

Example 5: Matrix Multiplication

```
double X[N][N], Y[N][N], Z[N][N];  
for (i=0; i<N; i++)  
    for (j=0; j<N; j++)  
        for (k=0; k<N; k++)  
            X[i][j] += Y[i][k] * Z[k][j];
```

Example 5: Matrix Multiplication

```
double X[N][N], Y[N][N], Z[N][N], tmp;
```

```
for (i=0; i<N; i++)
```

```
    for (j=0; j<N; j++){
```

```
        tmp = 0;
```

```
        for (k=0; k<N; k++)
```

```
            tmp += Y[i][k] * Z[k][j];
```

```
        X[i][j] = tmp; / Dot product inner loop
```

```
    } Y[0,0], Z[0,0], Y[0,1], Z[1,0], Y[0,2], Z[2,0] ... X[0,0],  
      Y[1,0], Z[0,1], Y[1,1], Z[1,1], Y[1,2], Z[2,1] ... X[0,1],  
      Y[2,0], Z[0,2], Y[2,1], Z[1,2], Y[2,2], Z[2,2] ... X[0,2],  
      ...
```

Example 5: Matrix Multiplication

```
double X[N][N], Y[N][N], Z[N][N], tmp;
```

```
for (i=0; i<N; i++)
```

```
    for (j=0; j<N; j++) {
```

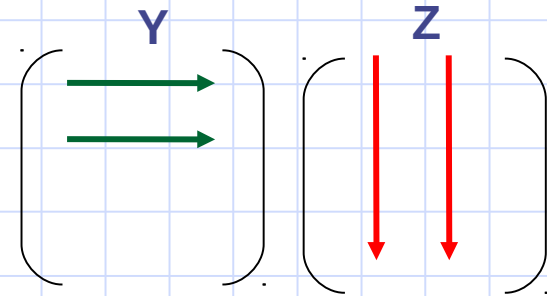
```
        tmp = 0;
```

```
        for (k=0; k<N; k++)
```

```
            tmp += Y[i][k] * Z[k][j];
```

```
        X[i][j] = tmp; / Dot product inner loop
```

```
    } Y[0,0], Z[0,0], Y[0,1], Z[1,0], Y[0,2], Z[2,0] ... X[0,0],  
      Y[1,0], Z[0,1], Y[1,1], Z[1,1], Y[1,2], Z[2,1] ... X[0,1],  
      Y[2,0], Z[0,2], Y[2,1], Z[1,2], Y[2,2], Z[2,2] ... X[0,2],  
      ...
```



Matmul: Loop Interchange

- We can interchange the 3 loops
- Example: Interchange i and k loops – make the loops “kji” instead of “ijk”

```
double X[N][N], Y[N][N], Z[N][N];
```

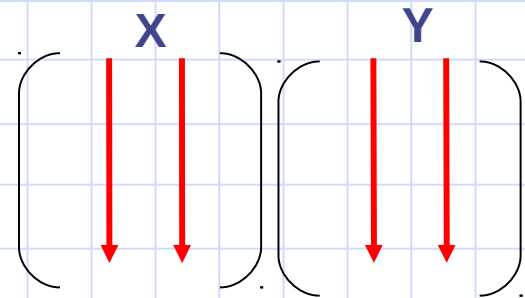
```
for (k=0; k<N; k++)
```

```
    for (j=0; j<N; j++)
```

```
        for (i=0; i<N; i++)
```

```
            X[i][j] += Y[i][k] * Z[k][j];
```

- For the innermost loop: $Z[k][j]$ can be loaded into register once for each (k,j) , reducing the number of memory references



“Loop Unrolling”

```
double X[10];  
for (i=0; i<10; i++)  
    X[i] = X[i] - 1;
```

Unrolled once:

```
for (i=0; i<10; i+=2){  
    X[i] = X[i] - 1;  
    X[i+1] = X[i+1] - 1;  
}
```

Fully unrolled:

```
X[0] = X[0] - 1;  
X[1] = X[1] - 1;  
X[2] = X[2] - 1;  
...  
X[9] = X[9] - 1;
```

Unrolling Matrix Multiplication

```
double X[N][N], Y[N][N], Z[N][N];  
for (i=0; i<N; i++)  
    for (j=0; j<N; j++)  
        for (k=0; k<N; k++)  
            X[i][j] += Y[i][k] * Z[k][j];
```

Let us unroll the k loop once

Matmul: k loop unrolled

```
double X[N][N], Y[N][N], Z[N][N];  
for (i=0; i<N; i++)  
    for (j=0; j<N; j++)  
        for (k=0; k<N; k+=2) /* k loop unrolled once  
            X[i][j] += Y[i][k] * Z[k][j] + Y[i][k+1] * Z[k+1][j];
```

Now, let us also unroll the j loop once

Matmul: k and j loops unrolled

```
double X[N][N], Y[N][N], Z[N][N];
```

```
for (i=0; i<N; i++)
```

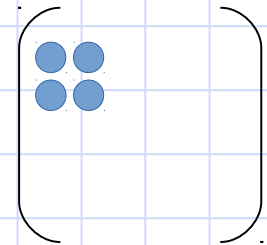
```
    for (j=0; j<N; j+=2)
```

```
        for (k=0; k<N; k+=2){    /* j and k loops unrolled once
```

```
            X[i][j] += Y[i][k] * Z[k][j] + Y[i][k+1] * Z[k+1][j];
```

```
            X[i][j+1] += Y[i][k] * Z[k][j+1] + Y[i][k+1] * Z[k+1][j+1];
```

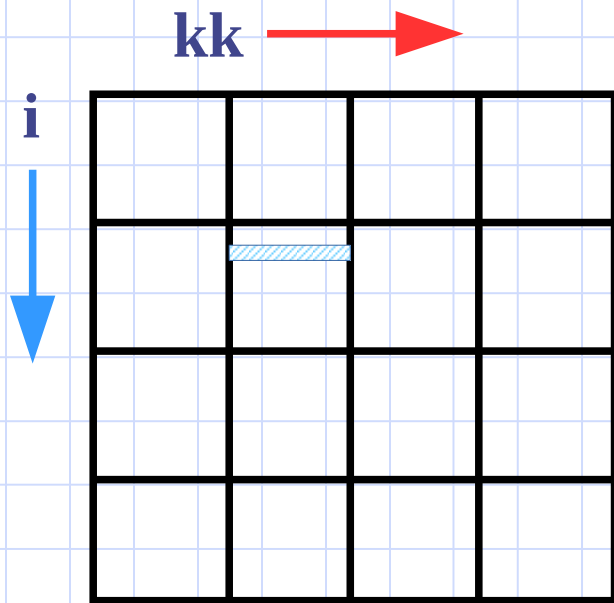
```
        }
```



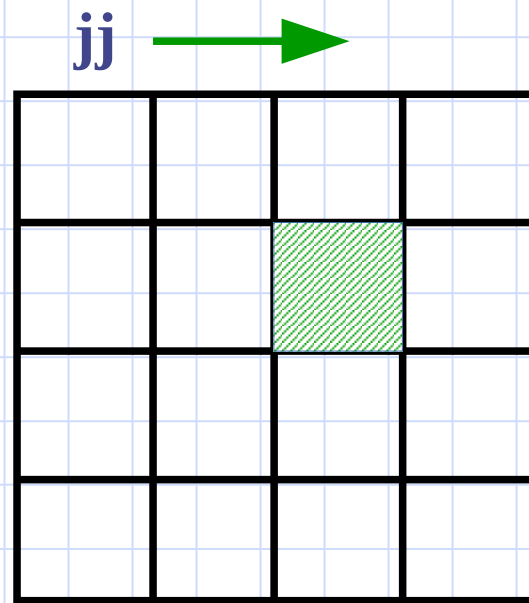
Exploits spatial locality for arrays Y, Z

Exploits temporal locality for array Y

Provides a programming idea for enhancing locality



kk

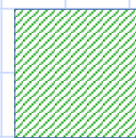
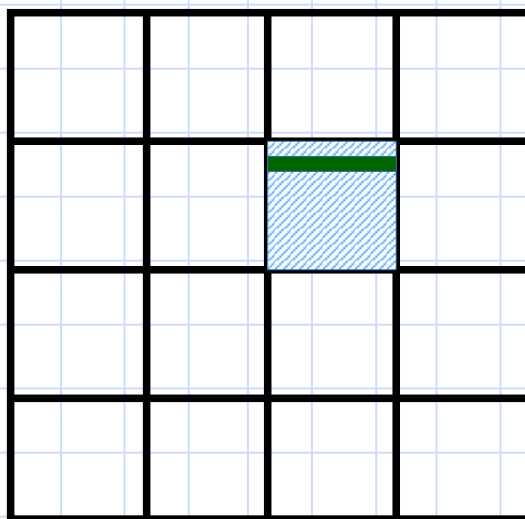


$Y_x Z$

$= X$

j

i



“Block” of
array elements

Block size, B



Row sliver

Matmul: “Blocking” or “Tiling”

```
double X[N][N], Y[N][N], Z[N][N];
for (jj=0; jj<N; jj+=B)
    for (kk=0; kk<N; kk+=B)
        for (i=0; i<N, i++){
            for (j=jj; j < min(jj+B, N); j++){
                sum = 0.0;
                for (k=kk; k<min(kk+B, N), k++){
                    sum += Y[i][k] * Z[k][j];
                }
                X[i][j] += sum;
            } /* for j */
        } /* for i */
```

Matmul: “Blocking” or “Tiling”

```
double X[N][N], Y[N][N], Z[N][N];
for (jj=0; jj<N; jj+=B)
    for (kk=0; kk<N; kk+=B)
        for (i=0; i<N, i++){
            for (j=jj; j < min(jj+B, N); j++){
                sum = 0.0;
                for (k=kk; k<min(kk+B, N), k++){
                    sum += Y[i][k] * Z[k][j];
                }
                X[i][j] += sum;
            } /* for j */
        } /* for i */
```