A General Data Dependence Analysis to Nested Loop Using Integer Interval Theory*

Jing Zhou [1,2], Guosun Zeng [1,2]

[1] Department of computer Science and Technology, Tongji University, Shanghai 200092, China
dinese@163.comm, gszeng@mail.tongji.edu.cn

[2] Tongji Branch, National Engineering & Technology Center of High Performance Computer, Shanghai 200092, China

Abstract

Many dependence tests have been proposed for loop parallelization in the case of arrays with linear subscripts, but little work has been done on the arrays with non-linear subscripts, which sometimes occur in parallel benchmarks and scientific and engineering applications. This paper focuses on array subscripts coupled integer power index variables. We attempt to use the integer interval theory to solve the above difficult dependence test problem. Some "interval solution" rules for polynomial equations have been proposed in this paper. Furthermore, based on the proposed rules, we present a novel approach to loop dependence analysis, which is termed the Polynomial Variable Interval test or PVI-test, and also develop a related algorithm. Some case studies show that the PVI-test is effective and efficient. Compared to the VI test, the PVI-test makes significant improvement, and is therefore a more general scheme of dependence test.

1 Introduction

Data dependence analysis is critical for parallel compiler to detect independent operations. Especially, a great deal of effort has been spent on loop dependence tests because of rich parallelism hidden in loops. The dependence analysis for array references, in fact, can be reduced to solving Diophantine equations [1]. It is a NP complete problem. Therefore a majority of dependence tests are heuristic methods in the literature. The GCD test [2] is based on the elementary number theory, which states that a linear equation has an integer solution if and only if the greatest common divisor of the coefficients divides the constant term. It is a necessary but not sufficient condition for data dependence. The Banerjee bound test [3] checks whether there is a real solution for the dependence equation. If the test produces yes answer, in practice, it implies the existence of dependence. The Omega test [4] is a general purpose algorithm that is based on Fourier-Motzkin variable elimination (FMVE) [5] and its integer extensions, while the Power test [6] combines the Generalized GCD test with FMVE [5] and takes loop limits and direction vector constraints into consideration. Though the two tests produce exact yes/no answers, they have worst-case exponential time complexity. The I-test [7] is a polynomial time test. Generally speaking, the I-test is a linear time exact test in most cases for single dimensional array references, but cannot precisely handle multidimensional array references involving coupled subscripts and rely on information known at compile time [8]. The DVI test [9] checks the existence of integer solutions and takes loop constant bounds and direction vector constraints into account for one-dimensional arrays. The GDVI test [10] extends the DVI test to handle loops with variable bounds. The IR test [11, 12] aims to a typical kind of loops with an arbitrary direction vector and triangular bound. The VI test [13], related to our research, has begun original work on dependence test using integer interval theory. But it can only prove or disprove the existence of data dependences in loops with linear subscripts. Other data dependence tests can be seen in the literature [14-21].

As mentioned above, in short, the most existing approaches to dependence test perform well only for nested loop with linear (affine) array subscripts, and in each test there is trade off between accuracy and efficiency. However, in the case of subscripts coupled integer power variable, such as $b_0+b_1\times i^1+b_2\times i^2+\ldots+b_n\times i^n$, where c_k is integer constants and i is loop index, the existing approaches can not work well. To our knowledge, for the more general form $b_0+\sum_{k=1}^{n}\sum_{j=1}^{m} b_{kj}\, i_k^{v_{kj}}$ of array subscripts occurring in some science application, few works have been done to deal with the situation. Thus this paper focuses on this kind of complicated problem. In order to determine the data dependence in array references with nonlinear subscripts coupled integer power variable, we develop a general dependence analysis method, named

*Supported by the 863 High-Tech Project under grant of 2004AA104340, the National Natural Science Foundation of China under grant of 60173026 and SEC E-Institute: Shanghai High Institutions Grid under grant of 200301-1.

PVI-test, based on the integer interval theory. The PVI-test presents a significant improvement over the VI-test through experiment.

The remainder of this paper is structured as follows. Section 2 presents problem description and introduces the integer interval theory. Section 3 proposes theories served the PVI-test. On basis of proposed theories, section 4 develops a novel algorithm for recognizing the data dependence in multidimensional arrays with non-linear subscripts. Then a case study is illustrated in section 5. Finally, Section 6 concludes the paper and pointes out the future work.

2 Preliminaries

2.1 Nested Loop with Nonlinear Subscripts Coupled Integer Power Variables

To describe clearly, we first depict the addressed loop program as shown in Figure1. Without loss of generality, we assume that all bounds of the loop are linear functions of the outer loop indices, and each value of lower and upper bounds is positive integer. The loop step is 1. S_1 and S_2 are statements in the loop body. There maybe exist dependence between the two array references. Their array subscripts, coupled integer power variables, have the following form of polynomial expression:

$$b_0+b_{11}\times i_1^{v_{11}}+b_{12}\times i_1^{v_{12}}+\dots+b_{21}\times i_2^{v_{21}}+b_{22}\times i_2^{v_{22}}+\dots$$

$$+b_{n1}\times i_n^{v_{n1}}+b_{n2}\times i_n^{v_{n2}}+\dots$$

where b_0 is an integer constant, b_{kj} such as b_{11} is integer constant coefficient and i_k is a loop index with power v_{xy}, $k, j, n \in Z^+$.

```
DO $i_1=p_{1,0}$ , $q_{1,0}$
   DO $i_2=p_{2,0}+p_{2,1}\times i_1$ , $q_{2,0}+q_{2,1}\times i_1$
      . . .
         DO $i_n = p_{n,0}+p_{n,1}\times i_1+\dots+p_{n,n-1}\times i_{n-1}$ ,
                  $q_{n,0}+q_{n,1}\times i_1+\dots+q_{n,n-1}\times i_{n-1}$
S1:            $A[f(i_1, i_2, \dots, i_n)]=\dots$
S2:            $\dots= A[g(i_1, i_2, \dots, i_n)]$
         ENDDO
      . . .
   ENDDO
ENDDO
```

Figure 1 A trapezoidal loop program with integer power variable model

Figure 1 carries data dependences if and only if there exist integer solutions for every loop variable i and i′ satisfying equations (1), where

$$f(i_1, i_2, \dots, i_n)=b_0+b_{11}\times i_1^{v_{11}}+\dots+b_{n2}\times i_n^{v_{n2}}+\dots$$

$$g(i'_1, i'_2, \dots, i'_n)=c_0+c_{11}\times i_1'^{v_{11}}+\dots+c_{n2}\times i_n'^{v_{n2}}+\dots$$

and the inequalities (2), where $1 \le k \le n$.

$$f(i_1, i_2, \dots, i_n)=g(i'_1, i'_2, \dots, i'_n) \qquad (1)$$

$$p_{k,0}+\sum_{j=1}^{n-1}p_{k,j}\times i_j \le i_k \le q_{k,0}+\sum_{j=1}^{n-1}q_{k,j}\times i_j \qquad (2)$$

To make it easier for us to solve the problem, we will change the above equation (1) subject to (2) to the following polynomial interval equation (3) subjected to (4).

$$\sum_{i=1}^{2n}\sum_{j=1}^{m}a_{ij}X_i^{\upsilon_{ij}}=[a_0, a_0]$$
(3)

$$P_{2k-1}(x) \le X_{2k-1} \le Q_{2k-1}(x)$$
$$P_{2k}(x) \le X_{2k} \le Q_{2k}(x)$$
$$1 \le k \le n, x=(X_1, X_2, \dots, X_{2n})$$
(4)

where $X_{2k-1}=i_k$ and $X_{2k}=i'_k$, are two instances of the same loop iteration variable contained power υ_{xy}. a_{ij} is integer constant coefficient of $X_i^{\upsilon_{xy}}$, and $a_0=c_0-b_0$.

2.2 Integer Interval Theory

As discussed above, it is rather difficult to solve the equation (1) subjected to the inequalities (2). So we give up finding the exact solutions. But the existence of solutions may be determined by integer interval theory. The followings are some concepts of integer interval theory.

Definition 1 We define the positive and negative part of an integer a, respectively, as

$$a^+=\begin{cases}a & \text{if } a>0\\ 0 & \text{otherwise}\end{cases}$$

$$a^-=\begin{cases}-a & \text{if } a<0\\ 0 & \text{otherwise}\end{cases}$$

Definition 2 Given an integer region $R \in Z^n$ and two functions L and U from R to Z, we define the variable integer interval, denoted [L(x), U(x)], as the union of all integer intervals for all values x_i of x in R:

$$[L(x), U(x)]=\bigcup_{x_i \in R}[L(x_i), U(x_i)]$$

Definition 3 Given an integer region $R \subseteq Z^n$ and three functions F, L, and U from R to Z, we define a variable interval equation of the following form:

$$F(x)=[L(x), U(x)]$$

The above equation is said to be integer solvable in R iff there exist a value of x, x_0□R such that $L(x_0) \le F(x_0) \le U(x_0)$.

3 Theories towards the Polynomial Variable Interval Test

To facilitate dependence test, we first present some "interval solution" rules for polynomial equations based on the integer interval theory.

Theorem 1 *The polynomial interval equation $F(x)+aX^2 = [L(x)+bX^2,U(x)+cX^2]$ is integer solvable subject to a set of constraints on x in R and the constraint $P^2(x)\le X^2\le Q^2(x)$, iff the polynomial interval equation $F(x)=[L(x)+(b-a)X^2, U(x)+ (c-a)X^2]$, is integer solvable subject to the same constraints.*

Proof Let us assume that the polynomial interval equation $F(x)= [L(x)+(b-a)X^2, U(x)+ (c-a) X^2]$ is integer solvable, iff there exist a value of x, x_0 □ R such that $L(x_0)\le F(x_0)\le U(x_0)$, which means

$L(x_0)\le F(x_0)=[L(x_0)+(b-a)X^2,U(x_0)+(c-a)X^2]\le U(x_0) \Leftrightarrow L(x_0)+aX^2\le F(x_0)+aX^2= [L(x_0)+bX^2, U(x_0) + cX^2] \le U(x_0)+aX^2$.

Thus, there exist x_0 such that $L(x_0)+ aX^2 \le F(x_0)+aX^2=[L(x_0)+bX^2,U(x_0)+cX^2]\le U(x_0)+aX^2$.

According to Definition 3 we can prove the polynomial interval equation:

$$F(x)+aX^2=[L(x)+bX^2,U(x)+ cX^2]$$

is integer solvable subject to the same constraints.

Theorem 2 *The variable integer interval $[L+\mu X^2,U+\nu X^2]$, where $P^2\le X^2 \le Q^2$ is equal to the integer interval $[L+\mu^+P^2-\mu^-Q^2,U+\nu^+Q^2-\nu^-P^2]$, iff $\mu\nu\le 0$, or $\mu\nu> 0$ and $U-L+(\nu-\mu)^+P^2-(\nu-\mu)^-Q^2 +1 \ge \min(|\mu|, |\nu|)$.*

Proof The proof of the theorem can be divided into the following two parts:

[part□]: if the condition of $\mu\nu\le 0$, or $\mu\nu> 0$ and $U-L+(\mu-\nu)^+P^2-(\mu-\nu)^-Q^2+1\ge \min(|\mu|, |\nu|)$ is satisfied, the variable integer interval $[L+\mu X^2, U + \nu X^2]$ is equal to the integer interval $[L+\mu^+P^2-\mu^-Q^2 , U+\nu^+Q^2-\nu^-P^2]$.

[part□]: if $[L+\mu X^2,U+\nu X^2]$ is equal to the integer interval$[L+\mu^+P^2-\mu^-Q^2,U+\nu^+Q^2-\nu^-P^2]$, where $P^2\le X_i^2\le Q^2$, we can conclude that the inequality, $U-L+(\mu-\nu)^+P^2-(\mu-\nu)^-Q^2+1\ge \min(|\mu|, |\nu|)$, exist where $\mu\nu > 0$. We will consider two conditions of $\mu\nu > 0$; one is $\mu> 0$ and $\nu> 0$ and another is $\mu <0$ and $\nu<0$.

Proof of part□

Because of $[L(x),U(x)]= \bigcup_{x_i\in R}[L(x_i),U(x_i)]$, where $P^2\le X_i^2\le Q^2$, according to Definition 2, we will prove that $\bigcup_{x_i\in R}[L(x_i),U(x_i)]$ is equal to the integer interval$[L+\mu^+P^2-\mu^-Q^2,U+\nu^+Q^2-\nu^-P^2]$ on the hypothesis of $\mu\nu\le 0$, or $\mu\nu > 0$ and $U-L+(\mu-\nu)^+P^2 -(\mu-\nu)^-Q^2+1\ge\min(|\mu|,|\nu|)$. We will take account of two aspects: (1) $\mu\nu\le 0$ or $\mu\nu > 0$ and (2) $U-L+ (\mu-\nu)^+P^2-(\mu-\nu)^-Q^2 +1\ge\min(|\mu|,|\nu|)$.

(1) $\mu\nu\le 0$.

Let us assume that $\mu\le 0$ and $\nu\ge 0$, then $L+\mu X^2$ is a decreasing function and $U+\nu X^2$ is an increasing function. If there exists a value X_0 such that $L+\mu X_0^2 \le U+\nu X_0^2$, for every X_1, X_2 such that $Q\ge X_2\ge X_1\ge X_0$

$[L+\mu X_1^2,U+\nu X_1^2]\subseteq[L+\mu X_2^2,U+\nu X_2^2]\subseteq \ldots$

$\subseteq[L+\mu Q^2,U+\nu Q^2]$

Thus, the final integer interval $[L+\mu Q^2,U+\nu Q^2]$ contains all of all the previous intervals. It means that $\bigcup_{x_i\in R}[L(x_i),U(x_i)]$ is equal to $[L+\mu^+P^2-\mu^-Q^2,U+\nu^+Q^2-\nu^-P^2]$. If no such X_0 exists, then each integer interval $[L+\mu X_i^2,U+\nu X_i^2]=\phi$, where $P^2\le X_i^2\le Q^2$, and also $[L+\mu^+P^2-\mu^-Q^2,U+\nu^+Q^2-\nu^-P^2]= \phi$.

In both cases the union of all integer intervals $[L+\mu X_i^2,U+\nu X_i^2]$, where $P^2\le X_i^2\le Q^2$, is equal to the integer interval $[L+\mu^+P^2-\mu^-Q^2,U+\nu^+Q^2-\nu^-P^2]$.

For the case of $\mu\ge 0$ and $\nu\le 0$ we can prove similarly.

(2) $\mu\nu\le 0$ and

$U-L+(\nu-\mu)^+P^2-(\nu-\mu)^-Q^2+1\ge\min(|\mu|,|\nu|)$

It is obvious all integer intervals $[L+\mu X_i^2, U + \nu X_i^2]$ are nonempty. Now we should consider two distinct cases: □μ,ν >0 or □μ,ν <0 and two sub cases for each case, where $\mu\ge\nu$ or $\mu<\nu$.

□ **μ >0 and ν >0**

In this case both bounds functions $L+\mu X^2$ and $U+\nu X^2$ are increasing. On the condition of $\mu\ge\nu$, the hypothesis is $U-L+(\nu-\mu)Q^2+1\ge\nu$, the increasing order of their bounds is as following form:

$[L+\mu P^2,U+\nu P^2], \ldots, [L+\mu X_i^2,U+\nu X_i^2], [L+\mu (X_i^2+1), U+\nu (X_i^2+1)], \ldots, [L+\mu Q^2,U+\nu Q^2]$.

For every two consecutive integer intervals $[L+\mu X_i^2,U+\nu X_i^2]$ and $[L+\mu(X_i^2+1),U+\nu(X_i^2+1)]$, where $P^2\le X_i^2\le Q^2-1$, we can derive from the hypothesis:

$U-L+(\nu-\mu)(X_i^2+1)+1\ge\nu \Leftrightarrow$

$L+\mu (X_i^2+1)\le U + \nu X_i^2 + 1$.

Therefore, the union interval of the pair of consecutive interval is $[L+\mu X_i^2,U+\nu(X_i^2+1)]$.

On the condition of $\mu <\nu$, the same conclusion can be obtained.

By induction we can prove that, the integer interval $[L+\mu P^2,U+\nu Q^2]$ is equal to the integer interval $[L+\mu^+P^2-\mu^-Q^2,U+\nu^+Q^2-\nu^-P^2]$, where $P^2\le X_i^2\le Q^2$.

□ **μ<0 and ν<0**

We can draw the same conclusion. Then the part □can be proved.

Proof of part □

According to Definition 2 and hypothesis, $\bigcup_{x_i\in R}[L(x_i),U(x_i)]$ is equal to the integer interval $[L+\mu^+P^2-\mu^-Q^2,U+\nu^+Q^2-\nu^-P^2]$. If $\mu\nu >0$, we should prove that $U-L+(\mu-\nu)^+P^2-(\mu-\nu)^-Q^2+1\ge \min(|\mu|,|\nu|)$, then we should consider two conditions: (1) $\mu >0$ and $\nu >0$ and (2) $\mu<0$ and $\nu<0$

(1) μ >0 and ν >0

In this case both bounds functions $L+\mu X^2$ and $U+\nu X^2$ are increasing, so the increasing order of the bounds of the integer interval $[L+\mu X_i^2,U+\nu X_i^2]$ is as following:

$[L+\mu P^2,U+\nu P^2], \ldots, [L+\mu X_i^2,U+\nu X_i^2]$,

$[L+\mu(X_i^2+1),U+\nu(X_i^2+1)], \ldots ,[L+\mu Q^2,U+\nu Q^2]$

Since $\bigcup_{x_i\in R}[L(x_i),U(x_i)]$ constitutes an integer interval, the union of every two consecutive integer intervals $[L+\mu X_i^2,U+\nu X_i^2]$ and $[L+\mu(X_i^2+1), U+\nu(X_i^2+1)]$, where $P^2\le X_i^2\le Q^2-1$, must also constitute an integer interval. Therefore, the lower bound of the second interval must be less or equal than the upper bound of the first interval plus one:

$L+\mu(X_i^2+1)\le U+\nu X_i^2+1 \Leftrightarrow U-L+(\nu-\mu)X_i^2+1\ge\mu$

Since the above inequality holds for all values of X_i^2 between P^2 and Q^2-1, it must hold for the minimum value of the expression on the left hand side. Therefore:

$U-L+(\nu-\mu)^+P^2-(\nu-\mu)^-(Q^2-1)+1\ge\mu \Leftrightarrow$
$U-L+(\nu-\mu)^+P^2-(\nu-\mu)^-Q^2+1\ge\mu-(\nu-\mu)^-=$
$\min(|\mu|,|\nu|)$.

The conclusion is proved in this case.

(2) $\mu<0$ and $\nu<0$

We can prove this case similarly in the same way. Then the part □ can be proved.

Theorem 3 *Consider the variable integer interval $[L(x)+\mu X^2,U(x)+\nu X^2]$, subject to a set of constraints on x in R and $P^2(x)\le X^2\le Q^2(x)$, where X does not appear in any of the constraints in R.*

If $\mu\nu\le 0$, or $\mu\nu>0$ and $\min(U(x)-L(x)+(\nu-\mu)^+P^2(x)-(\nu-\mu)^-Q^2(x)+1)\ge\min(|\mu|,|\nu|)$, then the above variable integer interval is equal to $[L(x)+\mu^+P^2(x)-\mu^-Q^2(x),U(x)+\nu^+Q^2(x)-\nu^-P^2(x)]$, subject to the same constraints on x in R and the constraint $P^2(x)\le Q^2(x)$.

Proof According to the hypothesis $\mu\nu\le 0$, or $\mu\nu>0$ and $\min(U(x)-L(x)+(\nu-\mu)^+P^2(x)-(\nu-\mu)^-Q^2(x)+1)\ge\min(|\mu|,|\nu|)$. Since the minimum value of the expression in the left hand side is greater or equal than the constant value on the right hand side, we can derive that $\mu\nu\le 0$, or $\mu\nu>0$ and $U(x_i)-L(x_i)+(\nu-\mu)^+P^2(x_i)-(\nu-\mu)^-Q^2(x_i)+1\ge\min(|\mu|,|\nu|)$, for all $x_i\in Z^n$ such that $P^2(x_i)\le Q^2(x_i)$.

According to Theorem 1, each variable integer interval $[L(x_i)+\mu X^2,U(x_i)+\nu X^2]$, where $P^2(x_i)\le X^2\le Q^2(x_i)$, is equal to the integer interval $[L(x_i)+\mu^+P^2(x_i)-\mu^-Q^2(x_i),U(x_i)+\nu^+Q^2(x_i)-\nu^-P^2(x_i)]$. Now according to Definition 2:

$$\bigcup_{P^2(x_i)\le X_j^2\le Q^2(x_i)}[L(x_i)+\mu X_j^2,U(x_i)+\upsilon X_j^2]=$$

$[L(x_i)+\mu^+P^2(x_i)-\mu^-Q^2(x_i),U(x_i)+\nu^+Q^2(x_i)-\nu^-P^2(x_i)]$, for all $x_i\in Z^n$, where $P^2(x_i)\le Q^2(x_i)$.

$$\bigcup_{P^2(x_i)\le Q^2(x_i)}\ \bigcup_{P^2(x_i)\le X_j^2\le Q^2(x_i)}[L(x_i)+\mu X_j^2,U(x_i)+\upsilon X_j^2]$$
$$=\bigcup_{P^2(x_i)\le Q^2(x_i)}[L(x_i)+\mu X_j^2,U(x_i)+\upsilon X_j^2]$$
$$=[L(x_i)+\mu^+P^2(x_i)-\mu^-Q^2(x_i),U(x_i)+\nu^+Q^2(x_i)-\nu^-P^2(x_i)]$$

By Definition 2 the double union of the integer intervals on the left hand side of the above equation is equal to the variable integer interval $[L(x_i)+\mu X^2, U(x_i)+\nu X^2]$, where x_i in Z^n and $P^2(x)\le X^2\le Q(x)^2$. Thus we conclude that the variable integer interval $[L(x_i)+\mu X^2,U(x_i)+\nu X^2]$ is equal to the variable integer interval $[L(x_i)+\mu^+P^2(x_i)-\mu^-Q^2(x_i), U(x_i)+\nu^+Q^2(x_i)-\nu^-P^2(x_i)]$, where $P^2(x)\le Q^2(x)$.

Theorem 4 *Consider the following variable interval equation:*

$F(x)+aX^2=[L(x)+bX^2,U(x)+cX^2]$

subject to a set of constraints on x in Z^n and $P^2(x)\le X^2\le Q^2(x)$, where X^2 does not appear in any of the constraints in R. If $(b-a)(c-a)\le 0$, or $(b-a)(c-a)>0$ and $\min(U(x)-L(x)+(c-b)^+P^2(x)-(c-b)^-Q^2(x)+1)\ge\min(|b-a|,|c-a|)$, then the equation above is integer solvable iff the variable interval equation:

$F(x)=[L(x)+(b-a)^+P^2(x)-(b-a)^-Q^2(x),U(x)+(c-a)^+Q^2(x)-(c-a)^-P^2(x)]$

is integer solvable subject to the same constraints on x in R and the constraint $P^2(x)\le Q^2(x)$.

Proof According to the Theorem 1, the polynomial interval equation

$F(x)+aX^2=[L(x)+bX^{2\ 2},U(x)+cX^2]$

is integer solvable subject to a set of constraints, where $P^2(x)\le X^2\le Q^2(x)$, iff the polynomial interval equation

$F(x)=[L(x)+(b-a)X^2,U(x)+(c-a)X^2]$

is integer solvable subject to the same constraints.

Also it can be concluded from Theorem 2,

$F(x)=[L(x)+(b-a)X^2,U(x)+(c-a)X^2]$

is integer solvable if the variable interval equation

$F(x)=[L(x)+(b-a)^+P^2(x)-(b-a)^-Q^2(x),U(x)+(c-a)^+Q^2(x)-(c-a)^-P^2(x)]$is integer solvable with the constrains that $(b-a)(c-a)\le 0$, or $(b-a)(c-a)>0$ and $\min(U(x)-L(x)+(c-b)^+P^2(x)-(c-b)^-Q^2(x)+1)\ge\min(|b-a|,|c-a|)$ and that concludes our proof.

Theorem 5 *Consider the following variable interval equation:*

$F(x)+aX^n=[L(x)+bX^n,U(x)+cX^n]$

subject to a set of constraints on x in R, $P^n(x)\le X^n\le Q^n(x)$ and $n\in Z^+$, where X^n does not appear in any of the constraints in R. If $(b-a)(c-a)\le 0$, or $(b-a)(c-a)>0$ and $\min(U(x)-L(x)+(c-b)^+P^n(x)-(c-b)^-Q^n(x)+1)\ge\min(|b-a|, |c-a|)$, then the equation above is integer solvable iff the variable interval equation:

$F(x)=[L(x)+(b-a)^+P^n(x)-(b-a)^-Q^n(x),U(x)+(c-a)^+Q^n(x)-(c-a)^-P^n(x)]$

is integer solvable subject to the same constraints on x in R and the constraint $P^n(x)\le Q^n(x)$.

Proof The proof is the same as Theorem 4.

Especially, when n=1, theorem 5 just is reduced to theorem 5 in the VI-test [13]. Also obviously theorem 2 is the instance of theorem 5. So theorem 5 is a general rule to solve the problem of array references either with linear subscripts or non-linear.

4 The Algorithm for the PVI-test

4.1 Basic idea

Our algorithm consists of three steps: the first is to rewrite the Diophantine equations (1) into the form of polynomial interval equation (3) subjected to inequalities (4). The second is to repeatedly eliminate variables which do not appear in any constraints in the equation, in order to decrease the constraints. The last is to judge whether dependence exists. During the step 2, according to Theorem 5, we treat the X^i as a variable with its own constrain like a usual variable X. At the end of the step 2, an integer interval equation with zero on the left-hand side and an integer interval on the right-hand side, is obtained. If zero belongs to the integer interval on the right-hand side, there exists integer solutions to the equation (1), that is to say that dependence exists in the loop.

Furthermore, before eliminating each variable from the polynomial interval equation, we should check following two accuracy conditions [13].

Accuracy Condition 1 For every variable X, eliminated from the variable interval equation:

$$F(x)+aX^n=[L(x)+bX^n,U(x)+cX^n], \qquad n\in Z$$

subject to a set of constraints on x in R and $P^n(x)\le X^n\le Q^n(x)$, where X does not appear in any of the constraints in R, the following inequalities need to be satisfied:

$(b-a)(c-a)\le 0$, or

$(b-a)(c-a)>0$, $\min(U(x)-L(x)+(c-b)^+P(x)^n-(c-b)^-Q(x)^n+1)\ge\min(|b-a|,|c-a|)$.

Accuracy Condition 2 For every variable X, eliminated from the variable interval equation,

$$F(x)+aX^n=[L(x)+bX^n,U(x)+cX^n],$$

subject to a set of constraints on x in R and the constraint $P^n(x)\le X^n\le Q^n(x)$, where X does not appear in any of the constraints in R, the following inequality needs to be satisfied: $\min(Q^n(x)-P^n(x))\ge 0$, $n\in Z$.

4.2 The algorithm

According to the theories and analysis about the PVI-test, we propose an algorithm for our dependence test towards the loop shown in Figure 1. The variables, which can be eliminated, are found by the procedure called Detector and pushed into a stack. The procedure acc_condition is used to judge whether variable satisfies above two accurate conditions, or adjust the lower bound to satisfy conditions and turn back the revised value of lower bound. The procedure Pos and Neg return the positive and negative part of an integer x, respectively.

Algorithm: PVI-test for data dependence

Input: $(a_0,a_{11},\ldots,a_{1m},a_{21},\ldots,a_{2m},\quad a_{n1},\ldots,a_{nm},l^{(11)},\ldots,l^{(1m)},\ldots,l^{(2n1)},\ldots,l^{(2nm)},u^{(11)},\ldots,u^{(1m)},\ldots,u^{(2n1)},\ldots,u^{(2nm)})$;

/*a_{ij} is the integer constant coefficient of the X_i^{υ}, $l^{(ij)}$ and $u^{(ij)}$ are the lower bound and upper bound of X_i^{υ}, respectively, as described above*/

Output: true means that the equation (1) is integer solvable and false means that the equation (1) is not integer solvable.

```
Procedure PVI (input, output)
{   /* Initialization */
    $A\leftarrow\{a_{11},\ldots,a_{1m},a_{21},\ldots,a_{2m},\ldots,a_{2n1},\ldots,a_{2nm}\}$;
    $B\leftarrow\{l^{(1m)},\ldots,l^{(2n1)},\ldots,l^{(2nm)},u^{(11)},\ldots,u^{(1m)},\ldots,u^{(2n1)},\ldots,u^{(2nm)}\}$;
    $X\leftarrow\{X^{(1m)},\ldots,X^{(2n1)},\ldots,X^{(2nm)},X^{(11)},\ldots,X^{(1m)},\ldots,X^{(2n1)},\ldots,X^{(2nm)}\}$;
    $F\leftarrow\{a_{11}X_1,\ldots,a_{12}X_1^2+\ldots+a_{1m}X_1^m+,\ldots,a_{2n1}X_{2n}+a_{2n2}X_{2n}^2+\ldots+a_{2nm}X_{2n}^m\}$;
    $L\leftarrow a_0$;   $U\leftarrow a_0$;
    While ($B\neq\varnothing$)
    {   Detector(X, F,stack);
        While (stack$\neq\varnothing$)
        {   elim $\leftarrow$stack;
            a$\leftarrow$the coefficient of elim in the function F;
            b$\leftarrow l^{(i)}$;
            c$\leftarrow u^{(i)}$;
            flag1$\leftarrow$accurate_condition (L, U, $l^{(i)}$, $u^{(i)}$, a, b, c );
            if flag1 then
            {   $L\leftarrow L + l^{(i)}\times Pos(b-a) - u^{(i)}\times Neg(b-a)$;
                $U\leftarrow U + l^{(i)}\times Pos(c-a) - u^{(i)}\times Neg(c-a)$;
                $F\leftarrow F - a_i\times elim$;
                $X\leftarrow X\backslash X^{(i)}$;
            }
            if ($0\ge L$ & $U\ge 0$ & flag1) then return true;
            else return false;
        }
    }
}
Procedure Pos(x)
{   if x>0 then return x   else return 0;}
Procedure Neg(x)
{   if x<0 then return –x   else return 0;}
Procedure Detector(X, F,stack)
{   for $\forall x_i^j\in X$
        /* X is the union of variables in the polynomial interval equation */
        if ($X_i^j$ not exist in B) & ($X_i^1$ not exist in B) then stack$\leftarrow X_i^j$;
}
Procedure acc_condition (L, U, l′, u′,a, b, c)
{   flag2$\leftarrow$false;
    if $(b-a)(c-a)\le 0$ then flag2$\leftarrow$true;
    else
```

```
            if  min  (U-L+ l′×Pos(c-b)*−u′×Neg(c-b)
              +1)≥ min(|b-a|,|c-a|) then
                flag2←true;
            else
                while not flag2
                { l′=l′−1;
                   acc_condition (L,U, l′−1,u′,a, b, c);
                }
        if flag2 then
            if min(u′- l′)≥0 then return true;
             else return false;
}
```

5 Case Study

To verify the proposed PVI-test in previous session, we demonstrate how to handle loops with subscripts coupled integer power index variable and detect the dependence based on interval solution rules.

```
    DO i = 1 , 5
          DO j = 5 − i , 2 × i + 7
    S:          A [i+i^3+j^2]=A[j^2−2×i+10]+…
          ENDDO
    ENDDO
```

Figure 2 An example loop program with integer variable interval

There is data dependence between the two array references in statement S in Figure2, if we can conclude that the right of the final interval equation includes zero through the proposed algorithm. We give the polynomial interval equation (5) subjected to inequalities (6) as follows:

$X_1+X_1^3+2X_2+X_3^2-X_4^2=[10,10]$ (5)

$1\le X_1\le 5$ $1\le X_2\le 5$

$5-X_1\le X_3\le 2X_1+7$

$(5-X_1)^2\le X_3^2\le (2X_1+7)^2$

$5-X_2\le X_4\le 2X_2+7$

$(5-X_2)^2\le X_4^2\le (2X_2+7)^2$ (6)

where X_1 and X_2 are instances of the loop variable i and X_3, X_4 are instances of the loop variable j.

The variables that may be eliminated, according to our algorithm, are X_3^2 and X_4^2. We may start with X_4^2. In this case a=−1, b=0, c=0, L(x)=10, U(x)=10, P(x)= $(5-X_2)^2$, Q(x)=$(2X_2+7)^2$. In addition, before eliminating, we should check two accuracy conditions as shown in the algorithm.

Checking Accuracy Condition 1

$(0-(-1))(0-(-1))=1>0$, and

$\min(10-10+0(5-X_2)^2-0(2X_2+7)^2+1)=1\ge 1=\min(|0-(-1)|, |0-(-1)|)$.

Checking Accuracy Condition 2

$\min((2X_2+7)^2-(5-X_2)^2)=\min((2X_2+7+5-X_2)(2X_2+7-5+X_2))\ge 0$.

Since both accuracy conditions are satisfied we can indeed eliminate X_4 and the polynomial interval equation is transformed into:

$X_1+X_1^3+2X_2+X_3^2=[10+(0-(-1))(5-X_2)^2,10+(0-(-1))(2X_2+7)^2]\Leftrightarrow X_1+X_1^2+2X_2+X_3^2=[X_2^2-10X_2+35,4X_2^2-28X_2+59]=[10+(5-X_2)^2,10+(2X_2+7)^2]$.

We continue by eliminating variable X_3^2. In this case a=1, b=0, c=0, L(x)=$X_2^2-10X_2+35$, U(x)=$4X_2^2-28X_2+59$, P(x)=$(5-X_1)^2$, Q(x)=$(2X_1+7)^2$. And check two accuracy conditions.

Checking Accuracy Condition 1

$(0-1))(0-1))=1>0$, and

$\text{Min}(4X_2^2-28X_2+59-(X_2^2-10X_2+35)+0(5-X_2)^2-0(2X_2+7)^2+1)=\min(10+(2X_2+7)^2-(10+(5-X_2)^2)+1)\ge 1=\min(|0-(-1)|,|0-(-1)|)$.

Checking Accuracy Condition 2

$\min((2X_1+7)^2-(5-X_1)^2)\ge 0$.

Since both accuracy conditions are satisfied we can indeed eliminate X_3^2 and the polynomial interval equation is transformed into:

$X_1+X_1^3+2X_2=$

$[10+(5-X_2)^2-(2X_1+7)^2, 10+(2X_2+7)^2-(5-X_1)^2]\Leftrightarrow$

$X_1+X_1^3+2X_2=[X_2^2-10X_2-4X_1^2-28X_1-14, 4X_2^2+28X_2-X_1^2+18X_1+34]$.

Next the variable X_2^2 can be eliminated. In this case a=0, b=1, c=4, L(x)= $-10X_2-4X_1^2-28X_1+24$, U(x)=$28X_2-X_1^2+18X_1-14$, P(x) =1, Q(x)=25.

Checking Accuracy Condition 1

$((1-0))(4-0))>0$

$\min(28X_2-X_1^2+18X_1+34-(-10X_2-4X_1^2-28X_1-144)+1)\ge 1=\min(|1-0|, |4-0|)$.

Checking Accuracy Condition 2

$\min(25-1)=24\ge 0$.

Since both accuracy conditions are satisfied we can indeed eliminate X_2 and the polynomial interval equation is transformed into:

$X_1+X_1^3+2X_2=[-10X_2-4X_1^2-28X_1-14+(1-0)\times 1,28X_2-X_1^2+18X_1+34+(4-0)\times 25]\Leftrightarrow$

$X_1+X_1^3+2X_2=[-10X_2-4X_1^2-28X_1-13,28X_2-X_1^2+18X_1+134]$

And we can eliminate X_2 in the same way, and get the polynomial interval equation:

$X_1+X_1^3=[-4X_1^2-28X_1-73,-X_1^2+18X_1+264]$

We continue by eliminating variable X_1^3. In this case a=1, b=0, c=0, L(x)=$-4X_1^2-28X_1-73$, U(x) =$-X_1^2+18X_1+234$,P(x)=1, Q(x)=5^3. And check the two accuracy conditions.

Checking Accuracy Condition 1

$((0-1))(0-1))>0$

$\min(-X_1^2+18X_1+264-(-4X_1^2-28X_1-73)+1)\ge 1=\min(|0-1|,|0-1|)$.

Checking Accuracy Condition 2

$\min((-X_1^2+18X_1+264)-(-4X_1^2-28X_1-73))\ge 0$.

Since both accuracy conditions are satisfied we can indeed eliminate X_2 and the polynomial interval equation is transformed into:

$X_1=[-4X_1^2-28X_1-198,28X_2-X_1^2+18X_1+263]$

We continue by eliminating variable X_1^2, X_1 following the algorithm, and obtain the final interval equation as follows:

$0=[-433,317]$

The PVI-Test concludes that there indeed exists an integer solution to the polynomial interval equation subject to the constraints.

Conclusion

Though many well-known research works have been done on the dependence analysis such as the Banerjee test, the I-test, the Omega test and the Power test, they tend to ignore the case of arrays with non-linear subscripts. But loops with non-linear subscripts are sometimes found in scientific source code, especially with polynomial subscripts coupled with integer power variables.

We have developed a novel dependence analysis scheme, the PVI-test, which can accurately handle loops with subscripts couple polynomial expressions and trapezoidal bounds. The test is based on "interval solution" theories for polynomial equation and solves the problem by repeatedly eliminating variables from the polynomial interval equation. In addition we illustrate how to use it to handle data dependence problems in loops and testify its accuracy. Compared to the others tests, the PVI-test is much more general and efficient to detect dependence in arrays either with linear subscripts or non-linear. Furthermore, the PVI-test extends the VI test which fails to solve problem with non-linear subscripts, as well as inherits all the benefits of the VI test. Our future work is to add the PVI-test into our parallel compiler.

References

[1] Z. Shen Z. Shen, Z. Li and P.C. Yew, An Empirical Study on Array Subscripts and Data Dependencies, Proceedings of 1989 International Conference on Parallel Processing, St. Charles, IL, pages 145-152, Aug. 1989.

[2] M.J. Wolfe, High Performance Compilers for Parallel Computing, Addison-Wesley Publishing, New York, NY, U.S.A.

[3] U. Banerjee, R. Eigenmann, A. Nicolau, and D. A. Padua, Automatic program parallelization, Proc. IEEE, 8(12): 211-243, 1993.

[4] W. Pough, A Practical Algorithm for Exact Array Dependence Analysis, Communications of the ACM, 35(8): 102-114, 1992.

[5] G. Dantzing, B. Eaves, Fourier-Motzkin Elimination and its Dual, Journal of Combinatorial Theory (A), 14(2): 288-297, 1973.

[6] M. Wolfe, C.W. Tseng, The power test for data dependence, IEEE Trans. Parallel Distributed Syst, 3(5): 591-601, 1992.

[7] X. Kong, D. Klappholz, K. Psarriss, The I test: an improved dependence test for automatic parallelization and vectorization, IEEE Trans.Parallel Distributed Syst., 2(3): 342-349, 1991.

[8] K.Psarris, S.Pande, An empirical study of the I test for exact data dependence, Prodeeding of the 1994 International conference on Parallel Processing, August 1994.

[9] K. Psarris, X.Y. Kong, D. Klappholz, The Direction Vector I Test, IEEE Trans. Parallel Distributed Syst, 4(11): 1280-1290, 1993.

[10] W.L. Chang, C.P. Chu, The generalized direction vector I test, Parallel Computing, 27(11): 1117-1144, 2001.

[11] T.C. Huang, C.M. Yang, Data dependence analysis for array references, The Journal of Systems and Software, 52(1): 55-65, 2000.

[12] T.C. Huang, C.M. Yang, Dependence analysis with direction vector for array references, Computer and Electrical Engineering, 27(3): 375-393, 2001.

[13] K. Kyriakopoulos, K. Psarris, Data Dependence Analysis techniques for increased accuracy and extracted parallelism, International Journal of parallel programming, 32(4): 317-359, 2004.

[14] Go. G., K. Kennedy, C.W. Tseng, Practical dependence testing, Proceedings of the ACM SIGPLAN '91 Conference on Programming Language Design and Implementation, Toronto, Ont. Canada, pages 15-29, 1991.

[15] T.C. Huang, C.M. Yang, An exact data dependence analysis for array reference: the IR test, The Tenth Annual International Conference on High Performance Computers, Ottawa, Canada, pages 1-24, 1996.

[16] Z. Li, P.C. Yew, C.Q. Zhu, Data dependence analysis on multi-dimensional array reference, International Conference on Supercomputing, pages 86-95, 1989.

[17] Z. Li, P.C. Yew, C.Q. Zhu, An efficient data dependence analysis for parallelizing compilers, IEEE Trans. Parallel Distributed Syst., 1(1): 26-34, 1990.

[18] D.E. Maydan, J.L. Hennessy, M.S. Lam, Efficient and extract data dependence analysis, Proceedings of the ACM SIGPLAN '91 Conference on Programming, Language Design and Implementation, Toronto, Ont. Canada, pages 1-14, 1991.

[19] P.M. Petersen, D.A. Padua, Static and dynamic evaluation of data dependence analysis techniques, IEEE Trans. Parallel Distributed Syst, 7(11): 1121-1132, 1996.

[20] W.Paugh, A Practical algorithm for extract array depend-

ence analysis, Commun. ACM, 35(8): 102-114, 1992.

[21] J.Subhlok, K.Kennedy, Integer programming for array subscript analysis, IEEE Trans. Parallel Distributed Syst, 6(6): 662-668, 1995.

Jing Zhou received her B.S and M.S degrees in computer science from Jiangxi Normal University in 2000 and 2003 respectively, and she is now a doctoral candidate in computer software and theory at Tongji University. Her main research interests include grid computing and high performance computing technology.

Guo-Sun Zeng achieved his Ph.D. degree from Shanghai Jiaotong University in 2000. He is concurrently a professor and Ph.D. supervisor of Tongji University. He is engaged in high performance computing technology.