

---

# Análisis de eficiencia de algoritmos BSP para la resolución de sistemas lineales tridiagonales

Antonio Zamora Gómez

## Tesis de Doctorado

**Facultad:** Escuela Politécnica Superior

**Director:** Dr. Joan Josep Climent Coloma

**2000**

---

Universidad de Alicante  
Departamento de Ciencia de la  
Computación e Inteligencia Artificial



Análisis de eficiencia de algoritmos BSP para la  
resolución de sistemas lineales tridiagonales

TESIS DOCTORAL

Presentada por:

Antonio Zamora Gómez

Dirigida por:

Joan Josep Climent Coloma

**Universidad de Alicante**  
**Departamento de Ciencia de la**  
**Computación e Inteligencia Artificial**

**Análisis de eficiencia de algoritmos BSP para la  
resolución de sistemas lineales tridiagonales**

Memoria presentada para optar al grado  
de Doctor Ingeniero en Informática por  
ANTONIO ZAMORA GÓMEZ.

Alicante, 14 de Diciembre de 1999

Don JOAN JOSEP CLIMENT COLOMA, Profesor Titular de Universidad del Departamento de Ciencia de la Computación e Inteligencia Artificial de la Universidad de Alicante,

CERTIFICA:

Que la presente memoria *Análisis de eficiencia de algoritmos BSP para la resolución de sistemas lineales tridiagonales*, ha sido realizada bajo su dirección, en el Departamento de Ciencia de la Computación e Inteligencia Artificial de la Universidad de Alicante, por el licenciado Don ANTONIO ZAMORA GÓMEZ, y constituye su tesis para optar al grado de Doctor Ingeniero en Informática. Para que conste, en cumplimiento de la legislación vigente, autoriza la presentación de la referida tesis doctoral ante la comisión de Doctorado de la Universidad de Alicante, firmando el presente certificado.

Alicante, 14 de Diciembre de 1999.

Fdo.: Joan Josep Climent Coloma

*A Carlos y Alberto, esos diablillos.*

*A Yolanda, por su amor y ayuda.*

*A mis padres, por el apoyo que me han dado.*

Quiero expresar mi más sincero agradecimiento:

Al profesor Joan Josep Climent Coloma por sus constantes orientaciones científicas, por mostrarme día a día qué significa tener capacidad de trabajo, por su incondicional ayuda sin la cual habría sido imposible la realización de esta memoria y sobre todo por su amistad.

Al profesor Leandro Tortosa, compañero de fatigas, por haber estado siempre ahí, por haber hecho mucho más agradable el trabajo y por haberme brindado su amistad desde el primer día.

A los miembros del Grupo de Computación Paralela de la Universidad de Alicante por el buen ambiente que se respira.

A los técnicos Ginés López, Francisco Martínez y Francisco Maciá por su decisiva y pronta ayuda en la resolución de los problemas surgidos en el uso de las distintas máquinas.

Esta tesis doctoral ha sido subvencionada por el proyecto número PB98-0977 de la Dirección General de Enseñanza Superior e Investigación Científica.

# Índice

<b>Índice de figuras</b>	<b>xii</b>
<b>Índice de tablas</b>	<b>xviii</b>
<b>Prólogo</b>	<b>xix</b>
<b>1 Computación Paralela</b>	<b>1</b>
1.1 Antecedentes de la computación paralela . . . . .	2
1.2 Modelos de computación paralela . . . . .	3
1.2.1 El modelo PRAM (1978) . . . . .	4
1.2.2 El modelo BSP (1990) . . . . .	5
1.2.3 El modelo LogP (1993) . . . . .	7
1.2.4 Data-Parallel . . . . .	8
1.2.5 Modelos de paso de mensajes (1989)(1992) . . . . .	8
1.3 Plataformas de computación paralela . . . . .	9
1.4 Medidas de paralelismo . . . . .	16
1.5 El modelo BSP . . . . .	17

---

1.5.1	Los parámetros del modelo BSP . . . . .	18
1.5.2	El modelo de programación . . . . .	20
1.5.3	El modelo de coste . . . . .	24
1.5.4	Librerías BSP disponibles . . . . .	32
<b>2</b>	<b>Método de las particiones superpuestas</b>	<b>47</b>
2.1	Preliminares . . . . .	47
2.2	Descripción del método. . . . .	48
2.3	Precisión del método . . . . .	57
2.4	Paralelización del método. . . . .	70
2.5	Algoritmos BSP . . . . .	76
2.6	Resultados numéricos . . . . .	81
<b>3</b>	<b>Métodos bidireccionales</b>	<b>111</b>
3.1	Factorización <i>LDU</i> de matrices tridiagonales . . . . .	112
3.2	Método bidireccional para dos procesadores . . . . .	114
3.2.1	Descripción del método . . . . .	114
3.2.2	Algoritmo BSP para dos procesadores . . . . .	122
3.3	Método bidireccional para un número par de procesadores . . . . .	125
3.3.1	Descripción del método . . . . .	125
3.3.2	Algoritmos BSP para un número par de procesadores . . . . .	129
3.4	Resultados numéricos . . . . .	141

---

<b>4</b>	<b>Método de Wang</b>	<b>171</b>
4.1	Descripción del método . . . . .	171
4.2	Paralelización del método . . . . .	181
4.3	Algoritmos BSP . . . . .	187
4.4	Coste computacional de los algoritmos basados en el método de Wang . . .	192
4.5	Resultados numéricos . . . . .	195
<b>5</b>	<b>Comparación entre métodos</b>	<b>223</b>
5.1	IBM SP2 . . . . .	225
5.1.1	<i>Switch</i> . . . . .	225
5.1.2	<i>Ethernet</i> . . . . .	229
5.2	<i>Cluster</i> de PC's . . . . .	229
5.3	CRAY T3D . . . . .	236
5.4	CRAY T3E . . . . .	245
	<b>Conclusiones y líneas futuras</b>	<b>255</b>
	<b>Bibliografía</b>	<b>257</b>

# Índice de figuras

1.1	Computador BSP: modelo arquitectónico. . . . .	6
1.2	Multiprocesador con memoria distribuida. . . . .	10
1.3	Topología de anillo. . . . .	12
1.4	Topología de malla bidimensional. . . . .	12
1.5	Topología de malla tridimensional. . . . .	13
1.6	Estructura básica de un IBM SP2. . . . .	15
1.7	Ejemplos de distintos tipos de patrones de comunicación que definen una $h$ -relación. . . . .	19
1.8	Espacio generado por los parámetros $p$ , $l$ y $g$ . . . . .	20
1.9	Segmento de un programa BSP. . . . .	22
1.10	Fases de una barrera de sincronización organizada en forma de árbol . . . .	23
1.11	Curva teórica de $g$ para los valores $n_{\frac{1}{2}} = 12$ y $g_{\infty} = 0.23\mu s$ . . . . .	28
1.12	Patrón de comunicación todos a todos para 4 procesadores. . . . .	29
1.13	Comparación de los valores $g(x)$ obtenidos experimentalmente con el patrón de comunicación <i>trid</i> y de la curva teórica de Miller en un IBM SP2 con 4 procesadores interconectados mediante <i>switch</i> . . . . .	34

1.14	Comparación de los valores $g(x)$ obtenidos experimentalmente con el patrón de comunicación <b>trid</b> y de la curva teórica de Miller en un IBM SP2 con 4 procesadores interconectados mediante <i>switch</i> . . . . .	35
1.15	Comparación de los valores $g(x)$ obtenidos experimentalmente con el patrón de comunicación <b>trid</b> y de la curva teórica de Miller en un IBM SP2 con 2 procesadores interconectados mediante <i>switch</i> . . . . .	36
1.16	Comparación de los valores $g(x)$ obtenidos experimentalmente con el patrón de comunicación <b>trid</b> y de la curva teórica de Miller en un IBM SP2 con 2 procesadores interconectados mediante <i>switch</i> . . . . .	37
1.17	Comparación de los valores $g(x)$ obtenidos experimentalmente con los patrones de comunicación <i>local</i> , <b>trid</b> y <i>all-to-all</i> en un IBM SP2 con 4 procesadores interconectados mediante <i>switch</i> . La curva teórica de Miller está ajustada al modelo <b>trid</b> . . . . .	38
1.18	Comparación de los valores $g(x)$ obtenidos experimentalmente con los patrones de comunicación <i>local</i> , <b>trid</b> y <i>all-to-all</i> en un IBM SP2 con 4 procesadores interconectados mediante <i>switch</i> . La curva teórica de Miller está ajustada al modelo <b>trid</b> . . . . .	39
2.1	Partición natural y superpuesta de $A$ y $\mathbf{d}$ para $p = 4$ . . . . .	56
2.2	Obtención de la solución general a partir de las soluciones parciales para $p = 4$ . . . . .	57
2.3	Valor de $m$ en función de $\delta$ para distintos valores de $\epsilon$ . . . . .	69
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 <i>switch</i> . . . . .	84
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 <i>switch</i> . . . . .	86

---

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 <i>switch</i> . . . . .	88
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 <i>ethernet</i> . . . . .	90
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 <i>ethernet</i> . . . . .	92
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 <i>ethernet</i> . . . . .	94
2.10	Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un <i>cluster</i> de PC's con 2 procesadores. . . . .	96
2.11	Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un <i>cluster</i> de PC's con 4 procesadores. . . . .	98
2.12	Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un <i>cluster</i> de PC's con 6 procesadores. . . . .	100
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 <i>switch</i> . . . . .	102
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 <i>ethernet</i> . . . . .	103
2.15	Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un <i>cluster</i> de PC's. . . . .	104
3.1	Cálculo en paralelo de los elementos de $D$ para una matriz con $n = 8$ . . . . .	118
3.2	Cálculo en paralelo de la solución del sistema (2.4) para una matriz con $n = 8$ . . . . .	119

---

3.3	Obtención de la solución general a partir de las soluciones parciales para $p = 8$ . . . . .	130
3.4	Esquema de comunicación para 8 procesadores del método descrito en el algoritmo 3.2. . . . .	132
3.5	Tiempos teóricos y experimentales del algoritmo 3.1 (TW) en un IBM SP2 con 2 procesadores interconectados mediante <i>switch</i> . . . . .	144
3.6	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2) en un IBM SP2 con 4 procesadores interconectados mediante <i>switch</i> . . . . .	146
3.7	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2) en un IBM SP2 con 6 procesadores interconectados mediante <i>switch</i> . . . . .	148
3.8	Tiempos teóricos y experimentales del algoritmo 3.1 (TW) en un IBM SP2 con 2 procesadores interconectados mediante <i>ethernet</i> . . . . .	150
3.9	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2) en un IBM SP2 con 4 procesadores interconectados mediante <i>ethernet</i> . . . . .	152
3.10	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2) en un IBM SP2 con 6 procesadores interconectados mediante <i>ethernet</i> . . . . .	154
3.11	Tiempos teóricos y experimentales del algoritmo 3.1 (TW) en un <i>cluster</i> de PC's para 2 procesadores. . . . .	157
3.12	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2) en un <i>cluster</i> de PC's para 4 procesadores. . . . .	159
3.13	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2) en un <i>cluster</i> de PC's para 6 procesadores. . . . .	161
3.14	Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 3.1 (TW), 3.2 (TW1) y 3.3 (TW2) en un IBM SP2 con <i>switch</i> . . . . .	162
3.15	Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 3.1 (TW), 3.2 (TW1) y 3.3 (TW2) en un IBM SP2 con <i>ethernet</i> . . . . .	163

---

3.16	Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 3.1 (TW), 3.2 (TW1) y 3.3 (TW2) en un <i>cluster</i> de PC's. . . .	164
4.1	Matriz de coeficientes para el caso $n = 16$ . . . . .	175
4.2	Transformación de la matriz de coeficientes después de la eliminación de los elementos situados por debajo de la diagonal principal, para el caso $n = 16$ . . . . .	176
4.3	Transformación de la matriz de coeficientes después de la eliminación los elementos situados por encima de la diagonal principal, para el caso $n = 16$ . . . . .	177
4.4	Transformación de la matriz de coeficientes después de anular los elementos $b_4, b_8$ y $b_{12}$ , para el caso $n = 16$ . . . . .	178
4.5	Transformación de la matriz de coeficientes después de anular los elementos $f_{4i+j}$ , con $i = 1, 2, 3$ y $j = 1, 2, 3, 4$ , para el caso $n = 16$ . . . . .	179
4.6	Transformación de la matriz de coeficientes después de anular los elementos $g_{4i+j}$ , con $i = 0, 1, 2, 3$ y $j = 1, 2, 3$ , para el caso $n = 16$ . . . . .	180
4.7	Transformación de la matriz de coeficientes después de eliminar los nuevos elementos no nulos, para el caso $n = 16$ . . . . .	181
4.8	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 2 procesadores interconectados mediante <i>switch</i> . . . . .	199
4.9	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 4 procesadores interconectados mediante <i>switch</i> . . . . .	201
4.10	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 6 procesadores interconectados mediante <i>switch</i> . . . . .	203
4.11	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3(WANG2) en un IBM SP2 con 2 procesadores interconectados mediante <i>ethernet</i> . . . . .	205
4.12	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 4 procesadores interconectados mediante <i>ethernet</i> . . . . .	207

- 
- 4.13 Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 6 procesadores interconectados mediante *ethernet*. . . 209
- 4.14 Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un *cluster* de PC's para 2 procesadores. . . . . 211
- 4.15 Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un *cluster* de PC's para 4 procesadores. . . . . 213
- 4.16 Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un *cluster* de PC's para 6 procesadores. . . . . 215
- 4.17 Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con *switch*. . . . 216
- 4.18 Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con *ethernet*. . . 217
- 4.19 Diferencias porcentuales entre los tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un *cluster* de PC's. . . . . 218
- 5.1 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.1 (TW), 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 2 procesadores interconectados mediante *switch*, para  $16384 \leq n \leq 524288$ . . . . . 226
- 5.2 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 4 procesadores interconectados mediante *switch*, para  $16384 \leq n \leq 524288$ . . . . . 227
- 5.3 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 6 procesadores interconectados mediante *switch*, para  $16128 \leq n \leq 516096$ . . . . . 228
- 5.4 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.1 (TW), 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 2 procesadores interconectados mediante *ethernet*, para  $16384 \leq n \leq 524288$ . . . . . 230

---

5.5	Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 4 procesadores interconectados mediante <i>ethernet</i> , para $16384 \leq n \leq 524288$ . . . . .	231
5.6	Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 con 6 procesadores interconectados mediante <i>ethernet</i> , para $16128 \leq n \leq 516096$ . . . . .	232
5.7	Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.1 (TW), 4.2 (WANG1) y 4.3 (WANG2) en un <i>cluster</i> de PC's, para 2 procesadores y $4096 \leq n \leq 65536$ . . . . .	233
5.8	Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un <i>cluster</i> de PC's, para 4 procesadores y $4096 \leq n \leq 65536$ . . . . .	234
5.9	Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un <i>cluster</i> de PC's, para 6 procesadores y $4032 \leq n \leq 64512$ . . . . .	235
5.10	Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.1 (TW), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3D, para 2 procesadores y $16384 \leq n \leq 524288$ . . . . .	237
5.11	Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3D, para 4 procesadores y $16384 \leq n \leq 524288$ . . . . .	238
5.12	Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3D, para 8 procesadores y $16384 \leq n \leq 524288$ . . . . .	239
5.13	Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3D, para 16 procesadores y $16384 \leq n \leq 524288$ . . . . .	240

- 
- 5.14 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3D, para 32 procesadores y  $16384 \leq n \leq 524288$ . . . . . 241
- 5.15 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3D, para 64 procesadores y  $16384 \leq n \leq 524288$ . . . . . 242
- 5.16 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3D, para 128 procesadores y  $16384 \leq n \leq 524288$ . . . . . 243
- 5.17 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3D, para 128 procesadores y  $16384 \leq n \leq 524288$ . . . . . 244
- 5.18 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.1 (TW), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3E, para 2 procesadores y  $16384 \leq n \leq 524288$ . . . . . 247
- 5.19 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3E, para 4 procesadores y  $16384 \leq n \leq 524288$ . . . . . 248
- 5.20 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3E, para 8 procesadores y  $16384 \leq n \leq 524288$ . . . . . 249
- 5.21 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3E, para 16 procesadores y  $16384 \leq n \leq 524288$ . . . . . 250
- 5.22 Comparación entre los algoritmos 2.2 (OPM1), 2.3 (OPM2), 3.2 (TW1), 3.3 (TW2), 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3E, para 32 procesadores y  $16384 \leq n \leq 524288$ . . . . . 251

# Índice de tablas

1.1	Valores de parámetros BSP. . . . .	31
1.2	Megaflops obtenidos para un IBM SP2 y un <i>cluster</i> de PC's. . . . .	31
1.3	Coste de comunicación en flops de una palabra de 32 bits para 4 procesadores en un IBM SP2 utilizando <i>switch</i> y distintos patrones de comunicación. . . . .	33
1.4	Valores de $g_\infty$ y $n_{\frac{1}{2}}$ para los patrones <i>local</i> , <i>trid</i> y <i>all-to-all</i> en un IBM SP2 con 4 procesadores interconectados mediante <i>switch</i> . . . . .	33
1.5	Funciones de la librería Green BSP. . . . .	41
1.6	Núcleo de la librería BSPLib. . . . .	43
2.1	Rango de la diagonal dominanza $\delta$ dependiendo del máximo error tolerado $\epsilon$ y del número de ecuaciones solapadas $m$ . . . . .	70
2.2	Valores de parámetros BSP. . . . .	81
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 <i>switch</i> , para $128 \leq n \leq 524288$ . . . . .	83
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 <i>switch</i> , para $128 \leq n \leq 524288$ . . . . .	85

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 <i>switch</i> , para $126 \leq n \leq 516096$ . . . . .	87
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 <i>ethernet</i> , para $128 \leq n \leq 524288$ . . . . .	89
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 <i>ethernet</i> , para $128 \leq n \leq 524288$ . . . . .	91
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 <i>ethernet</i> , para $126 \leq n \leq 516096$ . . . . .	93
2.9	Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un <i>cluster</i> de PC's, para 2 procesadores y $128 \leq n \leq 65536$ . . . . .	95
2.10	Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un <i>cluster</i> de PC's, para 4 procesadores y $128 \leq n \leq 65536$ . . . . .	97
2.11	Tiempos teóricos y experimentales de los algoritmos 2.2 (OPM1) y 2.3 (OPM2), medidos en un <i>cluster</i> de PC's, para 6 procesadores y $126 \leq n \leq 64512$ . . . . .	99
2.12	Algoritmo más rápido (teórica y experimentalmente) entre los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 <i>switch</i> . . . . .	105
2.13	Algoritmo más rápido (teórica y experimentalmente) entre los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 <i>ethernet</i> . . . . .	106
2.14	Algoritmo más rápido (teórica y experimentalmente) entre los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un <i>cluster</i> de PC's. . . . .	107

---

2.15	<i>Speed-up</i> ( $S_p$ ) y eficiencia ( $E_p$ ) de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un IBM SP2 y un <i>cluster</i> de PC's. . . . .	108
2.16	Valores de parámetros BSP. . . . .	109
2.17	<i>Speed-up</i> ( $S_p$ ) y eficiencia ( $E_p$ ) de los algoritmos 2.2 (OPM1) y 2.3 (OPM2) en un CRAY T3D y un CRAY T3E. . . . .	110
3.1	Valores de parámetros BSP. . . . .	142
3.2	Tiempos teóricos y experimentales del algoritmo 3.1 (TW), medidos en un IBM SP2 con 2 procesadores interconectados mediante <i>switch</i> , para $128 \leq n \leq 524288$ . . . . .	143
3.3	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2), medidos en un IBM SP2 con 4 procesadores interconectados mediante <i>switch</i> , para $128 \leq n \leq 524288$ . . . . .	145
3.4	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2), medidos en un IBM SP2 con 6 procesadores interconectados mediante <i>switch</i> , para $126 \leq n \leq 516096$ . . . . .	147
3.5	Tiempos teóricos y experimentales del algoritmo 3.1 (TW), medidos en un IBM SP2 con 2 procesadores interconectados mediante <i>ethernet</i> , para $128 \leq n \leq 524288$ . . . . .	149
3.6	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2), medidos en un IBM SP2 con 4 procesadores interconectados mediante <i>ethernet</i> , para $128 \leq n \leq 524288$ . . . . .	151
3.7	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2), medidos en un IBM SP2 con 6 procesadores interconectados mediante <i>ethernet</i> , para $126 \leq n \leq 516096$ . . . . .	153
3.8	Tiempos teóricos y experimentales del algoritmo 3.1 (TW) medidos en un <i>cluster</i> de PC's, para 2 procesadores y $128 \leq n \leq 65536$ . . . . .	156

3.9	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2) medidos en un <i>cluster</i> de PC's, para 4 procesadores y $128 \leq n \leq 65536$ . . .	158
3.10	Tiempos teóricos y experimentales de los algoritmos 3.2 (TW1) y 3.3 (TW2) medidos en un <i>cluster</i> de PC's, para 6 procesadores y $126 \leq n \leq 64512$ . . .	160
3.11	Algoritmo más rápido (teórica y experimentalmente) entre los algoritmos 3.2 (TW1) y 3.3 (TW2) en un IBM SP2 <i>switch</i> . . . . .	165
3.12	Algoritmo más rápido (teórica y experimentalmente) entre los algoritmos 3.2 (TW1) y 3.3 (TW2) en un IBM SP2 <i>ethernet</i> . . . . .	166
3.13	Algoritmo más rápido (teórica y experimentalmente) entre los algoritmos 3.2 (TW1) y 3.3 (TW2) en un <i>cluster</i> de PC's. . . . .	167
3.14	<i>Speed-up</i> ( $S_p$ ) y eficiencia ( $E_p$ ) de los algoritmos 3.1 (TW), 3.2 (TW1) y 3.3 (TW2) en un IBM SP2 y un <i>cluster</i> de PC's. . . . .	168
3.15	Valores de parámetros BSP. . . . .	169
3.16	<i>Speed-up</i> ( $S_p$ ) y eficiencia ( $E_p$ ) de los algoritmos 3.1 (TW), 3.2 (TW1) y 3.3 (TW2) en un CRAY T3D y un CRAY T3E. . . . .	170
4.1	Valores de parámetros BSP. . . . .	195
4.2	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2), medidos en un IBM SP2 con 2 procesadores interconectados mediante <i>switch</i> , para $128 \leq n \leq 524288$ . . . . .	198
4.3	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2), medidos en un IBM SP2 con 4 procesadores interconectados mediante <i>switch</i> , para $128 \leq n \leq 524288$ . . . . .	200
4.4	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2), medidos en un IBM SP2 con 6 procesadores interconectados mediante <i>switch</i> , para $126 \leq n \leq 516096$ . . . . .	202

---

4.5	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2), medidos en un IBM SP2 con 2 procesadores interconectados mediante <i>ethernet</i> , para $128 \leq n \leq 524288$ . . . . .	204
4.6	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2), medidos en un IBM SP2 con 4 procesadores interconectados mediante <i>ethernet</i> , para $128 \leq n \leq 524288$ . . . . .	206
4.7	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2), medidos en un IBM SP2 con 6 procesadores interconectados mediante <i>ethernet</i> , para $126 \leq n \leq 516096$ . . . . .	208
4.8	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) medidos en un <i>cluster</i> de PC's, para 2 procesadores y $128 \leq n \leq 65536$ . . . . .	210
4.9	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) medidos en un <i>cluster</i> de PC's, para 4 procesadores y $128 \leq n \leq 65536$ . . . . .	212
4.10	Tiempos teóricos y experimentales de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) medidos en un <i>cluster</i> de PC's, para 6 procesadores y $126 \leq n \leq 64512$ . . . . .	214
4.11	Tiempos teóricos de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) medidos en un CRAY T3D, para 256 procesadores y $512 \leq n \leq 524288$ . . . . .	219
4.12	<i>Speed-up</i> ( $S_p$ ) y eficiencia ( $E_p$ ) de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un IBM SP2 y un <i>cluster</i> de PC's. . . . .	220
4.13	Valores de parámetros BSP. . . . .	221
4.14	<i>Speed-up</i> ( $S_p$ ) y eficiencia ( $E_p$ ) de los algoritmos 4.2 (WANG1) y 4.3 (WANG2) en un CRAY T3D y un CRAY T3E. . . . .	222
5.1	Valores de parámetros BSP. . . . .	224

5.2	Número óptimo de procesadores en un CRAY T3D, para $16384 \leq n \leq 524288$ . . . . .	246
5.3	Número óptimo de procesadores en un CRAY T3E, para $16384 \leq n \leq 524288$ . . . . .	253