : $\quad \alpha \in \mathbb{Z} \Leftrightarrow_{\mathit{tol}} \alpha \in \mathbb{Z} + [-\epsilon_{\mathit{int}}, \epsilon_{\mathit{int}}]$, e.g., $0.9999999 =_{\mathit{tol}} 1$.

Feasibility and optimality in floating-point solvers

MIP solvers use numerical tolerances, typically in the range 10^{-6} to 10^{-9}:

- Integrality tolerance ϵ_{int} : $\alpha \in \mathbb{Z} \Leftrightarrow_{tol} \alpha \in \mathbb{Z} + [-\epsilon_{int}, \epsilon_{int}]$, e.g., $0.9999999 =_{tol} 1$.
- Feasibility tolerance ϵ_{feas} : $a^T x \leq b \Leftrightarrow_{tol} \ldots$

Absolute: $a^T x - b \leq \epsilon_{feas}$

Feasibility and optimality in floating-point solvers

MIP solvers use numerical tolerances, typically in the range 10^{-6} to 10^{-9}:

- Integrality tolerance ϵ_{int} : $\alpha \in \mathbb{Z} \Leftrightarrow_{tol} \alpha \in \mathbb{Z} + [-\epsilon_{int}, \epsilon_{int}]$, e.g., $0.9999999 =_{tol} 1$.
- Feasibility tolerance ϵ_{feas} : $a^T x \leq b \Leftrightarrow_{tol} \ldots$

Absolute: $$a^T x - b \leq \epsilon_{feas}$$

Relative: $$\frac{a^T x - b}{|b|} \leq \epsilon_{feas}$$

Feasibility and optimality in floating-point solvers

MIP solvers use numerical tolerances, typically in the range 10^{-6} to 10^{-9}:

- Integrality tolerance ϵ_{int} : $\alpha \in \mathbb{Z} \Leftrightarrow_{tol} \alpha \in \mathbb{Z} + [-\epsilon_{int}, \epsilon_{int}]$, e.g., $0.9999999 =_{tol} 1$.
- Feasibility tolerance ϵ_{feas} : $a^T x \leq b \Leftrightarrow_{tol} \dots$

Absolute: $$a^T x - b \leq \epsilon_{feas}$$

Relative: $$\frac{a^T x - b}{|b|} \leq \epsilon_{feas}$$ (problematic for $|b| \approx 0$)

Mixed (SCIP): $$\frac{a^T x - b}{\max\{|b|, 1\}} \leq \epsilon_{feas}$$

Feasibility and optimality in floating-point solvers

MIP solvers use numerical tolerances, typically in the range 10^{-6} to 10^{-9}:

- Integrality tolerance ϵ_{int} : $\alpha \in \mathbb{Z} \Leftrightarrow_{tol} \alpha \in \mathbb{Z} + [-\epsilon_{int}, \epsilon_{int}]$, e.g., $0.9999999 =_{tol} 1$.
- Feasibility tolerance ϵ_{feas} : $a^T x \leq b \Leftrightarrow_{tol} \dots$

Absolute:	$a^T x - b \leq \epsilon_{feas}$	
Relative:	$\dfrac{a^T x - b}{\lvert b \rvert} \leq \epsilon_{feas}$	(problematic for $\lvert b \rvert \approx 0$)
Mixed (SCIP):	$\dfrac{a^T x - b}{\max\{\lvert b \rvert, 1\}} \leq \epsilon_{feas}$	

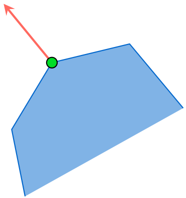
- LP tolerances for dual feasibility, barrier convergence, ...

Note: **not invariant under scaling**!

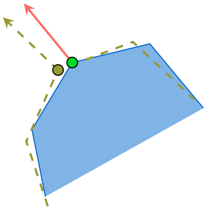
Feasibility and optimality in floating-point solvers

Hope:
Optimal solution with small residual errors is
close to an exact optimal solution without violations.

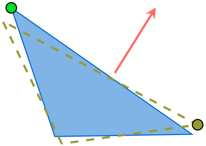
But really: exact solution to a **perturbed problem**



exact solution



good case



bad case

Sources of numerical issues: large big-M's

Example:

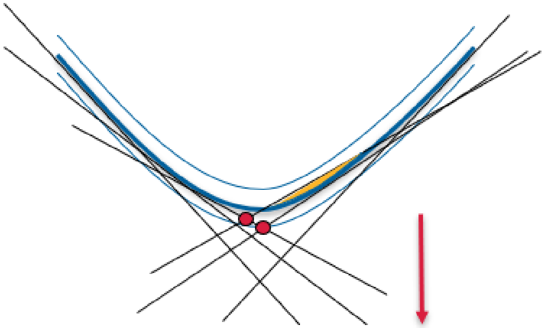
$$\begin{aligned} \min\ & x \\ \text{s.t. } & x \geq 1 \\ & x \leq 10^6 y \\ & y \in \{0,1\} \end{aligned} \qquad\qquad \begin{aligned} \min\ & x \\ \text{s.t. } & x \geq 1 \\ & 10^{-6} x \leq y \\ & y \in \{0,1\} \end{aligned}$$

Assuming an absolute tolerance of 10^{-6}, we have that:

- $x = 1, y = 0$ feasible in the scaled problem w.r.t. tolerances, but infeasible in the original
- $x = 1, y = 10^{-6}$ feasible in both, the scaled and original problem w.r.t. tolerances
- but when you fix $y = 0$ and reoptimize, the result will be infeasible
- $x = y = 1$ is exactly feasible

Sources of numerical issues: in MINLP solving

- Approximating convex functions by cutting planes can yield near-parallel rows in the LP and ill-conditioned basis matrices.
- Relaxations of nonconvex constraints over large domains can yield bad coefficients.
- ...



1. Introduction
2. Floating-point arithmetic and MIP tolerances
3. **Some guidelines and tools**
4. Iterative refinement for LP
5. Solving MIPs exactly

Some guidelines

- Good input, good output
  - Scale data to avoid extreme values: absolute and relative
    look at which units to use, e.g.
    ratio of largest to smallest coefficient $\leq 10^6$ in any row and column
  - Ensure that tolerances make sense relative to the input data.
  - Round insignificant, tiny data values to zero
  - Avoid using truncated or single-precision data

Some guidelines

- Good input, good output
  - Scale data to avoid extreme values: absolute and relative
    look at which units to use, e.g.
    ratio of largest to smallest coefficient $\leq 10^6$ in any row and column
  - Ensure that tolerances make sense relative to the input data.
  - Round insignificant, tiny data values to zero
  - Avoid using truncated or single-precision data
- Modelling and solving
  - Try different scaling parameters
  - Try to avoid large big-M's
  - If you don't have a reasonable *M*, use indicator or SOS constraints
  - If the objective is a hierarchical combination of multiple objective:
    try a sequential approach (akin to the ϵ-constraint method)

Some guidelines

- Good input, good output
  - Scale data to avoid extreme values: absolute and relative
    look at which units to use, e.g.
    ratio of largest to smallest coefficient $\leq 10^6$ in any row and column
  - Ensure that tolerances make sense relative to the input data.
  - Round insignificant, tiny data values to zero
  - Avoid using truncated or single-precision data
- Modelling and solving
  - Try different scaling parameters
  - Try to avoid large big-M's
  - If you don't have a reasonable *M*, use indicator or SOS constraints
  - If the objective is a hierarchical combination of multiple objective:
    try a sequential approach (akin to the ϵ-constraint method)
- Note: Poor scaling and imprecise input are **neither necessary nor sufficient** for numerical problems.

Tools: a posteriori

```
SCIP> checksol

check best solution
solution is feasible in original problem
Violation            :    absolute     relative
  bounds             : 3.18000e-10 3.18000e-10
  integrality        : 3.18000e-10           -
  LP rows            : 1.47428e-09 1.47428e-09
  constraints        : 1.47428e-09 1.47428e-09
```

Tools: a priori

```
Running HiGHS 1.7.2 (git hash: 8fce6250c): Copyright (c) 2024 HiGHS under MIT licence terms
Number of PL entries in BOUNDS section is 45
LP   mwe has 4203 rows; 194 cols; 23776 nonzeros
Coefficient ranges:
  Matrix [1e-09, 1e+07]
  Cost   [8e-02, 2e+00]
  Bound  [1e+03, 1e+06]
  RHS    [1e-13, 4e+11]
WARNING: Problem has excessively large bounds or RHS: consider scaling the bounds and RHS down
by at least 1e+2, or setting option user_bound_scale to -6 or less
```

Tools: check solver log during optimization

```
Warning: Model contains large matrix coefficient range
Consider reformulating model or setting
NumericFocus parameter to avoid numerical issues.
Warning: Markowitz tolerance tightened to 0.5
Warning: switch to quad precision
Numeric error
Numerical trouble encountered
Restart crossover...
Sub-optimal termination
Warning: ... variables dropped from basis
Warning: unscaled primal violation = ... and residual = ...
Warning: unscaled dual violation = ... and residual = ...
```

Tools: condition numbers

- Condition number κ of a matrix:
  bounds how error in the right-hand side can propagate to the solution vector

Tools: condition numbers

- Condition number κ of a matrix:
  bounds how error in the right-hand side can propagate to the solution vector
- For the simplex method: large κ of basis matrices indicates larger errors in the LP solutions

Tools: condition numbers

- Condition number κ of a matrix:
  bounds how error in the right-hand side can propagate to the solution vector
- For the simplex method: large κ of basis matrices indicates larger errors in the LP solutions
- For LP-based branch and bound: can compute or sample a “MIP-κ” / “attention level” / …as a **weighted average of encountered LP-κ’s**

Numerical information – Xpress final report

glass4 with default scaling:

```
Numerical issues encountered:
   Dual failures      :   3410 out of    508042 (ratio: 0.0067)
   Singular bases     :     18 out of    372616 (ratio: 0.0000)
   Nodes kappa stable      :          0 (ratio: 0.0000)
   Nodes kappa suspicious  :          0 (ratio: 0.0000)
   Nodes kappa unstable    :        260 (ratio: 0.0008)
   Nodes kappa ill-posed   :     307910 (ratio: 0.9992)
   Largest kappa seen      : 5.166264e+22
   Kappa attention level   : 0.9994
```

glass4 with SCALING=227 [scale big-M rows]

```
Numerical issues encountered:
   Dual failures      :    683 out of    531681 (ratio: 0.0013)
   Singular bases     :      4 out of    401058 (ratio: 0.0000)
   Nodes kappa stable      :     240371 (ratio: 0.8744)
   Nodes kappa suspicious  :      34412 (ratio: 0.1252)
   Nodes kappa unstable    :        102 (ratio: 0.0004)
   Nodes kappa ill-posed   :          0 (ratio: 0.0000)
   Largest kappa seen      : 1.654603e+12
   Kappa attention level   : 0.0014
```

Gurobi's model analyzer

GUROBI
OPTIMIZATION

Gurobi Model Analyzer

Search

CONTENTS:

Installation

Ill Conditioning Explainer

Quick Start Guide

Ill Conditioning Explainer

- Quick Start Guide
  - Introduction
  - Running the Explainer
  - Interpreting the Output
  - Suggested Usage Quick Start
- Advanced Usage Guide
  - Introduction
  - Interpreting the Explainer Output
  - Additional Function Arguments
- API Reference
  - `gurobi_modelanalyzer.kappa_explain()`

Xpress's solution refiner

Goal: reduce or remove primal, dual and integrality violations in incumbents and final solution by some of

- performing extra simplex iterations
- recomputing in quad precision
- pushing fractional integer variables out of the basis when possible
- performing additional branches to force integer variables to integer values
- fixing integer variables and solve remaining LP

```
    Node      BestSoln     BestBound    Sols Active  Depth     Gap     GInf   Time
*    197   15183.85668  15227.15725      8      1     19    0.28%       0      3
     200   15183.85668  15220.67819      8      1      8    0.24%       5      3
 *** Search completed ***     Time:      4 Nodes:        253
Number of integer feasible solutions found is 8
Best integer solution found is  15183.85668
Best bound is  15183.85668
Refining MIP solution (1.116e-06 fractionality 3.617e-05 abs infeasiblity)
P    253   15183.86782  15183.85668      9      0     10    0.00%       0      4
Refined MIP solution (0.0 fractionality 9.890e-07 abs infeasiblity)
```

MIP-DD: A Delta-Debugger for MIP

Goal: try to reproduce unwanted behavior in a MIP solver, e.g., a numerical issue, on a significantly smaller and easier to analyze MIP

- inspired by **delta debugging** heavily used in the SAT and SMT community (Zeller 1999, Brummayer and Biere 2009, Niemetz and Biere 2013, Kaufmann and Biere 2022, Paxian and Biere 2023)
- very successful in increasing the number of bug fixes in the last SCIP releases
- open-source package MIP-DD available at `github.com/scipopt/mip-dd`
- for details see Hoen, Kampp, G. 2024: "MIP-DD: A Delta Debugger for Mixed Integer Programming Solvers", `arxiv.org/abs/2405.19770v1`

1. Introduction
2. Floating-point arithmetic and MIP tolerances
3. Some guidelines and tools
4. **Iterative refinement for LP**
5. Solving MIPs exactly

Recall

Basic solution: primal-dual

- W.l.o.g. let $rank(A) = m < n$ and consider the computational form

$$\min\{c^T x \mid Ax = b, x \geq 0\} = \max\{y^T b \mid y^T A \leq c, y \text{ free}\}$$

- For every vertex there is a non-singular $m \times m$ sub-matrix B of A

$$A = \begin{bmatrix} B & N \end{bmatrix}$$

- and the corresponding basic solution is given by

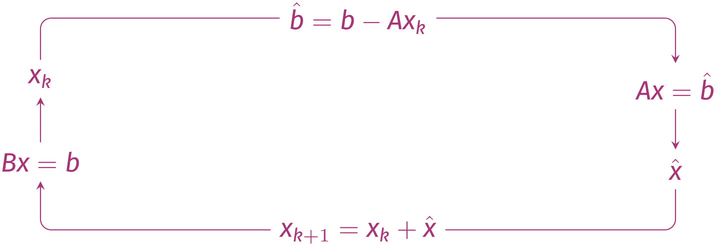
$$x_B = B^{-1}b, \qquad x_N = 0, \qquad y^T = c_B^T B^{-1}$$

- In practice: do not compute inverse, but solve linear systems $Bx_B = b$ and $B^T y = c_B$ by factorization $B = LU$
- Floating-point arithmetic results in residual errors $\hat{b} = b - Bx_B \neq 0$ and $\hat{c} = c_B - B^T y \neq 0$.

Iterative refinement for linear systems

(Wilkinson 1963, Ursic and Patarra 1983, Wan 2006, Pan 2011, Saunders et al. 2011)

- Drop subscript $x_B \rightsquigarrow x$ and suppose we want to solve $Bx = b$
- Idea: compute corrector solution $\hat{x}$ by using residual error as the right-hand side



Hybrid precision method: fast floating-point arithmetic (for linear system solve)
+ slower extended-precision or rational arithmetic (for residual computation and correction)

Iterative refinement for linear programs

(G., Steffy, Wolter 2012, 2016, 2020)

$$
\begin{aligned}
P : \mathbf{min} \quad & c^T x \\
\mathbf{s.\,t.} \quad & Ax = b \\
& x \geq \ell
\end{aligned}
$$

Iterative refinement for linear programs

(G., Steffy, Wolter 2012, 2016, 2020)

$$x_k, y_k$$

$$\uparrow$$

$$\begin{aligned} P : \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq \ell \end{aligned}$$

Iterative refinement for linear programs

(G., Steffy, Wolter 2012, 2016, 2020)

$$\hat{b} = b - Ax_k, \hat{\ell} = \ell - x_k, \hat{c} = c - A^T y_k$$

$$x_k, y_k$$

$$\uparrow$$

$$\begin{aligned} P : \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq \ell \end{aligned}$$

Iterative refinement for linear programs

(G., Steffy, Wolter 2012, 2016, 2020)

$$\hat{b} = b - Ax_k,\ \hat{\ell} = \ell - x_k,\ \hat{c} = c - A^T y_k$$

$$x_k, y_k$$

$$P: \min\ c^T x \quad \text{s.t.}\quad Ax = b,\quad x \geq \ell$$

$$\hat{P}: \min\ \Delta_k \hat{c}^T x \quad \text{s.t.}\quad Ax = \Delta_k \hat{b},\quad x \geq \Delta_k \hat{\ell}$$

Iterative refinement for linear programs

(G., Steffy, Wolter 2012, 2016, 2020)

$$\hat{b} = b - Ax_k, \hat{\ell} = \ell - x_k, \hat{c} = c - A^T y_k$$

$$x_k, y_k$$

$$\begin{aligned} P : \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq \ell \end{aligned}$$

$$\begin{aligned} \hat{P} : \min \quad & \Delta_k \hat{c}^T x \\ \text{s.t.} \quad & Ax = \Delta_k \hat{b} \\ & x \geq \Delta_k \hat{\ell} \end{aligned}$$

$$\hat{x}, \hat{y}$$

Iterative refinement for linear programs

(G., Steffy, Wolter 2012, 2016, 2020)

$$P : \min \; c^T x \quad \text{s.t.} \quad Ax = b, \quad x \geq \ell$$

$$\uparrow$$

$$x_k, y_k$$

$$\hat{b} = b - Ax_k, \; \hat{\ell} = \ell - x_k, \; \hat{c} = c - A^T y_k$$

$$\hat{P} : \min \; \Delta_k \hat{c}^T x \quad \text{s.t.} \quad Ax = \Delta_k \hat{b}, \quad x \geq \Delta_k \hat{\ell}$$

$$\downarrow$$

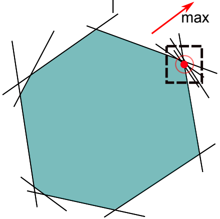
$$\hat{x}, \hat{y}$$

$$x_{k+1} = x_k + \frac{1}{\Delta_k}\hat{x}, \; y_{k+1} = y_k + \frac{1}{\Delta_k}\hat{y}$$

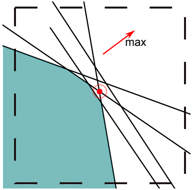
Iterative refinement for linear programs

(G., Steffy, Wolter 2012, 2016, 2020)

$$\hat{b} = b - Ax_k, \hat{\ell} = \ell - x_k, \hat{c} = c - A^T y_k$$



$$\Delta_k \to \infty$$

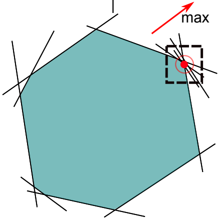


$$x_{k+1} = x_k + \frac{1}{\Delta_k}\hat{x}, y_{k+1} = y_k + \frac{1}{\Delta_k}\hat{y}$$

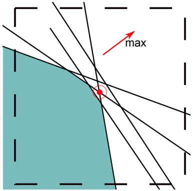
Iterative refinement for linear programs

(G., Steffy, Wolter 2012, 2016, 2020)

$$\hat{b} = b - Ax_k,\ \hat{\ell} = \ell - x_k,\ \hat{c} = c - A^T y_k$$



$$\Delta_k \to \infty$$



$$x_{k+1} = x_k + \frac{1}{\Delta_k}\hat{x},\ y_{k+1} = y_k + \frac{1}{\Delta_k}\hat{y}$$

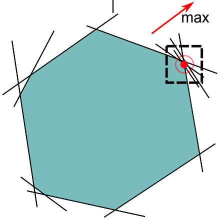
Hybrid precision method: double precision (for simplex)

+ rational arithmetic (for error computation, correction, rounding, LU)

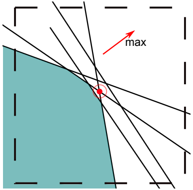
Boosted Iterative refinement for linear programs

(Applegate et al. 2007 + G., Steffy, Wolter 2012, 2016, 2020+ $\to$ G., Nicolas-Thouvenin, Eifler 2024)

$$\hat{b} = b - Ax_k, \hat{\ell} = \ell - x_k, \hat{c} = c - A^T y_k$$



$$\Delta_k \to \infty$$



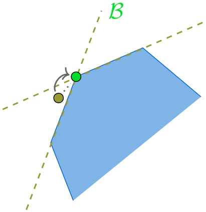
$$x_{k+1} = x_k + \frac{1}{\Delta_k}\hat{x}, y_{k+1} = y_k + \frac{1}{\Delta_k}\hat{y}$$

Hybrid precision method: double + extended precision (for simplex)
+ rational arithmetic (for error computation, correction, rounding, LU)

From accurate to exact

Method 1: Basis Verification

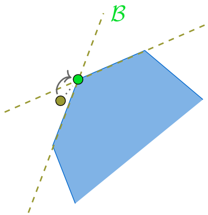
- compute exact basic solution via rat. LU factorization
- check primal and dual feasibility
- keep refining until optimal



From accurate to exact

Method 1: Basis Verification

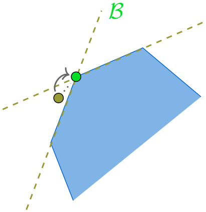
- compute exact basic solution via rat. LU factorization
- check primal and dual feasibility
- keep refining until optimal



Method 2: Output-Sensitive Rational Reconstruction

- round approximate solution by continued fractions approximation
- check primal and dual feasibility and complementary slackness
- increase denominator bound as residual errors decrease

From accurate to exact



Method 1: Basis Verification

- compute exact basic solution via rat. LU factorization
- check primal and dual feasibility
- keep refining until optimal

Method 2: Output-Sensitive Rational Reconstruction

- round approximate solution by continued fractions approximation
- check primal and dual feasibility and complementary slackness
- increase denominator bound as residual errors decrease

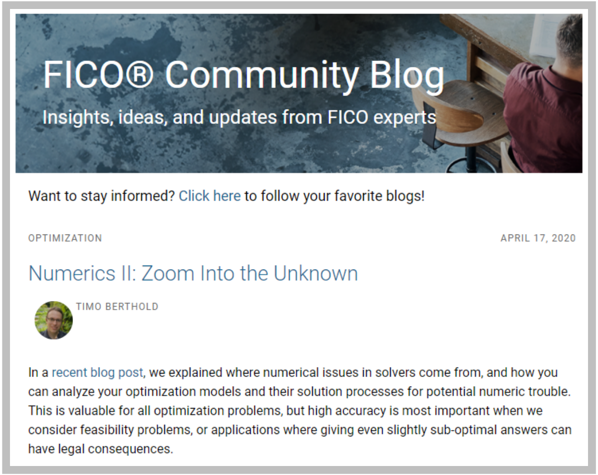
Can prove **oracle-polynomial running time** under some assumptions on the fp solver:
Exact solution reached after a polynomial number of refinements

- Method 1: $O((m^2\langle A\rangle + \langle b, c, \ell\rangle + n^2))$
- Method 2: $O(\max\{\log D, m^2\langle A\rangle\})$

Implemented in the open-source solver SoPlex: `github.com/scipopt/soplex`

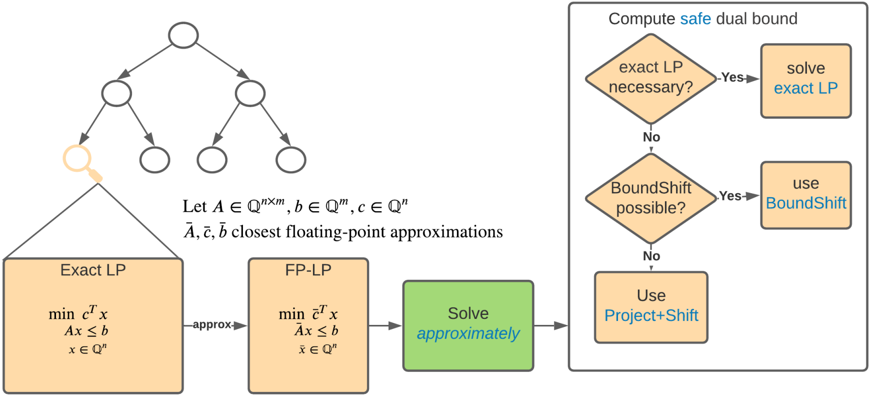
LP iterative refinement in industry

- Iterative refinement for solving linear systems in LP solvers is a standard technique to deal with ill-conditioned basis matrices,
- but also LP iterative refinement has been implemented with quad precision instead of rational arithmetic:
- see `community.fico.com/s/blog-post/a5Q2E000000cDPaUAM/fico2199`



1. Introduction
2. Floating-point arithmetic and MIP tolerances
3. Some guidelines and tools
4. Iterative refinement for LP
5. **Solving MIPs exactly**

From exact LP to exact MIP

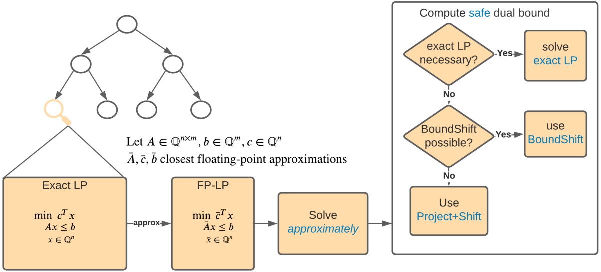


Hybrid-precision branch and bound (Cook, Koch, Steffy, Wolter 2013).

Uses floating-point + directed rounding + rational arithmetic.

Exact SCIP: Hybrid-precision branch-and-bound plus ...

- primal repair heuristics
  (Eifler, G. 2022)
- rational presolving through PaPILO
  (G. Gottwald, Hoen 2023)
- propagation and dual proof analysis (Borst, Eifler, G. 2024)
- MIR cuts (Eifler, G. 2024)
- certificates (Cheung, G., Steffy 2017)



Available at `github.com/scipopt/scip/tree/exact-rational`.