# Numeric-Symbolic Exact Rational Linear System Solver

B. David Saunders, David Wood, and Bryan Youse

University of Delaware

June 10, 2011

# Motivation

**$Ax = b$**

Given $A \in \mathbb{Z}^{m \times n}$ and $b \in \mathbb{Z}^m$, compute $x \in \mathbb{Q}^n$.

Core problem specifics:

- Square $m = n$ matrices
- Matrix entries of length $d$ bits or fewer

# Background

Competing Approaches

- Purely Symbolic [Dixon, 1982]
  - Solve system modulo $p$
  - Use Hensel lifting to obtain a $p$-adic expansion of solution
  - Rational reconstruction from $p$-adic approximants

# Background

Competing Approaches

- Purely Symbolic [Dixon, 1982]
  - Solve system modulo $p$
  - Use Hensel lifting to obtain a $p$-adic expansion of solution
  - Rational reconstruction from $p$-adic approximants
- Numeric-Symbolic [Wan, 2006]
  - Numeric iterative refinement to obtain dyadic number solution of high accuracy
  - Specifically, $2\lg(h)$, the Hadamard bound of the input system.
  - Rational reconstruction from dyadic approximants

**Fact**

*If two rational numbers $r_1 = a/b, r_2 = c/d$ are given in lowest terms, and $r_1 \neq r_2$, then $|r_1 - r_2| \geq 1/bd$.*

That is, though dense in the real number line, rational numbers with bound denominators are discrete.

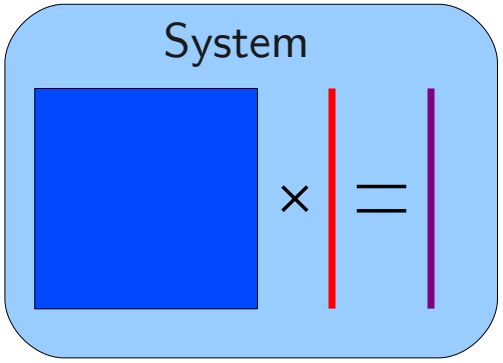
**Fact**

*If two rational numbers $r_1 = a/b, r_2 = c/d$ are given in lowest terms, and $r_1 \neq r_2$, then $|r_1 - r_2| \geq 1/bd$.*

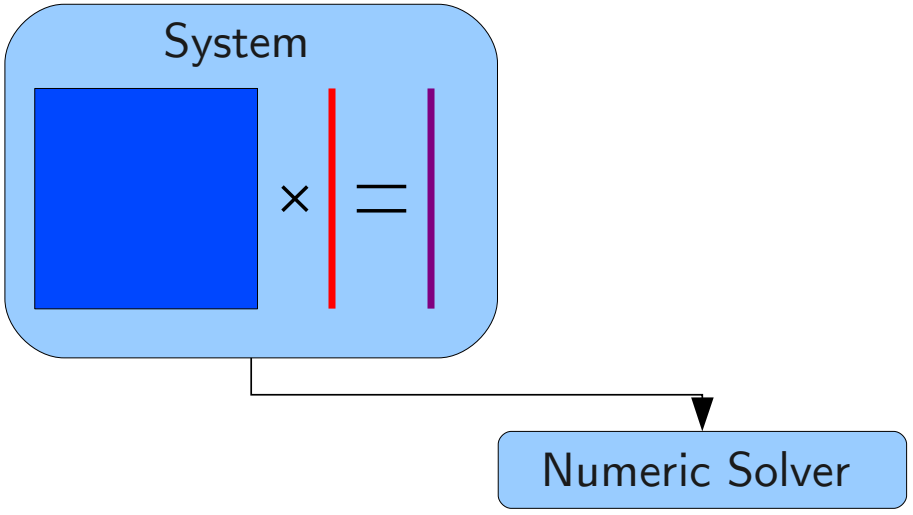
That is, though dense in the real number line, rational numbers with bound denominators are discrete.

A given real number $r$ can be represented by a continued fraction: $r = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$, where $a_i \in \mathbb{Z}$.

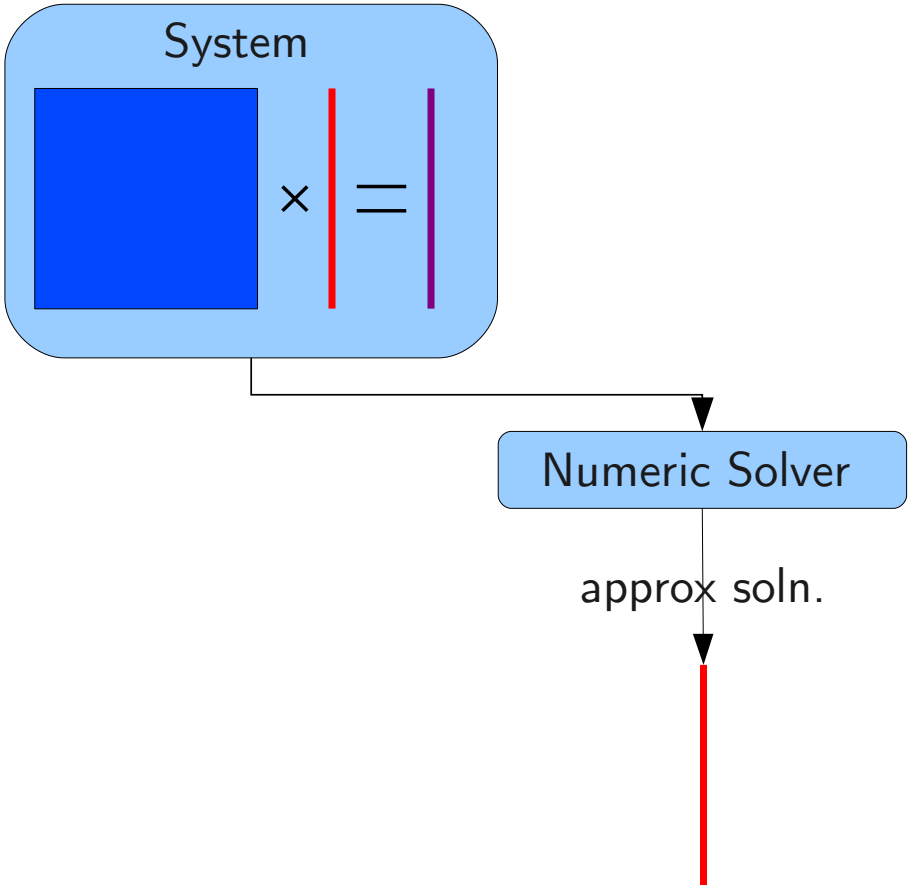
# Iterative Refinement



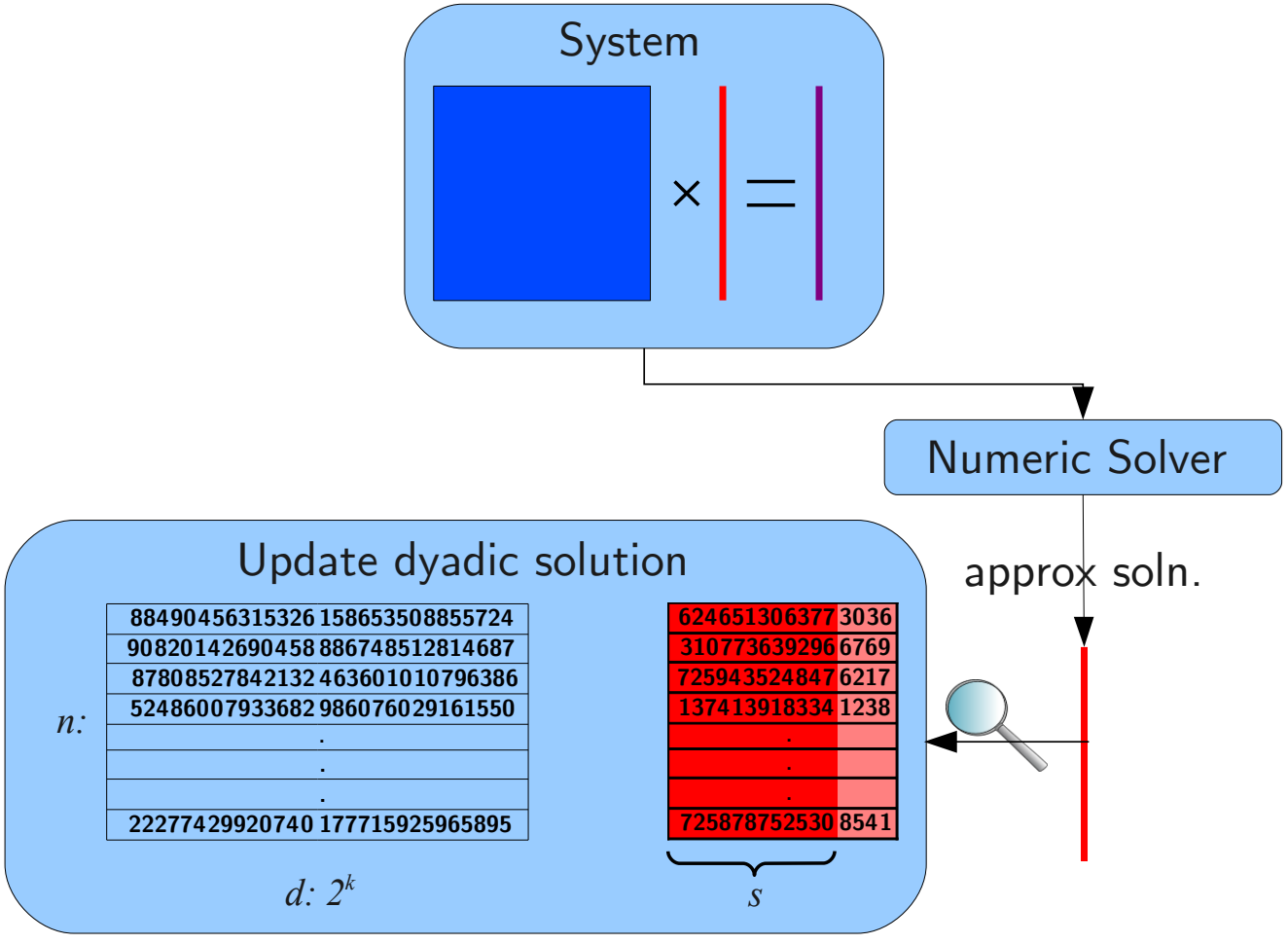
# Iterative Refinement



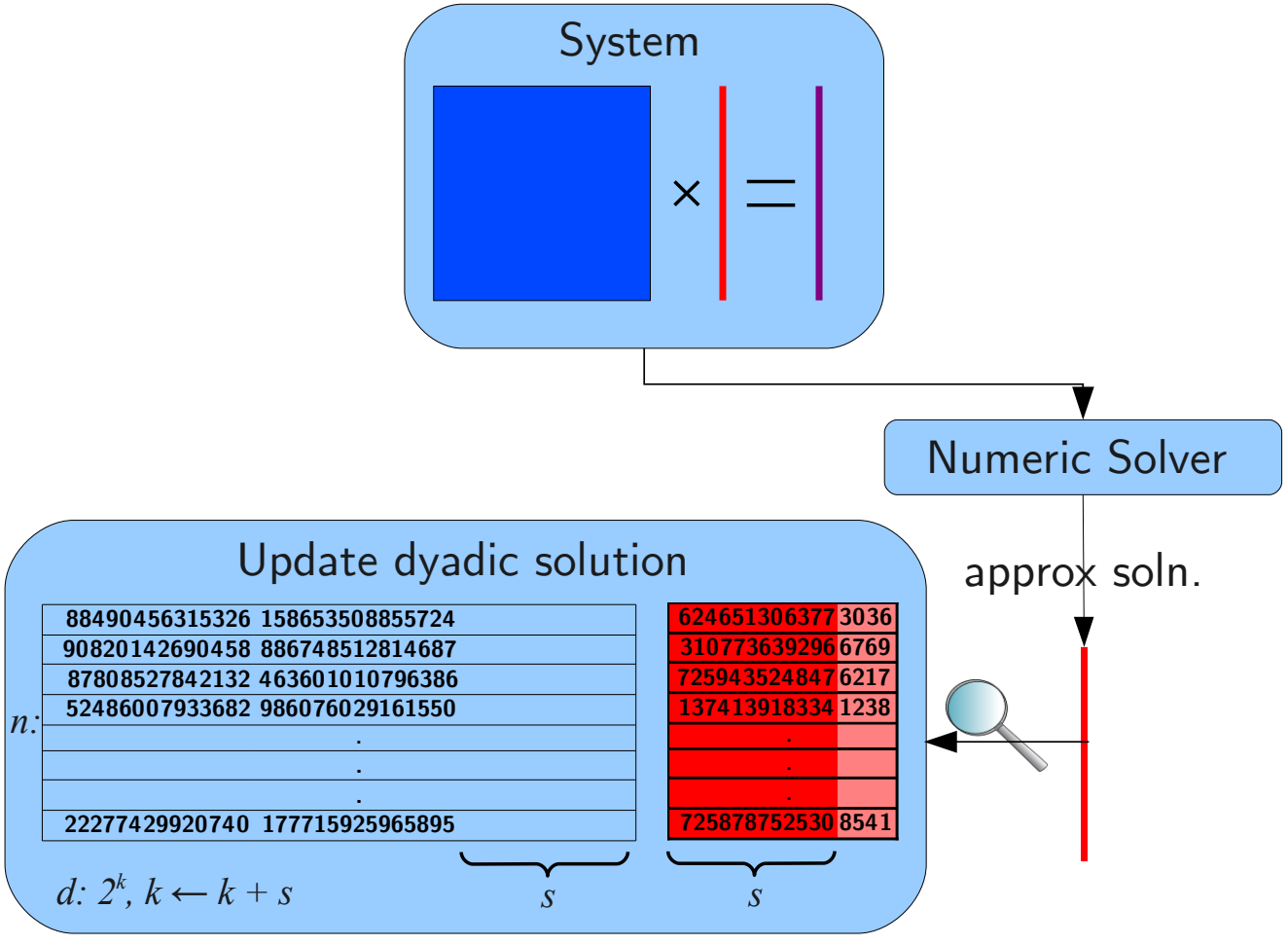
# Iterative Refinement



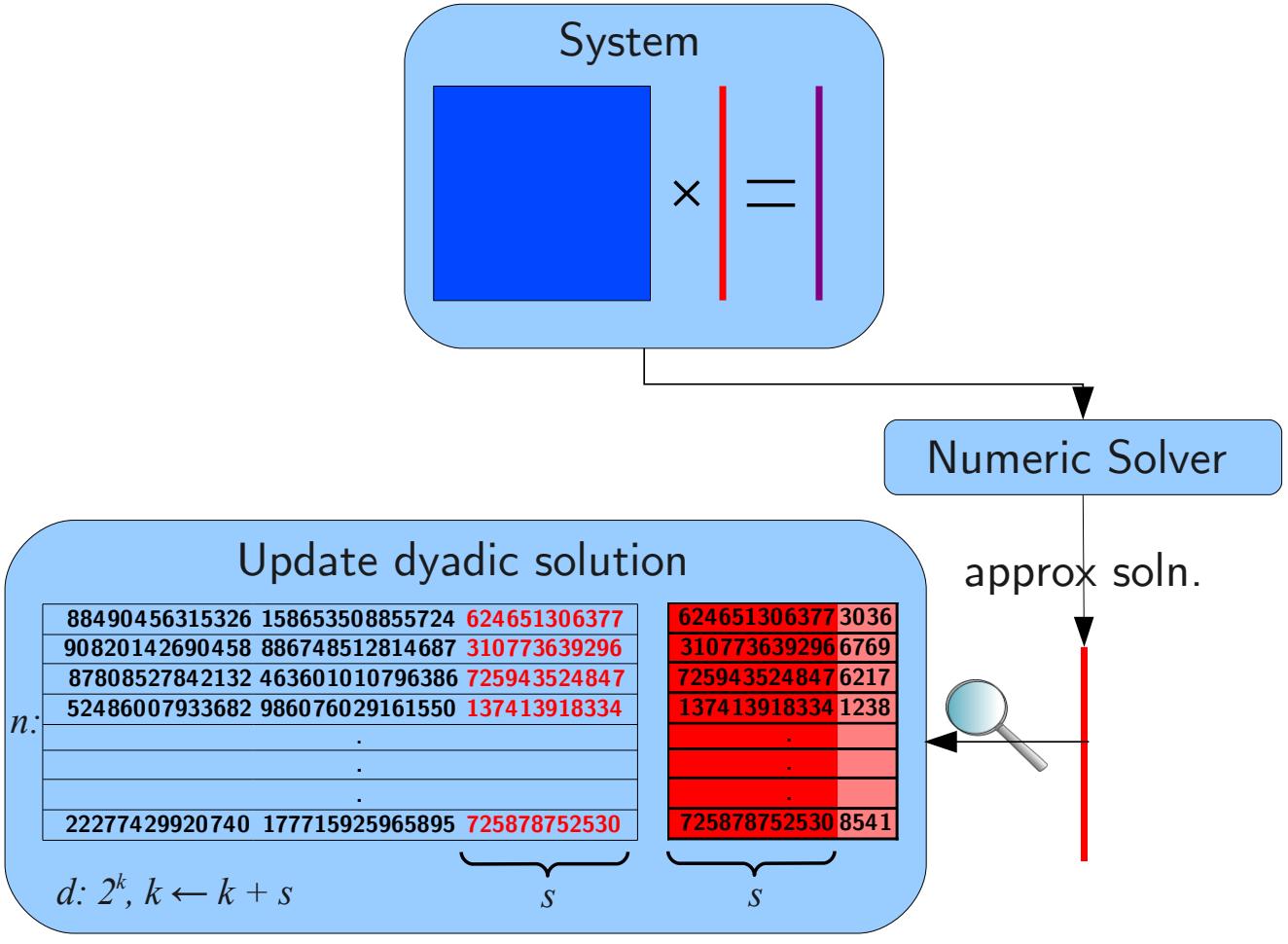
# Iterative Refinement



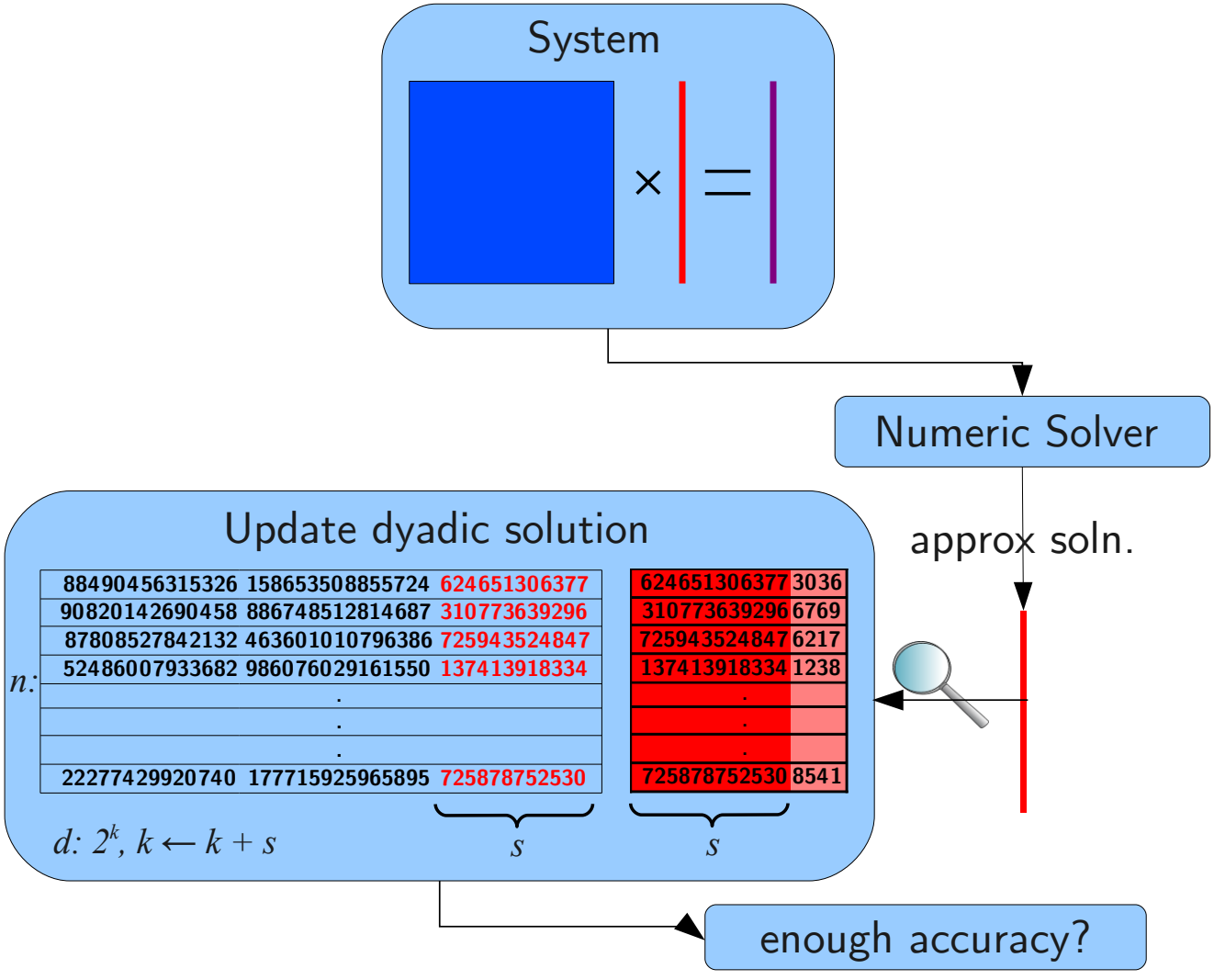
# Iterative Refinement



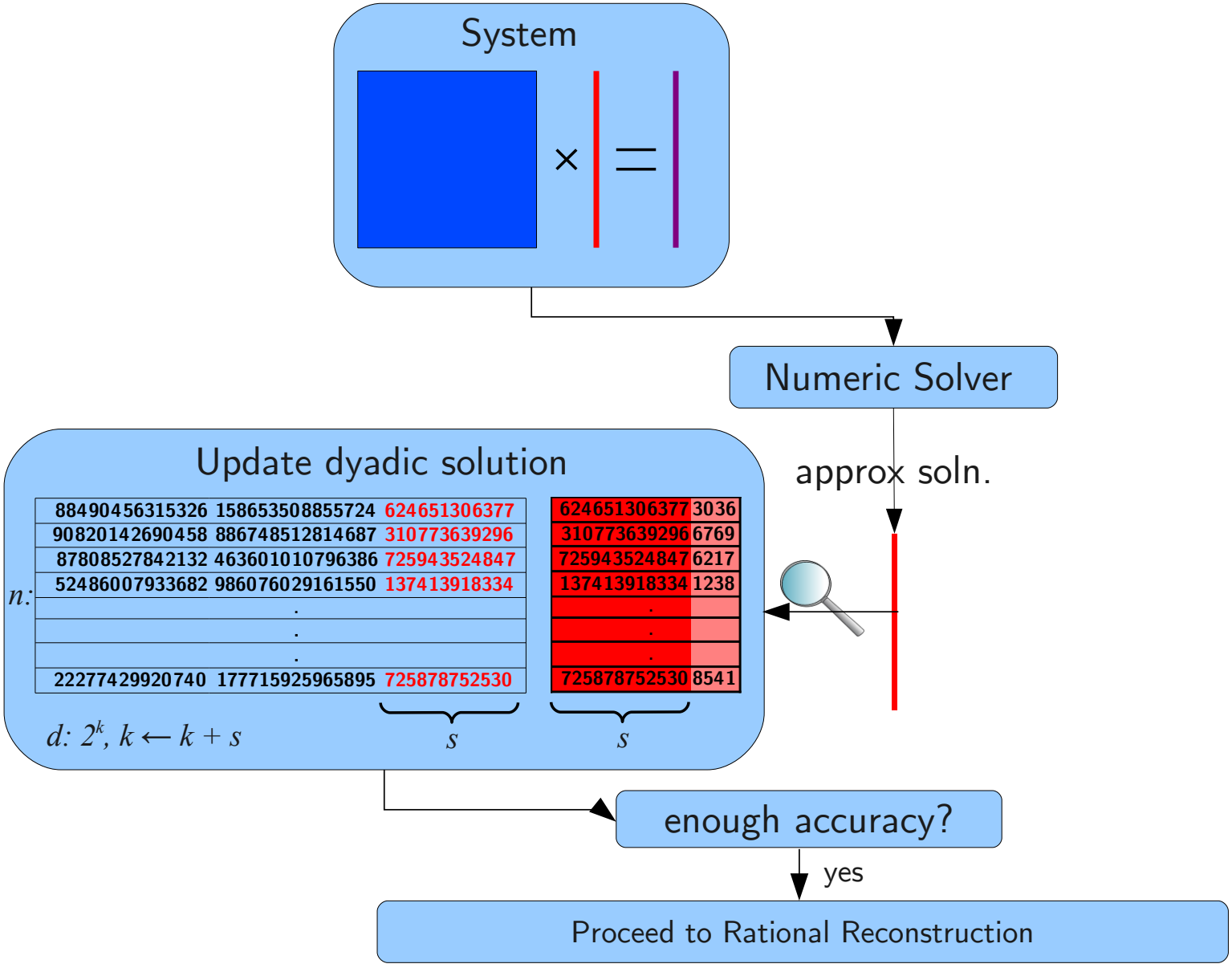
# Iterative Refinement



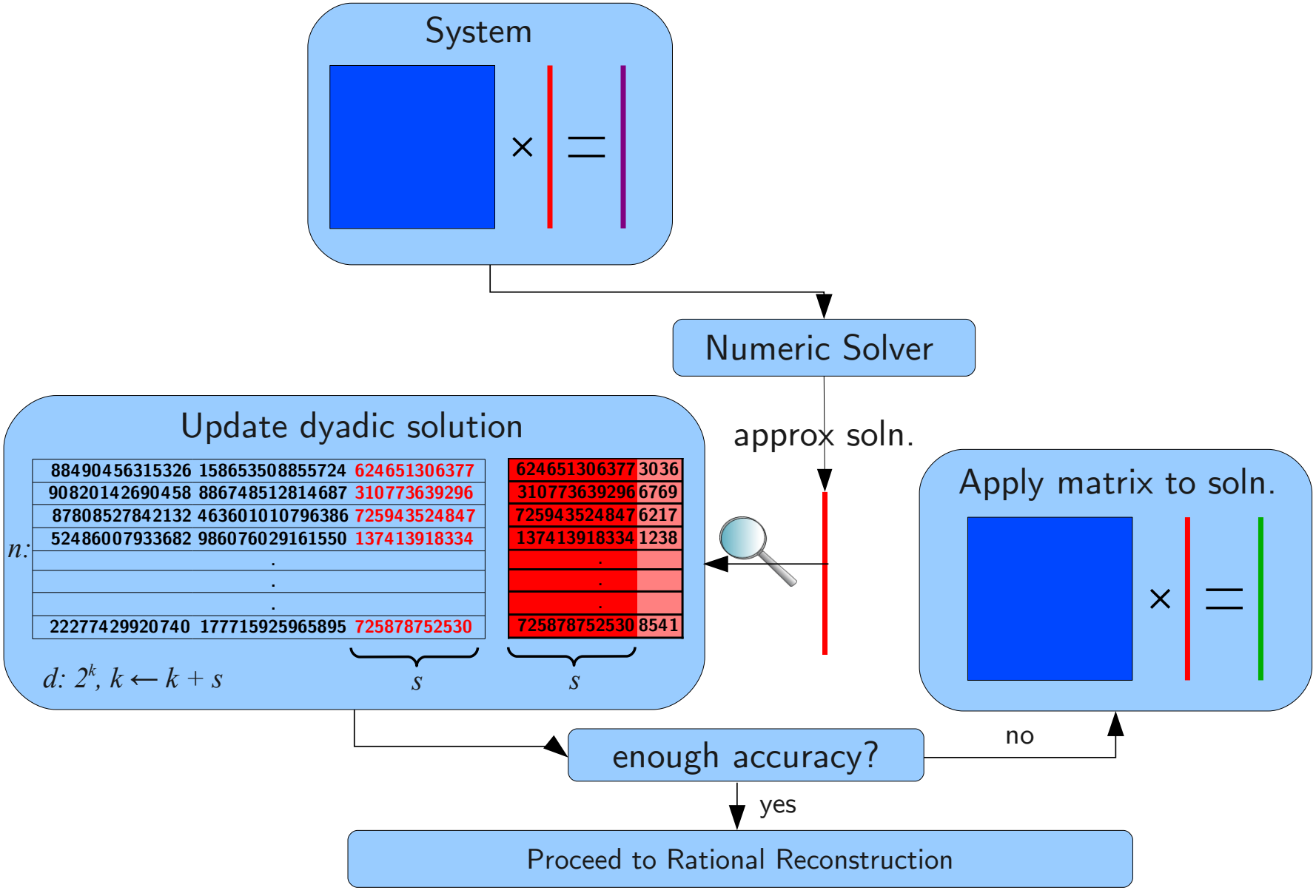
# Iterative Refinement



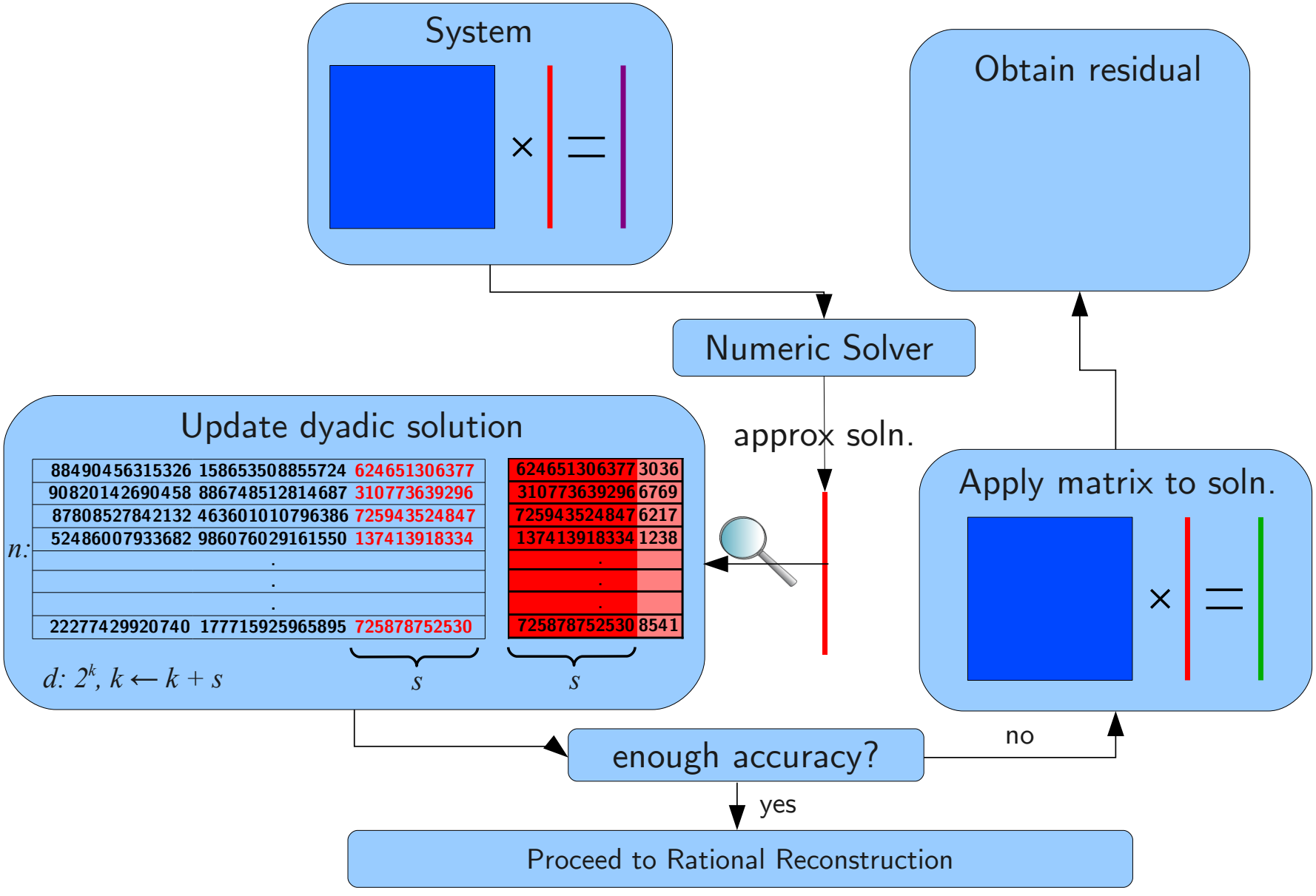
# Iterative Refinement



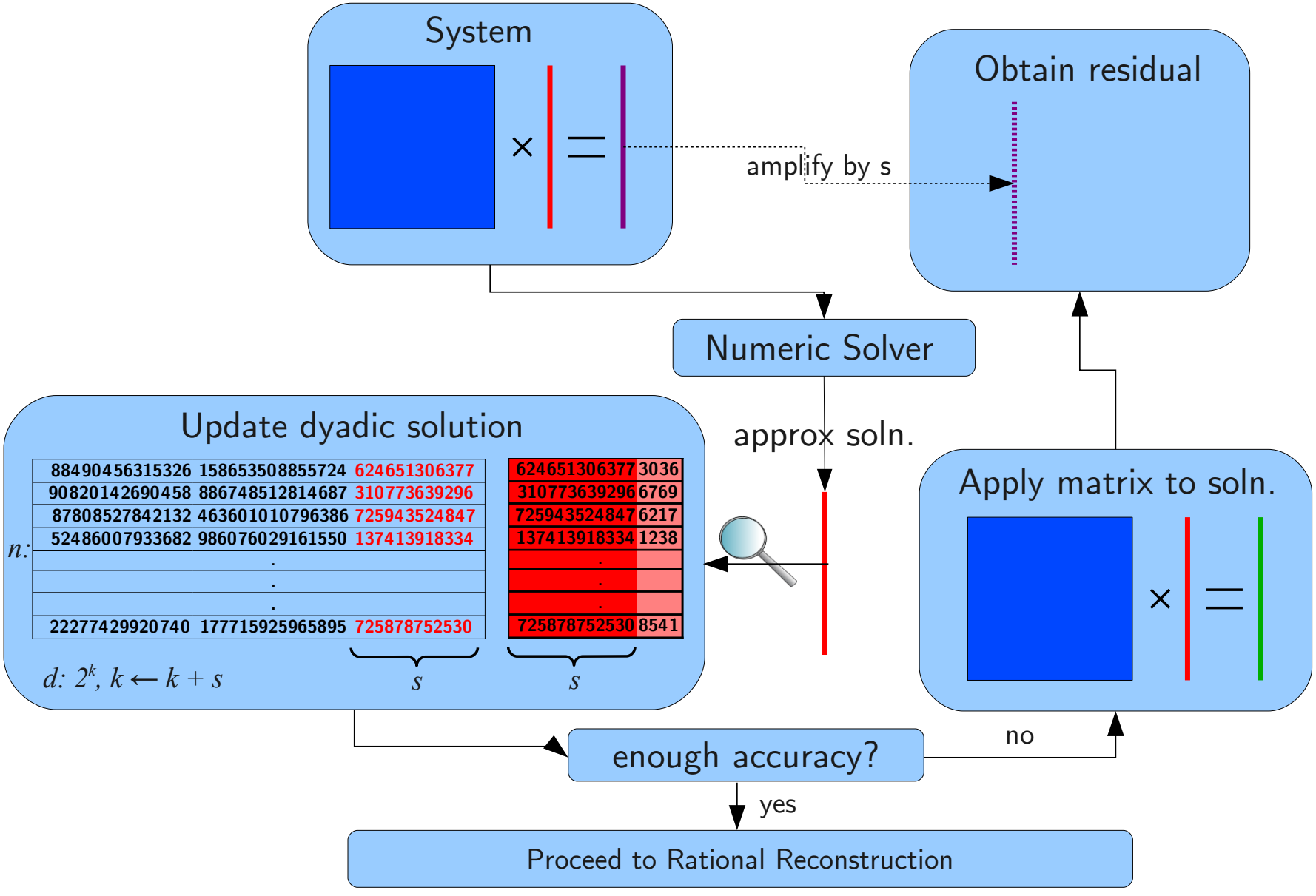
# Iterative Refinement



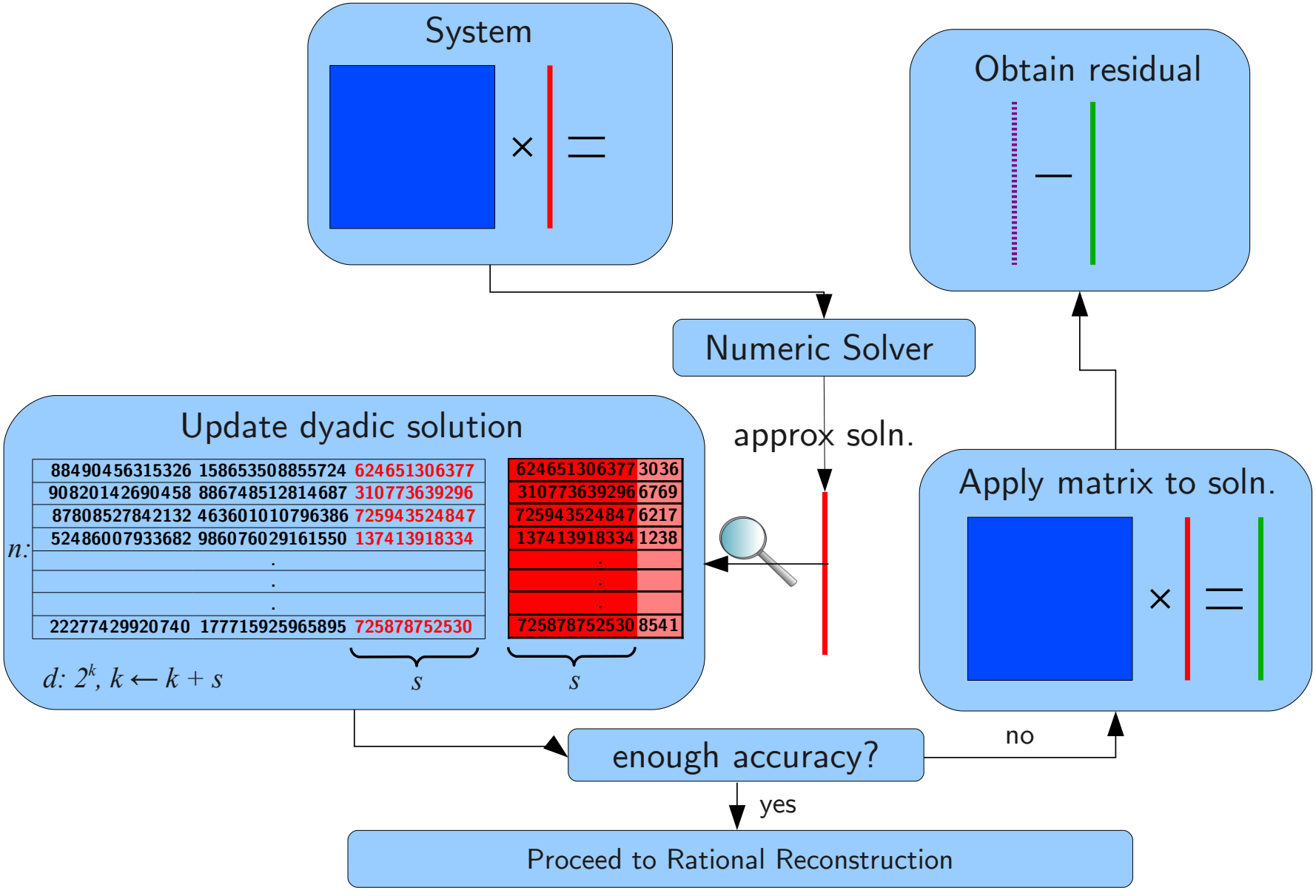
# Iterative Refinement



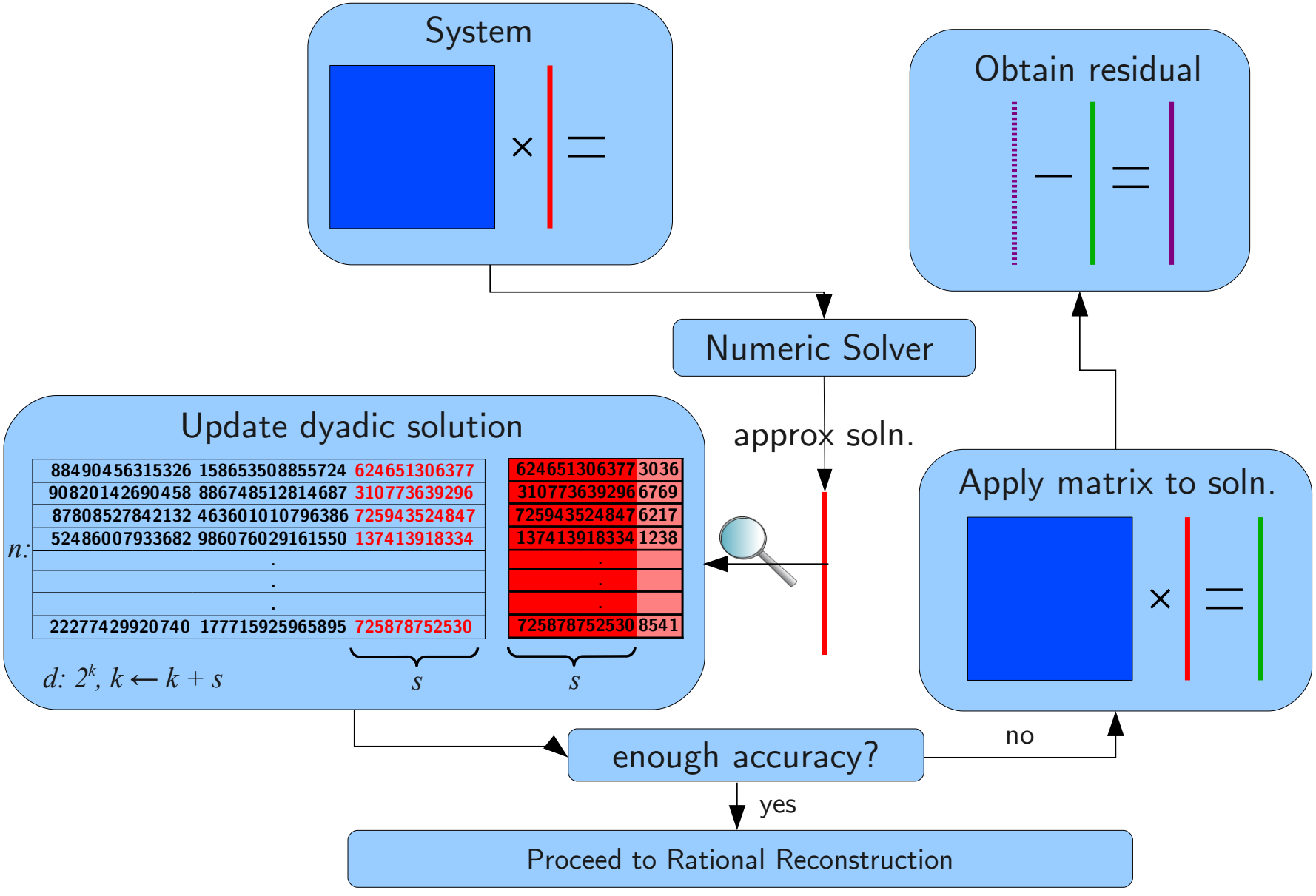
# Iterative Refinement



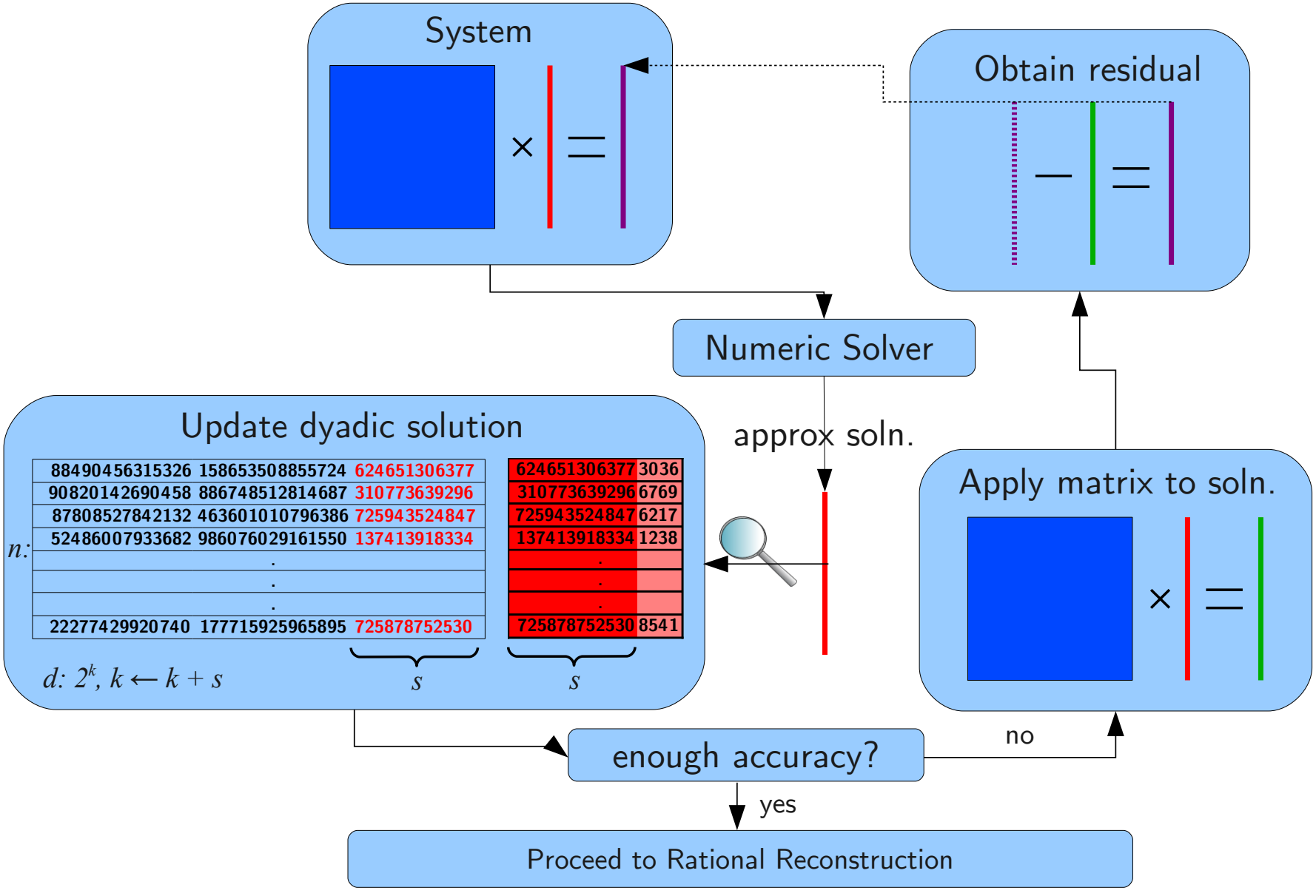
# Iterative Refinement



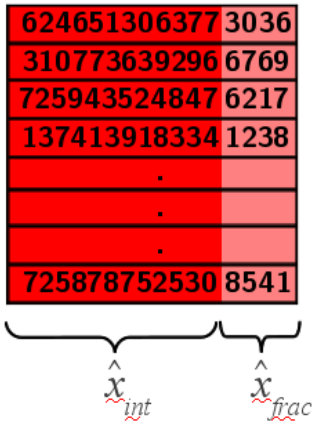
# Iterative Refinement



# Iterative Refinement



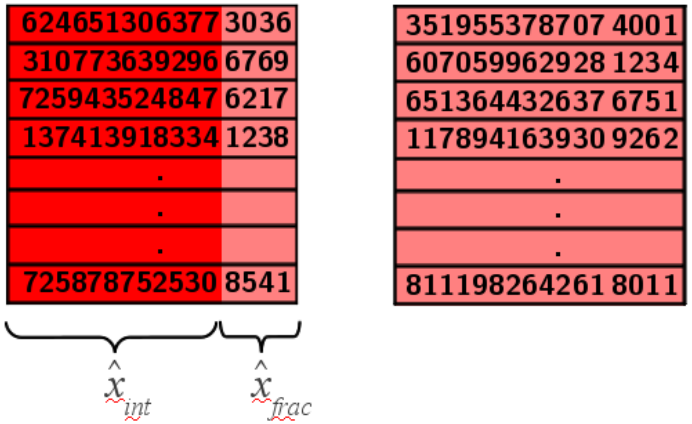
# Confirmed Continuation (Overlap)




At each iteration, we scale our partial solution $\hat{x}$, then split into:

- $\hat{x}_{int} = \lfloor \hat{x} \cdot 2^s \rfloor$
- $\hat{x}_{frac} = \hat{x} \cdot 2^s - \hat{x}_{int}$

# Confirmed Continuation (Overlap)

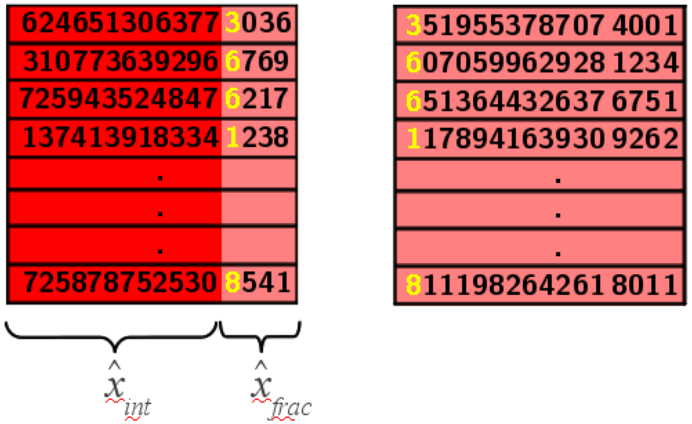



At each iteration, we scale our partial solution $\hat{x}$, then split into:

- $\hat{x}_{int} = \lfloor \hat{x} \cdot 2^s \rfloor$
- $\hat{x}_{frac} = \hat{x} \cdot 2^s - \hat{x}_{int}$

Before updating dyadic approximants, we *confirm continuation* of the iteration by verifying **overlap** between the current $\hat{x}'$ and the previous $\hat{x}_{frac}$.

# Confirmed Continuation (Overlap)




At each iteration, we scale our partial solution $\hat{x}$, then split into:

- $\hat{x}_{int} = \lfloor \hat{x} \cdot 2^s \rfloor$
- $\hat{x}_{frac} = \hat{x} \cdot 2^s - \hat{x}_{int}$

Before updating dyadic approximants, we *confirm continuation* of the iteration by verifying **overlap** between the current $\hat{x}'$ and the previous $\hat{x}_{frac}$.

- $\max |\hat{x}' - \hat{x}_{frac}| \leq \frac{1}{2^b}$ ensures $b$ bits of overlap
- One bit is (typically) sufficient.

# Rational Reconstruction

Like rational reconstruction from $p$-adic digits, Euclidean remainder sequence used to recover exact solution from dyadic approximants.

# Rational Reconstruction

Like rational reconstruction from $p$-adic digits, Euclidean remainder sequence used to recover exact solution from dyadic approximants.

- P-adically a remainder sequence entry is $(r_i, s_i, t_i)$, where $r_i = s_i x + t_i p^k$, the remainder $r_i$ is a *numerator candidate* and coefficient $s_i$ of the given residue $x$ is a denominator candidate.

# Rational Reconstruction

Like rational reconstruction from $p$-adic digits, Euclidean remainder sequence used to recover exact solution from dyadic approximants.

- P-adically a remainder sequence entry is $(r_i, s_i, t_i)$, where $r_i = s_i x + t_i p^k$, the remainder $r_i$ is a *numerator candidate* and coefficient $s_i$ of the given residue $x$ is a denominator candidate.
- Dyadically a remainder sequence entry is $(r_i, s_i, t_i)$, where $r_i = s_i x + t_i 2^k$, the remainder $r_i$ is a *measure of error* and coefficient $s_i$ of the given dyadic approximation $x$ is a denominator candidate.

# Rational Reconstruction - single element example

**Example**

Solving $Ax = b$, suppose:

- $x_1 = 9/17$
- the Hadamard bound for $\det(A)$ is $h = 2^8 = 256$.

In general we need $2\lg(h) = 16$ bits of approximation.

# Rational Reconstruction - single element example

**Example**

Solving $Ax = b$, suppose:

- $x_1 = 9/17$
- the Hadamard bound for $\det(A)$ is $h = 2^8 = 256$.

In general we need $2\lg(h) = 16$ bits of approximation.

Dyadic approximation to 16 bits: $\frac{n}{d} = \frac{34696}{65536} \approx \frac{a}{b}$

Stopping condition: $\left|\frac{a}{b} - \frac{n}{d}\right| \leq \frac{1}{2d}$

# Rational Reconstruction - single element example

**Example**

Solving $Ax = b$, suppose:

- $x_1 = 9/17$
- the Hadamard bound for $\det(A)$ is $h = 2^8 = 256$.

In general we need $2\lg(h) = 16$ bits of approximation.

Dyadic approximation to 16 bits: $\frac{n}{d} = \frac{34696}{65536} \approx \frac{a}{b}$

Stopping condition: $\left|\frac{a}{b} - \frac{n}{d}\right| \leq \frac{1}{2d} \rightarrow |bn - ad| \leq \frac{b}{2}$

# Rational Reconstruction - single element example

## Example

Solving $Ax = b$, suppose:

- $x_1 = 9/17$
- the Hadamard bound for $\det(A)$ is $h = 2^8 = 256$.

In general we need $2\lg(h) = 16$ bits of approximation.

Dyadic approximation to 16 bits: $\frac{n}{d} = \frac{34696}{65536} \approx \frac{a}{b}$
Stopping condition: $\left|\frac{a}{b} - \frac{n}{d}\right| \leq \frac{1}{2d} \rightarrow |bn - ad| \leq \frac{b}{2}$

$$65536 = 0 \cdot n + 1 \cdot d$$
$$34696 = 1 \cdot n + 0 \cdot d$$
$$30840 = -1 \cdot n + 1 \cdot d$$
$$3856 = 2 \cdot n + -1 \cdot d$$
$$3848 = -15 \cdot n + 8 \cdot d$$
$$8 = 17 \cdot n + -9 \cdot d$$

# Output sensitivity (early termination)

Dyadic approximation to 12 bits: $\frac{n}{d} = \frac{2168}{4096} \approx \frac{a}{b}$

$$4096 = 0 \cdot n + 1 \cdot d$$
$$2168 = 1 \cdot n + 0 \cdot d$$
$$1928 = -1 \cdot n + 1 \cdot d$$
$$240 = 2 \cdot n + -1 \cdot d$$
$$8 = -17 \cdot n + 9 \cdot d$$

- No random choice of prime, therefore no probabilistic early termination.
- But *guaranteed* early termination in some cases.
- Idea explored in-depth by Steffy [2010]

# Vector rational reconstruction

Suppose solution to Ax = b is (9/17, 3/11, 7/187) and we have computed the approximation to 12 bits: $\frac{2168,1117,153}{4096}$

# Vector rational reconstruction

Suppose solution to Ax = b is (9/17, 3/11, 7/187) and we have computed the approximation to 12 bits: $\frac{2168,1117,153}{4096}$

- $x_1$: As we saw, 9/17 is found with certainty.
  17 = lcm of denominators so far.

# Vector rational reconstruction

Suppose solution to Ax = b is (9/17, 3/11, 7/187) and we have computed the approximation to 12 bits: $\frac{2168,1117,153}{4096}$

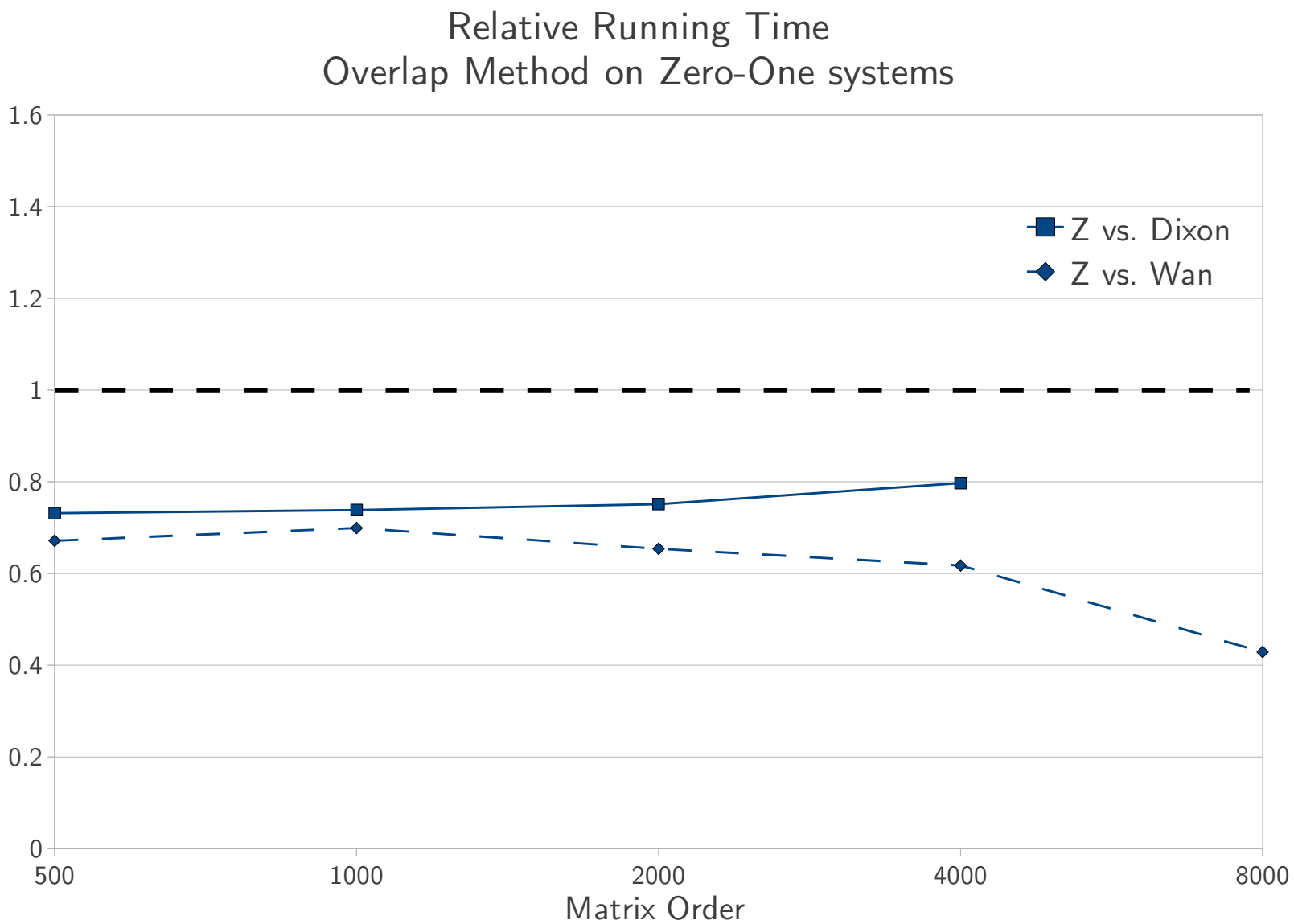
- $x_1$: As we saw, 9/17 is found with certainty.
  17 = lcm of denominators so far.
- $x_2$: Try division once- failure ($-1491 = 1117 \cdot 17 - 5 \cdot 4096$).
  Find 3/11 with single element reconstruction.
  187 = lcm of denominators so far.

# Vector rational reconstruction

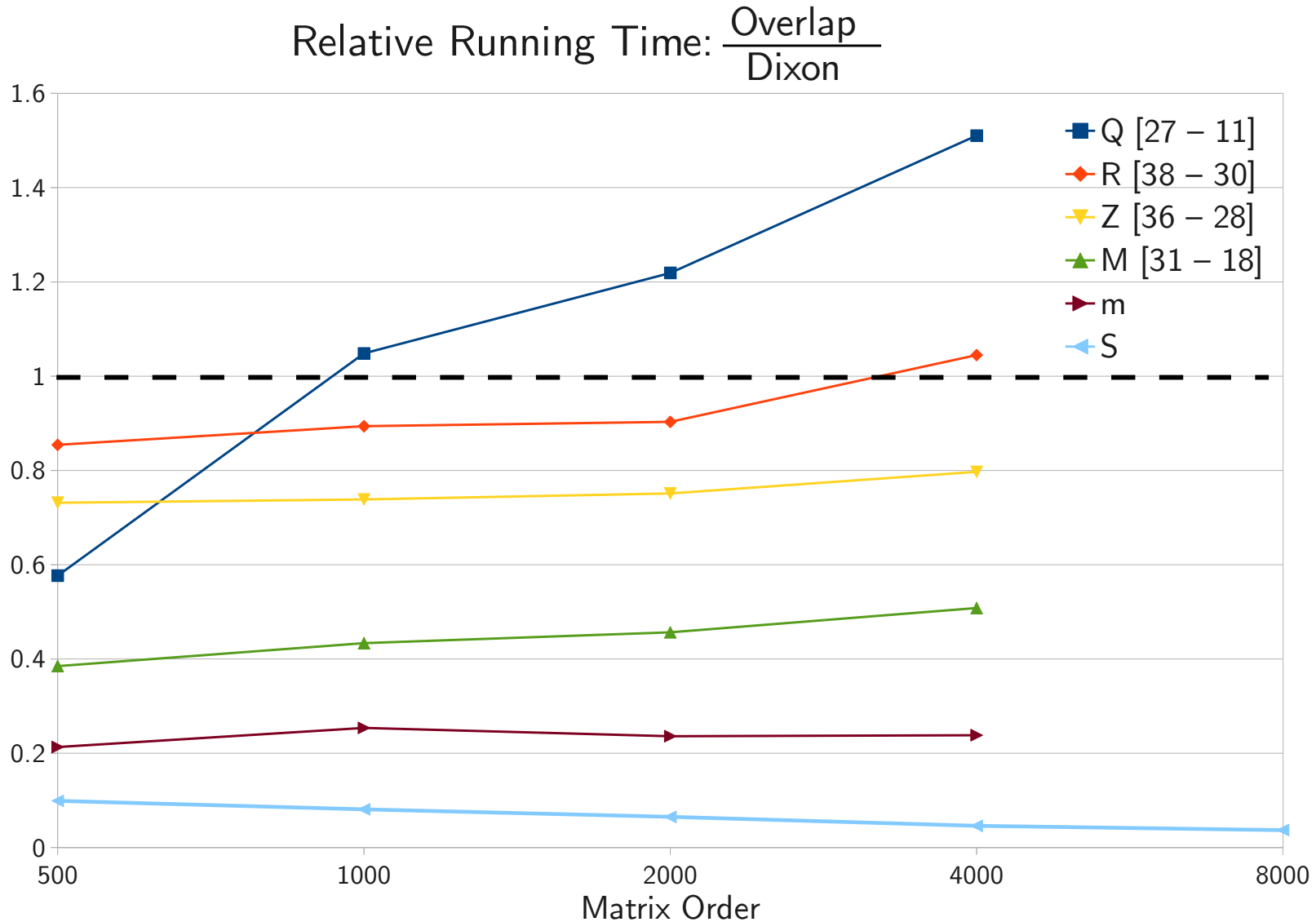
Suppose solution to Ax = b is (9/17, 3/11, 7/187) and we have computed the approximation to 12 bits: $\frac{2168,1117,153}{4096}$

- $x_1$: As we saw, 9/17 is found with certainty.
  17 = lcm of denominators so far.
- $x_2$: Try division once- failure $(-1491 = 1117 \cdot 17 - 5 \cdot 4096)$.
  Find 3/11 with single element reconstruction.
  187 = lcm of denominators so far.
- $x_3$: Try division once- success! $(-61 = 153 \cdot 187 - 7 \cdot 4096)$.
  7/187 found with small enough remainder $(61 \leq \frac{187}{2})$.

# Performance



# Performance



# Conclusion and Ongoing Work

Overlap method results in consistently faster, more robust performance over algorithm predecessor.

# Conclusion and Ongoing Work

Overlap method results in consistently faster, more robust performance over algorithm predecessor.

Output sensitive early termination is a proven avenue for runtime savings.

# Conclusion and Ongoing Work

Overlap method results in consistently faster, more robust performance over algorithm predecessor.

Output sensitive early termination is a proven avenue for runtime savings.

Specialized numeric solvers fit easily into the framework.

# Conclusion and Ongoing Work

Overlap method results in consistently faster, more robust performance over algorithm predecessor.

Output sensitive early termination is a proven avenue for runtime savings.

Specialized numeric solvers fit easily into the framework.

Work ongoing to incorporate highly-tuned direct (SuperLU) and iterative sparse solvers.

The End