SIAM J. SCI. STAT. COMPUT.
Vol. 10, No. 1, pp. 18-26, January 1989

002

ON PARALLEL JACOBI ORDERINGS*

FRANKLIN T. LUK† AND HAESUN PARK‡

Abstract. This paper presents a systematic description of five well-known Jacobi orderings, using the tool of a caterpillar track. The following surprising result is obtained: under certain assumptions, the orderings are all mathematically equivalent. Three new caterpillar-track orderings are then derived, but they, in turn, are shown to be equivalent to the known schemes.

Key words. parallel Jacobi methods, caterpillar track, caterpillar tractor tread, odd-even ordering, systolic arrays

AMS(MOS) subject classifications. 65F15, 68A10

1. Introduction. The matrix-based approach to signal processing is rapidly gaining popularity (cf. the collection of papers in [3]). Four matrix operations constitute the fundamental computing requirements [14]: matrix-matrix multiplication; the QR decomposition; the singular value decomposition (SVD); and the generalized singular value decomposition (GSVD). Although these operations are costly, VLSI and parallel processing technology will soon enable us to complete the tasks in real time. A signal processing machine, called a Systolic Linear Algebra Parallel Processor, is being built at the Naval Ocean Systems Center in San Diego [5]. The machine will be hardwired to implement parallel Jacobi-like methods [7], [9] for computing the SVD and the GSVD.

The Jacobi approach is to solve a "big" $n \times n$ problem as a sequence of "small" 2×2 subproblems. To define the method, we specify an "ordering" of the $n(n-1)/2$ distinct index pairs to denote the planes in which the transformations are to take place. For serial computing, we refer to such a sequence of $n(n-1)/2$ distinct index pairs as a "sweep." A usual choice is the cyclic by-row ordering ($n=4$):

$$(1,2)(1,3)(1,4)(2,3)(2,4)(3,4).$$

Now, if successive Jacobi transformations involve disjoint planes, e.g., $(1,4)(2,3)$ in the above ordering, then these transformations can occur simultaneously. For an efficient "parallel" implementation, we want to perform as many noninteracting transformations as possible at each time "stage." An example is the following ordering of Brent and Luk [1] ($n=8$):

$$\begin{array}{ll} \text{stage 1} & (1,2)(3,4)(5,6)(7,8) \\ \text{stage 2} & (1,4)(2,6)(3,8)(5,7) \\ \text{stage 3} & (1,6)(4,8)(2,7)(3,5) \\ \text{stage 4} & (1,8)(6,7)(4,5)(2,3) \\ \text{stage 5} & (1,7)(8,5)(6,3)(4,2) \\ \text{stage 6} & (1,5)(7,3)(8,2)(6,4) \\ \text{stage 7} & (1,3)(5,2)(7,4)(8,6). \end{array}$$

For parallel computing, we refer to the set of disjoint transformations at one stage as a "compound rotation" and define a "sweep" as the minimal sequence of compound

* Received by the editors July 3, 1986; accepted for publication (in revised form) April 4, 1988. This work was supported in part by the Office of Naval Research under contract N00014-85-K-0074.

† School of Electrical Engineering, Cornell University, Ithaca, New York 14853.

‡ Department of Computer Science, Cornell University, Ithaca, New York 14853. Present address, Department of Computer Science, University of Minnesota, Minneapolis, Minnesota 55455.

rotations that includes all $n(n-1)/2$ distinct pairs. Our goal is to seek a "good" ordering that satisfies these three criteria:

(a) Each sweep is completed in n or fewer stages. Since at most $\lfloor n/2 \rfloor$ transformations can be done in parallel, a minimum of n stages is needed if n is odd ($n-1$ stages if n is even).

(b) Only nearest-neighbor connections are required for data communication.

(c) Data movement between stages is systematic.

However, we know of only one "good" ordering, viz. [1], that completes one sweep in $n-1$ stages when n is even. Many other examples (see, e.g., [12], [15], [16]) require n stages for all values of n.

As a way of finding the relation among "good" orderings, we use the tool of a caterpillar track [12], which neatly illustrates the systematic manner by which the orderings are generated. We discover that the five orderings given in [1], [4], [12], [15], [16] can all be visualized as caterpillar-track orderings. These orderings have been proposed for calculating the symmetric eigenvalue decomposition [1], [4], [12], [13], [16]; the Schur decomposition [6], [15]; the SVD decomposition [1], [2], [9]; the GSVD decomposition [7]; the CS decomposition [11]; and the QR decomposition [8]. We then consider all possible "good" caterpillar-track orderings, and derive three new ones. However, they in turn are shown to be equivalent to the five known schemes.

This paper is organized as follows. Section 2 presents a description of five well-known Jacobi orderings. The *surprising* result that (under certain assumptions) they are all mathematically equivalent is derived in § 3. Three new caterpillar-track orderings are presented and discussed in § 4. The last section contains concluding remarks and Fig. 14 summarizes the equivalent relationships.

2. Various orderings. We present five different orderings of n indices for Jacobi methods. With two exceptions, they all require n stages to complete a sweep. The exceptions are the Brent–Luk ordering for even n (respectively, $n-1$) stages, and the Chen–Irani ordering for odd n (respectively, $n+1$ stages). Hence the latter is not a "good" ordering for odd n. To illustrate each ordering, we show one sweep of the case $n=6$ for even n, and of the case $n=5$ for odd n.

(i) Modi and Pryce [12] describe a "mobile" scheme in terms of a "caterpillar track." There are $2n$ places on the track to store the n indices in every other position. The indices move one step per stage in a counterclockwise direction, as shown in Fig. 1. Each index on the lower track is paired with the closest index to its right on the upper track. The compound rotations for each stage are shown in Fig. 2.

For $n=5$ we get the track as in Fig. 3 and the corresponding compound rotations as in Fig. 4.

(ii) Whiteside et al. [16] present an ordering in terms of a "caterpillar tractor tread." In this ordering, the n indices are stored on a caterpillar tractor tread of length n that rolls along a roadbed of processors. Each processor has access to the pair of

```
· ← 2 ← · ← 4 ← · ← 6      2 ← · ← 4 ← · ← 6 ← ·      · ← 4 ← · ← 6 ← · ← 5
↓                   ↑  →   ↓                   ↑  →   ↓                   ↑  →
1 → · → 3 → · → 5 → ·      · → 1 → · → 3 → · → 5      2 → · → 1 → · → 3 → ·

4 ← · ← 6 ← · ← 5 ← ·      · ← 6 ← · ← 5 ← · ← 3      6 ← · ← 5 ← · ← 3 ← ·
↓                   ↑  →   ↓                   ↑  →   ↓                   ↑
· → 2 → · → 1 → · → 3      4 → · → 2 → · → 1 → ·      · → 4 → · → 2 → · → 1
```

FIG. 1. *Caterpillar-track for $n=6$.*

1. (1, 2)(3, 4)(5, 6)
2. 2(1, 4)(3, 6)5
3. (2, 4)(1, 6)(3, 5)
4. 4(2, 6)(1, 5)3
5. (4, 6)(2, 5)(1, 3)
6. 6(4, 5)(2, 3)1

FIG. 2. *Caterpillar-track ordering for $n = 6$.*

```
·←2←·←4←·      2←·←4←·←5      ·←4←·←5←·
↓        ↑  →  ↓        ↑  →  ↓        ↑  →
1→·→3→·→5      ·→1→·→3→·      2→·→1→·→3

4←·←5←·←3      ·←5←·←3←·
↓        ↑  →  ↓        ↑
·→2→·→1→·      4→·→2→·→1
```

FIG. 3. *Caterpillar track for $n = 5$.*

1. (1, 2)(3, 4)5
2. 2(1, 4)(3, 5)
3. (2, 4)(1, 5)3
4. 4(2, 5)(1, 3)
5. (4, 5)(2, 3)1

FIG. 4. *Caterpillar-track ordering for $n = 5$.*

indices directly above it, on the lower and upper parts of the tread. The upper tread of the tractor advances one step per time stage and the lower tread stays stationary on the ground. Figure 5 gives an example for $n = 6$. Corresponding to the tread, the compound rotations for each time stage are exactly as those given in Fig. 2.

For n odd, the tread proceeds as in Fig. 6. Again the compound rotations are the same as those in Fig. 4.

(iii) Stewart [15] introduces the "odd-even" ordering. The paired indices are exchanged for the next stage, and pairings start with either the first or the second index at alternate time stages. So the ordering proceeds in exactly the same fashion as the caterpillar-track ordering portrayed in Figs. 2 and 4.

```
2 4 6     4 6     4 6 5     6 5     6 5 3     5 3
      → 2     5 →       → 4     3 →       → 6     1
1 3 5     1 3     2 1 3     2 1     4 2 1     4 2
```

FIG. 5. *Caterpillar tractor tread for $n = 6$.*

```
2 4        4 5   4 5        5 3   5 3
    5 → 2     →      3 → 4     →      1
1 3        1 3   2 1        2 1   4 2
```

FIG. 6. *Caterpillar tractor tread for $n = 5$.*

(iv) Brent and Luk [1] present an ordering that is often used in round-robin chess tournaments. For n even, this ordering requires only $n-1$ stages to complete a sweep. See Fig. 7 for an example and Fig. 8 for the corresponding compound rotations.

Our presentation here looks different from the original description in [1] that is illustrated in Fig. 9. However, one version is just the other applied to the inverted list of indices $(n, n-1, \cdots, 2, 1)$. For n odd, we simply add a dummy index ϕ to return to the even case. See Figs. 10 and 11 for illustrations.

(v) The final scheme is due to Chen and Irani [4]. They consider only the case of an even n. Like the "odd-even" ordering, pairings of indices start with the first or the second index at alternate time stages. However, the indices are exchanged after every *two* stages, according to the permutation

$$\sigma=(2,4,1,6,3,8,5,\cdots,n-5,n,n-3,n-1),$$

```
2←4  6     4←5  6     5←3  6     3←1  6     1←2  6
↓   ↖   →  ↓   ↖   →  ↓   ↖   →  ↓   ↖   →  ↓   ↖
1→3→5      2→1→3      4→2→1      5→4→2      3→5→4
```

FIG. 7. *Round-robin exchanges for $n=6$.*

1. (1, 2)(3, 4)(5, 6)
2. (2, 4)(1, 5)(3, 6)
3. (4, 5)(2, 3)(1, 6)
4. (5, 3)(4, 1)(2, 6)
5. (3, 1)(5, 2)(4, 6)

FIG. 8. *Brent–Luk ordering for $n=6$.*

```
2←4←6      4←6←5      6←5←3      5←3←2      3←2←4
↘   ↑  →   ↘   ↑  →   ↘   ↑  →   ↘   ↑  →   ↘   ↑
1  3→5     1  2→3     1  4→2     1  6→4     1  5→6
```

FIG. 9. *Original round-robin exchanges for $n=6$.*

```
2←4  ϕ     4←5  ϕ     5←3  ϕ     3←1  ϕ     1←2  ϕ
↓   ↖   →  ↓   ↖   →  ↓   ↖   →  ↓   ↖   →  ↓   ↖
1←3→5      2→1→3      4→2→1      5→4→2      3→5→4
```

FIG. 10. *Round-robin exchanges for $n=5$.*

1. (1, 2)(3, 4)5
2. (2, 4)(1, 5)3
3. (4, 5)(2, 3)1
4. (5, 3)(4, 1)2
5. (3, 1)(5, 2)4

FIG. 11. *Brent–Luk ordering for $n=5$.*

whose action is better explained by the following figure:

$$\begin{array}{lcl} 2 \leftarrow 4 \leftarrow 6 \leftarrow \cdots \leftarrow n \\ \downarrow \qquad\qquad\qquad \uparrow \\ 1 \rightarrow 3 \rightarrow 5 \rightarrow \cdots \rightarrow n-1 \end{array}$$

Figure 12 presents the compound rotations. We may extend the ordering to the odd n case by appending a dummy index ϕ. See Fig. 13.

3. Equivalence. From the descriptions and examples in § 2, we see that the five orderings are intimately related. This section is devoted to the proof of their mathematical equivalence. For any n, let us choose the same initial stage for each ordering:

$$(1,2)(3,4)(5,6)\cdots(n-1,n) \qquad \text{if } n \text{ is even,}$$

$$(1,2)(3,4)(5,6)\cdots(n-2,n-1)n \quad \text{if } n \text{ is odd.}$$

DEFINITION. Consider the two sequences S_1 and S_2 of compound rotations generated in one sweep of the orderings O_1 and O_2. We say that O_1 and O_2 are "identical" if $S_1 = S_2$, and that they are "stage-equivalent" if S_1 is a permutation of S_2. Note that two stage-equivalent orderings may not share the same numerical properties.

Since the caterpillar track makes it easy to visualize the orderings geometrically, we centralize our discussion on this theoretical machine. Modi and Pryce [12] have given the following result.

LEMMA 1. *The caterpillar track generates the same compound rotation and completes one sweep in every n stages.* □

Consequently, when proving stage-equivalence of some ordering to the caterpillar-track ordering, we need only consider any n consecutive stages of the track. We say that two stages are "equal" if they correspond to the same compound rotation.

LEMMA 2. *The caterpillar-track ordering is identical to the odd-even ordering.*

Proof. Suppose that n is even (the proof is similar for odd n). We use induction on the stage number. By assumption, the first stages are identical. Suppose that we get

1. (1, 2)(3, 4)(5, 6)
2. 1(2, 3)(4, 5)6
3. (2, 4)(1, 6)(3, 5)
4. 2(4, 1)(6, 3)5
5. (4, 6)(2, 5)(1, 3)
6. 4(6, 2)(5, 1)3

FIG. 12. *Chen-Irani ordering for $n = 6$.*

1. $(1, 2)(3, 4)(5, \phi)$
2. $1(2, 3)(4, 5)\phi$
3. $(2, 4)(1, \phi)(3, 5)$
4. $2(4, 1)(\phi, 3)5$
5. $(4, \phi)(2, 5)(1, 3)$
6. $4(\phi, 2)(5, 1)3$

FIG. 13. *Chen-Irani ordering for $n = 5$.*

the same $(2k+1)$st stages:

$$\begin{array}{l} \cdot \leftarrow a_2 \leftarrow \cdot \leftarrow a_4 \leftarrow \cdots \leftarrow \cdot \leftarrow a_n \\ \downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow \\ a_1 \rightarrow \cdot \rightarrow a_3 \rightarrow \cdot \rightarrow \cdots \rightarrow a_{n-1} \rightarrow \cdot \end{array} \qquad (a_1, a_2)(a_3, a_4)\cdots(a_{n-1}, a_n).$$

Then the $(2k+2)$nd and $(2k+3)$rd stages are the following:

$$\begin{array}{l} a_2 \leftarrow \cdot \leftarrow a_4 \leftarrow \cdot \leftarrow \cdots \leftarrow a_n \leftarrow . \\ \downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow \\ \cdot \rightarrow a_1 \rightarrow \cdot \rightarrow a_3 \rightarrow \cdots \rightarrow \cdot \rightarrow a_{n-1} \end{array} \qquad a_2(a_1, a_4)\cdots(a_{n-3}, a_n)a_{n-1}, \text{ and}$$

$$\begin{array}{l} \cdot \leftarrow a_4 \leftarrow \cdot \leftarrow a_6 \leftarrow \cdots \leftarrow \cdot \leftarrow a_{n-1} \\ \downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow \\ a_2 \rightarrow \cdot \rightarrow a_1 \rightarrow \cdot \rightarrow \cdots \rightarrow a_{n-3} \rightarrow . \end{array} \qquad (a_2, a_4)(a_1, a_6)\cdots(a_{n-3}, a_{n-1}). \qquad \square$$

LEMMA 3. *The caterpillar tractor tread ordering is identical to the odd-even ordering.*

Proof. The proof is similar to that of Lemma 2. $\square$

We now generalize the concept of a caterpillar track.

DEFINITION. An (o, e) caterpillar track is a caterpillar track that moves o steps at every odd time stage and e steps at every even time stage, both in the counterclockwise direction.

Thus, the original caterpillar track is a $(1, 1)$ caterpillar track. It can be shown that if $-2 \leqq o$, $e \leqq 3$, then we can implement the (o, e) caterpillar-track ordering on a multiprocessor array with only nearest-neighbor connections. In this section we discuss two specific choices: $(o, e) = (2, 2)$, and $(-1, 3)$. What follow are two obvious properties.

FACT 1. The $(2, 2)$ caterpillar track repeats itself in every n stages, and its ith stage equals the $(2i-1)$st stage of the $(1, 1)$ caterpillar track, for $i = 1, 2, \cdots, n$.

FACT 2. Consider the $(-1, 3)$ caterpillar track. Its ith stage equals the ith stage of the $(1, 1)$ caterpillar track when i is odd, and equals the $(i-2)$nd stage of the $(1, 1)$ track when i is even.

Note by Lemma 1 that stages $n+2, n+4, \cdots, 2n-1$ of the $(1, 1)$ track equal stages $2, 4, \cdots, n-1$ of the $(1, 1)$ track, respectively. If n and i are both even, then the index $i-2+n$ is even and stage i of the $(-1, 3)$ track equals stage $i-2+n$ of the $(1, 1)$ track. From these observations and Facts 1 and 2, we get the following two lemmas.

LEMMA 4. *For any odd n, the* $(2, 2)$ *caterpillar-track ordering is stage-equivalent to the* $(1, 1)$ *caterpillar-track ordering.* $\square$

LEMMA 5. *For any even n, the* $(-1, 3)$ *caterpillar-track ordering is stage-equivalent to the* $(1, 1)$ *caterpillar-track ordering.* $\square$

Let us relate the $(2, 2)$ and $(-1, 3)$ tracks to the two remaining orderings.

LEMMA 6. *For any odd n, the* $(2, 2)$ *caterpillar track generates the same ordering as the Brent–Luk scheme.*

Proof. We use induction for the proof. Stages 1 are the same by assumption. Suppose that the kth stages are equal, i.e.,

$$\begin{array}{l} \cdot \leftarrow a_2 \leftarrow \cdot \leftarrow a_4 \leftarrow \cdots \leftarrow a_{n-1} \leftarrow \cdot \\ \downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow \\ a_1 \rightarrow \cdot \rightarrow a_3 \rightarrow \cdot \rightarrow \cdots \rightarrow \cdot \rightarrow a_n \end{array} \qquad \begin{array}{l} a_2 \leftarrow a_4 \leftarrow \cdots \leftarrow a_{n-1} \quad \phi \\ \downarrow \qquad\qquad\qquad\qquad \nwarrow \\ a_1 \rightarrow a_3 \rightarrow \cdots \rightarrow a_{n-2} \rightarrow a_n \end{array} .$$

Then the $(k+1)$st stages become the following:

$$\begin{array}{l} \cdot \leftarrow a_4 \leftarrow \cdot \leftarrow a_6 \leftarrow \cdots \leftarrow a_n \leftarrow \cdot \\ \downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow \\ a_2 \rightarrow \cdot \rightarrow a_1 \rightarrow \cdot \rightarrow \cdots \rightarrow \cdot \rightarrow a_{n-2} \end{array} \qquad \begin{array}{l} a_4 \leftarrow a_6 \leftarrow \cdots \leftarrow a_n \quad \phi \\ \downarrow \qquad\qquad\qquad\qquad \nwarrow \\ a_2 \rightarrow a_1 \rightarrow \cdots \rightarrow a_{n-4} \rightarrow a_{n-2} \end{array} . \qquad \square$$

LEMMA 7. *For any even n, the* $(-1, 3)$ *caterpillar track generates the same ordering as the Chen–Irani scheme.*

Proof. We use induction on the stage number. Stages 1 are equal by assumption. Assume that the $(2k+1)$st stages are equal, i.e.,

$$\begin{array}{l} \cdot \leftarrow a_2 \leftarrow \cdot \leftarrow a_4 \leftarrow \cdots \leftarrow \quad \cdot \quad \leftarrow a_n \\ \downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow \\ a_1 \rightarrow \cdot \rightarrow a_3 \rightarrow \cdot \rightarrow \cdots \rightarrow a_{n-1} \rightarrow \cdot \end{array} \qquad (a_1, a_2)(a_3, a_4) \cdots (a_{n-1}, a_n).$$

Then the $(2k+2)$nd and $(2k+3)$rd stages of the two schemes are the following:

$$\begin{array}{l} a_1 \leftarrow \cdot \leftarrow a_2 \leftarrow \cdot \leftarrow \cdots \leftarrow a_{n-2} \leftarrow \cdot \\ \downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow \\ \cdot \rightarrow a_3 \rightarrow \cdot \rightarrow a_5 \rightarrow \cdots \rightarrow \quad \cdot \quad \rightarrow a_n \end{array} \qquad a_1(a_2, a_3) \cdots (a_{n-2}, a_{n-1})a_n$$

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

$$\begin{array}{l} \cdot \leftarrow a_4 \leftarrow \cdot \leftarrow a_6 \leftarrow \cdots \leftarrow \quad \cdot \quad \leftarrow a_{n-1} \\ \downarrow \qquad\qquad\qquad\qquad\qquad\qquad \uparrow \\ a_2 \rightarrow \cdot \rightarrow a_1 \rightarrow \cdot \rightarrow \cdots \rightarrow a_{n-3} \rightarrow \end{array} \qquad (a_2, a_4)(a_1, a_6) \cdots (a_{n-3}, a_{n-1}). \qquad \square$$

We summarize our results in a theorem.

THEOREM 1. *The* $(1, 1)$ *caterpillar track, the caterpillar tractor tread, and the odd-even orderings are all identical. For n odd, the three orderings are stage-equivalent to the Brent-Luk ordering, while for n even, they are stage-equivalent to the Chen-Irani ordering.* $\square$

4. New orderings. First, we give the conditions under which an (o, e) caterpillar track generates a Jacobi ordering.

THEOREM 2. *In order that an* (o, e) *caterpillar track generate all the distinct index pairs, the three parameters must satisfy either one of the following two relations*:

$$\gcd(n, o+e) = 1, \quad \textit{or}$$

$$\gcd(n, o+e) = 2 \quad \textit{and } o \textit{ is odd.}$$

The track then completes a sweep in at most $2n$ *stages.*

Proof. We assume that the first stages are the same. Then stage $(2i+1)$ of the (o, e) track equals stage $1+(o+e)i$ of the regular track, for $i = 0, 1, \cdots, n-1$. But if n and $o+e$ are relatively prime, then the set $S = \{(o+e)i \bmod n, \text{ for } i = 1, 2, \cdots, n-1\}$ equals the set $\{1, 2, \cdots, n-1\}$. Now if $\gcd(n, o+e) = 2$, then the set S equals only $\{2, 4, \cdots, n-2\}$. Thus, we need an odd o so that we can use the set of even stages of the (o, e) track. If $\gcd(n, o+e) \geqq 3$, then this patchwork fails and not all index pairs can be generated. $\square$

COROLLARY. *The* (o, o) *and* $(1, 1)$ *caterpillar-track orderings are stage-equivalent if*

$$\gcd(n, o) = 1. \qquad \square$$

We now consider only those orderings that satisfy $-2 \leqq o, e \leqq 3$, a condition imposed by the nearest-neighbor communication restriction. We are particularly interested in caterpillar tracks that complete one sweep in n stages. Clearly, there is no need to discuss the cases where $o = 0$, $e = 0$, or $o + e \leqq 0$.

(i) $o+e = 1, 3, 5$. According to Theorem 2, the ordering generates all possible pairs in at most $2n$ stages so long as $\gcd(n, o+e) = 1$. The caterpillar track finishes one sweep in n stages when $n = 2|o|$. For example, the $(2, -1)$ caterpillar track requires more than n stages when $n > 4$ because its fifth stage equals its second stage.

(ii) $o+e=2$. The three possibilities are: (1, 1), (−1, 3), and (3, −1). The first two are described in detail in the previous sections, and the third ordering is obviously stage-equivalent to the second when n is even.

(iii) $o+e=4$. The three choices are: (2, 2), (1, 3), and (3, 1). The first choice is the Brent–Luk ordering, and the other two satisfy the conditions of Theorem 2 for any n that is not a multiple of 4.

(iv) $o+e=6$. From the corollary, the (3, 3) caterpillar track completes one sweep in n stages for any n that is not a multiple of 3.

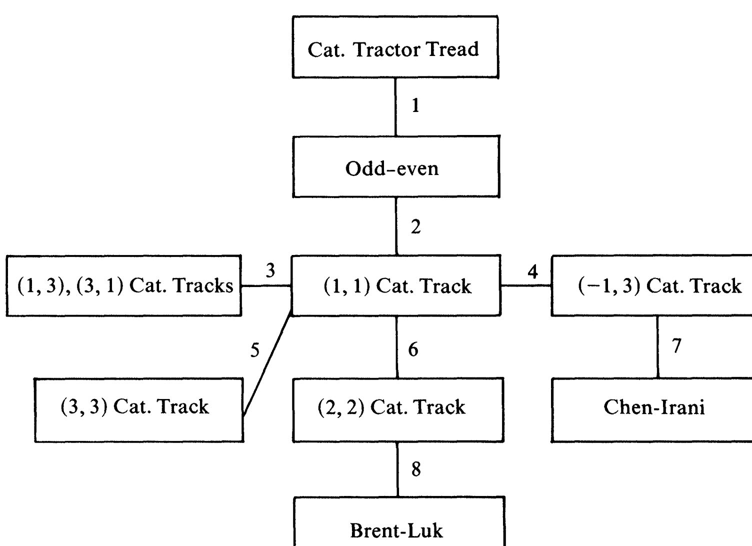
We present the equivalence properties of the three new orderings in the next two lemmas. Their proofs are like those in the previous section.

LEMMA 8. *The* (1, 3), (3, 1), *and* (1, 1) *caterpillar-track orderings are stage-equivalent if*

$$n \bmod 4 = 2. \qquad \square$$

LEMMA 9. *The* (3, 3) *and* (1, 1) *caterpillar-track orderings are stage-equivalent if n is not a multiple of* 3. □

5. Concluding remarks. In this paper, we study five well-known parallel Jacobi orderings and derive three new ones via the tool of a caterpillar track. We show that all eight orderings are either identical or stage-equivalent, and that they constitute all possible "good" orderings that can be generated by a caterpillar track. Figure 14 summarizes the equivalence relations from which the convergence of various Jacobi SVD methods can be derived [10].



1. Identical by Lemma 3.
2. Identical by Lemma 2.
3. Stage-equivalent by Lemma 8 when $n \bmod 4 = 2$.
4. Stage-equivalent by Lemma 5 when n is even.
5. Stage-equivalent by Lemma 9 when $n \bmod 3 \neq 0$.
6. Stage-equivalent by Lemma 4 when n is odd.
7. Identical by Lemma 7 when n is even.
8. Identical by Lemma 6 when n is odd.

FIG. 14. *Relationships among the orderings.*

REFERENCES

[1] R. P. BRENT AND F. T. LUK, *The solution of singular-value and symmetric eigenvalue problems on multiprocessor arrays*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 69–84.

[2] R. P. BRENT, F. T. LUK, AND C. F. VAN LOAN, *Computation of the singular value decomposition using mesh-connected processors*, J. VLSI Comput. Syst., 1 (1985), pp. 242–270.

[3] K. BROMLEY AND J. M. SPEISER, *Signal processing algorithms, architectures, and applications*, in Tutorial 31, SPIE 27th Annual International Technical Symposium, San Diego, CA, 1983.

[4] K-W. CHEN AND K. B. IRANI, *A Jacobi algorithm and its implementation on parallel computers*, in Proc. 18th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, 1980, pp. 564–573.

[5] B. L. DRAKE, F. T. LUK, J. M. SPEISER, AND J. J. SYMANSKI, SLAPP: *A Systolic Linear Algebra Parallel Processor*, IEEE Computer, 20 (1987), pp. 45–49.

[6] P. J. EBERLEIN, *On the Schur decomposition of a matrix for parallel computation*, IEEE Trans. Comput., 36 (1987), pp. 167–174.

[7] F. T. LUK, *A parallel algorithm for computing the generalized singular value decomposition*, J. Parallel Distrib. Comput., 2 (1985), pp. 250–260.

[8] ———, *A rotation algorithm for computing the* QR-*decomposition*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 452–459.

[9] ———, *A triangular processor array for computing singular values*, J. Linear Algebra Appl., 77 (1986), pp. 259–273.

[10] F. T. LUK AND H. PARK, *A proof of convergence for two parallel Jacobi* SVD *algorithms*, IEEE Trans. Comput., 38 (1989), to appear.

[11] F. T. LUK AND S. QIAO, *Computing the* CS-*decomposition on systolic arrays*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1121–1125.

[12] J. J. MODI AND J. D. PRYCE, *Efficient implementation of Jacobi's diagonalization method on the* DAP, Numer. Math., 46 (1985), pp. 443–454.

[13] A. H. SAMEH, *On Jacobi and Jacobi-like algorithms for a parallel computer*, Math. Comput., 25 (1971), pp. 579–590.

[14] J. M. SPEISER AND H. J. WHITEHOUSE, *A review of signal processing with systolic arrays*, in Real Time Signal Processing VI, Proc. SPIE 431, 1983, pp. 2–6.

[15] G. W. STEWART, *A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 853–864.

[16] R. A. WHITESIDE, N. S. OSTLUND, AND P. G. HIBBARD, *A parallel Jacobi diagonalization algorithm for a loop multiple processor system*, IEEE Trans. Comput., 33 (1984), pp. 409–413.