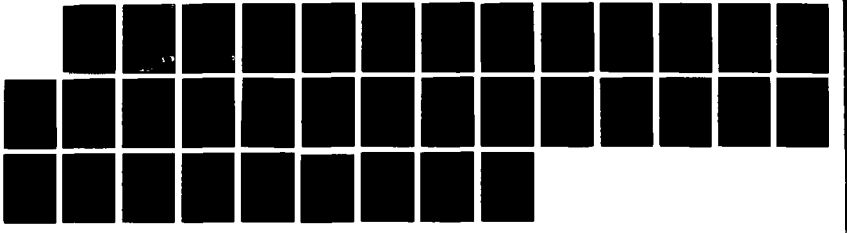
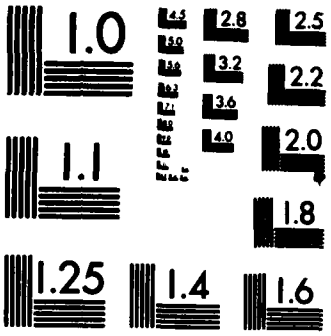
AD-A181 613 TRANSFORMATION OF BROADCASTING INTO PIPELINING(U) YALE 1/1
UNIV NEW HAVEN CT DEPT OF COMPUTER SCIENCE
Y WONG ET AL. JUN 87 YALEU/DCS/RR-544 DAAL03-86-K-0158
UNCLASSIFIED F/G 12/7 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Transformation of Broadcasting into Pipelining

Yiwan Wong[1] and Jean-Marc Delosme[2]

Research Report YALEU/DCS/RR-544
June, 1987

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

5

A-

ABSTRACT

A characteristic shared by many computation intensive algorithms is the repeated usage of a few data values in a sequence of computations. An efficient parallel implementation of these data dependences often requires the simultaneous transfer, or *broadcasting*, of the data values to all the processors that need them. Unfortunately, direct realization of this broadcasting operation on VLSI processor arrays, especially on systolic arrays, usually results in severe performance degradation.

A technique for decomposing broadcasting dependences into propagation dependences at the algorithm level is presented in this paper. Such propagation dependences, when physically realized, result in pipelining. The determination of a feasible propagation scheme is formulated as a linear algebra problem. ~~We prove~~ that all broadcastings can be decomposed into propagations and we propose a systematic method for finding such decompositions.

It is proven

Transformation of Broadcasting into Pipelining

Yiwan Wong[1] and Jean-Marc Delosme[2]

Research Report YALEU/DCS/RR-544

June, 1987

[1]Department of Computer Science

[2]Department of Electrical Engineering

The work presented in this paper was supported by the Army Research Office under contract number DAAL03-86-K-0158.

1. INTRODUCTION

A major obstacle to the efficient implementation of algorithms on processor arrays with distributed memory is that random data access in constant time is not possible. The data access time in a processor array depends on the physical distance between the processors which generate and use the data, which is a function of the global interconnection pattern used for data routing.

A characteristic shared by many computation intensive algorithms is the repeated usage of input data or of intermediate results in sequences of computations, see, *e.g.*, the algorithms in [1]. While this kind of data dependence poses no problem in a uniprocessor environment, a minimal time parallel implementation on an ideal machine with negligible communication cost would often require the simultaneous transfer, or *broadcasting*, of data to all the processors which use them in subsequent computations. Simulation of the broadcasting operation, however, usually results in severe performance degradation in array implementations because the I/O bandwidth between the processor array and the host is limited, and the *fan-out* degree of the processors is bounded.

There are two commonly known solutions to the broadcasting problem on processor arrays: *architectural transformation* and *algorithmic transformation*. Architectural transformation is based on the *retiming* procedure proposed by Leiserson and Saxe [2] for inserting delays in the broadcasting paths on the array, effectively replacing data broadcasting with *pipelining*, see the example in [3]. While this approach is systematic and well formulated, the derivation of the broadcast-free array has a complexity proportional to the product of the number of processors and the number of interconnections in the array. The algorithmic transformation approach, on the other hand, eliminates broadcasting at the algorithm level. A powerful type of transformation is the decomposition of broadcasting dependences into *propagation* dependences [4], which, when physically realized, result in pipelining. The decomposition approach deals with broadcasting at a higher level than architectural transformation and hence, may result in parallel implementations of better performance.

To illustrate the decomposition approach, consider the 1-D recursive filtering algorithm. The output signal y is computed from the input signal x and two sets of weights w and r according to the equation

$$y_i = \sum_{j=1}^{k} w_j x_{i-j} + \sum_{j=1}^{k} r_j y_{i-j} \quad , \quad k \leq i \leq n \quad , \quad k \ll n \quad .$$

The equation may be expressed in *single assignment* form:

$$y(i,j) = y(i,j+1) + \bar{w}(i-1,j)\bar{x}(i-1,j-1) + \bar{r}(i-1,j)y(i-j,1)$$
$$\bar{w}(i,j) = \bar{w}(i-1,j)$$
$$\bar{x}(i,j) = \bar{x}(i-1,j-1)$$
$$\bar{r}(i,j) = \bar{r}(i-1,j),$$

with boundary conditions

$$y(i,k+1) = 0,\ \bar{w}(k-1,j) = w_j,\ \bar{r}(k-1,j) = r_j,$$
$$\bar{x}(k-1,j) = x_{k-j-1},\ \bar{x}(i,0) = x_i \quad .$$

The final results are $y_i = y(i,1)$. Each computed value $y(l,1)$ is needed by several computations

of $y(i, j)$; the value of $y(l, 1)$ has to be made available to all the computations $y(i, j)$, where $i-j = l$. The data dependences associated to this broadcasting are shown in Figure 1a, in which $k = 6$ and $l = 8$.

To eliminate the broadcasting, we introduce a *propagation variable*, $\overline{y}$, defined by the conditional recurrence

$$\overline{y}(i, j) = \begin{cases} \overline{y}(i-1, j-1) & \text{if } j \neq 1 \\ y(i-1, j) & \text{otherwise} \end{cases},$$

and replace the dependence of $y(i, j)$ on $y(i-j, 1)$ with the dependence on $\overline{y}(i, j)$,

$$y(i, j) = y(i, j+1) + \overline{w}(i-1, j)\overline{x}(i-1, j-1) + \overline{r}(i-1, j)\overline{y}(i, j).$$

The function of $\overline{y}(i, j)$ is to transfer, or propagate, the value of $y(i-j, 1)$ to all the computations that use it. For example, the direct dependence of computation $y(12,4)$ on the value of $y(8,1)$ is now replaced by the dependence path

$$y(12,4) \rightarrow \overline{y}(12,4) \rightarrow \overline{y}(11,3) \rightarrow \overline{y}(10,2) \rightarrow \overline{y}(9,1) \rightarrow y(8,1) .$$

Note that all the $y(i,j)$'s along the dependence path need value $y(8,1)$ and that the propagation variable $\overline{y}$ brings them that value. With propagation, the broadcasting dependences in Figure 1a are decomposed into the *propagation dependences* shown in Figure 1b.

The idea of replacing broadcasting with propagation is not new. Many innovative processor array implementations have been derived through the use of such a transformation: algebraic path

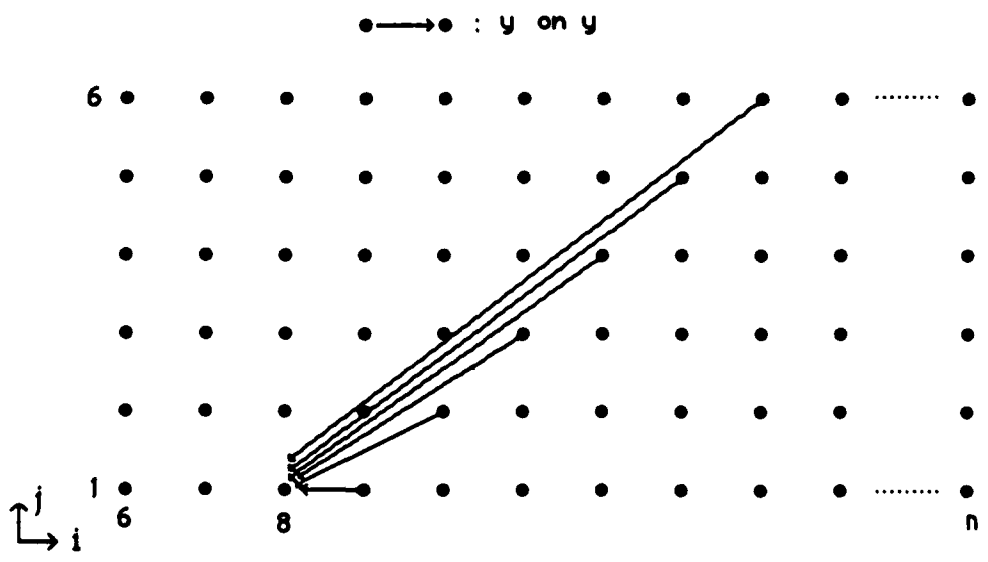


Figure 1a : Broadcasting dependences on the value of $y(8,1)$ in the 1-D recursive filtering example.

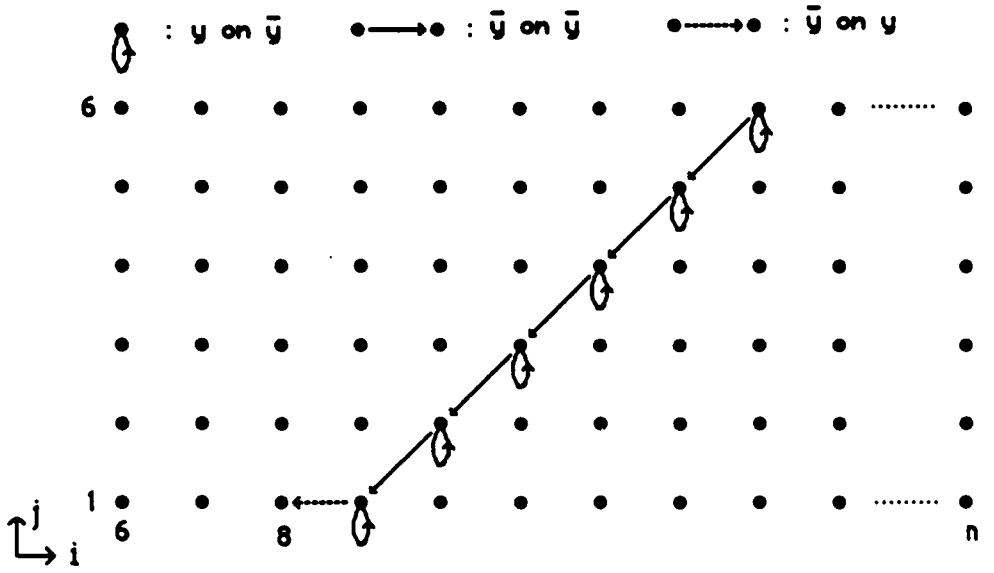


Figure 1b : Decomposing the broadcasting dependences shown in Figure 1a into propagation dependences.

problem [5], LDU decomposition of matrices [4], dynamic programming [6], *etc*. Without a rigid formalism, however, algorithm transformations must be carried out through tedious data dependence analyses, which can handle in practice only the simplest forms of broadcasting.

In this paper, we formulate the decomposition of broadcasting dependences into propagation dependences and present a systematic method for determining the appropriate decomposition. In Section 2, the algorithm model, upon which our discussion is based, is presented. In Section 3 propagation is defined mathematically and all broadcastings are proved to be decomposable into propagations. In Section 4, a special case of broadcasting, in which the data to be broadcast come from external sources (input data), is considered. Open problems and current research in the area are discussed in Section 5.

2. ALGORITHM MODEL

The Linear Dependence Algorithms (LDA's), are a generalization of the Regular Iterative Algorithms (RIA's) [4] and Uniform Recurrence Equations (URE's) [7]. An LDA is expressed as a set of r recurrence equations defined within an n-dimensional index space called the *computation domain*, $\mathbf{C}$. The data dependences among the computations of an LDA are linear (or affine) functions of the coordinates of the computation points,

$$a_i(P) = f_i(a_{j_1}(d_{ij_1}(P)), a_{j_2}(d_{ij_2}(P)), \cdots); 1 \leq i \leq r, j_m \in \mathbf{N}^+, P \in \mathbf{C}_i \subset \mathbf{C}.$$

The *computed variables*, a_i, $1 \leq i \leq r$, represent the quantities computed by the LDA, and the f_i's are the functions evaluated in order to compute these values. Note that, unlike RIA's, in which the full set of the r computed variables is computed at every index point of the computation domain (except, possibly, for index points on the boundary of the domain), LDA's feature selective computations of subsets of these r variables at different index locations throughout the domain: variable a_i is computed at all the points in the subset, $\mathbf{C}_i$, of $\mathbf{C}$. The *input variables*, a_j, $j > r$, of the LDA are data values supplied to the LDA from external sources.

The *dependence mapping*, d_{ij}, is an *affine mapping* which defines the dependence between the indices of the computed variable a_i, and the indices of the variable a_j; the computation of $a_i(P)$ requires the use of the quantity $a_j(d_{ij}(P))$. A dependence mapping comprises a linear part, B, which is an $m \times n$ integral matrix, and a constant part, Δ_{ij}, which is a constant integral column m-vector,

$$d_{ij}(P) = B_{ij}P - \Delta_{ij},$$

where the value of m is n if $j \leq r$, *i.e.*, if a_j is also a computed variable, and satisfies $1 \leq m \leq n$, otherwise. Here, we assume that all the computed variables are *fully indexed*, *i.e.*, each of them is indexed by an n-vector. This can be achieved by introducing one or more additional indices to those computed variables which are not already fully-indexed. There are simple heuristics for adding these missing indices [8].

The *dependence vector*, $v_{ij}(P)$, associated to the dependence mapping d_{ij}, $i,j \leq r$, at an index point P at which a_i is computed, is given by

$$v_{ij}(P) = P - d_{ij}(P) = (I_n - B_{ij})P + \Delta_{ij}$$

where I_n is the order n identity matrix. A dependence mapping d_{ij}, $i,j \leq r$ is a *translation* if $B_{ij} = I_n$ and then

$$v_{ij}(P) = \Delta_{ij} \quad \forall P \in \mathbf{C}_i.$$

The RIA's and URE's are particular cases of LDA's in which all the dependence mappings are translations and $\mathbf{C}_i = \mathbf{C}$, $1 \leq i \leq r$. The dependence vectors are constant in these cases.

3. BROADCASTING

In this section, we formally define the decomposition technique for the elimination of broadcasting in LDA's. Two kinds of propagation are considered, *elementary* and *composite* propagation. In elementary propagation, the propagation dependence paths are constructed from elementary (or canonical) vectors. We present a necessary and sufficient condition under which a broadcasting dependence is decomposable into elementary propagation. A polynomial time algorithm for constructing the propagation variables is given. Composite propagation is the generalization of elementary propagation, with propagation dependence vectors that are not necessary canonical. We show that all broadcasting dependences can be decomposed into composite propagations. In this section, we consider only the broadcasting of the values of computed variables in a LDA. The broadcasting of input variables, which is a special case, will be discussed in Section 4.

3.1. Elementary Propagation

An LDA requires broadcasting if the value of a computed variable is needed by several other computations of the same or a different variable. This can be detected easily from the recurrence definitions of the LDA:

Definition:

A dependence mapping d_{ij}, where a_i and a_j are computed variables of the LDA, is a *broadcasting* dependence mapping if the linear part of d_{ij}, the $n \times n$ integral matrix B_{ij}, is *rank deficient.*

The computation of a_i at point P uses the value of a_j computed at point $d_{ij}(P)$. If B_{ij} is rank deficient then there exist P_1, P_2, $\cdots$, such that $d_{ij}(P_1) = d_{ij}(P_2) = \cdots = P_0$, say. The computations of a_i at points P_1, P_2, $\cdots$, all need the value of a_j computed at point P_0, hence the value of $a_j(P_0)$ must be broadcast to all these points. For convenience, we drop the subscripts and rename a_i as a, a_j as b, and d_{ij} as d.

The basic idea of elementary propagation is as follows. We associate to each broadcasting dependence mapping, d, a set of *propagation variables*, $\bar{a}_{k_1}, \bar{a}_{k_2}, \cdots, \bar{a}_{k_n}$, where $\{k_1, k_2, \cdots, k_n\}$ is a permutation of the set of integers $\{1, \cdots, n\}$, for transferring the broadcast value, b, to the index points at which b is needed. Each of these propagation variables is responsible for propagation along one elementary direction, $\vec{e}_{k_q}$, where $\vec{e}_{k_q}$ is a canonical basis vector. The portion of dependence path in the direction of $\vec{e}_{k_q}$ is called the q th *section* of the overall propagation dependence path.

Let P_0 be the index point at which the value of b to be broadcast is computed and assume that the computation of a at P_1 needs the value of $b(P_0)$, hence, $d(P_1) = P_0$. The original data dependence

$$a(P_1) \rightarrow b(d(P_1)) = b(P_0)$$

is replaced by a sequence of propagation dependences starting with

$$a(P_1) \rightarrow \bar{a}_{k_1}(P_1) .$$

The variable $\bar{a}_{k_1}$ performs propagation in the direction of $\vec{e}_{k_1}$, the k_1th elementary direction. The objective is to create a dependence path from the current index point, P_1 in this case, to the index point P_2 whose k_1th component equals the k_1th component of P_0, the index point at which the required b is computed. The first section of the propagation path is therefore,

$$a(P_1) \rightarrow \bar{a}_{k_1}(P_1) \rightarrow \bar{a}_{k_1}(P_1+\epsilon_1\vec{e}_{k_1}) \rightarrow \bar{a}_{k_1}(P_1+2\epsilon_1\vec{e}_{k_1}) \rightarrow \cdots \rightarrow \bar{a}_{k_1}(P_2-\epsilon_1\vec{e}_{k_1}) ,$$

where $\epsilon_1 = 1$ or -1 or 0 is the *sense* of propagation in the direction $\vec{e}_{k_1}$. The second section of the propagation path, which extends in the direction of $\vec{e}_{k_2}$, begins at P_2 with propagation variable $\bar{a}_{k_2}$. Propagation in this direction is continued until the index point P_3 whose k_2th component equals that of P_0, is reached,

$$a(P_1) \rightarrow \bar{a}_{k_1}(P_1) \rightarrow \cdots \bar{a}_{k_1}(P_2-\epsilon_1\vec{e}_{k_1}) \rightarrow \bar{a}_{k_2}(P_2) \rightarrow \bar{a}_{k_2}(P_2 + \epsilon_2\vec{e}_{k_2}) \rightarrow \cdots \rightarrow \bar{a}_{k_2}(P_3 - \epsilon_2\vec{e}_{k_2}) .$$

Since propagation in the second section does not change the k_1th component, the k_1th component of P_3 equals the k_1th component of P_0. Repeating this procedure for each of the elementary directions, the final dependence path is

$$a(P_1) \rightarrow \bar{a}_{k_1}(P_1) \rightarrow \cdots \bar{a}_{k_2}(P_2) \rightarrow \cdots \rightarrow \bar{a}_{k_n}(P_n) \rightarrow \cdots \rightarrow \bar{a}_{k_n}(P_{n+1} - \epsilon_n \vec{e}_{k_n}) .$$

If we let b be $\bar{a}_{k_{n+1}}$, this dependence path is a decomposition of the original broadcasting dependence of $a(P_1)$ on $b(P_0)$, since $P_{n+1} \equiv P_0$. The introduction of the propagation variables does not modify the precedence relationship between the computations of a and b at index points within the computation domain, hence the computability of the transformed LDA is guaranteed. The feasibility of the elementary propagation scheme, however, depends on the existence of the appropriate propagation variables for performing the required propagations.

In general, the propagation variables, $\bar{a}_{k_q}$, $1 \leq q \leq n$, are defined by *conditional recurrences* of the form

$$\bar{a}_{k_q}(P) = \begin{cases} \bar{a}_{k_q}(P + \vec{e}_{k_q}) & \text{if } (P + \vec{e}_{k_q})\rangle_{k_q} < p_{0k_q} \\ \bar{a}_{k_q}(P - \vec{e}_{k_q}) & \text{if } (P - \vec{e}_{k_q})\rangle_{k_q} > p_{0k_q} \\ \bar{a}_{k_{q+1}}(P + \vec{e}_{k_q}) & \text{if } (P + \vec{e}_{k_q})\rangle_{k_q} = p_{0k_q} \\ \bar{a}_{k_{q+1}}(P - \vec{e}_{k_q}) & \text{if } (P - \vec{e}_{k_q})\rangle_{k_q} = p_{0k_q} \\ \bar{a}_{k_{q+1}}(P) & \text{if } (P)\rangle_{k_q} = p_{0k_q} \end{cases}, \quad \forall P , \tag{1}$$

where $(P + \vec{x})\rangle_{k_q}$ is the k_q th component of the vector sum $(P + \vec{x})$, and p_{0k_q} is the k_q th component of P_0, the index point at which the required data value b is located. The first two conditionals determine the sense (either positive or negative) of propagation and the remaining conditionals define the *terminating conditions* of the q th section. The value of p_{0k_q} depends on the starting index point, P_1, of the dependence path. Since P_1 is arbitrary, and hence P_0 is arbitrary, it must be possible to determine the value of p_{0k_q} from the current index point P and the constant part, Δ, of d for the propagation scheme to work with only one set of propagation variables for all P_1's.

From (1), the index points P on the q th section of the propagation path have the property that their k_r th component is equal to the k_r th component of P_0 for $r < q$, and to the k_r th component of P_1 for $r > q$,

$$p_{k_r} = \begin{cases} p_{0k_r} & , r < q \\ p_{1k_r} & , r > q \end{cases} .$$

These components are called the *invariant components* of the q th section of the propagation path. Hence, if d is such that the k_q th component of $P_0 (= d(P_1))$ can be determined from Δ and the invariant components of the q th section, $1 \leq q \leq n$, the elementary propagation scheme is applicable. In other words, if there exists an *order of propagation*, i.e., an assignment of $1, \cdots, n$ to $k_1, k_2, \cdots, k_n$, such that the above-mentioned condition is satisfied, then the broadcasting dependence can be decomposed into elementary propagation with the given ordering. To minimize the computation cost of the conditionals, the k_q th component of P_0 is constrained to be a linear function of the invariant components and of Δ.

The condition under which a broadcasting dependence mapping is decomposable into elementary propagation can be formulated in simple mathematical terms. Assume for the moment that the i th row of B is different from $\vec{e}_i^{\,T}$ for all i. The broadcasting dependence is

$$P_1 \rightarrow d(P_1) = BP_1 - \Delta = P_0.$$

Suppose the *natural ordering* is a feasible propagation order, *i.e.*, $k_q = q$, $1 \leq q \leq n$. The assumption that the q th component of P_0 is a linear function of Δ, of the first $q-1$ components of P_0 and of the last $n-q$ components of P_1 within the q th section, implies that

$$p_{0q} = -\sum_{j=1}^{q-1} l_{qj} p_{0q} + \sum_{j=q+1}^{n} u_{qj} p_{1q} - \sum_{j=1}^{n} x_{qj} \delta_j ,$$

where l_{qj}, u_{qj} and x_{qj} are rational numbers. Writing the above expression for $q = 1, \cdots, n$ in matrix notation and grouping the components of P_0 and P_1 on opposite sides, we get

$$\begin{bmatrix} 1 & & & \\ l_{21} & 1 & & 0 \\ l_{31} & l_{32} & \cdot & \\ \cdot & \cdot & \cdot & \\ \cdot & \cdot & & \cdot \\ l_{n1} & l_{n2} & \cdots & l_{nn-1} \quad 1 \end{bmatrix} \begin{bmatrix} p_{01} \\ p_{02} \\ \cdot \\ \cdot \\ \cdot \\ p_{0n} \end{bmatrix} = \begin{bmatrix} 0 & u_{12} & u_{13} & \cdots & u_{1n} \\ & \cdot & u_{23} & & \cdot \\ & & \cdot & & \cdot \\ & 0 & & \cdot & \cdot \\ & & & & u_{n-1n} \\ & & & & 0 \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ \cdot \\ \cdot \\ \cdot \\ p_{1n} \end{bmatrix} - X\Delta .$$

This implies that $P_0 = L^{-1}(UP_1 - X\Delta)$, hence

$$B = L^{-1}U \ , \text{ and } X = L \ ,$$

where L is *unit lower* triangular and U is *strictly upper* triangular. Since the actual value of Δ is unimportant, for simplicity, we will assume from now on that Δ is a null vector, hence $d(P) = BP$.

Theorem 3.1:

A broadcasting dependence mapping d is *decomposable* into elementary propagation using m propagation variables if there exists an order n permutation matrix Q such that

$$Q^{-1}BQ = \begin{bmatrix} L^{-1}U & H \\ 0 & I_{n-m} \end{bmatrix}$$

where L is an $m \times m$ unit lower triangular matrix, U is an $m \times m$ strictly upper triangular matrix, H is an $m \times (n-m)$ matrix, and I_{n-m} is the identity matrix of order $n-m$.

Proof:

First, note that a *symmetric permutation* of rows and columns of B by Q is equivalent to reordering the indices of the computation n-space. Second, if the i th row of B is equal to $\vec{e}_i^{\,T}$ then $P_1 \rightarrow P_0 = d(P_1)$ implies that $p_{0i} = p_{1i}$, hence propagation in the direction of $\vec{e}_i$ is unnecessary. Thus, from our previous analysis, if B can be symmetrically permuted into the stated form, Q defines an order of propagation and the appropriate elementary propagation may be carried out by m propagation variables, one for each of the first m elementary directions selected by the column permutation Q and defined with conditional recurrences as in (1). □

Suppose B is decomposable into elementary propagations, then

$$BP_1 = P_0 \Rightarrow Q^{-1}BQ\,(Q^{-1}P_1) = Q^{-1}P_0\,. \tag{2}$$

Partition vector $Q^{-1}P$ into $\begin{bmatrix}\vec{p}_1\\ \vec{p}_2\end{bmatrix}$, where

$$\vec{p}_1 = (p_{k_1}, p_{k_2}, \cdots, p_{k_m})^{\mathrm{T}}$$
$$\vec{p}_2 = (p_{k_{m+1}}, p_{k_{m+2}}, \cdots, p_{k_n})^{\mathrm{T}}.$$

By Theorem 3.1, equation (2) is equivalent to

$$\begin{cases} L^{-1}U\vec{p}_{11} + H\vec{p}_{12} = \vec{p}_{01} \\ \vec{p}_{12} = \vec{p}_{02} \end{cases}$$

The first equation implies that $U\vec{p}_{11} + LH\vec{p}_{12} = L\vec{p}_{01}$, hence,

$$p_{0k_q} = \vec{u}_q^{\mathrm{T}}\vec{p}_{11} - (\vec{l}_q - \vec{\eta}_q)^{\mathrm{T}}\vec{p}_{01} + \vec{l}_q^{\mathrm{T}}H\vec{p}_{12}\ ,\ 1 \le q \le m\ , \tag{3}$$

where $L = (\vec{l}_1\ \vec{l}_2\ \cdots\ \vec{l}_m)^{\mathrm{T}}$, $U = (\vec{u}_1\ \vec{u}_2\ \cdots\ \vec{u}_m)^{\mathrm{T}}$, and $\vec{\eta}_q$ is the m-vector with all 0's except for the q th component which is a 1. Since L is unit lower triangular, the expression $(\vec{l}_q - \vec{\eta}_q)^{\mathrm{T}}\vec{p}_{01}$ is a linear function of the first $q-1$ components of $Q^{-1}P_0$. Similarly, since U is strictly upper triangular, $(\vec{u}_q^{\mathrm{T}}\vec{p}_{11} + \vec{l}_q^{\mathrm{T}}H\vec{p}_{12})$ is a linear function of the last $n-q$ components of $Q^{-1}P_1$. Hence, the q th component, p_{0k_q}, of $Q^{-1}P_0$ can be determined from the invariant components of the q th section. Thus, once the matrices Q, L, U and H are known, the conditionals for defining the propagation variables, $\bar{a}_{k_q}$, $1 \le q \le m$, can be derived from (1) and (3).

Example 1:

Let the indices of the four-dimensional computation space be (i_1, i_2, i_3, i_4). Consider the broadcasting dependence

$$a\,(i_1,i_2,i_3,i_4) \rightarrow b\,(i_2+2i_3, 2i_2+4i_3+i_4, i_2+i_3+i_4, i_2+i_3+i_4),\ \text{or,}$$

$$B = \begin{bmatrix} 0 & 1 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

The broadcasting is decomposable into elementary propagation with order of propagation $\{k_1, k_2, k_3, k_4\} = \{1, 3, 2, 4\}$, as defined by the following matrices :

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},\ L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix},\ U = \begin{bmatrix} 0 & 2 & 1 & 0 \\ 0 & 0 & \frac{1}{2} & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

For example, the 3rd component of P_0 can be determined from any index point P on the second section of the propagation path using equation (3) :

$$p_{03} = \vec{u}_2^{\mathrm{T}} Q^{-1} P - (\vec{l}_2 - \vec{\eta}_2)^{\mathrm{T}} Q^{-1} P = \frac{i_1 + i_2}{2} + i_4 .$$

Similarly,

$$p_{01} = i_2 + 2i_3 , \quad P \in \text{section 1}$$
$$p_{02} = 2i_1 + i_4 , \quad P \in \text{section 3}$$
$$p_{04} = i_3 , \quad P \in \text{section 4}$$

Thus, for example, the definition of the propagation variable $\bar{a}_3$ is

$$\bar{a}_3(P) = \begin{cases} \bar{a}_3(P + \vec{e}_{k_2}) & \text{if } (i_3+1) < \dfrac{i_1 + i_2}{2} + i_4 \\ \bar{a}_3(P - \vec{e}_{k_2}) & \text{if } (i_3-1) > \dfrac{i_1 + i_2}{2} + i_4 \\ \bar{a}_2(P + \vec{e}_{k_2}) & \text{if } (i_3+1) = \dfrac{i_1 + i_2}{2} + i_4 \\ \bar{a}_2(P - \vec{e}_{k_2}) & \text{if } (i_3-1) = \dfrac{i_1 + i_2}{2} + i_4 \\ \bar{a}_2(P) & \text{if } i_3 = \dfrac{i_1 + i_2}{2} + i_4 \end{cases} ,$$

which is valid for all index points P within the computation domain.

For example, the dependence $a(1,1,1,2) \rightarrow b(3,8,4,4)$ is now replaced by a propagation dependence path consisting of four sections:

$$a(1,1,1,2) \rightarrow \bar{a}_1(1,1,1,2) \rightarrow \bar{a}_1(2,1,1,2) \rightarrow$$
$$\bar{a}_3(3,1,1,2) \rightarrow \bar{a}_3(3,1,2,2) \rightarrow \bar{a}_3(3,1,3,2) \rightarrow$$
$$\bar{a}_2(3,1,4,2) \rightarrow \bar{a}_2(3,2,4,2) \rightarrow \bar{a}_2(3,3,4,2) \rightarrow \cdots \rightarrow \bar{a}_2(3,7,4,2) \rightarrow$$
$$\bar{a}_4(3,8,4,2) \rightarrow \bar{a}_4(3,8,4,3) \rightarrow b(3,8,4,4).$$

For the computation of $a(3,1,1,2)$, which depends on the same value $b(3,8,4,4)$, the propagation dependence path starts as

$$a(3,1,1,2) \rightarrow \bar{a}_1(3,1,1,2) \rightarrow \bar{a}_3(3,1,1,2)$$

and then joins the path shown above. It can be verified easily that the definitions of the propagation variables do trace out these propagation paths. □

Despite the simple mathematical formulation given in Theorem 3.1, determining whether a broadcasting dependence mapping is elementary decomposable and finding a feasible order of propagation are non-trivial tasks. A simple-minded approach is to find Q by exhaustive search, possibly with the help of heuristics; if such Q exists then the broadcasting mapping is decomposable. This approach is not very appealing as the worst case complexity is exponential. Fortunately, as will be shown next, decomposability of a broadcasting mapping into elementary propagation is related directly to the non-zero structure in B, the linear part of the broadcasting mapping. The proof of this claim provides a polynomial time algorithm for constructing a feasible order of propagation.

For the following discussion, we assume that the ith row of B is different from $\vec{e}_i^{\,T}$ for all i. If this is not true, we can find a permutation matrix R such that $R^{-1}BR = \begin{bmatrix} B_{11} & B_{12} \\ 0 & I_{n-m} \end{bmatrix}$, where the $m \times m$ matrix B_{11} satisfies the assumption, and let $B \leftarrow B_{11}$, $n \leftarrow m$.

Let $G(B) = (V,E)$ be the *directed graph*, or the *digraph*, associated with the matrix $B = \left[\beta_{ij}\right]$, with a vertex $v_i \in V$ for each of the n indices of the n-space and an edge, $\xi_{ij} \in E$, from v_i to v_j if β_{ij} is non-zero. The connectivity of the digraph $G(B)$ corresponds to the non-zero structure in B. Let $F(G(B)) = \{\ v_i \mid v_i \in V,\ \text{indegree}(v_i) = 0\ \}$ be the set of *free vertices* in the graph $G(B)$, and $\overline{V} = V - F(G(B))$. We say that the matrix B satisfies the *reachability condition* if its associated digraph $G(B)$ has the property that every vertex in $\overline{V}$ is *reachable* from at least one of the free vertices, *i.e.*, if for each $v_j \in \overline{V}$, there exists at least one $v_i \in F(G(B))$ such that there is a directed path from v_i to v_j in $G(B)$. The digraph associated with the broadcasting dependence mapping in Example 1 is shown in Figure 2.

The following theorem constitutes the main result of this subsection:

Theorem 3.2:

The following statements are equivalent :

(1) B satisfies the reachability condition,

(2) there exists a permutation matrix Q such that

$$Q^{-1}BQ = L^{-1}U\ ,$$

where L is a unit lower triangular matrix and U is a strictly upper triangular matrix.

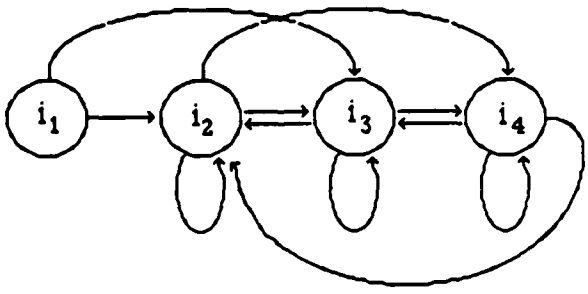


Figure 2 : The directed graph, G(B) associated with the broadcasting dependence B of Example 1. The vertices are labelled by the indices they represent. B satisfies the reachability condition.

Proof

not (1) $\Rightarrow$ not (2):

Suppose (1) does not hold and let $v_r \in \bar{V}$ be a vertex that is not reachable from the free vertices; v_r necessarily belongs to a strongly connected component in which all the vertices are not reachable from the free vertices. Without loss of generality, let this strongly connected component contain the set of vertices $\bar{V}_1 = \{v_r, v_{r+1}, \cdots, v_n\}$, *i.e.*, assume that the matrix B is of the form

$$B = \begin{bmatrix} B_{11} & 0 \\ B_{21} & B_{22} \end{bmatrix}, \; B_{22} = \begin{bmatrix} \beta_{rr} & \cdots & \beta_{rn} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \beta_{nr} & \cdots & \beta_{nn} \end{bmatrix},$$

where the non-zero structure of B_{22} corresponds to the connectivity of the strongly connected component formed by the vertices in $\bar{V}_1$.

Suppose (2) holds. Since U is strictly upper triangular, Q must be such that the diagonal and the subdiagonal elements of $Q^{-1}BQ$ can be eliminated using the elements on its first superdiagonal as pivots. Consider the diagonal element β_{rr} in the submatrix B_{22}, there are three possible choices of Q :

(*i*) Leave the position of β_{rr} unchanged. Statement (2) does not hold unless $\beta_{jr} = 0$ for all $j \geq r$, which contradicts the assumption that vertices in $\bar{V}_1$ are strongly connected.

(*ii*) Reorder elements within B_{22}. Suppose the j th row and j th column, $r < j \leq n$, are swapped with the r th row and r th column. The r th column becomes

$$(\beta_{1j} \; \beta_{2j} \; \cdots \; \beta_{r-1,j} \; \beta_{jj} \; \beta_{r+1,j} \; \cdots \; \beta_{j-1,j} \; \beta_{rj} \; \beta_{j+1,j} \; \cdots \; \beta_{nj})^{\mathrm{T}},$$

which is equal to

$$(0 \; 0 \; \cdots \; 0 \; \beta_{jj} \; \beta_{r+1,j} \; \cdots \; \beta_{j-1,j} \; \beta_{rj} \; \beta_{j+1,j} \; \cdots \; \beta_{nj})^{\mathrm{T}}.$$

As in case (*i*), statement (2) holds only if $\beta_{ij} = 0$ for all $i \geq r$, which contradicts the assumption that the vertices in $\bar{V}_1$ are strongly connected.

(*iii*) Move β_{rr} out of B_{22}. Suppose the r th row and r th column are swapped with the i th row and i th column, $i < r$. The i th column becomes

$$(\beta_{1r} \; \beta_{2r} \; \cdots \; \beta_{i-1,r} \; \beta_{rr} \; \beta_{i+1,r} \; \cdots \; \beta_{r-1,r} \; \beta_{ir} \; \beta_{r+1,r} \; \cdots \; \beta_{nr})^{\mathrm{T}},$$

which is equal to

$$(0 \; 0 \; \cdots \; 0 \; \beta_{rr} \; 0 \; \cdots \; 0 \; \beta_{ir} \; \beta_{r+1,r} \; \cdots \; \beta_{nr})^{\mathrm{T}}.$$

Since $\beta_{jr} = 0$, $j \leq i-1$, statement (2) does not hold unless $\beta_{jr} = 0$ for all j, a contradiction to the assumption that the vertices in $\bar{V}_1$ are strongly connected.

Thus, not (1) implies not (2).

To prove the converse, we need the following lemma :

Lemma 3.1 :

Let $F(G(B))$ be $\{v_1, v_2, \cdots, v_f\}$ and $\overline{V}$ be $\{v_{f+1}, v_{f+2}, \cdots, v_n\}$, $1 \leq f \leq n$. If B satisfies the reachability condition then there exists a pivot, β_{ij}, $1 \leq i \leq f$, $f < j \leq n$, such that B^e, the order $n-1$ matrix obtained from B by

(i) eliminating the jth column of B using β_{ij} as pivot, and,

(ii) deleting the ith row and ith column,

also satisfies the reachability condition.

Proof of Lemma 3.1 :

By induction on the cardinality, l, of $\overline{V}$, the set of non-free vertices in $G(B)$.

Base case : $l = 1$,

$$B_1 = \begin{bmatrix} 0 & \cdots & 0 & \beta_{1,f+1} \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & \beta_{f,f+1} \\ 0 & \cdots & 0 & \beta_{f+1,f+1} \end{bmatrix}.$$

Since v_{f+1} is reachable from at least one of the vertices in $F(G(B_1))$, there exists a $\beta_{i,f+1}$, $1 \leq i \leq f$, which is non-zero. Using this element as pivot, we have

$$B_1^e = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix},$$

which satisfies the reachability condition. Therefore, the base case is true.

Suppose B_k, $k > 1$, satisfies the reachability condition. By picking an appropriate permutation R of order $(f+k)$, it is always possible to rearrange the rows and columns of B_k such that the principal submatrix, B_{k-1}, obtained from deleting the last row and last column of $R^{-1}BR$ satisfies the reachability condition. To do this, first find the depth-first spanning forest [9] of the graph $G(B_k)$ and, supposing v_u, $f < u \leq f+k$, is a non-isolated leaf node, form R to swap the uth column and $(f+k)$th column. For clarity of presentation, we now assume that R is the identity matrix.

We complete the induction step of the proof by showing that either the element, β_{rq}, $1 \leq r \leq f$, $f < q \leq k-1$, which is a feasible pivot for the case $l = k-1$, or the element β_{rk} is a feasible pivot for $l = k$.

Suppose the proposition is true for $l = k-1$, that is, there exists a pivot β_{ij} such that B_{k-1}^e satisfies the reachability condition. Without loss of generality, let β_{1q}, $f+1 \leq q \leq k-1$, be such a pivot, then,

$$B_{k-1}^{e} = \begin{bmatrix} 0 & \cdots & 0 & \hat{\beta}_{2,f+1} & \cdots & \hat{\beta}_{2,q-1} & 0 & \hat{\beta}_{2,q+1} & \cdots & \hat{\beta}_{2,k-1} \\ \cdot & & & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & & & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & & & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ \cdot & & & \cdot & & \cdot & \cdot & \cdot & & \cdot \\ 0 & \cdots & 0 & \hat{\beta}_{k-1,f+1} & \cdots & \hat{\beta}_{k-1,q-1} & 0 & \hat{\beta}_{k-1,q+1} & \cdots & \hat{\beta}_{k-1,k-1} \end{bmatrix}$$

where $\hat{\beta}_{ij} = \beta_{ij} - \alpha_i \beta_{1j}$, with $\alpha_i = \dfrac{\beta_{iq}}{\beta_{1q}}$, $2 \leq i \leq (k-1)$, $(f+1) \leq j \leq (k-1)$.

Now, if we pick β_{1q} as pivot for B_k, we have,

$$B_k^{e} = \begin{bmatrix} & & & \hat{\beta}_{2k} \\ & & & \hat{\beta}_{3k} \\ & & & \cdot \\ & B_{k-1}^{e} & & \cdot \\ & & & \cdot \\ & & & \hat{\beta}_{k-1,k} \\ 0 & \cdots & 0 \ \hat{\beta}_{k,f+1} \ \cdots \ \hat{\beta}_{k,q-1} & 0 \ \hat{\beta}_{k,q+1} \ \cdots & \hat{\beta}_{kk} \end{bmatrix},$$

where $\hat{\beta}_{ij} = \beta_{ij} - \alpha_i \beta_{1j}$ with $\alpha_k = \dfrac{\beta_{kq}}{\beta_{1q}}$, $i = k$ or $j = k$. Since, by assumption, B_{k-1}^{e} satisfies the reachability condition, B_k^{e} does not satisfy the reachability condition only if

$\hat{\beta}_{ik} = 0$, $2 \leq i \leq (k-1)$, and,

$\hat{\beta}_{kk} \neq 0$,

where the inequality ensures that v_k does not belong to $F(G(B_k^{e}))$. Thus, if β_{1q} is not a feasible pivot for B_k, then

$$B_k = \begin{bmatrix} & & & \beta_{1k} \\ & & & \alpha_2\beta_{1k} \\ & B_{k-1} & & \alpha_3\beta_{1k} \\ & & & \cdot \\ & & & \cdot \\ & & & \alpha_{k-1}\beta_{1k} \\ 0 & \cdots & 0 \ \beta_{k,f+1} \ \cdots \ \beta_{k,k-1} & \beta_{kk} \end{bmatrix}.$$

Since v_k is assumed to be reachable from at least one of the vertices in $F(G(B_k))$, $\beta_{1k} \neq 0$. Picking β_{1k} as pivot, we have,

$$B_k' = \begin{bmatrix} & & & 0 \\ & & & 0 \\ & B_{k-1}^{\bullet} & & \cdot \\ & & & \cdot \\ & & & \cdot \\ & & & 0 \\ 0 & \cdots & 0 \;\; \bar{\beta}_{k,f+1} \;\; \cdots \;\; \bar{\beta}_{k,k-1} & 0 \end{bmatrix},$$

where $\bar{\beta}_{kj} = \beta_{kj} - \frac{\beta_{kk}}{\beta_{1k}}\beta_{1j}$, $(f+1) \leq j < k$, which satisfies the reachability condition since $v_k \in F(G(B_k'))$. Therefore, either β_{1q} or β_{1k} of B_k is a feasible pivot and hence, the proposition is also true for $l = k$. □

Proof of Theorem 3.2 (continued) :

(1) ⇒ (2):

Suppose B satisfies the reachability condition and let $F(G(B))$ be $\{v_1, \cdots, v_f\}$. The following procedure finds the permutation matrix Q, a unit lower triangular matrix L, and a strictly upper triangular matrix U such that $Q^{-1}BQ = L^{-1}U$:

$Q \leftarrow I_n$; $L \leftarrow I_n$; $l \leftarrow 1$; $B_1 \leftarrow B$;

Repeat

{

$u \leftarrow n - l + 1$; $f_l \leftarrow |F(G(B_l))|$;

Find a feasible pivot $\beta_{ij} \in B_l$, $1 \leq i \leq f_l < j \leq u$, (Lemma 3.1); (†)

$R_l \leftarrow (i,1)$ row permutation matrix;

$H \leftarrow R_l B_l R_l^{\mathrm{T}}$;

$$L_l \leftarrow \begin{bmatrix} 1 & & & \\ -\alpha_2 & \cdot & & 0 \\ -\alpha_3 & & \cdot & \\ \cdot & & & \cdot \\ \cdot & & 0 & \cdot \\ -\alpha_u & & & 1 \end{bmatrix}, \; \alpha_k = \frac{h_{kj}}{h_{1j}};$$

$H^{\bullet} \leftarrow LH$ with the first row and column removed;

$$L \leftarrow \begin{bmatrix} I_{n-u} & 0 \\ 0 & L_l R_l \end{bmatrix} L;$$

$$Q \leftarrow \begin{bmatrix} I_{n-u} & 0 \\ 0 & R_l \end{bmatrix} Q;$$

$Q_l^{\mathrm{T}} \leftarrow$ column permutation which moves the zero columns of $H^{\bullet}$ to the left;

$B_{l+1} \leftarrow Q_l H^{\bullet} Q_l^{\mathrm{T}}$;

$$L \leftarrow \begin{bmatrix} I_{n-u+1} & 0 \\ 0 & Q_l \end{bmatrix} L;$$

$$Q \leftarrow \begin{bmatrix} I_{n-s+1} & 0 \\ 0 & Q_l \end{bmatrix} Q \ ;$$

$l \leftarrow l+1;$

}

until B_l becomes a null matrix;

$U \leftarrow L\,BQ^{T};$

$L \leftarrow LQ^{T};$

$Q \leftarrow Q^{T};$

The elimination step in the above repeat loop has to be done with *exact arithmetic* as the connectivity of the digraph depends on the exact values of the matrix entries.

By Lemma 3.1, if B_l satisfies the reachability condition then a feasible pivot can be found at line (†) such that B_{l+1} also satisfies the reachability condition. Hence, if B_1 satisfies the reachability condition, each iteration of the repeat loop eliminates at least one diagonal element and the associated subdiagonal elements and a strictly upper triangular matrix results when the procedure terminates. Thus, (1) implies (2). □

Theorem 3.2 establishes the rather surprising property that the decomposability of a rank deficient matrix into the product of a unit lower triangular matrix and a strictly upper triangular matrix (up to symmetric permutation) depends only on the zero (or non-zero) structure of the matrix and not on the value of its non-zero entries. Utilizing this result, the test for decomposability of a broadcasting dependence mapping into elementary propagation can be accomplished simply by finding all the strongly connected components in the digraph $G(B)$ and verifying that each strongly connected component has at least one incoming edge from a vertex external to that component. The test has complexity of $O(\mathrm{MAX}(n, |E|))$ [9]. The procedure detailed in the proof of Theorem 3.2, which has complexity of $O(n^4)$, can be used to find a feasible order of propagation, if one exists.

Example 2 (Back-substitution solver) :

Given a non-singular $n \times n$ lower triangular band matrix A of bandwidth p, and an n-vector b, the solution x of the system of equations

$$Ax = b$$

can be found by back-substitution. The algorithm can be expressed in the following sequential program loop :

for $i = 1$ to n do

$$x_i = (b_i - \sum_{j<i} a_{ij}\, x_j)/a_{ii}$$

which can be translated into the following LDA:

$$x(i,j) = \begin{cases} x(i,j-1) - a_{ij}\, x(j,j) & i \neq j \\ x(i,i-1)/a_{ii} & i = j \end{cases}$$

The domain of computation is $\mathbf{C} = \{\, 1 \leq i \leq n \ , \max(1, i-p+1) \leq j \leq i \,\}$, with the boundary condition $x(i, j_0) = b_i$, where $j_0 = \max(1, i-p+1) - 1$, and the final results are

$x_i = x(i, i)$. The dependence of $x(i, j)$ on $x(j, j)$ is a broadcasting, $B = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$. Since the second row of B is equal to $\vec{e}_i^{\,T}$, we let $B \leftarrow B_{11} = [\,0\,]$, and hence $G(B)$ is a digraph with only one vertex (representing index i), which is free. Since B satisfies the reachability condition, the broadcasting can be decomposed into elementary propagation with one propagation variable, where

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, L = \begin{bmatrix} 1 \end{bmatrix}, U = \begin{bmatrix} 0 \end{bmatrix}, H = \begin{bmatrix} 1 \end{bmatrix}.$$

Introducing the propagation variable $\overline{x}$, the LDA now becomes

$$x(i, j) = \begin{cases} x(i, j-1) - a_{ij}\,\overline{x}(i, j) & i \neq j \\ x(i, i-1)/a_{ii} & i = j \end{cases}$$

$$\overline{x}(i, j) = \begin{cases} \overline{x}(i-1, j) & \text{if } i-1 \neq j \\ x(i-1, j) & \text{otherwise} \end{cases}$$

In the recurrence of $\overline{x}$, only two conditionals are needed because the coordinates of the points in

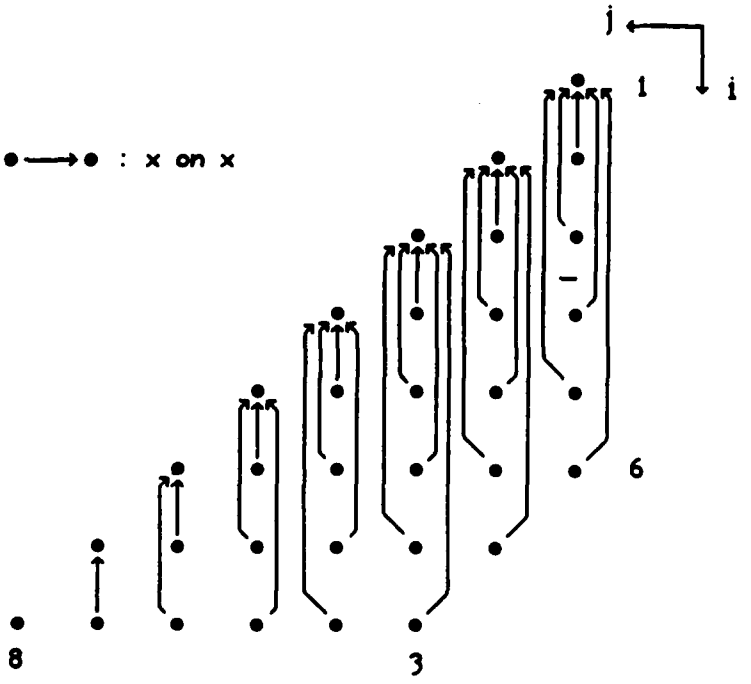


Figure 3a : Broadcasting dependences in the back-substitution algorithm.

the computation domain satisfy $i \geq j$. The broadcasting dependences and the corresponding propagation dependences after decomposition for the case $p = 6$ and $n = 8$ are shown in Figure 3a and 3b, respectively. □

Although it is powerful, elementary propagation does not handle all of the broadcasting dependence mappings commonly found in numerical algorithms. For example, the broadcasting in the 1-D recursive filtering algorithm shown at the beginning of this section is not decomposable into elementary propagation. In the next subsection, we generalize our approach to cover all broadcastings.

3.2. Composite Propagation

As noted in the previous subsection, constraining the propagation dependences to elementary dependence vectors is too restrictive. In this subsection, we extend the basic principle of propagation to allow propagation dependence vectors which are not necessarily elementary. This *composite propagation* technique is a generalization of the elementary propagation scheme discussed in the

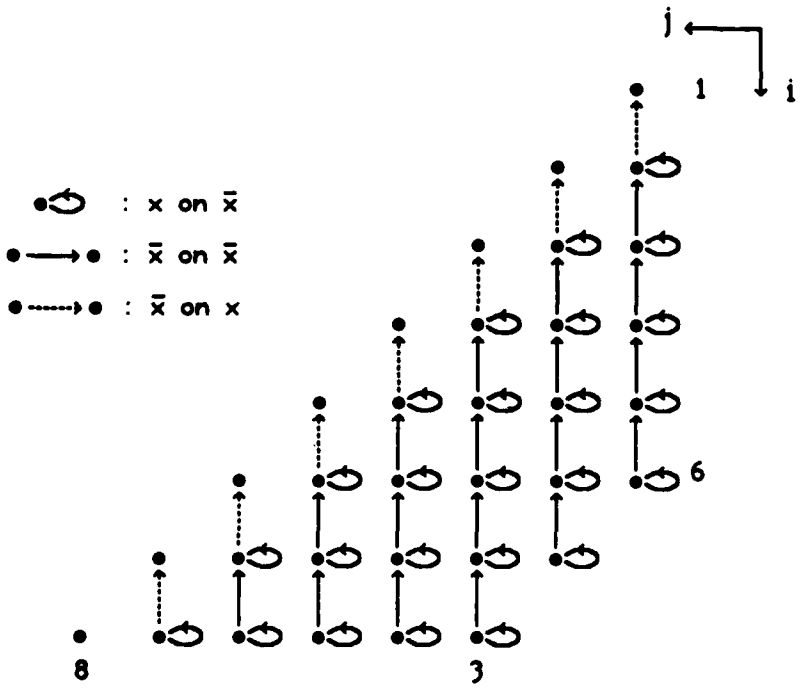


Figure 3b : Decomposition of broadcasting dependences shown in Figure 3a into elementary propagation dependences.

previous subsection. We prove that, with such generalization, any broadcasting dependence can be transformed into propagation.

A major complication in allowing non-elementary dependence vectors on a propagation path is that the conditionals for defining the propagation variables are no longer dependent on the index components which remain invariant in a section of the path. This can be seen in the following example :

Example 3 :

Consider the broadcasting dependence

$$a(i_1, i_2) \rightarrow b(i_1+i_2, i_1+i_2) \ , \ i.e., \ B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} .$$

It is not transformable into elementary propagation as B does not satisfy the reachability condition. By using the non-elementary propagation dependence vectors, $\vec{w}_1 = (1, -1)^T$ and $\vec{w}_2 = (1, -2)^T$, however, this broadcasting mapping can be transformed into propagation :

$$a(i_1, i_2) \rightarrow \overline{a}_1(i_1, i_2) ,$$

where,

$$\overline{a}_1(P) = \begin{cases} \overline{a}_1(P + \vec{w}_1) & \text{if } (i_1 + 2i_2 - 1) > 0 \\ \overline{a}_1(P - \vec{w}_1) & \text{if } (i_1 + 2i_2 + 1) < 0 \\ \overline{a}_2(P + \vec{w}_1) & \text{if } (i_1 + 2i_2 - 1) = 0 \\ \overline{a}_2(P - \vec{w}_1) & \text{if } (i_1 + 2i_2 + 1) = 0 \\ \overline{a}_2(P) & \text{if } (i_1 + 2i_2) = 0 \end{cases} , \quad \overline{a}_2(P) = \begin{cases} \overline{a}_2(P + \vec{w}_2) & \text{if } (i_1 - i_2 + 3) < 0 \\ \overline{a}_2(P - \vec{w}_2) & \text{if } (i_1 - i_2 - 3) > 0 \\ b(P + \vec{w}_2) & \text{if } (i_1 - i_2 + 3) = 0 \\ b(P - \vec{w}_2) & \text{if } (i_1 - i_2 - 3) = 0 \\ b(P) & \text{if } (i_1 - i_2) = 0 \end{cases}$$

Some of the broadcasting dependences and the corresponding propagation dependence paths are shown in Figure 4a and 4b. Interested readers can verify that the propagation scheme does indeed implement the broadcasting correctly. □

Unlike elementary propagation, in which the index points along a section share some common invariant components that can be used for determining the sense and extent of the section, composite propagation does not necessarily preserve any of the components of index points on a section. Consequently, the technique outlined in the previous subsection for defining the appropriate propagation variables does not apply directly. By restricting our attention to particular classes of propagation dependence vectors, however, we are able to derive a precise mathematical description of composite propagation. This is detailed in the following.

Mathematical formulation

Let P be an index point in the computation domain, $\mathbf{C}$. Originally, P is expressed as a linear combination of the n elementary vectors, $\{\vec{e}_i\}$, which make up the *canonical basis*,

$$P = p_1\vec{e}_1 + p_2\vec{e}_2 + \cdots + p_n\vec{e}_n .$$

Suppose we perform a *change of basis* [10] to $\mathbf{C}$ from the canonical basis to the basis defined by the n *linearly independent* integral vectors, $\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_n$, which form the columns of a matrix W.

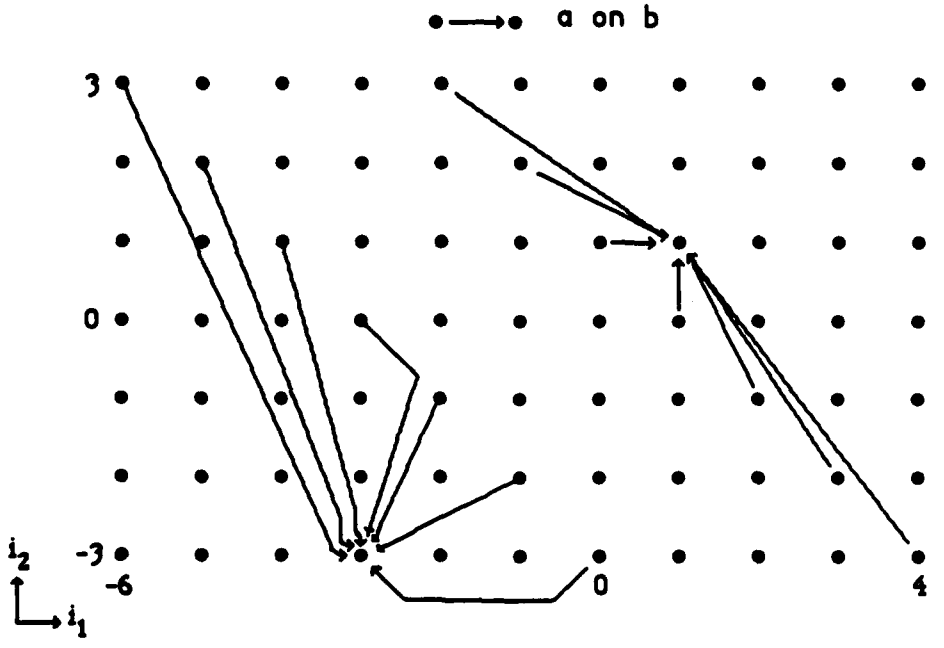


Figure 4a : Some of the broadcasting dependences of Example 3. The broadcast data are located along the line $i_1 = i_2$.

With respect to this new basis, the index point P becomes $\bar{P}$ where

$$P = \bar{p}_1\vec{w}_1 + \bar{p}_2\vec{w}_2 + \cdots + \bar{p}_n\vec{w}_n \Rightarrow \bar{P} = W^{-1}P ,$$

that is, a change of basis from $\{\vec{e}_i\}$ to the columns of W corresponds to a linear mapping, W^{-1}, which maps index points $P \in \mathbf{C}$ to index points $\bar{P}$ of the *transformed* computation domain, $\bar{\mathbf{C}}$. To ensure that $\bar{\mathbf{C}}$ contains only integral points, W must be *unimodular*, *i.e.*, $|\det(W)| = 1$.

Let $d = B$ be a broadcasting dependence mapping in $\mathbf{C}$. Under the change of basis to W, the broadcasting dependence $P \rightarrow BP$ becomes

$$W^{-1}P \rightarrow W^{-1}BP \Rightarrow \bar{P} \rightarrow W^{-1}BW(W^{-1}P) = W^{-1}BW\bar{P}$$

in $\bar{\mathbf{C}}$. Thus, the broadcasting dependence B undergoes a *similarity transformation* [10], $\bar{B} = W^{-1}BW$. Since $\bar{B}$ represents also a broadcasting dependence in $\bar{\mathbf{C}}$, if $\bar{B}$ is elementary transformable, the elementary propagation paths of $\bar{B}$ in $\bar{\mathbf{C}}$ can be mapped by W into propagation paths in $\mathbf{C}$ that implement the broadcasting dependence B. This is formally stated in the following theorem.

Theorem 3.3 :

A broadcasting dependence B is decomposable into *composite propagation* with propagation

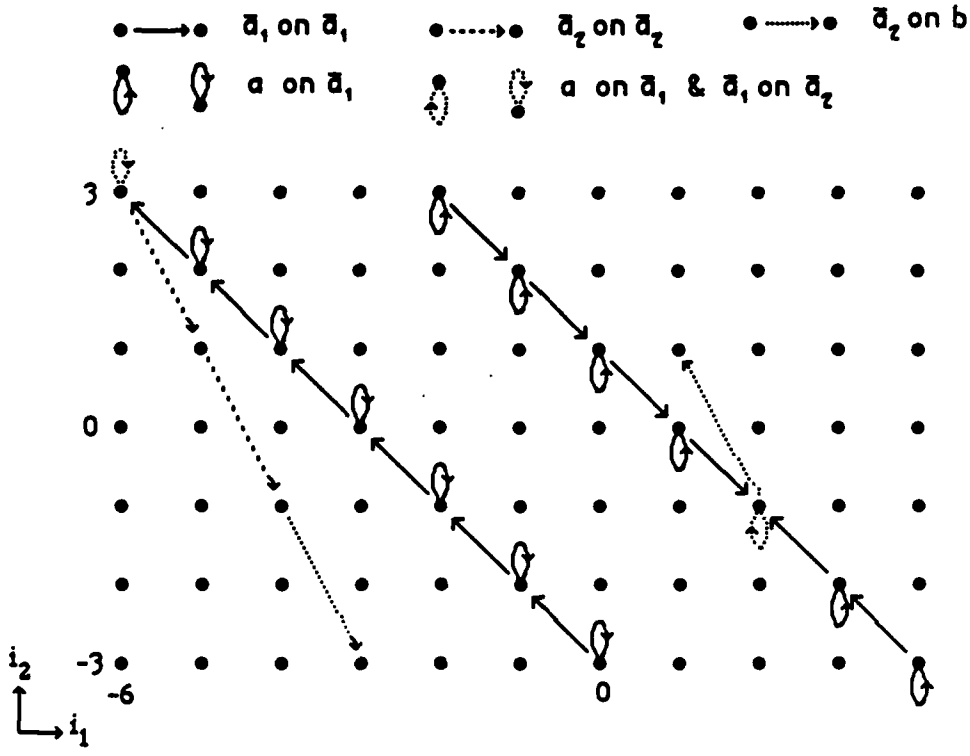


Figure 4b : Decomposing broadcasting dependences shown in Figure 4a into propagation dependences with dependence vectors $(1\ -1)^T$ and $(1\ -2)^T$.

dependence vectors $\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_m$, if there exists an $n \times n$ unimodular integral matrix $W = (\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_n)$, called a *feasible basis*, such that

$$W^{-1}BW = \bar{B} = \begin{bmatrix} L^{-1}U & H \\ 0 & I_{n-m} \end{bmatrix}$$

where L is an $m \times m$ unit lower triangular matrix, U is an $m \times m$ strictly upper triangular matrix, H is an $m \times (n-m)$ matrix, and I_{n-m} is the identity matrix of order $n-m$.

Proof :

By Theorem 3.1, the broadcasting dependence $\bar{B}$ can be decomposed into elementary propagation with propagation variables, $\bar{a}_i$, defined with respect to the transformed domain, $\bar{P} \in \bar{C}$,

$$\bar{a}_i(\bar{P}) = \begin{cases} \bar{a}_i(\bar{P} + \bar{e}_i) & \text{if } (\bar{P} + \bar{e}_i)\rangle_i < \bar{p}_{0i} \\ \bar{a}_i(\bar{P} - \bar{e}_i) & \text{if } (\bar{P} - \bar{e}_i)\rangle_i > \bar{p}_{0i} \\ \bar{a}_{i+1}(\bar{P} + \bar{e}_i) & \text{if } (\bar{P} + \bar{e}_i)\rangle_i = \bar{p}_{0i} \\ \bar{a}_{i+1}(\bar{P} - \bar{e}_i) & \text{if } (\bar{P} - \bar{e}_i)\rangle_i = \bar{p}_{0i} \\ \bar{a}_{i+1}(\bar{P}) & \text{if } (\bar{P})\rangle_i = \bar{p}_{0i} \end{cases} , \quad \forall \bar{P} \in \bar{\mathbf{C}}, i \leq m , \tag{4}$$

where $\bar{e}_i$ is the ith canonical vector (in the transformed domain) and $\bar{p}_{0i}$ is the ith component of the index point $\bar{P}_0$ at which the broadcast quantity is located. Using equation (3), $\bar{p}_{0i}$ can be expressed as the invariant component of the ith section of the elementary propagation path in $\bar{\mathbf{C}}$. Applying the linear transformation W to the domain $\bar{\mathbf{C}}$, and substituting $W^{-1}P$ and $\vec{w}_i$ for $\bar{P}$ and $\bar{e}_i$, respectively, in the conditionals, expression (4) becomes

$$\bar{a}_i(P) = \begin{cases} \bar{a}_i(P + \vec{w}_i) & \text{if } (W^{-1}P + \vec{w}_i)\rangle_i < p_{0i} \\ \bar{a}_i(P - \vec{w}_i) & \text{if } (W^{-1}P - \vec{w}_i)\rangle_i > p_{0i} \\ \bar{a}_{i+1}(P + \vec{w}_i) & \text{if } (W^{-1}P + \vec{w}_i)\rangle_i = p_{0i} \\ \bar{a}_{i+1}(P - \vec{w}_i) & \text{if } (W^{-1}P - \vec{w}_i)\rangle_i = p_{0i} \\ \bar{a}_{i+1}(P) & \text{if } (W^{-1}P)\rangle_i = p_{0i} \end{cases} , \quad \forall P \in \mathbf{C}, i \leq m , \tag{5}$$

where p_{0i} is obtained from $\bar{p}_{0i}$, which is a linear function of $\bar{p}_{0j}$, $j \neq i$, with the appropriate substitutions. Thus, the broadcasting dependence B can be transformed into composite propagation with propagation dependence vectors $\vec{w}_i$, $1 \leq i \leq m$. Obviously, the elementary propagation scheme discussed in the previous subsection is a particular case of composite propagation in which the feasible basis, W, is an order n permutation matrix, Q. □

Example 4 :

Apply Theorem 3.3 to Example 3:

Let $W = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix}$, then

$$\bar{B} = W^{-1}BW = \begin{bmatrix} 0 & -3 \\ 0 & 2 \end{bmatrix}$$

$$= L^{-1}U = \begin{bmatrix} 1 & 0 \\ -\frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 0 & -3 \\ 0 & 0 \end{bmatrix} .$$

Hence $\bar{B}$ is decomposable into elementary propagation with propagation variables

$$\bar{a}_1(\bar{P}) = \begin{cases} \bar{a}_1(\bar{P} + \bar{e}_1) & \text{if } (\bar{P} + \bar{e}_1)\rangle_1 < -3\bar{p}_2 \\ \bar{a}_1(\bar{P} - \bar{e}_1) & \text{if } (\bar{P} - \bar{e}_1)\rangle_1 > -3\bar{p}_2 \\ \bar{a}_2(\bar{P} + \bar{e}_1) & \text{if } (\bar{P} + \bar{e}_1)\rangle_1 = -3\bar{p}_2 \\ \bar{a}_2(\bar{P} - \bar{e}_1) & \text{if } (\bar{P} - \bar{e}_1)\rangle_1 = -3\bar{p}_2 \\ \bar{a}_2(\bar{P}) & \text{if } (\bar{P})\rangle_1 = -3\bar{p}_2 \end{cases} ,$$

$$\bar{a}_2(\bar{P}) = \begin{cases} \bar{a}_2(\bar{P} + \bar{e}_2) & \text{if } (\bar{P} + \bar{e}_2)\!\downarrow_2 < -\frac{2}{3}\bar{p}_1 \\ \bar{a}_2(\bar{P} - \bar{e}_2) & \text{if } (\bar{P} - \bar{e}_2)\!\downarrow_2 > -\frac{2}{3}\bar{p}_1 \\ b(\bar{P} + \bar{e}_2) & \text{if } (\bar{P} + \bar{e}_2)\!\downarrow_2 = -\frac{2}{3}\bar{p}_1 \\ b(\bar{P} - \bar{e}_2) & \text{if } (\bar{P} - \bar{e}_2)\!\downarrow_2 = -\frac{2}{3}\bar{p}_1 \\ b(\bar{P}) & \text{if } (\bar{P})\!\downarrow_2 = -\frac{2}{3}\bar{p}_1 \end{cases}.$$

Some of the elementary propagation paths in the transformed domain are displayed in Figure 4c. Applying the linear transformation W to the domain $\bar{\mathbf{C}}$ and performing the substitutions

$W^{-1}P = (\bar{p}_1, \bar{p}_2)^T = ((2i_1 + i_2),\ -(i_1 + i_2))^T$,
$(W^{-1}P \pm \bar{w}_1)\!\downarrow_1 = (\bar{p}_1 + w_{11}) = 2i_1 + i_2 \pm 1$,
$(W^{-1}P)\!\downarrow_1 = \bar{p}_1 = 2i_1 + i_2$,
$(W^{-1}P \pm \bar{w}_2)\!\downarrow_2 = (\bar{p}_2 + w_{22}) = -(i_1 + i_2 \pm 2)$, and
$(W^{-1}P)\!\downarrow_2 = \bar{p}_2 = -(i_1 + i_2)$.

in the conditionals in (5), we get the propagation variable definitions given in Example 3.

The choice of W is non-unique. For example, by selecting $W = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$, a different set of composite propagation paths is obtained (Figure 4d). □

Existence of a feasible basis

From Theorem 3.2, a broadcasting dependence B is decomposable into elementary propagation if and only if B satisfies the reachability condition. The following corollary of Theorem 3.3 is a trivial extension of Theorem 3.2 :

Corollary 3.1 :

A broadcasting dependence B is decomposable into composite propagation if and only if there exists a unimodular transformation W such that $\bar{B} = W^{-1}BW$ satisfies the reachability condition.

Hence, given B, theoretically one could determine all the feasible bases by checking the non-zero structure of $\bar{B}$ for all unimodular transformations W. The next theorem is the major contribution of this subsection.

Theorem 3.4 :

All broadcasting dependences can be decomposed into composite propagations.

We prove the theorem by showing that for any rank deficient matrix, B, there exists at least one unimodular transformation W such that $\bar{B} = W^{-1}BW$ satisfies the reachability condition. For the trivial case where $B = 0$, Theorem 3.4 is automatically satisfied as $\bar{B}$ is a null matrix for all unimodular transformations W. In the following discussion, we assume that $B \neq 0$. To prove the

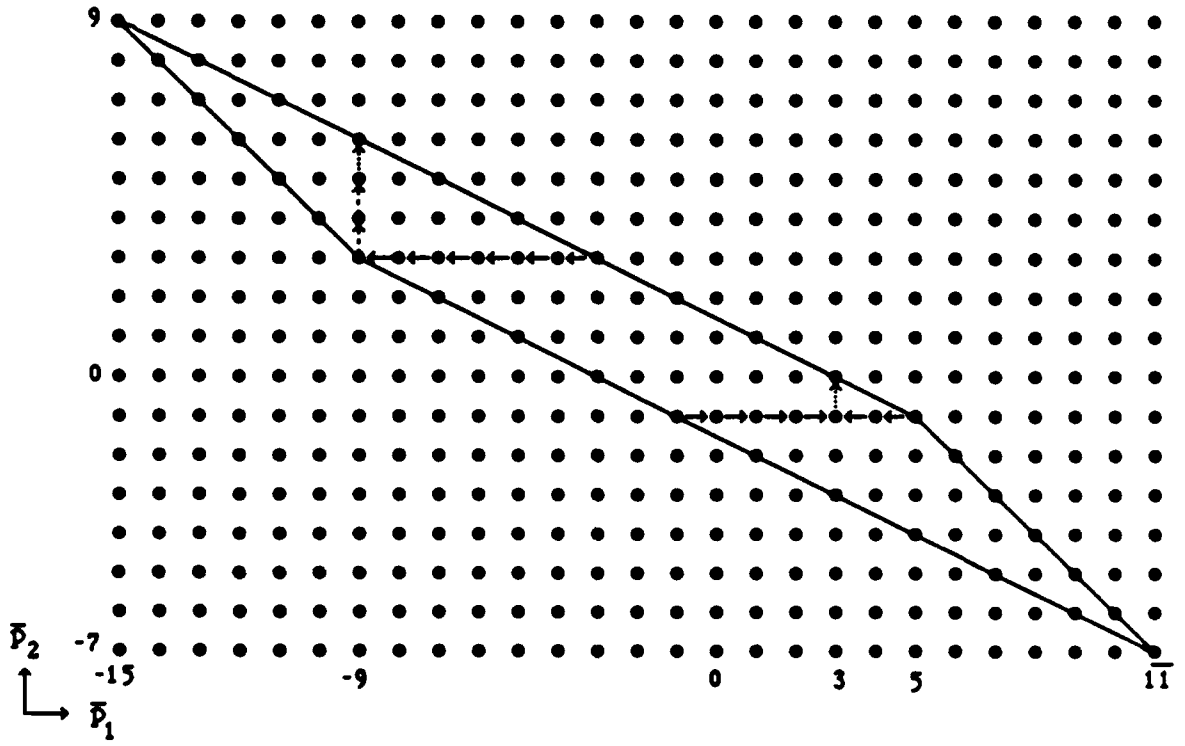


Figure 4c : Elementary propagation dependences obtained if a change of basis, W, is applied to the domain shown in Figure 4b. For clarity, the self-dependences are not displayed.

main result, we need the following lemma:

Lemma 3.2 :

Given a rank deficient matrix $B \neq 0$, there exist integral n-vectors $\vec{w}_1$ and $\vec{q}$, satisfying

$B\vec{w}_1 = \vec{0}$,

$\vec{q}^T\vec{w}_1 = 1$, and

$\vec{q}^T B \neq k\vec{q}^T$, for any scalar k.

Proof of Lemma 3.2 :

Since B is rank deficient there always exists an integral vector with coprime components, $\vec{w}_1$, such that $B\vec{w}_1 = \vec{0}$. For any such $\vec{w}_1$, there exists n linearly independent integral vectors, $\vec{q}_i$,

$1 \leq i \leq n$, which satisfy $\vec{q}_i^{\,T}\vec{w}_1 = 1$. If for each of these vectors there exists a scalar k_i such that

$$\vec{q}_i^{\,T}B = k_i\,\vec{q}_i^{\,T}$$

then

$$\vec{q}_i^{\,T}B\vec{w}_1 = k_i\,\vec{q}_i^{\,T}\vec{w}_1,$$

hence $k_i = 0$. Therefore, $\vec{q}_i^{\,T}B = 0$ for n linearly independent vectors which implies that $B = 0$, a contradiction. □

Let $\vec{q}$ and $\vec{w}_1$ be integral vectors that satisfy the conditions stated in Lemma 3.2. Let $\vec{s} = B^T\vec{q}$. Since $\vec{s}^{\,T} \neq k\vec{q}^{\,T}$ by assumption, the components of $\vec{s}$ and $\vec{q}$ can be reordered with the same permutation such that

$$s_1q_2 \neq s_2q_1\,. \qquad (c)$$

Without loss of generality, we assume that no reordering is required.

We prove Theorem 3.4 by constructing the integral vectors $\vec{w}_i$, $2 \leq i \leq n$, such that

(*i*) the first row of W^{-1} is $\vec{q}^{\,T}$,

(*ii*) the matrix $W = (\vec{w}_1, \vec{w}_2, \cdots, \vec{w}_n)$ is unimodular, and,

(*iii*) $\vec{q}^{\,T}B\vec{w}_i = \vec{s}^{\,T}\vec{w}_i \neq 0$, for all $i \geq 2$,

where $\vec{q}$ and $\vec{w}_1$ satisfy Lemma 3.2 and condition (*c*). For such a matrix W, $\bar{B} = W^{-1}BW$ has the form

$$\bar{B} = \begin{bmatrix} 0 & x & \cdots & x \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ 0 & & & \end{bmatrix}, \; x \neq 0\,,$$

which always satisfies the reachability condition, hence W is a feasible basis.

Proof of Theorem 3.4 :

Requirements (*i*) and (*ii*) are equivalent to

(*iv*) $\vec{q}^{\,T}\vec{w}_i = 0$, $i \geq 2$, and

(*v*) $\det(\vec{q}\;\;\vec{w}_2\;\cdots\;\vec{w}_n) = \pm 1$.

This follows directly from the fact that the first row of W^{-1} is $\vec{u}^{\,T}/\det(W)$, where $\vec{u}$ is the column vector whose ith component is the cofactor of the $(i, 1)$ element of W, and from the formulas

$$\det(W) = \vec{u}^{\,T}\vec{w}_1,$$

$$\det(\vec{q}\;\;\vec{w}_2\;\cdots\;\vec{w}_n) = \vec{u}^{\,T}\vec{q},$$

obtained by expanding the two determinants with respect to their first column.

We now construct, given $\vec{q}$ and $\vec{w}_1$ satisfying the conditions of Lemma 3.2 and condition (*c*), vectors $\vec{w}_2, \vec{w}_3, \cdots, \vec{w}_n$, which satisfy condition (*iii*) to (*v*). The associated matrix W is a feasible basis.

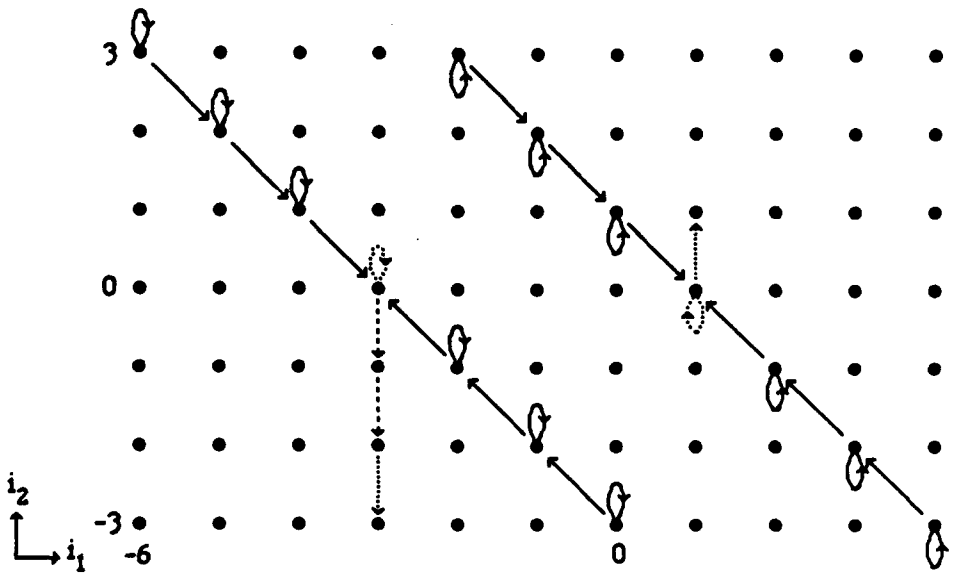


Figure 4d : Another feasible composite propagation scheme for the broadcasting dependences shown in Figure 4a. Note that the propagation paths are shorter than those shown in Figure 4b.

Let g_i, $1 \leq i \leq n$, be the gcd of the first i components of $\vec{q}$. The g_i's can be computed via the recurrence

$$g_1 = q_1,$$
$$g_i = \gcd(g_{i-1}, q_i),\ 2 \leq i \leq n,$$

hence, $\frac{q_i}{g_i}$ and $\frac{g_{i-1}}{g_i}$ are integers for $i \geq 2$.

The set of integral vectors, $\{\vec{w}_2, \vec{w}_3, \cdots, \vec{w}_n\}$, where

$$\vec{w}_i = (w_{i1}\ w_{i2}\ \cdots\ w_{i,i-1}\ \frac{g_{i-1}}{g_i}\ 0\ \cdots\ 0)^T\ ,\ 2 \leq i \leq n\ ,$$

$$\sum_{j=1}^{i-1} w_{ij}\, q_j = -\frac{g_{i-1}}{g_i} q_i\ ,\ \text{and,} \qquad (c1)$$

$$\sum_{j=1}^{i-1} w_{ij} s_j \neq -\frac{g_{i-1}}{g_i} s_i \ . \qquad (c2)$$

satisfies requirements (*iii*) to (*v*). This is shown next.

The Diophantine equation (*c1*) always has solutions, because the gcd of the coefficients on its left hand side, which is g_{i-1}, divides its right hand side [11]. Since, by assumption (*c*), the two hyperplanes defined by equation (*c1*) and equation

$$\sum_{j=1}^{i-1} w_{ij} s_j = -\frac{g_{i-1}}{g_i} s_i$$

are not parallel, there exists at least one solution, $\{w_{ij},\ j < i\}$, to equation (*c1*) that satisfies condition (*c2*). Therefore, the set $\{\vec{w}_2, \vec{w}_3, \cdots, \vec{w}_n\}$ is well defined.

It is straightforward to verify that condition (*c1*) implies (*iv*) and condition (*c2*) implies (*iii*). Thus the vectors $\vec{w}_i$, $i \geq 2$, satisfy requirements (*iii*) and (*iv*).

To see that $\vec{w}_2, \vec{w}_3, \cdots, \vec{w}_n$ satisfy requirement (*v*), consider the matrix

$$V = (\vec{q}, \vec{w}_2, \cdots, \vec{w}_n),$$

which has the following structure

$$V = \begin{bmatrix} q_1 & w_{21} & w_{31} & w_{41} & \cdot & \cdot & \cdot & w_{n1} \\ q_2 & w_{22} & w_{32} & \cdot & & & & w_{n2} \\ q_3 & 0 & w_{33} & \cdot & & & & w_{n3} \\ \cdot & \cdot & 0 & w_{44} & & & \cdot & \cdot \\ \cdot & & & 0 & \cdot & & & \cdot \\ \cdot & & & \cdot & 0 & \cdot & \cdot & \cdot \\ \cdot & & & & & 0 & w_{n-1,n-1} & \cdot \\ q_n & 0 & \cdot & \cdot & \cdot & 0 & 0 & w_{nn} \end{bmatrix}.$$

Partition the matrix V as

$$V = \begin{bmatrix} q_1 & \vec{r}^{\,T} \\ \vec{c} & \overline{V} \end{bmatrix}$$

where $\vec{r}^{\,T}$ is a row $(n-1)$-vector, $\vec{c}$ is a column $(n-1)$-vector and $\overline{V}$ is upper triangular.

Using Schur's determinantal formula [12], the determinant of V can be expressed as

$$\det(V) = (q_1 - \vec{r}^{\,T}\overline{V}^{-1}\vec{c})\det(\overline{V}) \ . \qquad (6)$$

Since, by construction, $\vec{q}^{\,T}\vec{w}_i = 0,\ 2 \leq i \leq n$,

$$q_1\vec{r}^{\,T} + \vec{c}^{\,T}\overline{V} = 0 \ ,$$

which yields the relation $\vec{r}^{\,T}\overline{V}^{-1} = -\dfrac{\vec{c}^{\,T}}{q_1}$. Substituting this relation into equation (6) and noting that

$$\det(\bar{V}) = = \prod_{i=2}^{n} w_{ii} = q_1,$$

the determinant of V is seen to satisfy

$$\det(V) = q_1^2 + \vec{c}^T\vec{c} = \vec{q}^T\vec{q} ,$$

which is equivalent to condition (v).

Therefore, the vectors $\vec{w}_i$, $i \geq 2$, satisfy the requirements (iii) to (v) and hence satisfy (i) to (iii). □

Given a broadcasting dependence, the basis W constructed in the proof of Theorem 3.4 can be used to transform the broadcasting dependences into composite propagation dependences. This is, however, not the only feasible choice of basis in general. It is unclear whether there exist efficient methods for determining all such feasible choices. In the concluding section, we will discuss current and future research topics in the area.

4. INPUT PIPELINING

Input values of an algorithm, which are used by more than one computation, should be pipelined to reduce the amount of communication between the host and the processor array on which the algorithm is implemented. To avoid performance degradation due to I/O bottleneck, it is essential to reduce the I/O bandwidth requirement of the algorithm so that the host is capable of delivering the input data at a rate which matches the computation rate of the array.

Example 5 :

Consider the one-dimensional convolution of signal x and w to form the filtered signal y as given by the following LDA consisting of a single recurrence,

$$y(i, j) = y(i, j-1) + w(j) \cdot x(i-j) ,$$

with computation domain $\{ 1 \leq i \leq n , 1 \leq j \leq i \}$. Thus, $B_{yy} = I_2$, $B_{yw} = [0\ 1]$ and $B_{yx} = [1\ -1]$.

The input streams w and x, which are needed for the computations of multiple instances of y, should be pipelined to reduce the I/O overhead. This can be realized by transforming the LDA into its equivalent *input-pipelined* form with the introduction of the *pipelined variables*, $\bar{w}$ and $\bar{x}$,

$$y(i, j) = y(i, j-1) + \bar{w}(i, j) \cdot \bar{x}(i, j)$$

$$\bar{w}(i, j) = \begin{cases} \bar{w}(i-1, j) & i > 1 \\ w(j) & \text{otherwise} \end{cases}$$

$$\bar{x}(i,j) = \begin{cases} \bar{x}(i-1, j-1) & i,j > 1 \\ x(i-j) & \text{otherwise} \end{cases}.$$

With this transformation, only one copy of each of the streams x and w is needed from the host. The dependence paths for pipelining the input variables are shown in Figure 5. □

In this section, we show that input pipelining can be treated as a special case of the broadcasting presented in the previous section and, therefore, that it can be handled with the same technique.

A variable a_l is an input stream if it is not computed within the LDA, *i.e.*, if $l > r$, the number of recurrences in the LDA. The following defines the pipelinability of an input stream :

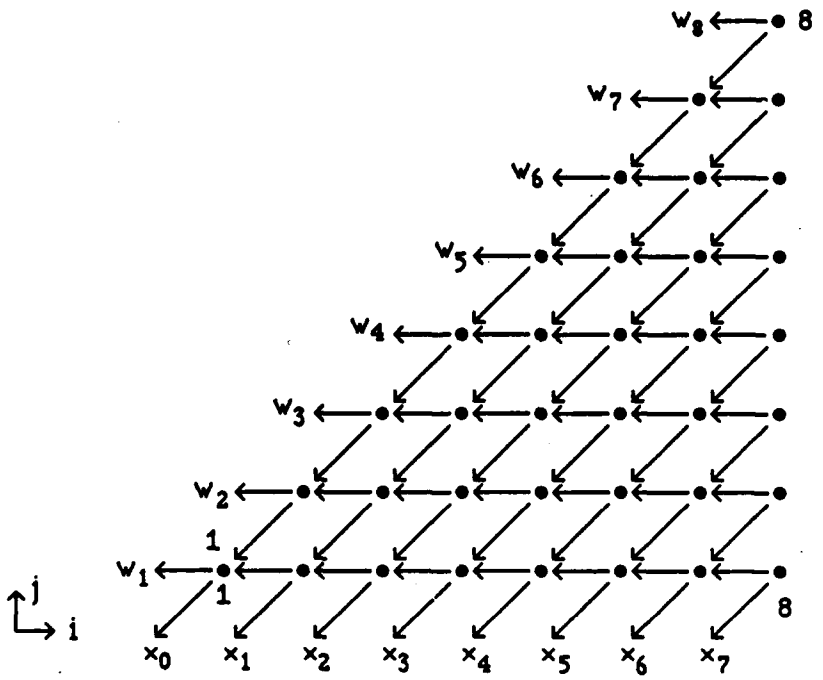


Figure 5 : Pipelining of input streams w and x in Example 5 ($n = 8$). The self-dependences are not displayed.

Definition:

An input stream a_l of an LDA is *pipelinable* if there exist one or more dependence mappings, d_{il}, $1 \leq i \leq r$, in the r recurrences of the LDA such that $\text{rank}(B_{il}) < n$.

Obviously, if all the B_{il}'s in the LDA have rank equal to n, the computation of a_i at each index point P requires a distinct value of a_l, $a_l(d_{il}(P))$. In this case, each of the input values of a_l is only used once and therefore the stream is not pipelinable.

Now, suppose there exists a B_{il} of size $m \times n$, $m \leq n$, with rank less than n. This resembles broadcasting in which a particular (computed) value is needed by several other computations. The only difference is that, since a_l is an input variable, we have the freedom to reassign the location of the broadcast value provided that the assignment does not result in conflicts, that is, we do not assign more than one a_{il} to an index point.

Let R_l be an $n \times m$ integral matrix, the *relocation transformation* of input variable a_l. R_l has to be chosen such that

$$R_l B_{il}(P_1 - P_2) = \vec{0} \text{ only if } B_{il}(P_1 - P_2) = \vec{0},$$

that is, R_l should have full column rank. Applying R_l to d_{il}, we have

$$\bar{d}_{il}(P) = \bar{B}_{il} P - \bar{\Delta}_{il} = R_l B_{il} P - R_l \Delta_{il}.$$

Since $\bar{B}_{il}$ is square and rank deficient, it can be treated as a broadcasting dependence mapping and the technique discussed in the previous section applies.

In Example 5, the chosen relocation transformations for the input variables w and x are

$$R_w = [0\ 1]^T \Rightarrow \bar{B}_{yw} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$R_x = [1\ 0]^T \Rightarrow \bar{B}_{yx} = \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix}.$$

By decomposing these two broadcastings respectively into elementary propagation and composite propagation (with $W = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$), we obtain the result given in Example 5.

Example 6 :

The matrix-matrix multiplication algorithm is an LDA comprising one recurrence equation

$$c(i,j,k) = c(i,j,k-1) + a(i,k) \cdot b(k,j), \quad 1 \leq i,j,k \leq N,$$

in which the input streams a and b are pipelinable. Applying the procedure described above,

$$R_a = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \Rightarrow \bar{B}_{ca} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$R_b = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow \bar{B}_{cb} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and transforming the broadcasting dependences into propagation, the input-pipelined version of the LDA is

$$c(i,j,k) = c(i,j,k-1) + \bar{a}(i,j,k) \cdot \bar{b}(i,j,k)$$

$$\bar{a}(i,j,k) = \begin{cases} \bar{a}(i,j-1,k) & \text{if } (j-1) > 0 \\ a(i,k) & \text{otherwise} \end{cases}$$

$$\bar{b}(i,j,k) = \begin{cases} \bar{b}(i-1,j,k) & \text{if } (i-1) > 0 \\ b(k,j) & \text{otherwise} \end{cases} \quad \square$$

5. CONCLUDING REMARKS

In this paper, we systematize the decomposition of broadcasting dependences into propagation dependences. Two kinds of propagation schemes, elementary and composite propagation, are introduced. It is shown that the derivation of the appropriate decomposition can be formulated as a linear algebra problem. Moreover, all broadcasting dependences are decomposable into propagation dependences.

In the discussion, we provided the theoretical framework for performing the decomposition but avoided the actual implementation issues because the optimal choice of propagation scheme depends on many factors; among others,

(1) the propagation dependence vectors must be selected in a way *compatible* with the other data dependences in the algorithm for the transformed algorithms to be executed efficiently,

(2) the number of propagation variables should be minimized to simplify the data flow and the control complexity of the processor array, and,

(3) the length of the propagation paths should be minimized as the computation time of the algorithm depends on the length of the longest dependence path in the computation domain.

Simple heuristics can be found for the optimal choice of propagation scheme. These results will be reported in a forthcoming paper.

Open problems and related research currently in progress include :

(1) Can efficient methods for determining all the feasible propagation schemes be found? This is equivalent to finding all the feasible bases W, such that the matrix $W^{-1}BW$ satisfies the reachability condition, a condition which depends on the non-zero structure of the matrix.

(2) Is it possible to classify and characterize all such feasible bases?

(3) Does there always exist a propagation scheme such that the transformed algorithm has the same order of run time as the original algorithm? In [4], it is shown that any *systolic algorithm* with computation domain of dimension s has $O(N)$ run time when

implemented on an $(s-1)$-dimensional systolic array, where N is the size parameter of the algorithm. We would like to know whether such a claim can be extended to LDA type algorithms

(4) How is the control and communication complexity of the array related to the propagation scheme used? We would like to justify the cost effectiveness of implementing algorithms with broadcasting dependences on regularly connected VLSI arrays.

The results reported in this paper will serve as stepping stones for further investigation of these topics.

Acknowledgement The authors would like to thank Professor Stan Eisenstat for many helpful discussions.

REFERENCES

[1] G.H. Golub, and C.F. Van Loan, *Matrix Computations*, John Hopkins University Press, Baltimore, MD, 1983

[2] C.E. Leiserson, and J.B. Saxe, "Optimizing Synchronous Systems," *Proceedings of the Twenty-Second Annual Symposium on Foundations of Computer Science*, IEEE, pp. 23-36, October 1981.

[3] R.P. Brent, and F.T. Luk, "The Solution of Singular-Value and Symmetric Eigenvalue Problems on Multiprocessor Arrays," *SIAM J. Sci. Stat. Comput.*, Vol. 6, No. 1, pp. 69-84, January 1985.

[4] S.K. Rao, "Regular Iterative Algorithms and their Implementation on Processor Arrays," *Ph.D. Dissertation*, Information System Laboratory, Stanford University, October 1985.

[5] S.Y. Kung, S.C. Lo, and P.S. Lewis, "Optimal Systolic Design for the Transitive Closure and the Shortest Path Problems," *IEEE Trans. Comput.*, Vol. C-36, No.5, pp. 603-614, May 1987.

[6] P. Gachet, B. Joinnault, and P. Quinton, "Synthesizing Systolic Arrays Using Diastol," in *Systolic Arrays*, pp. 25-36, W. Moore, *et al.*, eds., Adam Hilger, 1987.

[7] R.M. Karp, R.E. Miller, and S. Winograd, "The Organization of Computations for Uniform Recurrence Equations," *JACM*, Vol. 14, No. 3, pp. 563-590, 1967.

[8] J.-M. Delosme, and I.C.F. Ipsen, "Efficient Systolic Arrays for the Solution of Toeplitz Systems: An Illustration of a Methodology for the Construction of Systolic Architectures in VLSI," *Technical Report No. 370*, Department of Computer Science, Yale University, June 1985.

[9] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[10] B. Noble, and J. Daniel, *Applied Linear Algebra*, Prentice-Hall, Englewood Cliffs, N.J., 1977.

[11] L.J. Mordell, *Diophantine Equations*, Academic Press, New York, 1969.

[12] R.W. Cottle, "Manifestations of the Schur Complement," *J. Linear Algebra and its Applications*, Vol. 8, pp. 189-211, 1974.

END

7-87

DTIC