Identification, Assessment and Correction of Ill-Conditioning and Numerical Instability in Linear and Integer Programs

Ed Klotz (klotz@us.ibm.com), Math Programming Specialist, IBM

Objective

- Enable more precise assessment of ill conditioning in linear and integer programs
  - Multiple metrics available to assess ill conditioning
- Discuss some techniques to treat the symptoms and causes of ill conditioning in LP and MIP models

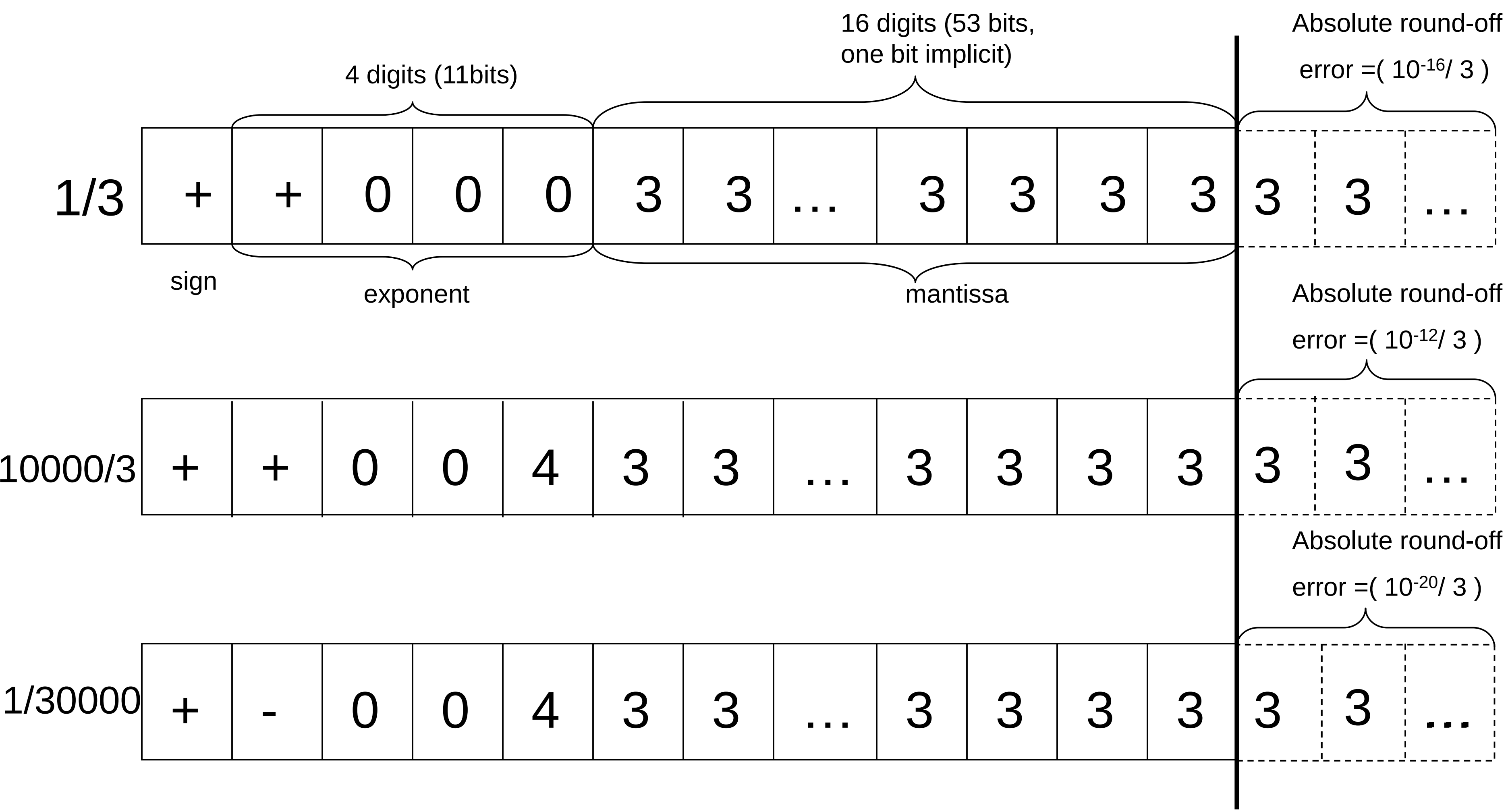
Outline

- Finite precision computing fundamentals
- Description of ill conditioning
- Assessment of ill conditioning
- Alternate metrics for ill conditioning
- Numerical stability of algorithms
- Identification and treatment of symptoms of ill conditioning
- Identification and treatment of sources of ill conditioning
- Examples that illustrate modeling pitfalls that can contribute to ill conditioning
  - Formulation alternatives
- Conclusions

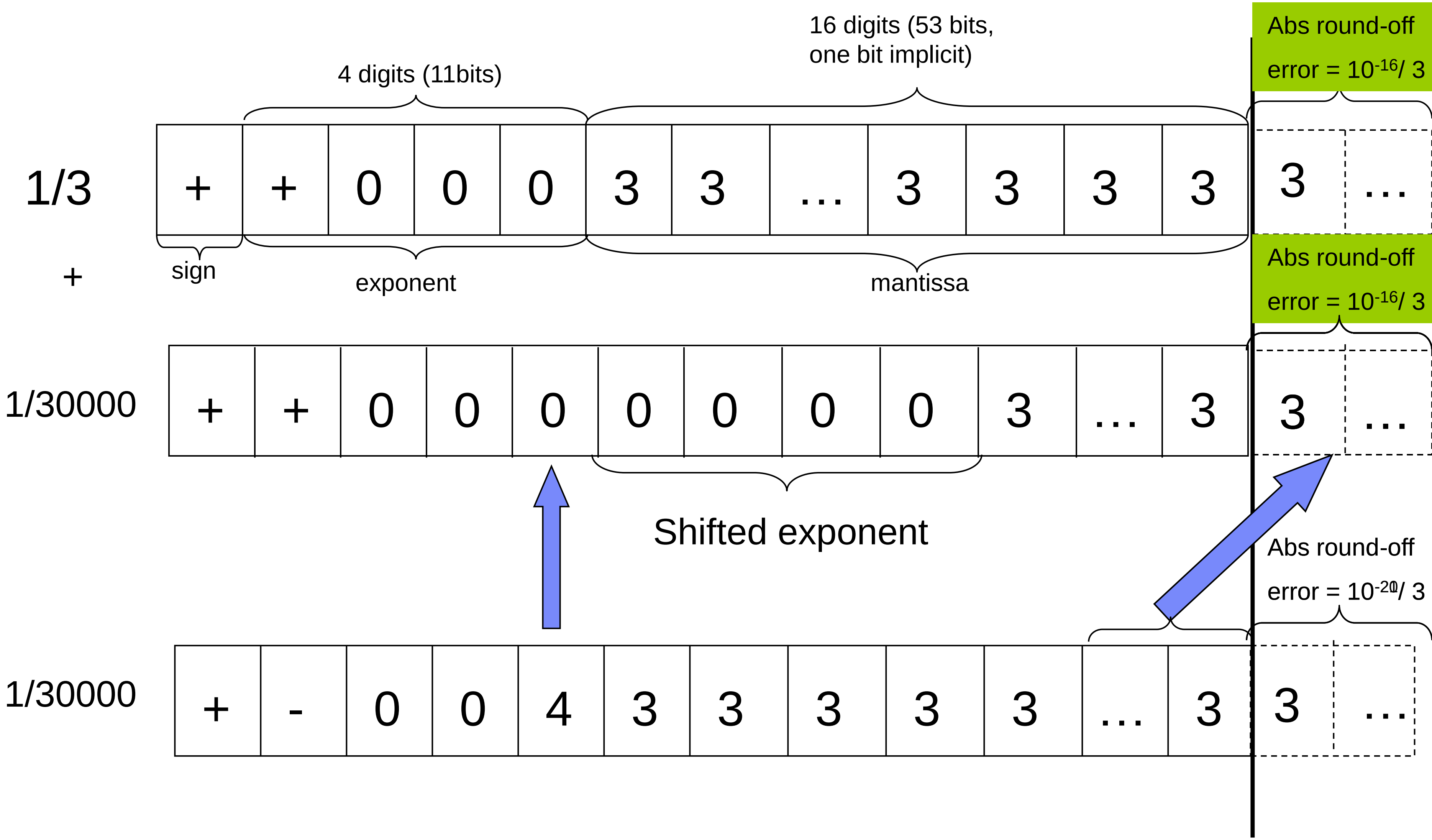
Finite Precision Computing Fundamentals

- 64 bit double precision is most commonly used in scientific applications
  - 32 bit single precision requires less memory, but is less accurate
    - Memory savings not significant for LP and MIP solvers anymore
  - 128 bit quad precision is more accurate, but requires more memory and computing time
- Many floating point numbers cannot be represented exactly
  - Base of floating point representation determines those that can
  - Base 2 typically used
    - Numbers that are sums of (positive and negative) powers of 2 can be represented exactly, within the limits of the minimum and maximum possible exponents
      - 8.0625 = 2^3 + 2^(-4) can be represented exactly
      - .333333…. cannot be represented exactly

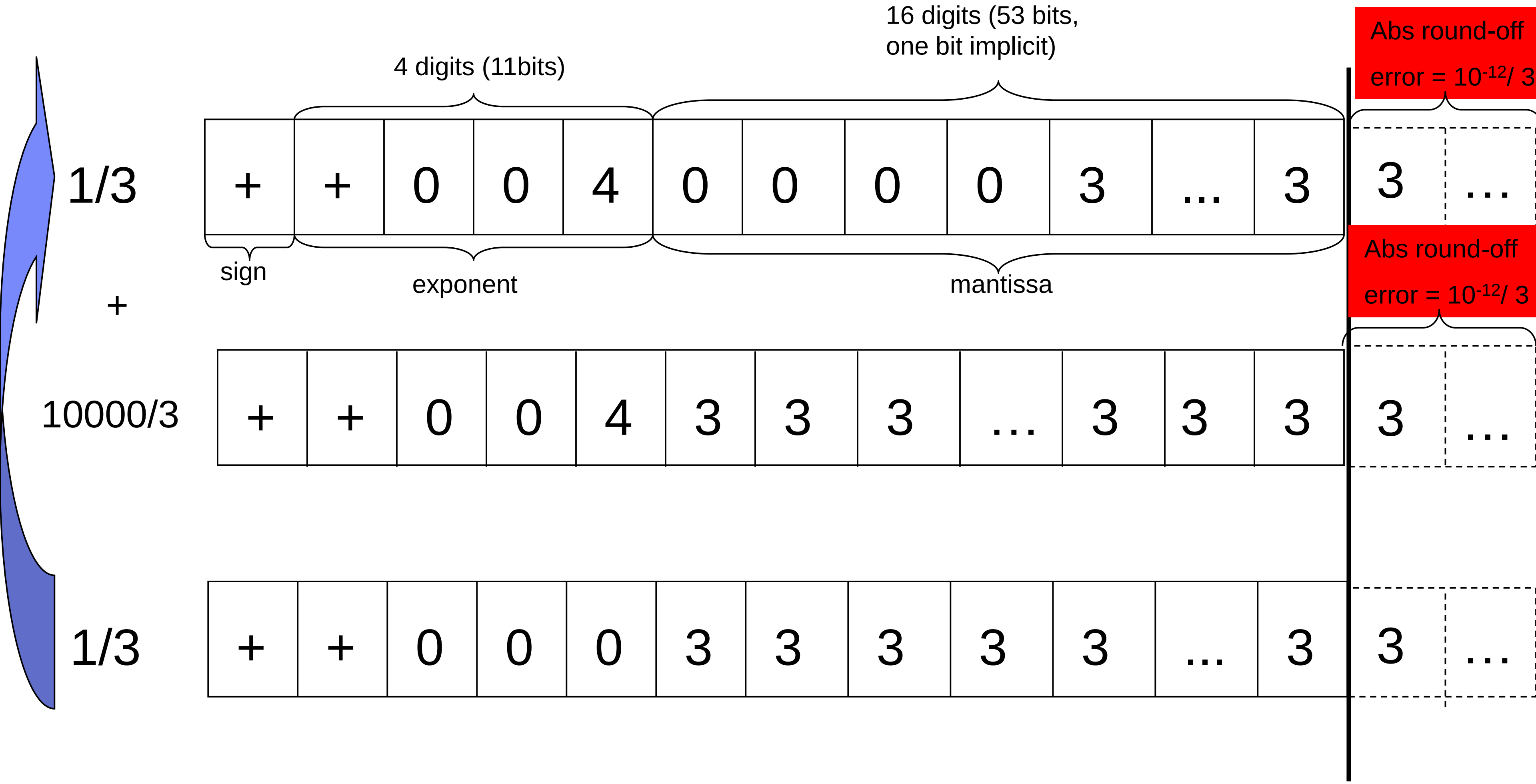
Example: 64 Bit IEEE Double Representation



IEEE 64 Bit Double Addition



IEEE 64 Bit Double Addition



IEEE 64 Bit Division

- Subtract exponents, divide mantissas
  - Errors in representation in mantissa determine magnitude of round-off error

- Don't divide big numbers by small numbers in data calculations

$$\text{For } a >> b, \text{ compare } a/b \text{ and } b/a \qquad (a = 3, b = 1/30000, \ \varepsilon = 10^{-8})$$

$$b/a \sim (b+\varepsilon)/a = b/a + \boxed{\varepsilon/a} \qquad (\text{error} \sim 10^{-8})$$

$$a/b \sim a/(b+\varepsilon) = a/b - \boxed{a\varepsilon / \left((b+\varepsilon)b\right)} \qquad (\text{error} \sim 10^{0})$$

$$(a = 3, b = 1/30000, \varepsilon = 10^{-16})$$

$$b/a \sim (b+\varepsilon)/a = b/a + \boxed{\varepsilon/a} \qquad (\text{error} \sim 10^{-16})$$

$$a/b \sim a/(b+\varepsilon) = a/b - \boxed{a\varepsilon / \left((b+\varepsilon)b\right)} \qquad (\text{error} \sim 10^{-8})$$

Implications of Finite Precision Representation

- Simply representing the model data can introduce round-off errors
  - Larger numbers have larger absolute round-off errors in their representations
- Arithmetic calculations can introduce additional round-off errors
- With 64 bit doubles, we have 16 accurate base 10 digits, or 53 accurate base 2 digits.
  - Larger numbers have more digits to the left of the decimal point.
- Arithmetic calculations on numbers of the same order of magnitude are more accurate than calculations on numbers of different orders of magnitude

Description

- Ill Conditioning
  - *Does the flap of a butterfly's wings in Brazil set off a tornado in Texas?*

Data to 3 decimal places (.506) → Meteorological Model →

Data to 6 decimal places (.506127) → Meteorological Model →

Problem definition

- Ill Conditioning
  - Small change in input leads to big change in output

$$\boxed{\begin{array}{l} \text{Given } x \in R^n, y \in R^m, y = f(x) \\ \text{For } y + \Delta y = f(x + \Delta x), \text{compute bound} \\ \kappa : \|\Delta \mathrm{y}\| \leq \kappa \|\Delta x\| \end{array}}$$

  - Can we quantitatively measure ill conditioning?
    - For many mathematical systems or models, quantitative measures have yet to be discovered. But, sometimes we can measure it.
    - Specifically, we can measure ill conditioning when solving square linear systems of equations

Condition Number of a Square Matrix (Turing, 1948; Rice, 1966)

- CPLEX solves square linear systems of form:
- exact solution is:
- How will a change to the input vector b affect the computed solution x?

$$Bx = b$$
$$x = B^{-1}b$$
$$\underbrace{x}+\delta x = \underbrace{B^{-1}(b}+\delta b)$$
$$\Rightarrow \delta x = B^{-1}\delta b$$

- Cauchy-Schwarz inequality:

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

- Cauchy-Schwarz for original system:

$$\|b\| \le \|B\| \cdot \|x\|$$

- Combine and rearrange:

$$\frac{\|\delta x\|}{\|x\|} \le \|B\| \cdot \|B^{-1}\| \cdot \frac{\|\delta b\|}{\|b\|}$$

Condition Number of a Square Matrix (ctd.)

- CPLEX solves square linear systems of form: $Bx = b$
- exact solution is: $x = B^{-1}b$
- How will a change <u>to the input matrix B</u> affect the computed solution x?

$$(B + \delta B)(x + \delta x) = b$$

$$\cancel{Bx} + \delta Bx + B\delta x + \delta B\delta x = \cancel{b}$$

$$\Rightarrow B\delta x = -\delta B(x + \delta x) \qquad \Rightarrow -\delta x = B^{-1}\delta B(x + \delta x)$$

- Cauchy-Schwarz inequality:

$$\|\delta x\| \le \|B^{-1}\| \cdot \|\delta b\| \cdot \|x + \delta x\|$$

- Rearrange:

$$\frac{\|\delta x\|}{\|x + \delta x\|} \le \|B^{-1}\| \cdot \|\delta B\|$$

- Multiply by $\|B\| / \|B\|$:

$$\frac{\|\delta x\|}{\|x + \delta x\|} \le \|B^{-1}\| \cdot \|B\| \cdot \frac{\|\delta B\|}{\|B\|}$$

Condition Number

- Condition number of B is defined as $\kappa(B) = \|B\| \cdot \|B^{-1}\|$

$$\|\delta x\| / \|x\| \le \kappa(B) \cdot \|\delta b\| / \|b\|$$

$$\frac{\|\delta x\|}{\|x + \delta x\|} \le \kappa(B) \cdot \|\delta B\| / \|B\|$$

- As condition number increases, potential change in solution relative to (normwise) change in data also increases
- Even if the modeler doesn‘t change the data, finite precision computers can introduce small changes
  - Machine precision for 64 bit double = 1e-16
  - Just moving from a Windows machine to an AIX machine can change precision enough to significantly influence results on an ill conditioned linear system

Assessment of Ill Conditioning

- What constitutes a large or small value?
  - Depends on machine, data and algorithm precision (ε), algorithm tolerances (t)
  - Ill conditioning can occur when round off error associated with machine precision is large enough to influence algorithm decisions

$$\underbrace{\|\delta x\|/\|x\|}_{\geq t\ ?} \leq \kappa(B)\cdot\underbrace{\|\delta b\|/\|b\|}_{\varepsilon}$$

  - Classify based on threshold defined by t / ε
    - Four distinct categories
  - Example: CPLEX has default algorithmic tolerances of 1e-6, runs double precision arithmetic on machines with precision of ~1e-16
    - t / ε = 1e-6/1e-16 = 1e+10 is a key threshold

Assessment of Ill Conditioning

- Condition number is a bound for the increase of the error:

$$\|\delta x\| / \|x\| \le \kappa(B) \cdot \|\delta b\| / \|b\|$$

- Basic epsilons:
  - Machine precision (double): 1e-16
  - Default feasibility and optimality tolerance: 1e-6

- Classification of condition numbers for LP bases:
  - Stable: $0 \le \kappa(B) < 1e+7$
  - Suspicious: $1e+7 \le \kappa(B) < 1e+10$
  - Unstable: $1e+10 \le \kappa(B) < 1e+14$
  - Ill-posed: $1e+14 \le \kappa(B)$

Assessment of Ill Conditioning

- What about MIP?
  - Optimality proof of MIP is based on pruning during tree search and thus not available with final solution
- How reliable is it?
  - Need to monitor **condition number of all optimal bases** used during Branch-and-Cut search
  - Performance impact
  - Can be mitigated by sampling

Assessment of Ill Conditioning

- MIP Kappa feature, available starting with CPLEX 12.2
  - Sample from the series of condition numbers
    - New parameter CPX_PARAM_MIPKAPPA with settings:
      - -1: off
      - 0: auto (defaults to off)
      - 1: sample
      - 2: use every optimal basis
  - Classification thresholds provide percentages of each category
  - Provide an assessment for users unfamiliar with ill conditioning
- If enabled, categorize condition numbers of optimal bases
    - Stable
    - Suspicious
    - Unstable
    - Ill-posed

Assessment of Ill Conditioning

- MIP Kappa sample output :

```
Branch-and-cut subproblem optimization:
Max condition number:                 3.5490e+16
Percentage of stable bases:            0.0%
Percentage of suspicious bases:       86.9%
Percentage of unstable bases:         13.0%
Percentage of ill-posed bases:         0.1%
Attention level:                       0.048893
CPLEX encountered numerical difficulties while solving this model.
```

- Attention level
  - =0 if only stable bases encountered
  - >0 if at least one basis encountered that is not stable
  - Max value is 1 (all bases ill-posed)
  - Not “linear”

Implications of Ill Conditioning

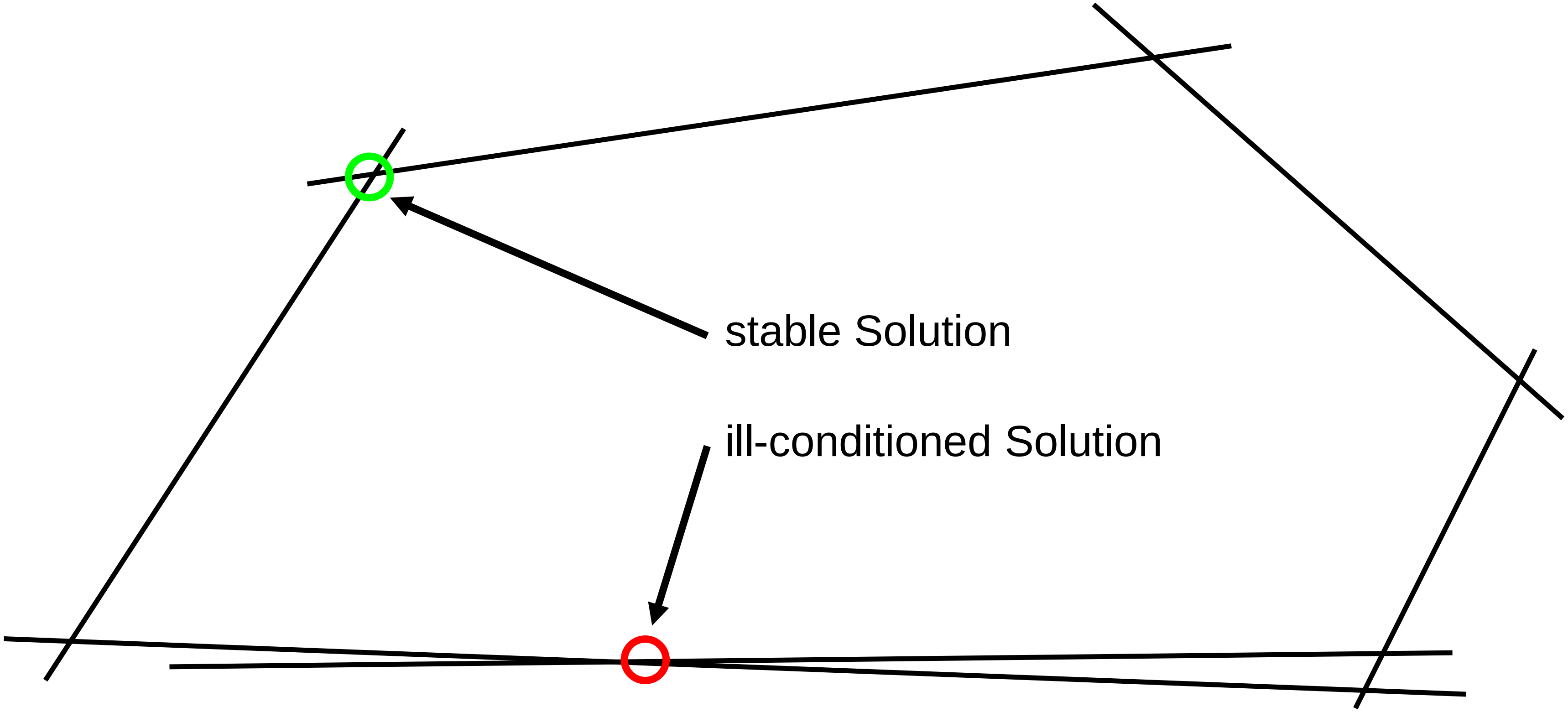
- Now that we can better assess the meaning of the basis condition numbers, what can we do about it?
  - Ill conditioning can occur under perfect arithmetic.
    - For some models, even perfect data, algorithm and machine precision may not address the problem.
      - Consider adjustments to existing formulation, or alternate formulations that provide the solution to the ill conditioned model.
  - But, in most cases, finite precision can perturb the exact system of equations we wish to solve, resulting in significant changes to the computed solution.
    - Calculate data, formulate model and configure algorithm to keep such perturbations as small as possible
  - Condition number provides a worst case bound on the effect
    - CPLEX provides good quality solutions on majority of models containing some basis condition numbers in [1e+10, 1e+14]

Implications of Ill Conditioning

- Sources of perturbations
  - Finite precision representation of exact data
  - Calculation of problem data in finite precision
  - Truncation of calculated data
    - Good idea if based on knowledge of the model and associated physical system (cleaning up the model data)
    - Bad idea if done arbitrarily without considering the implications for the model and associated physical system (garbage in, garbage out).
  - Errors in algorithmic calculations of data
    - Statistical methods to predict demand for production planning or asset returns for consideration in a financial portfolio
  - Errors in physical measurements of data values
  - Any other differences between the conceptual perfect precision calculation and the practical finite precision calculation
    - Example: addition and multiplication no longer associative and distributive under finite precision

Alternate interpretations of Ill Conditioned Basis

- Condition number of Simplex Solutions
  - Simplex solution is intersection point of n hyperplanes



Alternate interpretations of Ill Conditioned Basis

- Distance to singularity of a matrix is the reciprocal of its condition number (Gastinel, Kahan).

$$\mathrm{dist}_p(B) := \min_{\Delta B}\left\{\frac{\|\Delta B\|_p}{\|B\|_p} : B + \Delta B \text{ singular}\right\}$$

$$\mathrm{dist}_p(B) = 1/\mathrm{K}_p(B)$$

  - Implies that linear combinations of rows or columns of B that are close to 0 imply ill conditioning:

$$\text{if } B^T\lambda = v, \|v\| < \varepsilon, \quad \|\lambda\| >> \|\varepsilon\|,$$

$$B \text{ is close to singular, and hence ill conditioned}$$

  - λ provides a certificate of ill conditioning; its support identifies rows to examine

Implications for the Practitioner (Data Input)

- Can't avoid perturbations due to machine precision

$$\underbrace{\|\delta x\| / \|x\|}_{\geq t\,?} \leq \kappa(B) \cdot \underbrace{\|\delta b\| / \|b\|}_{\mathcal{E}}$$

  – But, increasing tolerances when condition number is high can prevent algorithmic decisions based on round off error associated with machine precision.

  – <u>More precise input values are better</u>

    - Always calculate and input model data in double precision
    - Machine precision for 32 bit floats ~1e-8
      - Condition numbers > 1e+2 could result in algorithmic decisions based on machine precision based round off error
      - If you really need to use single precision in the model data, increase the algorithms tolerances above the default of 1e-6

Implications for the Practitioner (Data Calculation)

- Minimize perturbations involving other factors

$$\underbrace{\|\delta x\|/\|x\|}_{\geq t\ ?} \leq \kappa(B)\cdot \underbrace{\|\delta b\|/\|b\|}_{\varepsilon}$$

  - Model data values
    - Don’t divide big numbers by small numbers in data calculations
      - Increases round off error
    - Make sure all procedures that calculate the data are implemented in a numerically stable manner
    - Less round off error if all data values of similar order of magnitude
      - Mix of large and small numbers results in more shifting of the exponents, loss of precision in the mantissa.
        - Use CPLEX’s aggressive scaling, numerical emphasis parameters if unavoidable

Implications for the Practitioner (Formulation)

- Avoid nearly linear dependent rows or columns

$$\text{if } B^T \lambda = v, \| v \| < \varepsilon, \quad \| \lambda \| >> \| \varepsilon \|,$$

$$B \text{ is close to singular, and hence ill conditioned}$$

  – Such linear combinations of rows and columns often arise from round off error in the data

Implications for the Practitioner (Formulation)

- Imprecise model data values and near singular matrices (example)
  - Avoid rounding if you can, or round as precisely as possible
  - Matrices can be ill conditioned despite small spread of coefficients

– Exact formulation:
Maximize x1 + x2
c1: 1/3 x1 + 2/3 x2 = 1
c2: x1 + 2 x2 = 3

$\text{if } B^T\lambda = v, \| v \| < \varepsilon, \quad \| \lambda \| >> \| \varepsilon \|,$

B is close to singular, and hence ill conditioned

– Imprecisely rounded, single [double] precision
Maximize x1 + x2
c1: .33333333 x1 + .66666667 x2 = 1 (results in near singular matrix)
[c1: .333333333333333 x1 + .666666666666667 x2 = 1] (better)
c2: x1 + 2x2 = 3

– Scale to integral value whenever possible:
Maximize x1 + x2
c1: x1 + 2 x2 = 3
c2: x1 + 2 x2 = 3 (best)

Numerical Stability of Algorithms

- Numerical instability and ill conditioning are not the same
  - Ill conditioning can occur under perfect precision; numerical instability is specific to finite precision
  - Informally, an algorithm is numerically unstable if it performs calculations that introduce unnecessarily large amounts of round-off error
  - Formally, numerical stability (or lack thereof) involves error analysis

$$\text{Given } x \in R^n, y \in R^m, y = f(x)$$

$$\text{Forward error analysis: } \Delta y = \left| fl(f(x)) - f(x) \right|$$

$$\text{Backward error analysis: } \Delta x : f(x + \Delta x) = fl(f(x))$$

- Forward: change in computed solution due to round-off errors
- Backward: change in model (under perfect precision) required to achieve finite precision result
- An algorithm is numerically stable when the bound on the backward error is small relative to the error in the input

Numerical Stability of Algorithms

- Sources of numerical instability in finite precision algorithms and calculations
  - Performing arithmetic operations on numbers of dramatically different orders of magnitude
    - Look for mathematically equivalent calculation on numbers of more similar magnitude
  - Algorithms that rely on ill conditioned subproblems
    - Example: Gomory cuts become almost parallel in cutting plane algorithm as it nears convergence
  - Ill-conditioned transformations of the problem*
    - Example: LU factorization calculated with numerically unstable pivot selections
  - Calculations involving large intermediate values compared to final solution values*
    - Small relative error for large intermediate values are much larger relative to final value

* source: Higham, Accuracy and Stability of Numerical Algorithms

Identification of symptoms of ill conditioning

- Tactics and tools for assessing presence of ill conditioning or excessive round-off error
  - Examine problem statistics of model before starting the optimization
    - Mixtures of large and small coefficients
    - Indications of nearly linearly dependent rows
      - Values with repeating decimal places
  - Examine node or iteration log during the optimization
    - Loss of feasibility for LP/QP solves
    - Large iteration counts for node relaxations
  - Examine solution quality after the optimization
    - Significant primal or dual solution residuals often indicate large basis condition numbers
  - Run the MIP Kappa feature for MIPs
  - (CPLEX 12.7 and later) Run CPLEX's Modeling Assistance tool by setting the datacheck parameter to 2

Identification of symptoms of ill conditioning: ns1687037 (http://plato.asu.edu/ftp/lptestset/)

- Problem statistics:

```
Variables              :   43749  [Nneg: 36001,  Box: 874,  Free: 6874]
Objective nonzeros     :   24000
Linear constraints     :   50622  [Greater: 38622,  Equal: 12000]
  Nonzeros             : 1406739
  RHS nonzeros         :   24000

Variables              : Min LB: 0.000000        Max UB: 3.000000
Objective nonzeros     : Min      : 1.000000      Max     : 100.0000
Linear constraints     :
  Nonzeros             : Min      : 1.987766e-08  Max     : 1364210.
  RHS nonzeros         : Min      : 0.0005000000 Max     : 5.030775e+07
```

Wide range of coefficients; smallest below default feasibility, optimality tolerances

Identification of symptoms of ill conditioning: ns1687037 (http://plato.asu.edu/ftp/lptestset/)

- Iteration log #1: Loss of feasibility after basis refactorization

```
Iteration: 674222   Dual objective     =          913.318204
Iteration: 674228   Dual objective     =          913.318204
…              <3 more refactorizations>
Iteration: 674256   Dual objective     =          913.318205
Iteration: 674258   Dual objective     =          913.318205
Removing perturbation.
Iteration: 674259   Scaled dual infeas =         12123.146176
Iteration: 674772   Scaled dual infeas =           595.276887
Elapsed time = 16959.32 sec. (6667412.82 ticks, 674876 iterations)
...
Elapsed time = 17138.76 sec. (6737439.02 ticks, 681930 iterations)
Iteration: 681949   Scaled dual infeas =             0.000002
...
Iteration: 682542   Scaled dual infeas =             0.000000
Iteration: 682624   Dual objective     =       -19559.930294
Iteration: 682896   Dual objective     =       -18109.597465
Elapsed time = 17160.88 sec. (6747443.75 ticks, 682941 iterations)
```

Should only refactor every 100+ iters

Massive loss of feasibility

Objective much worse

Identification, of symptoms of ill conditioning: ns1687037 (http://plato.asu.edu/ftp/lptestset/)

- Iteration log #2: Increase in Markowitz tolerance after frequent refactorizations of the basis

```
Iteration: 783543   Dual objective     =          3.635840
Iteration: 783548   Dual objective     =          3.635840
Iteration: 783550   Dual objective     =          3.635840
Iteration: 783553   Dual objective     =          3.635840
Iteration: 783556   Dual objective     =          3.635840
Removing shift (209).
Markowitz threshold set to 0.99999
Iteration: 783558   Dual objective     =          3.635695
```

Should only refactor every 100+ iters

CPLEX reacts to signs of trouble

Dual feasibility preserved

Identification, of symptoms of ill conditioning: ns1687037 (http://plato.asu.edu/ftp/lptestset/)

- Solution quality (available in all CPLEX APIs)

Max. unscaled (scaled) bound infeas. = 8.39528e-07 (8.39528e-07)

(Reduce feasibility tolerance, continue optimizing to decrease bound infeasibilities)

Max. unscaled (scaled) reduced-cost infeas. = 2.31959e-08 (2.31959e-08)

(Reduce optimality tolerance, continue optimizing to decrease reduced cost infeasibilities)

Max. unscaled (scaled) Ax-b resid. = 3.51461e-07 (1.16886e-12)

(If exceeds feasibility tolerance, CPLEX feasibility decisions based on round off error)

Max. unscaled (scaled) c-B'pi resid. = 1.18561e-13 (1.18561e-13)

(If exceeds optimality tolerance, CPLEX optimality decisions based on round off error)

Max. unscaled (scaled) |x| = 24139.1 (24139.1)

Max. unscaled (scaled) |slack| = 48278.2 (48278.2)

Max. unscaled (scaled) |pi| = 76.2637 (76.2637)

Max. unscaled (scaled) |red-cost| = 100 (100)

Condition number of scaled basis = 2.2e+12

(Use to assess sensitivity of solution to perturbations in the model data)

Treatment of symptoms of ill conditioning

- Distinguish the symptoms from the cause
- Treatment of symptoms often not as robust, but it may provide a quick resolution to a pressing problem
- CPLEX parameters to treat symptoms
  - Set the scale parameter to 1
    - Geometric mean based scaling works well on models with wide range of coefficients
  - Increase the Markowitz tolerance from its default of 0.01 to .90 or larger (max of .9999)
    - Tightens the pivot threshold in the row stability test of the LU factorization
    - Equivalently, tightens the bound on the sub diagonal elements of L from 1/.01 to 1/.9
  - Turn on the numerical emphasis parameter
    - Causes CPLEX to invoke internal logic to perform more accurate calculations (including quad precision for the LU factorization)

Treatment of symptoms of ill conditioning

- ns1687037
  - Problem stats indicated wide range of coefficients in matrix
    - Well suited for setting scale parameter to 1
    - Removed all problems (loss of feasibility, overly frequent basis refactorizations) seen in iteration logs
    - Results: Huge reductions in run times for dual simplex and barrier, modest reduction for primal simplex:

	Algorithm		
Settings	Primal	Dual	Barrier
Default	14214.6	21094.2	1258.23
Scaling=1	11164.64	907.5	83.52

- Solution quality was better as well

Treatment of causes of ill conditioning

- ns1687037
  - Problem stats indicated wide range of coefficients in matrix

    Linear constraints :
    Nonzeros : Min : 1.987766e-08 Max : 1364210.
    RHS nonzeros : Min : 0.0005000000 Max : 5.030775e+07
  - Modeling assistances output (datacheck = 2) confirms:

    CPLEX Warning 1045: Detected nonzero <= the maximum value of either CPX_PARAM_EPRHS or CPX_PARAM_EPOPT at constraint 'R0002627', variable 'C0025750'.
    CPLEX Warning 1045: Detected nonzero <= the maximum value of either CPX_PARAM_EPRHS or CPX_PARAM_EPOPT at constraint 'R0002635', variable 'C0025753'.
  - Are the small coefficients of 1e-8 meaningful or due to round-off error in the data calculations?
    - Changing them to 0 results in an LP that solves to optimality within 1 second, just with presolve
      - Suggests these coefficients have meaning, but may cause trouble for CPLEX's default feasibility or optimality tolerances of 1e-6
    - Modeller or data owner needs to assess if these coefficients are meaningful

Treatment of causes of ill conditioning

- ns1687037
  - Consider the constraints that contain the tiny coefficients
    - Fortunately, they appear repeatedly in small subsets of constraints
    - Reformulation of one subset will apply to other subsets

```
R0002624: 50150 C0024008 + 50150 C0024010 + 50150 C0024012 +
          50150 C0024014 + 50150 C0024016 + 113600 C0024020 +
          50150 C0024024 + 113600 C0024026 + 113600 C0024038 +
          ... +
          69070 C0025728 + 69070 C0025734 + 47585 C0025738 +
          50150 C0025742 + 50150 C0025744 + 69070 C0025748 +
          C0025749  = 50307748
R0002625: - C0025749 + C0025750 >= 0
R0002626: C0025749 + C0025750 >= 0
R0002627: 1.9877659e-8 C0025750 - C0025751  = 0
R0002628: C0000001 - C0025751 >= 0
R0002629: C0000002 - C0025751 >= -0.0005
R0002630: C0000003 - C0025751 >= -0.0008
R0002631: C0000004 - C0025751 >= -0.0009
```

These variables only appear in constraints on this slide

Treatment of causes of ill conditioning

- ns1687037

These vars appear in other constraints

R0002624: 50150 C0024008 + 50150 C0024010 + 50150 C0024012 +
50150 C0024014 + 50150 C0024016 + 113600 C0024020 +
50150 C0024024 + 113600 C0024026 + 113600 C0024038 +
... +
69070 C0025728 + 69070 C0025734 + 47585 C0025738 +
50150 C0025742 + 50150 C0025744 + 69070 C0025748 +
C0025749 = 50307748 // C0025749 free; all others ≥ 0
R0002625: - C0025749 + C0025750 >= 0
R0002626: C0025749 + C0025750 >= 0
R0002627: 1.9877659e-8 C0025750 - C0025751 = 0
R0002628: C0000001 - C0025751 >= 0
R0002629: C0000002 - C0025751 >= -0.0005
R0002630: C0000003 - C0025751 >= -0.0008
R0002631: C0000004 - C0025751 >= -0.0009

C25750 = | C0025749|

Scale abs. value of violation for R0002624

Penalty variables; appear here and in objective

Treatment of causes of ill conditioning

- ns1687037
  - Objective contains only variables like C0000001,…,C0000004
    - Piecewise linear, higher cost for larger absolute violation of R0002624
    - Move the scaling of absolute violation in the constraints to the objective
      - Dramatically improves the coefficient spread in the constraint matrix, LU factors
    - Better: use unscaled violation. Objective value will be larger, but, if needed, recapture actual value after the optimization

R0002627: 1.9877659 C0025750 – 1e+8 C0025751 = 0
R0002628: 1e+8 C0000001 – 1e+8 C0025751 >= 0
R0002629: 1e+8 C0000002 – 1e+8 C0025751 >= -50000
R0002630: 1e+8 C0000003 – 1e+8 C0025751 >= -80000
R0002631: 1e+8 C0000004 – 1e+8 C0025751 >= -90000

(x ' = (1e+8) x)

R0002627: 1.9877659 C0025750 – C0025751' = 0
R0002628: C0000001' – C0025751' >= 0
R0002629: C0000002' – C0025751' >= -50000
R0002630: C0000003' – C0025751' >= -80000
R0002631: C0000004' – C0025751' >= -90000

Treatment of causes of ill conditioning

- ns1687037
  - Reformulation:

```
R0002624: 50150 C0024008 + 50150 C0024010 + 50150 C0024012 +
          50150 C0024014 + 50150 C0024016 + 113600 C0024020 +
          50150 C0024024 + 113600 C0024026 + 113600 C0024038
          ... +
          69070 C0025728 + 69070 C0025734 + 47585 C0025738 +
          50150 C0025742 + 50150 C0025744 + 69070 C0025748 +
          C0025749  = 50307748
R0002625: - C0025749 + C0025750 >= 0
R0002626: C0025749 + C0025750 >= 0
R0002627: 1.9877659 C0025750 - C0025751  = 0
R0002628: C0000001 - C0025751 >= 0
R0002629: C0000002 - C0025751 >= -50000
R0002630: C0000003 - C0025751 >= -80000
R0002631: C0000004 - C0025751 >= -90000
```

Treatment of causes of ill conditioning

- ns1687037
  - Run times for original formulation:

	Algorithm		
Settings	Primal	Dual	Barrier
Default	14214.6	21094.2	1258.23
Scaling=1	11164.64	907.5	83.52

  - Run times for modified formulation:

	Algorithm		
Settings	Primal	Dual	Barrier
Default	2310.9	2926.5	41.4
Scaling=1	6890.8	1054.7	68.2

Common sources of ill conditioning

- Ill conditioning can be caused by large or small subsets of constraints and variables in the model
- Such subsets can be difficult to isolate
- Build up a list of common sources, use that before more model specific analysis

Common sources of ill conditioning

- Mixture of large and small coefficients in the model
  - Does not guarantee large basis condition numbers
  - Additional round-off in floating point representation, arithmetic calculations enables modest condition numbers to magnify error in computed solutions above optimizer tolerances
- Imprecise data resulting in near singular matrices
  - Near singular matrices have large condition numbers
- Long sequences of transfer constraints
  - Mixture of large and small coefficients is implicit rather than explicit
- This is not a comprehensive list
  - Add items based on your own modelling experiences

Common sources

$$\mathrm{dist_p}(B) := \min_{\Delta B}\left\{\frac{\|\Delta B\|_p}{\|B\|_p} : B + \Delta B \text{ singular}\right\}$$

$$\mathrm{dist_p}(B) = 1/\mathrm{K_p}(B)$$

- Imprecise model data values
  - Avoid rounding if you can, or round as precisely as possible
  - Matrices can be ill conditioned despite small spread of coefficients

– Exact formulation:
Maximize x1 + x2
c1: 1/3 x1 + 2/3 x2 = 1
c2: x1 + 2 x2 = 3

– Imprecisely rounded, single [double] precision
Maximize x1 + x2
c1: .33333333 x1 + .66666667 x2 = 1 (results in near singular matrix)
[c1: .3333333333333333 x1 + .6666666666666667 x2 = 1] (better)
c2: x1 + 2x2 = 3

– Scale to integral value whenever possible:
Maximize x1 + x2
c1: x1 + 2 x2 = 3 (best)
c2: x1 + 2 x2 = 3

Common sources of ill conditioning

- Long sequences of transfer constraints

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

- All coefficients have same order of magnitude
- All coefficients can be represented exactly as IEEE doubles
- How bad can it be?

Common sources of ill conditioning

- Long sequences of transfer constraints (ctd)
  - If any one variable > 0,
    all the others are basic as well
  - $K=3*2^n$
  - Bound from condition number is
    fairly tight

$$\tilde{B} = \begin{pmatrix} 1 & -2 & & & & & \\ & 1 & -2 & & & & \\ & & 1 & -2 & & & \\ & & & 1 & & & \\ & & & & \ddots & & \\ & & & & & 1 & -2 \\ & & & & & & 1 \end{pmatrix}$$

$$\tilde{B}^{-1} = \begin{pmatrix} 1 & 2 & 4 & 8 & & 2^{n-1} \\ & 1 & 2 & 4 & & 2^{n-2} \\ & & 1 & 2 & & 2^{n-3} \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & 1 & 2 \\ & & & & & 1 \end{pmatrix}$$

Common sources of ill conditioning

- Long sequences of transfer constraints (ctd)
  - Substitute out variables:

$$
\begin{aligned}
&x_1 = 2x_2 && \\
&x_2 = 2x_3 && (x_1 = 4x_3) \\
&x_3 = 2x_4 && (x_1 = 8x_4) \\
&\quad\vdots && \\
&x_{n-1} = 2x_n && (x_1 = 2^{n-1}x_n) \\
&x_n = 1 && \\
&x_j \geq 0 \quad \text{for } j = 1, \ldots, n &&
\end{aligned}
$$

  - Small change in x_n propagates into large change in x_1

Diagnostics for ill conditioning and numerical instability

- Consider the list of common sources first
- Look at solution values
  - Extremely large primal or dual values can identify small subsets of constraints and variables involved in the ill-conditioning
- Then look at the basis and its inverse for large values
  - C API programs available among IBM Technotes*
- For MIPs, consider MIP Kappa feature
  - C API program available to export node LPs with conditions number above a user supplied threshold available as well*
    - Look at solution values, basis values or inverse values after locating a node LP with ill conditioned optimal basis
- Run the modeling assistance tool

*http://www-01.ibm.com/support/docview.wss?uid=swg21662382

Examples from publicly available test sets

- ns1687037 (previously discussed)
  - Wide range of coefficients
  - Setting scaling to 1 helped
  - Recognize that smallest coefficients involved penalties on constraint violations that could be moved into the objective function, improving numerics of basis factorization
  - Reformulating the model to improve the numerics yielded additional improvements, addressed the underlying cause of the problem

Examples from publicly available test sets

- cdma (unsolved MIP from unstable test set of MIPLIB 2010)

```
        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                      -1.97987e+14  -6.51749e+17    71155     ---
      0     0  -6.33335e+16   704  -1.97987e+14  -6.33335e+16    71155     ---
      0     0  -6.31118e+16   702  -1.97987e+14    Cuts: 1870   185865     ---
      0     0  -6.26076e+16   631  -1.97987e+14    Cuts: 1870   292616     ---
...

      0     0 -5.87363e+16  1206  -3.21168e+15    Cuts: 1604  4545331    ---
*     0+    0                     -7.75173e+15  -5.87363e+16  4654552  657.72%
...

*     0+    0                     -1.16804e+16  -5.81032e+16  5626309  397.44%
      0     0  -5.80853e+16  1585  -1.16804e+16     Cuts: 566  5632163  397.29%
Heuristic still looking.
      0     2  -5.80853e+16  1583  -1.16804e+16  -5.80853e+16  5633601  397.29%
Elapsed time = 52951.48 sec. (11682283.45 ticks, tree = 0.01 MB, solutions = 17)
      1     3  -5.80295e+16  1444  -1.16804e+16  -5.80853e+16  5643488  397.29%
...

  12862 10763  -4.10127e+16  1032  -1.46845e+16  -4.29552e+16 29639775  192.52%
Elapsed time = 71901.33 sec. (18469780.15 ticks, tree = 33.05 MB, solutions = 24)
  12866 10767  -3.86482e+16   979  -1.46845e+16  -4.29552e+16 29661467  192.52%
```

Examples from publicly available test sets

- cdma (unsolved MIP from unstable test set of MIPLIB 2010)
  - Problem statistics:

Variables	:	7891 [Fix: 1, Box: 3655, Binary: 4235]
Objective nonzeros	:	2383
Linear constraints	:	9095 [Less: 8390, Greater: 645, Equal: 60]
Nonzeros	:	168227
RHS nonzeros	:	4145

Variables	: Min LB: 0.000000	Max UB: 1.000000e+07
Objective nonzeros	: Min : 1.000000	Max : 1.724400e+11
Linear constraints	:	
Nonzeros	: Min : 1.000000	Max : 5.000000e+07
RHS nonzeros	: Min : 1.000000	Max : 3000000.

Examples from publicly available test sets

- cdma (unsolved MIP from unstable test set of MIPLIB 2010)
  - Typical node LP* iteration log

Wasted iterations slow node throughput

Iteration log ...
Iteration: 1 Scaled dual infeas = 0.000130
Iteration: 8 Scaled dual infeas = 0.000069
Iteration: 12 Dual objective = -6614900586660791.000000
Iteration: 23 Scaled dual infeas = 0.000107
Iteration: 32 Dual objective = -6614900586660791.000000
Iteration: 46 Dual infeasibility = 0.000038
Iteration: 52 Dual objective = -6614900586660791.000000
Iteration: 58 Dual infeasibility = 0.000038
Iteration: 64 Dual objective = -6614900586660791.000000
Maximum unscaled reduced-cost infeasibility = 7.62939e-06.
Maximum scaled reduced-cost infeasibility = 7.62939e-06.
Dual simplex - Optimal: Objective = -6.6149005867e+15
...

* Node LP Program at http://www-01.ibm.com/support/docview.wss?uid=swg21400065

Examples from publicly available test sets

- cdma (unsolved MIP from unstable test set of MIPLIB 2010)
  - Solution quality for same node LP

Max. unscaled (scaled) bound infeas. = 1.81311e-07 (1.81311e-07)
Max. unscaled (scaled) reduced-cost infeas. = 7.62939e-06 (7.62939e-06)
Max. unscaled (scaled) Ax - b resid. = 9.99989e-10 (6.10345e-14)
Max. unscaled (scaled) c - B'pi resid. = 7.3125 (7.3125)
Max. unscaled (scaled) |x| = 5596 (55296)
Max. unscaled (scaled) |slack| = 6.12827e+06 (10.6908)
Max. unscaled (scaled) |pi| = 8.67329e+16 (4.02442e+17)
Max. unscaled (scaled) |red-cost| = 4.31401e+17 (4.31401e+17)
Condition number of scaled basis = 5.2e+08

Reasonable optimal basis condition number

Incoming variable choice based on round-off error

Examples from publicly available test sets

- cdma (unsolved MIP from unstable test set of MIPLIB 2010)
  - Large objective coefficients, not large basis condition numbers, cause slow node throughput
    - Model is numerically unstable, not ill conditioned
    - 16 base 10 digits of accuracy for IEEE doubles, objective coefficients on the order of 1e+11
      - Round-off error of 1e-5 just to represent
      - Modest basis condition numbers of 1e+8 can magnify to 1e+8*1e-5 = 1e+3
      - Default optimality tolerance: 1e-6
    - Simplex method pivots heavily influenced by round-off error
  - What can we do?
    - Take a closer look at the objective function

Examples from publicly available test sets

- cdma (unsolved MIP from unstable test set of MIPLIB 2010)
  - Histogram of objective coefficients*

OBJECTIVE Range	Count
[10^0,10^1]:	31
[10^9,10^10]:	1236
[10^11,10^12]:	1116

All binaries; relative contribution to objective is below CPLEX's default relative MIP gap

Current objective poorly scaled; would be well scaled if we deleted the 31 small objective coeffs.

  - Large objective coefficients problematic for dual feasibility of node LP, dual residuals in solution quality

*Using program from https://www-304.ibm.com/support/docview.wss?uid=swg21400100.

Examples from publicly available test sets

- cdma (unsolved MIP from unstable test set of MIPLIB 2010)
  - 31 binaries with relatively small objective coefficients have negligible impact on objective coefficients
    - Especially when current solutions have relative MIP gaps of over 150%
  - Remove them from the objective
    - Remaining objective coefficients are all on the order of [1e+9,1e+12]
    - Rescale by 1e+9
  - Much better results with adjusted model
    - Much faster node throughput
    - Much better intermediate results regarding MIP gap
    - Moderately better final MIP gap after ~20 hours
  - More to be done
    - MIP gap remains challenging
    - But at least now node throughput sufficiently fast to consider MIP parameter tuning, other changes to formulation

Examples from publicly available test sets

- cdma
- Implications
  - Mixture of large and small coefficients can be problematic
    - Consider solving sequence of problems with a hierarchical objective rather than solving a problem with a single, blended objective
  - Examined node log to discover slow node throughput was a major performance bottleneck
  - Examined node LP iteration log, solution quality, problem statistics to identify large dual residuals as the primary source of slow node LP solve times
  - Adjusted formulation to obtain a well scaled objective

Examples from publicly available test sets

- de063155 (LP from http://www.sztaki.hu/meszaros/public_ftp/lptestset/problematic/
  - CPLEX solves it in less than 0.1 seconds
  - Iterations logs indicate no sign of trouble
  - Problem statistics and solution quality raise questions regarding the solution and the associated physical system
  - Is the solution acceptable?
    - Depends
    - Examine the problem statistics and solution quality to find out.

Examples from publicly available test sets

- de063155 (LP from http://www.sztaki.hu/meszaros/public_ftp/lptestset/problematic/
  - Problem stats

Variables : 1488 [Nneg: 756, Fix: 205, Box: 215, Free: 228, Other: 84]
Objective nonzeros : 852
Linear constraints : 852 [Less: 360, Equal: 492]
Nonzeros : 4553
RHS nonzeros : 777

Variables : Min LB: -10000.00 Max UB: 30.90000
Objective nonzeros : Min : 1.279580e-05 Max : 1000.000
Linear constraints :
Nonzeros : Min : 2.106480e-07 Max : 8.354500e+11
RHS nonzeros : Min : 0.0002187500 Max : 4.227560e+17

Examples from publicly available test sets

- de063155 (LP from http://www.sztaki.hu/meszaros/public_ftp/lptestset/problematic/
  - Solution quality:

```
There are no bound infeasibilities.
There are no reduced-cost infeasibilities.
Max. unscaled (scaled) Ax-b resid.   = 747.949 (5.12641e-08)
Max. unscaled (scaled) c-B'pi resid. = 7.74852e-10 (8.51025e-06)
Max. unscaled (scaled) |x|           = 3.10148e+13 (3.76112e+07)
Max. unscaled (scaled) |slack|       = 3.75814e+07 (3.75814e+07)
Max. unscaled (scaled) |pi|          = 62061.1 (5.106e+09)
Max. unscaled (scaled) |red-cost|    = 6.78639e+09 (8.5923e+09)
Condition number of scaled basis     = 1.7e+08
```

  - Using aggressive scaling or turning on numerical emphasis does not improve the solution quality.

Examples from publicly available test sets

- de063155
  - Is the solution quality a problem?

```
Max. unscaled (scaled) Ax-b resid.   = 747.949 (5.12641e-08)
Max. unscaled (scaled) c-B'pi resid. = 7.74852e-10 (8.51025e-06)
Max. unscaled (scaled) |x|           = 3.10148e+13 (3.76112e+07)
Max. unscaled (scaled) |slack|       = 3.75814e+07 (3.75814e+07)
Max. unscaled (scaled) |pi|          = 62061.1 (5.106e+09)
Max. unscaled (scaled) |red-cost|    = 6.78639e+09 (8.5923e+09)
Condition number of scaled basis     = 1.7e+08
```

  - Unscaled primal residuals are large in an absolute sense, but not relative to primal solution values
  - Solution quality does not hinder performance
  - Practitioner must assess acceptability in context of the physical system being modelled
    - Look at the constraints with the large absolute residuals
    - Need more computing precision if not acceptable
    - Or need to reformulate model in order to eliminate large matrix and right hand side coefficients

Summary of examples from publicly available test sets

- ns1687037 (LP; previously discussed)
  - Wide range of coefficients
  - Setting scaling to 1 helped
  - Reformulating the model to improve the numerics yielded additional improvements, addressed the underlying cause of the problem
    - Moving the scaling issue from the matrix to the objective removed the numerical problems from the basis matrix
- cdma (MIP)
  - Basis condition numbers OK
  - Wide range of objective coefficients were the real problem
  - Separate large objective coefficients, rescale
  - Faster node throughput yields significantly better solutions faster, but solving MIP to optimality remains challenging

Summary of publicly available examples (ctd).

- de063155 (LP)
  - No performance problem; solves within a second
  - Problem statistics, solution quality are cause for concern
  - Large data values, significant absolute residuals that are relatively small
    - Need to assess whether residuals are acceptable in the context of the associated system being modelled

Key takeaways

- Finite precision representation and operations can easily introduce round-off error
  - Large basis condition numbers can magnify
- Monitor/assess conditioning of model with available tools
  - Make sure algorithms don't make decisions based on round-off error
  - Solution quality for LPs
  - MIP Kappa for MIPs
  - Node and iterations logs for signs of trouble
  - Modeling assistance tool
- Be careful about mixing large and small coefficients
- Compute data as accurately as possible

References/Further Reading

- More detailed discussion in INFORMS TutORials in Operations Research 2014
- Higham, Accuracy and Stability of Numeric Algorithms
- Duff, Erisman and Reid, Direct Methods for Sparse Matrices
- Gill, Murray and Wright, Practical Optimization
- Golub and Van Loan, Matrix Computations
- Floating point arithmetic: http://pages.cs.wisc.edu/~smoler/x86text/lect.notes/arith.flpt.html
- MIP Performance tuning and formulation strengthening: α
  - Klotz, Newman. Practical Guidelines for Solving Difficult Mixed Integer Programs

    http://www.sciencedirect.com/science/article/pii/S1876735413000020
- LP performance issues
  - Klotz, Newman. Practical Guidelines for Solving Difficult Linear Programs http://www.sciencedirect.com/science/article/pii/S1876735412000189
- Converting repeating decimals into rational fractions:

  http://en.wikipedia.org/wiki/Repeating_decimal#Converting_repeating_decimals_to_fractions

Other presentations of interest

- Today, 4-6:30 PM, Room 124A IBM Workshop, Latest news about IBM Decision Optimization
- Nov. 4, SB34, 11:45-12:30 PM. Solving Multiobjective problems with CPLEX, Ed Klotz
- Nov. 7, WC05, 2:30-2:50 PM. CPLEX Progress in 2018

Backup Material

- Backup Material

Problem Definition

- Ill Conditioning
  - Motivated by work of meteorologist & mathematician Edward Lorenz
  - Lorenz focused on small changes in initial conditions, resulting trajectories in nonlinear meteorological models
    - Lorenz subsequently became a pioneer in the field of Chaos Theory
  - Ill conditioning extends beyond the nonlinear meteorological models on which Lorenz worked
  - More generally, a mathematical model or system is ill conditioned when a small change in the input can result in a large change to the computed solution

Alternate interpretations of Ill Conditioned Basis

- Skeel's condition number: $\| \, |B^{-1}| \cdot |B| \, \|$
  - Invariant under row scaling
    - Not invariant under column scaling
  - If significantly smaller than regular condition number, some rows of the matrix have larger norms than others

Alternate interpretations of Ill Conditioned Basis

- Skeel's condition number:
  - Example (http://www.hsl.rl.ac.uk/specs/mc75.pdf):

c1: 3300 x1 + 1e-11 x2 = 1

c2: x1 + 3300 x2 + 1e-11 x3 = 1

c3: x2 + 3300 x3 + 1e-11 x4 = 1

c4: 10000 x2 + 10000 x3 + 330000000 x4 = 1 // much larger row norm

xj free, j=1,...,4

  - Skeel condition number: 1.0061
  - CPLEX exact condition number (no scaling): 100036
  - CPLEX exact condition number (default scaling): 10.0091

Alternate interpretations of Ill Conditioned Basis

- Skeel's condition number: $\| |B^{-1}| \cdot |B| \|$
  - Why does this metric measure sensitivity to perturbations?
    - We saw how $\kappa(B) = \|B\| \cdot \|B^{-1}\|$ measured potential magnification of error in the solution relative to perturbation in the input
    - What is the underlying theoretical justification for $\| |B^{-1}| \cdot |B| \|$?

Alternate interpretations of Ill Conditioned Basis

- What is the underlying theoretical justification for $\| |B^{-1}| \cdot |B| \|$?
  - Use absolute values on individual components instead of norms during derivation
  - Use componentwise perturbation instead of norms: $|\delta B| \le |\varepsilon B| \quad (\varepsilon > 0)$

(original system) (perturbed system)

$$Bx = b \qquad (B + \delta B)(x + \delta x) = b$$

(Combine and rearrange) $\Rightarrow -\delta x = B^{-1}\delta B(x + \delta x)$

(Componentwise abs value) $\Rightarrow |\delta x| = \left|B^{-1}\overbrace{\delta B}^{\le |\varepsilon B|}(x + \delta x)\right|$

$$\Rightarrow |\delta x| = \left|B^{-1}\delta B(x + \delta x)\right| \le \left|B^{-1}\right| \cdot |B| \cdot |(x + \delta x)|\varepsilon$$

$$\Rightarrow |\delta x| / |(x + \delta x)| \le \left|B^{-1}\right| \cdot |B| \cdot \varepsilon$$

Examples

- Consider alternate formulations to improve numerics
  - Fixed costs on continuous variables using big Ms:

$$\text{Minimize } c^T x + f^T z \qquad (c, f \geq 0)$$
$$\text{subject to } Ax = b$$
$$x_i - Mz_i \leq 0 \qquad (\text{only constraint with } z_i)$$
$$x_i \geq 0,\ 0 \leq z_i \leq 1$$
$$z_i \text{ integer}$$

(Mixture of large and small numbers)

- LP relaxation solution

$$x_i \leq Mz_i \Rightarrow x_i / M \leq z_i \Rightarrow z_i = x_i / M$$

- CPLEX default integrality tolerance: 1e-5

$$x_i = 100,\ M = 1e^{+10} \Rightarrow z_i = x_i / M = 1e^{-8}$$
$$z_i \text{ not eligible for branching unless } M \leq 1e^{+7}$$

“integer feasible” solution within integrality tolerance that violates intent of the model (trickle flow)

Examples

- To get correct answers with big-M formulation
  - Use smallest possible value of big-M that doesn't violate intent of model
    - Bound strengthening in CPLEX presolve often does this automatically
  - Set integrality tolerance to 0
  - Set simplex tolerances to minimum values, 1e-9
  - Ask for more accuracy on a potentially ill-conditioned system
    - Turn on numerical emphasis parameter
- Many users are unfamiliar with issues
  - Frequent source of CPLEX customer calls
  - One of most popular CPLEX FAQs
  - But should they have to be?

Examples

- Indicator constraint formulation for fixed costs on continuous variables

$$\text{Minimize}\; c^T x + f^T z \qquad (c,\, f \geq 0)$$
$$\text{subject to}\;\; Ax = b$$
$$\boxed{z_i = 0 \rightarrow x_i \leq 0}$$
$$x_i \geq 0,\;\; 0 \leq z_i \leq 1$$
$$z_i \;\; \text{integer}$$

(CPLEX branches on these directly)

  – LP relaxation solution

$$x_i = 100,\;\; z_i = 0$$

indicator constraint i requires branching

(integer feasible solutions aligned with intent of the model)

Examples

- Which approach to use?
  - Indicator formulation more precise representation of model
    - Indicator and big-M formulation equivalent when M=∞
  - If we can use modest values for big-M, indicator formulation tends to be weaker
  - Use indicator constraints, let CPLEX decide whether to replace with big-Ms if preprocessing can deduce big-M values of modest size
    - Presolve tightens the indicator formulation (improved further in CPLEX 12.2.0)
      - Presolve on indicators (improved)
      - Node presolve on indicators
      - Probing on by default
      - Probing on indicator constraints
      - Re-presolve by default