Accurate Eigenvalues and SVDs of Totally Nonnegative Matrices

Plamen Koev

San Jose State University

HMICGA, Dublin, Ireland, June 18, 2018

Supported by SJSU Woodward Fund for Applied Mathematics

Motivation

- Def: A matrix is Totally Nonnegative (TN) if all minors ≥ 0
- This talk: *All* matrix computations with TN matrices possible:
  - to high relative accuracy
  - in floating point arithmetic
  - at no extra cost
- Connection to computed-aided geometric design, e.g.,

  "When converting a curve expressed in a B-spline expansion into its Bézier form, corner cutting of the B-spline control polygon leads to the Bézier points exactly when the Bézier matrix is totally positive."

  Ref: Corner cutting algorithms for the Bézier representation of free form curves, Goodman, Micchelli, Linear Algebra Appl. 1998.

Examples of TN matrices

- Vandermonde, Hilbert, Pascal:

$$V = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2^2 & 2^3 \\ 1 & 3 & 3^2 & 3^3 \\ 1 & 5 & 5^2 & 5^3 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 \\ 1/2 & 1/3 & 1/4 & 1/5 \\ 1/3 & 1/4 & 1/5 & 1/6 \\ 1/4 & 1/5 & 1/6 & 1/7 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$$

- Also Cauchy, Said–Ball, etc.
- Ubiquitous in practice: Occupy an octant in n^2 space when properly parameterized (as will see)

The matrix eigenvalue problem

- Goal: compute all eigenvalues of a TN matrix in floating point arithmetic
- Including the zero Jordan structure!
- Very hard in general (per higher powers):

  "***The Jordan form is useful theoretically but is very hard to compute in a numerically stable fashion...***"

  James Demmel, Applied Numerical Linear Algebra.
- Why is that?

Floating point arithmetic

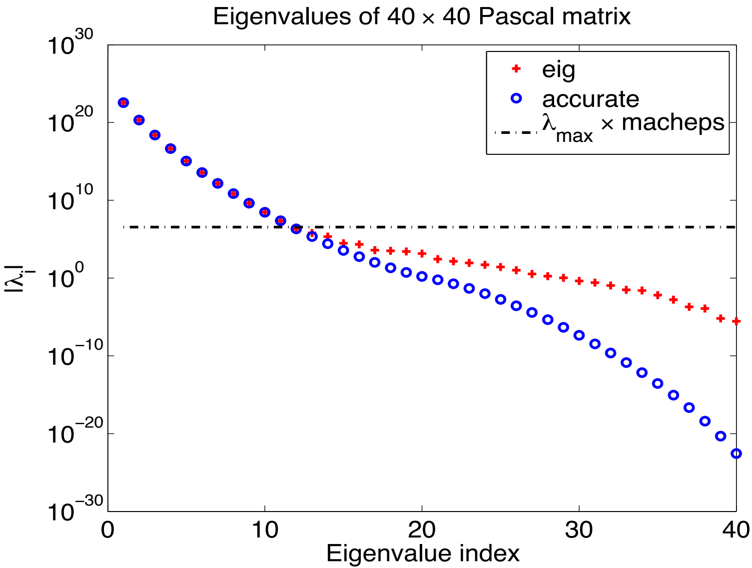
- Finite, countably many floating point numbers representing the infinite, uncountable $\mathbb{R}$
- Roundoff errors could make equal eigenvalues different and *destroy* the Jordan structure!
- Even accurate eigenvalues alone are problematic
- The determinant and eigenvalues of $\begin{bmatrix} 1 & 3 \\ 3 & 9 \end{bmatrix}$.

```
>> det([1 3; 3 9])
ans =
-4.9960e-16
>> eig([1 3; 3 9])
ans =
1.1102e-16
1.0000e+01
```

Eigenvalues of Pascal Matrix (which is TN)

$$P_n = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1 & 2 & 3 & 4 & \cdots \\ 1 & 3 & 6 & 10 & \cdots \\ 1 & 4 & 10 & 20 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

$$cond(P_{40}) = 6 \times 10^{44}$$



Eigenvalues of 40 × 40 Pascal matrix
eig
accurate
λmax × macheps
|λi|
Eigenvalue index
10^30
10^20
10^10
10^0
10^−10
10^−20
10^−30
0
10
20
30
40

Absolute vs. Relative Accuracy

- Absolute accuracy
$$|x - \hat{x}| \leq \varepsilon$$
  - Depends on the magnitude of x
  - $\varepsilon = 10^{-4}$ means what?
  - Does x equal distance between planets or between molecules?
- *Relative* accuracy
$$|x - \hat{x}| \leq \varepsilon |x|$$
  - Works fine regardless of magnitude of x
  - $\varepsilon = 10^{-4}$ means $\hat{x}$ has 4 correct decimal digits!
  - What if $x = 0$?

Reason accuracy is lost in floating point arithmetic

- $\text{fl}(a \odot b) = (a \odot b)(1 + \delta), \quad \odot \in \{+, -, \times, /\}$
- Relative accuracy preserved in $\times, +, /$
  Proof: $(1 + \delta)$ factors accumulate multiplicatively
- Subtractions of approximate quantities dangerous:

$$\begin{array}{rl} & .123456789\color{red}{xxx} \\ - & .123456789\color{red}{yyy} \\ \hline & .000000000\color{red}{zzz} \end{array}$$

  - subtraction of *exact* initial data is OK!
  - if *all* other subtractions avoided, we get accuracy
- Exactly what we do with TN matrices

Question:

- How do we know if a matrix is TN?

$$\begin{bmatrix} 1 & 2 & 6 \\ 4 & 13 & 69 \\ 28 & 131 & 852 \end{bmatrix}$$

Question:

- How do we know if a matrix is TN?

$$\begin{bmatrix} 1 & 2 & 6 \\ 4 & 13 & 69 \\ 28 & 131 & 852 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & 7 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 4 & 1 & \\ & 8 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 5 & \\ & & 9 \end{bmatrix} \begin{bmatrix} 1 & 2 & \\ & 1 & 6 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & 3 \\ & & 1 \end{bmatrix}$$

- Product of TN bidiagonals

Question:

- How do we know if a matrix is TN?

$$\begin{bmatrix} 1 & 2 & 6 \\ 4 & 13 & 69 \\ 28 & 131 & 852 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & 7 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 4 & 1 & \\ & 8 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 5 & \\ & & 9 \end{bmatrix} \begin{bmatrix} 1 & 2 & \\ & 1 & 6 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & 3 \\ & & 1 \end{bmatrix}$$

- Product of TN bidiagonals
- In general:

$$A = L_1 \cdots L_{n-1} \cdot D \cdot U_{n-1} \cdots U_1,$$

  where L_i lower bidiagonal, D diagonal, U_i upper bidiagonal
- Cauchy–Binet: TN $\times$ TN $=$ TN
- Each red entry $= \frac{\mathrm{minor}_1(A)}{\mathrm{minor}_2(A)} \cdot \frac{\mathrm{minor}_3(A)}{\mathrm{minor}_4(A)}$

Question:

- How do we know if a matrix is TN?

$$\begin{bmatrix} 1 & 2 & 6 \\ 4 & 13 & 69 \\ 28 & 131 & 852 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & 7 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 4 & 1 & \\ & 8 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 5 & \\ & & 9 \end{bmatrix} \begin{bmatrix} 1 & 2 & \\ & 1 & 6 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & 3 \\ & & 1 \end{bmatrix}$$

$$\downarrow$$

$$\mathcal{BD}(A) = \left\{ \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \right\}$$

The n^2 nontrivial entries of $\mathcal{BD}(A)$:

- parameterize class of nonsingular TN matrices
- allow for highly accurate computations:
  If $A \to B$, then $\mathcal{BD}(A) \to \mathcal{BD}(B)$ accurate

Starting point is the bidiagonal decomposition

$$A = \begin{bmatrix} 1 & & \\ & 1 & \\ & l_{31} & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ l_{21} & 1 & \\ & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & \\ & d_2 & \\ & & d_3 \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \\ & 1 & u_{23} \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & u_{13} \\ & & 1 \end{bmatrix}$$

- These are parameters the user must deliver
- Readily available for Vandermonde $A = \left[x_i^{j-1}\right]_{i,j=1}^{n}$

$$d_i = \prod_{j=1}^{i-1}(x_i - x_j), \quad l_{ik} = \prod_{j=n-k}^{i-1} \frac{x_{i+1} - x_{j+1}}{x_i - x_j}, \quad u_{ik} = x_{i+n-k}$$

Starting point is the bidiagonal decomposition

$$A = \begin{bmatrix} 1 & & \\ & 1 & \\ & l_{31} & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ l_{21} & 1 & \\ & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & \\ & d_2 & \\ & & d_3 \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \\ & 1 & u_{23} \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & u_{13} \\ & & 1 \end{bmatrix}$$

- Cauchy $A = \left[\frac{1}{x_i + y_j}\right]_{i,j=1}^{n}$

$$d_i = \prod_{k=1}^{i-1} \frac{(x_i - x_k)(y_i - y_k)}{(x_i + y_k)(y_i + x_k)}$$

$$l_{ik} = \frac{x_{n-k} + y_{i-n+k+1}}{x_i + y_{i-n+k+1}} \prod_{l=n-k}^{i-1} \frac{x_{i+1} - x_{l+1}}{x_i - x_l} \prod_{l=1}^{i-n+k-1} \frac{x_i + y_l}{x_{i+1} + y_l}$$

$$u_{ik} = \frac{y_{n-k} + x_{i-n+k+1}}{y_i + x_{i-n+k+1}} \prod_{l=n-k}^{i-1} \frac{y_{i+1} - y_{l+1}}{y_i - y_l} \prod_{l=1}^{i-n+k-1} \frac{y_i + x_l}{y_{i+1} + x_l}$$

Starting point is the bidiagonal decomposition

$$A = \begin{bmatrix} 1 & & \\ & 1 & \\ & l_{31} & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ l_{21} & 1 & \\ & l_{32} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & \\ & d_2 & \\ & & d_3 \end{bmatrix} \begin{bmatrix} 1 & u_{12} & \\ & 1 & u_{23} \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & u_{13} \\ & & 1 \end{bmatrix}$$

- Pascal: $d_i = l_{ij} = u_{ij} = 1$
- Reasonable requirement: Show me that your matrix is TN!
- Critical observation: $\mathcal{BD}(A)$ determines all eigenvalues accurately! (matrix entries do not!)
- I.e., small relative perturbations in $\mathcal{BD}(A)$ cause small relative perturbations to eigenvalues
- Eigenvalues “deserve” to be computed accurately
- The logic: The computed eigenvalues are rational functions of entries of $\mathcal{BD}(A)$
- So those rational functions *must have* only positive coefficients (intuition, not proof)

Eigenvalue algorithm

- Reduction to tridiagonal form (Cryer '76)
- Using only one operation: Addition/subtraction of one row to the next/previous

$$\begin{bmatrix} + & + & + & + \\ + & + & + & + \\ + & + & + & + \\ + & + & + & + \end{bmatrix} \to \begin{bmatrix} + & + & + & + \\ + & + & + & + \\ + & + & + & + \\ 0 & + & + & + \end{bmatrix} \to \begin{bmatrix} + & + & + & 0 \\ + & + & + & + \\ + & + & + & + \\ 0 & + & + & + \end{bmatrix} \to \begin{bmatrix} + & + & + & 0 \\ + & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \end{bmatrix} \to$$

$$\begin{bmatrix} + & + & 0 & 0 \\ + & + & + & + \\ 0 & + & + & + \\ 0 & + & + & + \end{bmatrix} \to \begin{bmatrix} + & + & 0 & 0 \\ + & + & + & + \\ 0 & + & + & + \\ 0 & 0 & + & + \end{bmatrix} \to \begin{bmatrix} + & + & 0 & 0 \\ + & + & + & 0 \\ 0 & + & + & + \\ 0 & 0 & + & + \end{bmatrix}$$

- $\lambda_i = \sigma_i^2$, $\sigma_i =$ singular values of bidiagonal Cholesky factor
- Solved accurately by Demmel and Kahan in 1989
- We implement on $\mathcal{BD}(A)$ and not on A!

Cryer's algorithm applied to A

- To create a 0 in position (3,1) of

$$\begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix}$$

we use similarity

$$\begin{bmatrix} 1 & & \\ & 1 & \\ & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & 1 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 0 & 1 & 7 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 6 & 4 \\ 1 & 12 & 9 \\ 0 & 8 & 7 \end{bmatrix}$$

Now, Cryer's applied to $\mathcal{BD}(A)$

- Subtracting a multiple of one row from next to create a 0 is equivalent to setting an entry of the BD to 0

$$\begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 1 & 1 & \\ & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & \\ & 1 & 3 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & 2 \\ & & 1 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} 1 & 2 & 4 \\ 1 & 3 & 9 \\ 0 & 1 & 7 \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ 1 & 1 & \\ & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & \\ & & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & \\ & 1 & 3 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & \\ & 1 & 2 \\ & & 1 \end{bmatrix}$$

- No arithmetic performed!
- New matrix still TN

Next: Completing the similarity

- Adding a multiple of one row/col to next/previous is done by changing the entries of the BD only

$$\begin{bmatrix}1&2&4\\1&3&9\\0&1&7\end{bmatrix} = \begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\1&1&\\&1&1\end{bmatrix}\begin{bmatrix}1&&\\&1&\\&&2\end{bmatrix}\begin{bmatrix}1&2&\\&1&3\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&2\\&&1\end{bmatrix}$$

$$\downarrow \qquad\qquad\qquad\qquad \downarrow$$

$$\begin{bmatrix}1&6&4\\1&12&9\\0&8&7\end{bmatrix} = \begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\2&1&\\&\frac{3}{2}&1\end{bmatrix}\begin{bmatrix}1&&\\&6&\\&&2\end{bmatrix}\begin{bmatrix}1&6&\\&1&\frac{4}{3}\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&2\\&&1\end{bmatrix}$$

- New entries are rational functions with >0 coefficients
- No subtractions $\Rightarrow$ accuracy
- New matrix is still TN (Cauchy–Binet)

Elementary bidiagonal matrix

- The only operation we need to compute everything TN

$$E_i(b,c) = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & c & & & \\ & & b & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix}$$

an *“elementary bidiagonal matrix”*, $b \geq 0, c \geq 0$

- Differs from I in two entries only
- Building block of $\mathcal{BD}(A)$: $A = (\prod E_i) \cdot (\prod E_i^T)$
- $E_i =$ addition/subtraction of multiple of one row/column from next previous

Multiplication by E_i

$$\begin{bmatrix}1 & & \\ & 1 & \\ & 0 & 1\end{bmatrix}\begin{bmatrix}1 & & \\ b & 1 & \\ & c & 1\end{bmatrix}\begin{bmatrix}d & & \\ & e & \\ & & f\end{bmatrix}\begin{bmatrix}1 & g & \\ & 1 & h \\ & & 1\end{bmatrix}\underline{\begin{bmatrix}1 & & \\ & 1 & k \\ & & 1\end{bmatrix}\color{red}{\begin{bmatrix}1 & & \\ & y & \\ & x & z\end{bmatrix}}}$$

Multiplication by E_i

$$\begin{bmatrix}1 & & \\ & 1 & \\ & 0 & 1\end{bmatrix}\begin{bmatrix}1 & & \\ b & 1 & \\ & c & 1\end{bmatrix}\begin{bmatrix}d & & \\ & e & \\ & & f\end{bmatrix}\begin{bmatrix}1 & g & \\ & 1 & h \\ & & 1\end{bmatrix}\underline{\begin{bmatrix}1 & & \\ & 1 & k \\ & & 1\end{bmatrix}\begin{bmatrix}1 & & \\ & y & \\ & x & z\end{bmatrix}}$$

$$\begin{bmatrix}1 & & \\ & 1 & \\ & 0 & 1\end{bmatrix}\begin{bmatrix}1 & & \\ b & 1 & \\ & c & 1\end{bmatrix}\begin{bmatrix}d & & \\ & e & \\ & & f\end{bmatrix}\begin{bmatrix}1 & g & \\ & 1 & h \\ & & 1\end{bmatrix}\underline{\begin{bmatrix}1 & & \\ & y' & \\ & x' & z'\end{bmatrix}\begin{bmatrix}1 & & \\ & 1 & k' \\ & & 1\end{bmatrix}}$$

$$\begin{aligned} x' &= x \\ y' &= y + kx \\ z' &= 1/y' \\ k' &= kz/y_1 \end{aligned}$$

- This is the LR algorithm, implemented as dqd

Multiplication by E_i

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\begin{bmatrix}d&&\\&e&\\&&f\end{bmatrix}\begin{bmatrix}1&g&\\&1&h\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&k\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&y&\\&x&z\end{bmatrix}$$

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\begin{bmatrix}d&&\\&e&\\&&f\end{bmatrix}\underline{\begin{bmatrix}1&g&\\&1&h\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&y'&\\&x'&z'\end{bmatrix}}\begin{bmatrix}1&&\\&1&k'\\&&1\end{bmatrix}$$

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\begin{bmatrix}d&&\\&e&\\&&f\end{bmatrix}\underline{\begin{bmatrix}1&&\\&y''&\\&x''&z''\end{bmatrix}\begin{bmatrix}1&g'&\\&1&h'\\&&1\end{bmatrix}}\begin{bmatrix}1&&\\&1&k'\\&&1\end{bmatrix}$$

Multiplication by E_i

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\begin{bmatrix}d&&\\&e&\\&&f\end{bmatrix}\begin{bmatrix}1&g&\\&1&h\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&k\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&y&\\&x&z\end{bmatrix}$$

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\begin{bmatrix}d&&\\&e&\\&&f\end{bmatrix}\begin{bmatrix}1&g&\\&1&h\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&y'&\\&x'&z'\end{bmatrix}\begin{bmatrix}1&&\\&1&k'\\&&1\end{bmatrix}$$

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\underline{\begin{bmatrix}d&&\\&e&\\&&f\end{bmatrix}\begin{bmatrix}1&&\\&y''&\\&x''&z''\end{bmatrix}}\begin{bmatrix}1&g'&\\&1&h'\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&k'\\&&1\end{bmatrix}$$

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\underline{\begin{bmatrix}1&&\\&1&\\&x'''&1\end{bmatrix}\begin{bmatrix}d&&\\&e'&\\&&f'\end{bmatrix}}\begin{bmatrix}1&g'&\\&1&h'\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&k'\\&&1\end{bmatrix}$$

Multiplication by E_i

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\begin{bmatrix}d&&\\&e&\\&&f\end{bmatrix}\begin{bmatrix}1&g&\\&1&h\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&k\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&y&\\&x&z\end{bmatrix}$$

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\begin{bmatrix}d&&\\&e&\\&&f\end{bmatrix}\begin{bmatrix}1&g&\\&1&h\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&y'&\\&x'&z'\end{bmatrix}\begin{bmatrix}1&&\\&1&k'\\&&1\end{bmatrix}$$

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\begin{bmatrix}d&&\\&e&\\&&f\end{bmatrix}\begin{bmatrix}1&&\\&y''&\\&x''&z''\end{bmatrix}\begin{bmatrix}1&g'&\\&1&h'\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&k'\\&&1\end{bmatrix}$$

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c&1\end{bmatrix}\begin{bmatrix}1&&\\&1&\\&x'''&1\end{bmatrix}\begin{bmatrix}d&&\\&e'&\\&&f'\end{bmatrix}\begin{bmatrix}1&g'&\\&1&h'\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&k'\\&&1\end{bmatrix}$$

$$\begin{bmatrix}1&&\\&1&\\&0&1\end{bmatrix}\begin{bmatrix}1&&\\b&1&\\&c+x'''&1\end{bmatrix}\begin{bmatrix}d&&\\&e'&\\&&f'\end{bmatrix}\begin{bmatrix}1&g'&\\&1&h'\\&&1\end{bmatrix}\begin{bmatrix}1&&\\&1&k'\\&&1\end{bmatrix}$$

Jordan blocks corresponding to zero eigenvalues

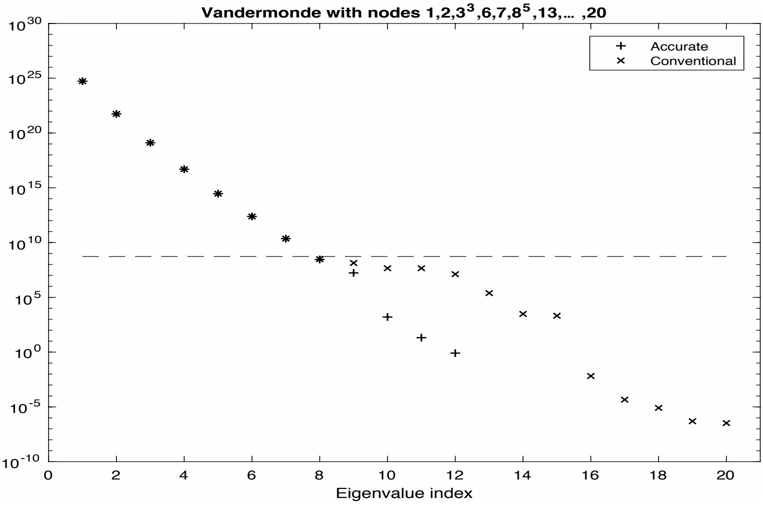
- OK, we get accurate eigenvalues
- Zero eigenvalues are exact!
- How about Jordan blocks?
  - $n - \mathrm{rank}(A) =$ # Jordan blocks
  - $\mathrm{rank}(A) - \mathrm{rank}(A^2) =$ # of Jordan blocks of size ≥ 2
  - ...
  - $\mathrm{rank}(A), \mathrm{rank}(A^2), \ldots$ readily obtainable from its BD
  - A^2 is TN (as a product of TN) and its BD is a TN-preserving op, thus BD accurate
  - need to form BD of $A^2, \ldots, A^n$

Example

```
A =
      3  3  2  1
      2  2  3  2
      1  1  2  3
      1  1  2  3

>> eig(A)
ans =
      7.828427124746188e+00
      2.171572875253811e+00
      5.247731480861326e-16
     -1.110223024625157e-16

>> [B,C]=STNBD(A); [e,jb]=STNEigenValues(B,C)
e =
      7.8284
      2.1716
      0
      0
jb =
      2
```



Vandermonde with nodes 1,2,3^3,6,7,8^5,13,... ,20
Accurate
Conventional
10^30
10^25
10^20
10^15
10^10
10^5
10^0
10^-5
10^-10
0
2
4
6
8
10
12
14
16
18
20
Eigenvalue index

Conclusions

- First example of a Jordan structure being computed to high relative accuracy
- Complete Jordan structure of *Irreducible* TN matrices (Nonzero eigenvalues are distinct per Fallat–Gekhtman.)
- Future work: Get formulas for non-unique BD of singular Vandermonde, etc.
- Software, paper: `http://www.math.sjsu.edu/~koev`