Plan for today

Intro to HW3, MPI data movements for

- Jacobi methods

Two problems. Find:

- the eigendecomposition (EVD) of a symmetric real matrix A

$$A = J\mathrm{diag}(\lambda_1, ..., \lambda_n)J^T,\ J^T J = JJ^T = I_n$$

  $J^T J = JJ^T = I_n$ means that J is orthogonal.

- the singular value decomposition (SVD) of a real matrix A

$$A = U\Sigma V^T,\ U^T U = I_m,\ V^T V = I_n,\ \Sigma = \begin{pmatrix} \mathrm{diag}(\sigma_1, ..., \sigma_n) \\ 0_{(m-n)\times n} \end{pmatrix}$$

  $U^T U = I_m$, $V^T V = I_n$ means that U is orthogonal of dimension m, V is orthogonal of dimension n,

Start with EVD and apply the concept to SVD

Two-sided Jacobi method for EVD

Want $A = J\text{diag}(\lambda_1, ..., \lambda_n)J^T$.

Jacobie's idea:

Generate a sequence of $J^{(i)}$, $i = 1, 2, ..., \infty$ so

$$J = \lim_{i \to \infty} J^{(i)}$$

- $J^{(i)}$ is orthogonal for $i = 1, 2, ..., \infty$
- for $A^{(i+1)} = (J^{(i)})^T A^{(i)} J^{(i)}$ there are $\frac{n}{2}$ zero elements above the diagonal and below the diagonal as($A^(i)$ is symmetric)

$$\lim_{i \to \infty} A^{(i)} = \text{diag}(\lambda_1, ..., \lambda_n)$$

Two-sided Jacobi - 2×2 case

The process starts with a 2×2 subproblem. Find c, s, $c^2 + s^2 = 1$ s.t.

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix}^T \begin{pmatrix} a_{p,p} & a_{p,q} \\ a_{q,p} & a_{q,q} \end{pmatrix} \underbrace{\begin{pmatrix} c & s \\ -s & c \end{pmatrix}}_{G(p,q)} = \begin{pmatrix} \hat{a}_{p,p} & 0 \\ 0 & \hat{a}_{q,q} \end{pmatrix}$$

$$\begin{aligned} orthogonality \ - \ G^T(p,q)G(p,q) &= I_2, \\ a_{p,p}^2 + a_{p,q}^2 + a_{p,q}^2 + a_{q,q}^2 &= \hat{a}_{p,p}^2 + \hat{a}_{q,q}^2. \end{aligned}$$

We want $0 = \hat{a}_{p,q}$, or

$$\begin{aligned} 0 &= a_{p,q}(c^2 - s^2) + (a_{p,p} - a_{q,q})cs = \hat{a}_{p,q}, \quad \text{divide by } \ a_{p,q}c^2 \\ 0 &= (1 - t^2) + \frac{a_{p,p} - a_{q,q}}{a_{p,q}}t, \ \ t = \frac{s}{c} \ \ (t \text{ stands for a tangent}) \\ 0 &= t^2 + 2\tau t - 1, \ \ \tau = \frac{a_{q,q} - a_{p,p}}{2a_{p,q}} \end{aligned}$$

Never: compute $\arctan(t)$!

The roots of the quadratic are

$$t^2 + 2\tau t - 1 = 0 \;\Rightarrow\; t = -\tau \pm \sqrt{1+\tau^2}$$

from which we obtain

$$c = \frac{1}{\sqrt{1+t^2}},\; s = ct.$$

From the two solutions for t, we take the **smaller** in absolute value,

$$t = \begin{cases} -\tau + \sqrt{1+\tau^2} & \text{if } \tau \geq 0 \\ -\tau - \sqrt{1+\tau^2} & \text{if } \tau < 0 \end{cases}$$

However, because of possible large roundoff errors in floating point arithmetic, we do not want to subtract numbers of the same sign.

Two-sided Jacobi - accurate cos *and* sin

$$t = \begin{cases} -\tau + \sqrt{1+\tau^2} & \text{if } \tau \geq 0 \\ -\tau - \sqrt{1+\tau^2} & \text{if } \tau < 0 \end{cases}$$

There is a way out of this predicament. Recall

$$(x-y)(x+y) = x^2 - y^2$$

Thus, multiply and divide t by

$$-\tau - \operatorname{sign}(\tau)\sqrt{1+\tau^2}$$

which gives the following formula for t

$$t = \frac{1}{\tau + \operatorname{sign}(\tau)\sqrt{1+\tau^2}}, \;\; c = \frac{1}{\sqrt{1+t^2}}, \;\; s = c \cdot c$$

Two-sided Jacobi - rotations in n dimensions

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix}^T \underbrace{\begin{pmatrix} a_{p,p} & a_{p,q} \\ a_{p,q} & a_{q,q} \end{pmatrix}}_{G(p,q)} \begin{pmatrix} c & s \\ -s & c \end{pmatrix} = \begin{pmatrix} \hat{a}_{p,p} & 0 \\ 0 & \hat{a}_{q,q} \end{pmatrix}$$

Now we embed $G(p,q)$ into an $n \times n$ matrix $J(p,q)$,

$$G(p,q) \longrightarrow J(p,q) = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & & \vdots & & \vdots & \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & \ddots & & \vdots & & \vdots & \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & \ddots & & \vdots & & \vdots & \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}$$

Two-sided Jacobi - a single step

Now we formaly (not a matrix-matrix mult) apply rotations

$$J^T(p,q)AJ(p,q) = \hat{A}.$$

J is orthogonal so $||A||_F = ||\hat{A}||_F$.

$$\hat{A} = \begin{pmatrix} a_{1,1} & \cdots & \hat{a}_{1,p} & \cdots & \hat{a}_{1,q} & \cdots & a_{1,n} \\ \vdots & \ddots & | & & | & \vdots & \\ \hat{a}_{p,1} & — & \hat{a}_{p,p} & — & \hat{\mathbf{a}}_{\mathbf{p,q}} & — & \hat{a}_{p,n} \\ \vdots & \ddots & | & & | & \vdots & \\ \hat{a}_{q,1} & — & \hat{\mathbf{a}}_{\mathbf{q,p}} & — & \hat{a}_{q,q} & — & \hat{a}_{q,n} \\ \vdots & \ddots & | & & | & \vdots & \\ a_{n,1} & \cdots & \hat{a}_{n,p} & \cdots & \hat{a}_{n,q} & \cdots & a_{n,n} \end{pmatrix}, \quad \hat{\mathbf{a}}_{\mathbf{p,q}} = \hat{\mathbf{a}}_{\mathbf{q,p}} = 0$$

Two-sided Jacobi - $\text{off}(A)^2$

Define

$$\text{off}(A)^2 = ||A||_F^2 - \sum_{i=1}^{n} a_{i,i}^2 = \sum_{i \neq j}^{n} a_{i,j}^2$$

$$(a_{pp}, a_{pq}, a_{qp}, a_{qq}) \xrightarrow{J(p,q)} (\hat{a}_{pp}, 0, \hat{a}_{qq}, 0)$$

$$\begin{aligned} a_{p,p}^2 + a_{p,q}^2 + a_{p,q}^2 + a_{q,q}^2 &= \hat{a}_{p,p}^2 + \hat{a}_{q,q}^2 \\ a_{p,p}^2 + a_{q,q}^2 &\leq \hat{a}_{p,p}^2 + \hat{a}_{q,q}^2 \end{aligned}$$

Let $A^{(1)} = J(p,q)^T A J(p,q)$. Then

- $||A^{(1)}||_F^2 = ||A||_F^2||$ (J does not change the Frobenius norm)
- $\sum_{i=1}^{n} a_{i,i}^2 \leq \sum_{i=1}^{n} \hat{a}_{i,i}^2$

$$\begin{aligned} \text{off}(A^{(1)})^2 = ||A^{(1)}||_F^2 - \sum_{i=1}^{n} \hat{a}_{i,i}^2 &= ||A||_F^2 - \sum_{i=1}^{n} \hat{a}_{i,i}^2 \leq ||A||_F^2 - \sum_{i=1}^{n} a_{i,i}^2 = \text{off}(A)^2 \\ \text{off}(A^{(1)})^2 &< \text{off}(A)^2 \end{aligned}$$

Two-sided Jacobi - convergence

- Going form $A^{(i)}$ to $A^{(i+1)}$ we decrease $\text{off}(A^{(i)})$.
- In the limit $\text{off}(A^{(\infty)}) = 0$

$$\left(\prod_{i=1}^{\infty}(J^{(i)})\right)^T A \underbrace{\left(\prod_{i=1}^{\infty} J^{(i)}\right)}_{U} = \underbrace{\text{diag}(\lambda_1, \ldots \lambda_n)}_{\Lambda}$$

- The matrix U is the matrix of eigenvectors of A.
- We need a final termination criterion, for example $\text{off}(A^{(i)}) \leq \epsilon$.
- If after K steps termination criterion is satisfied, then

$$\hat{U} = \prod_{i=1}^{K} J^{(i)} \approx U, \;\; \text{diag}(A^{(K)}) \approx \text{diag}(\lambda_1, \ldots \lambda_n)$$

Two-sided Jacobi row-by-row order

Need to select order for zeroing off-diagonal elements.

For example, one can follow the row-by-row order

$$
\begin{array}{ccc}
(1,2) & \cdots & (1,n) \\
(2,3) & \cdots & (2,n) \\
 & \ddots & \vdots \\
 & & (n-1,n)
\end{array}
$$

A **sweep** is a sequence of Jacobi rotations which annihilate all off-diagonal elements exactly once ($\frac{n(n-1)}{2}$ rotations)

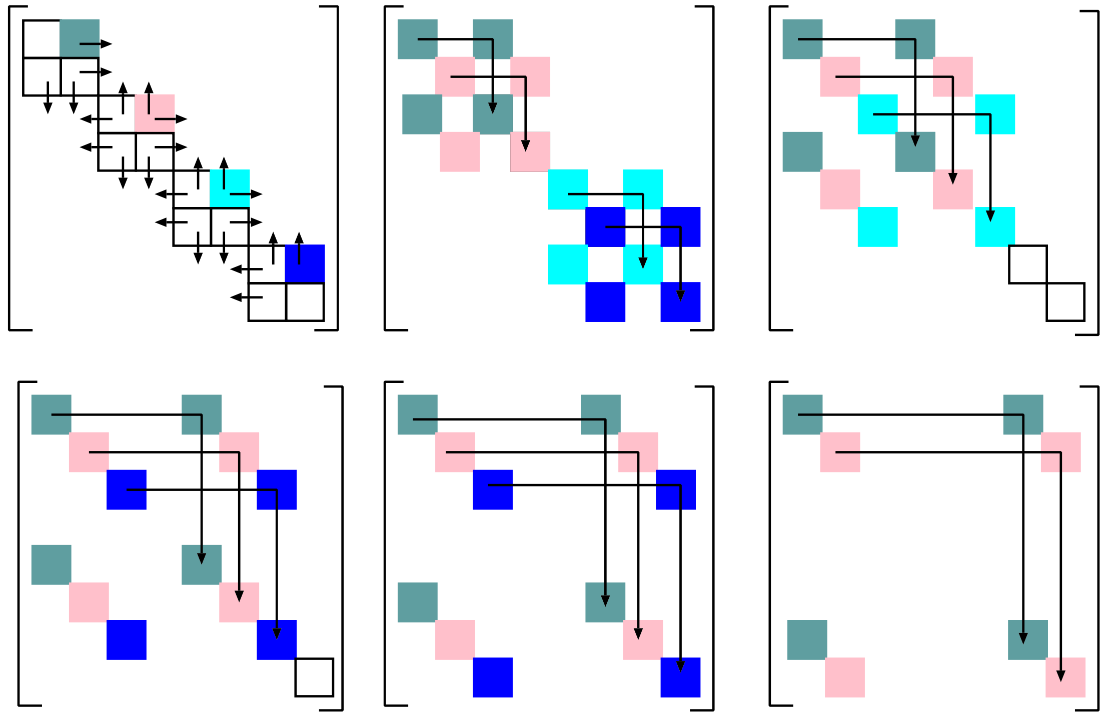
Two-sided Jacobi - parallelization

We want to parallelize the Jacobi method.

- a rotation acts on 2 columns (and 2 rows)
- there are n columns (and rows)
- $n/2$ independent rotations $J(p, q)$ can be computed simultaneously
- call simultaneous rotations a **compound** rotation
- $n/2$ rotations can be simultaneously applied from the right, followed by
- simultaneous application of the $n/2$ rotations from the left
- $(n-1)$ compound rotations makes a sweep

How do we find a sequence of $(n-1)$ compound rotations to realize a sweep?

Two-sided Jacobi - parallel row-by-row case



Shared memory - diminishing number of idependent rotations.

Distributed memory - columns (p, q) in "remote" PEs.

The "upper-left corner" in the figure allows $\frac{n}{2}$ independent rotations

The best parallel compound rotation:

$$\text{f}or\ n = 8, \quad J = \left(\begin{array}{cc|cc|cc|cc} c_1 & s_1 & & & & & & \\ -s_1 & c_1 & & & & & & \\ \hline & & c_2 & s_2 & & & & \\ & & -s_2 & c_2 & & & & \\ \hline & & & & c_3 & s_3 & & \\ & & & & -s_3 & c_3 & & \\ \hline & & & & & & c_3 & s_3 \\ & & & & & & -s_3 & c_3 \end{array}\right)$$

Two-sided Jacobi - compound roatation on the right

$$AJ = \left(\begin{array}{cc|cc|c} a_{11}c_1 - a_{12}s_1 & a_{11}s_1 + a_{12}c_1 & a_{13}c_2 - a_{14}s_2 & a_{13}s_2 + a_{14}c_2 & \cdots \\ a_{21}c_1 - a_{22}s_1 & a_{21}s_1 + a_{22}c_1 & a_{23}c_2 - a_{24}s_2 & a_{23}s_2 + a_{24}c_2 & \cdots \\ a_{31}c_1 - a_{32}s_1 & a_{31}s_1 + a_{32}c_1 & a_{33}c_2 - a_{34}s_2 & a_{33}s_2 + a_{34}c_2 & \cdots \\ a_{41}c_1 - a_{42}s_1 & a_{41}s_1 + a_{42}c_1 & a_{43}c_2 - a_{44}s_2 & a_{43}s_2 + a_{44}c_2 & \cdots \\ \vdots & \vdots & & \cdots & \end{array}\right)$$

or

$$AJ = \left(\begin{array}{cc|cc|c} a_{:,1}c_1 - a_{:,2}s_1 & a_{:,1}s_1 + a_{:,2}c_1 & a_{:,3}c_1 - a_{:,4}s_1 & a_{:,3}s_1 + a_{:,4}c_1 & \cdots \end{array}\right)$$

Cost $3n$ flops.

Two-sided Jacobi - compound rotation on the left

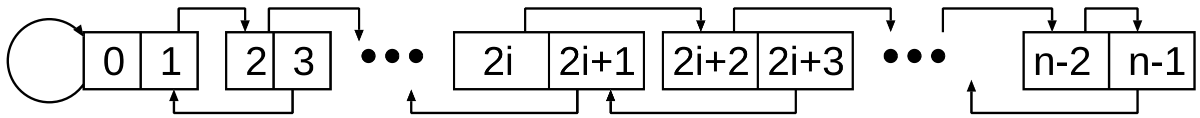
$$JA = \left(\begin{array}{c} c_1 a_{1,:} - s_1 a_{2,:} \\ s_1 a_{1,:} + c_1 a_{2,:} \\ \hline c_2 a_{1,:} - s_2 a_{2,:} \\ s_2 a_{1,:} + c_2 a_{2,:} \\ \hline \vdots \end{array} \right)$$

Can we find more compound rotations?

Idea: permute columns/rows to:

- bring them "together"
- always have $n/2$ independent rotations available
- permute by exchanging only neighboring columns (rows)

Two-sided Jacobi method



Consider pairs of columns/rows at consecutive positions $(2i, 2i+1)$, $i = 0, 1, 2, ..., n/2 - 1$, n even.

For all pairs, execute the "nearest neighbor" permutations:

If $0 \neq i \neq n/2 - 1$

move right the even position $(2i)$ to even $2(i+1)$

move left the odd position $2i + 3$ to odd $(2i + 1)$

If $i = 0$

move right odd (1) to even (2)

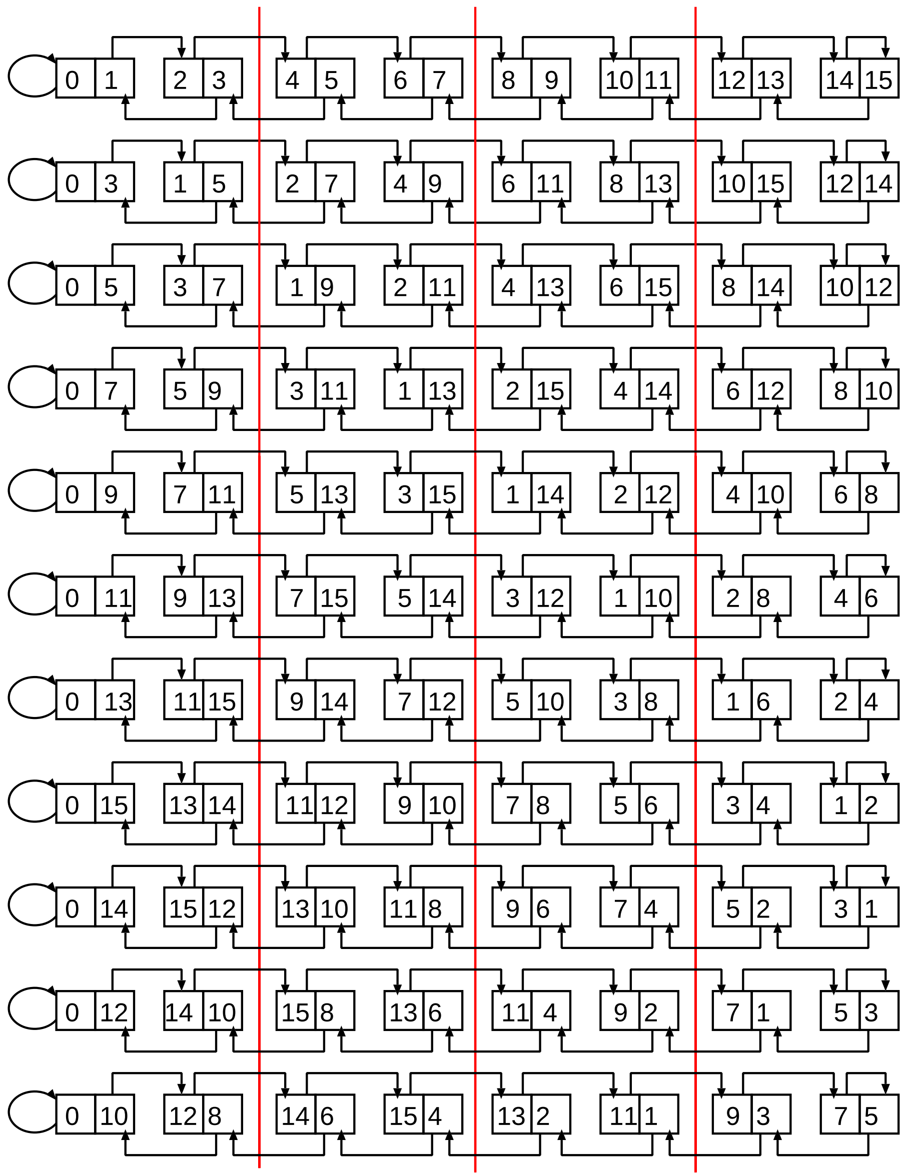
keep even (0) in even (0)

If $i = n/2 - 1$

move locally even $(n-2)$ to odd $(n/2 - 1)$, and

move left odd $(n/2 - 1)$ to odd $(n/2 - 3)$.

Two-sided Jacobi method



Two-sided Jacobi - 2D distribution

2D arrangement?

- compute rotations in "diagonal" PEs
- brodcast rotations along rows and columns (MNB)
- apply rotations locally
- exchange "boundary" data

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}
a_{31}	a_{12}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}
a_{41}	a_{22}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}
a_{51}	a_{12}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}
a_{61}	a_{22}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}
a_{71}	a_{12}	a_{73}	a_{74}	a_{75}	a_{76}	a_{77}	a_{78}
a_{81}	a_{22}	a_{83}	a_{84}	a_{85}	a_{86}	a_{87}	a_{88}

Two-sided Jacobi - 1D distribution

1D arrangement?

- compute rotations for diagonal blocks
- brodcast rotations to all PEs (MNB)
- apply rotations locally
- exchange "boundary" data

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}
a_{31}	a_{12}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}
a_{41}	a_{22}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}
a_{51}	a_{12}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}
a_{61}	a_{22}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}
a_{71}	a_{12}	a_{73}	a_{74}	a_{75}	a_{76}	a_{77}	a_{78}
a_{81}	a_{22}	a_{83}	a_{84}	a_{85}	a_{86}	a_{87}	a_{88}

One-sided Jacobi method

Can we eliminate the broadcast communications?

YES, if A is non-negative definite.

If it is indefinite, the Jacobi process computes the SVD of a matrix.

The idea is to find orthogonal V such that columns of AV are orthogonal,

$$AV = U\Sigma, \;\; V^TV = I, \;\; U^TU = I, \;\; \Sigma = \mathrm{diag}(\sigma_1, ..., \sigma_n), \; \sigma_i \geq 0$$

If A is non-negative definite then

$$U = V, \;\; \Sigma = \Lambda$$

Otherwise $A = U\Sigma V^T$ is the SVD od A.

One-sided Jacobi method

$$AV = U\Sigma, \;\; V^TV = I, \;\; U^TU = I, \;\; \Sigma = \mathrm{diag}(\sigma_1, ..., \sigma_n), \; \sigma_i \geq 0$$

One can find V as an infinite product of Jacobi rotations,

- each rotation orthogonalizes two columns of the current $A^{(i)}$
- rotations are organized into sweeps of $\frac{(n-1)n}{2}$ rotations acting on all different pairs of columns, and
- sweeps are repeated indefinitely.
- A finite product of rotations is accumulated to give an approximate V

One-sided Jacobi method - details

Let $A^{(i)}$ be the matrix after the ith sweep.

Select columns p and q to be orthogonalized. Let

$$[\hat{a}^{(i)}_{:,p},\ \hat{a}^{(i)}_{:,q}] = [a^{(i)}_{:,p},\ a^{(i)}_{:,q}]J(p,q)$$

Find $J(p,q)$ so $\hat{a}^{(i)}_{:,p}$ and $\hat{a}^{(i)}_{:,q}$ are orthogonal. That is, we want

$$\begin{aligned}\begin{pmatrix} ||\hat{a}^{(i)}_{:,p}||^2 & 0 \\ 0 & ||\hat{a}^{(i)}_{:,q}||^2 \end{pmatrix} &= [\hat{a}^{(i)}_{:,p},\ \hat{a}^{(i)}_{:,q}]^T[\hat{a}^{(i)}_{:,p},\ \hat{a}^{(i)}_{:,q}] \\ &= J^T(p,q)\underbrace{[a^{(i)}_{:,p},\ a^{(i)}_{:,q}]^T[a^{(i)}_{:,p},\ a^{(i)}_{:,q}]}_{B^{(i)}}J(p,q)\end{aligned}$$

$B^{(i)}$ is a 2×2 symmetric matrix, so the Jacobi's idea can be used here.

One-sided Jacobi method

Select a Jacobi rotation $J(p,q)$ so

$$J(p,q)^T B^{(i)} J(p,q) = J(p,q)^T \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{1,2} & b_{2,2} \end{pmatrix} J(p,q) = \begin{pmatrix} \hat{b}_{1,1} & 0 \\ 0 & \hat{b}_{2,2} \end{pmatrix}$$

Then apply $J(p,q)$ to $A^{(i)}$,

$$[\hat{a}^{(i)}_{:,p},\ \hat{a}^{(i)}_{:,q}] = [a^{(i)}_{:,p},\ a^{(i)}_{:,q}] J(p,q)$$

Apply rotations to $\frac{(n-1)n}{2}$ different pairs of columns. Let $J^{(i)}$ be the product of these rotations, which we called a sweep.

Repeat sweeps $J^{(i)}$ infinite number of times. In the limit

$$A\left(\prod_{i=1}^{\infty} J^{(i)}\right) = AV = U\Sigma \ \Rightarrow\ A = U\Sigma V^T$$

we obtain the SVD of A.

One-sided Jacobi method

From the uniqueness of the SVD and EVD, for a symmetric non-negative definite matrix we get $U = V$,

$$A = U\Sigma V^T = U\Lambda U^T$$

1D arrangement of PEs with the BL permutations is well suited for this algorithm.

One-sided Jacobi method

In "C" matrices are arranged row-wise.

As presented, we compute

$$AV = U\Sigma, \;\; V^T V = I \tag{1}$$

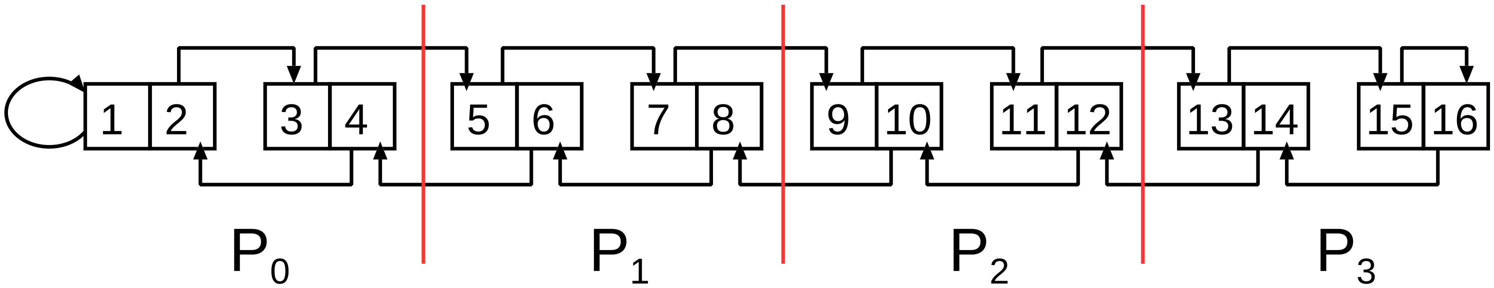
- for column-wise distribution, access to columns is not consecutive
- for row-wise distribution only part of a column resides in the same PE

Transpose (1) to get

$$V^T A^T = \Sigma U^T, \;\; V^T V = I \tag{2}$$

- orthogonalize rows
- row-wise distribution is optimal

Brent-Luk strategy



- Create and distribute data matrix $\frac{n}{P}$ rows per process
- Create (locally) the identity matrix
- Arrange processes into a conceptional 1D ring
- Compute and apply local compound rotations to local rows
- Exchange rows with neighboring processes
- Repeat $(n-1)$ times to complete a sweep
- After 8 sweeps check for convergence
  - Compute the "off" and "diagonal" norms for local columns
  - Reduce and distribute to all by `MPI_Allreduce`
  - Stop or continue

Often, one is interested only in sigular values of A, that is diagonal elements of the matrix Σ in

$$V^T A^T = \Sigma U^T, \;\; V^T V = I_n, \;\; U^T U = I_m$$

$$\sigma_{\geq} \sigma_2 \geq \cdots \geq \sigma_n \geq 0$$

Say, that we stop itereting after the kth sweep. Our transformed data is

- $A^{(k)}, \;\; V^{(k)}$
- $(V^{(k)})^T V^{(k)} = I_n$
- $A^{(k)} = [a_1^{(k)}, ..., a_n^{(n)}]$ has (approximately) orthogonal columns $a_i^{(k)}$.

$A^{(k)} = [a_1^{(k)}, ..., a_n^{(n)}]$

The singular values are (approximately) norms of $a_i^{(k)}$.

However, Jacobi method does not guarantee that

$$||a_1^{(k)}|| \geq ||a_2^{(k)}|| \geq \cdots \geq ||a_n^{(k)}||.$$

Thus the last step in Jacobi method is to compute norms $||a_i^{(k)}||$ and sort them.

Brent-Luk ordering

The major difficultes in the Brent-Luk Jacobi method are

- the implementation of data movements, and
- sorting of norms of columns of $A^{(k)}$

You are asked to address these two issues in Assignment 3.