

1. Optimizaciones.

1.1. Inversión de ciclos.

El método consiste en invertir los ciclos exteriores del método básico para lograr que la matriz de resultados se recorra por columnas, aprovechando mejor la memoria cache.

1.2. Multiplicación por columnas.

Este método realiza la multiplicación de matrices en forma incremental, efectuando todas las operaciones posibles con los elementos disponibles de una de las matrices y sumando los resultados parciales en la matriz resultado.

Utilizando las propiedades de las matrices se consiguen las siguientes equivalencias que proporcionan fundamento matemático al método. Por simplicidad, se utilizan matrices de 2x2.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$$

El producto del miembro izquierdo se puede escribir como

$$\left(\begin{bmatrix} a_{11} & 0 \\ a_{21} & 0 \end{bmatrix} + \begin{bmatrix} 0 & a_{12} \\ 0 & a_{22} \end{bmatrix} \right) \left(\begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \end{bmatrix} + \begin{bmatrix} 0 & b_{12} \\ 0 & b_{22} \end{bmatrix} \right)$$

Y expandiendo los paréntesis resulta

$$\begin{bmatrix} a_{11} & 0 \\ a_{21} & 0 \end{bmatrix} \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \end{bmatrix} + \begin{bmatrix} 0 & a_{12} \\ 0 & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \end{bmatrix} + \begin{bmatrix} a_{11} & 0 \\ a_{21} & 0 \end{bmatrix} \begin{bmatrix} 0 & b_{12} \\ 0 & b_{22} \end{bmatrix} + \begin{bmatrix} 0 & a_{12} \\ 0 & a_{22} \end{bmatrix} \begin{bmatrix} 0 & b_{12} \\ 0 & b_{22} \end{bmatrix}$$

Así, la matriz producto se puede descomponer en la siguiente suma

$$\begin{bmatrix} a_{11}b_{11} & 0 \\ a_{21}b_{11} & 0 \end{bmatrix} + \begin{bmatrix} a_{12}b_{21} & 0 \\ a_{22}b_{21} & 0 \end{bmatrix} + \begin{bmatrix} 0 & a_{11}b_{12} \\ 0 & a_{21}b_{12} \end{bmatrix} + \begin{bmatrix} 0 & a_{12}b_{22} \\ 0 & a_{22}b_{22} \end{bmatrix}$$

Como se puede apreciar, las tres matrices se acceden por columnas. Esto resulta en una mejor utilización de la memoria cache.

Sin embargo, para que funcione correctamente, se requiere que la matriz donde se almacena el resultado haya sido inicializada, ya sea previamente en 0 o con los primeros valores calculados de cada elemento.

1.3. Multiplicación por bloques

1.4. Básico con trasposición.

Al igual que en el método anterior, la trasposición de la primera matriz del producto permite que posiciones contiguas de memoria sean accedidas para las tres matrices, optimizando el uso de memoria cache.

Sin embargo, el desempeño de este método se ve impactado por la penalidad de calcular la matriz traspuesta, que requiere movimientos de grandes cantidades de datos.