JOHANNES KEPLER
UNIVERSITY LINZ

Stefan Amberger

ICA & RISC
amberger.stefan@gmail.com

A Parallel, In-Place, Rectangular Matrix Transpose Algorithm

Description of Algorithm and Correctness Proof

JOHANNES KEPLER
UNIVERSITY LINZ

Table of Contents

JOHANNES KEPLER
UNIVERSITY LINZ

Introduction

Rectangular Matrices

Large rectangular matrices are abundant

- Discrete Fourier transforms
- Finite element method
- Raster images in earth observation
- Computer graphics (e.g. radiosity equation)
- etc.

Current Situation in Computing

Moore's Law

- Number of transistors on a chip doubles every two years
- Maximum clock frequencies reached in 2005
- Maximum power density reached

→ multiple cores on CPUs

Memory

often the limiting factor

- medium-sized problems on mobile / embedded device
- large problem on computer

Example:
100.000 x 100.000 matrix: 75 GB

Need **parallel**, **in-place** algorithms

Rectangular Matrix Transpose

Mathematical Concept vs Implementation

Concept of Transpose

two-dimensional

$$\begin{array}{l}\textbf{double}\ \ A[M][N],\ \ B[N][M];\\ \textbf{for}\ \ (i=0;\ \ i<M;\ \ i{+}{+})\\ \quad \textbf{for}\ \ (j=0;\ \ j<N;\ \ j{+}{+})\\ \quad\quad B[j][i]\ =\ \mathrm{A}[i][j];\end{array}$$

Implementation on Computer

one-dimensional

$$\begin{array}{l}\textbf{double}\ \ A[M\cdot N],\ \ B[N\cdot M];\\ \textbf{for}\ \ (i=0;\ \ i<M;\ \ i{+}{+})\\ \quad \textbf{for}\ \ (j=0;\ \ j<N;\ \ j{+}{+})\\ \quad\quad B[j\cdot M+i]\ =\ A[i\cdot N+j];\end{array}$$

Rectangular Matrix Transpose

In-Place Transpose

In-Place Transpose of Square Matrix

using one temporary variable

M x (M-1)/2 permutation cycles

$$
\begin{array}{l}
\textbf{double}\ \ A[M \cdot M]; \\
\textbf{for}\ \ (i = 0;\ \ i < M;\ \ i{+}{+}) \\
\quad \textbf{for}\ \ (j = 0;\ \ j < i;\ \ j{+}{+}) \\
\quad\quad \mathrm{tmp}\ = \ A[j \cdot M + i]; \\
\quad\quad A[j \cdot M + i]\ = \ A[i \cdot M + j]; \\
\quad\quad A[i \cdot M + j]\ = \ \mathrm{tmp};
\end{array}
$$

In-Place Transpose of Rectangular Matrix

one-dimensional

$$
\pi(x) = \begin{cases} Mx \bmod MN - 1 & \text{if } \ x \neq MN - 1 \\ MN - 1 & \text{if } \ x = MN - 1 \end{cases}
$$

π, like every permutation, can be decomposed into disjoint, independent cycles

Rectangular Matrix Transpose

Parallel In-Place Transpose

Common Approach

Independence of Permutation Cycles

- Limited Parallelism
- Problem-dependent parallelism
- Permutation cycles are inherently serial

Our Approach

Divide and conquer

Transpose of **R**ectangular matrices, **I**n-place and in **P**arallel (**TRIP**)

- Highly parallel for all problem-sizes (see presentation 2)
- In-place
- Recursive

Description of Transpose Algorithm

TRIP Algorithm

If matrix is rectangular ***TRIP*** transposes sub-matrices, then combines the result with ***merge*** or ***split***

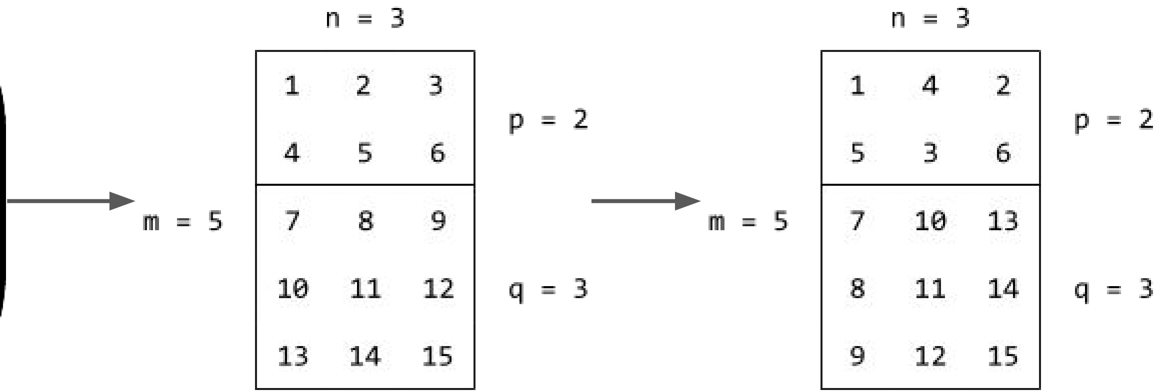
$$
\mathrm{TRIP}(A,\ m,\ n) = \begin{cases}
\mathrm{TRIP}(A(0:\lfloor \frac{m}{2} \rfloor,\ 0:n),\ \lfloor \frac{m}{2} \rfloor,\ n) \ \| \\
\mathrm{TRIP}(A(\lfloor \frac{m}{2} \rfloor : m,\ 0:n),\ \lceil \frac{m}{2} \rceil,\ n); & \text{if } m > n \\
\mathrm{merge}(\overline{A},\ \lfloor \frac{m}{2} \rfloor,\ \lceil \frac{m}{2} \rceil,\ n) \\
\\
\mathrm{TRIP}(A(0:m,\ 0:\lfloor \frac{n}{2} \rfloor),\ m,\ \lfloor \frac{n}{2} \rfloor) \ \| \\
\mathrm{TRIP}(A(0:m,\ \lfloor \frac{n}{2} \rfloor : n),\ m,\ \lceil \frac{n}{2} \rceil); & \text{if } m < n \\
\mathrm{split}(\overline{A},\ \lfloor \frac{n}{2} \rfloor,\ \lceil \frac{n}{2} \rceil,\ m) \\
\\
\mathrm{square\_transpose}(A,\ n) & \text{if } m = n
\end{cases}
$$

TRIP Example

Transpose of a Tall Matrix

original matrix is tall → it is divided by ***TRIP***
and the sub-matrices are in-place transposed

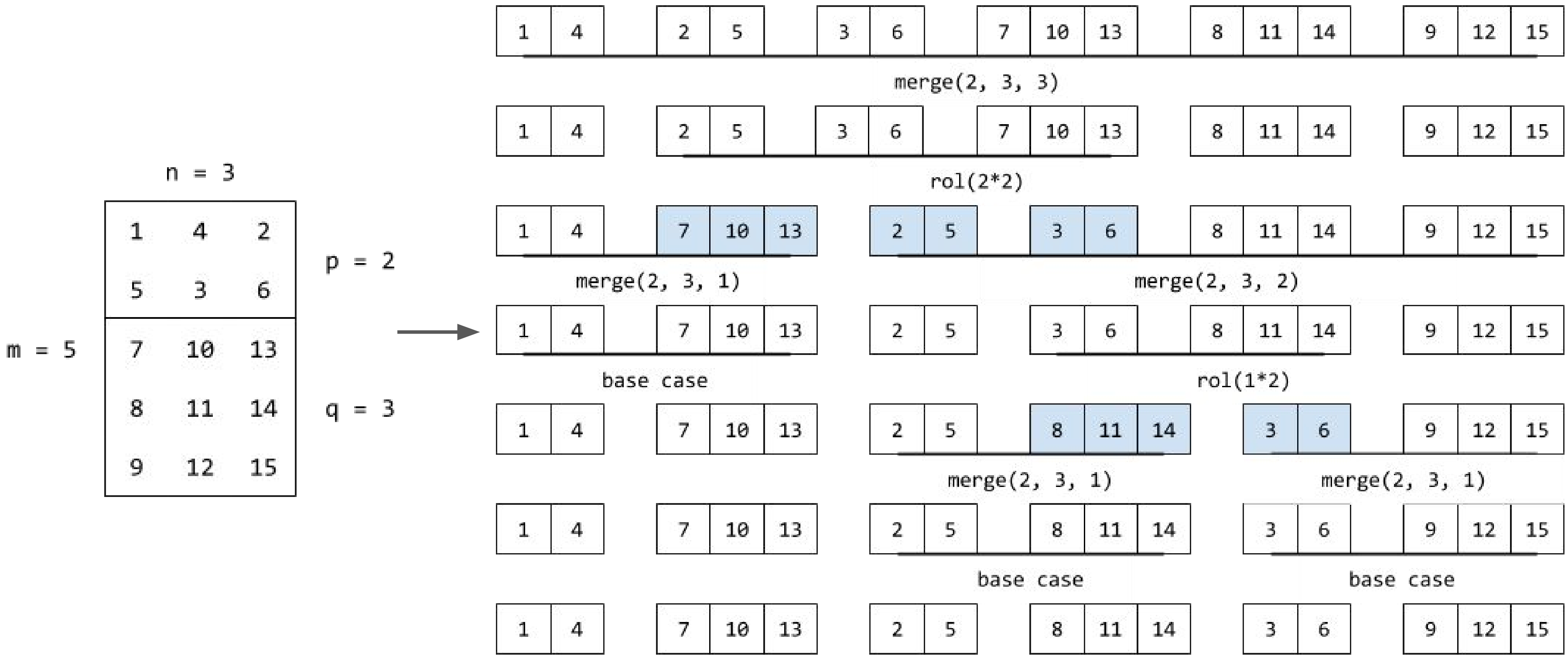
$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{pmatrix}$$



TRIP Example

Transpose of a Tall Matrix

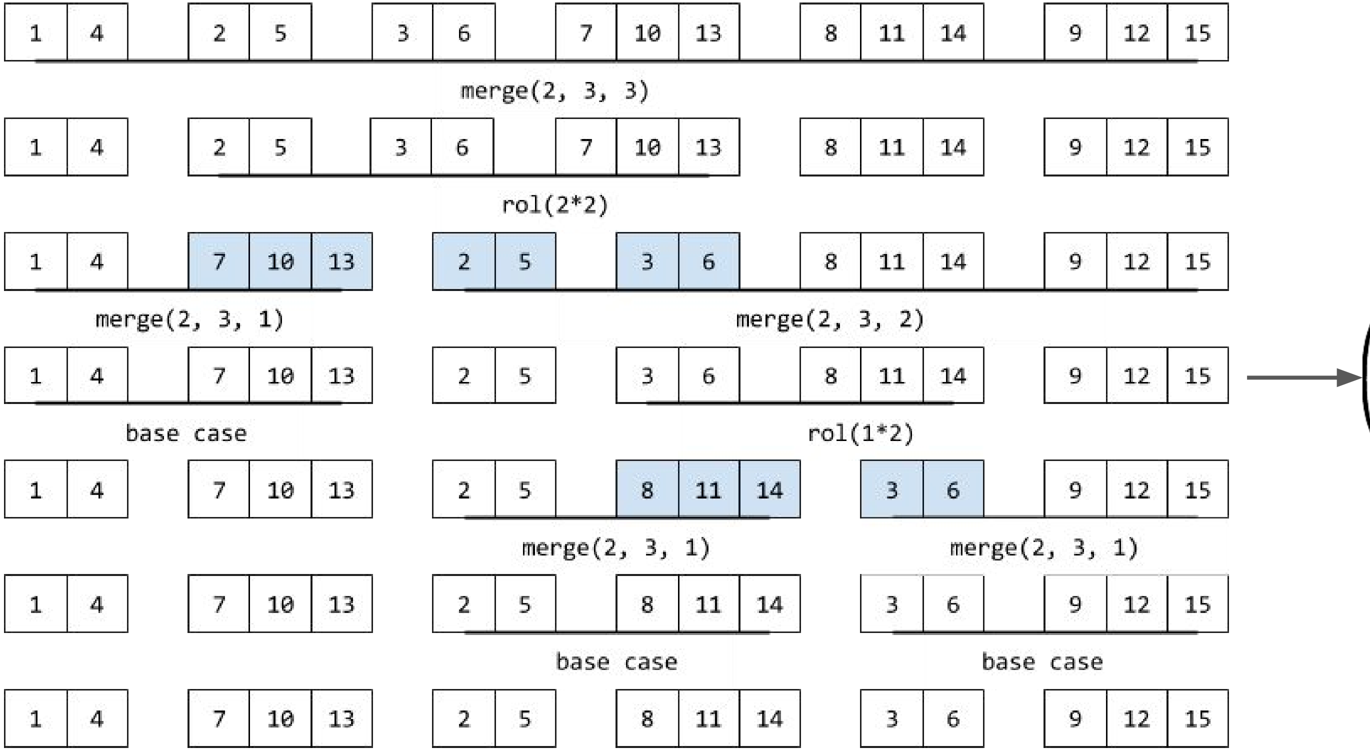
the transposed sub-matrices are combined by ***merge***



TRIP Example

Transpose of a Tall Matrix

the merged result can be reinterpreted as the transpose of the original matrix



$$\begin{pmatrix} 1 & 4 & 7 & 10 & 13 \\ 2 & 5 & 8 & 11 & 14 \\ 3 & 6 & 9 & 12 & 15 \end{pmatrix} = A^T$$

`merge` Algorithm

merge combines the transposes of sub-matrices of *tall* matrices

merge first rotates the middle part of the array, then recursively merges the left and right parts of the array

$$\operatorname{merge}(\overline{A},\ p,\ q,\ n) = \begin{cases} \operatorname{rol}(\overline{A}(\lfloor \frac{n}{2} \rfloor p : np + \lfloor \frac{n}{2} \rfloor q),\ \lceil \frac{n}{2} \rceil p); \\ \operatorname{merge}(\overline{A}(0 : \lfloor \frac{n}{2} \rfloor (p+q)),\ p,\ q,\ \lfloor \frac{n}{2} \rfloor) \parallel & \text{if } n > 1 \\ \operatorname{merge}(\overline{A}(\lfloor \frac{n}{2} \rfloor (p+q) : n(p+q)),\ p,\ q,\ \lceil \frac{n}{2} \rceil) \\ \\ \overline{A} & \text{if } n = 1 \end{cases}$$

rol(arr, k) … left rotation (circular shift) of array ***arr*** by ***k*** elements

`split` Algorithm

split combines the transposes of sub-matrices of *wide* matrices

split first recursively splits the left and right parts of the array, then rotates the middle part of the array

$$\text{split}(\overline{A},\ p,\ q,\ m) = \begin{cases} \text{split}(\overline{A}(0 : \lfloor \frac{m}{2} \rfloor (p+q)),\ p,\ q,\ \lfloor \frac{m}{2} \rfloor) \parallel \\ \text{split}(\overline{A}(\lfloor \frac{m}{2} \rfloor (p+q) : m(p+q)),\ p,\ q,\ \lceil \frac{m}{2} \rceil); & \text{if } \ m > 1 \\ \text{rol}(\overline{A}(\lfloor \frac{m}{2} \rfloor p : mp + \lfloor \frac{m}{2} \rfloor q),\ \lfloor \frac{m}{2} \rfloor q) & \\ \overline{A} & \text{if } \ m = 1 \end{cases}$$

split and ***merge*** are inverse to each other

JOHANNES KEPLER
UNIVERSITY LINZ

Correctness Proof

Correctness of ***merge***

Structure of Matrix and Transpose

Matrix is split into two parts $\longrightarrow$ transpose of Matrix is split into two parts

$$A = \underbrace{\begin{pmatrix} A_p \\ \\ A_q \end{pmatrix}}_{N} \begin{matrix} \left.\vphantom{A_p}\right\} p \\ \\ \left.\vphantom{A_q}\right\} q \end{matrix} \quad \longrightarrow \quad A^\top = \left.\begin{pmatrix} \underbrace{A_p^\top}_{p} & \underbrace{A_q^\top}_{q} \end{pmatrix}\right\} N$$

Correctness of ***merge***

Structure of Matrix after In-Place Transposition of Sub-Matrices

In-place transposition of sub-matrices results in *reshaped transposes* of sub-matrices

$$A = \underbrace{\begin{pmatrix} A_p \\ \\ A_q \end{pmatrix}}_{N} \begin{matrix} \}\, p \\ \\ \}\, q \end{matrix} \longrightarrow T = \underbrace{\begin{pmatrix} T_p \\ \\ T_q \end{pmatrix}}_{N} \begin{matrix} \}\, p \\ \\ \}\, q \end{matrix}$$

$$\overline{T_p} = \overline{A_p^\top} \quad \text{and} \quad \overline{T_q} = \overline{A_q^\top}$$

Correctness of ***merge***

Proof Sketch

Prove by induction: ***merge*** transforms T into the transpose of A

$$T = \underbrace{\begin{pmatrix} T_p \\ \\ T_q \end{pmatrix}}_{N} \begin{matrix} \}\, p \\ \\ \}\, q \end{matrix} \xrightarrow{\text{merge}} A^\top = \begin{pmatrix} \underbrace{A_p^\top}_{p} & \underbrace{A_q^\top}_{q} \end{pmatrix} \}\, N$$

$$\overline{T} = \prod_{0 \leq i < N} \left(\overline{A_p^\top}\right)_i \cdot \prod_{0 \leq i < N} \left(\overline{A_q^\top}\right)_i$$

$$\overline{A^\top} = \prod_{0 \leq i < N} \left(\left(\overline{A_p^\top}\right)_i \cdot \left(\overline{A_q^\top}\right)_i\right)$$

Correctness of ***merge***

Lemma (*merge*)

Lemma 1 *Let A be a matrix of dimension $M \times N$.*
Then

$$merge(\overline{T},\ p,\ q,\ N) = \overline{A^\top}$$

if T is composed of the reshaped transposes of A_p and A_q as described above, for $p,\ q > 0$ with $p + q = M$.

Correctness of *merge*

Proof of Lemma (*merge*)

$$\overline{T} = \prod_{0 \le i < N} \left(\overline{A_p^\top}\right)_i \cdot \prod_{0 \le i < N} \left(\overline{A_q^\top}\right)_i \xrightarrow{!} \overline{A^\top} = \prod_{0 \le i < N} \left(\left(\overline{A_p^\top}\right)_i \cdot \left(\overline{A_q^\top}\right)_i\right)$$

Base Case (k=1)

$$\overline{T} = \left(\overline{A_p^\top}\right)_0 \cdot \left(\overline{A_q^\top}\right)_0 = \overline{A_p^\top} \cdot \overline{A_q^\top} = \prod_{0 \le i < k} \left(\left(\overline{A_p^\top}\right)_i \cdot \left(\overline{A_q^\top}\right)_i\right) = \overline{A^\top}$$

Induction Hypothesis (k0 a.b.f)

`merge` transforms the array $\overline{T}$ from the shape

$$\overline{T} = \prod_{0 \le i < k_0} \left(\overline{A_p^\top}\right)_i \cdot \prod_{0 \le i < k_0} \left(\overline{A_q^\top}\right)_i$$

to the shape

$$\prod_{0 \le i < k_0} \left(\left(\overline{A_p^\top}\right)_i \cdot \left(\overline{A_q^\top}\right)_i\right) = \overline{A^\top}$$

Correctness of ***merge***

Proof of Lemma (*merge*)

$$\overline{T} = \prod_{0\leq i<N} \left(\overline{A_p^\top}\right)_i \cdot \prod_{0\leq i<N} \left(\overline{A_q^\top}\right)_i \xrightarrow{!} \overline{A^\top} = \prod_{0\leq i<N} \left(\left(\overline{A_p^\top}\right)_i \cdot \left(\overline{A_q^\top}\right)_i\right)$$

Induction Step (k0 ⟶ k0+1)

$$\overline{T} = \prod_{0\leq i<k_0+1} \left(\overline{A_p^\top}\right)_i \cdot \prod_{0\leq i<k_0+1} \left(\overline{A_q^\top}\right)_i$$

merge matches recursive case

$$\begin{aligned}&\mathrm{rol}(\overline{A}(\lfloor\tfrac{k_0+1}{2}\rfloor p : np + \lfloor\tfrac{k_0+1}{2}\rfloor q),\ \lceil\tfrac{k_0+1}{2}\rceil p);\\ &\mathrm{merge}(\overline{A}(0 : \lfloor\tfrac{k_0+1}{2}\rfloor(p+q)),\ p,\ q,\ \lfloor\tfrac{k_0+1}{2}\rfloor) \parallel\\ &\mathrm{merge}(\overline{A}(\lfloor\tfrac{k_0+1}{2}\rfloor(p+q) : k_0+1(p+q)),\ p,\ q,\ \lceil\tfrac{k_0+1}{2}\rceil)\end{aligned}$$

rol transforms T to

$$\overline{T} = \prod_{0\leq i<\lfloor\frac{k_0+1}{2}\rfloor} \left(\overline{A_p^\top}\right)_i \cdot \prod_{0\leq i<\lfloor\frac{k_0+1}{2}\rfloor} \left(\overline{A_q^\top}\right)_i \cdot \prod_{\lfloor\frac{k_0+1}{2}\rfloor\leq i<k_0+1} \left(\overline{A_p^\top}\right)_i \cdot \prod_{\lfloor\frac{k_0+1}{2}\rfloor\leq i<k_0+1} \left(\overline{A_q^\top}\right)_i$$

Correctness of ***merge***

Proof of Lemma (*merge*)

$$\overline{T} = \prod_{0 \le i < N} \left(\overline{A_p^\top}\right)_i \cdot \prod_{0 \le i < N} \left(\overline{A_q^\top}\right)_i \xrightarrow{!} \overline{A^\top} = \prod_{0 \le i < N} \left(\left(\overline{A_p^\top}\right)_i \cdot \left(\overline{A_q^\top}\right)_i\right)$$

finally: recursive ***merge*** calls on sub-arrays

$$\overline{T} = \prod_{0 \le i < \lfloor \frac{k_0+1}{2} \rfloor} \left(\overline{A_p^\top}\right)_i \cdot \prod_{0 \le i < \lfloor \frac{k_0+1}{2} \rfloor} \left(\overline{A_q^\top}\right)_i \cdot \prod_{\lfloor \frac{k_0+1}{2} \rfloor \le i < k_0+1} \left(\overline{A_p^\top}\right)_i \cdot \prod_{\lfloor \frac{k_0+1}{2} \rfloor \le i < k_0+1} \left(\overline{A_q^\top}\right)_i$$

↓ I.H.

$$\overline{T} = \prod_{0 \le i < \lfloor \frac{k_0+1}{2} \rfloor} \left(\left(\overline{A_p^\top}\right)_i \cdot \left(\overline{A_q^\top}\right)_i\right) \cdot \prod_{\lfloor \frac{k_0+1}{2} \rfloor \le i < k_0+1} \left(\left(\overline{A_p^\top}\right)_i \cdot \left(\overline{A_q^\top}\right)_i\right)$$

$$= \prod_{0 \le i < k_0+1} \left(\left(\overline{A_p^\top}\right)_i \cdot \left(\overline{A_q^\top}\right)_i\right) = \overline{A^\top}$$

$\square$

Correctness of ***split***

analogous, by induction

$$A = \left.\begin{pmatrix} \underbrace{A_p}_{p} & \underbrace{A_q}_{q} \end{pmatrix}\right\} M$$

$$T = \left.\begin{pmatrix} \underbrace{T_p}_{p} & \underbrace{T_q}_{q} \end{pmatrix}\right\} M$$

$$A^\top = \underbrace{\begin{pmatrix} A_p^\top \\ A_q^\top \end{pmatrix}}_{M} \begin{matrix} \} p \\ \} q \end{matrix}$$

Correctness of ***TRIP***

Proof by Induction

Theorem 1 *For all matrices A with dimension $M \times N$:*

$$TRIP(A,\ M,\ N) = A^\top$$

Induction on number of elements of matrix

Base Case (E=1):

$$\mathrm{TRIP}(A,\ 1,\ 1) = \mathrm{square\_transpose}(A,\ 1) = A = A^\top$$

Induction Hypothesis (E0 a.b.f.):

With E_0 a.b.f., for all matrices with dimension $M \times N$ such that $M \cdot N \leq E_0$:

$$\mathrm{TRIP}(A,\ M,\ N) = A^\top$$

Correctness of *TRIP*

Proof by Induction

Induction Step (E0 $\longrightarrow$ E0+1):

$M = N$ $\quad$ $\mathrm{TRIP}(A,\ M,\ N) = \mathrm{square\_transpose}(A,\ M) = A^\top$

$M > N$

Matrix is divided in two sub-matrices of dimension $p \times N$ and $q \times N$

with $p = \lfloor \frac{M}{2} \rfloor$ and $q = \lceil \frac{M}{2} \rceil$

Induction hypothesis applies, merge combines result.

$M < N$

Matrix is divided in two sub-matrices of dimension $M \times p$ and $M \times q$

with $p = \lfloor \frac{N}{2} \rfloor$ and $q = \lceil \frac{N}{2} \rceil$

Induction hypothesis applies, split combines result. $\square$

JOHANNES KEPLER
UNIVERSITY LINZ

Conclusions

JOHANNES KEPLER
UNIVERSITY LINZ

Conclusions

Novel Algorithm ***TRIP*** transposes rectangular matrices

- correctly
- in-place
- in highly parallel manner (see next presentation)