Precise Solutions for $\min_x \|Ax - b\|_2$

James Demmel, UC Berkeley
Yozo Hida, UC Berkeley
Xiaoye Li, Lawrence Berkeley Labs
Jason Riedy, UC Berkeley
Meghana Vishvanath, UC Berkeley
David Vu, UC Berkeley

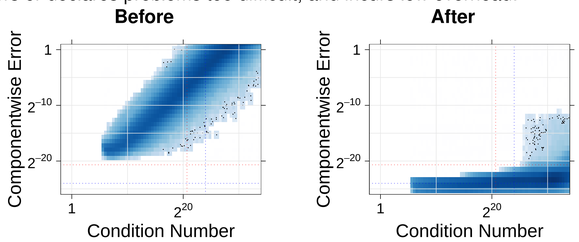
1. From Faster Computers to Better Answers

- Use faster processors and more memory to **improve accuracy**.
- We can trade a small amount of performance for results that are:
  - **Easy to understand** and **dependable**.
  - The **forward error is small** or the **problem is too difficult**.
  - Correctness both from some theory and **extensive experiments**.
- Compute a factorization and initial solution with hardware arithmetic.
- Refine the solution with some extra precision arithmetic.
- Smaller problems: often fast enough; add moderate overhead.
- Larger problems harness asymptotics: $O(mn^2)$ solve, $O(mn)$ refine.
- Relatively easy to parallelize; typical dense linear algebra.

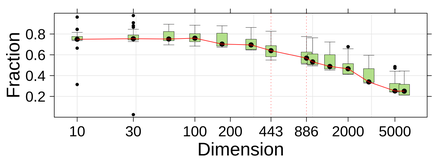
2. Prior Success for $Ax = b$

Applying extra precision within iterative refinement for $Ax = b$ produces small errors or declares problems too difficult, and incurs low overhead:

Before **After**



Fraction of time in refinement



- Factor in **hardware**. (Double)
- Refine in **software**. (Doubled-double)
- Includes condition estimation.

Vertical dotted lines: 2 matrices and 1 matrix fit in cache

- Accepted solutions are correct to $O(\varepsilon)$ **componentwise**.
- Reject solutions when the system is too ill-conditioned.

3. Differences between $Ax = b$ and $\min_x \|Ax - b\|_2$

- More reasonable error metrics.
- Require the **termination state** in addition to the **condition number** for detecting difficult problems.
  - $Ax = b$: All well-enough conditioned systems converged.
  - $\min_x \|Ax - b\|_2$: All condition numbers depend on the possibly inaccurate computed solution.
- Cannot avoid many forms of ill-conditioning.
  - Consider conditioning of **two** parts (x,r), not one (x).
  - No row equilibration to reduce ill-scaling.
  - If κ(augmented system) $\approx \kappa(A) \approx 1/\varepsilon$, A is practically rank deficient.
- Refinement requires **more work**!
  - More steps of refinement are necessary: worst case 50 v. 5
  - More work per step.
  - The ratio between factorization and refinement work is n, often $\ll m$.

4. Goals for Overdetermined Least-Squares

Solve with error $\le \sqrt{m+n} \cdot \varepsilon$ or declare too difficult.

- Two solution parts: the model coefficients **x** and the residual **r**.
- Two error metrics: **normwise** and **componentwise**

	Normwise	Componentwise
Solution x	$\frac{\|x - x_{\text{truth}}\|_\infty}{\|x_{\text{truth}}\|_\infty}$	$\max_i \left\|\frac{x(i) - x_{\text{truth}}(i)}{x_{\text{truth}}(i)}\right\|$
Residual r	$\frac{\|r - r_{\text{truth}}\|_\infty}{\|b\|_\infty}$	$\max_i \left\|\frac{r(i) - r_{\text{truth}}(i)}{r_{\text{truth}}(i)}\right\|$

- Residual r's compared to right-hand side b normwise.
  - Measures system's consistency.
- Each part and metric has an associated **specific condition number**.
  - Condition number too large ⇒ problem needs more precision throughout.
  - All condition numbers depend on **computed** x, r as well as A, b.

5. Iterative Refinement for Least-Squares

Apply Newton's method to the augmented linear system:

$$\begin{bmatrix} \alpha I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\alpha} r \\ x \end{bmatrix} = B_\alpha \begin{bmatrix} r_s \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

(*c.f.* Björck and Golub)

1. Factor $A = QR$ in working precision ε. $O(mn^2)$
2. Estimate parameter $\alpha \approx \frac{\|A\|_\infty}{\kappa_\infty(R)\sqrt{2}}$, so that $\kappa(B_\alpha) \approx \kappa(A)$. $O(mn + n^2)$
3. Using $A = QR$ and working precision ε, solve
$$\begin{bmatrix} \alpha I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r_s \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$ $O(mn)$
4. For $i = 1$ to max. iterations,
   (a) In extended precision $\varepsilon_r \le \varepsilon^2$, calculate
   $$\begin{bmatrix} e_r \\ e_x \end{bmatrix} \leftarrow \begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} \alpha I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r_s \\ x \end{bmatrix}.$$ $O(mn)$
   (b) Using $A = QR$ and working precision ε, solve
   $$\begin{bmatrix} \alpha I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} dr_s \\ dx \end{bmatrix} = \begin{bmatrix} e_r \\ e_x \end{bmatrix}.$$ $O(mn)$
   (c) **Check termination for x, r_s normwise and componentwise.** See below for details. $O(m+n)$
   (d) Accumulate, possibly keeping x, r_s in precision $\varepsilon_x < \varepsilon^2$,
   $$\begin{bmatrix} r_s \\ x \end{bmatrix} \leftarrow \begin{bmatrix} r_s \\ x \end{bmatrix} + \begin{bmatrix} dr_s \\ dx \end{bmatrix}.$$ $O(m+n)$
5. Return x, $r = \alpha r_s$, and error estimates.

6. When Does Refinement Stop?

- **Terminate** when nothing is **working**.
- Each part (x and r) and metric (normwise and componentwise) can be in one of four states. The mnemonic dv/v is defined below.
  - **Unstable**: The relative step size is too large ($dv/v > 0.5$). (only componentwise metrics may be considered unstable)
  - **Working**: Actively making progress.
  - **No Progress**: Step size does not decrease enough (new $dv/v \ge 0.7$ old dv/v).
  - **Converged**: The relative step size is tiny ($dv/v \le \varepsilon$).

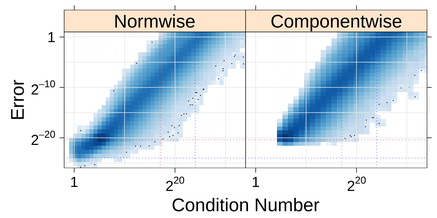
	Normwise	Componentwise
Solution x	$dv/v = \|dx\|_\infty / \|x\|_\infty$	$dv/v = \max_j \left\|dx_j/x_j\right\|$
Residual r	$dv/v = \|dr\|_\infty / \|b\|_\infty$	$dv/v = \max_j \left\|dr_j/r_j\right\|$

7. Results

Refinement converges & not too ill-conditioned ⇒ precise solutions!

Before: Solve with single-precision QR

Error in x ∝ condition number.

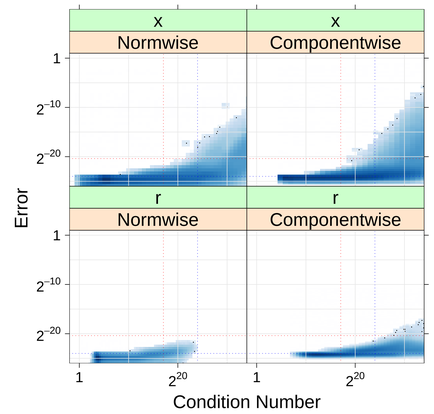


Test cases:
- Test one million single-precision 100×50 systems.
- Generate randomly conditioned A, random angle θ.
- Target $\kappa_2(A) \le 2^{26}$, θ in $[2^{-25}, 1] \cdot \pi/2$.
- Generate random b with $\|A^T b\|_2 = \|b\|_2 \cdot \cos\theta$.
- Solve for x, r with double precision.

After: Solve with single-precision QR and double-precision refinement

Error $\le \sqrt{m+n} \cdot \varepsilon$ if converged and condition number $< 1/(10\sqrt{m+n} \cdot \varepsilon)$.

Converged cases
Solution accepted if left of red line.



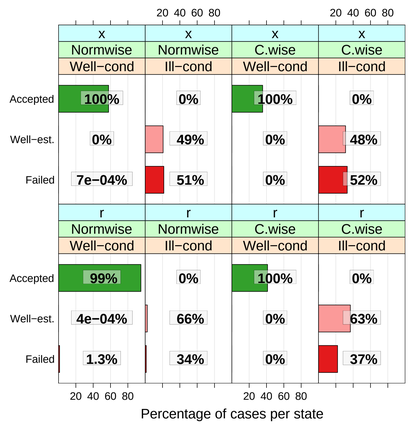
Accepted solutions:
- Have relative error $\le \sqrt{m+n} \cdot \varepsilon$.
- Refinement has converged.
- Part and metric well-*enough*-conditioned.
  - Need an extra $10\times$ safety factor over $Ax = b$.

Well-estimated solutions:
- Relative error within factor of 10 of returned estimate.
- Otherwise ill-conditioned and/or not converged.
- **Not a guarantee!** Accept at user's own risk.

Failed solutions:
- Cannot trust error or estimate.
- Indistinguishable from well-estimated solutions.

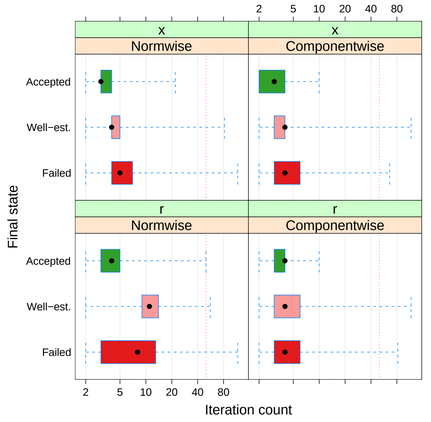
Dotted red lines at thresholds $\sqrt{m+n} \cdot \varepsilon$ and $1/(10\sqrt{m+n} \cdot \varepsilon)$. Dotted blue lines at ε and $1/\varepsilon$.

Acceptance, Estimation, and Failure



Labels: % within a box.
Length: % both conds by part & norm.

Iteration count



Dot at median; box encloses 50% of samples.
Dotted line: max count (50) for accepted solutions.

8th Bay Area Scientific Computing Day
Stanford 50
29-31 March 2007

L A P A C K
L -A P -A C -K
L A P A -C -K
L -A P -A -C K
L A -P -A C K
L -A -P A C -K