

Capítulo 2

Método de las particiones superpuestas

2.1 Preliminares

En general, se utilizará la siguiente notación para una matriz tridiagonal A de tamaño $n \times n$

$$A = \begin{bmatrix} a_1 & b_1 & & & \\ c_2 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-1} & a_{n-1} & b_{n-1} \\ & & & c_n & a_n \end{bmatrix}. \quad (2.1)$$

Se dice que una matriz tridiagonal A de tamaño $n \times n$ es **estrictamente diagonal dominante** si

$$|a_i| > |b_i| + |c_i|, \text{ para } i = 1, 2, \dots, n; \quad (2.2)$$

donde $c_1 = b_n = 0$.

Si $b_i \neq 0$ y $c_{i+1} \neq 0$, para $i = 1, 2, \dots, n - 1$ la matriz es **irreducible**.

Es conocido que si A es una matriz tridiagonal, estrictamente diagonal dominante, entonces es no singular y se puede realizar la eliminación Gaussiana sin intercambio de filas en cualquier sistema lineal cuya matriz de coeficientes sea A (véase por ejemplo Burden y Faires [21], páginas 366 y 372).

Para una matriz A de tamaño $n \times n$ tridiagonal e irreducible, se define la **diagonal dominanza** δ como

$$\delta = \min_{1 \leq i \leq n} \left\{ \frac{|a_i|}{|b_i| + |c_i|} \right\}. \quad (2.3)$$

Obviamente, si A es estrictamente diagonal dominante, entonces $\delta > 1$ por la expresión (2.2.)

Si \mathbf{x} es la solución de un sistema $A\mathbf{x} = \mathbf{d}$ y se toma otro vector \mathbf{s} como solución del mismo, la cantidad de **error** cometido viene determinada, por ejemplo, por $\|\mathbf{e}\|_\infty$, donde

$$\mathbf{e} = \mathbf{x} - \mathbf{s}.$$

2.2 Descripción del método.

Se considera el problema general de obtener la solución del sistema

$$A\mathbf{x} = \mathbf{d}, \quad (2.4)$$

donde A es una matriz tridiagonal, estrictamente diagonal dominante e irreducible, dada por la expresión (2.1) y

$$\mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}$$

es el vector de términos independientes.

Se supone que existen dos números naturales k y p tales que $k = \frac{n}{p}$ y se considera en A la siguiente partición por bloques

$$A = \begin{bmatrix} A_0 & B_0 & & & \\ C_1 & A_1 & B_1 & & \\ & \ddots & \ddots & \ddots & \\ & & C_{p-2} & A_{p-2} & B_{p-2} \\ & & & C_{p-1} & A_{p-1} \end{bmatrix} \quad (2.5)$$

donde cada uno de los bloques diagonales

$$A_i = \begin{bmatrix} a_{ik+1} & b_{ik+1} & & & \\ c_{ik+2} & a_{ik+2} & b_{ik+2} & & \\ & \ddots & \ddots & \ddots & \\ & & c_{(i+1)k-1} & a_{(i+1)k-1} & b_{(i+1)k-1} \\ & & & c_{(i+1)k} & a_{(i+1)k} \end{bmatrix}, \quad i = 0, 1, \dots, p-1,$$

es una matriz tridiagonal de tamaño $k \times k$ y, para $i = 0, 1, \dots, p-2$, cada bloque subdiagonal

$$C_{i+1} = \left[\begin{array}{cccc|c} 0 & 0 & \dots & 0 & c_{(i+1)k+1} \\ \hline & & & & 0 \\ & O & & & \vdots \\ & & & & 0 \\ & & & & 0 \end{array} \right]$$

y superdiagonal

$$B_i = \left[\begin{array}{c|cccc} 0 & & & & \\ 0 & & O & & \\ \vdots & & & & \\ 0 & & & & \\ \hline b_{(i+1)k} & 0 & \dots & 0 & 0 \end{array} \right]$$

es una matriz de tamaño $k \times k$ con un sólo elemento no nulo.

En los vectores \mathbf{x} y \mathbf{d} se considera una partición por bloques conforme con la realizada en la matriz de coeficientes A , es decir

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{p-2} \\ \mathbf{x}_{p-1} \end{bmatrix} \quad \text{y} \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{p-2} \\ \mathbf{d}_{p-1} \end{bmatrix}, \quad (2.6)$$

con

$$\mathbf{x}_i = \begin{bmatrix} x_{ik+1} \\ x_{ik+2} \\ \vdots \\ x_{(i+1)k-1} \\ x_{(i+1)k} \end{bmatrix} \quad \text{y} \quad \mathbf{d}_i = \begin{bmatrix} d_{ik+1} \\ d_{ik+2} \\ \vdots \\ d_{(i+1)k-1} \\ d_{(i+1)k} \end{bmatrix}, \quad \text{para } i = 0, 1, \dots, p-1.$$

En consecuencia, el sistema (2.4) se puede escribir como

$$\begin{bmatrix} A_0 & B_0 & & & & \\ C_1 & A_1 & B_1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & C_{p-2} & A_{p-2} & B_{p-2} & \\ & & & C_{p-1} & A_{p-1} & \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{p-2} \\ \mathbf{x}_{p-1} \end{bmatrix} = \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{p-2} \\ \mathbf{d}_{p-1} \end{bmatrix}. \quad (2.7)$$

Si se denota por \mathbf{s}_i la solución del sistema

$$A_i \mathbf{x}_i = \mathbf{d}_i, \quad i = 0, 1, \dots, p-1, \quad (2.8)$$

se obtiene el vector

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_{p-2} \\ \mathbf{s}_{p-1} \end{bmatrix},$$

que no coincide exactamente con el vector solución del sistema (2.4), ya que en su obtención se ha omitido alguna información del sistema (los bloques subdiagonales y superdiagonales); aunque se pueden utilizar las diferentes soluciones de los subsistemas independientes (2.8) para formar la solución del sistema (2.4).

Cada una de estas soluciones \mathbf{s}_i , para $i = 0, 1, \dots, p-1$, contiene una cierta cantidad de error. Como se puede observar en el siguiente ejemplo, dependiendo del valor particular de la diagonal dominanza de A y del tamaño de los bloques, este error se puede hacer muy pequeño en las componentes centrales de \mathbf{s}_i .

Ejemplo 2.1 En el sistema (2.4), se considera $n = 24$, $a_i = 30$, para $i = 1, 2, \dots, n$, $b_i = -c_{i+1} = 1$, para $i = 1, 2, \dots, n-1$ y $d_i = c_i + a_i + b_i$, para $i = 1, 2, \dots, n$ (con $b_n = c_1 = 0$).

Por la forma en que se ha construido, resulta inmediato que la solución \mathbf{x} del sistema tiene todas sus componentes iguales a 1.

Para $p = 4$ (y por tanto $k = 6$) el sistema se puede expresar como

$$\begin{bmatrix} A_0 & B_0 & & & & \\ C_1 & A_1 & B_1 & & & \\ & C_2 & A_2 & B_2 & & \\ & & C_3 & A_3 & & \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \end{bmatrix}.$$

El sistema $A_0 \mathbf{x}_0 = \mathbf{d}_0$, esto es

$$\begin{bmatrix} 30 & 1 & 0 & 0 & 0 & 0 \\ -1 & 30 & 1 & 0 & 0 & 0 \\ 0 & -1 & 30 & 1 & 0 & 0 \\ 0 & 0 & -1 & 30 & 1 & 0 \\ 0 & 0 & 0 & -1 & 30 & 1 \\ 0 & 0 & 0 & 0 & -1 & 30 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 31 \\ 30 \\ 30 \\ 30 \\ 30 \\ 30 \end{bmatrix},$$

tiene solución

$$\mathbf{s}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0.9989 \\ 1.0333 \end{bmatrix}.$$

Los sistemas $A_1 \mathbf{x}_1 = \mathbf{d}_1$ y $A_2 \mathbf{x}_2 = \mathbf{d}_2$ tienen la misma matriz de coeficientes

$$A_1 = A_2 = \begin{bmatrix} 30 & 1 & 0 & 0 & 0 & 0 \\ -1 & 30 & 1 & 0 & 0 & 0 \\ 0 & -1 & 30 & 1 & 0 & 0 \\ 0 & 0 & -1 & 30 & 1 & 0 \\ 0 & 0 & 0 & -1 & 30 & 1 \\ 0 & 0 & 0 & 0 & -1 & 30 \end{bmatrix}$$

y el mismo vector de términos independientes

$$\mathbf{d}_1 = \mathbf{d}_2 = \begin{bmatrix} 30 \\ 30 \\ 30 \\ 30 \\ 30 \\ 30 \end{bmatrix},$$

por tanto sus vectores solución coinciden

$$\mathbf{s}_1 = \mathbf{s}_2 = \begin{bmatrix} 0.9667 \\ 0.9989 \\ 1 \\ 1 \\ 0.9989 \\ 1.0333 \end{bmatrix}.$$

El sistema $A_3 \mathbf{x}_3 = \mathbf{d}_3$, o sea

$$\begin{bmatrix} 30 & 1 & 0 & 0 & 0 & 0 \\ -1 & 30 & 1 & 0 & 0 & 0 \\ 0 & -1 & 30 & 1 & 0 & 0 \\ 0 & 0 & -1 & 30 & 1 & 0 \\ 0 & 0 & 0 & -1 & 30 & 1 \\ 0 & 0 & 0 & 0 & -1 & 30 \end{bmatrix} \begin{bmatrix} x_{19} \\ x_{20} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \end{bmatrix} = \begin{bmatrix} 30 \\ 30 \\ 30 \\ 30 \\ 30 \\ 29 \end{bmatrix},$$

tiene solución

$$\mathbf{s}_3 = \begin{bmatrix} 0.9667 \\ 0.9989 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Si se toma

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{bmatrix}$$

como solución del sistema $A\mathbf{x} = \mathbf{d}$, para $i = 1, 2, 3$ en las componentes s_{6i}, s_{6i+1} se produce un error igual a 3.33×10^{-2} , en s_{6i-1}, s_{6i+2} el error que se comete es 0.11×10^{-2} (menor) y en el resto de componentes no se produce error.

En cambio, si en el sistema (2.4) se considera $a_i = 20$ para $i = 1, 2, \dots, 24$, $b_i = -c_{i+1} = 1$, para $i = 1, 2, \dots, 23$ y $d_i = c_i + a_i + b_i$, para $i = 1, 2, \dots, 24$ (con $b_{24} = c_1 = 0$) se obtienen las siguientes soluciones de los sistemas $A_i \mathbf{x}_i = \mathbf{d}_i$, para

$i = 0, 1, 2, 3;$

$$\mathbf{s}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1.0001 \\ 0.9975 \\ 1.0499 \end{bmatrix}, \quad \mathbf{s}_1 = \mathbf{s}_2 = \begin{bmatrix} 0.9501 \\ 0.9975 \\ 0.9999 \\ 1.0001 \\ 0.9975 \\ 1.0499 \end{bmatrix}, \quad \mathbf{s}_3 = \begin{bmatrix} 0.9501 \\ 0.9975 \\ 0.9999 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

En esta ocasión, el error se produce en las tres últimas componentes de \mathbf{s}_0 , en las tres primeras de \mathbf{s}_3 y en todas las componentes de los vectores \mathbf{s}_1 y \mathbf{s}_2 .

Por último, si en el sistema (2.4) se considera $a_i = 4$, para $i = 1, 2, \dots, 24$, y el resto de valores como en los dos casos anteriores, se obtienen los siguientes vectores solución de los sistemas $A_i \mathbf{x}_i = \mathbf{d}_i$, para $i = 0, 1, 2, 3$;

$$\mathbf{s}_0 = \begin{bmatrix} 0.9998 \\ 1.0007 \\ 0.9969 \\ 1.0132 \\ 0.9443 \\ 1.2361 \end{bmatrix}, \quad \mathbf{s}_1 = \mathbf{s}_2 = \begin{bmatrix} 0.7637 \\ 0.9450 \\ 0.9837 \\ 1.0101 \\ 0.9435 \\ 1.2359 \end{bmatrix}, \quad \mathbf{s}_3 = \begin{bmatrix} 0.7639 \\ 0.9443 \\ 0.9868 \\ 0.9969 \\ 0.9993 \\ 0.9998 \end{bmatrix}.$$

En este caso, se comete error en todas las componentes de los vectores \mathbf{s}_i , para $i = 0, 1, 2, 3$.

En los tres casos el tamaño de los bloques de la partición no ha variado, en cambio sí ha variado la diagonal dominante que en el primer caso es igual a 15, en el segundo 10 y en el último 2.

Si ahora se considera en el primer caso $p = 6$ (y por tanto $k = 4$), la diagonal dominante es la misma pero el tamaño de los bloques es distinto. El sistema (2.4) se

puede expresar como

$$\begin{bmatrix} A_0 & B_0 & & & & \\ C_1 & A_1 & B_1 & & & \\ & C_2 & A_2 & B_2 & & \\ & & C_3 & A_3 & B_3 & \\ & & & C_4 & A_4 & B_4 \\ & & & & C_5 & A_5 \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \\ \mathbf{x}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \mathbf{d}_2 \\ \mathbf{d}_3 \\ \mathbf{d}_4 \\ \mathbf{d}_5 \end{bmatrix}.$$

Se obtienen los siguientes vectores solución de los sistemas $A_i \mathbf{x}_i = \mathbf{d}_i$, para $i = 0, 1, \dots, 5$;

$$\mathbf{s}_0 = \begin{bmatrix} 1 \\ 1 \\ 0.9989 \\ 1.0333 \end{bmatrix}, \quad \mathbf{s}_1 = \mathbf{s}_2 = \mathbf{s}_3 = \mathbf{s}_4 = \begin{bmatrix} 0.9667 \\ 0.9989 \\ 0.9989 \\ 1.0333 \end{bmatrix}, \quad \mathbf{s}_5 = \begin{bmatrix} 0.9667 \\ 0.9989 \\ 1 \\ 1 \end{bmatrix}.$$

Ahora, el error se produce en las dos últimas componentes de \mathbf{s}_0 , en las dos primeras de \mathbf{s}_5 y en todas las componentes de las vectores centrales $\mathbf{s}_i, i = 1, 2, 3, 4$. ■

Para minimizar y acotar el error, el método OPM (*Overlapped Partitions Method*) propone considerar una nueva partición (véase Larriba, Jorba, y Navarro [72]) basada en la partición natural dada en la expresión (2.7), añadiendo $2m$ filas (respectivamente, componentes) a cada uno de los bloques diagonales de la matriz de coeficientes (respectivamente, vector de términos independientes) y m filas (respectivamente, componentes) al primer y último bloque de la matriz de coeficientes (respectivamente, vector de términos independientes). Como consecuencia, cada uno de los nuevos bloques está solapado con sus adyacentes (véase la figura 2.1) y se obtiene un nuevo conjunto de subsistemas

$$\hat{A}_i \hat{\mathbf{x}}_i = \hat{\mathbf{d}}_i, \quad i = 0, 1, \dots, p-1, \quad (2.9)$$

donde

- \hat{A}_0 es la submatriz de A de tamaño $(k+m) \times (k+m)$ formada por las filas y columnas $1, 2, \dots, k+m$ y $\hat{\mathbf{d}}_0$ es el vector formado por las componentes $1, 2, \dots, k+m$ de \mathbf{d} ;

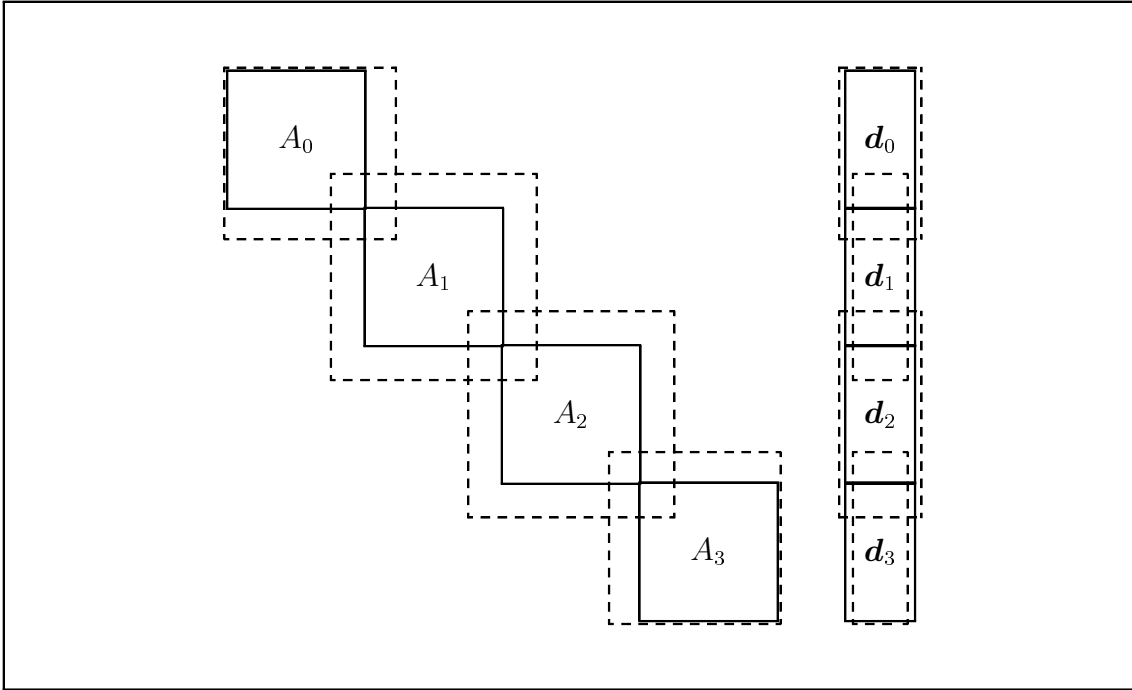


Figura 2.1: Partición natural y superpuesta de A y \mathbf{d} para $p = 4$.

- \hat{A}_i , para $i = 1, 2, \dots, p - 2$, es la submatriz de A de tamaño $(k + 2m) \times (k + 2m)$ formada por las filas y columnas $ik - m + 1, ik - m + 2, \dots, (i + 1)k + m$ y $\hat{\mathbf{d}}_i$ es el vector formado por las componentes $ik - m + 1, ik - m + 2, \dots, (i + 1)k + m$ de \mathbf{d} ;
- \hat{A}_{p-1} es la submatriz de A de tamaño $(k + m) \times (k + m)$ formada por las filas y columnas $j = n - k - m + 1, n - k - m + 2, \dots, n$ y, finalmente, $\hat{\mathbf{d}}_{p-1}$ es el vector formado por las componentes $n - k - m + 1, n - k - m + 2, \dots, n$ de \mathbf{d} .

En la figura 2.1 se muestran, para $p = 4$, los bloques A_i , \mathbf{d}_i marcados con trazo continuo y los bloques \hat{A}_i , $\hat{\mathbf{d}}_i$ con trazo discontinuo.

Para obtener la solución del sistema (2.4), se resuelven los subsistemas (2.9) por eliminación Gaussiana (no hay que realizar intercambio de filas por ser la matriz A estrictamente diagonal dominante), de cada subsistema intermedio se eligen las k componentes centrales del vector solución, del primer subsistema se toman las k primeras y del último las k últimas. En la figura 2.2 se muestra, para $p = 4$, la formación de la solución del sistema (2.4) a partir de las soluciones de los subsistemas (2.9).

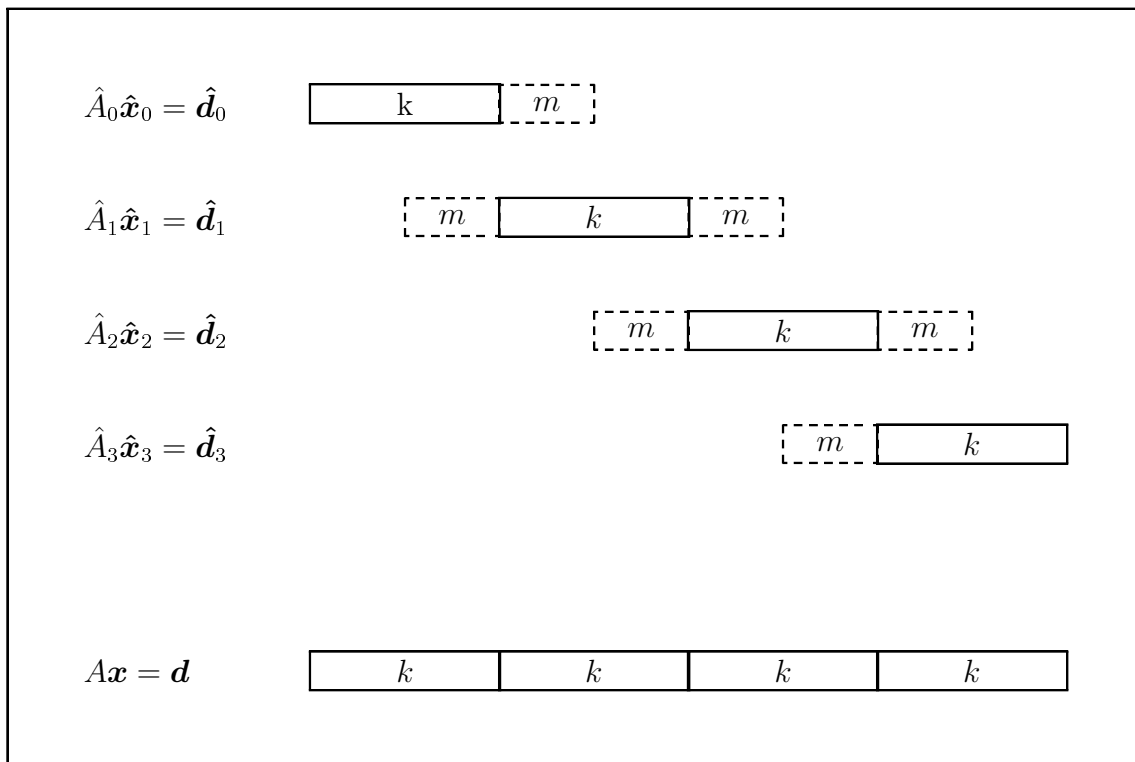


Figura 2.2: Obtención de la solución general a partir de las soluciones parciales para $p = 4$.

2.3 Precisión del método

En la sección 2.2 no se ha especificado nada acerca de la forma de calcular m , que representa el número de ecuaciones superpuestas entre dos bloques contiguos. En esta sección, el parámetro m se obtiene como una función de la diagonal dominanza δ de la matriz de coeficientes y el máximo error permitido ϵ .

Teorema 2.1 Si \mathbf{x} es la solución del sistema (2.4) se tiene

$$\|\mathbf{x}\|_{\infty} \leq \frac{\max_{1 \leq i \leq n} \left\{ \left| \frac{d_i}{a_i} \right| \right\}}{1 - \delta^{-1}}, \quad (2.10)$$

con δ la diagonal dominanza de la matriz A .

Demostración: La matriz A se puede descomponer como $A = D + B$, donde

$$D = \begin{bmatrix} a_1 & & & & \\ & a_2 & & & \\ & & \ddots & & \\ & & & a_{n-1} & \\ & & & & a_n \end{bmatrix} \quad \text{y} \quad B = \begin{bmatrix} 0 & b_1 & & & \\ c_2 & 0 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & c_{n-1} & 0 & b_{n-1} \\ & & & & c_n & 0 \end{bmatrix}.$$

El hecho de que la matriz A sea estrictamente diagonal dominante implica que es no singular por lo que D también lo es, con

$$D^{-1} = \begin{bmatrix} \frac{1}{a_1} & & & & \\ & \frac{1}{a_2} & & & \\ & & \ddots & & \\ & & & \frac{1}{a_{n-1}} & \\ & & & & \frac{1}{a_n} \end{bmatrix},$$

de manera que A se puede escribir como

$$A = D(I + D^{-1}B)$$

y así

$$\mathbf{x} = A^{-1}\mathbf{d} = (I + D^{-1}B)^{-1}D^{-1}\mathbf{d},$$

con lo que

$$\|\mathbf{x}\|_{\infty} \leq \|(I + D^{-1}B)^{-1}\|_{\infty} \|D^{-1}\mathbf{d}\|_{\infty}. \quad (2.11)$$

Ahora bien

$$\|(I + D^{-1}B)^{-1}\|_{\infty} \leq \frac{1}{1 - \|D^{-1}B\|_{\infty}} = \frac{1}{1 - \delta^{-1}} \quad (2.12)$$

ya que

$$\begin{aligned}\|D^{-1}B\|_{\infty} &= \max_{1 \leq i \leq n} \left\{ \frac{|b_i| + |c_i|}{|a_i|} \right\} \\ &= \frac{1}{\min_{1 \leq i \leq n} \left\{ \frac{|a_i|}{|b_i| + |c_i|} \right\}} \\ &= \delta^{-1},\end{aligned}$$

mientras que

$$\|D^{-1}\mathbf{d}\|_{\infty} = \max_{1 \leq i \leq n} \left\{ \left| \frac{d_i}{a_i} \right| \right\}. \quad (2.13)$$

Por tanto, de las expresiones (2.11), (2.12) y (2.13) se obtiene la desigualdad (2.10). ■

Lema 2.1 Si $D = \text{diag}(A)$ y

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

es la solución del sistema (2.4), se tiene

$$|x_i| \leq \frac{\delta^{-h+1}}{1 - \delta^{-2}} \|D^{-1}\mathbf{d}\|_{\infty}, \text{ para } i = 1, 2, \dots, n;$$

donde

- (i) $h = i$, si $d_2 = d_3 = \dots = d_n = 0$.
- (ii) $h = n - i + 1$, si $d_1 = d_2 = \dots = d_{n-1} = 0$.
- (iii) $h = \min\{i, n - i + 1\}$, si $d_2 = d_3 = \dots = d_{n-1} = 0$.

Demostración: De la expresión (2.13) se tiene

$$\mathbf{x} = (D^{-1}\mathbf{d}) - N(D^{-1}\mathbf{d}) + N^2(D^{-1}\mathbf{d}) - N^3(D^{-1}\mathbf{d}) + \dots$$

Debido a la especial estructura de la matriz N , se tiene que el primer término de la serie (2.3) que afecta a la componente x_i está en $N^{i-1}(D^{-1}\mathbf{d})$. Nótese que los términos $N^i(D^{-1}\mathbf{d})$, $N^{i+2}(D^{-1}\mathbf{d})$, \dots no contribuyen en el cálculo de la componente x_i , mientras que los términos $N^{i+1}(D^{-1}\mathbf{d})$, $N^{i+3}(D^{-1}\mathbf{d})$, \dots sí. Como consecuencia

$$\begin{aligned}
|x_i| &\leq \|N^{i-1}(D^{-1}\mathbf{d})\|_\infty + \|N^{i+1}(D^{-1}\mathbf{d})\|_\infty + \|N^{i+3}(D^{-1}\mathbf{d})\|_\infty + \dots \\
&\leq \|N\|_\infty^{i-1} \|D^{-1}\mathbf{d}\|_\infty + \|N\|_\infty^{i+1} \|D^{-1}\mathbf{d}\|_\infty + \|N\|_\infty^{i+3} \|D^{-1}\mathbf{d}\|_\infty + \dots \\
&= \left[(\delta^{-1})^{i-1} + (\delta^{-1})^{i+1} + (\delta^{-1})^{i+3} + \dots \right] \|D^{-1}\mathbf{d}\|_\infty \\
&= \delta^{-i+1} (1 + \delta^{-2} + \delta^{-4} + \dots) \|D^{-1}\mathbf{d}\|_\infty \\
&= \frac{\delta^{-i+1}}{1 - \delta^{-2}} \|D^{-1}\mathbf{d}\|_\infty.
\end{aligned}$$

(ii) En este segundo caso, los productos tienen la siguiente forma

$$D^{-1}\mathbf{d} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ * \end{bmatrix}$$

$$N(D^{-1}\mathbf{d}) = \begin{bmatrix} 0 & * & & & & & \\ \ddots & \ddots & \ddots & & & & \\ & * & 0 & * & & & \\ & & * & 0 & * & & \\ & & & * & 0 & * & \\ & & & & * & 0 & * \\ & & & & & * & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ * \\ 0 \end{bmatrix}$$

$$N^2(D^{-1}\mathbf{d}) = \begin{bmatrix} 0 & * & & & \\ \ddots & \ddots & \ddots & & \\ & * & 0 & * & \\ & & * & 0 & * \\ & & & * & 0 & * \\ & & & & * & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ * \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ * \\ 0 \\ * \end{bmatrix}$$

$$N^3(D^{-1}\mathbf{d}) = \begin{bmatrix} 0 & * & & & \\ \ddots & \ddots & \ddots & & \\ & * & 0 & * & \\ & & * & 0 & * \\ & & & * & 0 & * \\ & & & & * & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ * \\ 0 \\ * \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ * \\ 0 \\ * \\ 0 \end{bmatrix}$$

El primer término de la serie (2.3) que afecta a la componente x_i está en $N^{n-i}(D^{-1}\mathbf{d})$. Los términos $N^{n-i+1}(D^{-1}\mathbf{d})$, $N^{n-i+3}(D^{-1}\mathbf{d})$, \dots no influyen en el cálculo de x_i , en cambio los términos $N^{n-i+2}(D^{-1}\mathbf{d})$, $N^{n-i+4}(D^{-1}\mathbf{d})$, \dots sí lo hacen. En consecuencia

$$\begin{aligned} |x_i| &\leq \|N^{n-i}(D^{-1}\mathbf{d})\|_\infty + \|N^{n-i+2}(D^{-1}\mathbf{d})\|_\infty + \|N^{n-i+4}(D^{-1}\mathbf{d})\|_\infty + \dots \\ &\leq \|N\|_\infty^{n-i} \|D^{-1}\mathbf{d}\|_\infty + \|N\|_\infty^{n-i+2} \|D^{-1}\mathbf{d}\|_\infty + \|N\|_\infty^{n-i+4} \|D^{-1}\mathbf{d}\|_\infty + \dots \\ &= \left[(\delta^{-1})^{n-i} + (\delta^{-1})^{n-i+2} + (\delta^{-1})^{n-i+4} + \dots \right] \|D^{-1}\mathbf{d}\|_\infty \\ &= \delta^{i-n} (1 + \delta^{-2} + \delta^{-4} + \dots) \|D^{-1}\mathbf{d}\|_\infty \\ &= \frac{\delta^{i-n}}{1 - \delta^{-2}} \|D^{-1}\mathbf{d}\|_\infty \end{aligned}$$

(iii) Ahora, los productos tienen la forma

(respectivamente, vector de términos independientes) y m filas (respectivamente, componentes) al primer y último bloque de la matriz de coeficientes (respectivamente, vector de términos independientes). Como consecuencia, cada uno de los nuevos bloques está solapado con sus adyacentes y se obtiene el nuevo conjunto de subsistemas dado en la expresión (2.9).

Para simplificar, se redefine cualquier subsistema $\hat{A}_i \hat{\mathbf{x}}_i = \hat{\mathbf{d}}_i$ como $\hat{A} \hat{\mathbf{x}} = \hat{\mathbf{d}}$ y los índices del mismo desde 1 hasta r . La situación del subsistema $\hat{A} \hat{\mathbf{x}} = \hat{\mathbf{d}}$ en el sistema general tiene la siguiente forma

$$\begin{array}{rcccccc} \tilde{c}_1 \tilde{x}_0 & + & \hat{a}_1 \hat{x}_1 & + & \hat{b}_1 \hat{x}_2 & & = & \hat{d}_1 \\ & & \hat{c}_2 \hat{x}_1 & + & \hat{a}_2 \hat{x}_2 & + & \hat{b}_2 \hat{x}_3 & = & \hat{d}_2 \\ & & \ddots & & \ddots & & \ddots & \vdots \\ & & & & \hat{c}_{r-1} \hat{x}_{r-2} & + & \hat{a}_{r-1} \hat{x}_{r-1} & + & \hat{b}_{r-1} \hat{x}_r & = & \hat{d}_{r-1} \\ & & & & & & \hat{c}_r \hat{x}_{r-1} & + & \hat{a}_r \hat{x}_r & + & \tilde{b}_r \tilde{x}_{r+1} & = & \hat{d}_r \end{array}$$

donde \tilde{c}_1 y \tilde{b}_r son los elementos omitidos del sistema general al resolver el subsistema $\hat{A} \hat{\mathbf{x}} = \hat{\mathbf{d}}$. Nótese que $\tilde{c}_1 = 0$ para el primer subsistema y $\tilde{b}_r = 0$ para el último.

Si \mathbf{x} es la solución del sistema general, se representa por \mathbf{x}' a la restricción de \mathbf{x} a las variables del subsistema $\hat{A} \hat{\mathbf{x}} = \hat{\mathbf{d}}$. Se puede escribir entonces

$$\mathbf{x} = \begin{bmatrix} \vdots \\ \mathbf{x}' \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \hline x'_0 \\ x'_1 \\ \vdots \\ x'_r \\ \hline x'_{r+1} \\ \vdots \end{bmatrix},$$

verificándose

$$\begin{array}{l} \tilde{c}_1 x'_0 + \hat{a}_1 x'_1 + \hat{b}_1 x'_2 = \hat{d}_1 \\ \hat{c}_r x'_{r-1} + \hat{a}_r x'_r + \tilde{b}_r x'_{r+1} = \hat{d}_r. \end{array}$$

El error que se comete al tomar \mathbf{x}' como solución del subsistema $\hat{A}\hat{\mathbf{x}} = \hat{\mathbf{d}}$ viene determinado por

$$\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}',$$

siendo $\hat{\mathbf{x}}$ el vector solución del mismo. Como

$$\hat{A}\hat{\mathbf{x}} = \begin{bmatrix} \hat{d}_1 \\ \hat{d}_2 \\ \vdots \\ \hat{d}_{r-1} \\ \hat{d}_r \end{bmatrix}$$

y

$$\hat{A}\mathbf{x}' = \begin{bmatrix} \hat{a}_1 x'_1 + \hat{b}_1 x'_2 \\ \hat{d}_2 \\ \vdots \\ \hat{d}_{r-1} \\ \hat{c}_r x'_{r-1} + \hat{a}_r x'_r \end{bmatrix} = \begin{bmatrix} \hat{d}_1 - \tilde{c}_1 x'_0 \\ \hat{d}_2 \\ \vdots \\ \hat{d}_{r-1} \\ \hat{d}_r - \tilde{b}_r x'_{r+1} \end{bmatrix},$$

se tiene

$$\hat{A}\hat{\mathbf{x}} - \hat{A}\mathbf{x}' = \begin{bmatrix} \tilde{c}_1 x'_0 \\ 0 \\ \vdots \\ 0 \\ \tilde{b}_r x'_{r+1} \end{bmatrix} = \mathbf{c},$$

por lo que

$$\hat{A}\mathbf{e} = \mathbf{c}.$$

Aplicando el lema 2.1 al sistema anterior, se tiene la siguiente acotación para la i -ésima componente de \mathbf{e}

$$|e_i| \leq \frac{\delta^{-h+1}}{1 - \delta^{-2}} \|D^{-1}\mathbf{c}\|_\infty, \text{ para } i = 1, 2, \dots, n;$$

donde

$$h = \begin{cases} i, & \text{en el último subsistema,} \\ r - i + 1, & \text{en el primer subsistema,} \\ \min\{i, r - i + 1\}, & \text{en los subsistemas intermedios} \end{cases}$$

y $D = \text{diag}(\hat{A})$.

Ahora bien, para el primer subsistema

$$\frac{|\hat{a}_r|}{|\tilde{b}_r|} \geq \min_{1 \leq i \leq n} \left\{ \frac{|a_i|}{|b_i| + |c_i|} \right\} = \delta$$

con lo que

$$\|D^{-1}\mathbf{c}\|_{\infty} = \frac{|\tilde{b}_r| |x'_{r+1}|}{|\hat{a}_r|} \leq \delta^{-1} |x'_{r+1}|.$$

Para el último subsistema

$$\frac{|\hat{a}_1|}{|\tilde{c}_1|} \geq \delta$$

con lo que

$$\|D^{-1}\mathbf{c}\|_{\infty} = \frac{|\tilde{c}_1| |x'_0|}{|\hat{a}_1|} \leq \delta^{-1} |x'_0|.$$

Finalmente, para los subsistemas intermedios

$$\frac{|\hat{a}_1|}{|\tilde{c}_1|} \geq \delta \quad \text{y} \quad \frac{|\hat{a}_r|}{|\tilde{b}_r|} \geq \delta$$

con lo que

$$\|D^{-1}\mathbf{c}\|_{\infty} = \max \left\{ \frac{|\tilde{c}_1| |x'_0|}{|\hat{a}_1|}, \frac{|\tilde{b}_r| |x'_{r+1}|}{|\hat{a}_r|} \right\} \leq \max \{ \delta^{-1} |x'_0|, \delta^{-1} |x'_{r+1}| \}.$$

En cualquier caso

$$\|D^{-1}\mathbf{c}\|_{\infty} \leq \delta^{-1} \mu$$

donde μ es una cota superior de $\|\mathbf{x}\|_\infty$ y así

$$|e_i| \leq \frac{\delta^{-h}}{1 - \delta^{-2}} \mu, \quad \text{para } i = 1, 2, \dots, r. \quad (2.15)$$

Por otra parte, al ser $\delta > 1$, se cumple que

$$1 > \delta^{-1} > \delta^{-2} > \dots > \delta^{-r}.$$

En el último subsistema, las m primeras componentes de \mathbf{e} provienen de ecuaciones de solapamiento, en las demás se tiene la siguiente acotación, para $j = m + 1, m + 2, \dots, r$,

$$|e_j| \leq \frac{\delta^{-j}}{1 - \delta^{-2}} \mu < \frac{\delta^{-m}}{1 - \delta^{-2}} \mu, \quad (2.16)$$

En el primer subsistema las m últimas ecuaciones son de solapamiento. Las $r - m$ primeras componentes de \mathbf{e} cumplen

$$|e_j| \leq \frac{\delta^{-(r-j+1)}}{1 - \delta^{-2}} \mu < \frac{\delta^{-m}}{1 - \delta^{-2}} \mu, \quad (2.17)$$

con $j = 1, 2, \dots, r - m$.

En los subsistemas intermedios las m primeras y últimas ecuaciones son de solapamiento. Las $r - 2m$ componentes intermedias del vector de error están acotadas como sigue

$$|e_j| \leq \frac{\delta^{-\min\{j, n-j+1\}}}{1 - \delta^{-2}} \mu < \frac{\delta^{-m}}{1 - \delta^{-2}} \mu, \quad (2.18)$$

con $j = m + 1, m + 2, \dots, r - m$.

Si se toma m de manera que cumpla la desigualdad 2.14 entonces

$$m \log \delta^{-1} \leq \log \frac{\epsilon(1 - \delta^{-2})}{\mu},$$

esto es

$$\delta^{-m} \leq \frac{\epsilon(1 - \delta^{-2})}{\mu},$$

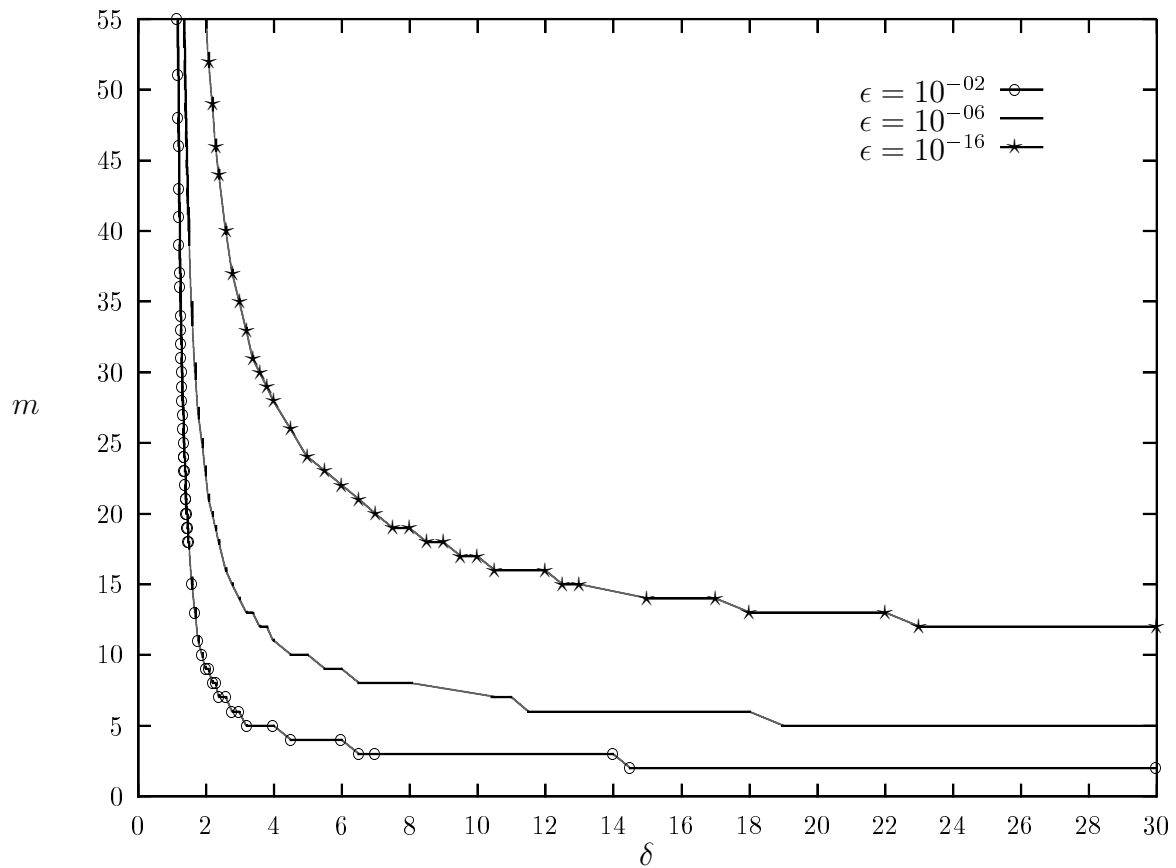


Figura 2.3: Valor de m en función de δ para distintos valores de ϵ .

o sea

$$\frac{\delta^{-m}}{1 - \delta^{-2}} < \epsilon,$$

lo que implica, por las expresiones (2.16), (2.17) y (2.18), que el error cometido es menor que ϵ . ■

Para cada sistema en particular, de las expresiones (2.14) y (2.10) se puede determinar m y obtener, aplicando el método, la solución \mathbf{x} .

En la figura 2.3 se muestra una gráfica de los diferentes valores de m en función de la diagonal dominante δ y el máximo error permitido ϵ , donde μ se ha calculado utilizando

	m				
	5	10	100	500	1000
10^{-02}	3.14899	1.88411	1.09877	1.02438	1.01332
ϵ 10^{-06}	18.23523	4.37983	1.19132	1.04131	1.02172
10^{-16}	1802.18329	42.55320	1.48022	1.08736	1.04409

Tabla 2.1: Rango de la diagonal dominanza δ dependiendo del máximo error tolerado ϵ y del número de ecuaciones solapadas m .

la expresión (2.10) para el caso concreto

$$\max_{1 \leq i \leq n} \left\{ \left| \frac{d_i}{a_i} \right| \right\} = 1.9.$$

Para otros valores de μ se obtienen gráficas con formas semejantes a la de la figura 2.3.

Se observa que para valores de la diagonal dominanza no muy próximos a 1, se obtienen valores de m pequeños (aumentando este valor cuanto menor sea el error ϵ tolerado) y que para valores de δ próximos a 1 el valor de m crece con tendencia a $+\infty$ (para $\delta = 1.01$ y $\epsilon = 10^{-16}$ se obtiene $m = 4626$). Además, diferentes pares (δ, ϵ) producen el mismo valor de m , debido a que este parámetro está determinado por una función entera de variable real.

En la tabla 2.1 se muestra el mínimo valor de δ necesario para no superar el error ϵ , dado un valor de solapamiento m fijo.

2.4 Paralelización del método.

Este método es fácilmente paralelizable pues basta con considerar que p es el número de procesadores (generalmente, los procesadores se representarán por P_i , para $i = 0, 1, \dots, p - 1$) y asignar a cada procesador la solución de uno de los subsistemas dados en la expresión (2.9). El siguiente algoritmo describe esa tarea.

Algoritmo 2.1 Método de las particiones superpuestas en paralelo**Paso 1**

Se calcula δ conforme a la expresión (2.3), se obtiene μ mediante la expresión (2.10) y utilizando la expresión (2.14) se determina m .

Paso 2

A partir de m , se asignan a cada procesador las ecuaciones necesarias para obtener la partición superpuesta.

Paso 3

Cada procesador resuelve el subsistema correspondiente dado por la expresión (2.9).

Paso 4

El vector solución del sistema original se obtiene tomando las k primeras componentes del vector solución de $\hat{A}_0 \hat{x}_0 = \hat{d}_0$, las k componentes centrales de los vectores solución de $\hat{A}_i \hat{x}_i = \hat{d}_i$, para $i = 1, 2, \dots, p-2$, y las k últimas componentes del vector solución de $\hat{A}_{p-1} \hat{x}_{p-1} = \hat{d}_{p-1}$.

Ejemplo 2.2 En el sistema (2.4), se considera $n = 24$, $a_i = 30$, para $i = 1, 2, \dots, n$, $b_i = -c_{i+1} = 1$, para $i = 1, 2, \dots, n-1$ y $d_i = c_i + a_i + b_i$, para $i = 1, 2, \dots, n$ (con $b_n = c_1 = 0$). Por la forma en que se ha construido, resulta inmediato que la solución del sistema tiene todas sus componentes iguales a 1.

La diagonal dominanza δ está dada por

$$\begin{aligned} \delta &= \min_{1 \leq i \leq 24} \left\{ \frac{|a_i|}{|b_i| + |c_i|} \right\} \\ &= \min \left\{ \frac{30}{1}, \frac{30}{1+1}, \dots, \frac{30}{1+1}, \frac{30}{1} \right\} \\ &= 15. \end{aligned}$$

El parámetro μ vale

$$\begin{aligned}
 \mu &= \frac{\max_{1 \leq i \leq 24} \left\{ \left| \frac{d_i}{a_i} \right| \right\}}{1 - \delta^{-1}} \\
 &= \frac{\max \left\{ \frac{31}{30}, \frac{30}{30}, \dots, \frac{29}{30}, \frac{29}{30} \right\}}{1 - 15^{-1}} \\
 &= \frac{31}{28} \\
 &= 1.1071.
 \end{aligned}$$

Si se quiere que el error cometido ϵ sea inferior a 10^{-3} , se debe tomar m de manera que

$$\begin{aligned}
 m &\geq \left\lceil \frac{1}{\log \delta^{-1}} \log \frac{\epsilon(1 - \delta^{-2})}{\mu} \right\rceil \\
 &= \left\lceil \frac{1}{\log 15^{-1}} \log \frac{10^{-3}(1 - 15^{-2})}{\frac{31}{28}} \right\rceil \\
 &= \left\lceil -\frac{1}{\log 15} \left(\log 10^{-3} + \log \frac{1026}{1141} \right) \right\rceil \\
 &= \left\lceil -\frac{1}{\log 15} \left(-3 + \log \frac{1026}{1141} \right) \right\rceil \\
 &= \lceil 2.5901 \rceil,
 \end{aligned}$$

por ejemplo $m = 3$.

El procesador P_0 resuelve el subsistema $\hat{A}_0 \hat{\mathbf{x}}_0 = \hat{\mathbf{d}}_0$, esto es

$$\begin{bmatrix} 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ \hline x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} 31 \\ 30 \\ 30 \\ 30 \\ 30 \\ 30 \\ \hline 30 \\ 30 \\ 30 \end{bmatrix},$$

cuya solución es

$$\hat{\mathbf{x}}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \hline 1 \\ 0.9989 \\ 1.0333 \end{bmatrix}.$$

Cada uno de los procesadores P_1 y P_2 deben resolver, respectivamente, el subsistema

$\hat{A}_1 \hat{x}_1 = \hat{d}_1$ y $\hat{A}_2 \hat{x}_2 = \hat{d}_2$, esto es

$$\begin{bmatrix} 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 \end{bmatrix} \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \hline x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ \hline x_{13} \\ x_{14} \\ x_{15} \end{bmatrix} = \begin{bmatrix} 30 \\ 30 \\ 30 \\ \hline 30 \\ 30 \\ 30 \\ 30 \\ 30 \\ 30 \\ \hline 30 \\ 30 \\ 30 \end{bmatrix}$$

y

$$\begin{bmatrix} 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 \end{bmatrix} \begin{bmatrix} x_{10} \\ x_{11} \\ x_{12} \\ \hline x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \\ x_{17} \\ x_{18} \\ \hline x_{19} \\ x_{20} \\ x_{21} \end{bmatrix} = \begin{bmatrix} 30 \\ 30 \\ 30 \\ \hline 30 \\ 30 \\ 30 \\ 30 \\ 30 \\ 30 \\ \hline 30 \\ 30 \\ 30 \end{bmatrix},$$

respectivamente, cuyas soluciones son

$$\hat{\mathbf{x}}_1 = \hat{\mathbf{x}}_2 = \begin{bmatrix} 0.9667 \\ 0.9989 \\ 1 \\ \hline 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \hline 1 \\ 0.9989 \\ 1.0333 \end{bmatrix}.$$

Por último, el procesador P_3 debe resolver el subsistema $\hat{A}_3 \hat{\mathbf{x}}_3 = \hat{\mathbf{d}}_3$, esto es

$$\begin{bmatrix} 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 30 \end{bmatrix} \begin{bmatrix} x_{16} \\ x_{17} \\ x_{18} \\ \hline x_{19} \\ x_{20} \\ x_{21} \\ x_{22} \\ x_{23} \\ x_{24} \end{bmatrix} = \begin{bmatrix} 30 \\ 30 \\ 30 \\ \hline 30 \\ 30 \\ 30 \\ 30 \\ 30 \\ 29 \end{bmatrix},$$

cuya solución viene dada por

$$\hat{\mathbf{x}}_3 = \begin{bmatrix} 0.9667 \\ 0.99891 \\ 1 \\ \hline 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

La solución del sistema original se obtiene tomando las seis primeras componentes de $\hat{\mathbf{x}}_0$, las seis componentes centrales de $\hat{\mathbf{x}}_1$ y $\hat{\mathbf{x}}_2$ y las seis últimas componentes de $\hat{\mathbf{x}}_3$. ■

2.5 Algoritmos BSP

La implementación del algoritmo 2.1 según el modelo BSP se describe en el algoritmo 2.2. Se supone que tanto la matriz de coeficientes como el vector de términos independientes se encuentran en el procesador principal P_0 .

Algoritmo 2.2 Algoritmo BSP del método OPM para un sistema tridiagonal

Superpaso 1

- El procesador principal P_0 calcula $k = \frac{n}{p}$ y los parámetros δ , μ y m usando las expresiones (2.3), (2.10) y (2.14).
- El procesador P_0 envía al procesador P_i , para $i = 0, 1, \dots, p-1$, el bloque \hat{A}_i y las componentes del vector $\hat{\mathbf{d}}_i$, correspondientes a la partición superpuesta.
- El procesador P_0 envía el valor de m al procesador P_i , para $i = 1, 2, \dots, p-1$.

Superpaso 2

- Para $i = 0, 1, \dots, p - 1$, cada procesador P_i resuelve el subsistema $\hat{A}_i \hat{x}_i = \hat{d}_i$ por eliminación Gaussiana.
- Para $i = 1, 2, \dots, p - 1$, el procesador P_i comunica al procesador P_0 las componentes $m + 1, m + 2, \dots, m + k$ de \hat{x}_i .
- El procesador principal forma la solución x del sistema (2.4), haciendo:
 - Para $j = 1, 2, \dots, k$, la componente j -ésima de x igual a la componente j -ésima de \hat{x}_0 .
 - Para $i = 1, 2, \dots, p - 1$ y $j = 1, 2, \dots, k$, la componente $(ki + j)$ -ésima de x igual a la componente $(m + j)$ -ésima de \hat{x}_i .

Para obtener el coste computacional del algoritmo, se calcula el coste de cada uno de los superpasos individuales.

Coste del superpaso 1. El coste aritmético de este superpaso viene determinado por el cálculo en el procesador principal de los parámetros k , δ , μ y m . La obtención de k requiere una operación, para calcular la diagonal dominante se necesita realizar $2n$ operaciones, para el cálculo de μ —de acuerdo con la expresión (2.10)— se necesitan $n + 3$ operaciones y para calcular m —mediante (2.14)— 9 operaciones aritméticas. Por tanto el coste aritmético de este superpaso es $3n + 13$.

En cuanto al coste de comunicación, el procesador que más comunica es el principal que envía a P_i , para $i = 1, 2, \dots, p - 1$, el valor de m , el bloque \hat{A}_i y las componentes del vector \hat{d}_i ; comunica por tanto $p - 1 + [4(k + 2m) - 2](p - 2) + 4(k + m) - 1$ elementos, por lo que el coste total del superpaso es

$$3n + 13 + (4n - 4k - 12m + 8mp - p + 2)g + l. \quad (2.19)$$

Coste del superpaso 2. Resolver un sistema tridiagonal de r ecuaciones mediante el método de eliminación de Gauss para sistemas tridiagonales requiere un total de $8r - 7$ operaciones, por lo que el coste aritmético del superpaso es $8(k + 2m) - 7$, que corresponde al número de operaciones realizadas por cualquiera de los procesadores intermedios.

Los procesadores P_i , para $i = 1, \dots, p-1$, comunican k componentes de su solución parcial al procesador principal, lo que supone un coste de comunicación $k(p-1)$. En consecuencia, el coste del superpaso 2 es

$$(8k + 16m - 7) + (n - k)g + l. \quad (2.20)$$

Sumando las expresiones (2.19) y (2.20), se obtiene que el coste total del algoritmo es

$$(3n + 8k + 16m + 6) + [5n - 5k - 12m + 8mp - p + 2]g + 2l.$$

El cálculo de los parámetros δ y μ requiere $3n + 3$ operaciones, tarea que puede realizarse en paralelo. Aunque esto conlleva un aumento del número de superpasos y un ligero incremento en el número de comunicaciones, en muchos casos supone una reducción en el coste total del algoritmo.

Para $i = 0, 1, \dots, p-1$, se definen

$$\delta_i = \min_{ik+1 \leq j \leq (i+1)k} \left\{ \frac{|a_j|}{|c_j| + |b_j|} \right\}, \quad (2.21)$$

$$\mu_i = \max_{ik+1 \leq j \leq (i+1)k} \left\{ \frac{|d_j|}{|a_j|} \right\}, \quad (2.22)$$

entonces

$$\delta = \min_{0 \leq i \leq p-1} \{\delta_i\}, \quad (2.23)$$

$$\mu = \frac{\max_{0 \leq i \leq p-1} \{\mu_i\}}{1 - \delta^{-1}}, \quad (2.24)$$

El siguiente algoritmo BSP modifica el algoritmo 2.3 para realizar el cálculo de δ y μ en paralelo.

Algoritmo 2.3 Algoritmo BSP del método OPM para un sistema tridiagonal, con cálculo de δ y μ en paralelo

Superpaso 1

- El procesador principal P_0 calcula $k = \frac{n}{p}$.
- Para $i = 1, 2, \dots, p - 1$, el procesador P_0 envía a P_i los elementos c_{ik+j} , a_{ik+j} , b_{ik+j} y d_{ik+j} , con $j = 1, 2, \dots, k$, excepto $b_n = 0$.

Superpaso 2

- Para $i = 0, 1, \dots, p - 1$, P_i calcula δ_i y μ_i de acuerdo con las expresiones (2.21) y (2.22).
- Para $i = 1, 2, \dots, p - 1$, P_i envía a P_0 los valores de los parámetros δ_i y μ_i .

Superpaso 3

- El procesador P_0 calcula los parámetros δ , μ y m haciendo uso de las expresiones (2.23), (2.24) y (2.14).
- Para $i = 1, 2, \dots, p - 2$, el procesador principal P_0 envía a P_i las filas j -ésima y $(m + k + j)$ -ésima de \hat{A}_i y $\hat{\mathbf{d}}_i$, con $j = 1, 2, \dots, m$.
- El procesador principal envía a P_{p-1} la fila j -ésima de \hat{A}_{p-1} y $\hat{\mathbf{d}}_{p-1}$, con $j = 1, 2, \dots, m$.
- Para $i = 1, 2, \dots, p - 1$, el procesador P_0 envía a P_i el valor de m .

Superpaso 4

Como el superpaso 2 del algoritmo 2.2.

Al repartir el cálculo de δ y μ entre p procesadores, el número de operaciones baja de $3n + 3$ a $3k + 3$ y el coste aritmético de $3n + 8k + 16m + 6$ a $11k + 16m + 6$; se han introducido dos superpasos más, por lo que el coste de sincronización aumenta de $2l$ a $4l$ y

se ha incrementado la comunicación en $2(p-1)$ elementos (ya que desde los procesadores remotos se envía a P_0 los parámetros δ_i y μ_i), en consecuencia, el coste de comunicación vale

$$[5n - 5k - 12m + 8mp - p + 2 + 2(p-1)]g = (5n - 5k - 12m + 8mp + p)g.$$

A continuación se obtendrá con detalle el coste del algoritmo 2.3.

Coste del superpaso 1. La obtención de k requiere una operación. Cada procesador recibe del principal $4k$ elementos, por lo que el coste de comunicación es $[4k(p-1) - 1]g$ y el coste del superpaso

$$1 + (4n - 4k - 1)g + l. \quad (2.25)$$

Coste del superpaso 2. El cálculo de δ_i en cualquier procesador requiere $2k$ operaciones mientras que el de μ_i necesita k operaciones. Cada procesador (excepto P_0) envía al procesador principal los valores de δ_i y μ_i que en total recibe $2(p-1)$ elementos; en consecuencia el coste del superpaso es

$$3k + (2p - 2)g + l. \quad (2.26)$$

Coste del superpaso 3. El procesador principal debe realizar 3 operaciones para calcular μ y 9 para obtener el valor de m por lo que el coste aritmético es 12. En este superpaso el procesador que más comunica es el principal que transmite m a todos los procesadores, envía a P_{p-1} la cantidad de $4m - 1$ elementos y a P_i , para $i = 1, 2, \dots, p-2$, un total de $8m - 2$ elementos; o sea, comunica $p - 1 + 4m - 1 + (8m - 2)(p - 2)$ elementos. Se tiene por tanto el siguiente coste para el superpaso 3

$$12 + (-12m + 8mp - p + 2)g + l. \quad (2.27)$$

Coste del superpaso 4. Puesto que este superpaso coincide con el superpaso 2 del algoritmo 2.2, su coste viene dado por la expresión (2.20)

La suma de las expresiones (2.25), (2.26), (2.27) y (2.20) da como resultado el coste global del algoritmo 2.3

$$(11k + 16m + 6) + (5n - 5k - 12m + 8mp + p)g + 4l. \quad (2.28)$$

s	p	l	g	$n_{\frac{1}{2}}$
45	1	423	2.3	26
	2	3294	9.5	25
	4	5366	12.4	25
	6	8164	12.5	25

(a) *IBM SP2 switch*

s	p	l	g	$n_{\frac{1}{2}}$
45	1	423	2.3	8
	2	20235	709.7	3
	4	54163	1362.6	9
	6	121958	3211.2	9

(b) *IBM SP2 ethernet*

s	p	l	g	$n_{\frac{1}{2}}$
16.4	1	23	0.2	22
	2	2556	6.9	5
	4	5152	7.4	4
	6	7538	6.8	4

(c) *Cluster de PC's***Tabla 2.2:** *Valores de parámetros BSP.*

2.6 Resultados numéricos

En esta sección se analizan los tiempos previstos teóricamente y los tiempos experimentales de los algoritmos 2.2 y 2.3 (a los que se referenciará en las tablas y figuras como OPM1 y OPM2 respectivamente). Las pruebas experimentales se han realizado en el IBM SP2 y el *cluster* de PC's cuyas características se han descrito en la subsección 1.5.3. Por comodidad, en la tabla 2.2 se repiten los parámetros obtenidos para esas máquinas que se muestran en la tabla 1.1.

El tiempo teórico se ha obtenido considerando que el tamaño de bloque de los mensajes que se comunican entre los procesadores es $k = \frac{n}{p}$ y que el coste de comunicación de una

palabra de 32 bits es

$$g(k) = \left(\frac{n_{\frac{1}{2}}}{k} + 1 \right) g_{\infty},$$

además se ha tomado $m = 5$. Los algoritmos 2.2 (OPM1) y 2.3 (OPM2) han sido implementados en Fortran usando la versión v1.3 de la librería BSPLib, se ha generado el sistema (2.4) obteniendo aleatoriamente los elementos de la matriz de coeficientes A y, para simplificar, eligiendo el vector de términos independientes \mathbf{d} de manera que la solución sea $\mathbf{x} = [1, 1, \dots, 1]^T$. A la elección aleatoria de los coeficientes se le ha añadido la restricción de que $m = 5$ (véase la tabla 2.1). Si en el ejemplo 2.2 se toma $\epsilon = 10^{-5}$, se obtiene de manera sencilla un sistema tridiagonal en el que $m = 5$.

En general, para medir cualquier tiempo se ha adoptado la estrategia que consiste en realizar un cierto número de veces el mismo experimento y quedarse con el valor mínimo, repitiéndose esta rutina en diversos días y horas, siempre con uso exclusivo de las máquinas (o los nodos necesarios en el caso del IBM SP2) y tomando como valor definitivo el mínimo de todos ellos. Esta estrategia, si es exhaustiva, permite medir el mejor rendimiento posible.

En las tablas 2.3, 2.4 y 2.5 y figuras 2.4, 2.5 y 2.6 se muestran los tiempos teóricos y experimentales, medidos en segundos, de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en el IBM SP2 para 2, 4 y 6 procesadores utilizando *switch*, mientras que en las tablas 2.6, 2.7 y 2.8 y figuras 2.7, 2.8 y 2.9 se muestran los tiempos teóricos y experimentales, medidos en segundos, de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en el IBM SP2 para 2, 4 y 6 procesadores utilizando *ethernet*. Estos tiempos han sido obtenidos para diferentes tamaños de la matriz de coeficientes en un rango que varía desde 128 a 524288 para 2 y 4 procesadores y desde 126 a 516096 para 6 procesadores. En las tablas 2.9, 2.10 y 2.11 y figuras 2.10, 2.11 y 2.12 se muestran los tiempos teóricos y experimentales, medidos en segundos, de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en el *cluster* de PC's para 2, 4 y 6 procesadores, en este caso el tamaño de la matriz de coeficientes varía desde 128 a 65536 para 2 y 4 procesadores y desde 126 a 64512 para 6 procesadores.

IBM SP2 2 procesadores <i>switch</i>				
n	OPM1		OPM2	
	Teórico	Experimental	Teórico	Experimental
128	0.0002	0.0002	0.0004	0.0004
256	0.0003	0.0006	0.0004	0.0009
512	0.0004	0.0009	0.0005	0.0011
1024	0.0006	0.0006	0.0007	0.0009
2048	0.0010	0.0011	0.0011	0.0011
4096	0.0019	0.0019	0.0019	0.0020
8192	0.0036	0.0034	0.0035	0.0036
16384	0.0070	0.0068	0.0066	0.0063
32768	0.0139	0.0144	0.0129	0.0123
65536	0.0276	0.0268	0.0255	0.0243
131072	0.0550	0.0527	0.0507	0.0484
262144	0.1098	0.1062	0.1012	0.0974
524288	0.2193	0.2187	0.2020	0.2012

Tabla 2.3: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un IBM SP2 con 2 procesadores interconectados mediante switch, para $128 \leq n \leq 524288$.*

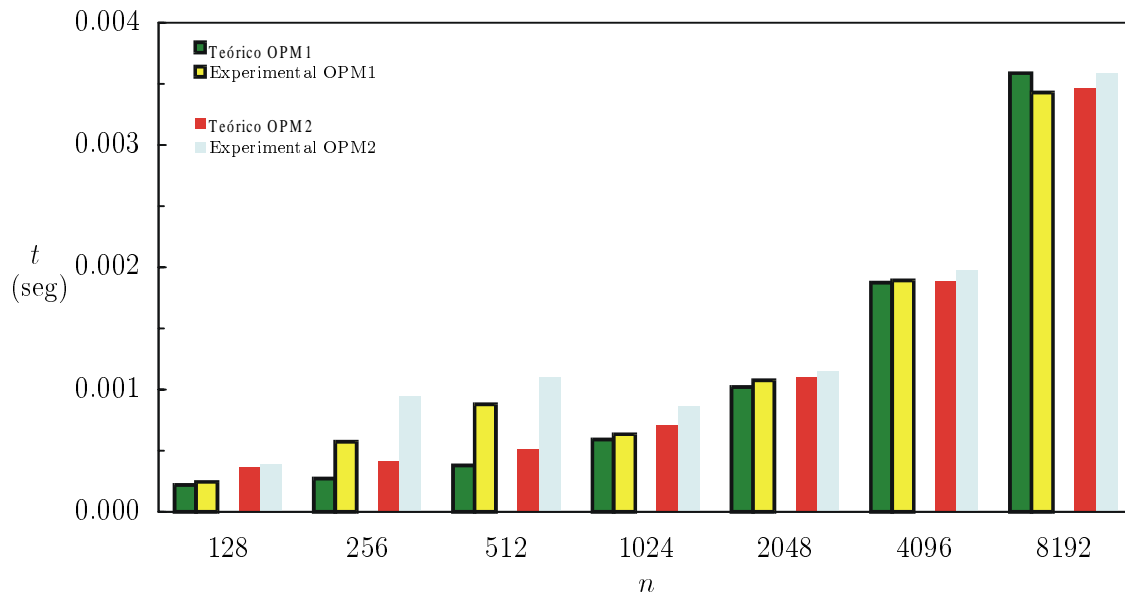
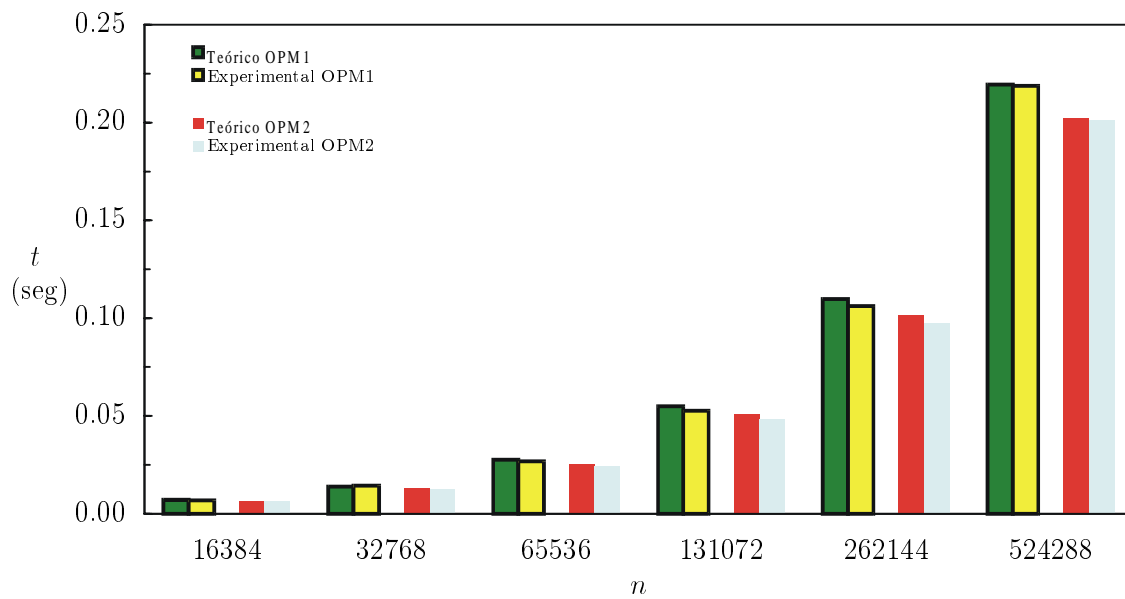
(a) $128 \leq n \leq 8192$ (b) $16384 \leq n \leq 524288$

Figura 2.4: Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 con 2 procesadores interconectados mediante switch.

IBM SP2 4 procesadores <i>switch</i>				
n	OPM1		OPM2	
	Teórico	Experimental	Teórico	Experimental
128	0.0004	0.0005	0.0006	0.0007
256	0.0005	0.0005	0.0007	0.0007
512	0.0006	0.0010	0.0008	0.0013
1024	0.0010	0.0011	0.0011	0.0014
2048	0.0016	0.0017	0.0017	0.0018
4096	0.0029	0.0027	0.0029	0.0027
8192	0.0055	0.0052	0.0053	0.0053
16384	0.0106	0.0105	0.0100	0.0099
32768	0.0209	0.0200	0.0195	0.0188
65536	0.0415	0.0403	0.0385	0.0370
131072	0.0828	0.0812	0.0764	0.0734
262144	0.1652	0.1649	0.1523	0.1517
524288	0.3301	0.3269	0.3041	0.3006

Tabla 2.4: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un IBM SP2 con 4 procesadores interconectados mediante switch, para $128 \leq n \leq 524288$.*

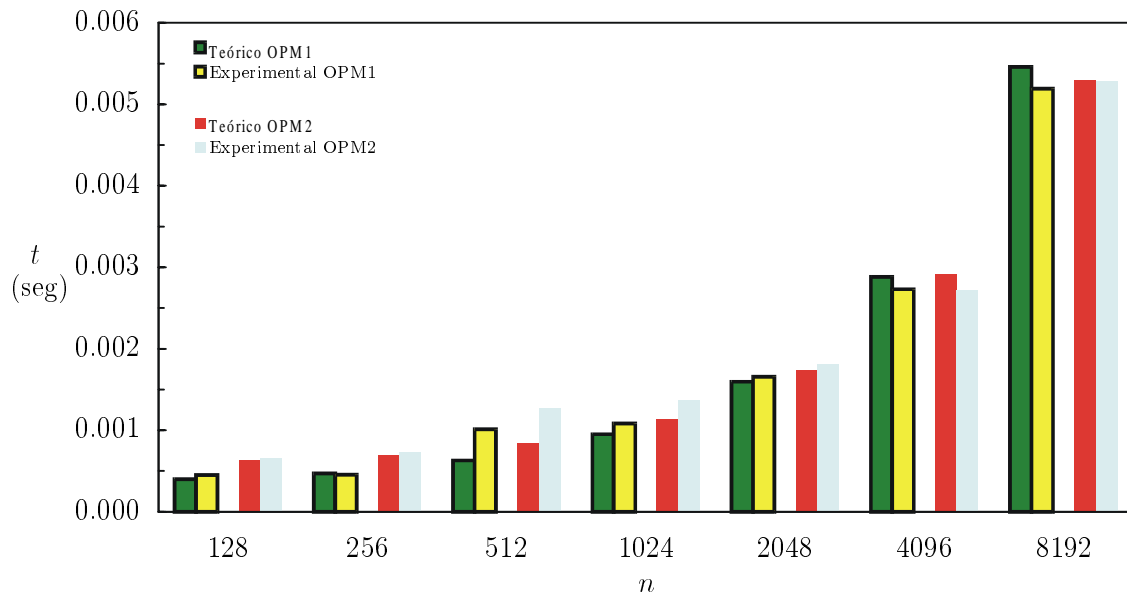
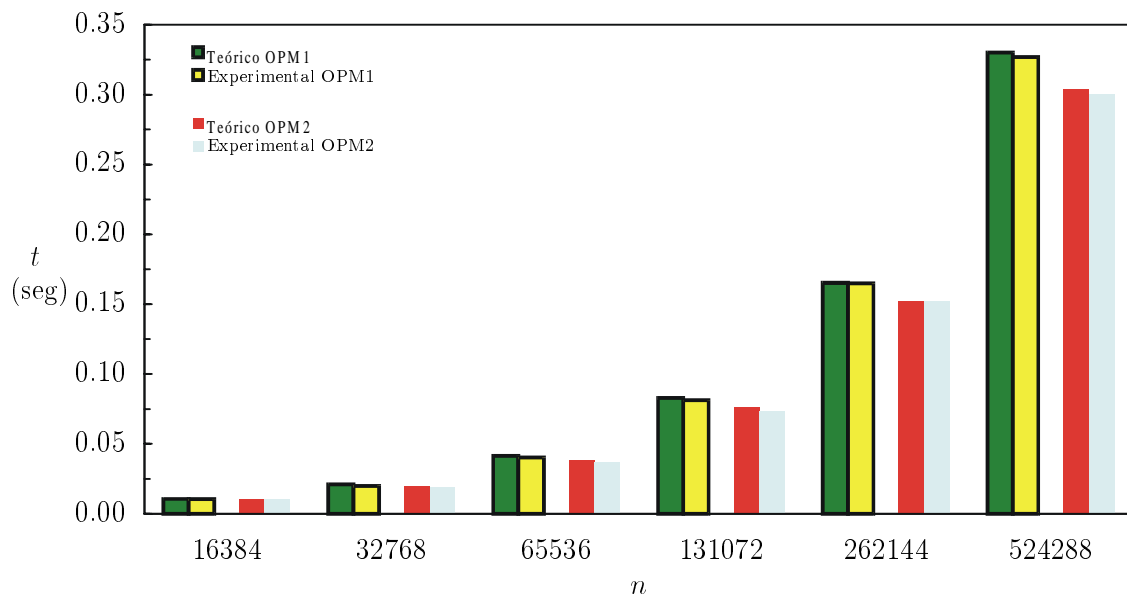
(a) $128 \leq n \leq 8192$ (b) $16384 \leq n \leq 524288$

Figura 2.5: Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 con 4 procesadores interconectados mediante switch.

IBM SP2 6 procesadores <i>switch</i>				
n	OPM1		OPM2	
	Teórico	Experimental	Teórico	Experimental
126	0.0006	0.0008	0.0009	0.0013
252	0.0007	0.0008	0.0010	0.0012
504	0.0008	0.0008	0.0012	0.0011
1008	0.0012	0.0015	0.0015	0.0018
2016	0.0018	0.0017	0.0021	0.0019
4032	0.0032	0.0029	0.0033	0.0031
8064	0.0059	0.0058	0.0058	0.0054
16128	0.0114	0.0113	0.0108	0.0106
32256	0.0223	0.0221	0.0209	0.0200
64512	0.0441	0.0432	0.0409	0.0401
129024	0.0877	0.0877	0.0809	0.0808
258048	0.1750	0.1682	0.1610	0.1591
516096	0.3494	0.3366	0.3211	0.3196

Tabla 2.5: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un IBM SP2 con 6 procesadores interconectados mediante switch, para $126 \leq n \leq 516096$.*

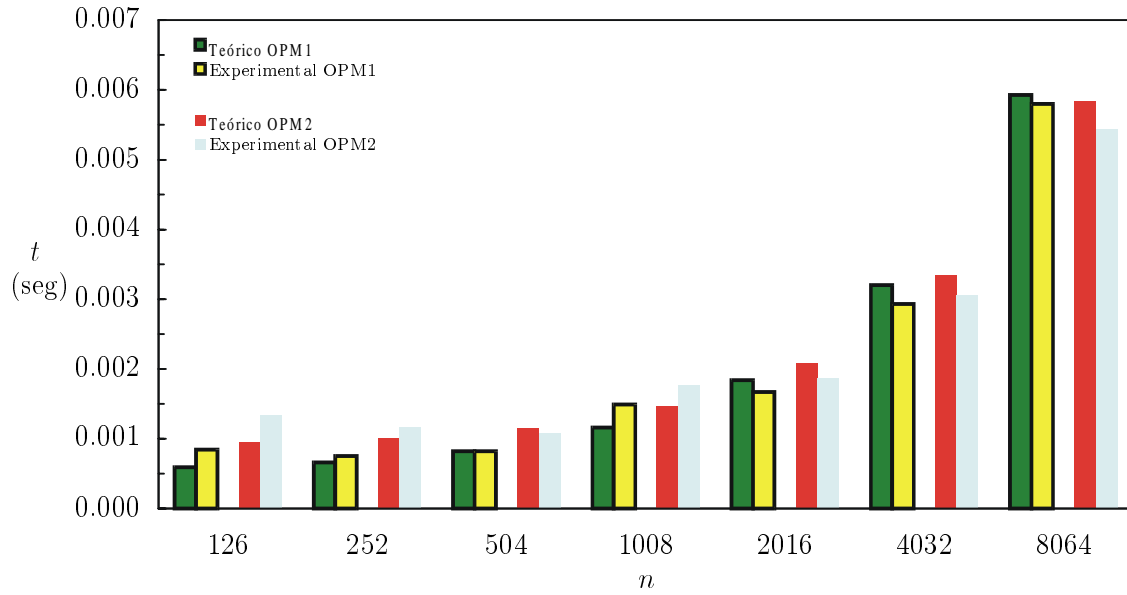
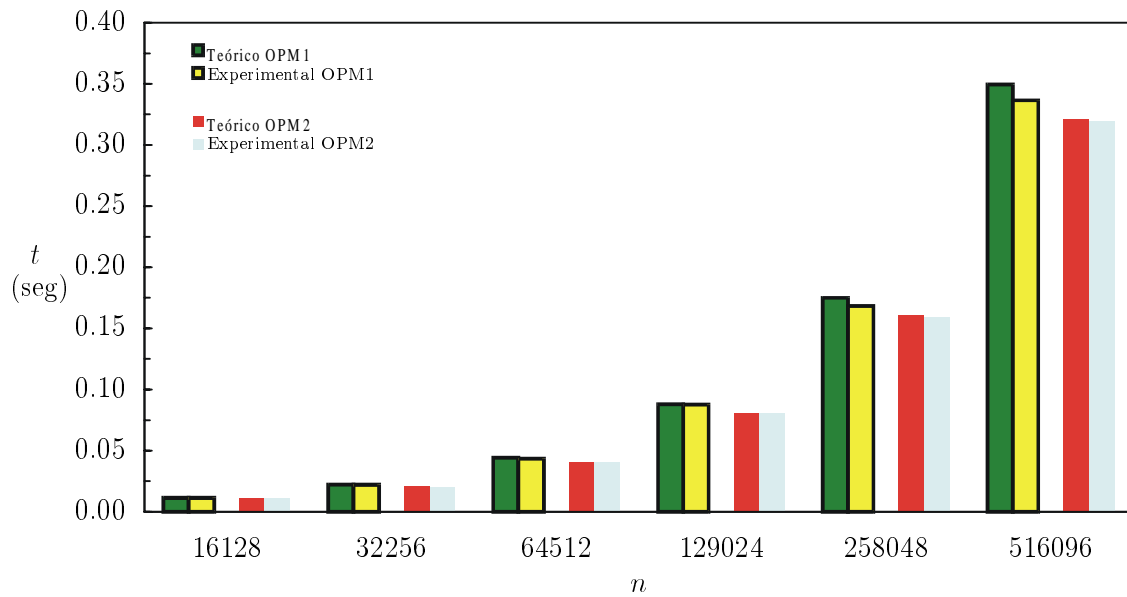
(a) $126 \leq n \leq 8064$ (b) $16128 \leq n \leq 516096$

Figura 2.6: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 con 6 procesadores interconectados mediante switch.*

IBM SP2 2 procesadores <i>ethernet</i>				
n	OPM1		OPM2	
	Teórico	Experimental	Teórico	Experimental
128	0.0037	0.0046	0.0046	0.0056
256	0.0063	0.0069	0.0072	0.0076
512	0.0114	0.0130	0.0122	0.0130
1024	0.0215	0.0235	0.0224	0.0264
2048	0.0419	0.0465	0.0427	0.0511
4096	0.0826	0.0782	0.0833	0.0815
8192	0.1639	0.1764	0.1646	0.1817
16384	0.3267	0.3213	0.3271	0.3125
32768	0.6522	0.6367	0.6520	0.6581
65536	1.3033	1.2936	1.3020	1.3145
131072	2.6053	2.6271	2.6019	2.5932
262144	5.2095	5.2019	5.2016	5.1770
524288	10.4178	10.3532	10.4012	10.3583

Tabla 2.6: Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un IBM SP2 con 2 procesadores interconectados mediante *ethernet*, para $128 \leq n \leq 524288$.

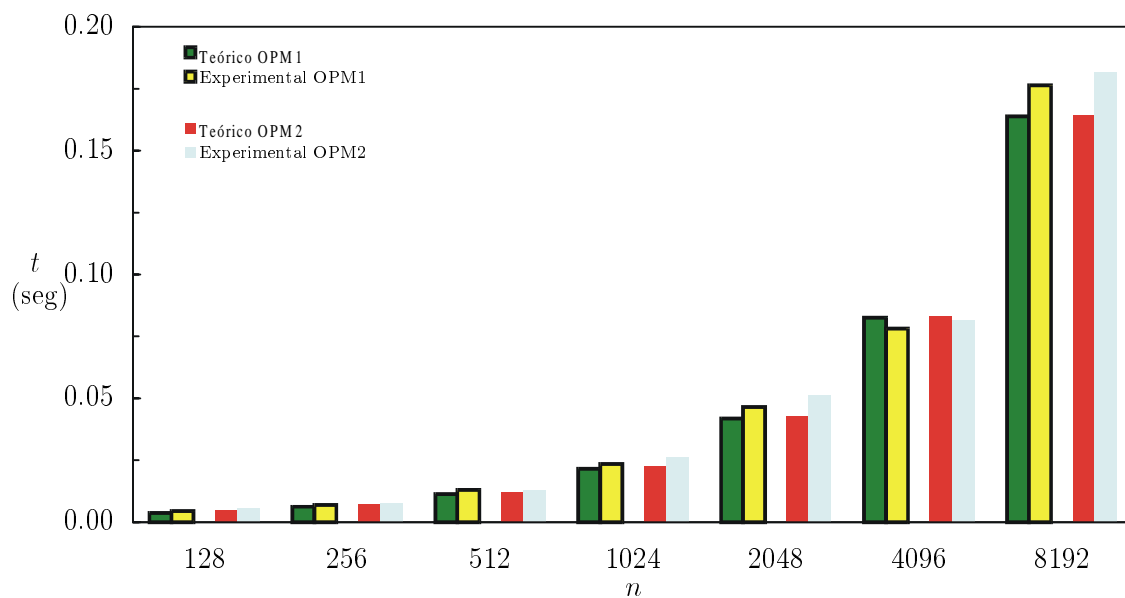
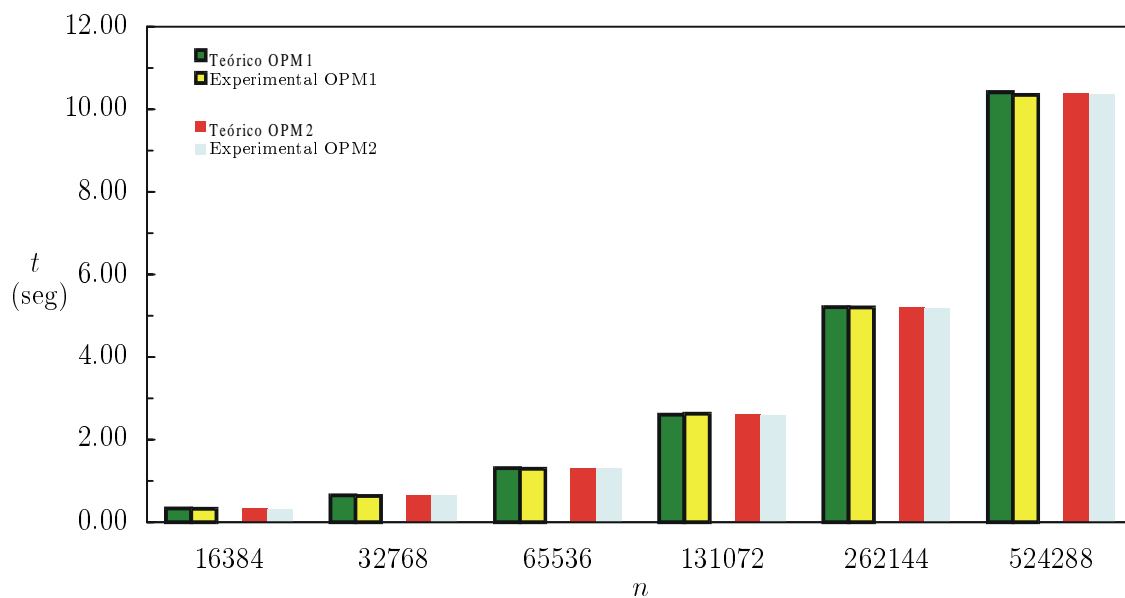
(a) $128 \leq n \leq 8192$ (b) $16384 \leq n \leq 524288$

Figura 2.7: Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 con 2 procesadores interconectados mediante ethernet.

IBM SP2 4 procesadores <i>ethernet</i>				
n	OPM1		OPM2	
	Teórico	Experimental	Teórico	Experimental
128	0.0136	0.0159	0.0162	0.0181
256	0.0207	0.0189	0.0232	0.0219
512	0.0352	0.0408	0.0376	0.0425
1024	0.0642	0.0770	0.0667	0.0757
2048	0.1225	0.1309	0.1249	0.1369
4096	0.2390	0.2467	0.2413	0.2562
8192	0.4720	0.4700	0.4741	0.4612
16384	0.9380	1.4150	0.9397	1.4889
32768	1.8700	1.8280	1.8709	1.7308
65536	3.7341	3.5331	3.7333	3.5249
131072	7.4623	7.3731	7.4582	7.3865
262144	14.9186	14.3351	14.9080	14.6099
524288	29.8313	29.2817	29.8075	28.4591

Tabla 2.7: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un IBM SP2 con 4 procesadores interconectados mediante ethernet, para $128 \leq n \leq 524288$.*

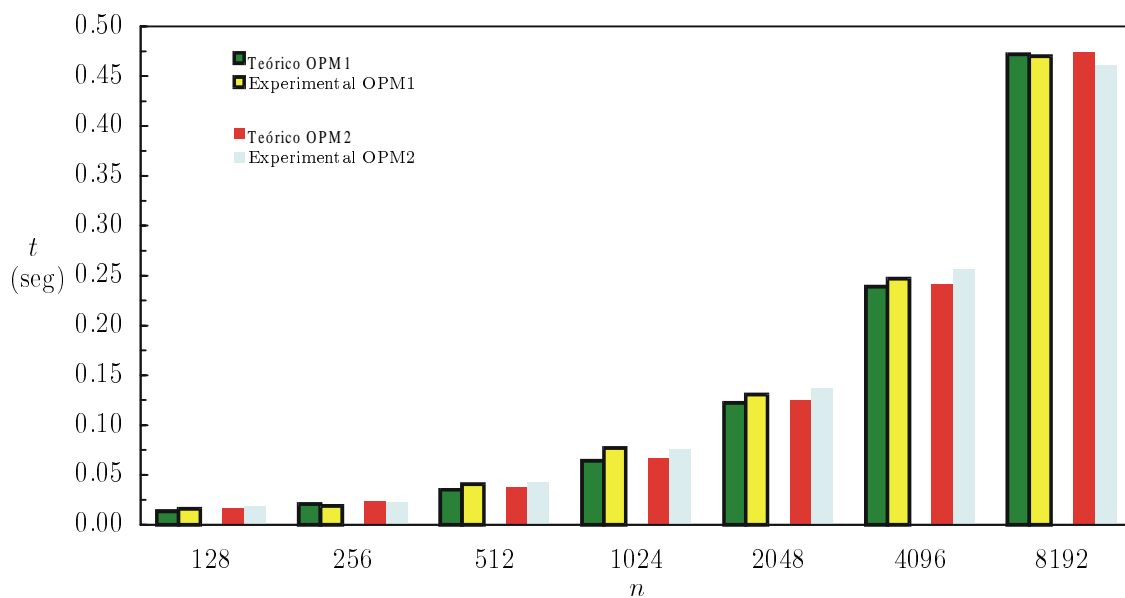
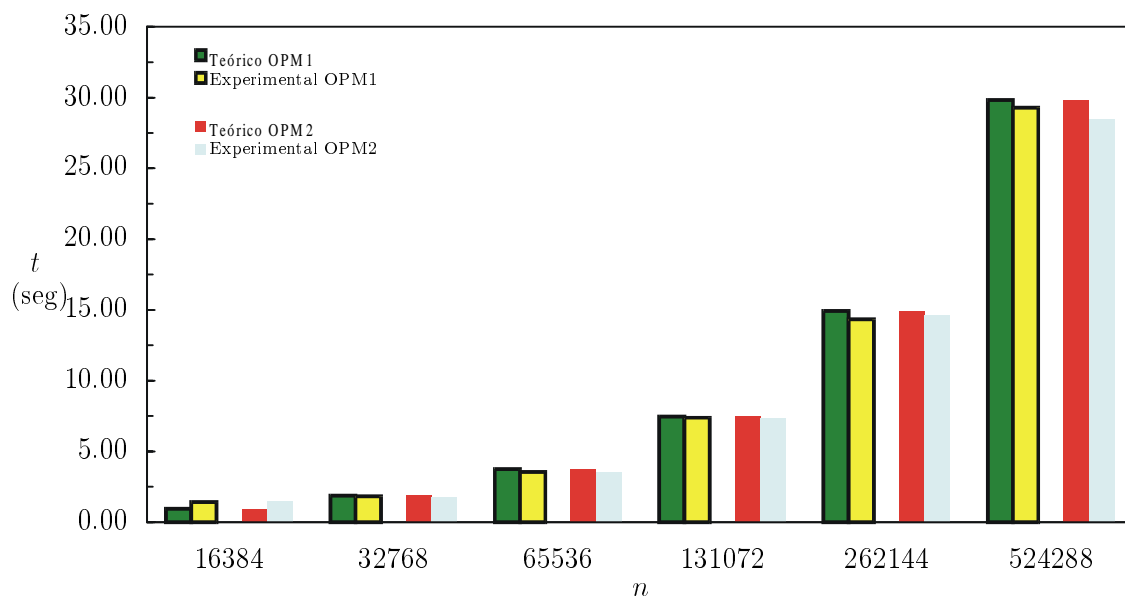
(a) $128 \leq n \leq 8192$ (b) $16384 \leq n \leq 524288$

Figura 2.8: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 con 4 procesadores interconectados mediante ethernet.*

IBM SP2 6 procesadores <i>ethernet</i>				
n	OPM1		OPM2	
	Teórico	Experimental	Teórico	Experimental
126	0.0412	0.0546	0.0471	0.0622
252	0.0586	0.0758	0.0644	0.0824
504	0.0954	0.1552	0.1012	0.1627
1008	0.1700	0.2054	0.1758	0.2019
2016	0.3198	0.3761	0.3255	0.3665
4032	0.6196	0.6954	0.6252	0.6916
8064	1.2194	1.1769	1.2247	1.1771
16128	2.4190	2.3189	2.4239	2.3334
32256	4.8182	5.1487	4.8222	5.0714
64512	9.6167	9.3015	9.6189	9.2169
129024	19.2137	18.8224	19.2123	18.1330
258048	38.4077	36.4179	38.3992	37.6399
516096	76.7957	75.1663	76.7728	71.9154

Tabla 2.8: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un IBM SP2 con 6 procesadores interconectados mediante ethernet, para $126 \leq n \leq 516096$.*

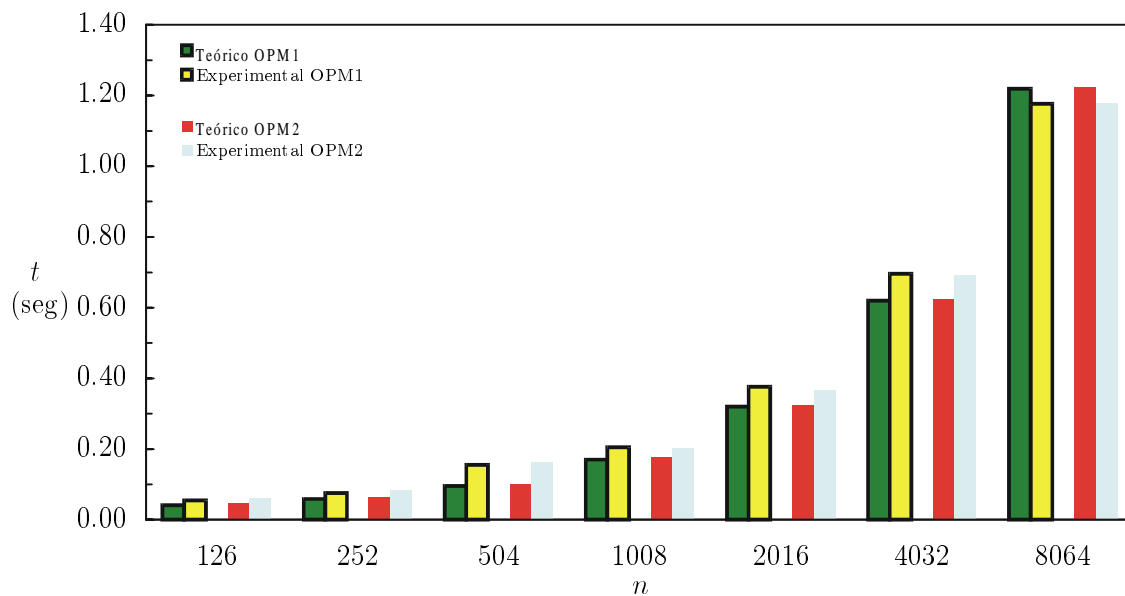
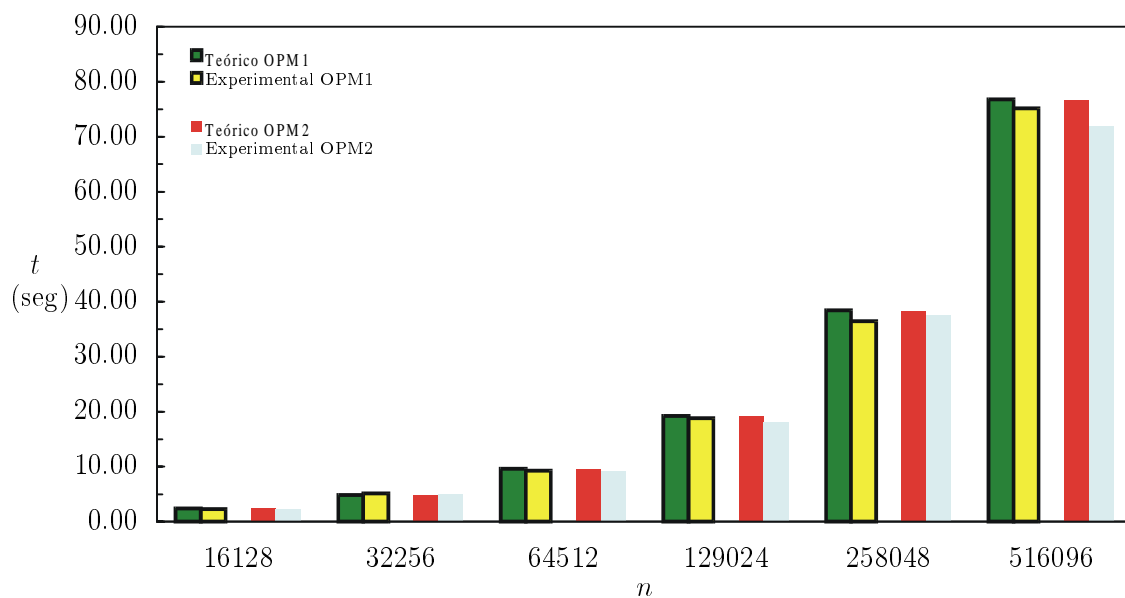
(a) $126 \leq n \leq 8064$ (b) $16128 \leq n \leq 516096$

Figura 2.9: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 con 6 procesadores interconectados mediante ethernet.*

<i>Cluster de PC's 2 procesadores</i>				
<i>n</i>	OPM1		OPM2	
	Teórico	Experimental	Teórico	Experimental
128	0.0026	0.0028	0.0050	0.0055
256	0.0027	0.0029	0.0052	0.0059
512	0.0030	0.0029	0.0054	0.0055
1024	0.0035	0.0039	0.0058	0.0064
2048	0.0044	0.0045	0.0067	0.0077
4096	0.0064	0.0064	0.0085	0.0086
8192	0.0103	0.0101	0.0120	0.0119
16384	0.0181	0.0175	0.0190	0.0190
32768	0.0336	0.0319	0.0331	0.0323
65536	0.0648	0.0641	0.0613	0.0592

Tabla 2.9: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un cluster de PC's, para 2 procesadores y $128 \leq n \leq 65536$.*

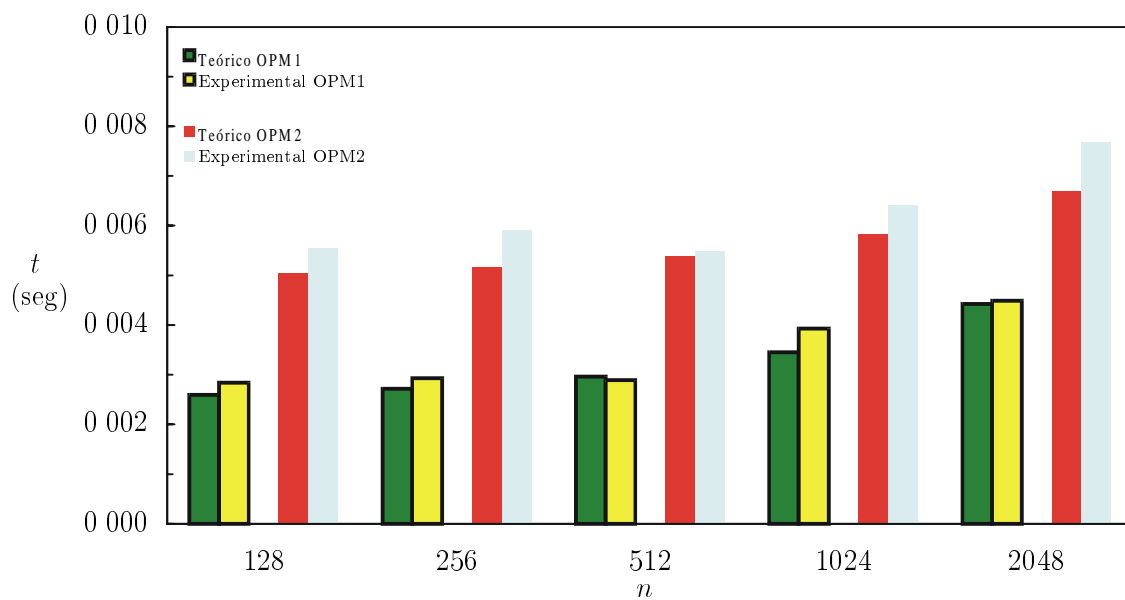
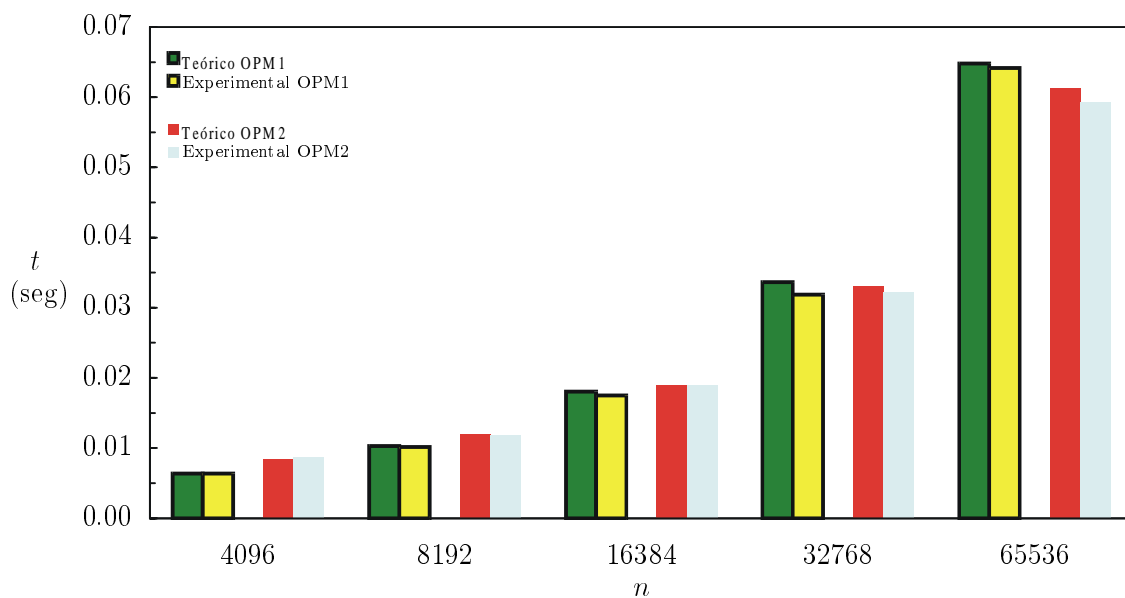
(a) $128 \leq n \leq 2048$ (b) $4096 \leq n \leq 65536$

Figura 2.10: Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un cluster de PC's con 2 procesadores.

<i>Cluster</i> de PC's 4 procesadores				
n	OPM1		OPM2	
	Teórico	Experimental	Teórico	Experimental
128	0.0068	0.0077	0.0134	0.0152
256	0.0069	0.0078	0.0135	0.0144
512	0.0072	0.0076	0.0137	0.0151
1024	0.0078	0.0079	0.0143	0.0136
2048	0.0090	0.0134	0.0153	0.0224
4096	0.0113	0.0109	0.0174	0.0169
8192	0.0160	0.0158	0.0215	0.0215
16384	0.0254	0.0239	0.0298	0.0292
32768	0.0442	0.0429	0.0463	0.0446
65536	0.0819	0.0807	0.0795	0.0782

Tabla 2.10: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un cluster de PC's, para 4 procesadores y $128 \leq n \leq 65536$.*

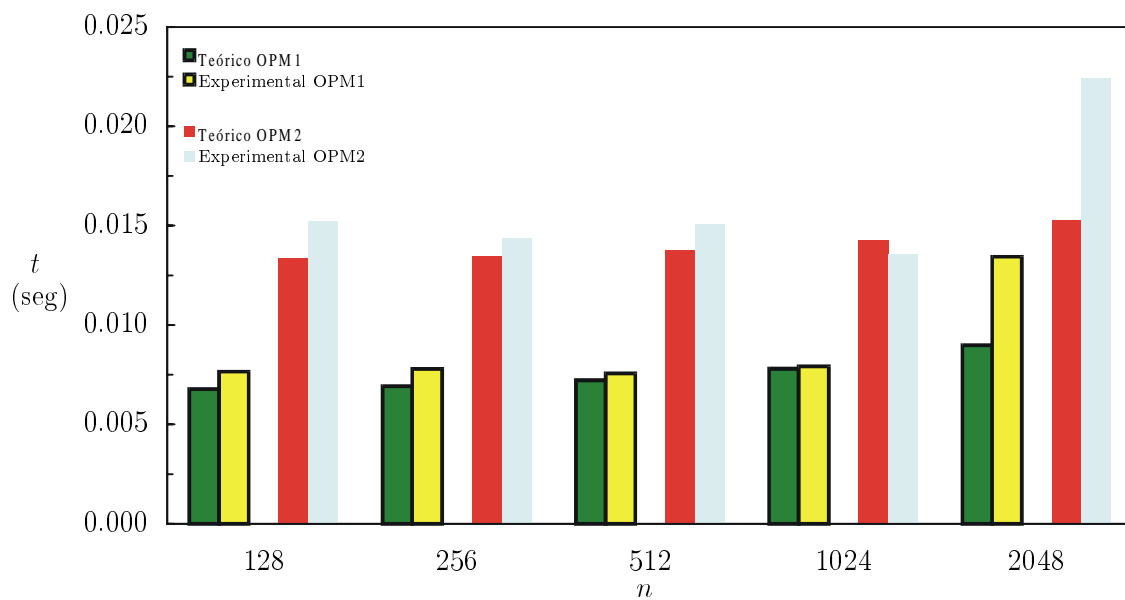
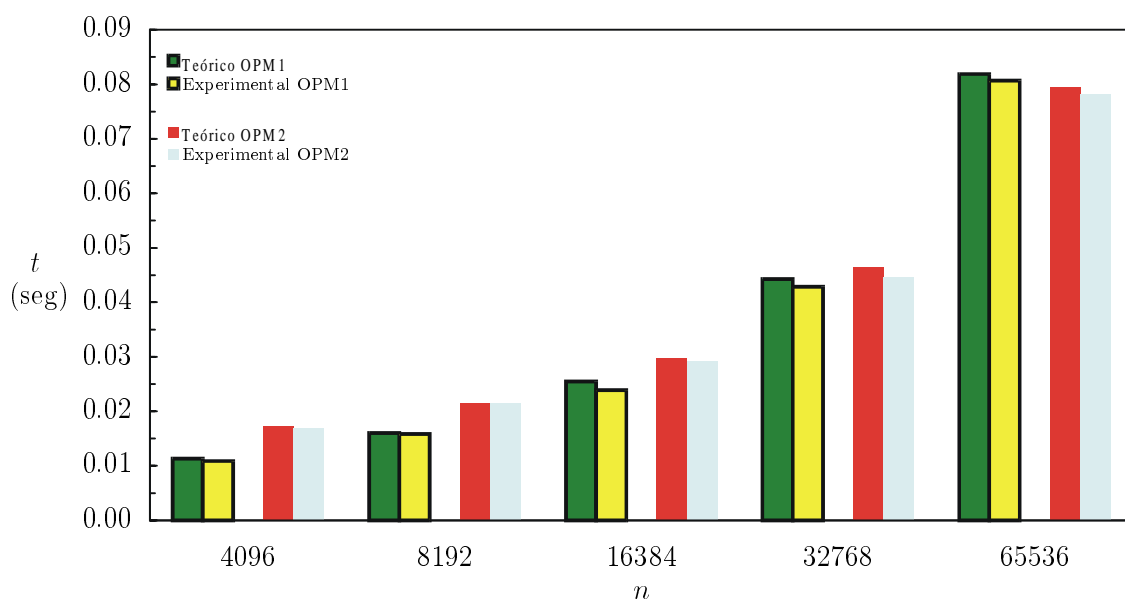
(a) $128 \leq n \leq 2048$ (b) $4096 \leq n \leq 65536$

Figura 2.11: Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un cluster de PC's con 4 procesadores.

<i>Cluster</i> de PC's 6 procesadores				
n	OPM1		OPM2	
	Teórico	Experimental	Teórico	Experimental
126	0.0150	0.0180	0.0299	0.0353
252	0.0152	0.0167	0.0300	0.0336
504	0.0155	0.0154	0.0302	0.0302
1008	0.0160	0.0150	0.0307	0.0294
2016	0.0172	0.0180	0.0317	0.0336
4032	0.0194	0.0186	0.0337	0.0330
8064	0.0240	0.0230	0.0376	0.0374
16128	0.0331	0.0310	0.0455	0.0451
32256	0.0512	0.0496	0.0612	0.0585
64512	0.0876	0.0867	0.0926	0.0912

Tabla 2.11: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un cluster de PC's, para 6 procesadores y $126 \leq n \leq 64512$.*

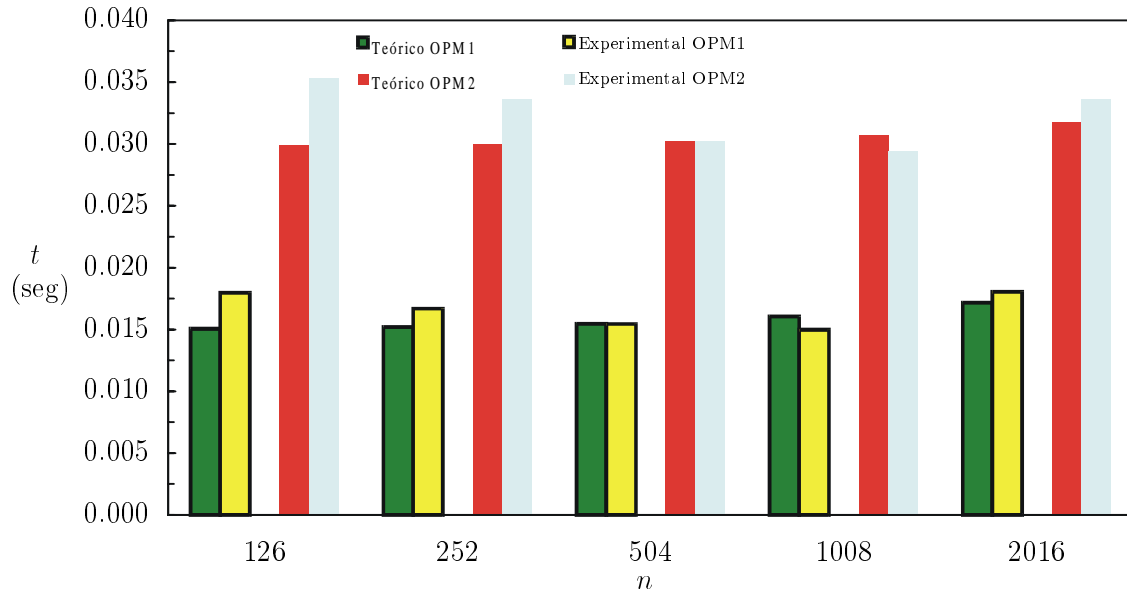
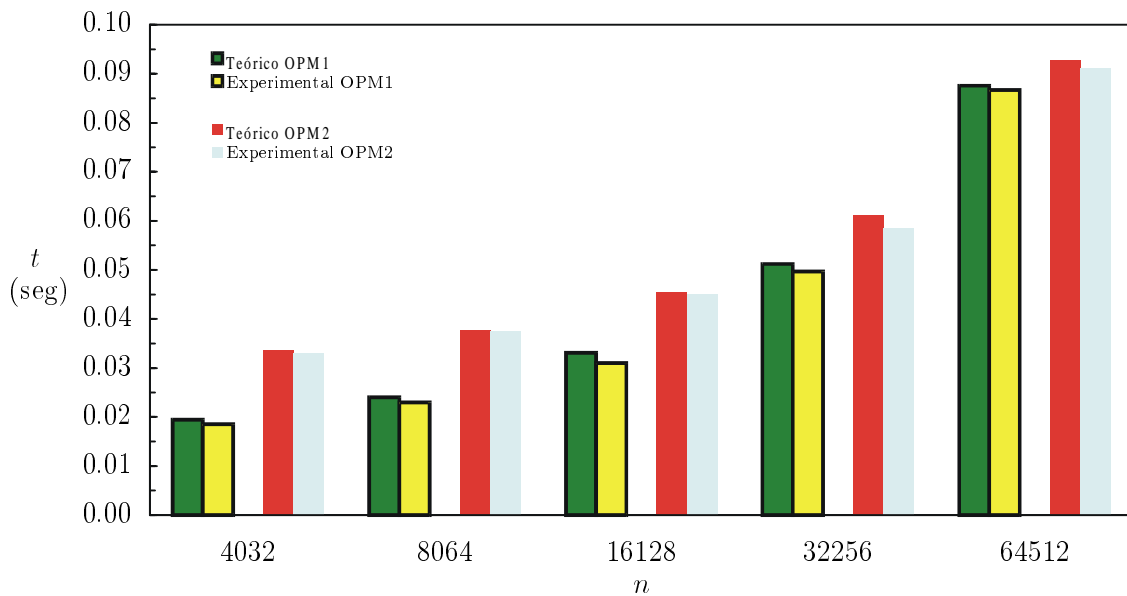
(a) $126 \leq n \leq 2016$ (b) $4032 \leq n \leq 64512$

Figura 2.12: *Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un cluster de PC's con 6 procesadores.*

En las figuras 2.13, 2.14 y 2.15 se muestra el porcentaje de desviación del tiempo experimental con respecto al teórico en función del tamaño del sistema¹.

Las mayores diferencias entre el tiempo previsto y el experimental se obtienen en los sistemas de tamaño pequeño en los que, por lo general, suele ser mayor el tiempo experimental que el teórico. A medida que aumenta el tamaño de sistema el tiempo obtenido experimentalmente se ajusta mejor al esperado, para tamaños grandes la desviación (salvo alguna excepción) oscila entre el 0% y el 5%.

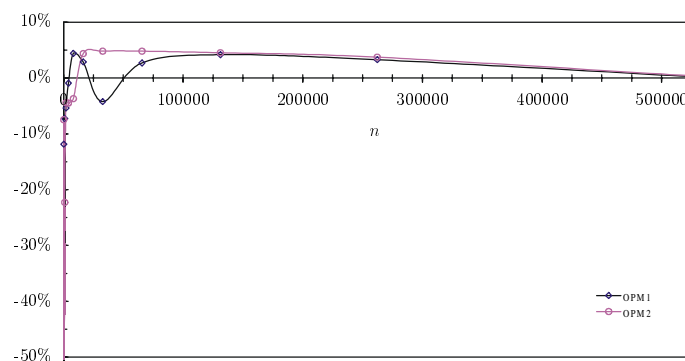
Este comportamiento obedece al ajuste teórico realizado, mediante la curva de Miller dada por la expresión (1.2), sobre los valores experimentales del coste de comunicación, $g(x)$, de una palabra de 32 bits para tamaños de mensaje x y patrón de comunicación *trid*. Por ejemplo, para el IBM SP2 *switch* y 2 procesadores los valores experimentales de $g(x)$ para tamaños de mensaje x comprendidos entre 64 y 512 palabras está por encima del coste teórico, como puede observarse en las figuras 1.15 y 1.15, en la práctica los tiempos obtenidos para valores de n comprendidos entre 128 y 1024 son² mayores que los previstos, como puede observarse en la figura 2.4(a). En esta figura se aprecia, además, que para valores de n comprendidos entre 2048 y 8192 el tiempo obtenido experimentalmente supera al predicho (aunque en menor proporción que para los tamaños comprendidos entre 128 y 1024). En este hecho influyen dos factores:

- el primero depende del ajuste realizado mediante la curva de Miller, que puede variar dependiendo de la estimación del valor de $n_{\frac{1}{2}}$;
- el segundo está relacionado con la propia estructura del algoritmo ya que se comunican algunos mensajes de tamaño 1 palabra de 32 bits y en el cómputo teórico se considera que su coste es el mismo que el de enviar una palabra para un bloque de tamaño k . El coste para tamaño 1 es mayor que para tamaño k y aunque tiene peso en el tiempo obtenido para tamaños de sistema pequeños es irrelevante para tamaños de sistema grandes.

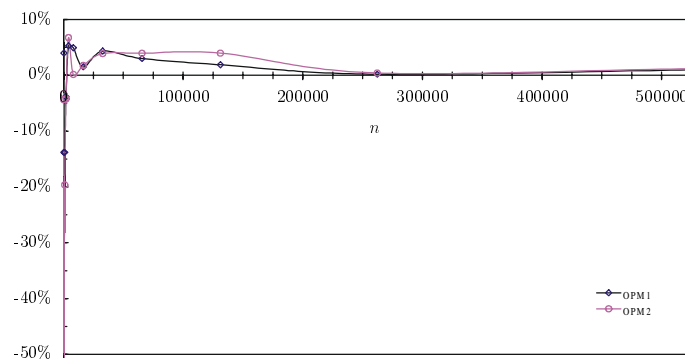
En las tablas 2.12, 2.13 y 2.14 se muestra el algoritmo más rápido de los dos, teórica y

¹Se considera que si el tiempo teórico es 100 y el tiempo experimental es 90, se ha producido un 10% de desviación y en cambio si el tiempo experimental es 110 se ha producido un -10% de desviación.

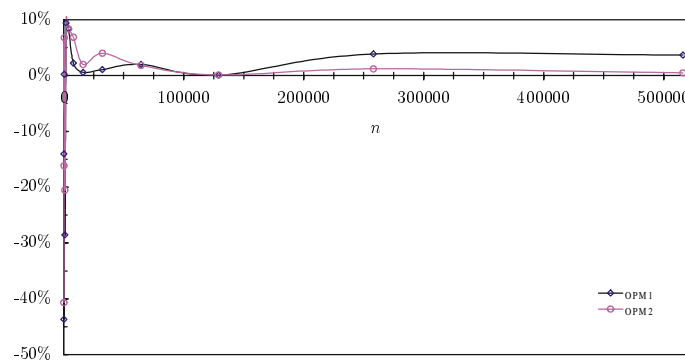
²Los tamaños de bloque enviados para un sistema de tamaño n y 2 procesadores son iguales a $\frac{n}{2}$.



(a) 2 procesadores

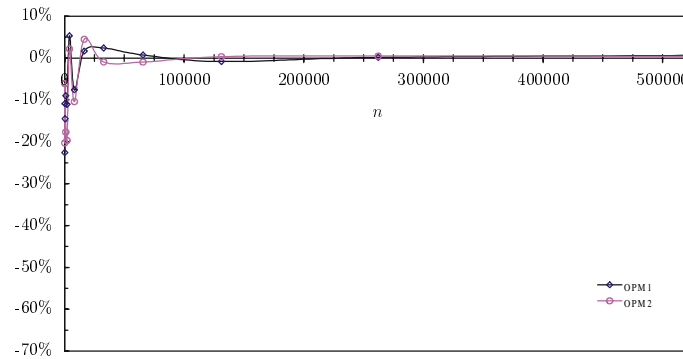


(b) 4 procesadores

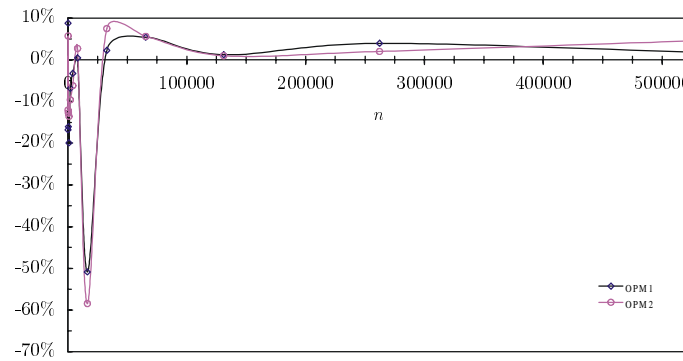


(c) 6 procesadores

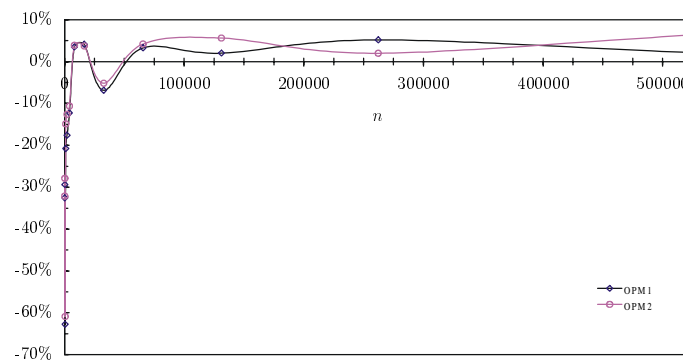
Figura 2.13: Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 con switch.



(a) 2 procesadores

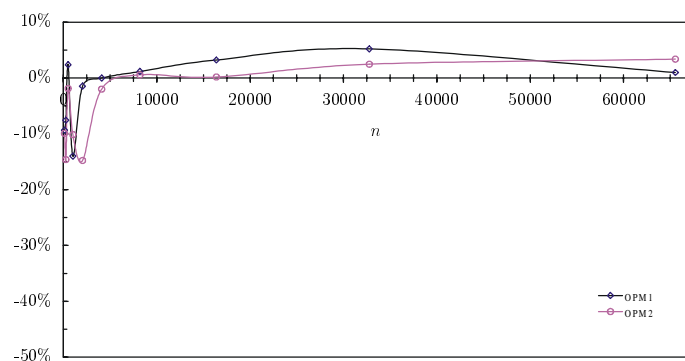


(b) 4 procesadores

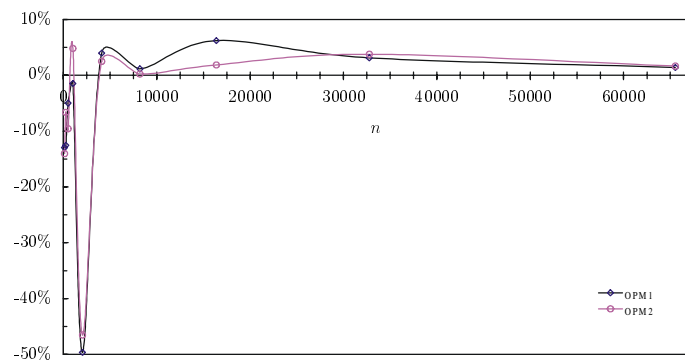


(c) 6 procesadores

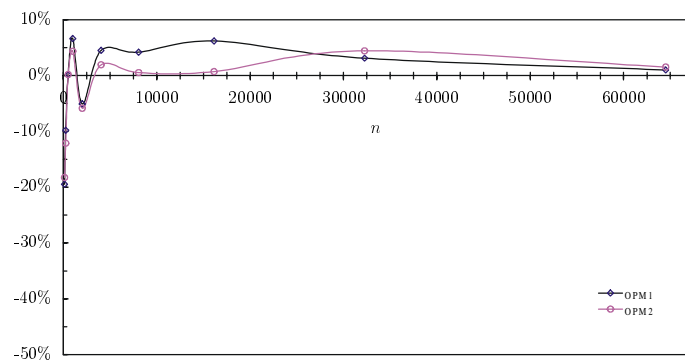
Figura 2.14: Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 con ethernet.



(a) 2 procesadores



(b) 4 procesadores



(c) 6 procesadores

Figura 2.15: Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un cluster de PC's.

IBM SP2 <i>switch</i>							
n	2 procesadores		4 procesadores		n	6 procesadores	
	Teór.	Exper.	Teór.	Exper.		Teór.	Exper.
128	OPM1	OPM1	OPM1	OPM1	126	OPM1	OPM1
256	OPM1	OPM1	OPM1	OPM1	252	OPM1	OPM1
512	OPM1	OPM1	OPM1	OPM1	504	OPM1	OPM1
1024	OPM1	OPM1	OPM1	OPM1	1008	OPM1	OPM1
2048	OPM1	OPM1	OPM1	OPM1	2016	OPM1	OPM1
4096	OPM1	OPM1	OPM1	OPM2	4032	OPM1	OPM1
8192	OPM2	OPM1	OPM2	OPM1	8064	OPM2	OPM2
16384	OPM2	OPM2	OPM2	OPM2	16128	OPM2	OPM2
32768	OPM2	OPM2	OPM2	OPM2	32256	OPM2	OPM2
65536	OPM2	OPM2	OPM2	OPM2	64512	OPM2	OPM2
131072	OPM2	OPM2	OPM2	OPM2	129024	OPM2	OPM2
262144	OPM2	OPM2	OPM2	OPM2	258048	OPM2	OPM2
524288	OPM2	OPM2	OPM2	OPM2	516096	OPM2	OPM2

Tabla 2.12: Algoritmo más rápido (teórica y experimentalmente) entre los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 *switch*.

experimentalmente, en cada una de las máquinas para los distintos tamaños de sistema. Se observa que, por lo general, a partir de 17000 ecuaciones (excepto en el IBM SP2 *switch*) es más rápido el algoritmo 2.3 (OPM2), que realiza el cálculo de m en paralelo.

No tiene mucho sentido estudiar el *speed-up* (incremento de velocidad) con respecto al tiempo de ejecución del propio algoritmo en un procesador por que se conoce el mejor algoritmo secuencial: el método de eliminación de Gauss para sistemas tridiagonales, cuyo coste para el sistema dado por la expresión 2.4 es $8n - 7$. Con respecto al método de eliminación de Gauss para sistemas tridiagonales se obtienen los resultados de la tabla 2.15, considerando los tiempos teóricos de los sistemas de tamaño máximo de los que se han obtenido resultados experimentales en cada una de las máquinas.

IBM SP2 <i>ethernet</i>							
n	2 procesadores		4 procesadores		n	6 procesadores	
	Teór.	Exper.	Teór.	Exper.		Teór.	Exper.
128	OPM1	OPM1	OPM1	OPM1	126	OPM1	OPM1
256	OPM1	OPM1	OPM1	OPM1	252	OPM1	OPM1
512	OPM1	OPM2	OPM1	OPM1	504	OPM1	OPM1
1024	OPM1	OPM1	OPM1	OPM2	1008	OPM1	OPM2
2048	OPM1	OPM1	OPM1	OPM1	2016	OPM1	OPM2
4096	OPM1	OPM1	OPM1	OPM1	4032	OPM1	OPM2
8192	OPM1	OPM1	OPM1	OPM2	8064	OPM1	OPM1
16384	OPM1	OPM2	OPM1	OPM1	16128	OPM1	OPM1
32768	OPM2	OPM1	OPM1	OPM2	32256	OPM1	OPM2
65536	OPM2	OPM1	OPM2	OPM2	64512	OPM1	OPM2
131072	OPM2	OPM2	OPM2	OPM1	129024	OPM2	OPM2
262144	OPM2	OPM2	OPM2	OPM1	258048	OPM2	OPM1
524288	OPM2	OPM1	OPM2	OPM2	516096	OPM2	OPM2

Tabla 2.13: Algoritmo más rápido (teórica y experimentalmente) entre los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 *ethernet*.

<i>Cluster de PC's</i>							
n	2 procesadores		4 procesadores		n	6 procesadores	
	Teór.	Exper.	Teór.	Exper.		Teór.	Exper.
128	OPM1	OPM1	OPM1	OPM1	126	OPM1	OPM1
256	OPM1	OPM1	OPM1	OPM1	252	OPM1	OPM1
512	OPM1	OPM1	OPM1	OPM1	504	OPM1	OPM1
1024	OPM1	OPM1	OPM1	OPM1	1008	OPM1	OPM1
2048	OPM1	OPM1	OPM1	OPM1	2016	OPM1	OPM1
4096	OPM1	OPM1	OPM1	OPM1	4032	OPM1	OPM1
8192	OPM1	OPM1	OPM1	OPM1	8064	OPM1	OPM1
16384	OPM1	OPM1	OPM1	OPM1	16128	OPM1	OPM1
32768	OPM2	OPM1	OPM1	OPM1	32256	OPM1	OPM1
65536	OPM2	OPM2	OPM2	OPM2	64512	OPM1	OPM1

Tabla 2.14: Algoritmo más rápido (teórica y experimentalmente) entre los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un cluster de PC's.

IBM SP2 <i>switch</i>				
p	OPM1		OPM2	
	S_p	E_p	S_p	E_p
2	0.42	21.25%	0.46	23.07%
4	0.28	7.06%	0.31	7.66%
6	0.26	4.38%	0.29	4.76%

(a) IBM SP2 *switch*

IBM SP2 <i>ethernet</i>				
p	OPM1		OPM2	
	S_p	E_p	S_p	E_p
2	0.01	0.45%	0.01	0.45%
4	0.00	0.08%	0.00	0.08%
6	0.00	0.02%	0.00	0.02%

(b) IBM SP2 *ethernet*

Cluster de PC's				
p	OPM1		OPM2	
	S_p	E_p	S_p	E_p
2	0.49	24.61%	0.52	26.02%
4	0.39	9.74%	0.40	10.03%
6	0.36	5.97%	0.34	5.65%

(c) Cluster de PC's

Tabla 2.15: Speed-up (S_p) y eficiencia (E_p) de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 y un cluster de PC's.

s	p	l	g	$n_{\frac{1}{2}}$
12	1	68	0.3	94
	2	164	0.7	71
	4	168	0.7	66
	8	175	0.8	59
	16	181	0.9	61
	32	201	1.1	28
	64	148	1	27
	128	301	1.1	20
	256	387	1.2	15

(a) CRAY T3D

s	p	l	g	$n_{\frac{1}{2}}$
46.7	1	86	2.12	9
	2	269	0.87	33
	4	357	0.87	40
	8	506	0.81	40
	16	751	1.04	38
	32	1252	1.31	45

(b) CRAY T3E

Tabla 2.16: Valores de parámetros BSP.

La eficiencia que se obtiene no es muy buena, y en caso del IBM SP2 con *ethernet* es pésima; sin duda un factor determinante es el valor de g . Para otras máquinas con menor valor de g se obtiene mejor eficiencia, como es el caso de un CRAY T3D o un CRAY T3E. Los valores de los parámetros de estas máquinas se muestran en la tabla 2.16; para las mismas se obtiene el *speed-up* y la eficiencia que figura en la tabla 2.17, considerando el mismo tamaño de sistema que para la tabla 2.15.

CRAY T3D				
p	OPM1		OPM2	
	S_p	E_p	S_p	E_p
2	1.13	56.43%	1.43	71.57%
4	1.27	31.68%	1.97	49.20%
8	1.39	17.38%	2.56	31.97%
16	1.42	8.91%	2.85	17.84%
32	1.35	4.22%	2.65	8.29%
64	1.43	2.23%	3.02	4.73%
128	1.38	1.08%	2.82	2.20%
256	1.32	0.52%	2.61	1.02%

(a) CRAY T3D

CRAY T3E				
p	OPM1		OPM2	
	S_p	E_p	S_p	E_p
2	0.99	49.45%	1.21	60.70%
4	1.13	28.15%	1.65	41.19%
8	1.39	17.32%	2.54	31.72%
16	1.35	8.41%	2.55	15.95%
32	1.22	7.60%	2.12	13.25%

(b) CRAY T3E

Tabla 2.17: *Speed-up* (S_p) y *eficiencia* (E_p) de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un CRAY T3D y un CRAY T3E.