Pinpointing Numerical Issues

A Tool for the Diagnosis and Explanation of Ill Conditioning in LPs and MILPs

Ed Klotz

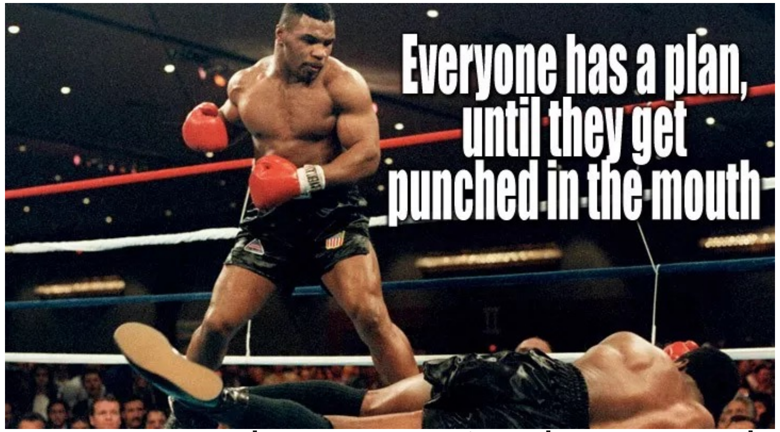
Agenda

- Introduction/Motivation
- Fundamental Concepts
- Three Common Sources of Ill Conditioning
- Quick Start
- **Advanced Usage:** Interpreting Larger Explanations
- Future Development
- Summary/Conclusions

Introduction/ Motivation

Introduction/Motivation

- OR Practitioners have to deal with unexpected, counterintuitive results from the solver



  - Need an organized approach to deal with the unexpected
- We should expand our tools to help explain such results
- We already have the infeasibility finder
- What other explainers might help our users?
- Based on our customer tickets, an Ill Conditioning Explainer would help

Fundamental Concepts

Condition number of a matrix

Computing the condition number κ of solving a linear system $Bx = b$

Change to input

- Perturb rhs $b' = b + \delta b$: $B(x + \delta x) = Bx + B\delta x = b + \delta b$

  $\delta x = B^{-1}\delta b$

Change to output

- Cauchy-Schwarz inequalities:

  $||\delta x|| \leq ||B^{-1}||\ ||\delta b||$

  $||b|| \leq ||B||\ ||x||$

- Combine:  $\frac{||\delta x||}{||x||} \leq ||B|| \cdot ||B^{-1}|| \frac{||\delta b||}{||b||}$ 

  $||B|| \cdot ||B^{-1}||$ → κ

- Condition number of matrix B:
  - Magnification factor of relative input error (rhs in this case, analogous for matrix B)
  - Worst case limit of relative result error given relative rhs error
  - RHS error of $||\delta b|| / ||b|| \sim 10^{-16}$ and $\kappa = 10^{10}$ can cause error up to $||\delta x|| / ||x|| \sim 10^{-6}$

10^{-16} → double precision

10^{-6} → default feasibility tolerance

Don't want solver decisions based on roundoff error

The Basic Idea

GUROBI OPTIMIZATION

- Use the following theorem (Gastinel and Kahan)

  Define $\kappa(B) = \|B\|\|B^{-1}\|$

  Define dist $(B) = min\{{}^{\|\Delta B\|}/_{\|B\|} : \ B + \ \Delta B \ singular\}$ // relative distance to singularity

  Theorem: dist (B) = $\kappa(B)^{-1}$

  Prove by showing that $\kappa(B)^{-1}$ is a lower bound for dist (B), then showing that a perturbation ΔB always exists that attains the lower bound.

  Easy

  Not so easy

- Approximate this by applying FeasRelax to minimize the sum of infeasibilities of

  $B^T y = 0$

  $e^T y = 1$

  $y \ free$

  Normalization of y $\neq 0$

- Rows of B with y component = 0 can be filtered out. **Support of y provides a row-based certificate of the ill conditioning**

Other computations of an explanation of ill conditioning

- Column based explanation
  - Approximate this by applying FeasRelax to minimize the sum of infeasibilities of

$$B\,y = 0$$
$$e^T y = 1$$
$$y \; free$$

    - **<u>Support of y provides a column-based certificate of the ill conditioning.</u>**

- Special case when pairs of almost parallel matrix rows or columns explain the ill conditioning
  - Just look at the angles of pairs of matrix rows and columns. Row or column vectors u and v from a nonsingular matrix are almost parallel if their angle is close to 0:
    - $|u^T v - \| u \| \| v \| | < \epsilon$
- Gurobi-modelanalyzer provides function calls for row, column and angle based explanations of ill conditioned basis matrices

Three Common Sources of Ill Conditioning

Sources of high condition numbers

- #1: Large matrix coefficient ratios
  - Especially rows and columns with common intersection.
  - Example:

Very common submatrix in ill conditioned basis matrices

$$\begin{pmatrix} 1 & 0 \\ M & 1 \end{pmatrix} \text{ has inverse } \begin{pmatrix} 1 & 0 \\ -M & 1 \end{pmatrix}$$

$$\Rightarrow \quad \|B\| \geq M, \|B^{-1}\| \geq M$$

$$\kappa = \|B\| \, \|B^{-1}\| = M^2$$

  - Remedies
    - Avoid unnecessarily large big M values
    - Avoid unnecessarily small units of measurement
      - Result in unnecessarily large values

Sources of high condition numbers

- #2: Inconsistent or unnecessarily truncated problem data calculations
  - Values that are supposed to be the same have slight differences
    - Calculated with different levels of precision in different parts of the model
    - Equivalent calculations under perfect precision may no longer hold under finite precision at the machine epsilon level

c306: **0.416666667** x80 – 100 x90 = 0
…
c11360: **0.416666666** x80 – 100 x90 = 0

Almost parallel rows with linear combination of (1, -1) close to 0

    - Two issues here
      - Values calculated inconsistently
      - Values only calculated to 9 base 10 digits when 64 bit doubles support 16
        - We see a bigger perturbation than is necessary

        - Could actually eliminate all loss of precision in this case (more soon)

Sources of high condition numbers

- Remedy: calculate your data in double or higher precision to limit round-off errors
  - Cancellation of valid digits of small numbers when adding/subtracting numbers of different magnitude due to shifting to equalize the exponents
  - Multiplication and division doesn't have shifted exponents
    - But dividing small numbers into bigger ones can be problematic
- Use integral data wherever possible
  - `bad:                0.3333333 x + 0.6666666 y = 1`
  - `better:0.3333333333333333 x + 0.6666666666666666 y = 1`
  - `best:                             1x + 2y = 3`
- When writing text-based model files (`*.mps`, `*.lp`) make sure to use maximum precision (16 decimal digits) when writing double precision values
  - Gurobi MPS files have higher precision than LP files
  - Row based LP files are more for model inspection than for solving numerically challenging models

Sources of high condition numbers

- Remedy: calculate your data in double or higher precision to limit round-off errors
- Use integral data wherever possible

```
bad:                 0.3333333 x + 0.6666666 y = 1
better:0.3333333333333333 x + 0.6666666666666666 y = 1
best:                         1x + 2y = 3
```

  - **Above coefficients are clearly meant to be 1/3 and 2/3, but what about numbers like 0.3823529412 or 0.1752577319587628 7?**
  - See `https://en.wikipedia.org/wiki/Repeating_decimal` for common truncated decimal representations
  - Fraction.limit_denominator () function from Python fractions package provides fractional representation of repeating decimal representations.

Sources of high condition numbers

- Use integral data wherever possible
  - Fraction.limit_denominator () function from Python fractions package provides fractional representation of repeating decimal values.

```
from fractions import Fraction
def converttofractions(vals):
    for v in vals:
        if not isinstance(v, float):
            print("Value ", v, " is not a float.  Cannot convert.")
            continue
        frac = Fraction(v).limit_denominator()
        print(f"The approx fraction of {v} is {frac}.")
```

```
>>> vals = [0.3823529412, 0.17525773195876287]
>>> converttofractions(vals)
The approx fraction of 0.3823529412 is 13/34.
The approx fraction of 0.17525773195876287 is 17/97.
```

Sources of high condition numbers

- Use integral data wherever possible
  - What about these numbers?

```
>>> vals = [0.8660254037844386, 1.4142135623730951,
0.7071067811865476]
>>> converttofractions(vals)
The approx fraction of 0.8660254037844386 is 489061/564719.
The approx fraction of 1.4142135623730951 is 665857/470832.
The approx fraction of 0.7071067811865476 is 665857/941664.
```

  - These are 16 decimal representations of the irrational numbers sqrt(3)/2, sqrt(2), sqrt(2)/2
    - Rational approximation already gives up accuracy
    - Multiplying by large denominator creates potential additional numerical problems
  - Remedies
    - Make sure to use all 16 base 10 digits
    - Check for rescaling to all rational coefficients

$$\sqrt{2}x_1 + \sqrt{2}/_2\, x_2 = 8\sqrt{8} \leftrightarrow 2\, x_1 + x_2 = 32$$

  - Similar issue with trigonometric data and other irrational values

Sources of high condition numbers

- #3: hidden mixtures of large and small coefficients

$\kappa = \|B\|\,\|B^{-1}\|$ is large if $\|B\|$ is large <u>OR</u> $\|B^{-1}\|$ is large

$\|B^{-1}\|$ can also be large for other reasons

- Cascade

$$\begin{pmatrix} 1 & -2 & & & \\ & 1 & -2 & & \\ & & \ddots & \ddots & \\ & & & 1 & -2 \\ & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} x \end{pmatrix} = \begin{pmatrix} b \end{pmatrix}$$

$$b = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \Rightarrow x = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} = B_1^{-1}$$

$$b = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \Rightarrow x = \begin{pmatrix} 2^{n-1} \\ 2^{n-2} \\ \vdots \\ 2 \\ 1 \end{pmatrix} = B_n^{-1}$$

- And many more non-obvious cases…
- In general, high condition numbers come from almost linear dependent rows or columns

Sources of high condition numbers

- Pivot on B to see the growth

$\kappa = \|B\| \, \|B^{-1}\|$ is large if $\|B\|$ is large **OR** $\|B^{-1}\|$ is large

$$\begin{pmatrix} 1 & -2 & 0 & 0 \\ & 1 & -2 & 0 \\ & & 1 & -2 \\ & & & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 & 0 \\ & 1 & 0 & 0 \\ & & 1 & 0 \\ & & & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & -4 & 0 \\ & 1 & -2 & 0 \\ & & 1 & -2 \\ & & & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 0 & 0 \\ & 1 & 0 & 0 \\ & & 1 & 0 \\ & & & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 & -8 \\ & 1 & 0 & -4 \\ & & 1 & -2 \\ & & & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 4 & 0 \\ & 1 & 2 & 0 \\ & & 1 & 0 \\ & & & 1 \end{pmatrix}$$

$$\Downarrow$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ & 1 & 0 & 0 \\ & & 1 & 0 \\ & & & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & 4 & 8 \\ & 1 & 2 & 4 \\ & & 1 & 2 \\ & & & 1 \end{pmatrix}$$

$$b = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \quad \Rightarrow \quad x = \begin{pmatrix} 2^{n-1} \\ 2^{n-2} \\ \vdots \\ 2 \\ 1 \end{pmatrix} = B_n^{-1}$$

Sources of high condition numbers

- Alternate way to see the hidden large coefficient

$$x_1 = 2x_2$$
$$x_2 = 2x_3$$
$$x_3 = 2x_4$$
$$\vdots$$
$$x_{n-1} = 2x_n$$
$$x_j \geq 0, j = 1, \cdots, n$$

$$x_1 = 2^{n-1} * x_n$$

Sources of high condition numbers

- Remedies for hidden mixtures of large and small coefficients
  - Not as straightforward as the previous sources
  - Need to assess why the activities at the start of the sequence are implicitly being rescaled to much larger values than those at the end of the sequence

$$
\begin{aligned}
x_1 &= 2x_2 \\
x_2 &= 2x_3 \\
x_3 &= 2x_4 \\
&\vdots \\
x_{n-1} &= 2x_n \\
x_j \geq 0, j &= 1, \cdots, n
\end{aligned}
$$

$$x_1 = 2^{n-1} * x_n$$

Quick Start Guide

Quick Start: Installation and Basic Usage

- Installation

  pip install gurobi-modelanalyzer

- Basic usage
  - API functions support other arguments for more advanced usage

```
import gurobipy as gp
import model_analyzer as ma
m=gp.read("myillconditionedmodel.mps")
m.optimize()
ma.kappa_explain(m)                    # row-based explanation; could also specify expltype="ROWS”
ma.kappa_explain(m, expltype="COLS")   # column-based explanation
ma.angle_explain(m)                    # one (or optionally more) tuples of almost parallel rows or columns
```


Quick Start: Interpreting the Output

- A simple, illustrative example
  - Modified the well conditioned NETLIB model afiro by adding a near parallel constraint
    - Row-based explanation

$$B^T y = 0$$
$$e^T y = 1$$
$$y \; free$$

```
GRB_Combined_Row: = -1.75613e-08
(mult=0.5000000001354858)R09bad: - 0.999999999 X02 + X01 - X03 = 0
(mult=0.4999999963548586)R09: X02 - X01 + X03 = 0
(mult=-4.999999859945595e-10)X46: - X03 + 0.109 X22 <= 0
(mult=4.716981000007684e-10)R10: - 1.06 X01 + X04 = 0
(mult=4.716981000007684e-10)R20: - 0.43 X22 + X26 = 0
(mult=-4.716981000007684e-10)X50: X04 + X26 <= 310
(mult=2.573301814737374e-10)X27: X22 <= 500
```

List the rows of the basis corresponding to sorted nonzero elements of y

Prefixed by the y value as a string

Original constraint name

Quick Start: Interpreting the Output

- A simple, illustrative example
  - Modified the well conditioned NETLIB model afiro by adding a near parallel constraint
    - Column-based explanation (Bigger and more cumbersome)

COLUMNS GRB_Combined_Column R09 1.1705685309948421e-09
(mult=1.240802673407655)X04 R10 1 (mult=1.240802673407655)X04 X50 1
(mult=-1.240802673407655)GRBslack_X50 X50 1
(mult=1.170568560989108)X02 R09 1
(mult=1.170568560989108)X02 X21 -1
(mult=1.170568560989108)X02 R09bad -0.999999999
(mult=1.1705685598185422)X01 R09 -1
(mult=1.1705685598185422)X01 R10 -1.06
(mult=1.1705685598185422)X01 X05 1
(mult=1.1705685598185422)X01 R09bad
…
(mult=-0.2516722403609866)X37 X49 -1

$$B\,y = 0$$
$$e^T y = 1$$
$$y \; free$$

List the columns of the basis corresponding to sorted nonzero elements of y

Original variable name

Prefixed by the y value as a string

Quick Start: Interpreting the Output

- A simple, illustrative example
  - Modified the well conditioned NETLIB model afiro by adding a near parallel constraint
    - Angle-based explanation

if $u^T v - \| u \| \| v \| < \epsilon$

Tuples of almost parallel basis rows

Tuples of almost parallel basis columns

```
>>> ma.angle_explain(m)
([(<gurobi.Constr R09>, <gurobi.Constr R09bad>)], [],
 <gurobi.Model Continuous instance basismodel: 28 constrs, 28 vars,
 No parameter changes>)
```

Model of the basis matrix from which the list was derived

Quick Start: Interpreting the Output

- A simple, illustrative example
  - Modified the well conditioned NETLIB model afiro by adding a near parallel constraint

- Takeaways
  - Try row, column and angle based explanations
  - Sometimes one method provides a much smaller explanation than others
  - For most users row or angle based explanations are easier to interpret

Advanced Usage – Interpreting Larger Explanations

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- LP relaxation of (easy) MIPLIB model neos-1603965
  - Problem stats:

Original:

```
Statistics for modelfile:
  Linear constraint matrix    : 28984 Constrs, 15003 Vars, 86947 NZs
  Variable types              : 7004 Continuous,
7999 Integer
  Matrix coefficient range    : [ 0.05, 1e+10 ]
  Objective coefficient range : [ 5, 55650.7 ]
  Variable bound range        : [ 0.5, 1 ]
  RHS coefficient range       : [ 6492, 1e+10 ]
>>>
```

Large matrix and rhs coefficients could cause trouble

Presolved:

```
Statistics for modelfile_pre:
  Linear constraint matrix    : 27980 Constrs, 13999 Vars, 83936 NZs
  Variable types              : 7004 Continuous,
6995 Integer (6995 Binary)
  Matrix coefficient range    : [ 0.05, 192552 ]
  Objective coefficient range : [ 5, 55650.7 ]
  Variable bound range        : [ 0.28125, 240690 ]
  RHS coefficient range       : [ 6017.25, 192552 ]
```

Gurobi's presolve reduces the large coefficients

  - Optimal relaxation basis condition number: 6.700000002285333e+22
    - Only 46552167.69344772 for presolved model
  - Row-based explanation size: 676 rows of basis, 677 columns
  - Column-based explanation size: 3997 rows of basis, 3998 columns
  - Angle_explain routine did not find anything

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- LP relaxation of (easy) MIPLIB model neos-1603965
  - Row-based explanation size: 676 rows of basis, 677 columns
    - Let's have a closer look
    - Most of the constraints just have +-1 coefficients and don't promote any growth in activity levels

      (mult=0.0014925373134321573)R24018: C7034 - C7035 >= 0
      (mult=0.0014925373134321573)R24019: C7035 - C7036 >= 0
      (mult=0.0014925373134321573)R24020: C7036 - C7037 >= 0
      …
    - Unlikely to contribute to the ill conditioning
    - Take a look at the few remaining constraints in the explanation:

      (mult=1.4925373134321572e-13)R8030: C3030 - **1e+10** C7034 <= 0
      (mult=1.4925373134321572e-13)R12700: - C3700 + C3701 - 0.05 C7001 + **1e+10** C7704 <= 1e+10
      (mult=1.4925373134321572e-13)R28684: C3700 = 6630
      (mult=-1.4925373134321572e-13)R28685: - C1701 + C3701 = 7065
      (mult=-1.4179104477605494e-13)R28014: C0030 + C3030 = 7165

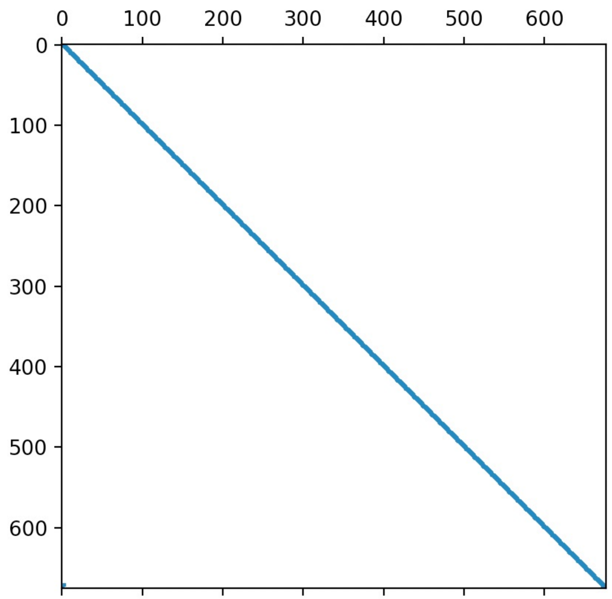
Just looking at these constraints gives a hint at the likely cause without understanding the complete explanation

But let's try to understand the complete explanation

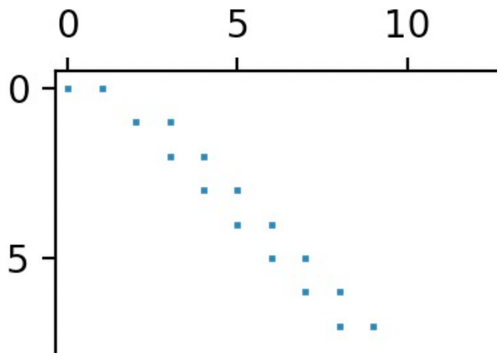
Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

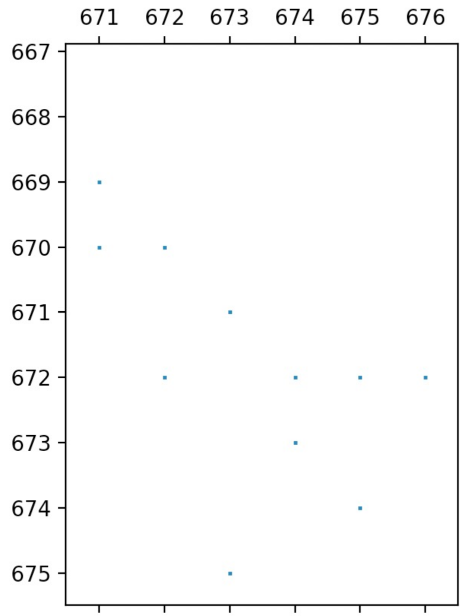
- Closer look at row-based explanation LP relaxation of neos-1603965
  - Bitmap of explanation using matrix_bitmap function in the package:



Top left



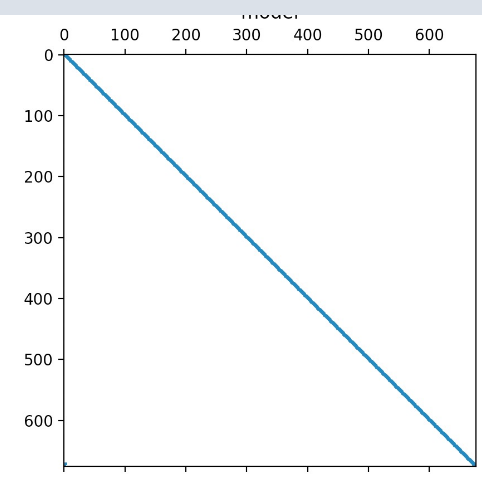
Bottom right



  - Bidiagonal with a few other coefficients at the beginning and end of the sequence of row doubleton constraints

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION



- Closer look at row-based explanation LP relaxation of neos-1603965
  - Row doubletons apparently all cancel except for the first and last variable:

```
(mult=0.0014925373134321573)R24018: C7034 – C7035 >= 0
(mult=0.0014925373134321573)R24019: C7035 – C7036 >= 0
(mult=0.0014925373134321573)R24020: C7036 – C7037 >= 0
…
(mult=0.0014925373134321573)R24120: C7136 – C7137 >= 0
(mult=0.0014925373134321573)R24121: C7137 – C7138 >= 0
(mult=0.0014925373134321573)R24122: C7138 – C7139 >= 0
…
(mult=0.0014925373134321573)R24685: C7701 – C7702 >= 0
(mult=0.0014925373134321573)R24686: C7702 – C7703 >= 0
(mult=0.0014925373134321573)R24687: C7703 – C7704 >= 0
```

  - Running a python script to add up these constraints confirms this.
  - Applying the y multipliers to these constraints:

```
mult=0.0014925373134321573)combined: C7034 – C7704 >= 0
```


Examples: Interpreting Larger Explanations

- Closer look at row-based explanation LP relaxation of neos-1603965
  - Despite the 676 rows in the explanation, we only need focus on these:

$$\begin{pmatrix} 1 & 0 \\ M & 1 \end{pmatrix} \text{ has inverse } \begin{pmatrix} 1 & 0 \\ -M & 1 \end{pmatrix}$$

$$\Rightarrow \quad \|B\| \geq M, \|B^{-1}\| \geq M$$

$$\kappa = \|B\| \, \|B^{-1}\| \geq M^2$$

mult=0.0014925373134321573)combined: ***C7034 - C7704*** >= 0
(mult=1.4925373134321572e-13)R8030: C3030 - **1e+10** C7034 <= 0
(mult=1.4925373134321572e-13)R12700: - C3700 + C3701 - 0.05 C7001 + **1e+10** C7704 <= 1e+10
(mult=1.4925373134321572e-13)R28684: C3700 = 6630
(mult=-1.4925373134321572e-13)R28685: - C1701 + C3701 = 7065
(mult=-1.4179104477605494e-13)R28014: C0030 + C3030 = 7165

```
Here’s what’s left:
GRB_Combined_Row: − 1.41791e−13 C0030 + 1.49254e−13 C1701
```

$$B^T y = 0$$
$$y \neq 0$$
$$y \; free$$

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- Summary and takeaways for LP relaxation of (easy) MIPLIB model neos-1603965
  - Unnecessarily large big M values in original formulation result in ill conditioned root LP relaxation
    - Fortunately Gurobi's presolve was able to reduce the big Ms to more reasonable values, so they were not problematic for the MIP optimization
      - But Gurobi may not always be able to do this, so understanding how to reduce them and whether they are problematic can be helpful
        - Gurobi Days Advanced Numerics Presentation describes how to do this
  - Ill conditioning explainer takeaways
    - Don't always need to completely understand every constraint in the explanation
      - Look for signs or large ratios, imprecisely truncated/rounded data, or cascades of constraints in subsets of the explanation
    - Use the matrix_bitmap to help understand the full explanation
    - Constraints with only +-1 coefficients are probably intermediate constraints that aren't the actual source of ill conditioning

Examples: Interpreting Larger Explanations

- LP Subproblem created by Gurobi from open MINLPLIB model topopt-cantilever_60x40_50
  - Problem stats indicate benign coefficients:

```
Linear constraint matrix    : 11923 Constrs, 33600 Vars, 223008 NZs
Matrix coefficient range    : [ 0.0664586, 2 ]
Objective coefficient range : [ 0, 0 ]
Variable bound range        : [ 1, 1 ]
RHS coefficient range       : [ 1, 1200 ]
```

  - Closer examination of problem data indicates no signs of imprecise data
  - All sorts of inconsistent results (feasible or infeasible) depending on Presolve and LP algorithm settings
  - Optimal basis condition number (solving with presolve off) 1.0128250880948158e+33
  - Row-based explanation size: 26 rows of basis, 26 columns
  - Column-based explanation size: 525 rows of basis, 438 columns
  - Angle_explain routine did not find anything

Examples: Interpreting Larger Explanations

- LP Subproblem created by Gurobi from open MINLPLIB model topopt-cantilever_60x40_50
  - Row-based explanation (variables have domains of $(-\infty, \infty)$):

$$x_1 = 2x_2,\quad x_2 = 2x_3,\quad x_3 = 2x_4,\quad \vdots,\quad x_{n-1} = 2x_n,\quad x_j \geq 0, j = 1, \cdots, n \;\rightarrow\; x_1 = 2^{n-1} * x_n$$

(mult=1.267949192397407)e11923: 0.221528652 x33590 = 0
(mult=-0.3397459621039425)e11803: 0.221528652 x32870 + 0.8267561847 x33590 = 0
(mult=0.09103465616606164)e11683: 0.221528652 x32150 + 0.8267561847 x32870 = 0
(mult=-0.02439266259987994)e11563: 0.221528652 x31430 + 0.8267561847 x32150 = 0
(mult=0.006535994244062434)e11443: 0.221528652 x30710 + 0.8267561847 x31430 = 0
(mult=-0.001751314379211212)e11323: 0.221528652 x29990 + 0.8267561847 x30710 = 0
(mult=0.0004692632735437659 6)e11203: 0.221528652 x29270 + 0.8267561847 x29990 = 0
…
(mult=4.614778914917941e-12)e9523: 0.221528652 x19190 + 0.8267561847 x19910 = 0
(mult=-1.236526283345254 8e-12)e9403: 0.221528652 x18470 + 0.8267561847 x19190 = 0
(mult=3.313262190006382 5e-13)e9283: 0.221528652 x17750 + 0.8267561847 x18470 = 0

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- LP Subproblem created by Gurobi from open MINLPLIB model topopt-cantilever_60x40_50
  - Row-based explanation (variables have domains of $(-\infty, \infty)$):

Let alpha = 0.8267561847/0.221528652 = 3.7320508080372377

(mult=1.267949192397407)e11923: x33590 = 0
(mult=-0.3397459621039425)e11803: x32870 + alpha x33590 = 0
(mult=0.09103465616606164)e11683: x32150 + alpha x32870 = 0
(mult=-0.02439266259987994)e11563: x31430 + alpha x32150 = 0
(mult=0.006535994244062434)e11443: x30710 + alpha x31430 = 0
(mult=-0.0017513143792112112)e11323: x29990 + alpha x30710 = 0
(mult=0.00046926327354376596)e11203: x29270+ alpha x29990 = 0
...

$$
\begin{aligned}
x_1 &= 2x_2 \\
x_2 &= 2x_3 \\
x_3 &= 2x_4 \\
&\vdots \\
x_{n-1} &= 2x_n \\
x_j &\geq 0, j = 1, \cdots, n
\end{aligned}
\quad \Rightarrow \quad x_1 = 2^{n-1} * x_n
$$

This has the same structure as the description of the cascade in the introduction. The only difference is that the multiple is 3.7320508080372377 instead of 2.0

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- https://plato.asu.edu/ftp/lptestset/irish-electricity.mps.bz2
  - From the Hans Mittelmann LP test set
  - Problem Stats:

Original:

```
Statistics for modelirish-electricity.mps:
  Linear constraint matrix    : 104259 Constrs, 61728 Vars, 523257 NZs
  Matrix coefficient range    : [ 1, 24827 ]
  Objective coefficient range : [ 1, 1 ]
  Variable bound range        : [ 1, 1 ]
  RHS coefficient range       : [ 0.333333, 4502.3 ]
```

Presolved:

```
Statistics for modelirish-electricity.mps_pre:
  Linear constraint matrix    : 69780 Constrs, 40306 Vars, 409389 NZs
  Matrix coefficient range    : [ 1, 445 ]
  Objective coefficient range : [ 9.10796, 24704.4 ]
  Variable bound range        : [ 0.333333, 445.446 ]
  RHS coefficient range       : [ 1, 4502.3 ]
```

No potential large matrix row or column ratios

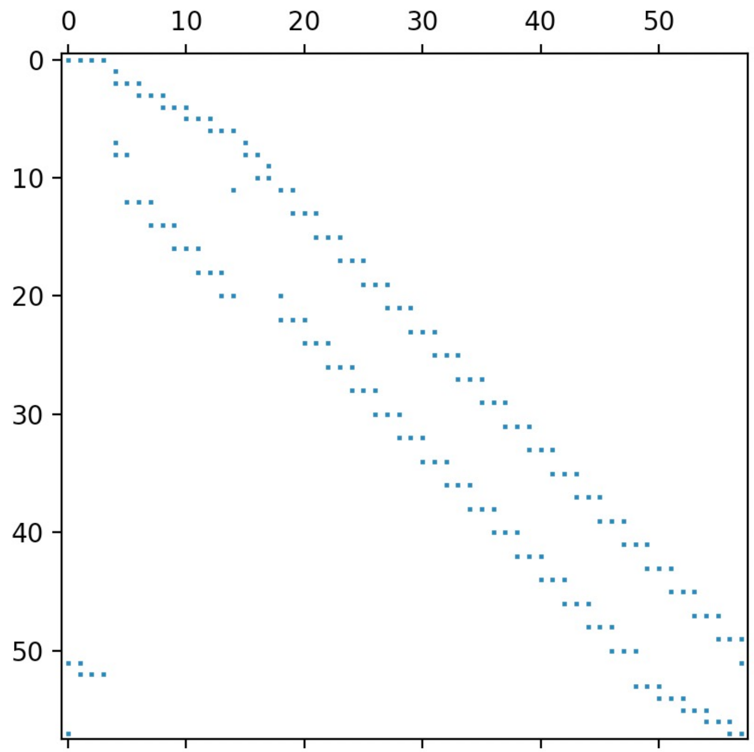
  - Optimal basis condition number: 2.22011060456565 3e+38
    - For presolved model: 4320535.811082865
  - Row-based explanation size: 57 rows of basis, 58 columns
  - Column-based explanation size: 4698 rows of basis, with 4431 columns
  - Angle_explain routine did not find anything

Condition numbers above 1e+30 so far have meant cascades

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- https://plato.asu.edu/ftp/lptestset/irish-electricity.mps.bz2
  - Row-based explanation looks more promising
    - Bitmap:



Pattern is consistent with a more elaborate cascade of constraints

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- irish-electricity.mps.bz2
  - Connection to cascade not immediately clear

Single power source? All constraints and variables have the same index of 24

Time period

$$
\begin{aligned}
x_1 &= 2x_2 \\
x_2 &= 2x_3 \\
x_3 &= 2x_4 \\
&\vdots \\
x_{n-1} &= 2x_n \\
x_j \geq 0, j &= 1, \cdots, n
\end{aligned}
$$

$$x_1 = 2^{n-1} * x_n$$

Link and RampUpSlow constraints for consecutive time periods are connected, but ordering in explanation seems disorganized

```
(mult=1.473670931920553)MinDownInit_24: v_24_13 = 0
(mult=-0.301174019398261)Link_u_v_24_14: v_24_13 + u_24_13 - v_24_14 >= 0
(mult=-0.1150382388171555)Link_u_v_24_15: v_24_14 + u_24_14 - v_24_15 >= 0
(mult=-0.04394069724688566)Link_u_v_24_16: v_24_15 + u_24_15 - v_24_16 >= 0
(mult=-0.01678385285894142)Link_u_v_24_17: v_24_16 + u_24_16 - v_24_17 >= 0
(mult=-0.006410861329938599)Link_u_v_24_18: v_24_17 + u_24_17 - v_24_18 >= 0

…
(mult=0.001128095639494215)RampUpSlow_P_24_14: - 165 u_24_13 - 165 v_24_14 + 165 u_24_14 <= 0
(mult=-0.0009353320626839298)Link_u_v_24_20: v_24_19 + u_24_19 - v_24_20 >= 0
(mult=0.0004308941917262418)RampUpSlow_P_24_15: - 165 u_24_14 - 165 v_24_15 + 165 u_24_15 <= 0
(mult=-0.00035726505717771395)Link_u_v_24_21: v_24_20 + u_24_20 - v_24_21 >= 0
(mult=0.00016458693568451054)RampUpSlow_P_24_16: - 165 u_24_15 - 165 v_24_16 + 165 u_24_16 <= 0
…
```


Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- irish-electricity.mps.bz2: How can we improve the ordering of the explanation?
  - $\kappa = \|B\|\,\|B^{-1}\|$ is large if $\|B\|$ is large **OR** $\|B^{-1}\|$ is large
  - Problem stats indicate tha $\|B\|$ is modest, so $\|B^{-1}\|$ must be large
    - Primal and dual variables involve linear combinations of columns and rows of $\mathbf{B}^{-1}$
      - $x = B^{-1}(b - A_N x_N); \quad y = c_B^T B^{-1}$
      - Large primal and dual values might help identify the order of variables or constraints in the explanation
      - No large primal variables, but we do find some large dual variables with the same power source and time period

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- irish-electricity.mps.bz2: How can we improve the ordering of the explanation?
  - No large primal variables, but we do find some large dual variables with the same power source and time period.

```
Constraint  Link_u_v_24_91  has value  108230.91037595687
Constraint  Link_u_v_24_90  has value  285128.83028436353
Constraint  Link_u_v_24_89  has value  748726.9041354554

…
Constraint  Link_u_v_24_16  has value  2.439434013360222e+36
Constraint  Link_u_v_24_15  has value  6.386521160289626e+36
Constraint  Link_u_v_24_14  has value  1.6720129467508656e+37

Constraint  RampUpSlow_P_24_85  has value  -131988.62730856577
Constraint  RampUpSlow_P_24_84  has value  -345543.70894560404
Constraint  RampUpSlow_P_24_83  has value  -904639.0794566364
…
Constraint  RampUpSlow_P_24_16  has value  -9.137291718601104e+33
Constraint  RampUpSlow_P_24_15  has value  -2.392174028442063e+34
Constraint  RampUpSlow_P_24_14  has value  -6.262792913466078e+34
Constraint  RampUpSlow_P_24_13  has value  -1.6396204711956172e+35
```

Not in the explanation

Dual variables grow as time period decreases

Reorder explanation with Link and Rampup pairs ordered by time period

In the explanation

Examples: Interpreting Larger Explanations

- irish-electricity.mps.bz2
  - Row-based explanation with Link and Rampup constraints in ordered by time period

```
(mult=-0.11503823888171555)Link_u_v_24_15: v_24_14 + u_24_14 - v_24_15 >= 0
(mult=0.00043089419172624l8)RampUpSlow_P_24_15: - 165 u_24_14 - 165 v_24_15 + 165 u_24_15 <= 0
(mult=-0.04394069724688566)Link_u_v_24_16: v_24_15 + u_24_15 - v_24_16 >= 0
(mult=0.00016458693568451054)RampUpSlow_P_24_16: - 165 u_24_15 - 165 v_24_16 + 165 u_24_16 <= 0
(mult=-0.01678385285894142)Link_u_v_24_17: v_24_16 + u_24_16 - v_24_17 >= 0
…
(mult=2.738820795076575e-13)RampUpSlow_P_24_37: - 165 u_24_36 - 165 v_24_37 + 165 u_24_37 <= 0
(mult=-2.824408944922718e-11)Link_u_v_24_38: v_24_37 + u_24_37 - v_24_38 >= 0
(mult=1.0270577981537156e-13)RampUpSlow_P_24_38: - 165 u_24_37 - 165 v_24_38 + 165 u_24_38 <= 0
(mult=-1.1297635779690872e-11)Link_u_v_24_39: v_24_38 + u_24_38 - v_24_39 >= 0
```

Absolute Link and RampUp multipliers get smaller as time period increases

Start with the last time period and work backwards

Examples: Interpreting Larger Explanations

- irish-electricity.mps.bz2
  - Start with the last time period and work backwards

```
-165  *  Link_u_v_24_39: v_24_38 + u_24_38 - v_24_39 >= 0       # sense changes to <=
+
RampUpSlow_P_24_38: - 165 u_24_37 - 165 v_24_38 + 165 u_24_38 <= 0

combinedcon0:  - 165 u_24_37 - 330 v_24_38 + 165 v_24_39 <= 0
+
-330 * Link_u_v_24_38: v_24_37 + u_24_37 - v_24_38 >= 0          # sense changes to <=

combinedcon1:  - 495 u_24_37 + -330 v24_37 + 165 v_24_39 <= 0
+
3 * RampUpSlow_P_24_37: - 165 u_24_36 - 165 v_24_37 + 165 u_24_37 <= 0

combinedcon2:  -825 v24_37 -495 u_24_36  + 165 v_24_39 <= 0
```


Examples: Interpreting Larger Explanations

- irish-electricity.mps.bz2
  - Start with the last time period and work backwards

```
combinedcon2:  -825 v24_37 - 495 u_24_36  + 165 v_24_39 <= 0
+
-825 * Link_u_v_24_37: v_24_36 + u_24_36 - v_24_37 >= 0          # sense changes

combinedcon3: -1320 u_24_36  - 825 v24_36 + 165 v_24_39 <= 0
+
8 * RampUpSlow_P_24_36: - 165 u_24_35 - 165 v_24_36 + 165 u_24_36 <= 0

combinedcon4:   -2145 v24_36 -1320 u_24_35+ 165 v_24_39 <= 0
+
-2145 * Link_u_v_24_36: v_24_35 + u_24_35 - v_24_36 >= 0         # sense changes

Combinedcon5:  -3465 u_24_35 - 2145 v_24_35 + 165 v24_39 <= 0
```


Examples: Interpreting Larger Explanations

- irish-electricity.mps.bz2
  - A pattern emerges:

```
combinedcon0:  – 330 v_24_38 – 165 u_24_37 + 165 v_24_39 <= 0
Combinedcon1: – 495 u_24_37 + –330 v24_37 + 165 v_24_39 <= 0
combinedcon2:  –825 v24_37 – 495 u_24_36  + 165 v_24_39 <= 0
combinedcon3: –1320 u_24_36  – 825 v24_36 + 165 v_24_39 <= 0
combinedcon4:   –2145 v24_36 –1320 u_24_35+ 165 v_24_39 <= 0
Combinedcon5:  –3465 u_24_35 – 2145 v_24_35 + 165 v24_39 <= 0
```

Largest coefficient Multiple of -165
2
3
5
8
13
21

Growing pretty rapidly, but is it exponential?

Look familiar?

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- irish-electricity.mps.bz2
  - Exponential growth?

Largest coefficient
Multiple of -165

```
combinedcon0:  − 330 v_24_38 − 165 u_24_37 + 165 v_24_39 <= 0          2
combinedcon1: − 495 u_24_37 + −330 v24_37 + 165 v_24_39 <= 0           3
combinedcon2:  −825 v24_37 − 495 u_24_36  + 165 v_24_39 <= 0           5
combinedcon3: −1320 u_24_36  − 825 v24_36 + 165 v_24_39 <= 0           8
combinedcon4:   −2145 v24_36 −1320 u_24_35+ 165 v_24_39 <= 0          13
Combinedcon5:  −3465 u_24_35 − 2145 v_24_35 + 165 v24_39 <= 0         21
```

The Fibonacci sequence: **0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, …**

$$F_n = F_{n-1} + F_{n-2}$$

Binet's formula: $F_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$

Yes, exponential growth, which explains the huge condition number and dual variable values

Examples: Interpreting Larger Explanations

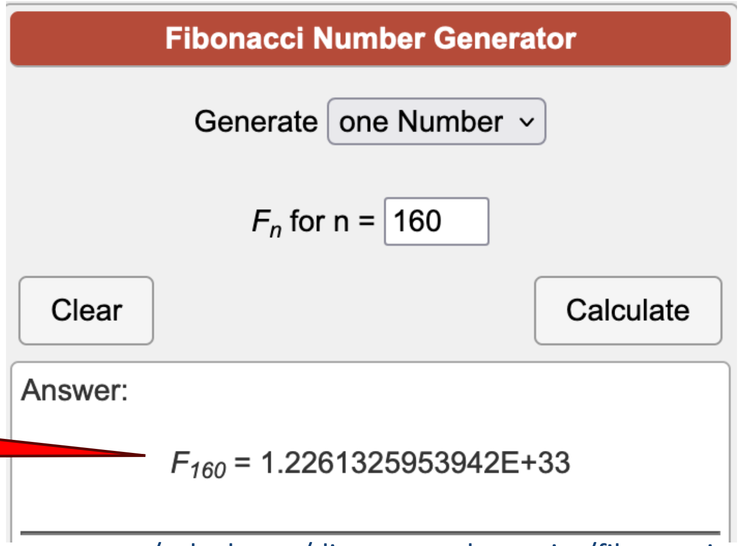
GUROBI OPTIMIZATION

- [irish-electricity.mps.bz2]
  - Yes, exponential growth

Largest coefficient
Multiple of -165

```
combinedcon0:  - 330 v_24_38 - 165 u_24_37 + 165 v_24_39 <= 0          2
combinedcon1: - 495 u_24_37 + -330 v24_37 + 165 v_24_39 <= 0            3
combinedcon2:  -825 v24_37 - 495 u_24_36  + 165 v_24_39 <= 0            5
combinedcon3: -1320 u_24_36  - 825 v24_36 + 165 v_24_39 <= 0            8
combinedcon4:   -2145 v24_36 -1320 u_24_35+ 165 v_24_39 <= 0           13
Combinedcon5:  -3465 u_24_35 - 2145 v_24_35 + 165 v24_39 <= 0          21
```

Binet's formula: $$F_n = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right)$$



https://www.calculatorsoup.com/calculators/discretemathematics/fibonacci-calculator.php

Two combined constraints for 80 time periods

In line with the dual values and basis condition number

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- irish-electricity.mps.bz2
  - We have a better understanding of the cause of the large basis condition number for the original, unpresolved model, but what does that tell us about the model itself?
    - Q: Why did the optimal basis for the presolved model have a much smaller condition number of 1e+6?
      - A: Examination of the LP file of the presolved model reveals that presolve removed all the u24_* and v_24* variables that were basic
    - Q: The original model had huge dual values at optimality, but no such huge primal values. Why?
      - A: Examination of the solution and basis files at optimality reveal that all the u24_* and v_24* basic variables that created the growth in basis inverse coefficients were basic at their lower bound of 0
    - Takeaway for the model
      - The huge condition number was relatively harmless in the sense it didn't adversely affect solution quality
      - However, a small change to the model that causes the u24_* and v24_* variables to take on positive values could result in huge primal values that probably don't make sense in the physical system
      - Will any reasonable changes to the model continue to ensure the stay basic at 0? If so, the high condition number is probably harmless. If not, then small changes to the model could make the high optimal basis condition number more problematic

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- irish-electricity.mps.bz2 - Lessons for ill conditioning explanation interpretation
  - Cascades may be harder to interpret as a source of ill conditioning than large ratios in basis rows or columns or imprecise rounding
  - Based on (admittedly anecdotal) experience, condition numbers above 1e+30 usually correspond to cascades rather than large ratios or imprecise rounding
  - Patterns in the multipliers associated with the certificate of ill conditioning may help with the investigation
  - Large primal or dual values associated with the solution associated with the ill conditioned basis may help with the interpretation of the explanation
  - May need to examine the underlying pivoting operations associated with computing the basis inverse
    - $\kappa = ||B|| \, ||B^{-1}||$ is large if $||B||$ is large <u>OR</u> $||B^{-1}||$ is large

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- LP relaxation of (hard) MIPLIB2010 model ns2122603.mps.bz2
  - From the “unstable” set at https://miplib2010.zib.de/miplib2010-unstable.php
  - Problem Stats:

Original:

```
Linear constraint matrix    : 24754 Constrs, 19300 Vars, 77044 NZs
Variable types              : 11712 Continuous, 7588 Integer
Matrix coefficient range    : [ 0.0416667, 1e+08 ]
Objective coefficient range : [ 1, 450 ]
Variable bound range        : [ 1, 1 ]
RHS coefficient range       : [ 0.0208334, 1e+08 ]
```

Presolved:

```
Linear constraint matrix    : 18936 Constrs, 15492 Vars, 63704
Variable types              : 7365 Continuous, 8127 Integer (6771 Binary)
Matrix coefficient range    : [ 0.016992, 1e+08 ]
Objective coefficient range : [ 0.0416667, 1e+08 ]
Variable bound range        : [ 0.0208333, 1.295e+08 ]
RHS coefficient range       : [ 0.0208333, 1.295e+08 ]
```

Large matrix range, possible imprecise rounding

Presolved model doesn’t look any better

  - Optimal relaxation basis condition number: 5624826146.234143
    - 4361746801650.753 for presolved model
  - Row-based explanation size: 129 rows of basis, 129 columns
  - Column-based explanation size: 2095 rows of basis, 2033 columns
  - Angle_explain routine did not find anything

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- LP relaxation of (hard) MIPLIB2010 model ns2122603.mps.bz2
  - Row-based explanation size: 129 rows of basis, 129 columns
    - Let’s have a closer look
    - Don’t need to understand the complete cause of the high condition number
    - There are obvious issues with imprecise rounding of repeating fractions and large big M values that probably can be reduced

1/24 using the converttofractions routine

```
(mult=0.023136276439040208)R1109: C5274 - C5275 - 0.0416667 C13063 >= 0
(mult=0.023136276439040208)R1110: C5275 - C5276 - 0.0416667 C13064 >= 0
(mult=0.023135065749535288)R1107: C5272 - C5273 - 0.0416667 C13061 >= 0
…
(mult=0.005919383956597007)R4893: C5279 - 8.750007 C13068 <= 0.02083335
(mult=0.005828887109140261)R4892: C5278 - 8.750007 C13067 <= 0.02083335
(mult=0.005738821199052356)R4891: C5277 - 8.750007 C13066 <= 0.02083335
…
(mult=8.545751103325234e-12)R12422: - 0.0416667 C10510 - 1e+08 C11770
    + 1e+08 C13030 >= -1e+08
```

Worse; the large coefficients will magnify the imprecise rounding. croutine

Examples: Interpreting Larger Explanations

- LP relaxation of (hard) MIPLIB2010 model ns2122603.mps.bz2
  - Remedies
    - Use the exact fractional representation and rescale:

```
(mult=0.0231362764390402O8)R1109: C5274 – C5275 – 0.0416667 C13063 >= 0

(mult=0.0231362764390402O8)R1109: C5274 – C5275 – 1/24 C13063 >= 0

(mult=0.0231362764390402O8)R1109: 24 C5274 – 24 C5275 – C13063 >= 0


(mult=0.005919383956597007)R4893: C5279 – 8.750007 C13068 <= 0.02083335

(mult=0.005919383956597007)R4893: C5279 – 8.750000 C13068 <= 0.02083333

(mult=0.005919383956597007)R4893: C5279 – 8.750000 C13068 <= 1/48

(mult=0.005919383956597007)R4893: 48 C5279 – 420 C13068 <= 1


…
(mult=8.545751103325234e-12)R12422: – 0.0416667 C10510 – 1e+08 C11770
    + 1e+08 C13030 >= –1e+08
```

1/48 using the converttofractions routine

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- LP relaxation of (hard) MIPLIB2010 model ns2122603.mps.bz2
  - Remedies (ctd)
    - Use the exact fractional representation and rescale

```
(mult=8.545751103325234e-12)R12422: - 0.0416667 C10510 - 1e+08 C1177
   + 1e+08 C13030 >= -1e+08
```

↓

```
(mult=8.545751103325234e-12)R12422: - 1/24 C10510 - 1e+08 C11770
   + 1e+08 C13030 >= -1e+08
```

↓

```
(mult=8.545751103325234e-12)R12422: - C10510 - 2.4e+09 C11770
   + 2.4e+09 C13030 >= -2.4e+09
```

↓

$[0,\infty)$ $\{0,1\}$ $\{0,1\}$

```
(mult=8.545751103325234e-12)R12422: C10510 <= 2.4e+09 (1 - C11770) + 2.4e+09 C13030
```

↓

$\{0,1\}$

$$Z \geq C11770 - C13030$$
$$Z \leq C11770$$
$$Z \leq 1 - C13030$$
$$Z == 0 \Rightarrow C10510 \leq 0$$

Replace big Ms with indicator constraint or derive tighter bound on C10510 to reduce big M

Examples: Interpreting Larger Explanations

GUROBI OPTIMIZATION

- LP relaxation of (hard) MIPLIB2010 model ns2122603.mps.bz2
  - Remedies (ctd)
    - Use the exact fractional representation and rescale:

```
(mult=0.023136276439040208)R1109: C5274 – C5275 – 0.0416667 C13063 >= 0
(mult=0.023136276439040208)R1109: C5274 – C5275 – 1/24 C13063 >= 0
(mult=0.023136276439040208)R1109: 24 C5274 – 24 C5275 – C13063 >= 0

(mult=0.005919383956597007)R4893: C5279 – 8.750007 C13068 <= 0.02083335
(mult=0.005919383956597007)R4893: C5279 – 8.750000 C13068 <= 0.02083333
(mult=0.005919383956597007)R4893: C5279 – 8.750000 C13068 <= 1/48
(mult=0.005919383956597007)R4893: 48 C5279 – 420 C13068 <= 1

…
(mult=8.545751103325234e-12)R12422: – 0.0416667 C10510 – 1e+08 C11770
   + 1e+08 C13030 >= -1e+08
```

1/48 using the converttofractions routine

Worse; the large coefficients will magnify the imprecise rounding. croutine

…g the converttofraction… routine

Roadmap

Future Developments

- Investigate ways to reduce explanation size
  - Automate the manual combining of constraints illustrated in the neos-1603965 example
  - Allow more informative filtering of constraints with small multipliers
- Consider eigenvalue/singular value based methods
  - Perhaps they will yield better explanations
- Based on user response, decide whether to incorporate into the C engine so it can be used within applications written in any Gurobi API
  - In contrast to the current Python implementation which is more of a model development/troubleshooting tool

Summary/Conclusions

Summary/Conclusions

- 3 different methods (row, column and angle) to compute an explanation of an ill conditioned basis
  - Different methods provide better explanations for different models
- 3 common sources of ill conditioning
  - Large basis matrix row or column ratios
  - Imprecise rounding, particularly of repeating fractions
  - Cascades of constraints with benign coefficients that implicitly hide a large ratio
- Explanations can be significant in size, not always easy to interpret
  - Don't always need to understand the full explanation
  - Look for signs of the 3 common sources
  - Combine constraints to reduce the size of the explanation
  - Multipliers associated with the certificate of ill conditioning can help in the interpretation

Questions?

Backup

Current State of Gurobi's Explanations

So why is my model infeasible?

Our Infeasibility Finder can provide identify a (hopefully) small subset of constraints that explains why your model is infeasible.   Or you can try our FeasRelax feature to look at constraints that need to be relaxed to explain the infeasibility; this could help if the infeasibility finder yields a large set of constraints that are difficult to interpret.

Thanks, I'll give it a shot.

Current State of Gurobi's Explanations

So why does Gurobi yield inconsistent results?

Our Kappa attribute indicates that your model has ill conditioned basis matrices.   This means that rounding errors in the finite precision computations the solver does can be magnified to values larger than the solver tolerances, leading to inconsistent results because of algorithmic decisions based on rounding errors. You should fix that.

If you aren't familiar with ill conditioning and finite precision, try reading https://pubsonline.informs.org/doi/10.1287/educ.2014.0130, or Accuracy and Stability of Numerical Algorithms by Higham

Ehhhh….ok?

- TODO: Add a few slides illustrating a column based explanation
- /models/miplib2010-1.0.2/instances/miplib2010/umts.mps.bz2 is a candidate

What does a basis actually mean?

- For those of us who think in terms of Linear Programs and the Simplex Method
  - A set of variables that are the only ones that can take on nonzero values
  - The variables or hyperplanes associated with a vertex solution in the simplex method
  - The subset of the constraint matrix columns that form the matrix that is factorized in the revised simplex method
- For those who think in terms of Linear Algebra, a basis is a coordinate system
  - A change in basis is a change in the coordinate system
  - A basis is a linear transformation from one coordinate system to another
  - Such a change or transformation will alter the view of an object in the coordinate system
  - The simplex method generates a sequence of different views of the polyhedron that defines the feasible region
    - Some may be more distorted than others
    - Highly distorted means the associated basis is ill conditioned.

What does a basis actually mean?

GUROBI OPTIMIZATION

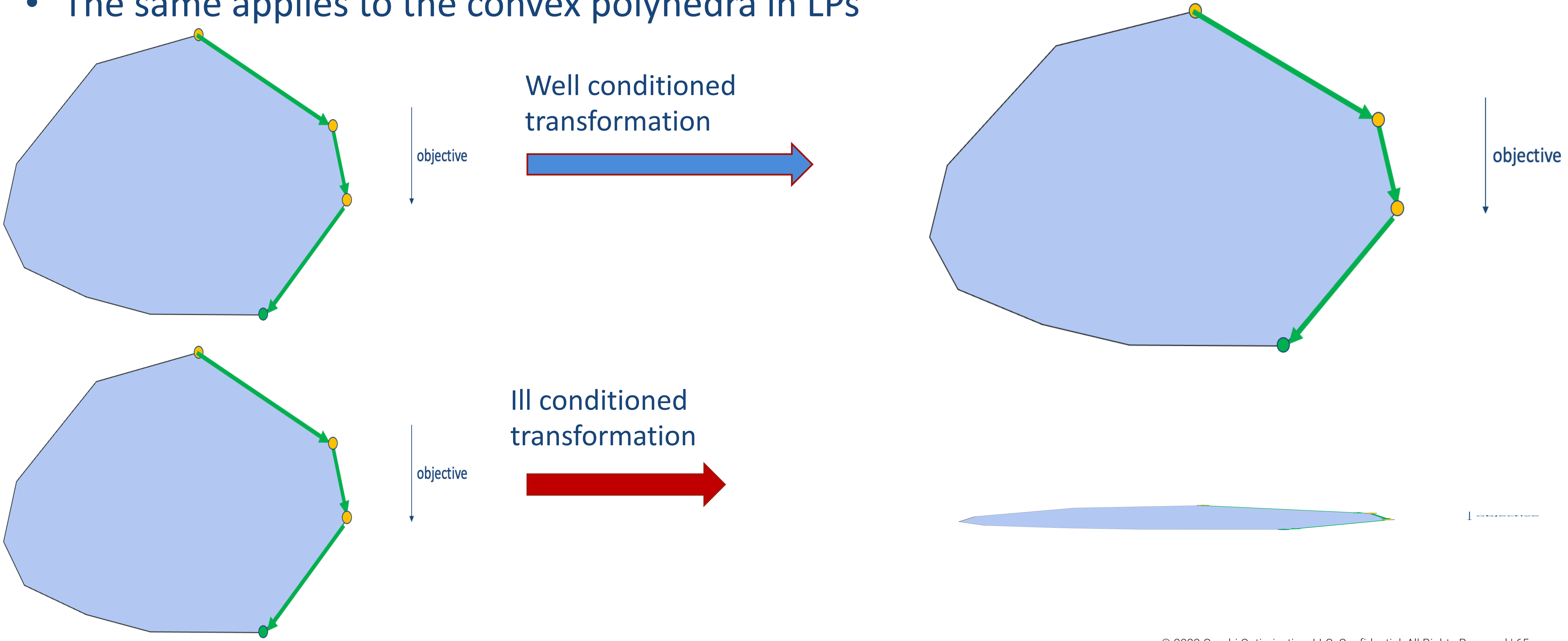
Well conditioned transformation

Ill conditioned transformation

Image by BrandonDanielArt on deviantart.com

What does a basis actually mean?

- The same applies to the convex polyhedra in LPs

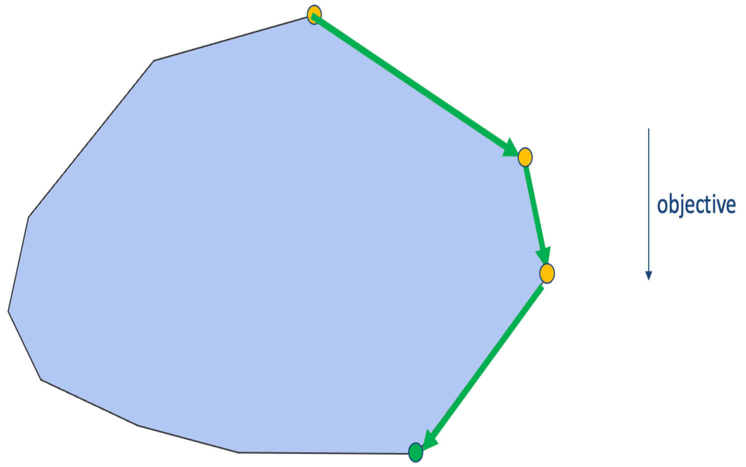


What does a basis actually mean?

- Consider each individual pivot in the simplex method
  - Rank one update matrix associated with a single pivot

Inverse

$$\begin{pmatrix} 1 & a & & & \\ & 1 & & & \\ & M & \ddots & \ddots & \\ & a & & 1 & \\ & a & & & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & -a & & & \\ & 1 & & & \\ & -M & \ddots & \ddots & \\ & -a & & 1 & \\ & -a & & & 1 \end{pmatrix}$$

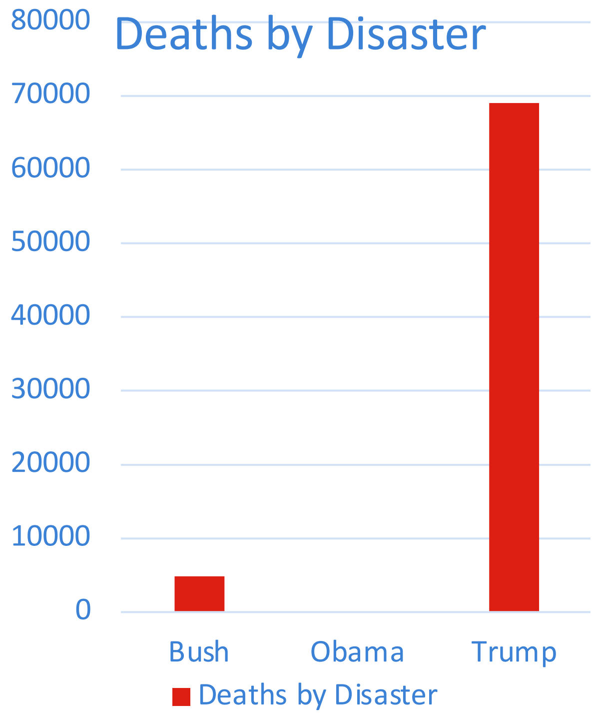


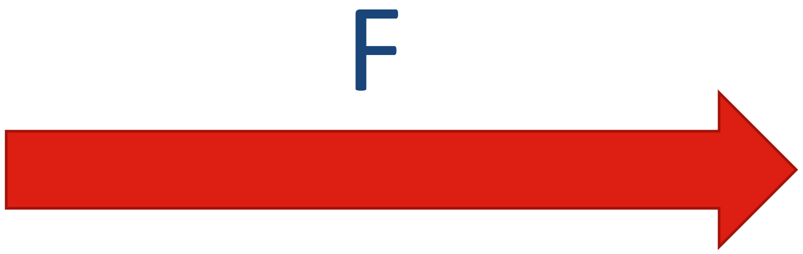
  - Condition number = M^2
    - Larger values of M imply more distorted change in the coordinate system resulting from a single pivot
    - Markowitz Tolerance, Harris Ratio Test and Numeric Focus all try to generate update matrices with smaller values of M

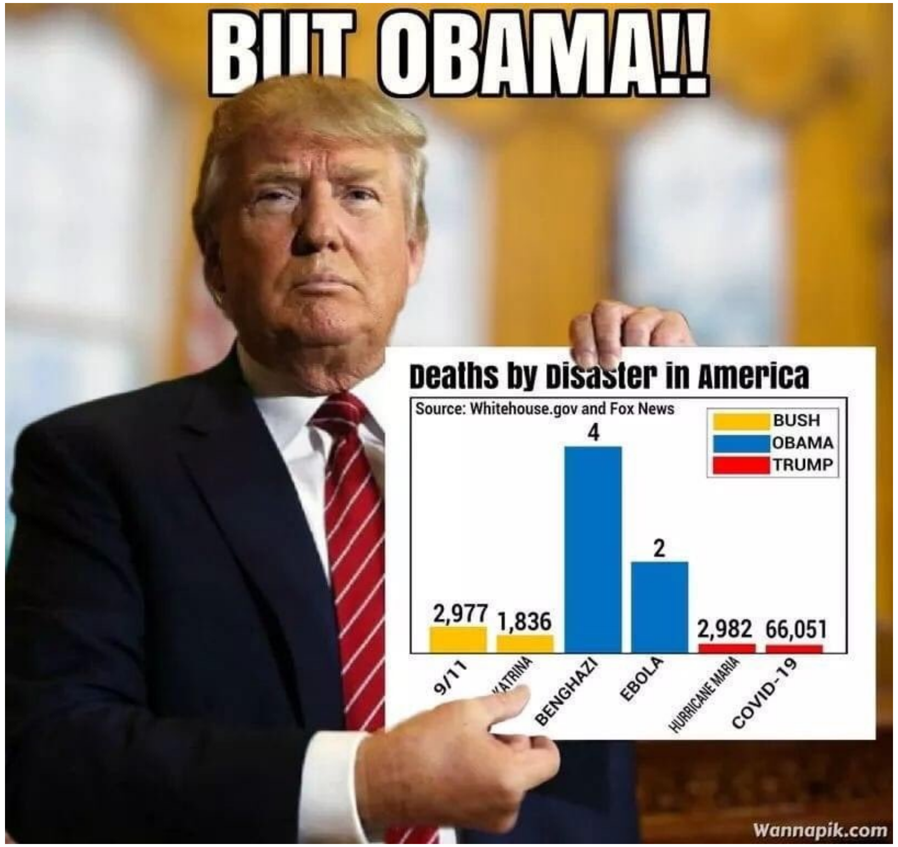
Ill conditioned matrix example

- Big ratio of largest to smallest diagonal in the basis factorization
- Deaths by disaster

  4813 Bush, 6 Obama, 69032 Trump







Other computations of an explanation of ill conditioning

- Column based explanation
  - Approximate this by applying FeasRelax to minimize the sum of infeasibilities of

$$B\,y = 0$$
$$e^T y = 1$$
$$y \; free$$

    - **Support of y provides a column-based certificate of the ill conditioning**
- Variants
  - FeasRelax can be instructed to minimize the sum (L1 norm) or sum of squares (L2 norm) of the infeasibilities.
    - FeasRelax can also minimize the number (L0 norm) of infeasibilities, but the Gastinel-Kahan Theorem requires a p-norm ≥ 1.

Other computations of an explanation of ill conditioning

- Add an objective function to try to reduce the number of elements in the support of the certificate of ill conditioning
  - $$\begin{aligned} \min \ & \| y \|_2^2 \\ & B\, y = 0 \\ & e^T y = 1 \\ & y \ free \end{aligned}$$
  - $$\begin{aligned} \min \ & y^+ + y^- \\ & B\,(y^+ - y^-) = 0 \\ & e^T (y^+ - y^-) = 1 \\ & y^+,\ y^- \geq 0 \end{aligned}$$

FeasRelax can be easily configured to optimize any objective on the set of solutions that minimize the infeasibilities

Other Approaches to Explaining Ill Conditioning

Other Possible Approaches

- Use the basis factorization to filter
  - Build down
    - Start with B = LDU, remove one row/column, discard if Kappa doesn’t decrease sufficiently.   Otherwise, restore and continue.
    - May be difficult to characterize “sufficient” decrease in Kappa 👎
    - Factorizations at the beginning involved ill conditioned basis matrices👎
  - Build up
    - Start with a single pivot element, systematically add one pivot at a time until ill conditioned.
      - Schur complement style factorization?
    - Once ill conditioned, apply build down approach to the (typically reduced size) square matrix.
      - Or use some other ill conditioning explainer on the smaller matrix.

Other Possible Approaches

- Look at minimum and maximum diagonal elements in B = LDU
  - Trace back the pivot history that led to the largest and smallest elements, filter out the other rows and columns since they weren't involved
- Eigenvalues
  - Condition number of basis is also defined as the ratio of maximum to minimum eigenvalues
  - Will decompositions based on eigenvalues or singular values give us any additional information lacking from the LDU factorization?
    - Principal Components Analysis
- Other decompositions
  - Gram Schmidt?

Examples: Mixed

- Open MIPLIB model neos-4230265-orari
  - No information on MIPLIB website regarding the model history
    - Nature of model not obvious based on quick inspection
  - Current best solutions on site:

	ID	Objective	Exact	Int. Viol	Cons. Viol	Obj. Viol	Submitter	Date	Description
2	2	81765.21	1e+100	8e-06	1e-07	0	Edward Rothberg	2020-09-14	Found with Gurobi
1	1	91310.20		0e+00	0e+00	0	Robert Ashford and Alkis Vazacopoulus	2019-12-18	Found using ODH|CPlex

ODH recently added support for Gurobi

  - Root node relaxation condition number: 3.69638104277258 8e+17
  - See if improving the condition number helps performance

Examples: Mixed

- Open MIPLIB model neos-4230265-orari
  - Run time for row and column based ill conditioning explainers below one minute
  - Row based explanation has 2497 rows, not obvious what to interpret
  - Column based explanation has only 40 columns, one in particular appears worthy of a closer look

```
(mult=1.9999872600811533e-08)C27728  R18172    25
(mult=1.9999872600811533e-08)C27728  R30572    -239
(mult=1.9999872600811533e-08)C27728  R35302    -239
(mult=1.9999872600811533e-08)C27728  R42404    50000000
(mult=1.9999872600811533e-08)C27728  R42405    -0.01
(mult=1.9999872600811533e-08)C27728  R44764    -50000000
(mult=1.9999872600811533e-08)C27728  R44765    0.01
(mult=1.9999872600811533e-08)C27728  R47124    1
(mult=1.9999872600811533e-08)C27728  R49502    -1
```

Large coefficient ratio in a single matrix column

Examples: Mixed

- Open MIPLIB model neos-4230265-orari

```
(mult=1.9999872600811533e-08)C27728  R42404    50000000
(mult=1.9999872600811533e-08)C27728  R42405    -0.01
(mult=1.9999872600811533e-08)C27728  R44764    -50000000
(mult=1.9999872600811533e-08)C27728  R44765    0.01
```

- Relevant constraints:

binary

```
R42404: – C9547 – 0.01 C27727 + 5e+07 C27728 >= -25
R44764: C9547 + 0.01 C27727 – 5e+07 C27728 >= -4.9999975e+07
C27778 = 0 → R44764 nonbinding
C27778 = 1 → R44764: C9547 + 0.01 C27727 >= 25
Eliminate the large rhs via
R44764*: C9547 + 0.01 C27727 >= 25 C27728
```

These two variables have lowerbounds of 0

Examples: Mixed

- Open MIPLIB model neos-4230265-orari
  - Examination of previous constraints suggests trickle flow is present in solutions with integrality violations of 1e-6
    - Obtained solution with objective 74299 after improving formulation as in previous slide, but examination of IIS for the fixed model confirms presence of trickle flow:

binary

```
 R38669: C0528 + C2923 - 5e+07 C28725 <= 0
Bounds
 C2923 >= 0.005
 -infinity <= C28725 <= 0
Generals
 C28725
```

Positive lower bound implies binary C28725 = 1

Can set C28725 = .005/5e+7 and satisfy default integrality tolerance of 1e-5

Examples: Mixed

- Takeaways for open MIPLIB model neos-4230265-orari
  - Column based ill conditioning explanation identified some numerical issues in the model
  - Improved formulation yielded a better MIPLIB incumbent both in terms of objective and solution quality:

```
Maximum violation:
  Bound       : 4.53573422e-09 (C10022)
  Constraint  : 1.99673137e-08 (R54649)
  Integrality : 5.11530058e-06 (C32980)
```

Int. Viol	Cons. Viol	Obj. Viol
8e-06	1e-07	0

  - But integrality violation indicates this solution has trickle flow
    - True meaning of model is subverted.
  - Should run with feasibility and integrality tolerances of 1e-8 and IntegralityFocus = 1
    - So far nothing as good as the ODH/CPLEX or ODH/Gurobi solution.

