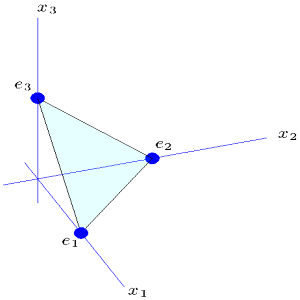
Efficient Validation of Basic Solutions via the Roundoff-Error-Free Factorization Framework

Adolfo R. Escobedo and Erick Moreno-Centeno



ARIZONA STATE UNIVERSITY

ICMS: July 12th, 2016

Outline

1. Introduction
   - Roundoff Errors in LP and MIP
   - State-of-the-Art Exact LP

2. The Roundoff-Error-Free (REF) Factorization Framework
   - Integer-Preserving Gaussian Elimination (IPGE)
   - REF Algorithms
   - Computational Tests

Roundoff errors in LP and MIP

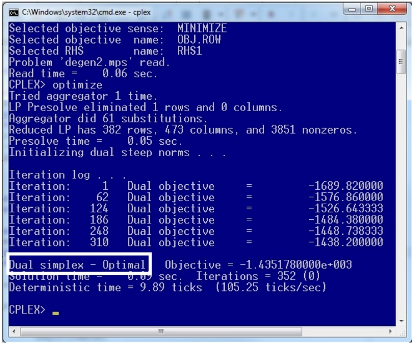
The Bottom Line

"Existing LP-solvers do not claim to solve LPs to optimality. In fact, **they come with hardly any guarantee**. Feasible problems may be classified as infeasible and vice versa, a solution returned is not guaranteed to be feasible, and the objective value returned comes with no approximation guarantee." (Dhiflaoui et al. 2003)

Roundoff errors in LP and MIP

The Bottom Line

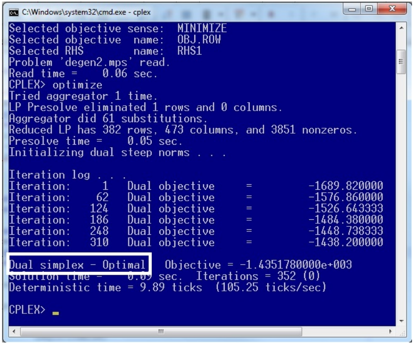
"Existing LP-solvers do not claim to solve LPs to optimality. In fact, **they come with hardly any guarantee**. Feasible problems may be classified as infeasible and vice versa, a solution returned is not guaranteed to be feasible, and the objective value returned comes with no approximation guarantee." (Dhiflaoui et al. 2003)



Roundoff errors in LP and MIP

The Bottom Line

"Existing LP-solvers do not claim to solve LPs to optimality. In fact, **they come with hardly any guarantee**. Feasible problems may be classified as infeasible and vice versa, a solution returned is not guaranteed to be feasible, and the objective value returned comes with no approximation guarantee." (Dhiflaoui et al. 2003)



Additional concerns

- Not exclusive to large problems
- "In mixed-integer optimization problems, numerical stability is of overriding importance to prevent wrong fathoming."(Suhl and Suhl 1990)

Roundoff errors in LP and MIP

CPLEX 7.1 Primal	$-6,398.71$	85.76
CPLEX 7.1 Dual	$-6,484.44$	0.03
CPLEX 9.0 Primal	$-6,406.78$	77.69
CPLEX 9.0 Dual	$-6,484.47$	0.00
CPLEX 11.0 Primal	$-6,425.87$	58.60
CPLEX 11.0 Dual	$-6,484.46$	0.01
CPLEX 12.1 Primal	$-6,425.87$	58.60
CPLEX 12.1 Dual	$-6,484.47$	0.00
CPLEX 12.4 Primal	$-6,424.23$	60.24
CPLEX 12.4 Dual	$-6,441.56$	42.91
CPLEX 12.4 Barrier	$-6,460.43$	24.04
Gurobi 2.0 Primal	$-6,484.47$	0.00
Gurobi 2.0 Dual	$-6,484.47$	0.00
XPress-15 Primal	$-6,380.45$	104.02
XPress-15 Dual	$-6,344.30$	140.17
XPress-20 Primal	$-6,349.93$	134.54
XPress-20 Dual	$-6,408.02$	76.45
QSopt Primal	$-6,419.94$	64.53
QSopt Dual	$-6,480.33$	4.14
CLP-1.02.01	$-6,480.95$	3.52
CLP-1.12.0	$-6,481.26$	3.21
GLPK-4.37	$-6,463.66$	20.81
GLPK-4.44	$-6,484.47$	0.00
MOSEK 6.0	$-6,292.06$	192.41
SoPlex 1.2.2	$-6,473.33$	11.14

sgpf5y6 (Mittelmann 2006)

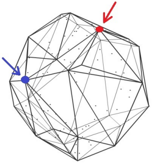
Inconsistent "optimality"

- **Exact:** ≈ -6484.47
- **Smallest:** ≈ -6292.06
- **Largest:** ≈ -6484.47

Adapted from (Steffy 2011)

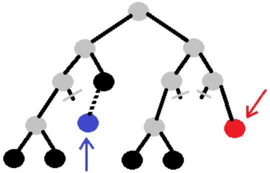
Roundoff errors in LP and MIP

Effect on LP



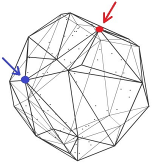
$\Rightarrow$

Effect on MIP



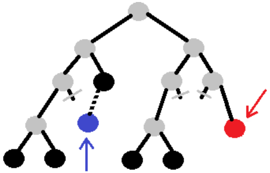
Roundoff errors in LP and MIP

Effect on LP



$\Rightarrow$

Effect on MIP

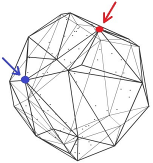


Mathematical applications

- Computer-assisted mathematics
- Numerically difficult problems
- Cut generation

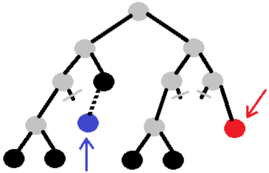
Roundoff errors in LP and MIP

Effect on LP



$\Rightarrow$

Effect on MIP



Mathematical applications

- Computer-assisted mathematics
- Numerically difficult problems
- Cut generation

Other applications

- Combinatorial auctions
- Robot control
- Chip design verification

Why not use an extended-precision FULL LP solver?

Extended precision comes at a high cost

Why not use an extended-precision FULL LP solver?

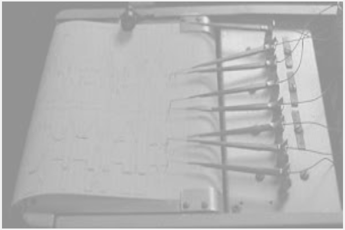
Extended precision comes at a high cost

Exactness of every operation is excessive

Why not use an extended-precision FULL LP solver?

Extended precision comes at a high cost

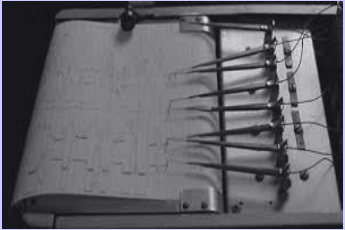
Exactness of every operation is excessive



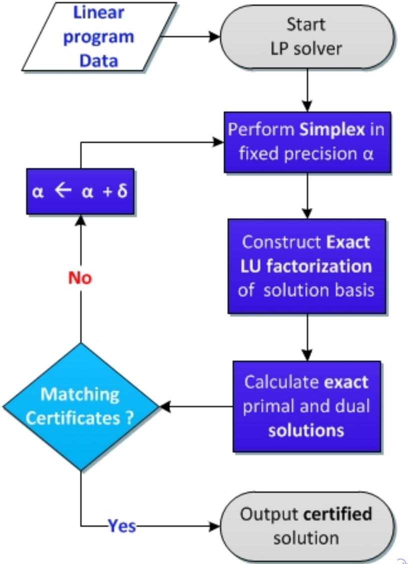
Why not use an extended-precision FULL LP solver?

Extended precision comes at a high cost

Exactness of every operation is excessive

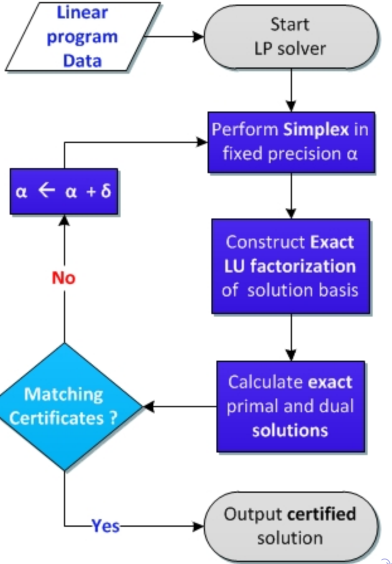


The Certify-and-Repair LP strategy (Dhiflaoui et al. 2003)



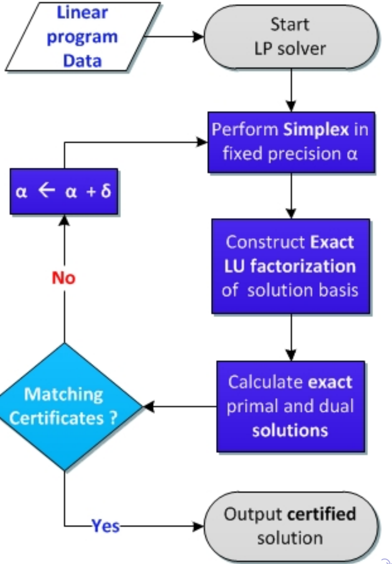
The Certify-and-Repair LP strategy (Dhiflaoui et al. 2003)

- Bulk of operations are **floating-point**



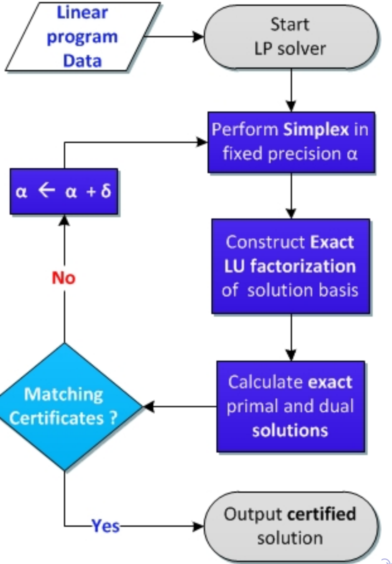
The Certify-and-Repair LP strategy (Dhiflaoui et al. 2003)

- Bulk of operations are **floating-point**
- Certify output via **unlimited-precision**



The Certify-and-Repair LP strategy (Dhiflaoui et al. 2003)

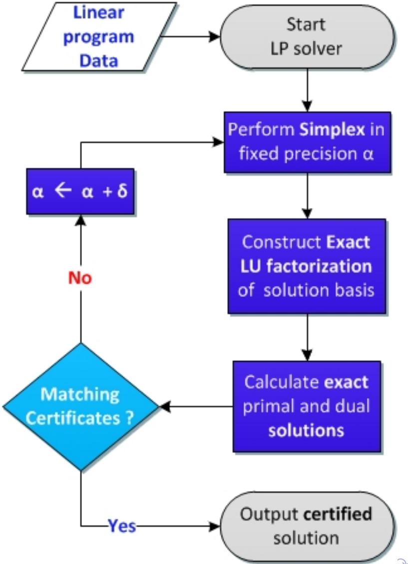
- Bulk of operations are **floating-point**
- Certify output via **unlimited-precision**
- Two state-of-the-art implementations



The Certify-and-Repair LP strategy (Dhiflaoui et al. 2003)

- Bulk of operations are **floating-point**
- Certify output via **unlimited-precision**
- Two state-of-the-art implementations

$\Rightarrow$ **Hybrid strategies**



State-of-the-art exact LP solvers

- Precision-boosting approach used in QSopt_ex
  - A **rational LU factorization** of the basis calculated after solver concludes for some precision level

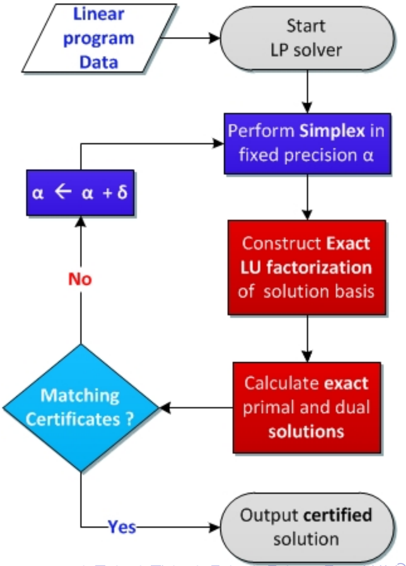
State-of-the-art exact LP solvers

- Precision-boosting approach used in QSopt_ex
  - A **rational LU factorization** of the basis calculated after solver concludes for some precision level
- Exact solving capability of SoPlex release 2.2.0:
  - A **rational LU factorization** of the basis calculated after promising refinement rounds
  - Rational reconstruction via continued fraction approximations

⇒ Improvements to **exact LU factorization** benefit both solvers

The Certify-and-Repair LP strategy

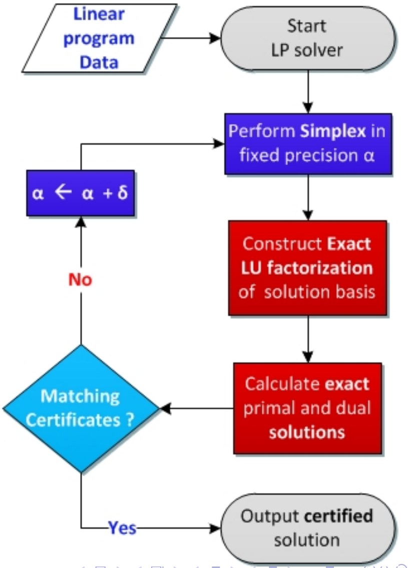
- Procedure bottleneck in **validation:**



The Certify-and-Repair LP strategy

- Procedure bottleneck in **validation:**

⋆ Exact rational arithmetic

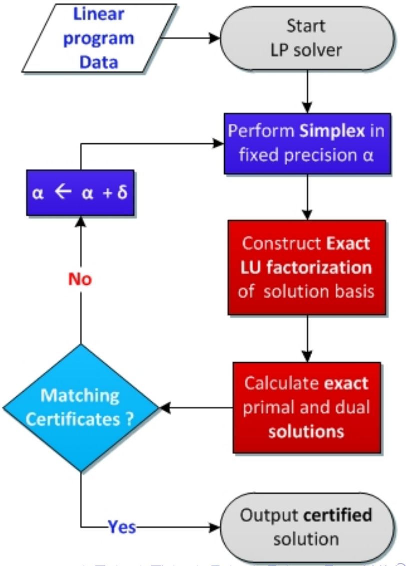


The Certify-and-Repair LP strategy

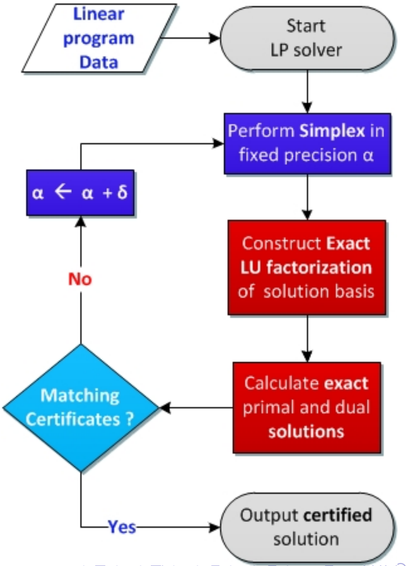
- Procedure bottleneck in **validation:**

⋆ Exact rational arithmetic

⋆ Extra storage



- Procedure bottleneck in **validation:**
  - ⋆ Exact rational arithmetic
  - ⋆ Extra storage
  - ⋆ Repeated (**GCD**) operations



Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$

Gaussian elimination using rational arithmetic

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$
4.	549	147	$a \leftarrow a - b$

Gaussian elimination using rational arithmetic

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$
4.	549	147	$a \leftarrow a - b$
5.	402	147	$a \leftarrow a/2$

Gaussian elimination using rational arithmetic

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$
4.	549	147	$a \leftarrow a - b$
5.	402	147	$a \leftarrow a/2$
6.	201	147	$a \leftarrow a - b$

Gaussian elimination using rational arithmetic

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$
4.	549	147	$a \leftarrow a - b$
5.	402	147	$a \leftarrow a/2$
6.	201	147	$a \leftarrow a - b$
7.	54	147	$a \leftarrow a/2$

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$
4.	549	147	$a \leftarrow a - b$
5.	402	147	$a \leftarrow a/2$
6.	201	147	$a \leftarrow a - b$
7.	54	147	$a \leftarrow a/2$
8.	27	147	$b \leftarrow b - a$

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$
4.	549	147	$a \leftarrow a - b$
5.	402	147	$a \leftarrow a/2$
6.	201	147	$a \leftarrow a - b$
7.	54	147	$a \leftarrow a/2$
8.	27	147	$b \leftarrow b - a$
9.	27	120	$b \leftarrow b/2$

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	**2343**	**147**	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$
4.	549	147	$a \leftarrow a - b$
5.	402	147	$a \leftarrow a/2$
6.	201	147	$a \leftarrow a - b$
7.	54	147	$a \leftarrow a/2$
8.	27	147	$b \leftarrow b - a$
9.	27	120	$b \leftarrow b/2$
10.	27	60	$b \leftarrow b/2$

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	**2343**	**147**	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$
4.	549	147	$a \leftarrow a - b$
5.	402	147	$a \leftarrow a/2$
6.	201	147	$a \leftarrow a - b$
7.	54	147	$a \leftarrow a/2$
8.	27	147	$b \leftarrow b - a$
9.	27	120	$b \leftarrow b/2$
10.	27	60	$b \leftarrow b/2$

Step	a	b	Action
11.	27	30	$b \leftarrow b/2$
12.	27	15	$a \leftarrow a - b$
13.	12	15	$a \leftarrow a/2$
14.	6	15	$a \leftarrow a/2$
15.	3	15	$b \leftarrow b - a$
16.	3	12	$b \leftarrow b/2$
17.	3	6	$b \leftarrow b/2$
18.	3	3	$b \leftarrow b - a$
19.	3	0	$b \leftarrow b - a$

Gaussian elimination using rational arithmetic

Binary Euclid's GCD Algorithm:

Step	a	b	Action
1.	$\mathbf{2343}$	$\mathbf{147}$	$a \leftarrow a - b$
2.	2196	147	$a \leftarrow a/2$
3.	1098	147	$a \leftarrow a/2$
4.	549	147	$a \leftarrow a - b$
5.	402	147	$a \leftarrow a/2$
6.	201	147	$a \leftarrow a - b$
7.	54	147	$a \leftarrow a/2$
8.	27	147	$b \leftarrow b - a$
9.	27	120	$b \leftarrow b/2$
10.	27	60	$b \leftarrow b/2$

Step	a	b	Action
11.	27	30	$b \leftarrow b/2$
12.	27	15	$a \leftarrow a - b$
13.	12	15	$a \leftarrow a/2$
14.	6	15	$a \leftarrow a/2$
15.	3	15	$b \leftarrow b - a$
16.	3	12	$b \leftarrow b/2$
17.	3	6	$b \leftarrow b/2$
18.	3	3	$b \leftarrow b - a$
19.	3	0	$b \leftarrow b - a$

$O(N^2)$ $\Rightarrow$ **19 steps**: GCD$(2343, 147) = 3$ $\Rightarrow$ $a^3_{4,4} = \dfrac{-781}{49}$

Outline

1. Introduction
   - Roundoff Errors in LP and MIP
   - State-of-the-Art Exact LP
2. The Roundoff-Error-Free (REF) Factorization Framework
   - Integer-Preserving Gaussian Elimination (IPGE)
   - REF Algorithms
   - Computational Tests

Is there a better recipe?

Want an algorithm that...

- Avoids rational arithmetic
- Bound word length polynomially
- Has a predictable number of steps

Is there a better recipe?

Want an algorithm that...

- Avoids rational arithmetic
- Bound word length polynomially
- Has a predictable number of steps

$$a_{i,j}^{k} = a_{i,j}^{k-1} - \frac{a_{k,j}^{k-1}}{a_{k,k}^{k-1}} a_{i,k}^{k-1} = \frac{a_{k,k}^{k-1} a_{i,j}^{k-1} - a_{k,j}^{k-1} a_{i,k}^{k-1}}{a_{k,k}^{k-1}} \quad \Leftarrow \quad \textbf{GE}$$

$$\Downarrow$$

Is there a better recipe?

Want an algorithm that...

- Avoids rational arithmetic
- Bound word length polynomially
- Has a predictable number of steps

$$a_{i,j}^{k} = a_{i,j}^{k-1} - \frac{a_{k,j}^{k-1}}{a_{k,k}^{k-1}} a_{i,k}^{k-1} = \frac{a_{k,k}^{k-1} a_{i,j}^{k-1} - a_{k,j}^{k-1} a_{i,k}^{k-1}}{a_{k,k}^{k-1}} \quad \Leftarrow \quad \textbf{GE}$$

$$\Downarrow$$

$$a_{i,j}^{k} = \frac{a_{k,k}^{k-1} a_{i,j}^{k-1} - a_{k,j}^{k-1} a_{i,k}^{k-1}}{\boldsymbol{a_{k-1,k-1}^{k-2}}} \quad \Leftarrow \textbf{IPGE}$$

IPGE properties

- Its divisions are **exact** (Edmonds 1967):

  Given $A^0 \in \mathbb{Z}^{n\times n+1}$, $A^k \in \mathbb{Z}^{n\times n+1} \;\; \forall k$

- Its entries have a **special structure** (Bareiss 1968):

$$a^k_{i,j} = \begin{cases} (-1)^{i+k} \det\left(A^{1,..,k}_{1,..,i-1,i+1,..,k,j}\right) & \text{if } i \leq k \\ \det\left(A^{1,..,k,i}_{1,..,k,j}\right) & \text{otherwise} \end{cases}$$

- Each entry's bit-length is bounded **polynomially** (Bareiss 1972) :

  $\lceil \log(|a^k_{i,j}|) \rceil \leq \lceil \log(\sigma^n n^{\frac{n}{2}}) \rceil = \lceil n \log(\sigma\sqrt{n}) \rceil$

- **No gcd calculations are required!**

LU and Cholesky factorizations

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}}_{\textbf{A}} = \underbrace{\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix}}_{\textbf{L}} \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}}_{\textbf{U}}$$

LU and Cholesky factorizations

A **L** **U**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Forward substitution

$$\begin{bmatrix} {\color{red}l_{11}} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} {\color{red}y_1} \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} {\color{red}b_1} \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

LU and Cholesky factorizations

A **L** **U**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Forward substitution

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & {\color{red}l_{22}} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} y_1 \\ {\color{red}y_2} \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ {\color{red}b_2} \\ b_3 \\ b_4 \end{bmatrix}$$

LU and Cholesky factorizations

A **L** **U**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Forward substitution

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

LU and Cholesky factorizations

A **L** **U**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Forward substitution

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & \color{red}{l_{44}} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \color{red}{y_4} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \color{red}{b_4} \end{bmatrix}$$

LU and Cholesky factorizations

A **L** **U**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Backward substitution

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

LU and Cholesky factorizations

A **L** **U**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Backward substitution

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

LU and Cholesky factorizations

A **L** **U**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Backward substitution

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

LU and Cholesky factorizations

A **L** **U**

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

Backward substitution

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

The Roundoff-Error-Free Factorizations

REF LU:

$$\begin{bmatrix} \boldsymbol{a^0_{1,1}} & & & & \\ a^0_{2,1} & \boldsymbol{a^1_{2,2}} & & & \\ a^0_{3,1} & a^1_{3,2} & \boldsymbol{a^2_{3,3}} & & \\ \vdots & \vdots & \vdots & \ddots & \\ a^0_{n,1} & a^1_{n,2} & a^2_{n,3} & \dots & \boldsymbol{a^{n-1}_{n,n}} \end{bmatrix} \boldsymbol{D^{-1}} \begin{bmatrix} \boldsymbol{a^0_{1,1}} & a^0_{1,2} & a^0_{1,3} & \dots & a^0_{1,n} \\ & \boldsymbol{a^1_{2,2}} & a^1_{2,3} & \dots & a^1_{2,n} \\ & & \boldsymbol{a^2_{3,3}} & \dots & a^2_{3,n} \\ & & & \ddots & \vdots \\ & & & & \boldsymbol{a^{n-1}_{n,n}} \end{bmatrix}$$

$$\boldsymbol{D} = diag\left(\boldsymbol{a^0_{1,1}}\ ,\ \boldsymbol{a^0_{1,1}a^1_{2,2}}\ ,\ \boldsymbol{a^1_{2,2}a^2_{3,3}}\ ,\ \dots\ ,\ \boldsymbol{a^{n-2}_{n-1,n-1}a^{n-1}_{n,n}}\right)$$

REF LU:

$$\begin{bmatrix} \boldsymbol{a^0_{1,1}} & & & & \\ a^0_{2,1} & \boldsymbol{a^1_{2,2}} & & & \\ a^0_{3,1} & a^1_{3,2} & \boldsymbol{a^2_{3,3}} & & \\ \vdots & \vdots & \vdots & \ddots & \\ a^0_{n,1} & a^1_{n,2} & a^2_{n,3} & \cdots & \boldsymbol{a^{n-1}_{n,n}} \end{bmatrix} \boldsymbol{D^{-1}} \begin{bmatrix} \boldsymbol{a^0_{1,1}} & a^0_{1,2} & a^0_{1,3} & \cdots & a^0_{1,n} \\ & \boldsymbol{a^1_{2,2}} & a^1_{2,3} & \cdots & a^1_{2,n} \\ & & \boldsymbol{a^2_{3,3}} & \cdots & a^2_{3,n} \\ & & & \ddots & \vdots \\ & & & & \boldsymbol{a^{n-1}_{n,n}} \end{bmatrix}$$

$$\boldsymbol{D} = diag\left(\boldsymbol{a^0_{1,1}}\ ,\ \boldsymbol{a^0_{1,1}a^1_{2,2}}\ ,\ \boldsymbol{a^1_{2,2}a^2_{3,3}}\ ,\ \ldots\ ,\ \boldsymbol{a^{n-2}_{n-1,n-1}a^{n-1}_{n,n}}\right)$$

REF Cholesky:

$$(\boldsymbol{L}\sqrt{\boldsymbol{D^{-1}}})(\boldsymbol{L}\sqrt{\boldsymbol{D^{-1}}})^T = \boldsymbol{L}\boldsymbol{D^{-1}}\boldsymbol{L^T}$$

Exact forward and backward substitution

$$\begin{bmatrix} \mathbf{3} & 0 & 0 & 0 \\ 5 & \mathbf{-49} & 0 & 0 \\ 6 & -87 & \mathbf{-17} & 0 \\ 7 & -47 & 527 & \mathbf{884} \end{bmatrix} \quad \begin{bmatrix} \frac{\mathbf{1}}{\mathbf{3}} & 0 & 0 & 0 \\ 0 & \frac{\mathbf{1}}{\mathbf{3(-49)}} & 0 & 0 \\ 0 & 0 & \frac{\mathbf{1}}{\mathbf{-49(-17)}} & 0 \\ 0 & 0 & 0 & \frac{\mathbf{1}}{\mathbf{-17(884)}} \end{bmatrix} \quad \begin{bmatrix} \mathbf{3} & 11 & 8 & 7 \\ 0 & \mathbf{-49} & -31 & -20 \\ 0 & 0 & \mathbf{-17} & 57 \\ 0 & 0 & 0 & \mathbf{884} \end{bmatrix}$$

Solve $Ax = b$ using $(\boldsymbol{LD^{-1}U})x = b$:

$$b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

Exact forward and backward substitution

$$\begin{bmatrix} \mathbf{3} & 0 & 0 & 0 \\ 5 & \mathbf{-49} & 0 & 0 \\ 6 & -87 & \mathbf{-17} & 0 \\ 7 & -47 & 527 & \mathbf{884} \end{bmatrix} \quad \begin{bmatrix} \frac{\mathbf{1}}{\mathbf{3}} & 0 & 0 & 0 \\ 0 & \frac{\mathbf{1}}{\mathbf{3(-49)}} & 0 & 0 \\ 0 & 0 & \frac{\mathbf{1}}{\mathbf{-49(-17)}} & 0 \\ 0 & 0 & 0 & \frac{\mathbf{1}}{\mathbf{-17(884)}} \end{bmatrix} \quad \begin{bmatrix} \mathbf{3} & 11 & 8 & 7 \\ 0 & \mathbf{-49} & -31 & -20 \\ 0 & 0 & \mathbf{-17} & 57 \\ 0 & 0 & 0 & \mathbf{884} \end{bmatrix}$$

Solve $Ax = b$ using $(\boldsymbol{L}\boldsymbol{D}^{-1}\boldsymbol{U})x = b$:

$$b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \Rightarrow$$

$\boldsymbol{L}y = b$: $\quad 3y_1 = 1 \Rightarrow y_1 = \frac{1}{3}$

Exact forward and backward substitution

$$\begin{bmatrix} \mathbf{3} & 0 & 0 & 0 \\ 5 & \mathbf{-49} & 0 & 0 \\ 6 & -87 & \mathbf{-17} & 0 \\ 7 & -47 & 527 & \mathbf{884} \end{bmatrix} \quad \begin{bmatrix} \frac{\mathbf{1}}{\mathbf{3}} & 0 & 0 & 0 \\ 0 & \frac{\mathbf{1}}{\mathbf{3(-49)}} & 0 & 0 \\ 0 & 0 & \frac{\mathbf{1}}{\mathbf{-49(-17)}} & 0 \\ 0 & 0 & 0 & \frac{\mathbf{1}}{\mathbf{-17(884)}} \end{bmatrix} \quad \begin{bmatrix} \mathbf{3} & 11 & 8 & 7 \\ 0 & \mathbf{-49} & -31 & -20 \\ 0 & 0 & \mathbf{-17} & 57 \\ 0 & 0 & 0 & \mathbf{884} \end{bmatrix}$$

Solve $Ax = b$ using $(\boldsymbol{L}\boldsymbol{D}^{-1}\boldsymbol{U})x = b$:

$$b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \Rightarrow$$

$\boldsymbol{L}y = b$: $\quad 3y_1 = 1 \quad \Rightarrow \quad y_1 = \frac{1}{3}$

$(\boldsymbol{L}\boldsymbol{D}^{-1})y = b$:

$$3\left(\frac{1}{3}\right)y_1 = 1 \quad \Rightarrow \quad y_1 = 1$$

$$\vdots$$

$$\frac{-1}{17}y_4 = 4 - \frac{7}{3} - \frac{47}{3(-49)} + \frac{620}{49}$$

Roundoff-Error-Free Substitution

REF Forward Substitution

$$\psi_{i,r} = \begin{cases} b_i & \text{if } r = 0 \\ \frac{l_{r,r}\psi_{i,r-1} - l_{i,r}\psi_{r,r-1}}{l_{r-1,r-1}} & \text{if } 0 < r < i \end{cases} \quad \text{for} \quad i = 1..m$$

REF Backward Substitution

$$x'_i = \frac{1}{u_{i,i}}\left(y'_i - \sum_{h=i+1}^{m} x'_h u_{i,h}\right) \quad \text{for} \quad i = m \dots 1$$

$$b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

Roundoff-Error-Free Substitution

REF Forward Substitution

$$\psi_{i,r} = \begin{cases} b_i & \text{if } r = 0 \\ \frac{l_{r,r}\psi_{i,r-1} - l_{i,r}\psi_{r,r-1}}{l_{r-1,r-1}} & \text{if } 0 < r < i \end{cases} \quad \text{for} \quad i = 1..m$$

REF Backward Substitution

$$x'_i = \frac{1}{u_{i,i}}\left(y'_i - \sum_{h=i+1}^{m} x'_h u_{i,h}\right) \quad \text{for} \quad i = m \dots 1$$

$$b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \Rightarrow y = \begin{bmatrix} 1 \\ 1 \\ -20 \\ -238 \end{bmatrix}$$

Roundoff-Error-Free Substitution

REF Forward Substitution

$$\psi_{i,r} = \begin{cases} b_i & \text{if } r = 0 \\ \frac{l_{r,r}\psi_{i,r-1} - l_{i,r}\psi_{r,r-1}}{l_{r-1,r-1}} & \text{if } 0 < r < i \end{cases} \quad \text{for} \quad i = 1..m$$

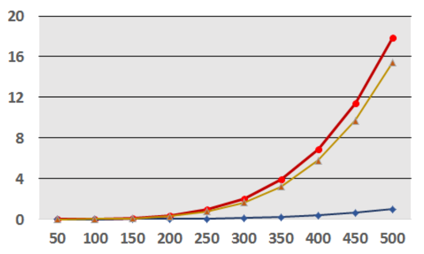
REF Backward Substitution

$$x'_i = \frac{1}{u_{i,i}} \left(y'_i - \sum_{h=i+1}^{m} x'_h u_{i,h} \right) \quad \text{for} \quad i = m \dots 1$$

$$b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \Rightarrow y = \begin{bmatrix} 1 \\ 1 \\ -20 \\ -238 \end{bmatrix} \Rightarrow x' = \begin{bmatrix} 476 \\ -74 \\ 242 \\ -238 \end{bmatrix}$$

Construction and solution (in secs)

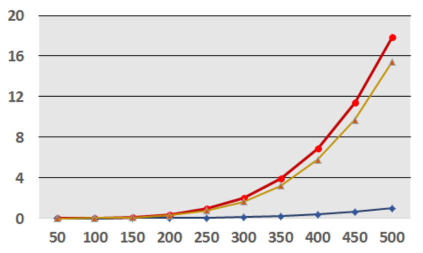
REF LU vs $\mathbb{Q}$LU (n,secs)



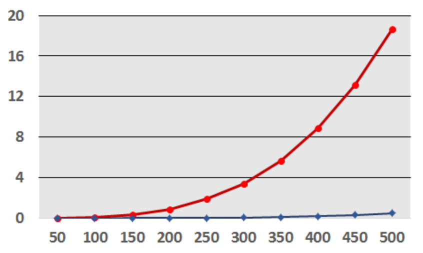
Computational advantages (1)

Construction and solution (in secs)

REF LU vs $\mathbb{Q}$LU (n,secs)

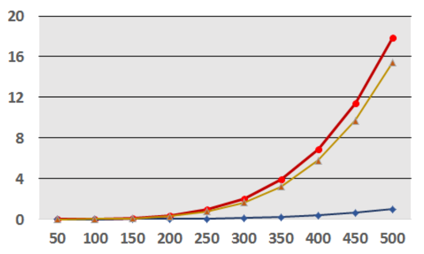


REF Sub vs $\mathbb{Q}$ Sub (n,secs)

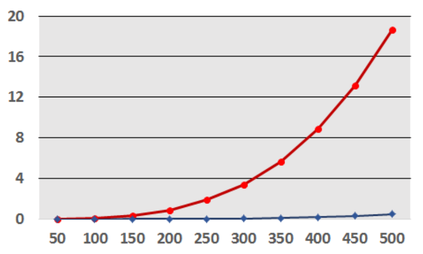


Construction and solution (in secs)

REF LU vs $\mathbb{Q}$LU (n,secs)



REF Sub vs $\mathbb{Q}$ Sub (n,secs)



$\Rightarrow$ Construction times $\geq \mathbf{15}\times$ faster

$\Rightarrow$ Substitution times $\geq \mathbf{35}\times$ faster

An alternative IPGE-based optimization approach

Simplex algorithm based on IPGE:

- Q-matrix framework (Edmonds and Maurras 1997)
- Q-matrix revised simplex method (Azulay and Pique 2001)

Methodology is **obscure** and **unutilized:**

$\Rightarrow$ A full unlimited-precision LP solver

An alternative IPGE-based optimization approach

Simplex algorithm based on IPGE:

- Q-matrix framework (Edmonds and Maurras 1997)
- Q-matrix revised simplex method (Azulay and Pique 2001)

Methodology is **obscure** and **unutilized:**

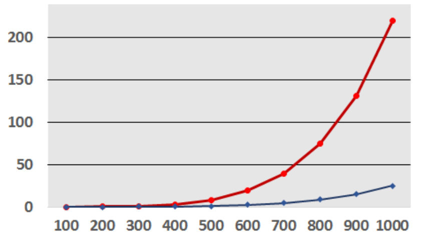
$\Rightarrow$ A full unlimited-precision LP solver

An additional comparison

- Efficient adaptation of Q-matrix framework
- **Construction routine**: Use IPGE to obtain Adj$(A) = \det(A)A^{-1}$
- **Solution routine**: Perform matrix-vector multiplication Adj$(A)b$

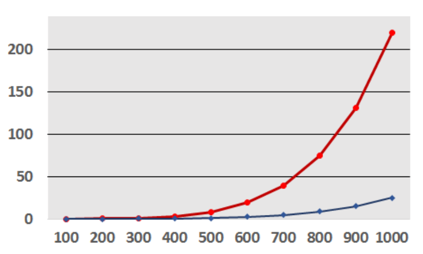
Construction (in mins) and **solution** (in secs)

Adj Q-Mat vs REF LU (n, mins)

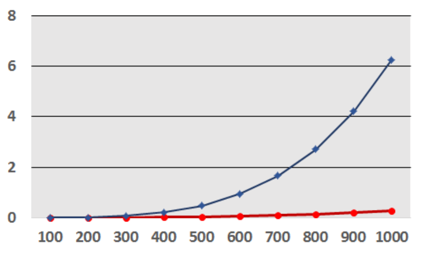


Construction (in mins) and **solution** (in secs)

Adj Q-Mat vs REF LU (n, mins)



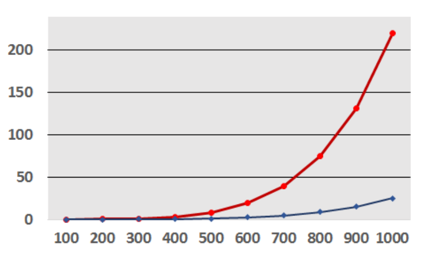
Adj$(A)b$ vs REF Sub (n, secs)



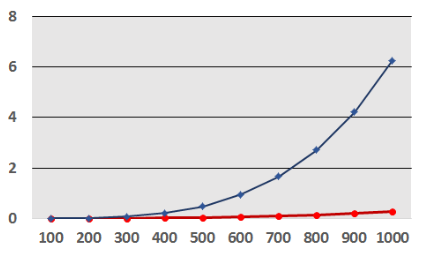
Computational advantages (2)

Construction (in <u>mins</u>) and **solution** (in secs)

Adj Q-Mat vs REF LU (n, <u>mins</u>)



Adj$(A)b$ vs REF Sub (n, secs)

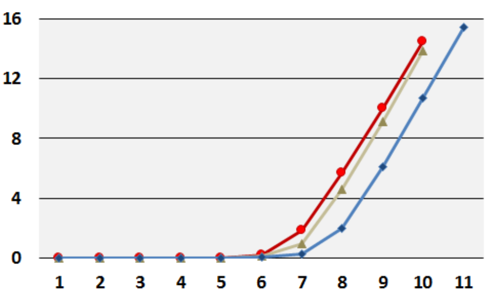


$\Rightarrow$ Construction times $\geq 7\times$ faster

$\Rightarrow$ Substitution times $\geq 7\times$ slower

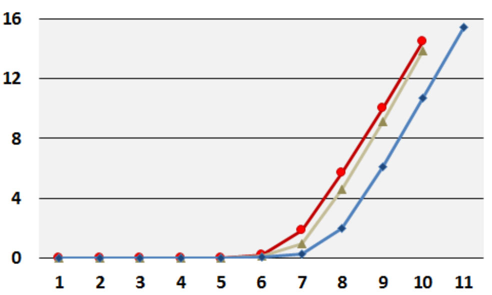
Asymptotic performance $(\log_2 n, \log_2 \text{secs})$

$\mathbb{Q}$LU vs Adj Q-Mat vs REF LU



Asymptotic performance $(\log_2 n, \log_2 \text{secs})$

$\mathbb{Q}$LU vs Adj Q-Mat vs REF LU



$\Rightarrow$ $\mathbb{Q}$LU $\approx$ Adj Q-Mat for larger n

$\Rightarrow$ REF LU maintains its edge

Storage (in limbs)

n	**Crout/REF** AVG	**Doolittle/REF** AVG	**Q-Mat/REF** AVG
50	1.9583	1.9574	2.77
100	1.9757	1.9753	2.97
150	1.9822	1.9823	3.03
200	1.9862	1.9862	3.03
250	1.9890	1.9891	3.09
300	1.9908	1.9909	3.13
350	1.9920	1.9921	3.10
400	1.9929	1.9929	3.11
450	1.9937	1.9937	3.12
500	1.9944	1.9944	3.12

Storage (in limbs)

n	Crout/REF AVG	Doolittle/REF AVG	Q-Mat/REF AVG
50	1.9583	1.9574	2.77
100	1.9757	1.9753	2.97
150	1.9822	1.9823	3.03
200	1.9862	1.9862	3.03
250	1.9890	1.9891	3.09
300	1.9908	1.9909	3.13
350	1.9920	1.9921	3.10
400	1.9929	1.9929	3.11
450	1.9937	1.9937	3.12
500	1.9944	1.9944	3.12

$\Rightarrow$ $\mathbb{Q}$LU requires **2x the storage** of REF LU

$\Rightarrow$ Adj Q-Mat requires $>$**3x the storage** of REF LU

Questions??

Thank you!

(email: adRes@asu.edu)