某循环如下：

for i = 1 to 4 do

for j = 1 to 4 do

H(i,j);

其依赖距离向量为(1,0)和(0,1)，循环迭代依赖图如下。



上述为某循环迭代依赖图。其依赖距离向量为(1,0)和(0,1)，假设均为流依赖。所以，该循环外层和内层均无法直接并行化，内层也无法向量化。现在对该循环施加单模变换，引入单模矩阵U(矩阵行列式绝对值为1):

而循环依赖矩阵 D 为：

则变换后的依赖矩阵为DU:

变换后的依赖（距离）向量均为正向量，说明该变换是正确的。

此外，依赖向量为(1,1)和(1,0)。可以看出，此时变换后循环存在仅由外层循环所携带或主导的跨迭代依赖关系，因此，可以对内层循环实施并行化或向量化。

那么，变换后循环如何构成呢？设变换后循环为：

for k1 …

for k2 …

H’(k1,k2);

则，(k1,k2) = (i,j) U，即，

k1 = i + j

k2 = i

由原来的循环迭代变量范围，有：

将i，j由k1,k2来表示，代入上述不等式，有：

按k2, k1的次序，依次消除相关变量：

1）先将不等式改写为以k2主导的如下形式：

则，我们可以得到k2的范围：

2）进一步，我们得到关于k1的不等式表示：

最后，我们得到k1的范围如下：

因此，变换后循环：

for k1 = 2 to 8 do

for k2 = to do

//pardo doall k2层循环可并行化

begin

i = k2;

j = k1 – k2;

H(i,j);

End

练习：

（a）某循环如下：

for i = 1 to 100 do

for j = 1 to 100 do

H(i,j);

其依赖距离向量为(1,-1)和(0,1)，尝试某个单模变换，可以并行化内层循环。

考虑单模变换矩阵：

（b）某循环如下：

for i = 1 to 100 do

for j = 1 to 100 do

for k = 1 to 100 do

H(i,j);

该循环依赖距离向量为(1,0,0) , (0,1,0) 和 ( 0,0,1)。

针对该循环，做单模变换 U = ，即(k1,k2,k3) = (i,j,k) U

给出变换后循环：

for k1= …

for k2 = …

for k3 = …

H’(k1,k2,k3);