Computación de Altas Prestaciones 2022-2023
Sesión 5

Computación de Altas Prestaciones

- Computación en precisión finita, conceptos básicos
- Normas vectoriales
- Normas matriciales
- Optimización de algoritmos matriciales

Estudio de Caso: Resolución de un Sistema de ecuaciones lineales

- Supongamos que nos encargan resolver un Sistema de ecuaciones lineales,

$$Ax = b, A \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^{n}$$

en una aplicación en la que la precisión es importante (debemos comprobar que la solución está calculada correctamente)

para simular el proceso, vamos a resolver un sistema en Matlab, usando una matriz de Hilbert de tamaño 12

```
>>A=hilb(12);                    (A(i,j)=1/(i+j-1))
```

Estudio de Caso: Resolución de un Sistema de ecuaciones lineales

Vamos a resolver el sistema con la barra de Matlab (\) que sirve para resolver sistemas de ecuaciones lineales; Para comprobar que funciona correctamente, vamos a probar con un sistema del cual conocemos la solución

```
>>b=A*ones(12,1);
```

Ahora, al resolver el sistema Ax=b me debería dar x todo unos.

```
 >>z=A\b
z =
   1.0000
   1.0000
   1.0000
   1.0000
   0.9997
   1.0018
   0.9935
   1.0141
   0.9809
   1.0156
   0.9929
   1.0014
```

Porque no funciona?

Computación con precisión finita

- Los cálculos en el ordenador NO SON 100% EXACTOS.
- Cada número se representa en el ordenador con un número finito de bits; en muchos casos, no se pueden guardar todos los dígitos, y se comete error de redondeo.
- A menudo esto no tiene influencia, pero a veces sí (la matriz de Hilbert es un caso "patológico", pero los casos patológicos a veces suceden...)

¿La solución que hemos obtenido, es aceptable?

¿Hasta que punto es una buena solución?

Tendríamos que MEDIR el error

Medir Errores

Recordemos: Dado un escalar x y una aproximación a ese escalar z:

-El ERROR ABSOLUTO de la aproximación es $|x - z|$

-El ERROR RELATIVO de la aproximación es $\frac{|x-z|}{|x|}$

-El ERROR EN TANTO POR CIEN de la aproximación es $\frac{|x-z|}{|x|} * 100$

-Si tenemos un error relativo menor que 10^{-k}, nuestra aproximación tiene al menos k dígitos significativos (correctos)

Sin embargo, en nuestro problema la solución y la aproximación no son números, sino vectores. Necesitamos una forma de medir el error en vectores: NORMAS VECTORIALES

Normas Vectoriales

- Una norma vectorial es una aplicación de Rn en R+ (A cada vector le corresponde un número positivo)
- La norma euclídea es el ejemplo más conocido de norma vectorial:

Dado el vector (3,2,-5) su norma euclidea es $\sqrt[2]{3^2+2^2+(-5)^2} = \sqrt{38}$

Dado el vector $(x_1, x_2, \cdots, x_n)$ su norma es $\sqrt[2]{x_1^2+x_2^2+\cdots+x_n^2}$

$f: \mathbb{R}^n \to \mathbb{R}$ es una norma vectorial si:

1) $f(x) \geq 0, y\ f(x) = 0 \leftrightarrow x = 0$
2) $f(x+y) \leq f(x) + f(y), \forall\, x, y \in \mathbb{R}^n$
3) $f(\alpha x) = |\alpha| f(x), \forall \alpha \in \mathbb{R}, \forall x \in \mathbb{R}^n$

Normas Vectoriales

-Aparte de la norma euclidea, también conocida como norma-2, las normas mas conocidas y fáciles de calcular son la norma 1 y la norma infinito:

$$\|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|$$

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|$$

-Cualquier norma da una idea del "tamaño" de un vector

-Para calcular el error absoluto de un vector respecto x a otro z, calculamos la norma de la diferencia: $\|x - z\|$, y el error relativo: $\frac{\|x-z\|}{\|x\|}$

En nuestro ejemplo:

```
>> x=ones(12,1); z=A\b; norm(z-x)
ans =    0.0301
 >> norm(z-x)/norm(x)
ans =    0.0087
```

Análisis del error

– El error obtenido es aproximadamente un 0.3%. ¿esto es bueno o malo? Se pueden resolver sistemas con mayor precisión?

– Probemos con una matriz aleatoria

```
>> A=rand(12); b=A*ones(12,1); z=A\b;
>> norm(z-ones(12,1))/norm(ones(12,1))
ans =   3.5705e-15
```

Mucho mejor...pero no es 0. Probemos a multiplicar A por 1000 y b por 20

```
>> A=A*1000; b=A*ones(12,1)*20; z=A\b; norm(z-
ones(12,1)*20)/norm(ones(12,1)*20)
```

ans =   2.9026e-15 (prácticamente igual)

Análisis del error

Casi igual al de antes; Si lo intentamos para cualquier sistema, no podremos bajar de 10^(-16).

Para un tipo de datos dado, existe un límite en la precisión que podemos alcanzar; viene dado por el “epsilon” de la máquina;

```
>> help eps
 eps  Spacing of floating point numbers.
    D = eps(X), is the positive distance from ABS(X) to the next larger in
    magnitude floating point number of the same precision as X.
    X may be either double precision or single precision.
    For all X, eps(X) is equal to eps(ABS(X)).

    eps, with no arguments, is the distance from 1.0 to the next larger double
    precision number, that is eps with no arguments returns 2^(-52).
```

Análisis del error

```
>> eps                          //doble precisión
ans = 2.2204e-16

>> eps(single(1))               //simple precisión
ans =  1.1921e-07
```

En Doble precisión cada número se guarda con 52 dígitos binarios ~16 dígitos decimales. Lógicamente, no se puede (salvo accidente) obtener una precisión relativa mayor que 10^(-16).

En nuestro ejemplo de la matriz de Hilbert, el error relativo es 3* 10^-3, menor que 10^-2. Tenemos dos dígitos correctos.

Normas Matriciales

Con frecuencia, el resultado de nuestros cálculos serán matrices.

Por ejemplo, la descomposición LU de una matriz A obtiene matrices L (triangular inferior) y U (triangular superior) tal que el producto de L por U es igual a A.

Para medir la precisión del cálculo, podemos multiplicar L por U y compararlo con A; Para hacer esta comparación necesitamos normas matriciales

$f: \mathbb{R}^{m\times n} \to \mathbb{R}$ es una norma matricial si:

1) $f(A) \geq 0, y\ f(A) = 0 \leftrightarrow A = 0$

2) $f(A+B) \leq f(A) + f(B), \forall\ A, B \in \mathbb{R}^{m\times n}$

3) $f(\alpha A) = |\alpha| f(A), \forall \alpha \in \mathbb{R}, \forall A \in \mathbb{R}^{m\times n}$

Normas Matriciales

Norma de Frobenius:

$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j}^2 \right)^{\frac{1}{2}}$$

Error de descomposición LU de matriz de hilbert:

```
>> A=rand(12);[L,U]=lu(A);norm(L*U-A)
```

Conclusiones

-Los cálculos en el ordenador son fiables...pero no siempre. Hay que ser precavidos, sobre todo cuando el problema es muy grande.

-Las normas vectoriales dan una idea de como de “grandes” son los números contenidos en un vector; las podemos utilizar para medir el error

-Las normas matriciales dan una idea de como de “grandes” son los números contenidos en una matriz; las podemos utilizar para medir el error

-Los números reales en precisión simple tienen aprox. 7 dígitos decimales significativos, 16 en precisión doble;

Optimización de algoritmos matriciales para matrices con estructura

Es muy frecuente que las matrices con las que tenemos que trabajar tengan una cierta estructura

MATRICES CON ESTRUCTURA POR "CEROS".

-Matrices Banda

-Matrices triangulares

-Matrices diagonales, ...

Optimización de algoritmos matriciales para matrices con estructura

$$\begin{bmatrix} x & x & x & 0 & 0 \\ x & x & x & x & 0 \\ 0 & x & x & x & x \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

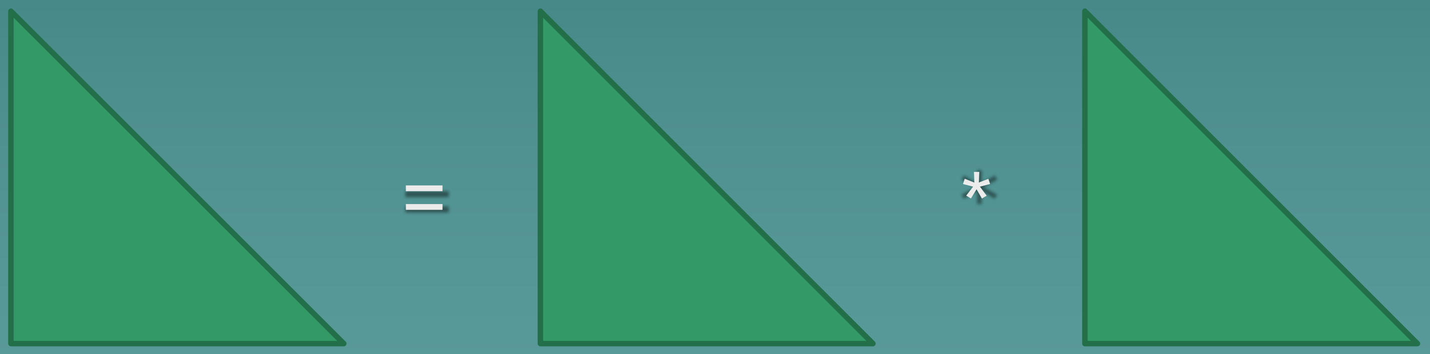
Dada una matriz A $\in \Re^{n*m}$ tiene anchura de banda inferior **p** si $i>j+p \rightarrow a_{i,j}=0$

Dada una matriz A $\in \Re^{n*m}$ tiene anchura de banda superior **q** si $j>i+q \rightarrow a_{i,j}=0$

Optimización de algoritmos matriciales para matrices con estructura

Si la estructura de la matriz es conocida, a menudo los algoritmos se pueden optimizar mucho con muy poco esfuerzo:

Optimización del producto de dos matrices triangulares inferiores



Optimización de algoritmos matriciales para matrices con estructura

Partimos del algoritmo para producto de matrices generales:

```
C=0
For i=1:n
        For j=1:n
                For k=1:n
                        C(i,j)=C(i,j)+A(i,k)*B(k,j)
                End
        End
End
```

¿Cómo se optimiza?
Teniendo en cuenta estos dos detalles:

1) El producto de dos matrices triangulares inferiores es triangular inferior

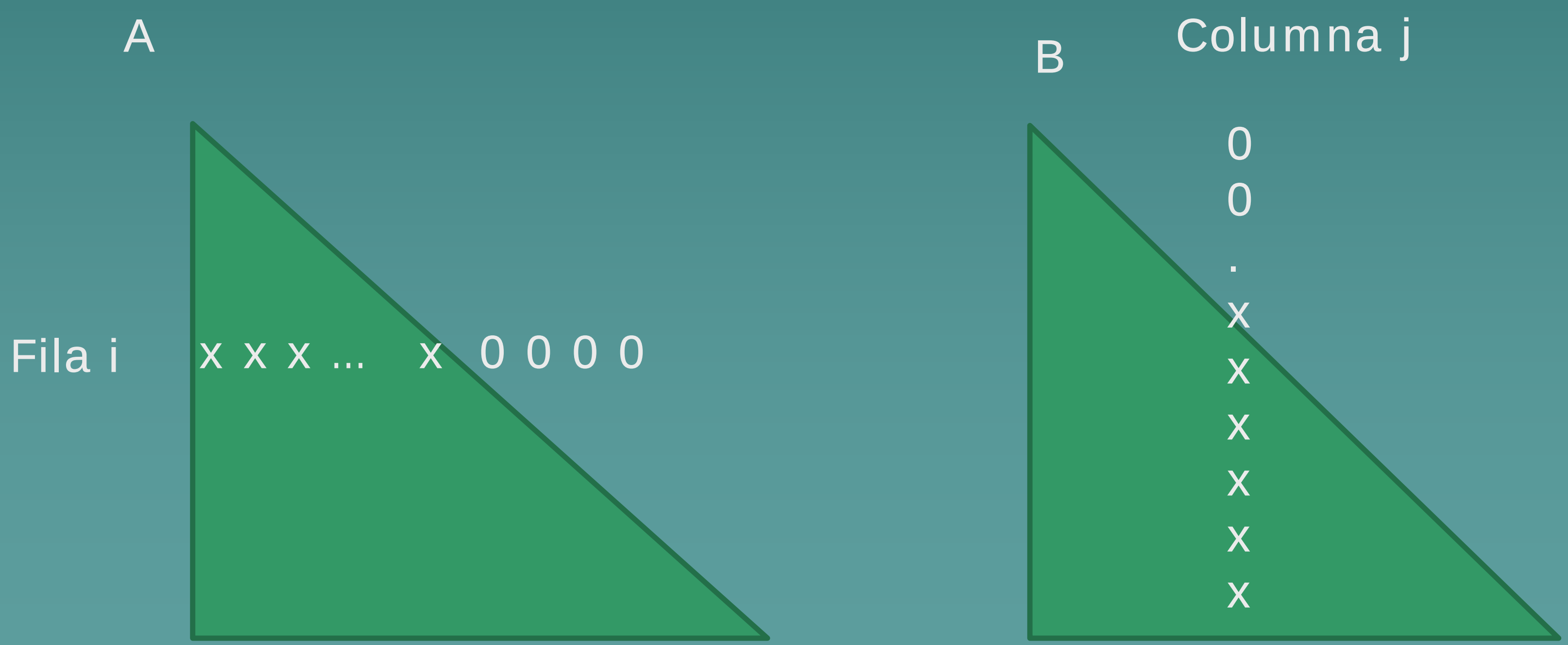
2) El elemento (i,j) de C se obtiene como un producto escalar “reducido”.

Optimización de algoritmos matriciales para matrices con estructura

Cálculo de $C(i,j)$: $i>=j$ por ser triangular inferior

$$C(i,j)=A(i,1)*B(1,j)+A(i,2)*B(2,j)+\ldots+A(i,j)*B(j,j)+\ldots$$

$$+\ldots+A(i,i)*B(i,j)+A(i,i+1)*B(i+1,j)+\ldots+A(i,n)*B(n,i)$$

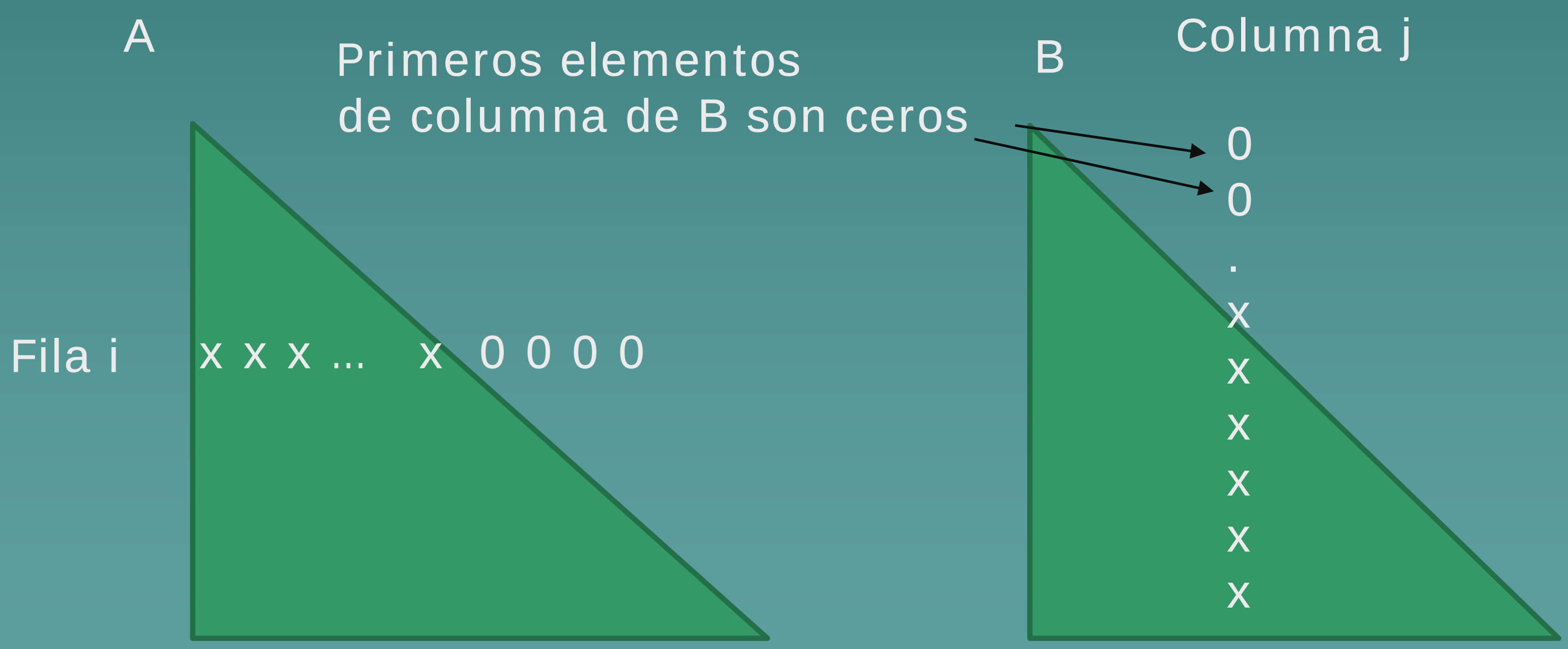


Optimización de algoritmos matriciales para matrices con estructura

Cálculo de C(i,j): i>=j por ser triangular inferior

$$C(i,j)=\cancel{A(i,1)*B(1,j)}+\cancel{A(i,2)*B(2,j)}+\ldots+A(i,j)*B(j,j)+\ldots$$

$$+\ \ldots+A(i,i)*B(i,j)+A(i,i+1)*B(i+1,j)+\ldots+A(i,n)*B(n,i)$$

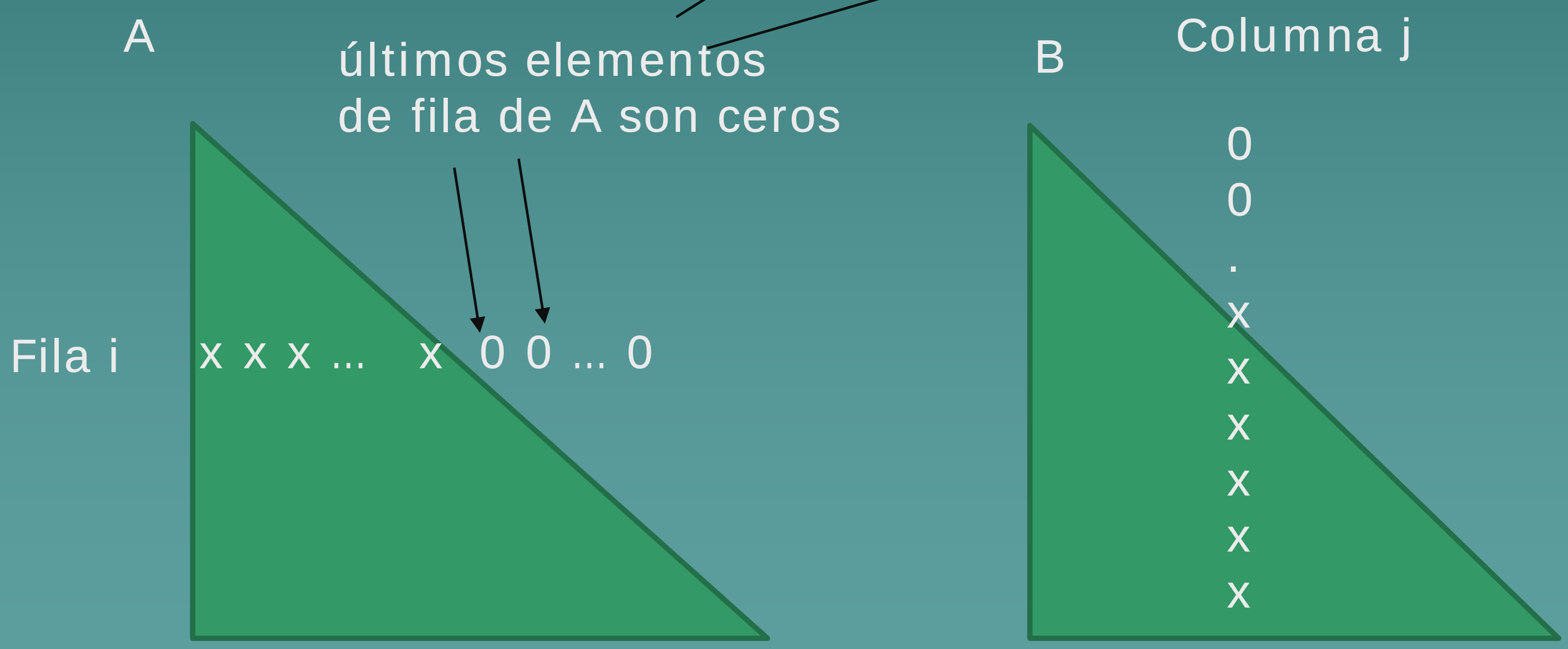


Optimización de algoritmos matriciales para matrices con estructura

Cálculo de C(i,j): i>=j por ser triangular inferior

$$C(i,j)=\cancel{A(i,1)*B(1,j)}+\cancel{A(i,2)*B(2,j)}+\ldots+A(i,j)*B(j,j)+\ldots$$

$$+\ \ldots+A(i,i)*B(i,j)+\ \cancel{A(i,i+1)*B(i+1,j)}+\ldots+\cancel{A(i,n)*B(n,i)}$$



Optimización de algoritmos matriciales para matrices con estructura

Algoritmo resultante

cambios

```
C=0
For i=1:n
        For j=1:i
                For k=j:i
                        C(i,j)=C(i,j)+A(i,k)*B(k,j)
                End
        End
End
```

Coste: $n^3/3$

Optimización de algoritmos matriciales para matrices con estructura

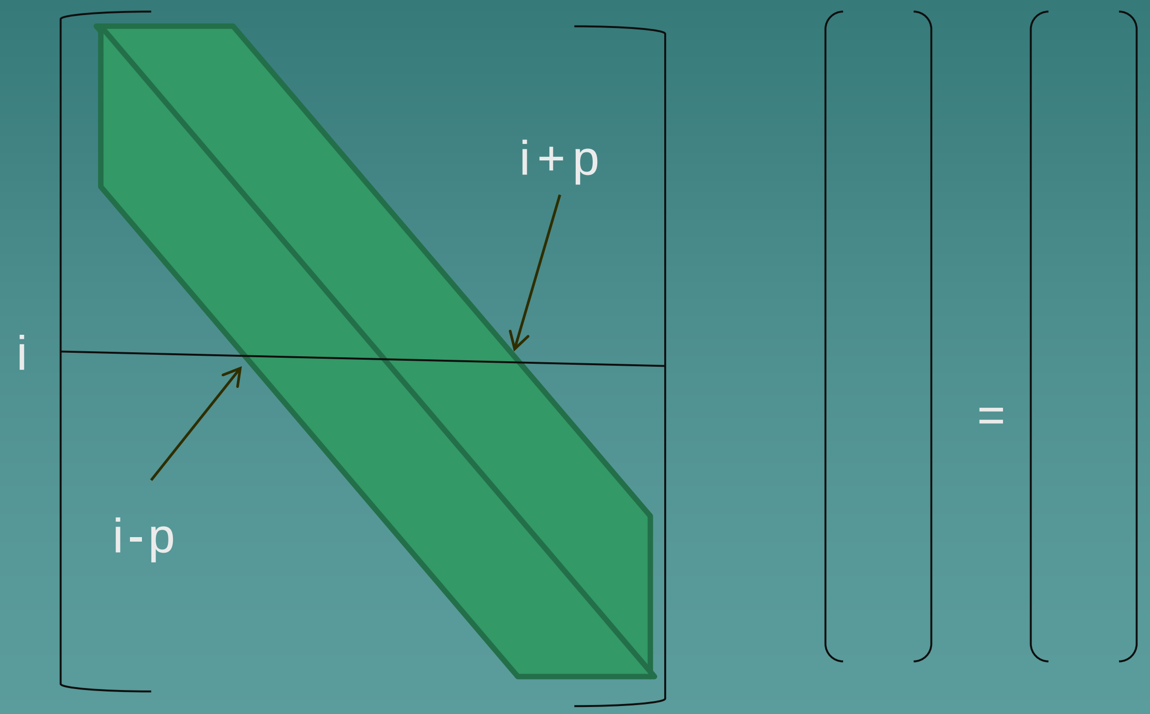
Consideremos el producto de matriz por vector:

```
for i=1:m
    for j=1:n
        y_i=y_i+A_i,j*x_j
    end
end
```

Supongamos que la matriz A es banda, con ancho de banda p (tanto inferior como superior) . Como podemos optimizar este algoritmo?

Optimización de algoritmos matriciales para matrices con estructura

Y(i)=A(i,1)*x(1)+A(i,2)*x(2)+..+A(i-p,i)*x(i-p)+...+A(i,i)*x(i)+....+ A(i+p,i)+....A(i,n)*x(n)



Optimización de algoritmos matriciales para matrices con estructura

Solución:

```
for i=1:m
    for j=max(1,i-p):min(n,i+p)
        y_i=y_i+A_i,j*x_j
    end
end
```

$$y_i = y_i + A_{i,j} * x_j$$