The Quadratic Test: An Exact Data Dependence Test for Quadratic Expressions

Jia-Hwa Wu and Chih-Ping Chu

Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan, Taiwan 701, R.O.C.
Phone Number: +886-6-2757575 Ext. 62527
E-mail: {wujh,chucp}@csie.ncku.edu.tw
Fax Number: +886–6–2747076

Abstract

In this paper, we present an exact dependence test, called the Quadratic Test, that can handle array subscripts with quadratic expression exactly. This method can find all of the integer solutions. Using this information, it is helpful for further program parallelization.

Index Terms – Data-dependence test, Parallel compiler.

I. Introduction

It is well known that the problem of finding integer-valued solutions to a system of linear or quadratic equations is NP-hard [2]. Therefore, in practice, most famous data dependence analysis algorithms are used to solve as many particular cases of this problem as possible. There are several well-known methods of data dependence analysis for dealing with one-dimensional arrays under constant bounds or variable bounds: the GCD test and Banerjee’s method [1, 2], the I test and the direction vector I test [7,9], and the interval reduction test [6]. There are also several famous methods of data dependence analysis to testing multi-dimensional coupled arrays under constant bounds or variable bounds: the generalized GCD test [1,2], the Lambda test [8], the Power test [12], and the Omega test [11]. There are several well-known data dependence analysis algorithms applicable for arrays with linear subscripts with symbolic coefficients or with non-linear subscripts under symbolic bounds: the infinity Banerjee test [2,10], and the Range test [4].

In this paper, we will present the Quadratic Test, a dependence test that can handle quadratic expression array subscripts. This method first checks whether quadratic equation is monotonically increasing or decreasing, then reduces the integer solution interval of each variable by repeated projection. Once the effective integer solution interval of any dependence function shrinks to empty, this dependence constraint system has no integer solution. Otherwise, all of the integer solutions can be found.

The rest of this paper is organized as follows. In Section II, the problem of data dependence for arrays with quadratic expression array subscripts is reviewed. Section III gives an overview of the range propagation to determine whether quadratic equation is monotone function. In Section IV, the Quadratic test is presented. An example that can be parallelized by the Quadratic test but cannot be handled effectively by current parallelization schemes is given in Section V. In Section VI, experimental results are presented to demonstrate the effectiveness of the Quadratic test. Finally, a conclusion is given in Section VII.

II. Background

Consider the do-loop in Figure 1. The array A in statement S₁ contains one quadratic reference and the array A in statement S₂ includes one linear reference. The lower and upper bounds of the loop are, respectively, 1 and 5. Therefore, the bounds of the do-loop are constants. This do-loop executes five iterations by consecutively assigning the values 1, 2, ..., 5 to I and in each iteration the body (the main statements S₁ and S₂) is executed exactly once.

```
DO I=1, 5
   ...
S₁:   A(I²) = ...
   ...
S₂:   ... = A(3I+2)
   ...
ENDDO
```

Figure 1: a loop with quadratic referenced array and constant bound.
To ascertain whether two references to the one-dimensional array $A$ may refer to the same element of $A$, we have to check if the following quadratic equation

$$x^2 - 3y - 2 = 0$$

that has a simultaneous integer solution of $x$ and $y$ under the constant bounds $1 \leq x, y \leq 5$.

The Range Test [4] is mainly used to check whether there is dependence relation for nonlinear expressions. The Range Test works as follows. For a given iteration $i$ of a loop $L$, the accessed array subscript range, range($i$), is considered as a symbolic expression; if this range does not overlap with the range accessed in the next iteration, $i + 1$, then there is no cross-iteration dependence for $L$; the two ranges do not overlap if $\max(\operatorname{range}(i)) < \min(\operatorname{range}(i+1))$. The Range Test disproves carried dependences between $A(f(i))$ and $A(g(j))$ for a loop $L$, by proving that the range of elements taken by $f$ and $g$ do not overlap.

### III. The Range Propagation Techniques

The Quadratic Test, to be introduced in Section IV, is correct only if dependence function is with monotonically increasing or decreasing subscripts. Determining monotony is simple for linear subscripts. However, it can also be done for many nonlinear expressions by testing whether the difference between two consecutive values is always positive or always negative. Before we describe the Quadratic test, we must define the property of monotone for a particular loop index.

**Definition 1**: A function $f(i_1, i_2, \ldots, i_p)$ is monotonically increasing for index $i_k$ (with lower bound $l_k$ and upper bound $u_k$) if and only if

$$f(i_1, i_2, \ldots, a_k, \ldots, i_p) < f(i_1, i_2, \ldots, b_k, \ldots, i_p),$$

where $l_k \leq a_k < b_k \leq u_k$ and $1 \leq k \leq p$. Similarly, a function $f(i_1, i_2, \ldots, i_p)$ is monotonically decreasing for index $i_k$ if and only if

$$f(i_1, i_2, \ldots, a_k, \ldots, i_p) > f(i_1, i_2, \ldots, b_k, \ldots, i_p),$$

where $l_k \leq a_k < b_k \leq u_k$ and $1 \leq k \leq p$.

We can prove whether a function is monotonically increasing/decreasing for a loop level $k$ by proving that the difference $f(i_1, i_2, \ldots, i_k+1, \ldots, i_p) - f(i_1, i_2, \ldots, i_k, \ldots, i_p)$ is always greater/less than zero using the range propagation techniques [5].

Now, we will describe how the information collected by the range propagation algorithm can be used to determine if a function is with monotonically increasing or decreasing subscripts. We calculate the integer range for the difference $f(i_1, i_2, \ldots, i_k+1, \ldots, i_p) - f(i_1, i_2, \ldots, i_k, \ldots, i_p)$, then determine whether this range is always positive or always negative. This integer range is calculated by repeatedly substituting ranges for variables in the difference function then simplifying the function, until all variables are eliminated.

For example, suppose we wish to determine that $y=x^2-x+10$ is with monotonically increasing or decreasing subscripts for loop level $x$, where the bounds of $x$ are $[1: 9]$. First, we calculate the difference, which is $((x+1)^2-(x+1)+10) - (x^2-x+10)$. After we simplify this difference function, we get the difference function $2x$. Then, we substitute $[1: 9]$ for $x$ in $2x$, getting $2*[1: 9]$. After simplification, we get the range $[2: 18]$. In this range $x$ is always greater than zero. Using the Definition 1, we can determine $y$ is monotonically increasing in $1\leq x \leq 9$.

### IV. The Quadratic Test

In the dependence problem for real programs, it is observed that most dependence equations are with no more than two variables [2]. Thus, in this section, we will focus on two-variable quadratic equation problem. The method is easily extended to dependence equation with more than two variables. We begin the discussion of the Quadratic test by presenting some notations.

**Definition 2**: Let $a$ be an integer. Define

$$a^+ = \max \{a, 0\},$$

$$a^- = \max \{-a, 0\}$$

**Lemma 1 (Zima and Chapman, 1991)**: Let $L$, $U$ be positive integers such that $L \leq U$, then

1. $\min \{ax \mid L \leq x \leq U\}=a(L-a)U$
2. $\max \{ax \mid L \leq x \leq U\}=a(U-a)L$

**Lemma 2**: Let $P_1([x_1], y_1)$ and $P_2([x_1], y_2)$ be two points on the line $ax+by=c$ or monotony quadratic function $ax^2+by=c$, where $x_1$ is an arbitrary real number, $[x_1]$ denotes the largest integer smaller than or equal to $x_1$, and $\lceil x_1 \rceil$ denotes the smallest integer larger than or equal to $x_1$. Then, the line segment $P_1P_2$ contains no integer point if and only if $y_1$ and $y_2$ are non-integers.

**Proof**: As shown in Fig. 2(a), since $[x_1]$ and $\lceil x_1 \rceil$ are the two integers most close to $x_1$, it is very trivial that the only integer points must appear at $P_1$ and $P_2$. If $y_1$ and $y_2$ are integers, then $x=-(a(x_1+c)b)/ab$ and $y=(a(x_1+c)b)/b$. In other words, $y_1$ and $y_2$ must be integers. Similar, the function $ax^2+by$ is monotonically increasing or decreasing (Fig. 2(b)). Results can also be inferred for. This concludes the proof.
Lemma 3: Let L, U be positive integers, then
\begin{align*}
(1) \quad & \min \{ax^2 \mid L \leq x \leq U\} = aL^2 - aL^2 U^2 \\
(2) \quad & \max \{ax^2 \mid L \leq x \leq U\} = aU^2 - aL^2 U^2.
\end{align*}

Proof: To Lemma 1, we replace x with x², then we get lemma 3.

Lemma 4: Let L, U be positive integers, then
\begin{align*}
(1) \quad & \min \{ax^2 + by \mid L \leq x \leq U \text{ and } L \leq y \leq U\} \\
&= (a^2 L^2 - a^2 L^2 U^2) + (bL^2 - bL^2 U) \\
(2) \quad & \max \{ax^2 + by \mid L \leq x \leq U \text{ and } L \leq y \leq U\} \\
&= (a^2 U^2 - a^2 L^2 U^2) + (bU^2 - bU^2 L).
\end{align*}

Proof: By Lemmas 1 and 3, we can, respectively, obtain the following inequalities:
\begin{itemize}
  \item \( bL - bU \leq y \leq b^2U - bL \)
  \item \( aL^2 - aU^2 \leq ax^2 \leq a^2U^2 - aL^2 L^2 \).
\end{itemize}

According to (4-1), we have
\[ (a^2 L^2 - a^2 L^2 U^2) + (bL^2 - bL^2 U) \leq ax^2 + by \leq (a^2 U^2 - a^2 L^2 U^2) + (bU^2 - bU^2 L). \]

This completes the proof of this lemma.

In this section, we focus on a two-variable dependence equation of this form:
\[ ax^2 + by = c \quad \text{(Eq-1)} \]
subject to
\[ l_1 \leq x \leq u_1 \quad \text{and} \quad l_2 \leq y \leq u_2 \quad \text{(Eq-2)} \]
where a, b, c are integer constants and x, y are integer variables. Apparently, the integer solutions will be located within the rectangular region bounded by \( l_1 \leq x \leq u_1 \) and \( l_2 \leq y \leq u_2 \). To find these integer solutions, if we project this line onto axis x, then the effective solution interval of x will be the intersection of the projection interval of this line restricted by \( l_2 \leq y \leq u_2 \) on axis y and the original constraint interval \( l_1 \leq x \leq u_1 \), that is, the physical solution interval of x will be only a subset of its original constraint interval.

The projection interval of line \( ax^2 + by = c \) restricted by \( l_2 \leq y \leq u_2 \) on axis x is
\[ x^2 = \frac{-b}{a} y + \frac{c}{a}, \text{ where } l_2 \leq y \leq u_2. \]

Lemma 1 enables us to compute the bounds of x:
\[
\left[ \frac{-b}{a} l_2^+ - \frac{c}{a}, \frac{-b}{a} l_2^- + \frac{c}{a} \right] \cap \left[ \frac{-b}{a} u_2^+ - \frac{c}{a}, \frac{-b}{a} u_2^- + \frac{c}{a} \right]
\]

Let
\[
u_1^{(i)} = \left[ \frac{-b}{a} l_2^+ - \frac{c}{a}, \frac{-b}{a} l_2^- + \frac{c}{a} \right]
\]
denote x's lower bound and upper bound respectively. Since, according to the original constraint that \( x \) must lie in between \( l_1 \) and \( u_1 \), the solution interval of x becomes
\[ [l_1^{(i)} : u_1^{(i)}] \cap [l_1 : u_1] = [l_1^{(i)} : u_1^{(i)}] \]

Because \( l_1^{(i)} \) and \( u_1^{(i)} \) may not be integers, by lemma 2, if equation (Eq-1) contains integer solutions, they must exist between the closed interval \([l_1^{(i)}] \cap [u_1^{(i)}] \). If now \([l_1^{(i)}] > [u_1^{(i)}] \), then the closed interval is empty and equation (Eq-1) subject to the constraints of (Eq-2) has no integer solution. Otherwise, since the solution interval of x has been reduced (see the illustration stated below), the solution interval of y will change correspondingly.

With the same argument as above, we project...
this line onto axis y. Let the reduced solution interval of y be \([l_2^{(i)}, u_2^{(i)}]\). If now \([l_2^{(i)}, u_2^{(i)}]\) then equation (Eq-1) subject to the constraints of (Eq-2) has no integer solution. Otherwise, we project this line onto axis x again to obtain the reduced solution interval of x. Repeating this procedure \(m\) times we have

\[
\begin{align*}
  x \in &\left[l_1^{(m)} : u_1^{(m)}\right] \\
y \in &\left[l_2^{(m)} : u_2^{(m)}\right]
\end{align*}
\]

Repeating \((m+1)\) times we have

\[
\begin{align*}
  x \in &\left[l_1^{(m+1)} : u_1^{(m+1)}\right] \\
y \in &\left[l_2^{(m+1)} : u_2^{(m+1)}\right]
\end{align*}
\]

In the process of finding integer solution of equation (Eq-1) subject to the constraints of (Eq-2), we use the following method.

**Method:** Given a two-variable dependence equation as in (Eq-1) subject to the constraints in (Eq-2), we project this equation onto axis x and axis y in turn, to obtain the reduced solution intervals of x and y respectively. Repeating this procedure \(m\) times we get

\[
\begin{align*}
  x \in &\left[l_1^{(m)} : u_1^{(m)}\right] \\
y \in &\left[l_2^{(m)} : u_2^{(m)}\right]
\end{align*}
\]

Repeating \((m+1)\) times we have

\[
\begin{align*}
  x \in &\left[l_1^{(m+1)} : u_1^{(m+1)}\right] \\
y \in &\left[l_2^{(m+1)} : u_2^{(m+1)}\right]
\end{align*}
\]

(1) If the solution interval of x or y has been reduced to empty, i.e., \([l_1^{(m)} : u_1^{(m)}]\) or \([l_2^{(m)} : u_2^{(m)}]\) equation (Eq-1) subject to the constraints in (Eq-2) has no integer solution.

(2) If the solution intervals of x and y are unable to be reduced, i.e., \([l_1^{(m)} : u_1^{(m)}]\) and \([l_2^{(m)} : u_2^{(m)}]\) then equation (Eq-1) subject to the constraints of (Eq-2) contains at least one integer solution:

(a) If both solution intervals have shrunk to one point respectively, i.e., \([l_1^{(m)} : u_1^{(m)}]=l_1^{(m)}\) and \([l_2^{(m)} : u_2^{(m)}]=l_2^{(m)}\) then equation (Eq-1) subject to the constraints of (Eq-2) has exactly one integer solution, which is \((x, y)=(l_1^{(m)}, l_2^{(m)})\).

(b) Otherwise, equation (Eq-1) subject to the constraints of (Eq-2) contains at least two integer solutions. These two integer solutions are

\[
\begin{align*}
  &\{l_1^{(m)} : [u_1^{(m)}, l_2^{(m)}]\}, \\
  &\{u_1^{(m)} : [l_2^{(m)}, u_2^{(m)}]\}
\end{align*}
\]

3) To find other integer solutions in case (2b), we reduce the solution intervals of x and y, respectively, by adding one to their lower bounds and subtracting one from their upper bounds. Use them as new constraint intervals and repeat the above procedure until one of them becomes empty. Eventually, all of the integer solutions can be evaluated.

This method is easily extended to dependence equation with more than two variables.

**V. An Example**

Let’s use an example to show that the Quadratic Test is more accurate than the Range Test.

Consider the loop in Figure 1, the dependence equation between \(S_1\) and \(S_2\) is \(y=(x^2-2)/3\). For starters, we prove whether \((x^2-2)/3\) is monotonically increasing or decreasing for loop \(I\), using the techniques described in Section III. We must compare \((x^2+1)^2/3\) with \((x^2-2)/3\), where \(x=1:5\), to see if the difference is always greater than zero or less than zero. First, we calculate the difference, which is \((x^2+1)/3\). Then, we substitute \([1:5]\) for \(x\) in \((2x^2+1)/3\), getting \([1:11/3]\). From this range, we can see that \((2x^2+1)/3=[1:11/3]\), is always greater than zero. Function \((x^2-2)/3\) is monotonically increasing.

Now, we project this equation onto axis x, we have

\[x^2-3y+2\text{ subject to }1\leq x\leq 5\text{ and }1\leq y\leq 5\]

By Lemma 1, the lower bound and the upper bound of x are respectively:

\[
\begin{align*}
  l_1^{(i)} &= \sqrt{5} \\
u_1^{(i)} &= \sqrt{17}
\end{align*}
\]

The effective solution interval of x becomes \([\sqrt{5} : \sqrt{17}]\cap[1 : 5]=[\sqrt{5} : \sqrt{17}]\].

Hence

\[
x \in [\sqrt{5} : \sqrt{17}]
\]

Since this interval is not empty, we project \(y=(x^2-2)/3\) with \(3\leq x\leq 5\) onto axis y next. The effective solution interval of y becomes \([3 : 4]\). Since \(l_1^{(2)} \leq u_2^{(2)}\), we project \(x^2-3y+2\) onto axis x next. The effective solution interval of x becomes \([\sqrt{11} : \sqrt{14}]\cap[3 : 4]= [\sqrt{11} : \sqrt{14}]\].

Hence

\[
x \in [\sqrt{11} : \sqrt{14}]
\]

Now, since \([\sqrt{11} : \sqrt{14}]\) does not contain any integer. This means that there is no data dependence between \(S_1\) and \(S_2\).

For the dependence equation in the above, if we apply the Range Test, we get \(1\leq t\leq 25\), and \(5\leq 31+2\leq 17\). Because the range of elements taken by \(f(x)=t^2\) and \(g(x)=31+2\) has overlap, it will conclude that there may exist a solution. This means that there may exist data dependence. In this example, it is obvious that the Quadratic Test is more accurate than the Range Test.
VI. Experimental Results

To evaluate the performance of the Quadratic test, we implemented it in C language, and run the programs on Personal Computer equipped with Intel Pentium CPU. The benchmarks are cited from five numerical packages EISPACK, LINPACK, Parallel loops, Livermore loops, and Vector loops [3]. In this experiment, we examine totally 169 subroutines in which 303 pairs of one-dimensional array references with constant bounds and quadratic equation are involved. It is indicated from Table 1 that the independent quadratic equations are 156 pairs, dependent quadratic equations are 147 pairs. The results obtained (Table 1) reveal the Quadratic test is an exact test. From Table 1, there are 49% cases where their solutions were exactly found and the parallelism for these cases could be improved. The “accuracy rate” in Table 1 refers to, when given a set of one-dimensional arrays with constant bounds and quadratic equation, how often the Quadratic test detects a case where there is an integer-valued solution. It is very obvious from Table 1 that the accuracy of our proposed method for quadratic cases with constant bound is very high.

Table 1. Testing capability of the Quadratic test for 303 pairs of one-dimensional array references with constant bounds and quadratic equation.

<table>
<thead>
<tr>
<th>The Number of array checked</th>
<th>Definitive Results</th>
<th>Accuracy Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Independent</td>
<td>Dependent</td>
</tr>
<tr>
<td>303</td>
<td>156</td>
<td>147</td>
</tr>
<tr>
<td>(100%)</td>
<td>(51%)</td>
<td>(49%)</td>
</tr>
</tbody>
</table>

VII. Conclusions

Linear expressions are the most common subscript pattern for the referenced arrays. Nevertheless, Petersen et al. [10] indicated there are many nonlinear cases, near to 6503 cases, in the analyzed Perfect Benchmarks and many of them are in quadratic form.

In this paper, we present an exact dependence test, called the Quadratic Test, that can handle quadratic expression array subscripts exactly. This method can find all of the integer solutions. Using this information, it is helpful for further parallelization. Mathematically, the Quadratic test can be thought as an extension version of the IR test [6].

At present, the Quadratic test is only suitable for the loops in which the loop bounds are rectangular at compiling-time. Our next objectives will loosen this restriction to accept more general criterion, such as loop limit is triangular, or unknown, or a function of outer loop indices, and to extend it to check for the data dependence subject to an arbitrary direction vector.

References