Unimodular Transformation Of Double Loops

-Utpal Banerjee

Seminar By

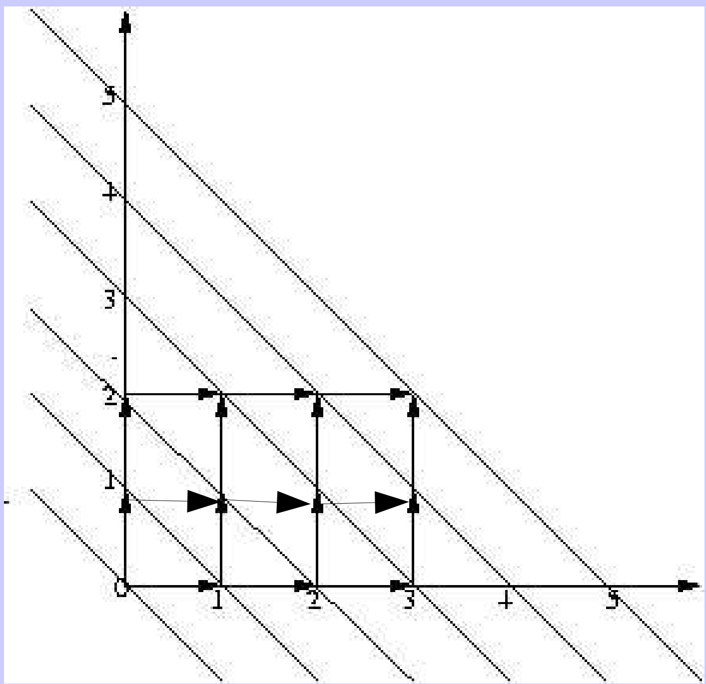
Sandeep, Shrirang, Amit

MTech I (CSE)

Outline

- Wavefront Transformation
- Basics : Unimodular Matrices
- Algorithm to apply Unimodular Transform ations
- Special Cases of Transform ations
- Algorithm s to find U.
- Summary

Wavefront Transformation




```
do I1= 0 , 3 , 1
    do I2 = 0 , 2 , 1
       A(I1,I2)=A(I1-1,I2)+A(I1,I2-1)
    end do
end do
```

```
do K1= 0 , 5 , 1
    do K2 = max {0,K1-3} , min {2,K1} ,1
       A(K1-K2,K2)=A(K1-K2-1,K2)
                       +A(K1-K2,K2-1)
    end do
end do
```

Unimodular Matrices

Original Loop:

```
L1:   do I1 = n1 , N1 , 1
L2:       do I2 = n2 , N2 , 1
              H( I1 , I2 )
          end do
      end do
```

Transformed Loop:

$L1^U$: do k1 = m1 , M1 , 1

$L2^U(K1)$: do k2 = m2 , M2 , 1

$H^U(K1, K2)$

end do

end do

(K1,K2)=(I1,I2) U

where U is integer unimodular matrix (det(U)=1).

First Column defines the direction of the wavefront

Second Column is chosen such that U is unimodular

$$U = \begin{bmatrix} u11 & u12 \\ u21 & u22 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

Distance and Direction Vectors

- (I1,I2)>(J1,J2) if either I1>J1 or I1=J1 and I2>J2
- Distance Vector (D1,D2).
- Thus D>0.

- Also that direction vector d=(s1,s2) > 0
- If direction vector if of the form (1,s2) then level-1 dependence
- If direction vector if of the form (0,1) then level-2 dependence

Outline of transforming a nest using Banerjee's Unimodular Transformations

- We have a nest $(L1, L2)$
- We have an appropriate unimodular matrix say 'U'
- Banerjee's algorithm transforms the nest into a new nest $(L1, L2)^U$ using 'U'
- The transformation applied for converting from $(L1, L2)$ to $(L1, L2)^U$, depends upon the value of 'U'
- The transformation could be any of Loop interchange, Loop reversal or simply a parallelizing transformation.

Example

Original Loop $(L1, L2)$:

```
L1:     do I1 = n1 ,  N1 ,  1
L2:             do I2 = n2 ,  N2 , 1
                        Body ( I1 , I2 )
                end do
        end do
```

Example :

```
L1:     do I1 = 0 ,  3 ,  1
L2:             do I2 = 0 ,  2 , 1
                        A (I1 , I2) = A (I1–1 , I2)
                                    + A (I1 , I2-1)
                end do
        end do
```

Given a unimodular matrix 'U':

$$\begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix}$$

u11 = 1

u12 = 0

u21 = 1

u22 = 1

n1 = 0

N1 = 3

n2 = 0

N2 = 2

Algorithm for transforming the nest

Step 1:

Set D = determinant(U)

If |D| not 1, then terminate (U is not Unimodular)

$$\begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix} \qquad D = 1$$

Step 2:

Find Expressions for I1 and I2 in terms of K1 and K2

Replace-

I1 by D. u22. K1 + (-D). u21. K2

Replace-

I2 by (-D). u12. K1 + D. u11. K2

in the loop body to get transformed body

u11 = 1   u12 = 0   u21 = 1   u22 = 1

D = 1

I1 = (1). (1). K1 + (-1). (1). K2   = K1 - K2

I2 = (-1). (0). K1 + (1). (1). K2   = K2

Old Body:

A (I1 , I2) = A (I1–1 , I2) + A (I1 , I2-1)

Transformed body:

A (K1–K2, K2) = A (K1–K2-1, K2) + A (K1–K2, K2-1)

Step 3:

In the Transformed Loop $(L1, L2)^U$:

$L1^U$: do K1 = m1 , M1 , 1

$L2^U(K1)$: do K2 = m2(K1) , M2(K1) , 1

Body ^{U}(K1 , K2)

end do

end do

Set,

$$m1 = (u11^+ . n1 - u11^- . N1) + (u21^+ . n2 - u21^- . N2)$$

$$M1 = (u11^+ . N1 - u11^- . n1) + (u21^+ . N2 - u21^- . n2)$$

We know,

n1 = 0	N1 = 3	n2 = 0	N2 = 2
u11 = 1	u12 = 0	u21 = 1	u22 = 1

m1 = ((1). (0) – (0). (3)) + ((1). (0) – (0). (2)) = 0

M1 = ((1). (3) – (0). (0)) + ((1). (2) – (0). (0)) = 5

Step 4:

```
If (-D. u21 > 0 )
then
        LB1=(n1–D.u22.K1)/(-D.u21)
        UB1=(N1–D.u22.K1)/(-D.u21)
else
        UB1=(n1–D.u22.K1)/(-D.u21)
        LB1=(N1–D.u22.K1)/(-D.u21)
end

If (D. u11 > 0 )
then
        LB2=(n2+D.u12.K1)/(D.u11)
        UB2=(N2+D.u12.K1)/(D.u11)
else
        UB2=(n2+D.u12.K1)/(D.u11)
        LB2=(N2+D.u12.K1)/(D.u11)
end
```

We know, D = 1 (Already Calculated)

n1=0 N1=3 n2=0 N2=2

u11=1 u12=0 u21=1 u22=1

So,

-D.u21=-1

Expression	Value	-D.u21>0	-D.u21<0
(n1–D.u22.K1)/(-D.u21)	K1	~~LB1~~	UB1
(N1–D.u22.K1)/(-D.u21)	K1-3	~~UB1~~	LB1

Also, D.u11=1

		D.u11>0	D.u11<0
(n2+D.u12.K1)/(D.u11)	0	LB2	~~UB2~~
(N2+D.u12.K1)/(D.u11)	2	UB2	~~LB2~~

Step 5:

m2(K1) = ceiling(max(LB1,LB2))

M2(K1) = floor(min(UB1,UB2))

From Previous Slide -

-D.u21=-1

Expression	Value	-D.u21>0	-D.u21<0
(n1–D.u22.K1)/(-D.u21)	K1	~~LB1~~	UB1
(N1–D.u22.K1)/(-D.u21)	K1-3	~~UB1~~	LB1
		Also, D.u11=1	
		D.u11>0	D.u11<0
(n2+D.u12.K1)/(D.u11)	0	LB2	~~UB2~~
(N2+D.u12.K1)/(D.u11)	2	UB2	~~LB2~~

So, m2(K1) = ceiling(max(0,K1-3))

and,M2(K1) = floor(min(2,K1))

Step 6: Substitute the calculated values in the foll nest-

Transformed Nest :

```
L1^U :      do K1 = m1 , M1 , 1
L2^U(K1):        do K2 = m2(K1) , M2(K1) , 1
                      Body^U ( K1 , K2 )
                 end do
            end do
```

We have calculated-

m1=0   M1 = 5   m2(K1) = ceiling(max(0,K1-3))   M2(K1) = floor(min(2,K1))

$Body^U$(K1,K2) : A (K1–K2, K2) = A (K1–K2-1, K2) + A (K1–K2, K2-1)

SUBSTITUTING, we get the ***Transformed Nest*** :

```
L1:      do K1 = 0 , M1 , 5
L2:           do K2 = ceiling(max(0,K1-3)) , floor(min(2,K1)) , 1
                   A (K1–K2, K2) = A (K1–K2-1, K2) + A (K1–K2, K2-1)
              end do
         end do
```

Outer Loop reversal

Reversal Matrix:

$$U = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Original Loop

```
do I1 = n1, N1, 1
        do I2 = n2, N2, 1
              H(I1, I2)
        end do
end do
```

Transformed Loop

```
do K1 = -N1, -n1, 1
        do K2 = n2, N2, 1
              H(-K1, K2)
        end do
end do
```

Loop Interchange

Interchange Matrix

$$U = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Original Loop

```
do I1 = n1, N1, 1
        do I2 = n2, N2, 1
              H(I1, I2)
        end do
end do
```

Transformed Loop

```
do K1 = n2, N2, 1
        do K2 = n1, N1, 1
              H(K2, K1)
        end do
end do
```

Loop Skewing

Skewing Matrix

$$U = \begin{bmatrix} 1 & q \\ 0 & 1 \end{bmatrix}$$

Original Loop

```
do I1 = n1, N1, 1
        do I2 = n2, N2, 1
              H(I1, I2)
        end do
end do
```

Transformed Loop

```
do K1 = n1, N1, 1
        do K2 = n2 + q*K1, N2 + q*K1, 1
              H(K1, K2 - q*K1)
        end do
end do
```

Algorithms to find Unimodular Matrix 'U'

- Given a nest (L1, L2)
- Algorithm 1
  - Transformed nest is equivalent to the original nest.
  - The outer loop L1 can be executed in parallel.
  - The iteration count of L1 is maximized.
- Algorithm 2
  - Transformed nest is equivalent to the original nest.
  - The inner loop L2 can be executed in parallel.
  - The iteration count of L1 is minimized.

Algorithm 1

1. Find all dependence distance vectors.

```
L1:        do I1 = 5, 100, 1
L2:        do I2 = 16, 80, 1
      S:        A(I1, I2) = A(I1 – 2, I2 – 4) + A(I1 – 3, I2 – 6)
           end do
      end do
```

Distance Vectors:

(2, 4)

(3, 6)

Algorithm 1 (cont.)

2. If there are no distance vectors, then set

$$U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$ If N1 – n1 >= N2 – n2,

$$U = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$ Otherwise,

And go to step 5.

3. If all depedence vectors are of the form c*(a1, a2) goto next step.
   Else “No unimodular transformation such that outer loop can be executed in parallel”.

   (2, 4) = 2(1, 2)

   (3, 6) = 3(1, 2)

Algorithm 1 (cont.)

4. Find g = gcd(a1, a2) and a pair of integers (u12, u22)

   such that a1*u12 + a2*u22 = g.

Calculate,

(u11, u21) = (a2/g, -a1/g)

gcd (1, 2) = 1

1*u12 + 2*u22 = 1

u12 = 1, u22 = 0

u11 = 2 / 1 = 2

u21 = -1 / 1 = -1

Solution

$$U = \begin{bmatrix} u11 & u12 \\ u21 & u22 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ -1 & 0 \end{bmatrix}$$

Algorithm to Parallelize Inner Loop

1. Find all the dependence distances of form (D1,D2) and direction vectors in (L1,L2).
2. Initialize List={ $(1,0)^T$, $(0,1)^T$, $(0,-1)^T$, $(1,1)^T$ }
3. Generate Final List

    if the nest has the direction vector (1,0) then

        delete from List : $(0,-1)^T$ $(0,1)^T$

    if the nest has the direction vector (0,1) then

        delete from List : $(1,0)^T$ $(0,-1)^T$

    if the nest has the direction vector (1,1) then

        delete from List : $(0,-1)^T$

    if the nest has the direction vector (1,-1) then

    begin

        delete from List : $(0,1)^T$ $(1,1)^T$

        set m = ceil(max(-D2/D1)) + 1 , where Maximum is taken over all distance vectors (D1,D2) such that D1>0 and D2<0.

        Add to the list $(m,1)^T$

    end

Algorithm to Parallelize Inner Loop (Cont..)

4. Get the best choice for u11,u21

    If (N1-n1)>(N2-n2)

        Take first vector from the following sequence which is in the list

        $(0,1)^T$ $(0,-1)^T$ $(1,0)^T$ $(1,1)^T$ $(m,1)^T$

    else

        Take first vector from the following sequence which is in the list

        $(1,0)^T$ $(0,1)^T$ $(0,-1)^T$ $(1,1)^T$ $(m,1)^T$

5. If the best choice of (u11,u21)=(1,0) then simply mark the inner loop as parallel.

    Otherwise,

    Form U by taking the first column from best choice and set u12=1 ,u22=0.

    Then transform the nest using algorithm seen previously.

Example

```
do I1 = 5, 100, 1
    do I2 = 5, 100, 1
          A(I1, I2) = A(I1, I2 – 1) + A(I1 – 2,I2+3)
                             +A(I1-3,I2+7)
   end do
end do
```

- The distance vectors are (0,1) , (2,-3) and (3,-7)
- The direction vectors are (0,1) and (1,-1)
- Initialize List={ $(1,0)^T$ $(0,1)^T$ $(0,-1)^T$ $(1,1)^T$ }
- Consider pair (0,1) from direction vectors. so delete pairs $(0,-1)^T$ $(1,0)^T$ from List.
- Consider pair (1,-1) so
- Delete pairs $(0,1)^T$ $(1,1)^T$ from List.

  Now m=ceil(max{-(-3)/2,(-7)/2})+1=3

  Add to the List $(3,1)^T$
- Thus the only element left in the list is $(3,1)^T$ And will be selected as best choice. Thus U is

$$U = \begin{bmatrix} u11 & u12 \\ u21 & u22 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 1 & 0 \end{bmatrix}$$

Summary

- Wavefront Transformation
- Basics : Unimodular Matrices
- Algorithm to apply Unimodular Transform ations
- Special Cases of Transform ations
- Algorithm s to find U.

Thank You!