The blind-spot of dynamically regularized factorizations

Jack Poulson

Department of Mathematics
Stanford University

Linear Algebra and Optimization Seminar
Stanford University
February 26, 2015

Context

- Had just finished preliminary tests for distributed (C++11 over MPI) sparse primal-dual IPM for

$$\min_x \frac{1}{2}x^T Q x + c^T x$$
$$\text{s.t. } Ax = b, Gx + s = h, s \geq 0$$

  when I signed up for this talk.
- Had tested synthetic BP, BPDN, CP, DS, EN, LAV, NNLS, SVM, and TV with great success
- Decided that solving $\min_x \|Ax - b\|_\infty$ meant that I should be able to solve

$$Ax = b$$

  for nonsymmetric sparse A.
- Then I ran into a fundamental instability in a widely-used technique
- This talk is focused on this instability

Context

- Had just finished preliminary tests for distributed (C++11 over MPI) sparse primal-dual IPM for

$$\min_x \frac{1}{2}x^T Qx + c^T x$$
$$\text{s.t. } Ax = b, Gx + s = h, s \geq 0$$

  when I signed up for this talk.
- Had tested synthetic BP, BPDN, CP, DS, EN, LAV, NNLS, SVM, and TV with great success
- Decided that solving $\min_x \|Ax - b\|_\infty$ meant that I should be able to solve

$$Ax = b$$

  for nonsymmetric sparse A.
- Then I ran into a fundamental instability in a widely-used technique
- This talk is focused on this instability

Context

- Had just finished preliminary tests for distributed (C++11 over MPI) sparse primal-dual IPM for

$$\min_x \frac{1}{2} x^T Q x + c^T x$$
$$\text{s.t. } Ax = b, Gx + s = h, s \geq 0$$

  when I signed up for this talk.
- Had tested synthetic BP, BPDN, CP, DS, EN, LAV, NNLS, SVM, and TV with great success
- Decided that solving $\min_x \|Ax - b\|_\infty$ meant that I should be able to solve

$$Ax = b$$

  for nonsymmetric sparse A.
- Then I ran into a fundamental instability in a widely-used technique
- This talk is focused on this instability

Context

- Had just finished preliminary tests for distributed (C++11 over MPI) sparse primal-dual IPM for

$$\min_x \frac{1}{2}x^T Qx + c^T x$$
$$\text{s.t. } Ax = b, Gx + s = h, s \geq 0$$

  when I signed up for this talk.
- Had tested synthetic BP, BPDN, CP, DS, EN, LAV, NNLS, SVM, and TV with great success
- Decided that solving $\min_x \|Ax - b\|_\infty$ meant that I should be able to solve

$$Ax = b$$

  for nonsymmetric sparse A.
- Then I ran into a fundamental instability in a widely-used technique
- This talk is focused on this instability

Context

- Had just finished preliminary tests for distributed (C++11 over MPI) sparse primal-dual IPM for

$$\min_x \frac{1}{2} x^T Q x + c^T x$$
$$\text{s.t. } Ax = b, Gx + s = h, s \geq 0$$

when I signed up for this talk.
- Had tested synthetic BP, BPDN, CP, DS, EN, LAV, NNLS, SVM, and TV with great success
- Decided that solving $\min_x \|Ax - b\|_\infty$ meant that I should be able to solve

$$Ax = b$$

for nonsymmetric sparse A.
- Then I ran into a fundamental instability in a widely-used technique
- This talk is focused on this instability

Dynamic regularization

- Sparse Cholesky can be vastly less expensive than Bunch-Kaufman (same for LU w/ and w/o pivoting)
- In a distributed-memory context, dynamic pivoting prevents a priori load balancing
- Forming $A^H A$ or AA^H should be avoided for both stability and sparsity reasons
- Many sparse-direct solvers (e.g., `Pardiso`, `SuperLU_Dist`, and `WSMP`) support dynamically regularized pivots
- Each diagonal modification implies a small **and** rank-one perturbation of the original problem, and so, with exact arithmetic, we could expect a great preconditioner
- Interior Point solvers (e.g., [Altman/Gondzio-1998]) often intelligently choose the sign of the perturbation of quasi-semidefinite matrix [Vanderbei-1993]

Dynamic regularization

- Sparse Cholesky can be vastly less expensive than Bunch-Kaufman (same for LU w/ and w/o pivoting)
- In a distributed-memory context, dynamic pivoting prevents a priori load balancing
- Forming $A^H A$ or AA^H should be avoided for both stability and sparsity reasons
- Many sparse-direct solvers (e.g., `Pardiso`, `SuperLU_Dist`, and `WSMP`) support dynamically regularized pivots
- Each diagonal modification implies a small **and** rank-one perturbation of the original problem, and so, with exact arithmetic, we could expect a great preconditioner
- Interior Point solvers (e.g., [Altman/Gondzio-1998]) often intelligently choose the sign of the perturbation of quasi-semidefinite matrix [Vanderbei-1993]

Dynamic regularization

- Sparse Cholesky can be vastly less expensive than Bunch-Kaufman (same for LU w/ and w/o pivoting)
- In a distributed-memory context, dynamic pivoting prevents a priori load balancing
- Forming $A^H A$ or AA^H should be avoided for both stability and sparsity reasons
- Many sparse-direct solvers (e.g., `Pardiso`, `SuperLU_Dist`, and `WSMP`) support dynamically regularized pivots
- Each diagonal modification implies a small **and** rank-one perturbation of the original problem, and so, with exact arithmetic, we could expect a great preconditioner
- Interior Point solvers (e.g., [Altman/Gondzio-1998]) often intelligently choose the sign of the perturbation of quasi-semidefinite matrix [Vanderbei-1993]

Dynamic regularization

- Sparse Cholesky can be vastly less expensive than Bunch-Kaufman (same for LU w/ and w/o pivoting)
- In a distributed-memory context, dynamic pivoting prevents a priori load balancing
- Forming $A^H A$ or AA^H should be avoided for both stability and sparsity reasons
- Many sparse-direct solvers (e.g., `Pardiso`, `SuperLU_Dist`, and `WSMP`) support dynamically regularized pivots
- Each diagonal modification implies a small **and** rank-one perturbation of the original problem, and so, with exact arithmetic, we could expect a great preconditioner
- Interior Point solvers (e.g., [Altman/Gondzio-1998]) often intelligently choose the sign of the perturbation of quasi-semidefinite matrix [Vanderbei-1993]

Dynamic regularization

- Sparse Cholesky can be vastly less expensive than Bunch-Kaufman (same for LU w/ and w/o pivoting)
- In a distributed-memory context, dynamic pivoting prevents a priori load balancing
- Forming $A^H A$ or AA^H should be avoided for both stability and sparsity reasons
- Many sparse-direct solvers (e.g., `Pardiso`, `SuperLU_Dist`, and `WSMP`) support dynamically regularized pivots
- Each diagonal modification implies a small **and** rank-one perturbation of the original problem, and so, with exact arithmetic, we could expect a great preconditioner
- Interior Point solvers (e.g., [Altman/Gondzio-1998]) often intelligently choose the sign of the perturbation of quasi-semidefinite matrix [Vanderbei-1993]

Dynamic regularization

- Sparse Cholesky can be vastly less expensive than Bunch-Kaufman (same for LU w/ and w/o pivoting)
- In a distributed-memory context, dynamic pivoting prevents a priori load balancing
- Forming $A^H A$ or AA^H should be avoided for both stability and sparsity reasons
- Many sparse-direct solvers (e.g., `Pardiso`, `SuperLU_Dist`, and `WSMP`) support dynamically regularized pivots
- Each diagonal modification implies a small **and** rank-one perturbation of the original problem, and so, with exact arithmetic, we could expect a great preconditioner
- Interior Point solvers (e.g., [Altman/Gondzio-1998]) often intelligently choose the sign of the perturbation of quasi-semidefinite matrix [Vanderbei-1993]

The irrelevance of pivot magnitudes

Suppose someone handed you

$$A = \begin{pmatrix} 1 & 2 & & & & \\ 2 & 5 & 2 & & & \\ & 2 & 5 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & & 5 & 2 \\ & & & & 2 & 5 \end{pmatrix} = \begin{pmatrix} 1 & & & & & \\ 2 & 1 & & & & \\ & 2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & \\ & & & & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & & & & \\ & 1 & 2 & & & \\ & & 1 & \ddots & & \\ & & & \ddots & & \\ & & & & 1 & 2 \\ & & & & & 1 \end{pmatrix}$$

and asked you to solve $Ax = b$.

Despite

$$\operatorname{diag}(L) = \operatorname{ones}(n, 1), \quad \|L\|_{\max} = 2,$$

for even $n = 100$, the residual will tend to be $O(1)$.

Because A is extremely ill-conditioned, there is no contradiction with the good backwards stability implied by $|A|$ and $|L||L^H|$.

While the $\lambda(L) = \{1\}$, the ϵ-pseudospectrum is a large disk containing the origin

And forward substitution can lead to 2^{n-1} element growth...

The irrelevance of pivot magnitudes

Suppose someone handed you

$$A = \begin{pmatrix} 1 & 2 & & & & \\ 2 & 5 & 2 & & & \\ & 2 & 5 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & & 5 & 2 \\ & & & & 2 & 5 \end{pmatrix} = \begin{pmatrix} 1 & & & & & \\ 2 & 1 & & & & \\ & 2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & \\ & & & & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & & & & \\ & 1 & 2 & & & \\ & & 1 & \ddots & & \\ & & & \ddots & & \\ & & & & 1 & 2 \\ & & & & & 1 \end{pmatrix}$$

and asked you to solve $Ax = b$.

Despite

$$\mathrm{diag}(L) = \mathrm{ones}(n, 1), \quad \|L\|_{\max} = 2,$$

for even $n = 100$, the residual will tend to be $O(1)$.

Because A is extremely ill-conditioned, there is no contradiction with the good backwards stability implied by $|A|$ and $|L||L^H|$.

While the $\lambda(L) = \{1\}$, the ϵ-pseudospectrum is a large disk containing the origin

And forward substitution can lead to 2^{n-1} element growth...

The irrelevance of pivot magnitudes

Suppose someone handed you

$$A = \begin{pmatrix} 1 & 2 & & & & \\ 2 & 5 & 2 & & & \\ & 2 & 5 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & & 5 & 2 \\ & & & & 2 & 5 \end{pmatrix} = \begin{pmatrix} 1 & & & & & \\ 2 & 1 & & & & \\ & 2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & \\ & & & & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & & & & \\ & 1 & 2 & & & \\ & & 1 & \ddots & & \\ & & & \ddots & & \\ & & & & 1 & 2 \\ & & & & & 1 \end{pmatrix}$$

and asked you to solve $Ax = b$.

Despite

$$\operatorname{diag}(L) = \operatorname{ones}(n, 1), \quad \|L\|_{\max} = 2,$$

for even $n = 100$, the residual will tend to be $O(1)$.

Because A is extremely ill-conditioned, there is no contradiction with the good backwards stability implied by $|A|$ and $|L||L^H|$.

While the $\lambda(L) = \{1\}$, the ϵ-pseudospectrum is a large disk containing the origin

And forward substitution can lead to 2^{n-1} element growth...

The irrelevance of pivot magnitudes

Suppose someone handed you

$$A = \begin{pmatrix} 1 & 2 & & & & \\ 2 & 5 & 2 & & & \\ & 2 & 5 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & & 5 & 2 \\ & & & & 2 & 5 \end{pmatrix} = \begin{pmatrix} 1 & & & & & \\ 2 & 1 & & & & \\ & 2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & \\ & & & & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & & & & \\ & 1 & 2 & & & \\ & & 1 & \ddots & & \\ & & & \ddots & & \\ & & & & 1 & 2 \\ & & & & & 1 \end{pmatrix}$$

and asked you to solve $Ax = b$.

Despite

$$\text{diag}(L) = \text{ones}(n, 1), \quad \|L\|_{\max} = 2,$$

for even $n = 100$, the residual will tend to be $O(1)$.

Because A is extremely ill-conditioned, there is no contradiction with the good backwards stability implied by $|A|$ and $|L||L^H|$.

While the $\lambda(L) = \{1\}$, the ϵ-pseudospectrum is a large disk containing the origin

And forward substitution can lead to 2^{n-1} element growth...

The irrelevance of pivot magnitudes

Suppose someone handed you

$$A = \begin{pmatrix} 1 & 2 & & & & \\ 2 & 5 & 2 & & & \\ & 2 & 5 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & & 5 & 2 \\ & & & & 2 & 5 \end{pmatrix} = \begin{pmatrix} 1 & & & & & \\ 2 & 1 & & & & \\ & 2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & & 1 & \\ & & & & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & & & & \\ & 1 & 2 & & & \\ & & 1 & \ddots & & \\ & & & \ddots & & \\ & & & & 1 & 2 \\ & & & & & 1 \end{pmatrix}$$

and asked you to solve $Ax = b$.

Despite

$$\mathrm{diag}(L) = \mathrm{ones}(n, 1), \quad \|L\|_{\max} = 2,$$

for even $n = 100$, the residual will tend to be $O(1)$.

Because A is extremely ill-conditioned, there is no contradiction with the good backwards stability implied by $|A|$ and $|L||L^H|$.

While the $\lambda(L) = \{1\}$, the ϵ-pseudospectrum is a large disk containing the origin

And forward substitution can lead to 2^{n-1} element growth...

A well-conditioned extension

Consider

$$K = \begin{pmatrix} 4J_{1/2}(n)^T J_{1/2}(n) & I \\ I & -I \end{pmatrix},$$

which, for $n = 100$, has $\mathrm{cond}(K) = 14.719$

Then LDL^T factorization without pivoting will fail catastrophically before invoking regularization

K is an example of a well-conditioned quasi-definite matrix where dynamic regularization dramatically fails.

A well-conditioned extension

Consider

$$K = \begin{pmatrix} 4J_{1/2}(n)^T J_{1/2}(n) & I \\ I & -I \end{pmatrix},$$

which, for $n = 100$, has $\mathrm{cond}(K) = 14.719$

Then LDL^T factorization without pivoting will fail catastrophically before invoking regularization

K is an example of a well-conditioned quasi-definite matrix where dynamic regularization dramatically fails.

A well-conditioned extension

Consider

$$K = \begin{pmatrix} 4J_{1/2}(n)^T J_{1/2}(n) & I \\ I & -I \end{pmatrix},$$

which, for $n = 100$, has $\mathrm{cond}(K) = 14.719$

Then LDL^T factorization without pivoting will fail catastrophically before invoking regularization

K is an example of a well-conditioned quasi-definite matrix where dynamic regularization dramatically fails.

The stability of quasi-definite LDL^H factorization

Suppose

$$K = \begin{pmatrix} G & A \\ A^H & -H \end{pmatrix}, \quad G, H \succ 0.$$

[Gill et al.-1996] showed that

$$\text{Econd(K)} = \left(1 + \frac{\max\{\|A^H G^{-1} A\|_2, \|A H^{-1} A^H\|_2\}}{\|K\|_2}\right) \text{cond}(K),$$

which allows for the failure of the previous example since $\|G^{-1}\|_2$ was very large.

The stability of quasi-definite LDL^H factorization

Suppose

$$K = \begin{pmatrix} G & A \\ A^H & -H \end{pmatrix}, \quad G, H \succ 0.$$

[Gill et al.-1996] showed that

$$\text{Econd(K)} = \left(1 + \frac{\max\{\|A^H G^{-1} A\|_2, \|A H^{-1} A^H\|_2\}}{\|K\|_2}\right) \text{cond}(K),$$

which allows for the failure of the previous example since $\|G^{-1}\|_2$ was very large.

Preconditioning with regularization

[Saunders-1995,1996] suggests instead using a factorization of the matrix

$$K_{\delta,\delta} \equiv \begin{pmatrix} G + \delta I & A \\ A^H & -H - \delta I \end{pmatrix}$$

as a preconditioner, where $\delta = 10^{-4}$ is typical.

He showed that, for $\sigma_{\min}(A) < \delta < \|A\|_2$, $G = H = 0$,

$$\text{Econd}(K_{\delta,\delta}) \approx \left(\frac{\|A\|_2}{\delta}\right)^2 \approx \text{cond}(A^H A + \delta^2 I),$$

which might often be 10^8. When A has a dense row, the former is often much preferred to $A^H A + \delta^2 I$.

Elemental's IPMs now default to $\delta = 10^{-4}$ and use the iteratively refined solution against $K_{\delta,\delta}$ as a preconditioner for FGMRES(10).

Preconditioning with regularization

[Saunders-1995,1996] suggests instead using a factorization of the matrix

$$K_{\delta,\delta} \equiv \begin{pmatrix} G + \delta I & A \\ A^H & -H - \delta I \end{pmatrix}$$

as a preconditioner, where $\delta = 10^{-4}$ is typical.

He showed that, for $\sigma_{\min}(A) < \delta < \|A\|_2$, $G = H = 0$,

$$\mathrm{Econd}(K_{\delta,\delta}) \approx \left(\frac{\|A\|_2}{\delta}\right)^2 \approx \mathrm{cond}(A^H A + \delta^2 I),$$

which might often be 10^8. When A has a dense row, the former is often much preferred to $A^H A + \delta^2 I$.

Elemental's IPMs now default to $\delta = 10^{-4}$ and use the iteratively refined solution against $K_{\delta,\delta}$ as a preconditioner for FGMRES(10).

Preconditioning with regularization

[Saunders-1995,1996] suggests instead using a factorization of the matrix

$$K_{\delta,\delta} \equiv \begin{pmatrix} G+\delta I & A \\ A^H & -H-\delta I \end{pmatrix}$$

as a preconditioner, where $\delta = 10^{-4}$ is typical.

He showed that, for $\sigma_{\min}(A) < \delta < \|A\|_2$, $G = H = 0$,

$$\text{Econd}(K_{\delta,\delta}) \approx \left(\frac{\|A\|_2}{\delta}\right)^2 \approx \text{cond}(A^H A + \delta^2 I),$$

which might often be 10^8. When A has a dense row, the former is often much preferred to $A^H A + \delta^2 I$.

Elemental's IPMs now default to $\delta = 10^{-4}$ and use the iteratively refined solution against $K_{\delta,\delta}$ as a preconditioner for FGMRES(10).

The history of augmented systems for least squares

There is a long history [Siegel-1965,Björck-1967,Hatchel-1974,Björck-1992] of

$$\min_x \|Ax - b\|_2 \Rightarrow A^H(Ax - b) = 0$$

being represented as

$$K_\alpha \begin{pmatrix} s \\ x \end{pmatrix} \equiv \begin{pmatrix} \alpha I & A \\ A^H & 0 \end{pmatrix} \begin{pmatrix} r/\alpha \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

- cond(K_α) roughly varies between cond(A) and cond$(A)^2$
- [Björck-1967,1992] showed $\alpha = \sigma_{\min}(A)$ is quasi-optimal for both $[s; x]$ and x via straight-forward argument from eigenpairs of K_α
- Used by MATLAB until mid-1990's [Matstoms-1994]; discarded because of expense of pivoted LU

The history of augmented systems for least squares

There is a long history [Siegel-1965,Björck-1967,Hatchel-1974,Björck-1992] of

$$\min_x \|Ax - b\|_2 \Rightarrow A^H(Ax - b) = 0$$

being represented as

$$K_\alpha \begin{pmatrix} s \\ x \end{pmatrix} \equiv \begin{pmatrix} \alpha I & A \\ A^H & 0 \end{pmatrix} \begin{pmatrix} r/\alpha \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

- cond(K_α) roughly varies between cond(A) and cond$(A)^2$
- [Björck-1967,1992] showed $\alpha = \sigma_{\min}(A)$ is quasi-optimal for both $[s; x]$ and x via straight-forward argument from eigenpairs of K_α
- Used by MATLAB until mid-1990's [Matstoms-1994]; discarded because of expense of pivoted LU

The history of augmented systems for least squares

There is a long history [Siegel-1965,Björck-1967,Hatchel-1974,Björck-1992] of

$$\min_x \|Ax - b\|_2 \Rightarrow A^H(Ax - b) = 0$$

being represented as

$$K_\alpha \begin{pmatrix} s \\ x \end{pmatrix} \equiv \begin{pmatrix} \alpha I & A \\ A^H & 0 \end{pmatrix} \begin{pmatrix} r/\alpha \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

- cond(K_α) roughly varies between cond(A) and cond(A)^2
- [Björck-1967,1992] showed $\alpha = \sigma_{\min}(A)$ is quasi-optimal for both $[s; x]$ and x via straight-forward argument from eigenpairs of K_α
- Used by MATLAB until mid-1990's [Matstoms-1994]; discarded because of expense of pivoted LU

The history of augmented systems for least squares

There is a long history [Siegel-1965,Björck-1967,Hatchel-1974,Björck-1992] of

$$\min_x \|Ax - b\|_2 \Rightarrow A^H(Ax - b) = 0$$

being represented as

$$K_\alpha \begin{pmatrix} s \\ x \end{pmatrix} \equiv \begin{pmatrix} \alpha I & A \\ A^H & 0 \end{pmatrix} \begin{pmatrix} r/\alpha \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

- $\text{cond}(K_\alpha)$ roughly varies between $\text{cond}(A)$ and $\text{cond}(A)^2$
- [Björck-1967,1992] showed $\alpha = \sigma_{\min}(A)$ is quasi-optimal for both $[s; x]$ and x via straight-forward argument from eigenpairs of K_α
- Used by MATLAB until mid-1990's [Matstoms-1994]; discarded because of expense of pivoted LU

The history of augmented systems for least squares

There is a long history [Siegel-1965,Björck-1967,Hatchel-1974,Björck-1992] of

$$\min_x \|Ax - b\|_2 \Rightarrow A^H(Ax - b) = 0$$

being represented as

$$K_\alpha \begin{pmatrix} s \\ x \end{pmatrix} \equiv \begin{pmatrix} \alpha I & A \\ A^H & 0 \end{pmatrix} \begin{pmatrix} r/\alpha \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}.$$

- cond(K_α) roughly varies between cond(A) and cond(A)^2
- [Björck-1967,1992] showed $\alpha = \sigma_{\min}(A)$ is quasi-optimal for both $[s; x]$ and x via straight-forward argument from eigenpairs of K_α
- Used by MATLAB until mid-1990's [Matstoms-1994]; discarded because of expense of pivoted LU

Conclusions, future work, availability

- The dynamic regularization commonly used by distributed sparse linear and IPM solvers can fail catastrophically
- The (carefully-scaled!) augmented system is likely to have a sparser factorization than the normal equations (and they share the same *effective* conditioning)
- Given a Lanczos procedure for estimating $\sigma_{\min}(A)$, we have a powerful tool for distributed sparse-direct linear and least squares problems
- Augmented systems should be revived, though a careful usage of extended precision may be needed; perhaps computation of $|A|$ and $|L||D||L|^T$ could predict the need for refactoring in extended precision

Thanks: Michael Saunders and Stephen Boyd's entire group for extended discussions

Availability: Highly configurable python interface to distributed sparse QPs available at `libelemental.org`

Conclusions, future work, availability

- The dynamic regularization commonly used by distributed sparse linear and IPM solvers can fail catastrophically
- The (carefully-scaled!) augmented system is likely to have a sparser factorization than the normal equations (and they share the same *effective* conditioning)
- Given a Lanczos procedure for estimating $\sigma_{\min}(A)$, we have a powerful tool for distributed sparse-direct linear and least squares problems
- Augmented systems should be revived, though a careful usage of extended precision may be needed; perhaps computation of $|A|$ and $|L||D||L|^T$ could predict the need for refactoring in extended precision

Thanks: Michael Saunders and Stephen Boyd's entire group for extended discussions

Availability: Highly configurable python interface to distributed sparse QPs available at `libelemental.org`

Conclusions, future work, availability

- The dynamic regularization commonly used by distributed sparse linear and IPM solvers can fail catastrophically
- The (carefully-scaled!) augmented system is likely to have a sparser factorization than the normal equations (and they share the same *effective* conditioning)
- Given a Lanczos procedure for estimating $\sigma_{\min}(A)$, we have a powerful tool for distributed sparse-direct linear and least squares problems
- Augmented systems should be revived, though a careful usage of extended precision may be needed; perhaps computation of $|A|$ and $|L||D||L|^T$ could predict the need for refactoring in extended precision

Thanks: Michael Saunders and Stephen Boyd's entire group for extended discussions

Availability: Highly configurable python interface to distributed sparse QPs available at `libelemental.org`

Conclusions, future work, availability

- The dynamic regularization commonly used by distributed sparse linear and IPM solvers can fail catastrophically
- The (carefully-scaled!) augmented system is likely to have a sparser factorization than the normal equations (and they share the same *effective* conditioning)
- Given a Lanczos procedure for estimating $\sigma_{\min}(A)$, we have a powerful tool for distributed sparse-direct linear and least squares problems
- Augmented systems should be revived, though a careful usage of extended precision may be needed; perhaps computation of $|A|$ and $|L||D||L|^T$ could predict the need for refactoring in extended precision

Thanks: Michael Saunders and Stephen Boyd's entire group for extended discussions

Availability: Highly configurable python interface to distributed sparse QPs available at `libelemental.org`