



Statement-Level Communication-Free Partitioning Techniques for Parallelizing Compilers

KUEI-PING SHIH AND JANG-PING SHEU* steven@axpl.csie.ncu.edu.tw, sheujp@csie.ncu.edu.tw
*Department of Computer Science and Information Engineering, National Central University,
Chung-Li 32054, Taiwan*

CHUA-HUANG HUANG chh@cc.ndhu.edu.tw
*Department of Computer Science and Information Engineering, National Dong-Hua University,
Hualien, Taiwan*

Received August 1998; final version accepted January 1999

Abstract. This paper addresses the problem of communication-free partition of iteration spaces and data spaces along hyperplanes. To finding more possible communication-free hyperplane partitions, we treat statements within a loop body as separate schedulable units. Instead of using the information about data dependence distance or direction vectors, our technique explicitly formulates array references as transformations from statement-iteration spaces to data spaces. Based on these transformations, the necessary and sufficient conditions for communication-free partition along hyperplanes to be feasible have been proposed. This approach can be applied to all programs with an imperfectly nested loop or sequences of imperfectly nested loops, whose array references are affine functions of outer loop indices or loop invariant variables. The proposed approach is more practical than existing methods in finding the data and computation distribution patterns that can cause the processor to execute fully-parallel on multicomputers without any interprocessor communication.

Keywords: Communication-free, data communication, distributed-memory multicomputers, hyperplane partition, parallelizing compilers

1. Introduction

Local memory access is much faster than memory access involving interprocessor communication on distributed-memory multicomputers. If data and computation are not properly distributed across processors, heavy interprocessor communication will result. The problem of data distribution is of critical importance to the efficiency of parallel programs in distributed memory multicomputers. Mace [14] proved that finding optimal data storage patterns for parallel processing is NP-complete, even when limited to one- and two-dimensional arrays. In addition, Li and Chen [11, 12] showed that the problem of finding the optimal data alignment is also NP-complete.

* Corresponding author: Jang-Ping Sheu, Department of Computer Science and Information Engineering, National Central University, Chung-Li 32054, Taiwan.

Thus, in previous work, a number of researchers have developed parallelizing compilers that require programmers to specify data storage patterns. Based on programmer-specified data partition, parallelizing compilers can automatically generate parallel programs with appropriate message passing constructs for multicomputers. Projects using this approach include the Fortran D compiler project [4, 5, 18], the SUPERB project [21], the Kali project [9, 10], and the DINO project [17]. For the same purpose, the Crystal project [11, 12] and the Id Nouveau compiler [16] deal with functional languages and generate parallel programs with message passing constructs. The parallel programs generated by most of these systems are in the SPMD (Single-Program Multiple Data) [8] model.

Recently, automatic data partitioning technique has been an attractive research topic in the field of parallelizing compilers. Many researchers have developed systems which help programmers deal with the problem of data distribution by automatically determining the data distribution at compile time. The PARADIGM project [3] and the SUIF project [1, 19] both have the same goal. These systems can automatically determine appropriate data distribution patterns in order to minimize the communication overhead and generate the SPMD code with appropriate message passing constructs for distributed memory multicomputers.

Since excessive interprocessor communication will offset the benefit of parallelization even if the program in question has a high degree of parallelism, parallelizing compilers must pay more attention to the distribution of computation and data across processors in order to reduce or even eliminate the communication overhead. Communication-free partition, therefore, has become an interesting and worthwhile area of research for distributed-memory multicomputers. In recent years, much research has focused on partitioning iteration spaces and/or data spaces to reduce interprocessor communication to achieve high-performance computing.

Ramanujam and Sadayappan [15] considered the problem of communication-free partition of data spaces along hyperplanes for distributed memory multicomputers. They presented a matrix-based formulation of the problem of determining the existence of communication-free partitions of data arrays. Their approach only proposes array decompositions and does not take the iteration space partitions into consideration. In addition, they concentrate on fully parallel nested loops and focus on two-dimensional data arrays.

Huang and Sadayappan [7] generalized the approach proposed in [15]. They dealt with the issue of communication-free hyperplane partition by explicitly modeling the iteration and data spaces, and provided the feasibility conditions for communication-free hyperplane partition. However, they did not deal with imperfectly nested loops. Moreover, their approach is restricted to loop-level partition; i.e., all statements within a loop body must be scheduled together as an indivisible unit.

Chen and Sheu [2] partitioned iteration spaces first according to the data dependence vectors obtained by analyzing all the reference patterns in a nested loop, and then grouped all data elements accessed by the same iteration partition. Two communication-free partitioning strategies, nonduplicate data and duplicate data strategies, were proposed in their paper. Nevertheless, they require that the loop contain only uniformly generated references, and that the problem domain be

restricted to a single perfectly nested loop. They also treated all statements within a loop body as an indivisible unit.

Lim and Lam [13] used affine processor mappings for statements to assign statement-iterations to processors and to maximize the degree of parallelism available in programs. Their approach does not treat the loop body as an indivisible unit and can assign different statement-iterations to different processors. However, they considered only statement-iteration space partition and did not address the issue of data space partition. Furthermore, their uniform affine processor mappings can cause a large number of processors to be idle if the affine mappings are nonunimodular transformations.

In this paper, communication-free partition of statement-iteration spaces and data spaces along hyperplanes is considered. We explicitly formulate array references as transformations from statement-iteration spaces to data spaces. Based on these transformations, we then present the necessary and sufficient feasibility conditions for communication-free hyperplane partitions. Currently, most of the existing partitioning schemes take an iteration instance as a basic schedulable unit that can be allocated to a processor. However, when the loop body contains multiple statements, it is difficult to make the loop execute in a communication free manner by allocating iteration instances among processors. The probability of communication-free execution using these methods is very low. For finding more possible communication-free hyperplane partitions, we treat statements within a loop body as separate schedulable units. Our method does not consider either the iteration space or the data space, but both of them. As in [13], our method can be extended to handle more general loop models and can be applied to programs with imperfectly nested loops and affine array references.

The rest of the paper is organized as follows. In Section 2, we introduce notations and terminology used throughout the paper. Section 3 describes the characteristics of statement-level communication-free hyperplane partition. The technique of statement-level communication-free hyperplane partition for a perfectly nested loop is presented in Section 4. The necessary and sufficient conditions for the feasibility of communication-free hyperplane partition are also given. Extension to general case for sequences of imperfectly nested loops is described in Section 5. Finally, conclusions are given in Section 6.

2. Preliminaries

This section explains the statement-iteration space and the data space. It also defines the statement-iteration hyperplane and the data hyperplane.

2.1. Statement-iteration space and data space

Let \mathbf{Q} , \mathbf{Z} , and \mathbf{Z}^+ denote the set of rational numbers, the set of integers and the set of positive integer numbers, respectively. The symbol \mathbf{Z}^d represents the set of d -tuple of integers. Traditionally, the iteration space is composed of discrete points

where each point represents the execution of *all* statements in one iteration of a loop [20]. Instead of viewing each iteration as indivisible, an iteration can be divided into statements that are enclosed in the iteration; i.e., each statement is a schedulable unit and has its own iteration space. We use another term, *statement-iteration space*, to denote the iteration space of a statement in a nested loop.

The following example illustrates the notion of iteration spaces and statement-iteration spaces:

Example 1. Consider the following nested loop L_1 :

```

do  $i_1 = 1, N$ 
  do  $i_2 = 1, N$ 
 $s_1$ :  $A[i_1, -i_1 - i_2 - 1]$ 
       $= A[i_1 - i_2 - 1, -i_1 + i_2 + 1] + B[i_1 + i_2, i_1 + 2i_2 - 1]$ 
 $s_2$ :  $B[i_1 - i_2 + 1, i_1 - 2i_2 + 1]$ 
       $= A[i_2 - 1, i_1 - i_2] * B[i_1 + i_2 - 1, i_1 + i_2 - 2]$ 
  enddo
enddo

```

Figure 1 illustrates the iteration space and statement-iteration spaces of loop L_1 for $N = 5$. In Figure 1(a), a circle means an iteration and includes two rectangles which are black and gray. The black rectangle indicates statement s_1 , and the gray one indicates statement s_2 . In Figure 1(b) and Figure 1(c), each statement is an individual unit, and the collection of statements forms two statement-iteration spaces. \square

The representation of statement-iteration spaces, data spaces and the relations among them is described as follows. Let \mathcal{S} denote the set of statements in the targeted problem domain, and let \mathcal{D} be the set of array variables that are referenced by \mathcal{S} . Consider statement $s \in \mathcal{S}$, which is enclosed in a d -nested loop. The statement-iteration space of s , denoted by $SIS(s)$, is a subspace of \mathbf{Z}^d and is defined as $SIS(s) = \{[I_1, I_2, \dots, I_d]^t \mid LB_i \leq I_i \leq UB_i, \text{ for } 1 \leq i \leq d\}$, where I_i is the loop index variable, and LB_i and UB_i are the lower and upper bounds of the loop index variable I_i , respectively. The superscript t is a transpose operator. The column vector $I_s = [I_1, I_2, \dots, I_d]^t$ is called a *statement-iteration* in statement-iteration space $SIS(s)$, $LB_i \leq I_i \leq UB_i$, for $i = 1, 2, \dots, d$. On the other hand, from the geometric point of view, an array variable also forms a space, and each array element is a point in the space. To exactly describe an array variable, we use *data space* to represent an n -dimensional array v , which is denoted by $DS(v)$, where $v \in \mathcal{D}$. An array element $v[D_1, D_2, \dots, D_n]$ has a corresponding data index in the data space $DS(v)$. We denote this data index by a column vector $D_v = [D_1, D_2, \dots, D_n]^t$.

The relations between statement-iteration spaces and data spaces can be built via *array reference functions*. An array reference function is a transformation from a

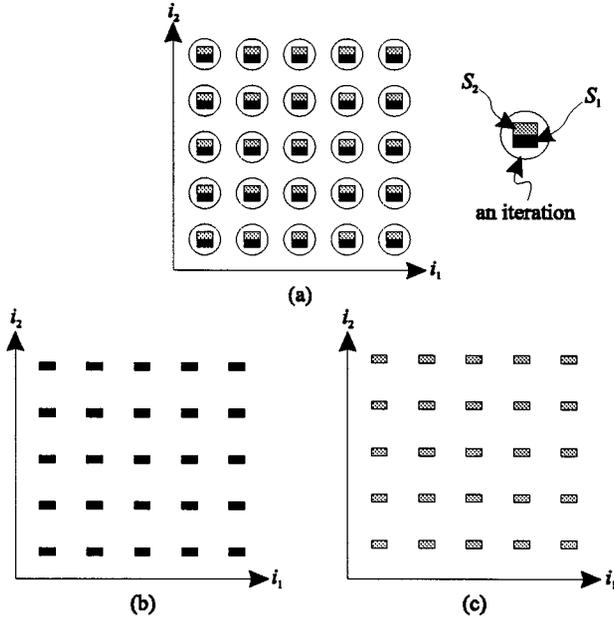


Figure 1. Loop (L_1)'s iteration space and its corresponding statement-iteration spaces, assuming $N = 5$. (a) $IS(L_1)$, iteration space of loop (L_1). (b) $SIS(s_1)$, statement-iteration space of statement s_1 . (c) $SIS(s_2)$, statement-iteration space of statement s_2 .

statement-iteration space into a data space. Like most of the existing methods, we require the array references to be affine functions of outer loop indices or loop invariant variables. Suppose statement s is enclosed in a d -nested loop and has an array reference pattern $v[a_{1,1}I_1 + a_{1,2}I_2 + \dots + a_{1,d}I_d + a_{1,0}, a_{2,1}I_1 + a_{2,2}I_2 + \dots + a_{2,d}I_d + a_{2,0}, \dots, a_{n,1}I_1 + a_{n,2}I_2 + \dots + a_{n,d}I_d + a_{n,0}]$, where $a_{i,j}$ are integer constants for $1 \leq i \leq n$ and $0 \leq j \leq d$; then the array reference function can be written as

$$Ref^{s,v}(I_s) = F^{s,v} \cdot I_s + f^{s,v},$$

where

$$F^{s,v} = \begin{bmatrix} a_{1,1} & \dots & a_{1,d} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \dots & a_{n,d} \end{bmatrix}, \quad \text{and} \quad f^{s,v} = \begin{bmatrix} a_{1,0} \\ \vdots \\ a_{n,0} \end{bmatrix}.$$

We term $F^{s,v}$ the *array reference coefficient matrix* and $f^{s,v}$ the *array reference constant vector*. If data index $D_v \in DS(v)$ is referenced in statement-iteration $I_s \in SIS(s)$, then $Ref^{s,v}(I_s) = D_v$. Take the array reference pattern $A[i - 2j + 3k - 4, 4i + 3j - 2k + 1]$ as an example. The array reference coefficient matrix and constant vector of $A[i - 2j + 3k + 4, 4i + 3j - 2k + 1]$ are $F^{s,A} = \begin{bmatrix} 1 & -2 & 3 \\ 4 & 3 & -2 \end{bmatrix}$ and $f^{s,A} = \begin{bmatrix} -4 \\ 1 \end{bmatrix}$, respectively.

We define statement-iteration hyperplanes and data hyperplanes in the next subsection.

2.2. Statement-iteration hyperplane and data hyperplane

A *statement-iteration hyperplane* on statement-iteration space $SIS(s)$, denoted by $\Psi(s)$, is a hyperspace [6] of $SIS(s)$ and is defined as $\Psi_h(s) = \{[I_1, I_2, \dots, I_d] | \delta_1 I_1 + \delta_2 I_2 + \dots + \delta_d I_d = c_h\}$, where δ_1, \dots , and $\delta_d \in \mathbf{Q}$ are the coefficients of the statement-iteration hyperplane and $c_h \in \mathbf{Q}$ is the constant term of the hyperplane. The formula can be abbreviated as $\Psi_h(s) = \{I_s | \Delta \cdot I_s = c_h\}$, where $\Delta = [\delta_1, \dots, \delta_d]$ is the statement-iteration hyperplane coefficient vector. Similarly, a *data hyperplane* on data space $DS(v)$, denoted by $\Phi(v)$, is a hyperspace of $DS(v)$ and is defined as $\Phi_g(v) = \{[D_1, D_2, \dots, D_n] | \theta_1 D_1 + \theta_2 D_2 + \dots + \theta_n D_n = c_g\}$, where θ_1, \dots , and $\theta_n \in \mathbf{Q}$ are the coefficients of the data hyperplane and $c_g \in \mathbf{Q}$ is the constant term of the hyperplane. In the same way, the formula also can be abbreviated as $\Phi_g(v) = \{D_v | \Theta \cdot D_v = c_g\}$, where $\Theta = [\theta_1, \dots, \theta_n]$ is the data hyperplane coefficient vector. Hyperplanes that include at least one integer point are considered in this paper.

Statement-iteration hyperplanes and data hyperplanes are used to characterize communication-free partition. We discuss some of these characteristics in the next section.

3. Characteristics of communication-free hyperplane partition

A program execution is communication-free if all operations on each of all processors access only data elements allocated to that processor. A trivial partitioning strategy allocates all statement-iterations and data elements to a single processor. The program execution of this trivial partition is communication-free. However, we are not interested in such single processor program execution because it does not exploit the potential of parallelization and it conflicts with the goal of parallel processing. Hence, in this paper, we consider only nontrivial partition, specifically, hyperplane partition.

The formal definition of a communication-free hyperplane partition is as follows. Let *partition group*, G ,

$$G = \bigcup_{s \in \mathcal{S}} \Psi_h(s) \cup \bigcup_{v \in \mathcal{D}} \Phi_g(v)$$

be the set of hyperplanes that should be assigned to one processor. The definition of a communication-free hyperplane partition is given in the following.

Definition 1 Hyperplane partitions of statement-iteration spaces and data spaces are said to be communication-free if and only if for any partition group $G = \bigcup_{s \in \mathcal{S}} \Psi_h(s) \cup \bigcup_{v \in \mathcal{D}} \Phi_g(v)$,

$$\forall I_s \in \Psi_h(s), \text{Ref}^{s,v}(I_s) \in \Phi_g(v), \quad \forall s \in \mathcal{S}, v \in \mathcal{D}. \quad \square$$

As mentioned above, statement-iterations which access the same array element should be allocated to the same statement-iteration hyperplane. Therefore, it is important to determine which statement-iterations that access the same array element. The following lemma states the necessary and sufficient condition under that two statement-iterations will access the same array element.

Lemma 1 *For some statement $s \in \mathcal{S}$ and its referenced array $v \in \mathcal{D}$, I_s and I'_s are two statement-iterations on $SIS(s)$ and $Ref^{s,v}$ is the array reference function from $SIS(s)$ into $DS(v)$ as defined above. Then*

$$Ref^{s,v}(I_s) = Ref^{s,v}(I'_s) \Leftrightarrow (I'_s - I_s) \in Ker(F^{s,v}),$$

where $Ker(S)$ denotes the null space of S [6].

Proof: (\Rightarrow): Suppose that $Ref^{s,v}(I_s) = Ref^{s,v}(I'_s)$. Since $Ref^{s,v}(I_s) = F^{s,v} \cdot I_s + f^{s,v}$, it follows that

$$\begin{aligned} Ref^{s,v}(I_s) &= Ref^{s,v}(I'_s) \\ \Rightarrow F^{s,v} \cdot I_s + f^{s,v} &= F^{s,v} \cdot I'_s + f^{s,v} \\ \Rightarrow F^{s,v} \cdot (I'_s - I_s) &= 0 \\ \Rightarrow (I'_s - I_s) &\in Ker(F^{s,v}). \end{aligned}$$

Thus $(I'_s - I_s) \in Ker(F^{s,v})$.

(\Leftarrow): Conversely, suppose that $(I'_s - I_s) \in Ker(F^{s,v})$ and $dim(Ker(F^{s,v})) = n$, $n \in \mathbf{Z}$. Let $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ be a basis of $Ker(F^{s,v})$; then vectors belonging to $Ker(F^{s,v})$ can be represented as a linear combination of vectors in $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$. Since $(I'_s - I_s) \in Ker(F^{s,v})$, it follows that $(I'_s - I_s) = c_1 \alpha_1 + c_2 \alpha_2 + \dots + c_n \alpha_n$, where $c_1, c_2, \dots, c_n \in \mathbf{Z}$. Hence, $I'_s = I_s + c_1 \alpha_1 + c_2 \alpha_2 + \dots + c_n \alpha_n$. Then,

$$\begin{aligned} Ref^{s,v}(I'_s) &= F^{s,v} \cdot I'_s + f^{s,v} \\ &= F^{s,v} \cdot (I_s + c_1 \alpha_1 + c_2 \alpha_2 + \dots + c_n \alpha_n) + f^{s,v} \\ &= F^{s,v} \cdot I_s + F^{s,v} \cdot (c_1 \alpha_1 + c_2 \alpha_2 + \dots + c_n \alpha_n) + f^{s,v} \\ &= F^{s,v} \cdot I_s + f^{s,v} \\ &= Ref^{s,v}(I_s). \end{aligned}$$

Thus $Ref^{s,v}(I_s) = Ref^{s,v}(I'_s)$. □

We illustrate Lemma 1 using the following example.

Example 2. Consider the array reference $A[i + j, i + j]$. The array reference coefficient matrix, $F^{s,A}$, is $[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}]$. The null space of $F^{s,A}$ is $Ker(F^{s,A}) = \{r[1, -1]^t | r$

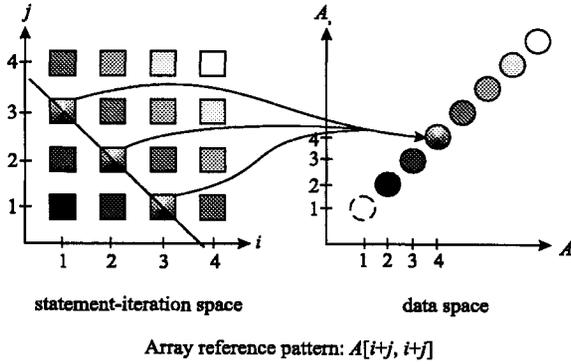


Figure 2. Those statement-iterations whose differences are in $\text{Ker}(F^{s,v})$ will access the same array element.

$\in \mathbf{Z}$). By Lemma 1, any two statement-iterations with a difference of $r[1, -1]^t$ will access the same array element, where $r \in \mathbf{Z}$. In Figure 2, statement-iterations with the same gray color denote that they access the same array element and their differences are in $\text{Ker}(F^{s,v})$. As Figure 2 shows, the statement-iterations $\{(1, 3), (2, 2), (3, 1)\}$ all access the same array element $A[4, 4]$. \square

We explain the significance of Lemma 1 and show how this lemma can help to find communication-free hyperplane partitions. Communication-free hyperplane partition requires that those statement-iterations which access the same array element be allocated to the same statement-iteration hyperplane. According to Lemma 1, two statement-iterations access the same array element if and only if the difference of these two statement-iterations belongs to the kernel of $F^{s,v}$. Hence $\text{Ker}(F^{s,v})$ should be a subspace of the statement-iteration hyperplane. Since there may exist many different array references, in partitioning a statement-iteration space, we must consider all the array references which appear in the statement. Thus, the space spanned from $\text{Ker}(F^{s,v})$ for all array references appearing in the same statement should be a subspace of the statement-iteration hyperplane. The dimension of a statement-iteration hyperplane is one less than the dimension of the statement-iteration space. If there exists a statement s such that the dimension of the spanning space of $\text{Ker}(F^{s,v})$ is equal to the dimension of $\text{SIS}(s)$, then the spanning space cannot be a subspace of a statement-iteration hyperplane. Therefore, there exists no nontrivial communication-free hyperplane partition. From the above observation, we obtain the following theorem.

Theorem 1 *If $\exists s \in \mathcal{S}$ such that*

$$\dim(\text{span}(\cup_{v \in \mathcal{D}} \text{Ker}(F^{s,v}))) = \dim(\text{SIS}(s)),$$

then there exists no non trivial communication-free hyperplane partition for \mathcal{S} and \mathcal{D} . \square

Example 3. Consider matrix multiplication:

```

do i = 1, N
  do j = 1, N
    do k = 1, N
s:      C[i, j] = C[i, j] + A[i, k]* B[k, j]
    enddo
  enddo
enddo

```

In the above program, there are three array variables, A , B , and C , with three distinct array references involved in statement s . The three array reference coefficient matrices, $F^{s,A}$, $F^{s,B}$, and $F^{s,C}$, are $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, and $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, respectively. Thus, $\text{Ker}(F^{s,A}) = \{r_1[0, 1, 0]^t | r_1 \in \mathbf{Z}\}$, $\text{Ker}(F^{s,B}) = \{r_2[1, 0, 0]^t | r_2 \in \mathbf{Z}\}$, and $\text{Ker}(F^{s,C}) = \{r_3[0, 0, 1]^t | r_3 \in \mathbf{Z}\}$. $\text{Ker}(F^{s,A})$, $\text{Ker}(F^{s,B})$, and $\text{Ker}(F^{s,C})$ span \mathbf{Z}^3 , which has the same dimensionality as the statement-iteration space. By Theorem 1, matrix multiplication has no nontrivial communication-free hyperplane partition. \square

Theorem 1 can be useful for determining nested loops that have no nontrivial communication-free hyperplane partition. Furthermore, when a nontrivial communication-free hyperplane partition exists, Theorem 1 can also be useful for finding the hyperplane coefficient vectors. We state this result in the following corollary.

Corollary 1 *For any communication-free statement-iteration hyperplane $\Psi_h(s) = \{I_s | \Delta \cdot I_s = c_h\}$, the following two conditions must hold:*

- (1) $\text{span}(\bigcup_{v \in \mathcal{S}} \text{Ker}(F^{s,v})) \subseteq \Psi_h(s)$,
- (2) $\Delta^t \in (\text{span}(\bigcup_{v \in \mathcal{S}} \text{Ker}(F^{s,v})))^\perp$,

where \mathcal{S}^\perp denotes the orthogonal complement space of \mathcal{S} .

Proof: By Lemma 1, two statement-iterations access the same data element using array reference $F^{s,v}$ if and only if the difference between these two statement-iterations belongs to the kernel of $F^{s,v}$. Therefore, the kernel of $F^{s,v}$ should be contained in the statement-iteration hyperplane, $\Psi_h(s)$. This should be true for all array references appearing in the same statement. Hence, $\text{span}(\bigcup_{v \in \mathcal{S}} \text{Ker}(F^{s,v})) \subseteq \Psi_h(s)$. The first condition is obtained.

Since $\Psi_h(s) = \{I_s | \Delta \cdot I_s = c_h\}$, Δ is the normal vector of $\Psi_h(s)$. That is, Δ^t is orthogonal to $\Psi_h(s)$. Condition (1) implies that Δ^t is orthogonal to the subspace $\text{span}(\bigcup_{v \in \mathcal{S}} \text{Ker}(F^{s,v}))$. Thus, Δ^t belongs to the orthogonal complement of $\text{span}(\bigcup_{v \in \mathcal{S}} \text{Ker}(F^{s,v}))$ [6]. \square

Corollary 1 gives the range of the communication-free statement-iteration hyperplane coefficient vectors. It can be used to find communication-free statement-

iteration hyperplane coefficient vectors. On the other hand, the range of the communication-free data hyperplane coefficient vectors is also given as described below.

As mentioned before, the relations between statement-iteration spaces and data spaces can be established via array references. Moreover, the statement-iteration hyperplane coefficient vectors and data hyperplane coefficient vectors are related. The following lemma expresses the relation between these two kinds of hyperplane coefficient vectors. A similar result was given in [7].

Lemma 2 *For any statement $s \in \mathcal{S}$ and its referenced array $v \in \mathcal{D}$, $Ref^{s,v}$ is the array reference function from $SIS(s)$ into $DS(v)$. $\Psi_h(s) = \{I_s | \Delta \cdot I_s = c_h\}$ and $\Phi_g(v) = \{D_v | \Theta \cdot D_v = c_g\}$ are communication-free hyperplane partitions if and only if $\Delta = \alpha \Theta \cdot F^{s,v}$, for some α , $\alpha \neq 0$.*

Proof: (\Rightarrow): Suppose that $\Psi_h(s) = \{I_s | \Delta \cdot I_s = c_h\}$ and $\Phi_g(v) = \{D_v | \Theta \cdot D_v = c_g\}$ are communication-free hyperplane partitions. Let I'_s and I''_s be two distinct statement-iterations which belong to the same statement-iteration hyperplane, $\Psi_h(s)$. If D'_v and D''_v are two data indices such that $Ref^{s,v}(I'_s) = D'_v$ and $Ref^{s,v}(I''_s) = D''_v$, from the above assumptions, D'_v and D''_v should belong to the same hyperplane, $\Phi_g(v)$.

Because I'_s and I''_s belong to the same statement-iteration hyperplane, $\Psi_h(s)$, it follows that $\Delta \cdot I'_s = c_h$ and $\Delta \cdot I''_s = c_h$. Therefore,

$$\begin{aligned} \Delta \cdot I'_s &= \Delta \cdot I''_s \\ \Rightarrow \Delta \cdot (I'_s - I''_s) &= 0. \end{aligned}$$

On the other hand, since D'_v and D''_v belong to the same data hyperplane, $\Phi_g(v)$, we can see that $\Theta \cdot D'_v = c_g$ and $\Theta \cdot D''_v = c_g$. Thus,

$$\begin{aligned} \Theta \cdot D'_v &= \Theta \cdot D''_v \\ \Rightarrow \Theta \cdot (D'_v - D''_v) &= 0 \\ \Rightarrow \Theta \cdot (Ref^{s,v}(I'_s) - Ref^{s,v}(I''_s)) &= 0 \\ \Rightarrow \Theta \cdot (F^{s,v} \cdot (I'_s - I''_s)) &= 0 \\ \Rightarrow (\Theta \cdot F^{s,v}) \cdot (I'_s - I''_s) &= 0. \end{aligned}$$

Since I'_s and I''_s are any two statement-iterations on the statement-iteration hyperplane $\Psi_h(s)$, $(I'_s - I''_s)$ is a vector on the statement-iteration hyperplane. Furthermore, both $\Delta \cdot (I'_s - I''_s) = 0$ and $(\Theta \cdot F^{s,v}) \cdot (I'_s - I''_s) = 0$; hence we can conclude that Δ and $\Theta \cdot F^{s,v}$ are linearly dependent. This implies $\Delta = \alpha \Theta \cdot F^{s,v}$, for some α , $\alpha \neq 0$ [6].

(\Leftarrow): Suppose $\Psi_h(s) = \{I_s | \Delta \cdot I_s = c_h\}$ and $\Phi_g(v) = \{D_v | \Theta \cdot D_v = c_g\}$ are hyperplane partitions for $SIS(s)$ and $DS(v)$, respectively, and that $\Delta = \alpha \Theta \cdot F^{s,v}$, for

some α , $\alpha \neq 0$. We claim that $\Psi_h(s)$ and $\Phi_g(v)$ are communication-free partition. According to Definition 1, what we have to do is to prove $\forall I_s \in \Psi_h(s), Ref^{s,v}(I_s) \in \Phi_g(v)$.

Let I_s be any statement-iteration on statement-iteration hyperplane $\Psi_h(s)$. Then $\Delta \cdot I_s = c_h$. From the assumption that $\Delta = \alpha \Theta \cdot F^{s,v}$, we have

$$\begin{aligned} & (\alpha \Theta \cdot F^{s,v}) \cdot I_s = c_h \\ \Rightarrow & \Theta \cdot (F^{s,v} \cdot I_s) = \frac{1}{\alpha} c_h \\ \Rightarrow & \Theta \cdot (F^{s,v} \cdot I_s + f^{s,v}) = \frac{1}{\alpha} c_h + \Theta \cdot f^{s,v} \\ \Rightarrow & \Theta \cdot Ref^{s,v}(I_s) = \frac{1}{\alpha} c_h + \Theta \cdot f^{s,v}. \end{aligned}$$

Let $c_g = (1/\alpha)c_h + \Theta \cdot f^{s,v}$; then $Ref^{s,v}(I_s) \in \Phi_g(v)$. We have shown that $\forall I_s \in \Psi_h(s), Ref^{s,v}(I_s) \in \Phi_g(v)$. It then follows that $\Psi_h(s)$ and $\Phi_g(v)$ are communication-free partition. \square

By Lemma 2, the statement-iteration hyperplane coefficient vector Δ can be decided if the data hyperplane coefficient vector Θ has been determined. If $F^{s,v}$ is invertible, the statement-iteration hyperplane coefficient vectors can be decided first; then the data hyperplane coefficient vectors can be derived by $\Theta = \alpha' \Delta (F^{s,v})^{-1}$, for some α' , $\alpha' \neq 0$. The range of the communication-free data hyperplane coefficient vectors can be derived from this lemma. Corollary 1 shows the range of the statement-iteration hyperplane coefficient vectors. The next corollary provides the range of the data hyperplane coefficient vectors.

Corollary 2 For any communication-free data hyperplane $\Phi_g(v) = \{D_v | \Theta \cdot D_v = c_g\}$, the following condition must hold:

$$\Theta^t \in \left(\bigcup_{s \in \mathcal{S}'} Ker((F^{s,v})^t) \right)^t,$$

where \mathcal{S}' denotes the complement of \mathcal{S} .

Proof: This paper considers nontrivial hyperplane partition, which requires that Δ be a nonzero vector. By Lemma 2, $\Delta = \alpha \Theta \cdot F^{s,v}$. Therefore, $\Theta \cdot F^{s,v}$ is not equal to 0. This implies that $\Theta^t \notin Ker((F^{s,v})^t)$. The condition should be true for all $s, s \in \mathcal{S}$. Hence, $\Theta^t \notin (\bigcup_{s \in \mathcal{S}} Ker((F^{s,v})^t))$. It follows that Θ^t belongs to the complement of $(\bigcup_{s \in \mathcal{S}} Ker((F^{s,v})^t))$, i.e., $\Theta^t \in (\bigcup_{s \in \mathcal{S}'} Ker((F^{s,v})^t))$. \square

Example 4. Consider the following loop:

```

do  $i = 1, N$ 
  do  $j = 1, N$ 
     $s_1: A[i, j] = B[i, i]$ 
     $s_2: B[i, j] = A[j, j]$ 
  enddo
enddo

```

The nested loop is communication-free if and only if the statement-iteration hyperplane coefficient vectors for s_1 and s_2 and data hyperplane coefficient vectors for v_1 and v_2 are $\Delta_1 = [r, 0]$, $\Delta_2 = [0, r]$, $\Theta_1 = [r, 0]$, and $\Theta_2 = [0, r]$, respectively, where $v_1 = A$, $v_2 = B$, and $r \in \mathbf{Q} - \{0\}$. We will show that Δ_1 and Δ_2 satisfy the Corollary 1 as follows:

$$\begin{aligned}
& \text{span}(\cup_{v \in \mathcal{D}} \text{Ker}(F^{s_1, v})) = \{c_1[0, 1]^t | c_1 \in \mathbf{Q} - \{0\}\}, \\
\Rightarrow & (\text{span}(\cup_{v \in \mathcal{D}} \text{Ker}(F^{s_1, v})))^\perp = \{[r_1, 0]^t | r_1 \in \mathbf{Q} - \{0\}\}, \\
\Rightarrow & \Delta_1^t = [r, 0]^t \in (\text{span}(\cup_{v \in \mathcal{D}} \text{Ker}(F^{s_1, v})))^\perp \\
& \cdot \text{span}(\cup_{v \in \mathcal{D}} \text{Ker}(F^{s_2, v})) = \{c_2[1, 0]^t | c_2 \in \mathbf{Q} - \{0\}\}, \\
\Rightarrow & (\text{span}(\cup_{v \in \mathcal{D}} \text{Ker}(F^{s_2, v})))^\perp = \{[0, r_2]^t | r_2 \in \mathbf{Q} - \{0\}\}, \\
\Rightarrow & \Delta_2^t = [0, t]^t \in (\text{span}(\cup_{v \in \mathcal{D}} \text{Ker}(F^{s_2, v})))^\perp.
\end{aligned}$$

The test of Corollary 2 for Θ_1 and Θ_2 is as follows:

$$\begin{aligned}
& \cup_{s \in \mathcal{S}} \text{Ker}((F^{s, v_1})^t) = \{c_3[1, -1]^t | c_3 \in \mathbf{Q} - \{0\}\}, \\
\Rightarrow & (\cup_{s \in \mathcal{S}} \text{Ker}((F^{s, v_1})^t))^\vee = \{[r_3, r_4]^t | r_3 \neq -r_4, r_3, r_4 \in \mathbf{Q} - \{0\}\}, \\
\Rightarrow & \Theta_1^t = [r, 0]^t \in (\cup_{s \in \mathcal{S}} \text{Ker}((F^{s, v_1})^t))^\vee \\
& \cdot \cup_{s \in \mathcal{S}} \text{Ker}((F^{s, v_2})^t) = \{c_4[1, -1]^t | c_4 \in \mathbf{Q} - \{0\}\}, \\
\Rightarrow & (\cup_{s \in \mathcal{S}} \text{Ker}((F^{s, v_2})^t))^\vee = \{[r_5, r_6]^t | r_5 \neq -r_6, r_5, r_6 \in \mathbf{Q} - \{0\}\}, \\
\Rightarrow & \Theta_2^t = [r, 0]^t \in (\cup_{s \in \mathcal{S}} \text{Ker}((F^{s, v_2})^t))^\vee. \quad \square
\end{aligned}$$

The next section describes the communication-free hyperplane partitioning technique. The necessary and sufficient conditions for communication-free hyperplane partition for a single perfectly nested loop will be presented.

4. Communication-free hyperplane partition for a perfectly nested loop

Each data array has a corresponding data space. However, a nested loop with multiple statements may have multiple statement-iteration spaces. In this section, we will consider additional conditions on multiple statement-iteration spaces for communication-free hyperplane partition. These conditions are also used in determining statement-iteration hyperplanes and data hyperplanes.

Suppose $\mathcal{S} = \{s_1, s_2, \dots, s_m\}$ and $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$, where $m, n \in \mathbf{Z}^+$. The number of occurrences of array variable v_j in statement s_i is $r_{i,j}$, where $r_{i,j} \in \mathbf{Z}^+ \cup \{0\}$, $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$. If s_i does not reference v_j , $r_{i,j}$ is set to 0. The previous representation of an array reference function can be modified slightly to describe the array reference of statement s_i to variable v_j in the k -th occurrence as $Ref_k^{s_i, v_j}(I_{s_i})$, where $1 \leq k \leq r_{i,j}$. The related representations will be changed accordingly, for example, $Ref_k^{s_i, v_j}(I_{s_i}) = F_k^{s_i, v_j} \cdot I_{s_i} + f_k^{s_i, v_j} = D_{v_j}$.

In this section, a partition group that contains a statement-iteration hyperplane for each statement-iteration space and a data hyperplane for each data space is considered. Suppose that the data hyperplane in data space $DS(v_j)$ is $\Phi_g(v_j) = \{D_{v_j} | \Theta_j \cdot D_{v_j} = c_{g_j}\}$, for all j , $1 \leq j \leq n$. Since $D_{v_j} = Ref_k^{s_i, v_j}(I_{s_i})$, for $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$ and $k = 1, 2, \dots, r_{i,j}$ and $\Theta_j \cdot D_{v_j} = c_{g_j}$, we have

$$\begin{aligned} \Theta_j \cdot D_{v_j} &= c_{g_j} \\ \Leftrightarrow \Theta_j \cdot (F_k^{s_i, v_j} \cdot I_{s_i} + f_k^{s_i, v_j}) &= c_{g_j} \\ \Leftrightarrow (\Theta_j \cdot F_k^{s_i, v_j}) \cdot I_{s_i} &= c_{g_j} - (\Theta_j \cdot f_k^{s_i, v_j}). \end{aligned}$$

Let

$$\Delta_i = \Theta_j \cdot F_k^{s_i, v_j}, \quad [1]$$

$$c_{h_i} = c_{g_j} - (\Theta_j \cdot f_k^{s_i, v_j}). \quad [2]$$

As a result, those statement-iterations that access the data on the data hyperplane $\Phi_g(v_j) = \{D_{v_j} | \Theta_j \cdot D_{v_j} = c_{g_j}\}$ will be located on the statement-iteration hyperplane $\Psi_h(I_{s_i}) = \{I_{s_i} | (\Theta_j \cdot F_k^{s_i, v_j}) \cdot I_{s_i} = c_{g_j} - (\Theta_j \cdot f_k^{s_i, v_j})\}$.

To simplify the presentation, we assume that all variables v_j appear in every statement s_i . To satisfy the requirement that each statement-iteration space contains a unique statement-iteration hyperplane, the following two conditions should be met:

- (i) $\forall i, \Theta_j \cdot F_k^{s_i, v_j} = \Theta_{j'} \cdot F_{k'}^{s_i, v_{j'}}$, ($j \neq j' \vee k \neq k'$), for $j, j' = 1, 2, \dots, n$; $k = 1, 2, \dots, r_{i,j}$ and $k' = 1, 2, \dots, r_{i,j'}$.
- (ii) $\forall i, c_{g_j} - (\Theta_j \cdot f_k^{s_i, v_j}) = c_{g_{j'}} - (\Theta_{j'} \cdot f_{k'}^{s_i, v_{j'}})$, ($j \neq j' \vee k \neq k'$), for $j, j' = 1, 2, \dots, n$; $k = 1, 2, \dots, r_{i,j}$ and $k' = 1, 2, \dots, r_{i,j'}$.

Condition (i) infers the following two equivalent equations:

$$\Theta_j \cdot F_k^{s_i, v_j} = \Theta_j \cdot F_1^{s_i, v_j}, \quad [3]$$

for $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$ and $k = 2, 3, \dots, r_{i,j}$.

$$\Theta_j \cdot F_1^{s_i, v_j} = \Theta_1 \cdot F_1^{s_i, v_1}, \quad [4]$$

for $i = 1, 2, \dots, m$; $j = 2, 3, \dots, n$. Condition (ii) leads to the following two equations, and vice versa:

$$\Theta_j \cdot f_k^{s_i, v_j} = \Theta_j \cdot f_1^{s_i, v_j}, \quad [5]$$

for $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$ and $k = 2, 3, \dots, r_{i,j}$.

$$c_{g_j} = c_{g_1} - \Theta_1 \cdot f_1^{s_i, v_1} + \Theta_j \cdot f_1^{s_i, v_j}, \quad [6]$$

for $i = 1, 2, \dots, m$; $j = 2, 3, \dots, n$.

Equation [6] can be used to evaluate the data hyperplane constant terms while some constant term is fixed, say c_{g_1} . Furthermore, we obtain the following results. For some j , c_{g_j} should be the same for all i , $1 \leq i \leq m$. Therefore,

$$c_{g_1} - \Theta_1 \cdot f_1^{s_i, v_1} + \Theta_j \cdot f_1^{s_i, v_j} = c_{g_1} - \Theta_1 \cdot f_1^{s_1, v_1} + \Theta_j \cdot f_1^{s_1, v_j}, \quad [7]$$

for $i = 2, 3, \dots, n$ and $j = 2, 3, \dots, n$. Equation [7] further infers the following equation:

$$\Theta_j \cdot (f_1^{s_i, v_j} - f_1^{s_1, v_j}) = \Theta_1 \cdot (f_1^{s_i, v_1} - f_1^{s_1, v_1}) \quad [8]$$

for $i = 2, 3, \dots, m$ and $j = 2, 3, \dots, n$.

After describing the conditions for satisfying the communication-free hyperplane partitioning constraints, we have the following theorem.

Theorem 2 Let $\mathcal{S} = \{s_1, s_2, \dots, s_m\}$ and $\mathcal{D} = \{v_1, v_2, \dots, v_n\}$ be the sets of statements and array variables, respectively. $\text{Ref}_k^{s_i, v_j}$ is the array reference function for statement s_i which accesses array variables v_j at the k -th occurrence in s_i , where $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$ and $k = 1, 2, \dots, r_{i,j}$. $\Psi_h(I_{s_i}) = \{I_{s_i} | \Delta_i \cdot I_{s_i} = c_{h_i}\}$ is the statement-iteration hyperplane in $\text{SIS}(s_i)$, for $i = 1, 2, \dots, m$. $\Phi_g(D_{v_j}) = \{D_{v_j} | \Theta_j \cdot D_{v_j} = c_{g_j}\}$ is the data hyperplane in $\text{DS}(v_j)$, for $j = 1, 2, \dots, n$. $\Psi_h(I_{s_i})$ and $\Phi_g(D_{v_j})$ are communication-free hyperplane partitions if and only if the following conditions hold:

(C1) $\forall i, \Theta_j \cdot F_k^{s_i, v_j} = \Theta_j \cdot F_1^{s_i, v_j}$, for $j = 1, 2, \dots, n$; $k = 2, 3, \dots, r_{i,j}$.

(C2) $\forall i, \Theta_j \cdot F_1^{s_i, v_j} = \Theta_1 \cdot F_1^{s_i, v_1}$, for $j = 2, 3, \dots, n$.

(C3) $\forall i, \Theta_j \cdot f_k^{s_i, v_j} = \Theta_j \cdot f_1^{s_i, v_j}$, for $j = 1, 2, \dots, n$; $k = 2, 3, \dots, r_{i,j}$.

(C4) $\Theta_j \cdot (f_1^{s_i, v_j} - f_1^{s_1, v_j}) = \Theta_1 \cdot (f_1^{s_i, v_1} - f_1^{s_1, v_1})$, for $i = 2, 3, \dots, m$; $j = 2, 3, \dots, n$.

- (C5) $\forall j, \Theta_j^t \in (\cup_{i=1}^m \cup_{k=1}^{r_{i,j}} \text{Ker}(F_k^{s_i, v_j})^t)$.
- (C6) $\forall i, \Delta_i = \Theta_j \cdot F_k^{s_i, v_j}$, for some $j, k, j \in \{1, 2, \dots, n\}; k \in \{1, 2, \dots, r_{i,j}\}$.
- (C7) $\forall i, \Delta_i^t \in (\text{span}(\cup_{j=1}^n \cup_{k=1}^{r_{i,j}} \text{Ker}(F_k^{s_i, v_j})))^\perp$.
- (C8) $\forall j, j = 2, 3, \dots, n, c_{g_j} = c_{g_1} - \Theta_1 \cdot f_1^{s_i, v_1} + \Theta_j \cdot f_1^{s_i, v_j}$, for some $i, i \in \{1, 2, \dots, m\}$.
- (C9) $\forall i, c_{h_i} = c_{g_j} - (\Theta_j \cdot f_k^{s_i, v_j})$, for some $j, k, j \in \{1, 2, \dots, n\}, k \in \{1, 2, \dots, r_{i,j}\}$. \square

Theorem 2 can be used to determine whether a nested loop is communication-free. It can also be used as a procedure for finding a communication-free hyperplane partition systematically. Conditions (C1) to (C4) in Theorem 2 are used to find the data hyperplane coefficient vectors. Condition (C5) can be used to check whether the data hyperplane coefficient vectors found in the preceding steps are within the legal range. Following the determination of the data hyperplane coefficient vectors, the statement-iteration hyperplane coefficient vectors can be obtained using Condition (C6). Similarly, Condition (C7) can be used to check whether the statement-iteration hyperplane coefficient vectors are within the legal range. The data hyperplane constant terms and statement-iteration hyperplane constant terms can be obtained using Conditions (C8) and (C9), respectively. If one of the conditions is violated, the whole procedure will stop and verify that the nested loop has no communication-free hyperplane partition.

On the other hand, combining Eqs. [3] and [5] together, a sufficient condition of communication-free hyperplane partition can be derived as follows:

$$\Theta_j \left(F_1^{s_i, v_j} - F_2^{s_i, v_j}, F_1^{s_i, v_j} - F_3^{s_i, v_j}, \dots, F_1^{s_i, v_j} - F_{r_{i,j}}^{s_i, v_j}, \right. \\ \left. f_1^{s_i, v_j} - f_2^{s_i, v_j}, f_1^{s_i, v_j} - f_3^{s_i, v_j}, \dots, f_1^{s_i, v_j} - f_{r_{i,j}}^{s_i, v_j} \right) = 0,$$

for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$. To satisfy the constraint that Θ is a nonzero row vector, the following condition should be satisfied:

$$\text{Rank} \left(F_1^{s_i, v_j} - F_2^{s_i, v_j}, \dots, F_1^{s_i, v_j} - F_{r_{i,j}}^{s_i, v_j}, \right. \\ \left. f_1^{s_i, v_j} - f_2^{s_i, v_j}, \dots, f_1^{s_i, v_j} - f_{r_{i,j}}^{s_i, v_j} \right) < \dim(DS(v_j)), \quad [9]$$

for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$. Note that this condition is similar to the result in [7] for loop-level hyperplane partition. We conclude with the following corollary.

Corollary 3 Suppose $\mathcal{S} = \{s_1, s_2, \dots, s_m\}$ and $\mathcal{D} = \{v_1, v_2, \dots, v_n\}$ are the set of statements and array variables, respectively. $F_k^{s_i, v_j}$ and $f_k^{s_i, v_j}$ are the array reference coefficient matrix and constant vector, respectively, where $i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}$, and $k \in \{1, 2, \dots, r_{i,j}\}$. If communication-free hyperplane partition exists, then Eq. [9] must hold. \square

Theorem 1 and Corollary 3 can be used to check for the absence of communication-free hyperplane partition for a nested loop because these conditions are

sufficient but not necessary. Theorem 1 is the statement-iteration space dimension test, and Corollary 3 is the data space dimension test. To determine the existence of a communication-free hyperplane partition, we need to check the conditions in Theorem 2. We will use the following example to explain how communication-free hyperplanes of statement-iteration spaces and data spaces are found.

Example 5. Reconsider loop $L1$. Since $m = 2$ and $n = 2$, the set of statements \mathcal{S} is $\{s_1, s_2\}$, and the set of array variables \mathcal{D} is $\{v_1, v_2\}$, where $v_1 = A$ and $v_2 = B$. The number of occurrences of array variables is $r_{1,1} = 2$, $r_{1,2} = 1$, $r_{2,1} = 1$, and $r_{2,2} = 2$. From Section 2.1, the array reference coefficient matrices and constant vectors for statements s_1 and s_2 are as listed below, respectively:

$$\begin{aligned} F_1^{s_1, v_1} &= \begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix}, & F_2^{s_1, v_1} &= \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, & F_1^{s_1, v_2} &= \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, \\ f_1^{s_1, v_1} &= \begin{bmatrix} 0 \\ -1 \end{bmatrix}, & f_2^{s_1, v_1} &= \begin{bmatrix} -1 \\ 1 \end{bmatrix}, & f_1^{s_1, v_2} &= \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \\ F_1^{s_2, v_2} &= \begin{bmatrix} 1 & -1 \\ 1 & -2 \end{bmatrix}, & F_1^{s_2, v_1} &= \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}, & F_2^{s_2, v_2} &= \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \\ f_1^{s_2, v_2} &= \begin{bmatrix} 1 \\ 1 \end{bmatrix}, & f_1^{s_2, v_1} &= \begin{bmatrix} -1 \\ 0 \end{bmatrix}, & f_2^{s_2, v_2} &= \begin{bmatrix} -1 \\ -2 \end{bmatrix}. \end{aligned}$$

Since $\dim(\text{span}(\bigcup_{j=1}^2 \bigcup_{k=1}^{r_{i,j}} \text{Ker}(F_k^{s_i, v_j}))) = 1$ is less than $\dim(\text{SIS}(s_i)) = 2$, for $i = 1, 2$. By Theorem 1, a communication-free hyperplane partition may exist for loop L_1 . Again, using Corollary 3, the loop can be tested for the possible existence of a nontrivial communication-free hyperplane partition. For array variable v_1 , the following inequality is satisfied:

$$\text{Rank}(F_1^{s_1, v_1} - F_2^{s_1, v_1}, f_1^{s_1, v_1} - f_2^{s_1, v_1}) = 1 < \dim(\text{DS}(v_1)) = 2.$$

Similarly, with respect to the array variable v_2 , the following inequality is obtained:

$$\text{Rank}(F_1^{s_2, v_2} - F_2^{s_2, v_2}, f_1^{s_2, v_2} - f_2^{s_2, v_2}) = 1 < \dim(\text{DS}(v_2)) = 2.$$

Although Eq. [9] holds for all array variables, it still can not ensure that the loop has a nontrivial communication-free hyperplane partition.

Using Theorem 2, we can further check loop $L1$ for the existence of a nontrivial communication-free hyperplane partition. In the mean time, the statement-iteration and data hyperplanes will be derived if they exist. Recall that the dimensions of data spaces $\text{DS}(v_1)$ and $\text{DS}(v_2)$ are two, and that Θ_1 and Θ_2 can be assumed to be $[\theta_{11}, \theta_{12}]$ and $[\theta_{21}, \theta_{22}]$, respectively. The conditions listed in Theorem 2 will be checked to determine the hyperplane coefficient vectors and constants. By Condition (C1) in Theorem 2, the following equations are obtained:

$$i = 1, j = 1, \text{ and } k = 2: \quad \Theta_1 \cdot F_2^{s_1, v_1} = \Theta_1 \cdot F_1^{s_1, v_1},$$

$$i = 2, j = 2, \text{ and } k = 2: \quad \Theta_2 \cdot F_2^{s_2, v_2} = \Theta_2 \cdot F_1^{s_2, v_2}.$$

By Condition (C2) in Theorem 2, we have

$$i = 1 \text{ and } j = 2: \quad \Theta_2 \cdot F_1^{s_1, u_2} = \Theta_1 \cdot F_1^{s_1, u_1},$$

$$i = 2 \text{ and } j = 2: \quad \Theta_2 \cdot F_1^{s_2, u_2} = \Theta_1 \cdot F_1^{s_2, u_1}.$$

By Condition (C3) in Theorem 2,

$$i = 1, j = 1, \text{ and } k = 2: \quad \Theta_1 \cdot f_2^{s_1, u_1} = \Theta_1 \cdot f_1^{s_1, u_1},$$

$$i = 2, j = 2, \text{ and } k = 2: \quad \Theta_2 \cdot f_2^{s_2, u_2} = \Theta_2 \cdot f_1^{s_2, u_2}.$$

By Condition (C4) in Theorem 2,

$$i = 2 \text{ and } j = 2: \quad \Theta_2 \cdot (f_1^{s_2, u_2} - f_1^{s_1, u_2}) = \Theta_1 \cdot (f_1^{s_2, u_1} - f_1^{s_1, u_1}).$$

Substituting $[\theta_{11}, \theta_{12}]$ and $[\theta_{21}, \theta_{22}]$ for Θ_1 and Θ_2 , respectively, the above equations form a homogeneous linear system. Solving this homogeneous linear system, we obtain the general solution $(\theta_{11}, \theta_{12}, \theta_{21}, \theta_{22}) = (2r, r, 3r, -2r)$, where $r \in \mathbf{Q} - \{0\}$. Therefore, $\Theta_1 = [2r, r]$ and $\Theta_2 = [3r, -2r]$. Next, we will check whether Θ_1 and Θ_2 satisfy Condition (C5) or not. From Condition (C5): $(\cup_{i=1}^m \cup_{k=1}^{i-1} \text{Ker}((F_k^{s_i, u_1})^t))$, we obtain $\{c_1[1, 1]^t | c_1 \in \mathbf{Q} - \{0\}\}$. This implies that $(\cup_{i=1}^m \cup_{k=1}^{i-1} \text{Ker}((F_k^{s_i, u_1})^t))^c = \{[r_1, r_2]^t | r_1 \neq r_2, r_1, r_2 \in \mathbf{Q} - \{0\}\}$. Thus, $\Theta_1^t = [2r, r]^t \in (\cup_{i=1}^m \cup_{k=1}^{i-1} \text{Ker}((F_k^{s_i, u_1})^t))^c$. Similarly, $\Theta_2^t = [3r, -2r]^t \in (\cup_{i=1}^m \cup_{k=1}^{i-1} \text{Ker}((F_k^{s_i, u_2})^t))^c$. Since Θ_1^t and Θ_2^t satisfy Condition (C5), they are legal.

From Loop $L1$, the two statement-iteration hyperplane coefficient vectors can be determined using Condition (C6) in Theorem 2:

$$\Delta_1 = \Theta_1 \cdot F_1^{s_1, u_1} = \Theta_1 \cdot F_2^{s_1, u_1} = \Theta_2 \cdot F_1^{s_1, u_2} = (r, -r),$$

$$\Delta_2 = \Theta_2 \cdot F_1^{s_2, u_2} = \Theta_1 \cdot F_1^{s_2, u_1} = \Theta_2 \cdot F_2^{s_2, u_2} = (r, r).$$

Note that the statement-iteration hyperplane coefficient vectors may be obtained using many different equations; e.g., Δ_1 can be obtained using $\Theta_1 \cdot F_1^{s_1, u_1}$, $\Theta_1 \cdot F_2^{s_1, u_1}$, or $\Theta_2 \cdot F_1^{s_1, u_2}$. Conditions (C1) and (C2) in Theorem 2 ensure that all the equations lead to the same result.

In the next step, we use Condition (C7) to check whether Δ_1 and Δ_2 are legal or not. The check of legality is similar to Example 4. We have that Δ_1 and Δ_2 are legal. Next, we will determine the data hyperplane constant terms. Because the hyperplanes are related to each other, once a hyperplane constant term is determined, the other constant terms will be determined accordingly. Assuming that c_{g_1} is known, c_{g_2} , c_{h_1} , and c_{h_2} can be determined using Conditions (C8) and (C9) as follows:

$$c_{g_2} = c_{g_1} - \Theta_1 \cdot f_1^{s_1, u_1} + \Theta_2 \cdot f_1^{s_1, u_2} = c_{g_1} - \Theta_1 \cdot f_1^{s_2, u_1} + \Theta_2 \cdot f_1^{s_2, u_2} = c_{g_1} + 3r,$$

$$c_{h_1} = c_{g_1} - \Theta_1 \cdot f_1^{s_1, u_1} = c_{g_1} - \Theta_1 \cdot f_2^{s_1, u_1} = c_{g_2} - \Theta_2 \cdot f_1^{s_1, u_2} = c_{g_1} + r,$$

$$c_{h_2} = c_{g_2} - \Theta_2 \cdot f_1^{s_2, v_2} = c_{g_1} - \Theta_1 \cdot f_1^{s_2, v_1} = c_{g_2} - \Theta_2 \cdot f_2^{s_2, v_2} = c_{g_1} + 2r.$$

Similarly, statement-iteration and data hyperplane constant terms can be evaluated using many different equations. However, Conditions (C3) and (C4) in Theorem 2 ensure that they all lead to the same values.

It is clear that there exists at least one set of nonzero statement-iteration and data hyperplane coefficient vectors such that the conditions listed in Theorem 2 are all satisfied. By Theorem 2, this fact implies that the nested loop has a nontrivial communication-free hyperplane partition. The partition group is defined as the set of statement-iteration and data hyperplanes that are allocated to a processor. The partition group for this example is as follows. $G = \Psi_{h_1}(I_{s_1}) \cup \Psi_{h_2}(I_{s_2}) \cup \Phi_{g_1}(D_{v_1}) \cup \Phi_{g_2}(D_{v_2})$, where

$$\Psi_{h_1}(I_{s_1}) = \{I_{s_1}[[r, -r] \cdot I_{s_1} = c_{g_1} + r],$$

$$\Psi_{h_2}(I_{s_2}) = \{I_{s_2}[[r, r] \cdot I_{s_2} = c_{g_1} + 2r],$$

$$\Phi_{g_1}(D_{v_1}) = \{D_{v_1}[[2r, r] \cdot D_{v_1} = c_{g_1}],$$

$$\Phi_{g_2}(D_{v_2}) = \{D_{v_2}[[3r, -2r] \cdot D_{v_2} = c_{g_1} + 3r].$$

Given loop bounds $1 \leq i_1 \leq 5$ and $1 \leq i_2 \leq 5$, for $r = 1$, the constant term c_{g_1} corresponding to statement-iteration hyperplane coefficient vectors Δ_1 and Δ_2 range from -5 to 3 and from 0 to 8 , respectively. The intersecting part of these two ranges means that the two statement-iteration hyperplanes have to be coupled together onto a processor. Only one statement-iteration hyperplane, either Δ_1 or Δ_2 , is allocated to a processor. The constant terms c_{g_2} , c_{h_1} , and c_{h_2} are evaluated to obtain the following values:

$$-2 \leq c_{g_2} \leq 11, \quad -4 \leq c_{h_1} \leq 4, \quad \text{and} \quad 2 \leq c_{h_2} \leq 10.$$

The corresponding parallelized program is as follows:

```

doall  $c = -5, 8$ 
  do  $i_1 = \max(c - 3, 1), \min(c + 1, 5)$ 
     $i_2 = -i_1 + c + 2$ 
     $B[i_1 - i_2 + 1, i_1 - 2i_2 + 1]$ 
     $= A[i_2 - 1, i_1 - i_2] * B[i_1 + i_2 - 1, i_1 + i_2 - 2]$ 
  enddo
  do  $i_1 = \max(c + 2, 1), \min(c + 6, 5)$ 
     $i_2 = i_1 - c - 1$ 
     $A[i_1, -i_1 - i_2 - 1]$ 
     $= A[i_1 - i_2 - 1, -i_1 + i_2 + 1] + B[i_1 + i_2, i_1 + 2i_2 - 1]$ 
  enddo
enddoall

```

Figure 3 illustrates the communication-free hyperplane partitions for a particular partition group, where $r = 1$ and $c_{g_1} = 2$. □

The communication-free hyperplane partitioning technique for a perfectly nested loop has been discussed in this section. Our method treats statements within a loop body as separate schedulable units and considers both iteration and data spaces at the same time. Partitioning groups are determined using affine array reference functions directly, instead of using data dependence vectors.

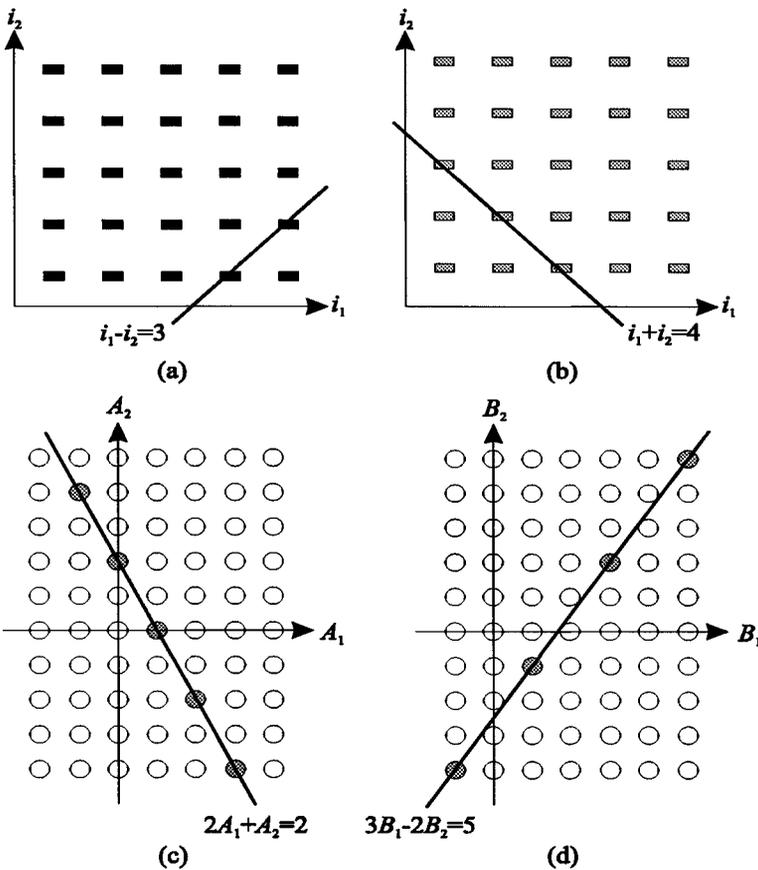


Figure 3. Communication-free statement-iteration hyperplanes and data hyperplanes for a partition group of loop (L_1) , where $r = 1$ and $c_{g_1} = 2$. (a) Statement-iteration hyperplane of $SIS(s_1)$. (b) Statement-iteration hyperplane of $SIS(s_2)$. (c) Data hyperplane of $DS(A)$. (d) Data hyperplane of $DS(B)$.

5. Communication-free hyperplane partition for sequences of imperfectly nested loops

The conditions presented in Section 4 for communication-free hyperplane partition can be applied to the general case for sequences of imperfectly nested loops. In a perfectly nested loop, all statements are enclosed at the same depth of the nested loop; i.e., the statement-iteration space of each statement has the same dimensionality. The statement-iteration spaces of two statements in imperfectly nested loops may have different dimensions. Since each statement-iteration is a schedulable unit and the partitioning technique is independent of the dimensionality of statement-iteration spaces, Theorem 2 can be directly applied to sequences of imperfectly nested loops. The following example demonstrates the use of the technique applied to sequences of imperfectly nested loops.

Example 6. Consider the following sequences of nested loops L_2 :

```

do  $i_1 = 1, N$ 
  do  $i_2 = 1, N$ 
 $s_1: A[i_1 + i_2, 1] = B[i_1 + i_2 + 1, i_1 + i_2 + 2]$ 
       $+ C[i_1 + 1, -2i_1 + 2i_2, 2i_1 - i_2 + 1]$ 
    do  $i_3 = 1, N$ 
 $s_2: B[i_1 + i_3 + 1, i_2 + i_3 + 1]$ 
       $= A[2i_1 + 2i_3, i_2 + i_3]$ 
       $+ C[i_1 + i_2 + 1, -i_2 + i_3 + 1, i_1 - i_2 + 1]$ 
    enddo
  enddo
enddo
do  $i_1 = 1, N$ 
  do  $i_2 = 1, N$ 
    do  $i_3 = 1, N$ 
 $s_3: C[i_1 + 1, i_2, i_2 + i_3] = A[2i_1 + 3i_2 + i_3, i_1 + i_2 + 2]$ 
       $+ B[i_1 + i_2, i_1 - i_3 + 1]$ 
    enddo
 $s_4: A[i, i_2 + 3] = B[i_1 - i_2, i_1 - i_2 + 2] + C[i_1 + i_2, -i_2, -i_2]$ 
  enddo
enddo

```

(L_2)

The set of statements \mathcal{S} is $\{s_1, s_2, s_4, s_4\}$. The set of array variables is $\mathcal{D} = \{v_1, v_2, v_3\}$, where v_1, v_2 , and v_3 represent A, B , and C , respectively. The values of $r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}, r_{41}, r_{42}$, and r_{43} are all 1. We use Theorem 1 and Corollary 3 to verify whether (L_2) has no communication-free hyperplane partition. Since $\dim(\sum_{v \in \mathcal{D}} \text{Ker}(F_1^{s_i, v})) = 1$, which is smaller than $\dim(\text{SIS}(s_i))$, for

$i = 1, \dots, 4$, Theorem 1 can not ensure that $(L2)$ has no communication-free hyperplane partition. Corollary 3 is useless here because all the values of r_{ij} are 1, for $i = 1, \dots, 4; j = 1, \dots, 3$. Further examination is necessary because Theorem 1 and Corollary 3 can not prove that $(L2)$ has no communication-free hyperplane partition. From Theorem 2, if a communication-free hyperplane partition exists, the conditions listed in Theorem 2 should be satisfied; otherwise, $(L2)$ has no communication-free hyperplane partition.

Because the dimensions of the data spaces $DS(v_1)$, $DS(v_2)$, and $DS(v_3)$ are 2, 2, and 3, respectively, without loss of generality, the data hyperplane coefficient vectors can be, respectively, assumed to be $\Theta_1 = [\theta_{11}, \theta_{12}]$, $\Theta_2 = [\theta_{21}, \theta_{22}]$, and $\Theta_3 = [\theta_{31}, \theta_{32}, \theta_{33}]$. In what follows, the requirements which must be satisfied for communication-free hyperplane partition to be feasible are examined one-by-one.

There is no need to examine Conditions (C1) and (C3) because all the values of r_{ij} are 1. By Condition (C2), we obtain

$$\begin{aligned}\Theta_2 \cdot F_1^{s_1, v_2} &= \Theta_1 \cdot F_1^{s_1, v_1}, & \Theta_3 \cdot F_1^{s_1, v_3} &= \Theta_1 \cdot F_1^{s_1, v_1}, \\ \Theta_2 \cdot F_1^{s_2, v_2} &= \Theta_1 \cdot F_1^{s_2, v_1}, & \Theta_3 \cdot F_1^{s_2, v_3} &= \Theta_1 \cdot F_1^{s_2, v_1}, \\ \Theta_2 \cdot F_1^{s_3, v_2} &= \Theta_1 \cdot F_1^{s_3, v_1}, & \Theta_3 \cdot F_1^{s_3, v_3} &= \Theta_1 \cdot F_1^{s_3, v_1}, \\ \Theta_2 \cdot F_1^{s_4, v_2} &= \Theta_1 \cdot F_1^{s_4, v_1}, & \Theta_3 \cdot F_1^{s_4, v_3} &= \Theta_1 \cdot F_1^{s_4, v_1}.\end{aligned}$$

By Condition (C4), we obtain

$$\begin{aligned}\Theta_2 \cdot (f_1^{s_2, v_2} - f_1^{s_1, v_2}) &= \Theta_1 \cdot (f_1^{s_2, v_1} - f_1^{s_1, v_1}), \\ \Theta_3 \cdot (f_1^{s_2, v_3} - f_1^{s_1, v_3}) &= \Theta_1 \cdot (f_1^{s_2, v_1} - f_1^{s_1, v_1}), \\ \Theta_2 \cdot (f_1^{s_3, v_2} - f_1^{s_1, v_2}) &= \Theta_1 \cdot (f_1^{s_3, v_1} - f_1^{s_1, v_1}), \\ \Theta_3 \cdot (f_1^{s_3, v_3} - f_1^{s_1, v_3}) &= \Theta_1 \cdot (f_1^{s_3, v_1} - f_1^{s_1, v_1}), \\ \Theta_2 \cdot (f_1^{s_4, v_2} - f_1^{s_1, v_2}) &= \Theta_1 \cdot (f_1^{s_4, v_1} - f_1^{s_1, v_1}), \\ \Theta_3 \cdot (f_1^{s_4, v_3} - f_1^{s_1, v_3}) &= \Theta_1 \cdot (f_1^{s_4, v_1} - f_1^{s_1, v_1}).\end{aligned}$$

After solving the above linear system, the general solutions are $(\theta_{11}, \theta_{12}, \theta_{21}, \theta_{22}, \theta_{31}, \theta_{32}, \theta_{33}) = (t, -t, 2t, -t, t, t, t)$, $t \in \mathbf{Q} - \{0\}$. Therefore, $\Theta_1 = [t, -t]$, $\Theta_2 = [2t, -t]$ and $\Theta_3 = [t, t, t]$.

The verification of Condition (C5) is as follows:

$$\begin{aligned}\bigcup_{i=1}^m \text{Ker}((F_1^{s_i, v_1})^t) &= \{c_1[0, 1]^t | c_1 \in \mathbf{Q} - \{0\}\} \\ \Rightarrow \Theta_1^t &\in \left(\bigcup_{i=1}^m \text{Ker}((F_1^{s_i, v_1})^t) \right)' \\ &\cdot \bigcup_{i=1}^m \text{Ker}((F_1^{s_i, v_2})^t) \\ &= \{c_2[1, -1]^t | c_2 \in \mathbf{Q} - \{0\}\} \cup \{c_3[1, 1]^t | c_3 \in \mathbf{Q} - \{0\}\}\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad \Theta'_2 &\in \left(\bigcup_{i=1}^m \text{Ker} \left((F_1^{s_i, v_2})^t \right) \right)' \\
&\cdot \bigcup_{i=1}^m \text{Ker} \left((F_1^{s_i, v_3})^t \right) \\
&= \{c_4[2, -1, -2]^t | c_4 \in \mathbf{Q} - \{0\}\} \cup \{c_5[0, 1, -1]^t | c_5 \in \mathbf{Q} - \{0\}\} \\
\Rightarrow \quad \Theta'_3 &\in \left(\bigcup_{i=1}^m \text{Ker} \left((F_1^{s_i, v_3})^t \right) \right)'.
\end{aligned}$$

All the data hyperplane coefficient vectors are within the legal range.

The statement-iteration hyperplane coefficient vectors can be determined by Condition (C6) as follows:

$$\begin{aligned}
\Delta_1 &= \Theta_1 \cdot F_1^{s_1, v_1} = \Theta_2 \cdot F_1^{s_1, v_2} = \Theta_3 \cdot F_1^{s_1, v_3} = [t, t], \\
\Delta_2 &= \Theta_1 \cdot F_1^{s_2, v_1} = \Theta_2 \cdot F_1^{s_2, v_2} = \Theta_3 \cdot F_1^{s_2, v_3} = [2t, -t, t], \\
\Delta_3 &= \Theta_1 \cdot F_1^{s_3, v_1} = \Theta_2 \cdot F_1^{s_3, v_2} = \Theta_3 \cdot F_1^{s_3, v_3} = [t, 2t, t], \\
\Delta_4 &= \Theta_1 \cdot F_1^{s_4, v_1} = \Theta_2 \cdot F_1^{s_4, v_2} = \Theta_3 \cdot F_1^{s_4, v_3} = [t, -t].
\end{aligned}$$

The legality of these statement-iteration hyperplane coefficient vectors can then be checked using Condition (C7) as follows:

$$\begin{aligned}
&\text{span} \left(\bigcup_{j=1}^n \text{Ker} (F_1^{s_1, v_j}) \right) = \{c_6[1, -1]^t | c_6 \in \mathbf{Q} - \{0\}\} \\
\Rightarrow \quad \Delta'_1 &\in \left(\text{span} \left(\bigcup_{j=1}^n \text{Ker} (F_1^{s_1, v_j}) \right) \right)^\perp, \\
&\text{span} \left(\bigcup_{j=1}^n \text{Ker} (F_1^{s_2, v_j}) \right) = \{c_7[1, 1, -1]^t | c_7 \in \mathbf{Q} - \{0\}\} \\
\Rightarrow \quad \Delta'_2 &\in \left(\text{span} \left(\bigcup_{j=1}^n \text{Ker} (F_1^{s_2, v_j}) \right) \right)^\perp, \\
&\text{span} \left(\bigcup_{j=1}^n \text{Ker} (F_1^{s_3, v_j}) \right) = \{c_8[1, -1, 1]^t | c_8 \in \mathbf{Q} - \{0\}\} \\
\Rightarrow \quad \Delta'_3 &\in \left(\text{span} \left(\bigcup_{j=1}^n \text{Ker} (F_1^{s_3, v_j}) \right) \right)^\perp, \\
&\text{span} \left(\sum_{j=1}^n \text{Ker} (F_1^{s_4, v_j}) \right) = \{c_9[1, 1]^t | c_9 \in \mathbf{Q} - \{0\}\} \\
\Rightarrow \quad \Delta'_4 &\in \left(\text{span} \left(\bigcup_{j=1}^n \text{Ker} (F_1^{s_4, v_j}) \right) \right)^\perp.
\end{aligned}$$

From the above observation, all the statement-iteration and data hyperplane coefficient vectors are legal. This fact reveals that the nested loops have communication-free hyperplane partitions. Next, the data and statement-iteration hyperplanes constant terms are decided.

First, let one data hyperplane constant term be fixed, say c_{g_1} . The rest of the data hyperplane constant terms can be determined by Condition (C8):

$$\begin{aligned} c_{g_2} &= c_{g_1} - \Theta_1 \cdot f_1^{s_1, \nu_1} + \Theta_2 \cdot f_1^{s_1, \nu_2} = c_{g_1} - \Theta_1 \cdot f_1^{s_2, \nu_1} + \Theta_2 \cdot f_1^{s_2, \nu_2} \\ &= c_{g_1} - \Theta_1 \cdot f_1^{s_3, \nu_1} + \Theta_2 \cdot f_1^{s_3, \nu_2} = c_{g_1} - \Theta_1 \cdot f_1^{s_4, \nu_1} + \Theta_2 \cdot f_1^{s_4, \nu_2} \\ &= c_{g_1} + t. \end{aligned}$$

$$\begin{aligned} c_{g_3} &= c_{g_1} - \Theta_1 \cdot f_1^{s_1, \nu_1} + \Theta_3 \cdot f_1^{s_1, \nu_3} = c_{g_1} - \Theta_1 \cdot f_1^{s_2, \nu_1} + \Theta_3 \cdot f_1^{s_2, \nu_3} \\ &= c_{g_1} - \Theta_1 \cdot f_1^{s_3, \nu_1} + \Theta_3 \cdot f_1^{s_3, \nu_3} = c_{g_1} - \Theta_1 \cdot f_1^{s_4, \nu_1} + \Theta_3 \cdot f_1^{s_4, \nu_3} \\ &= c_{g_1} + 3t. \end{aligned}$$

Similarly, the statement-iteration hyperplane constant terms can be determined by Condition (C9) after the data hyperplane constant terms have been decided:

$$\begin{aligned} c_{h_1} &= c_{g_1} - \Theta_1 \cdot f_1^{s_1, \nu_1} = c_{g_2} - \Theta_2 \cdot f_1^{s_1, \nu_2} = c_{g_3} - \Theta_3 \cdot f_1^{s_1, \nu_3} = c_{g_1} + t, \\ c_{h_2} &= c_{g_1} - \Theta_1 \cdot f_1^{s_2, \nu_1} = c_{g_2} - \Theta_2 \cdot f_1^{s_2, \nu_2} = c_{g_3} - \Theta_3 \cdot f_1^{s_2, \nu_3} = c_{g_1}, \\ c_{h_3} &= c_{g_1} - \Theta_1 \cdot f_1^{s_3, \nu_1} = c_{g_2} - \Theta_2 \cdot f_1^{s_3, \nu_2} = c_{g_3} - \Theta_3 \cdot f_1^{s_3, \nu_3} = c_{g_1} + 2t, \\ c_{h_4} &= c_{g_1} - \Theta_1 \cdot f_1^{s_4, \nu_1} = c_{g_2} - \Theta_2 \cdot f_1^{s_4, \nu_2} = c_{g_3} - \Theta_3 \cdot f_1^{s_4, \nu_3} = c_{g_1} + 3t. \end{aligned}$$

The corresponding partition group is as follows: $G = \Psi_{h_1}(I_{s_1}) \cup \Psi_{h_2}(I_{s_2}) \cup \Psi_{h_3}(I_{s_3}) \cup \Psi_{h_4}(I_{s_4}) \cup \Phi_{g_1}(D_{v_1}) \cup \Phi_{g_2}(D_{v_2}) \cup \Phi_{g_3}(D_{v_3})$, where

$$\begin{aligned} \Psi_{h_1}(I_{s_1}) &= \{I_{s_1}[[t, t] \cdot I_{s_1} = c_{g_1} + t\}, \\ \Psi_{h_2}(I_{s_2}) &= \{I_{s_2}[[2t, -t, t] \cdot I_{s_2} = c_{g_1}\}, \\ \Psi_{h_3}(I_{s_3}) &= \{I_{s_3}[[t, 2t, t] \cdot I_{s_3} = c_{g_1} + 2t\}, \\ \Psi_{h_4}(I_{s_4}) &= \{I_{s_4}[[t, -t] \cdot I_{s_4} = c_{g_1} + 3t\}, \\ \Phi_{g_1}(D_{v_1}) &= \{D_{v_1}[[t, -t] \cdot D_{v_1} = c_{g_1}\}, \\ \Phi_{g_2}(D_{v_2}) &= \{D_{v_2}[[2t, -t] \cdot D_{v_2} = c_{g_1} + t\}, \\ \Phi_{g_3}(D_{v_3}) &= \{D_{v_3}[[t, t, t] \cdot D_{v_3} = c_{g_1} + 3t\}. \end{aligned}$$

Figure 4 illustrates the communication-free hyperplane partitions for a partition group, where $t = 1$ and $c_{g_1} = 0$. The corresponding parallelized program is as

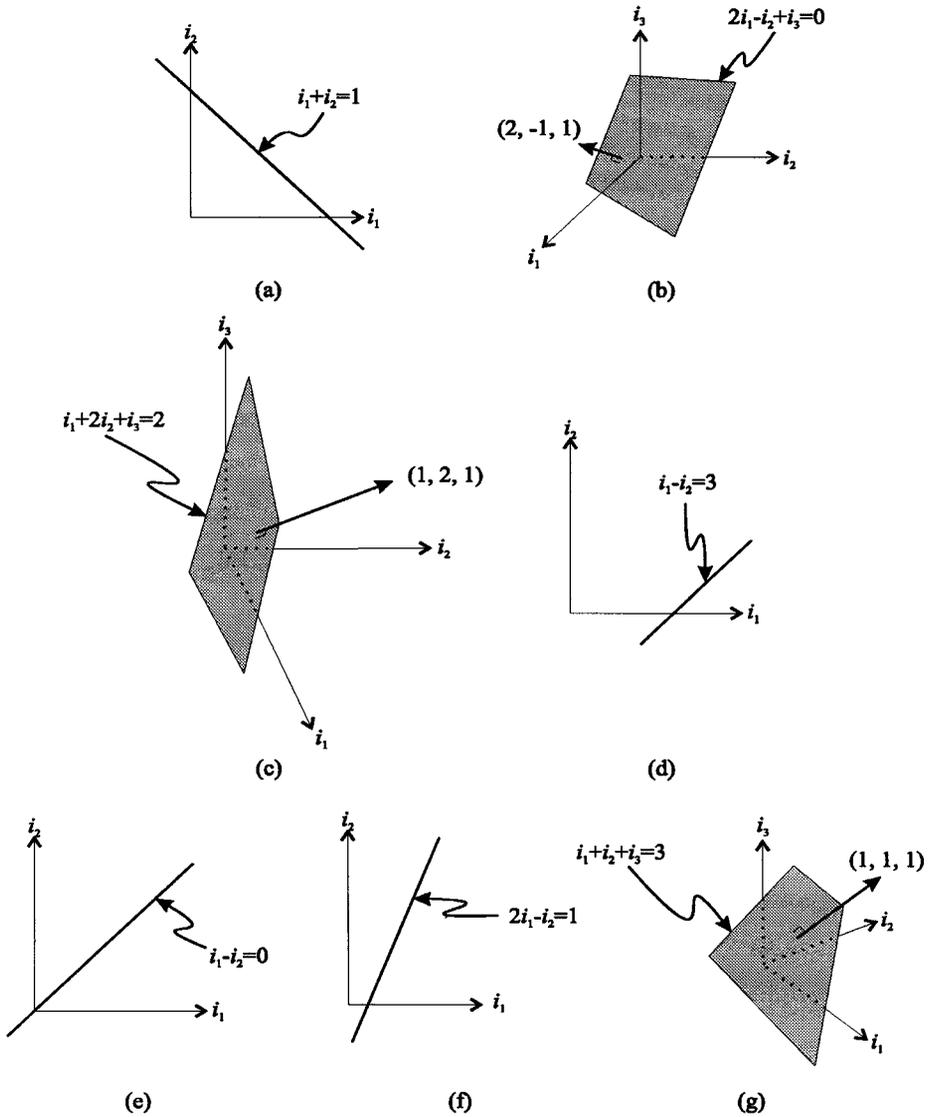


Figure 4. Communication-free statement-iteration hyperplanes and data hyperplanes for a partition group of loop (L_2) , where $t = 1$ and $c_{g_1} = 0$. (a) Statement-iteration hyperplane of $SIS(s_1)$. (b) Statement-iteration hyperplane of $SIS(s_2)$. (c) Statement-iteration hyperplane of $SIS(s_3)$. (d) Statement-iteration hyperplane of $SIS(s_4)$. (e) Data hyperplane of $DS(A)$. (f) Data hyperplane of $DS(B)$. (g) Data hyperplane of $DS(C)$.

follows:

```

doall  $c = -7, 18$ 
  do  $i_1 = \max\left(\min\left(c - 4, \left\lceil \frac{c - 4}{2} \right\rceil\right), 1\right), \min\left(\max\left(c, \left\lfloor \frac{c + 4}{2} \right\rfloor\right), 5\right)$ 
    if( $\max(c - 4, 1) \leq i_1 \leq \min(c, 5)$ )
       $i_2 = c - i_1 + 1$ 
       $A[i_1 + i_2, 1] = B[i_1 + i_2 - 1, i_1 + i_2 + 2]$ 
         $+ C[i_1 + 1, -2i_1 + 2i_2, 2i_1 - i_2 + 1]$ 
    endif
    do  $i_2 = \max(2i_1 - c + 1, 1), \min(2i_1 - c + 5, 5)$ 
       $i_3 = c - 2i_1 + i_2$ 
       $B[i_1 + i_3 + 1, i_2 + i_3 + 1]$ 
       $= A[2i_1 + 2i_3, i_2 + i_3] + C[i_1 + i_2 + 1, -i_2 + i_3 + 1, i_1 - i_2 + 1]$ 
    enddo
  enddo
  do  $i_1 = \max(c - 13, 1), \min(c - 1, 5)$ 
    do  $i_2 = \max\left(\left\lceil \frac{c - i_1 - 3}{2} \right\rceil, 1\right), \min\left(\left\lfloor \frac{c - i_1 + 1}{2} \right\rfloor, 5\right)$ 
       $i_3 = c - i_1 - 2i_2 + 2$ 
       $C[i_1 + 1, i_2, i_2 + i_3] = A[2i_1 + 3i_2 + i_3, i_1 + i_2 + 2]$ 
         $+ B[i_1 + i_2, i_1 - i_3 + 1]$ 
    enddo
  enddo
  do  $i_1 = \max(c + 4, 1), \min(c + 8, 5)$ 
     $i_2 = i_1 - c - 3$ 
     $A[i_1, i_2 + 3] = B[i_1 - i_2, i_1 - i_2 + 2] + C[i_1 + i_2, -i_2, -i_2]$ 
  enddo
enddoall

```

6. Conclusions

This paper has presented techniques for finding statement-level communication-free hyperplane partition for a perfectly nested loop and sequences of imperfectly nested loops. The necessary and sufficient conditions for communication-free partition along hyperplanes to be feasible have been proposed. The techniques can be applied to loops with affine array references and do not use any information about data dependence distances or direction vectors.

Although our goal is to determine communication-free partition for loops, in reality, most loops are not communication-free. If a program is not communication-free, then the technique can be used to identify subsets of statement-iteration and data spaces which are communication-free. For other statement-iterations, it is necessary to generate communication code. Two important tasks in our future work will be to develop heuristics for searching a subset of statement-iterations which is communication-free, and to generate efficient code when communication is inevitable.

Note

1. Note that Δ is a row vector. However, it is Δ^t , but not Δ , that is orthogonal to $\Psi_i(s)$.

Acknowledgments

This work was supported by National Science Council of the Republic of China under grant NSC 88-2213-E-008-020.

References

1. J. M. Anderson and M. S. Lam. Global optimizations for parallelism and locality on scalable parallel machines. In *Proceedings of the ACM SIGPLAN'93 Conference on Programming Language Design and Implementation*, pp. 112–125, June 1993.
2. T. S. Chen and J. P. Sheu. Communication-free data allocation techniques for parallelizing compilers on multicomputers. *IEEE Transactions on Parallel and Distributed Systems*, 5:924–938, Sept. 1994.
3. M. Gupta and P. Banerjee. Demonstration of automatic data partitioning techniques for parallelizing compilers on multicomputers. *IEEE Transactions on Parallel and Distributed Systems*, 3:179–193, Mar. 1992.
4. S. Hiranandani, K. Kennedy, and C. W. Tseng. Compiling Fortran D for MIMD distributed-memory machines. *Communications of the ACM*, 35:66–80, Aug. 1992.
5. S. Hiranandani, K. Kennedy, and C. W. Tseng. Evaluating compiler optimizations for Fortran D. *Journal of Parallel and Distributed Computing*, 21:27–45, 1994.
6. K. Koffman and R. Kunze. *Linear Algebra*, 2nd ed. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.
7. C.-H. Huang and P. Sadayappan. Communication-free hyperplane partitioning of nested loops. *Journal of Parallel and Distributed Computing*, 19:90–102, 1993.
8. A. H. Karp. Programming for parallelism. *IEEE Comput. Mag.*, 20:43–57, May 1987.
9. C. Koelbel. Compiling programs for nonshared memory machines. PhD thesis, Dept. of Computer Science, Purdue University, Nov. 1990.
10. C. Koelbel and P. Mehrotra. Compiling global name-space parallel loops for distributed execution. *IEEE Transactions on Parallel and Distributed Systems*, 2:440–451, Oct. 1991.
11. J. Li and M. Chen. Index domain alignment: Minimizing cost of cross-referencing between distributed arrays. In *Proceedings 3rd Symposium on the Frontiers of Massively Computation*, pp. 424–433, Oct. 1990.
12. J. Li and M. Chen. The data alignment phase in compiling programs for distributed-memory machines. *Journal of Parallel and Distributed Computing*, 13:213–221, 1991.

13. A. W. Lim and M. S. Lam. Communication-free parallelization via affine transformations. In *Proceedings of the 7th Workshop on Programming Languages and Compilers for Parallel Computing*, Aug. 1994.
14. M. Mace. *Memory Storage Patterns in Parallel Processing*. Kluwer Academic Publishers, 1987.
15. J. Ramanujam and P. Sadayappan. Compile-time techniques for data distribution in distributed memory machines. *IEEE Transactions on Parallel and Distributed Systems*, 2:472–482, Oct. 1991.
16. A. Rogers and K. Pingali. Process decomposition through locality of reference. In *Proceedings of the ACM SIGPLAN'89 Conference on Programming Language Design and Implementation*, pp. 69–80, June 1989.
17. M. Rosing, R. B. Schnabel, and R. P. Weaver. The dino parallel programming language. *Journal of Parallel and Distributed Computing*, 13:30–42, 1991.
18. C.-W. Tseng. An optimizing Fortran D compiler for MIMD distributed-memory machines, Ph.D. thesis, Dept. of Computer Science, Rice University, Jan. 1993.
19. M. E. Wolf and M. S. Lam. A loop transformation theory and an algorithm to maximize parallelism. *IEEE Transactions on Parallel and Distributed Systems*, 2:452–471, Oct. 1991.
20. M. J. Wolfe. *High Performance Compilers for Parallel Computing*. Addison-Wesley Publishing Company, 1996.
21. H. P. Zima, P. Brezany, and B. M. Chapman. SUPERB and Vienna Fortran. *Parallel Computing*, 20:1487–1517, 1994.