Practical Alternatives for Parallel Pivoting

Jason Riedy and James Demmel
UC Berkeley

June 20, 2003

DOE and NSF

1. Outline

- Direct method scalability
  - Target applications
  - Issues with dynamic pivoting
- Practical pivoting alternatives
  - Coping with dynamic pivoting
  - Static pivoting ideas
- Static pivoting results
  - Matching style
  - Perturbation size
  - Number and location of perturbations
- Structure-limited pivoting

2. Target Applications

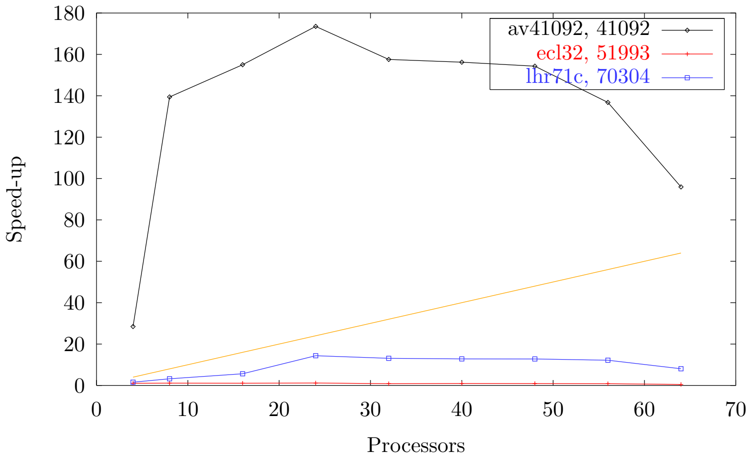
Direct solution of sparse, unsymmetric linear systems through LU factorization.

- Factor **many related** matrices
  - inverse problems, fluid flow, optimization, eigenvalues
  - Values change, structure doesn't
- Factor **large** matrices
  - 8 million columns (supercomputer-sized)

3. Issues with Dynamic Pivoting

Parallel numerical work scales pretty well.
Analysis work is unpredictable.

- Scalability: A larger problem requires proportionally more resources for same speed.
- Users and computer facilities want predictability.
- Parallel analysis work (new) is discrete and not predictable...



Amortize analysis work over many numerical runs.

4. Issues with Dynamic Pivoting

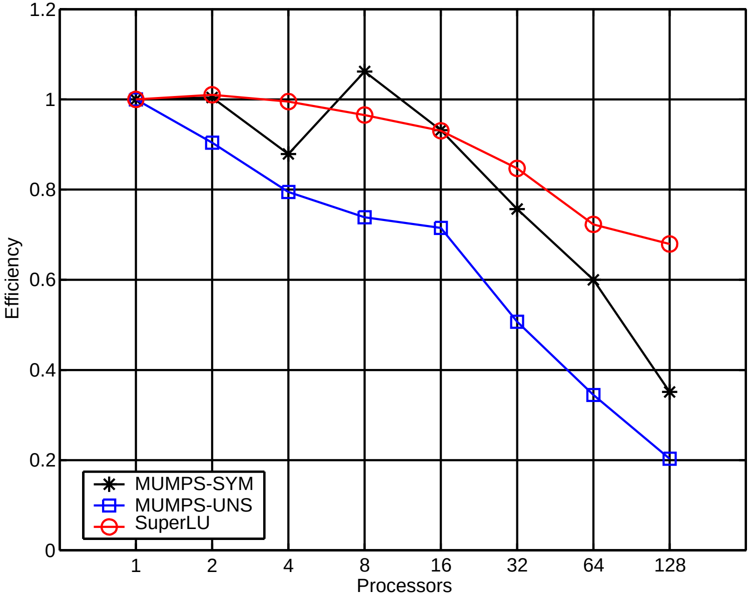
Dynamic pivoting can change the explicit entry structure at every step.

Changing the entry structure also changes:

- Column dependencies
  - Wrecks parallel scheduling.
- Data structures
  - Imposes communication and memory overheads.
- Computation and communication patterns
  - Causes load imbalancing.

Want to *decouple* analysis and numeric work.

5. SuperLU v. MUMPS



Amestoy, Duff, L'Excellent, and Li

- "Perfect" example: 3-D grid with nested dissection
- MUMPS achieves higher MFLOP/s until 128 processors
- SuperLU uses 2-D distribution and static pivoting

6. Practical Pivoting Alternatives

Something needs to control element growth.

- Coping with dynamic pivots:
  - Assume all pivoting is satisfied within the front, delaying unavailable pivots
  - Include all (heuristically) possible pivots in structure
- Avoiding structure-changing pivoting altogether:
  - Static pivoting through matchings
  - Structure-limited pivoting
  - Iterative methods rather than refinement

7. Coping with Dynamic Pivots

Dynamic pivots require dynamic response or over-estimation.

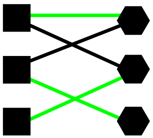
- Do everything dynamically
  - MUMPS: Duff, Amestoy, *et al*. (CERFACS)
  - Scheduling, load-balancing, *etc*. must be dynamic.
  - Assume pivots are local, delay those that fail
- Consider all possible pivots
  - WSMP: Gupta (IBM)
  - Data dependencies include all (heuristically) reasonable pivots
  - Frontal matrices constructed dynamically

8. Static Pivoting

Few entries should interact during sparse factorization, so large elements shouldn't change much.

- Pre-pivot with a matching [Olshowka, Neumaier] [Duff, Koster]
- Matrix A gives bipartite graph $G(A) = \{\mathcal{R}, \mathcal{C}; \mathcal{E}\}$
- Find a maximum weight matching on $G(A)$
  - Matching corresponds to a permutation M
  - Also provides a particular row scaling S_r and column scaling S_c

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{12} & 0 & a_{23} \\ 0 & a_{32} & 0 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Factor $LU = P_c^T M^T S_r A S_c P_c = P_c^T M^T A_s P_c$
- No dynamic structural changes.

9. Static Pivoting with Perturbations

Tiny pivots still occur from cancellation. But this kind of cancellation implies round-off...

- Perturb small diagonal entries [Li and Demmel]
  - Similar ideas for symmetric indefinite [Gill, Murray, Wright], [Eskow, Schnabel], [Cheng, Higham]
- Tiny pivots encourage drastic element growth.
- Drastic element growth leads to unintended cancellation.
- Change tiny pivots to some larger number.

$$P_c^T M^T A_s P_c = \begin{bmatrix} L_{11} & 0 & 0 \\ \lambda_{21} & 1 & 0 \\ L_{31} & 0 & I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & \pi_{22} & s_{23} \\ 0 & s_{32} & S_{33} \end{bmatrix} \begin{bmatrix} U_{11} & \nu_{12} & U_{13} \\ 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix}$$

10. Static Pivoting with Perturbations

Tiny pivots still occur from cancellation. But this kind of cancellation implies round-off...

- Perturb small diagonal entries [Li and Demmel]
  - Similar ideas for symmetric indefinite [Gill, Murray, Wright], [Eskow, Schnabel], [Cheng, Higham]
- Tiny pivots encourage drastic element growth.
- Drastic element growth leads to unintended cancellation.
- Change tiny pivots to some larger number.

$$\begin{bmatrix} L_{11} & 0 & 0 \\ \lambda_{21} & 1 & 0 \\ L_{31} & 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 & 0 \\ 0 & \pi_{22} & s_{23} \\ 0 & s_{32} & S_{33} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} U_{11} & \nu_{12} & U_{13} \\ 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix}$$

11. Static Pivoting with Perturbations

Tiny pivots still occur from cancellation. But this kind of cancellation implies round-off...

- Perturb small diagonal entries [Li and Demmel]
  - Similar ideas for symmetric indefinite [Gill, Murray, Wright], [Eskow, Schnabel], [Cheng, Higham]
- Tiny pivots encourage drastic element growth.
- Drastic element growth leads to unintended cancellation.
- Change tiny pivots to some larger number.

$$\begin{bmatrix} L_{11} & 0 & 0 \\ \lambda_{21} & 1 & 0 \\ L_{31} & 0 & I \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & \pi_{22} & s_{23} \\ 0 & s_{32} & S_{33} \end{bmatrix} \begin{bmatrix} U_{11} & \nu_{12} & U_{13} \\ 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

12. Static Pivoting with Perturbations

Tiny pivots still occur from cancellation. But this kind of cancellation implies round-off...

- Perturb small diagonal entries [Li and Demmel]
  - Similar ideas for symmetric indefinite [Gill, Murray, Wright], [Eskow, Schnabel], [Cheng, Higham]
- Tiny pivots encourage drastic element growth.
- Drastic element growth leads to unintended cancellation.
- Change tiny pivots to some larger number.

$$P_c^T M^T A_s P_c = LU + D$$

- D is diagonal with only a few small diagonal entries.

13. Static Pivoting Parameters

One downside: There are many more parameters and tuning choices.

- What scaling to use?
  - Matching-based, or typical equilibration
- What type of matching? What are the weights of $G(A)$?
  - Pure structural or value-based
- How large a perturbation?
  - SuperLU: $\sqrt{\varepsilon}\|A_s\|_1$
    - $\|A_s\|_1$: "weight" of column
    - $\sqrt{\varepsilon}$: half-precision perturbation
- How do we improve the solution?
  - Iterative refinement or an iterative method (GMRES)

Determine through experiments
(MatrixMarket and UF collections)

14. Scaling

Cheap is good.

- Computing Olshowka and Neumaier's scaling is expensive.
- Has no effect beyond (very cheap) equilibration.
- True for both static and dynamic pivoting.

	No Equilibration	Equilibration
No O&N scaling	no improvement	fixes scaling
O&N scaling	fixes scaling	no incremental improvement

15. Pattern-based Matching

We can't get away from using the values.

Pattern-based:

- Maximum cardinality matching
  - All edges have weight one.
  - Quickly encounter element blow-up.
  - Solves fail. Factors are far off the mark.
- Minimize the Markowitz cost
  - Edge weight $= n^2 - (r-1)(c-1)$, the worst case fill from the given pattern.
  - Most solves *still* fail. Same problems.

16. Value-based Matching

We can't get away from using the values.

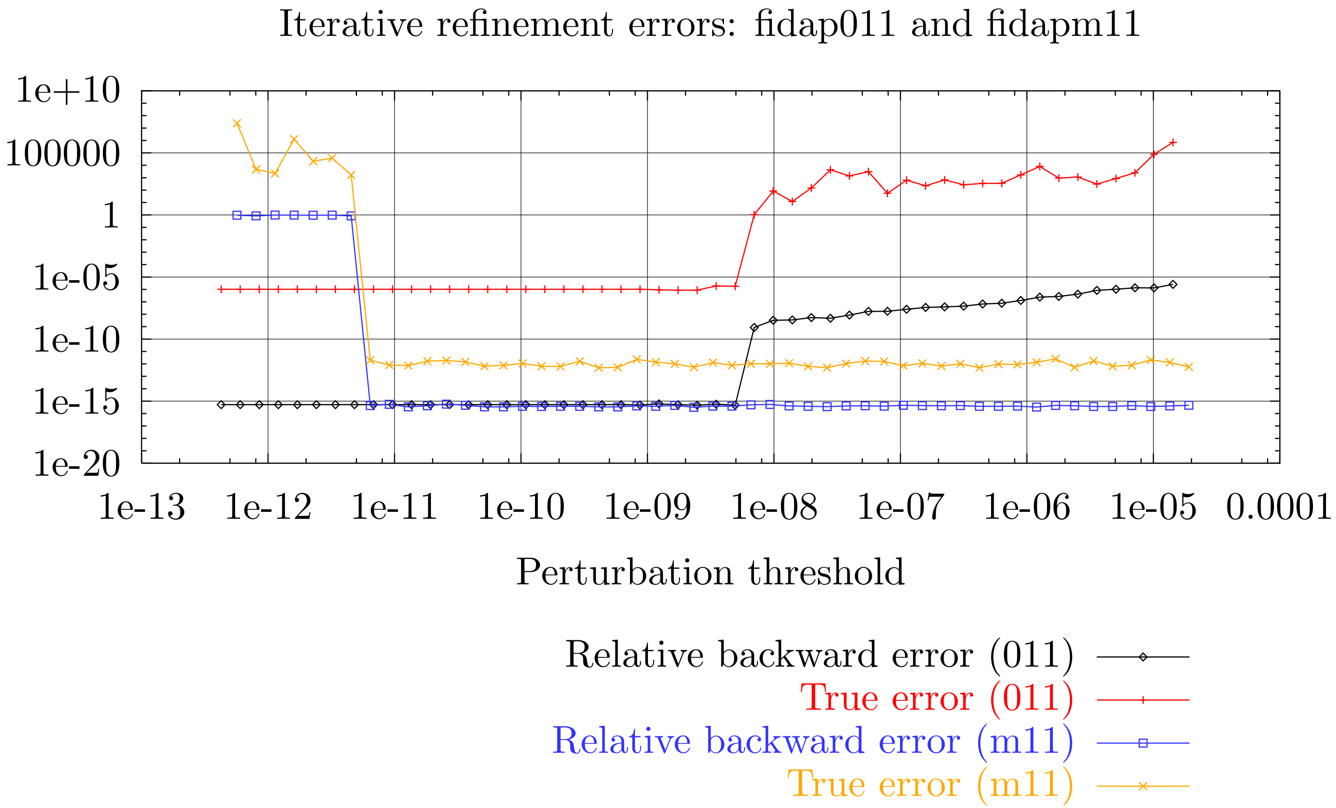
Value-based:

- Maximize the least entry
  - Relatively expensive; sequence of matchings
  - Works sometimes.
- Maximize the diagonal's sum
  - Edge (i, j) has weight $|a_{ij}|$
- Maximize the diagonal's product
  - Edge (i, j) has weight $\log |a_{ij}|$

The latter matchings with refinement fail to converge for few matrices (5-8 / 50). Maximizing the product requires fewer perturbations, and using integer exponents is equivalent.

17. Perturbation Size

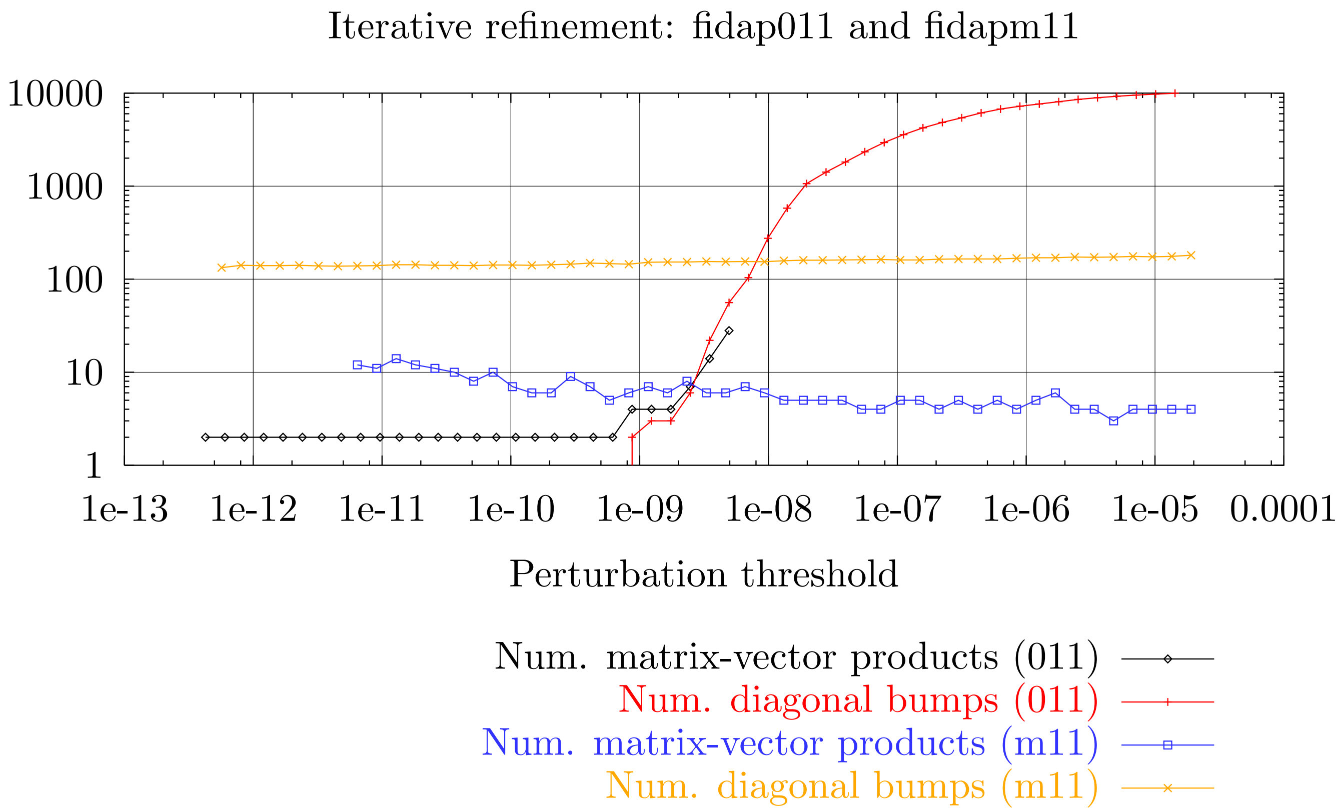
Size matters. Smaller is better.



rcondest(fidapm11) $\approx 10^{-5}$, rcondest(fidap011) $\approx 10^{-12}$

18. Number of Perturbations

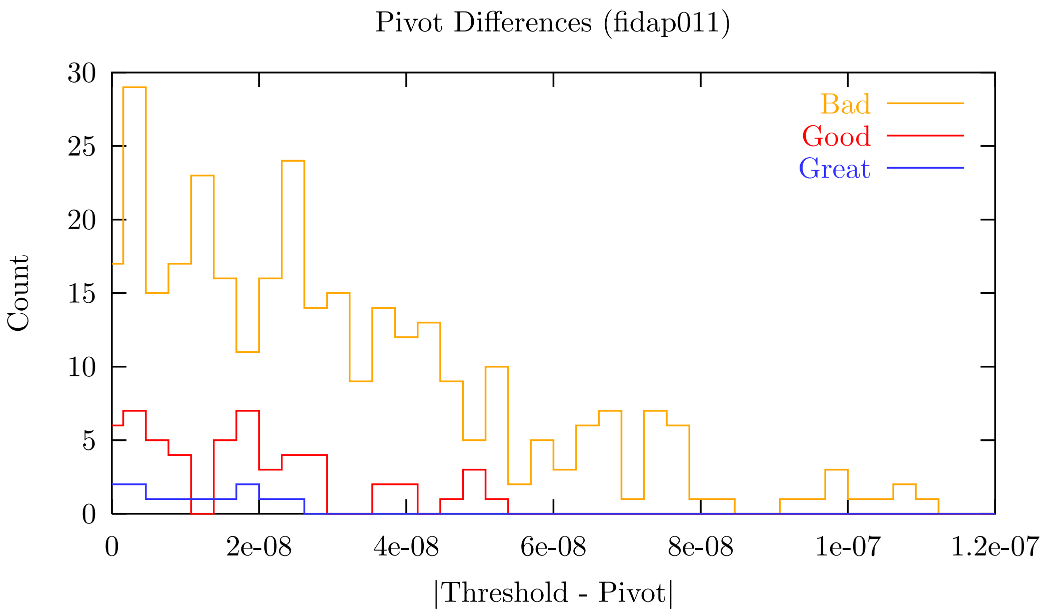
The number matters, sometimes.



Number of supernodes with pivots grows about half as quickly.

19. Perturbation Size: fidap011

The failing case changed *many* pivots by a tiny amount.



Pick thresholds around a critial point. rcondest(fidap011) $\approx 10^{-12}$
Factored approx. 100 levels before reaching perturbations.

20. Other Matrices

- fidapm11
  - Matching perfomed “poorly”.
  - Two large entries forced many tiny ones onto the diagonal.
- Hopeless cases
  - Really large condition numbers (lhr71c) fail.
- Complex: works?
  - Don't have many examples.
  - Use magnitude for matching.

21. Static Pivoting's Drawbacks

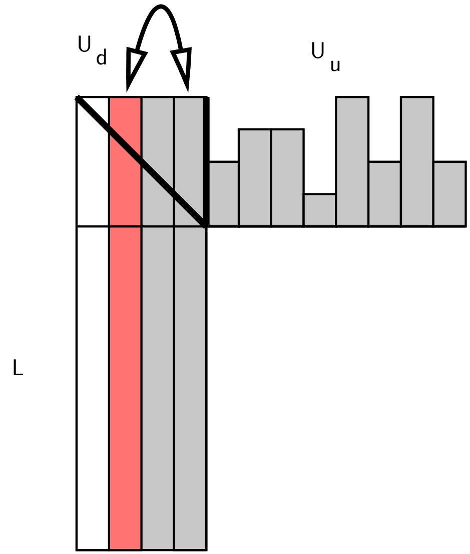
Is this still a direct method?

- When iterative refinement fails, use a general iterative method.
  - LU factors of a low-rank perturbation, so GMRES
  - Refinement depends on $(A + D)^{-1} D \ldots$
  - Pivot changes of order rcond...
- Lose backwards stability?
- Lose predictability?
- Promising direction: Merge iterative and direct methods.
  - Can we control the perturbations enough to prove things about the combination in *floating-point* arithmetic?

22. Structure-Limited Pivoting

Can we exploit block structure for pivoting?

- Perturbations in $< 30\%$ of a supernode's columns.

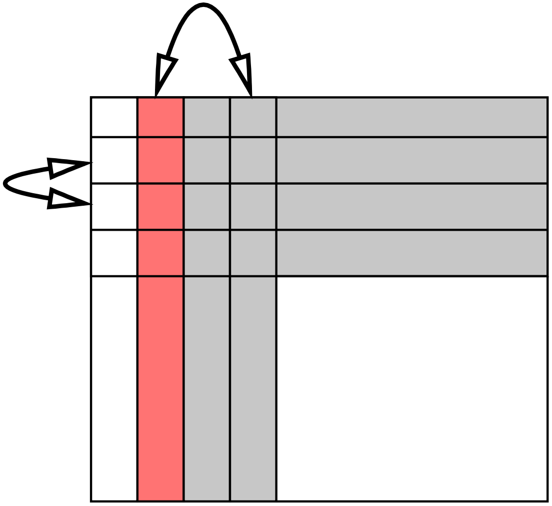


- Factor $P_c^T M^T A P_c Q = LU = L(U_d + U_u Q)$
- Don't need to modify U_u storage or indexing.

23. Structure-Limited Pivoting

Can we exploit block structure for pivoting?

- Row supernodes or frontal matrices can use limited row pivoting.



- Factor $Q_r P_c^T M^T A P_c = LU = (Q_r L_l + L_d)U$
- Could even use limted complete pivoting, *etc*.

24. Observations

- Static pivoting is practical and promising.
  - Separates numerical and analysis work.
  - Can calculate static pivots in parallel.
  - Needs work to keep all the benefits of a *direct* method
- Structure-limited pivoting is also promising.
  - Takes more communication.
  - Should preserve directness.