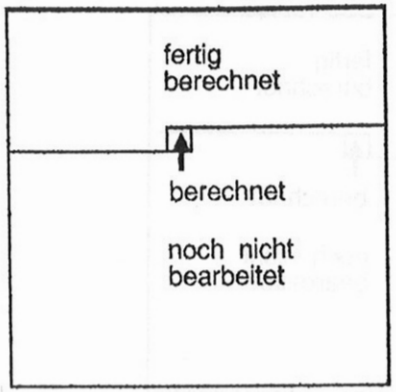
Matrix-Matrix-Multiplikation

Datenzugriff bei Indexreihenfolge ijk



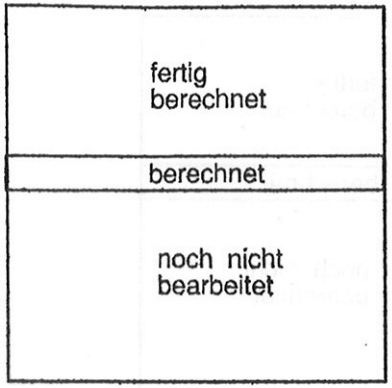
Punktalgorithmus

Vektorisierung: nicht über die innere Schleife
Parallelisierung: nicht über äußere oder mittlere Schleife
Datenzugriff: Stride m

```
      do 10 i=1,m
        do 10 j=1,n
          do 10 k=1,l
10          c(i,j)= c(i,j)+ a(i,k)* b(k,j)
```

Matrix-Matrix-Multiplikation

Datenzugriff bei Indexreihenfolge ikj



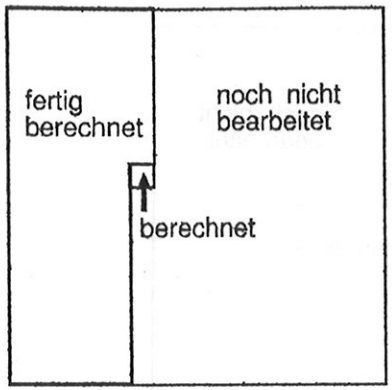
Zeilenalgorithmus

Vektorisierung: über die innere Schleife, Vektorlänge 2n (bzw. m für zwei Vektoren)
Parallelisierung: nicht über äußere oder mittlere Schleife
Datenzugriff: Stride m

```
      do 10 i=1,m
        do 10 k=1,l
          do 10 j=1,n
10          c(i,j)= c(i,j)+ a(i,k)* b(k,j)
```

Matrix-Matrix-Multiplikation

Datenzugriff bei Indexreihenfolge jik



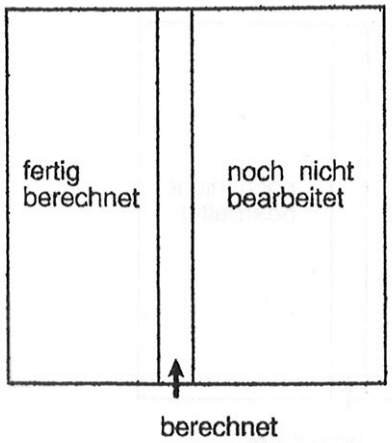
Punktalgorithmus

Vektorisierung: nicht über die innere Schleife
Parallelisierung: nicht über äußere oder mittlere Schleife
Datenzugriff: Stride 1

```
      do 10 j=1,n
        do 10 i=1,m
          do 10 k=1,l
10          c(i,j)= c(i,j)+ a(i,k)* b(k,j)
```

Matrix-Matrix-Multiplikation

Datenzugriff bei Indexreihenfolge jki



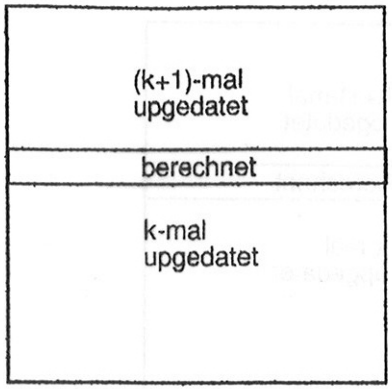
Spaltenalgorithmus

Vektorisierung: über die innere Schleife, Vektorlänge 2m (bzw. m für zwei Vektoren)
Parallelisierung: nicht über äußere oder mittlere Schleife
Datenzugriff: Stride 1

```
      do 10 j=1,n
        do 10 k=1,l
          do 10 i=1,m
10          c(i,j)= c(i,j)+ a(i,k)* b(k,j)
```

Matrix-Matrix-Multiplikation

Datenzugriff bei Indexreihenfolge kij



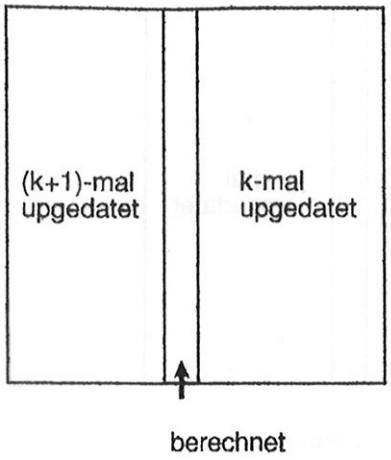
Zeilenalgorithmus

Vektorisierung: über die innere Schleife, Vektorlänge 2n (bzw. n für zwei Vektoren)
Parallelisierung: über die mittlere Schleife, Granularität m/#Prozessoren
Datenzugriff: Stride m

```
    do 10 k=1,l
      do 10 i=1,m
        do 10 j=1,n
10        c(i,j)= c(i,j)+ a(i,k)* b(k,j)
```

Matrix-Matrix-Multiplikation

Datenzugriff bei Indexreihenfolge kji



Spaltenalgorithmus

Vektorisierung: über die innere Schleife, Vektorlänge 2m (bzw. m für zwei Vektoren)
Parallelisierung: über die mittlere Schleife, Granularität n/#Prozessoren
Datenzugriff: Stride 1

```
      do 10 k=1,l
        do 10 j=1,n
          do 10 i=1,m
10          c(i,j)= c(i,j)+ a(i,k)* b(k,j)
```