

---

# Algoritmos divide y vencerás para la resolución de sistemas lineales tridiagonales en un computador BSP

Leandro Tortosa Grau

## 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



Algoritmos divide y vencerás para la  
resolución de sistemas lineales tridiagonales  
en un computador BSP

TESIS DOCTORAL

Presentada por: Leandro Tortosa Grau

Dirigida por: Joan Josep Climent Coloma



**Universidad de Alicante**

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

**Algoritmos divide y vencerás para la resolución de sistemas lineales tridiagonales  
en un computador BSP**

Memoria presentada para optar al grado de doctor Ingeniero en In-  
formática por LEANDRO TORTOSA GRAU.

Alicante, 14 de Diciembre de 1999



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

CERTIFICA:

Que la presente memoria *Algoritmos divide y vencerás para la resolución de sistemas lineales tridiagonales en un computador BSP*, ha sido realizada bajo su dirección, en el Departamento de Ciencia de la Computaci' on e Inteligencia Artificial de la Universidad de Alicante, por el licenciado Don LEANDRO TORTOSA GRAU, y constituye su tesis para optar al grado de Doctor en 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 Carmen, su apoyo constante y confianza en mí  
me han proporcionado las fuerzas necesarias  
para seguir hacia adelante.*



En primer lugar, quiero expresar mi agradecimiento al profesor Joan Josep Climent por todos los conocimientos, consejos y amistad que me ha transmitido a lo largo de todos estos años. Su paciencia y habilidad han sido determinantes para que este trabajo llegara a buen puerto, así como su enorme capacidad de trabajo y disciplina.

En segundo lugar, agradecer a Antonio Zamora su amistad y comprensión diaria. Ha sido el compañero perfecto en tan larga travesía. Juntos hemos celebrado las alegrías y hemos sufrido las decepciones propias del trabajo diario.

Una persona a la que admiro mucho como profesional y como persona es Natividad Llorca, a la que conocí en Novelda cuando comenzaba mi carrera docente y, por tanto, mi experiencia era nula. Ella me puso en contacto con el profesor Joan Josep Climent.

Quiero también agradecer a algunas personas la confianza y apoyo que me han transmitido, lo que ha resultado fundamental para mí, pues me han dado ánimos para superar los momentos difíciles. En primer lugar está mi familia, especialmente mi padre, con el que siempre puedo contar y mis suegros, a los que aprovecho para agradecer que siempre me hayan tratado como a un hijo más. Su apoyo y confianza en mí han sido muy importantes.

Mi labor profesional durante estos últimos años se ha desarrollado en el I.E.S. Haygón de Sant Vicent del Raspeig y la Universidad de Alicante. En la Universidad formo parte del grupo de Computación Paralela del Departamento de Ciencia de la Computación e Inteligencia Artificial. Agradezco a todos los miembros del mismo el clima de amistad y buen ambiente del cual he disfrutado estos últimos años.

En el Instituto Haygón he encontrado unos compañeros y amigos que han contribuido a que mi labor docente e investigadora se desarrollara en un ambiente óptimo. El director del mismo, Jorge Mateo, siempre me ha apoyado y facilitado la asistencia a las Jornadas

y Conferencias en las que he participado estos últimos años. Un buen amigo es Angel Luis García, profesor de Inglés y compañero desde hace bastantes años. Su ingenio nunca deja de sorprenderme. A él he acudido en diversas ocasiones y siempre me ha demostrado su amistad y generosidad. Desde el verano pasado desempeño la labor de secretario del Centro. Todos los compañeros me han ayudado mucho en esta labor completamente nueva para mí. Quiero destacar a una persona que sufre diariamente mis preocupaciones, mis despistes y mi desorganización. Es María Dolores Belda, a la que quiero agradecer su capacidad de trabajo y, por encima de todo, la amistad que me demuestra día a día.

Dos personas trabajan conmigo en temas de educación matemática y tecnología en la enseñanza de las matemáticas. Son Javier Santacruz y Rosario Martín. Juntos hemos compartido algún largo viaje y lo hemos pasado bien juntos. Su amistad es muy importante para mí.

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

# Índice

Lista de tablas	viii
Lista de figuras	xii
Prólogo	xiii
<b>1 Planteamiento del problema</b>	<b>1</b>
1.1 Preliminares . . . . .	1
1.2 Algoritmos del tipo divide y vencerás . . . . .	7
1.3 Métodos de resolución de sistemas lineales tridiagonales . . . . .	12
1.3.1 Métodos <i>divide y vencerás</i> para sistemas tridiagonales . . . . .	13

---

1.3.2	Método del <i>recursive doubling</i> . . . . .	15
1.3.3	Método de reducción cíclica . . . . .	19
1.4	Métodos <i>divide y vencerás</i> para calcular la inversa de una matriz . . . . .	27
1.5	Breve sumario de aplicaciones de los sistemas tridiagonales . . . . .	28
<b>2</b>	<b>El modelo BSP</b>	<b>31</b>
2.1	Breve introducción a la computación paralela . . . . .	31
2.2	El modelo de computación BSP . . . . .	35
2.3	El modelo de programación . . . . .	39
2.4	El modelo de coste . . . . .	40
2.5	Valores de $s$ , $g$ y $l$ en distintas máquinas . . . . .	42
<b>3</b>	<b>Algoritmos <i>divide y vencerás</i> basados en la fórmula de Sherman-Morrison</b>	<b>47</b>
3.1	Introducción . . . . .	47
3.2	Un algoritmo recursivo para sistemas tridiagonales . . . . .	49
3.3	Un algoritmo BSP de tipo <i>fan-in</i> . . . . .	65

---

3.3.1	Descripción del algoritmo . . . . .	65
3.3.2	Coste computacional . . . . .	73
3.4	Un algoritmo BSP basado en el método <i>recursive doubling</i> . . . . .	78
3.4.1	Descripción del algoritmo . . . . .	78
3.4.2	Coste computacional . . . . .	82
3.5	Resultados teóricos y comparaciones . . . . .	83
3.5.1	Tiempos en un IBM SP2 . . . . .	83
3.5.2	Tiempos en un CRAY T3D . . . . .	88
3.5.3	Tiempos en un cluster de Pentiums . . . . .	91
3.5.4	Estudio del <i>speedup</i> . . . . .	93
<b>4</b>	<b>Métodos divide y vencerás basados en la fórmula de Sherman-Morrison-Woodbury</b>	<b>101</b>
4.1	Introducción . . . . .	101
4.2	Un algoritmo general del tipo <i>divide y vencerás</i> . . . . .	104
4.2.1	Descripción del método . . . . .	104
4.2.2	Coste computacional . . . . .	117

---

4.3	Un algoritmo BSP del tipo <i>divide y vencerás</i> general . . . . .	118
4.3.1	Descripción del método . . . . .	118
4.3.2	Coste computacional . . . . .	121
4.4	Algoritmos BSP <i>divide y vencerás</i> utilizando el método <i>recursive doubling</i> . . . . .	123
4.4.1	Descripción del método . . . . .	124
4.4.2	Coste computacional . . . . .	132
4.5	Algoritmo BSP <i>divide y vencerás</i> basado en el método de reducción cíclica . . . . .	138
4.5.1	Descripción del método . . . . .	138
4.5.2	Coste computacional . . . . .	143
4.6	Resultados teóricos y comparaciones . . . . .	146
4.6.1	Tiempos en un IBM SP2 . . . . .	147
4.6.2	Tiempos en un CRAY T3D . . . . .	148
4.6.3	Tiempos en un Cluster de Pentiums . . . . .	154
4.6.4	Estudio del speedup . . . . .	154

---

5.1	Introducción . . . . .	163
5.2	Descripción del método . . . . .	166
5.3	Algoritmos BSP divide y vencerás . . . . .	176
5.3.1	Algoritmo BSP basado en el método de Bondeli . . . . .	176
5.3.2	Coste computacional . . . . .	180
5.3.3	Algoritmo BSP modificado basado en el método de Bondeli . . . . .	182
5.3.4	Coste computacional . . . . .	185
5.3.5	Algoritmo BSP utilizando el método <i>recursive doubling</i> . . . . .	187
5.3.6	Coste computacional . . . . .	191
5.4	Resultados teóricos y comparaciones . . . . .	193
5.4.1	Tiempos en un IBM SP2 . . . . .	193
5.4.2	Tiempos en un CRAY T3D . . . . .	196
5.4.3	Tiempos en un cluster de Pentiums . . . . .	199
5.4.4	Estudio del speedup . . . . .	202

---

6.1	Algoritmos <i>divide y vencerás</i> en el IBM SP2 . . . . .	209
6.2	Algoritmos <i>divide y vencerás</i> en el CRAY T3D . . . . .	219
6.3	Algoritmos <i>divide y vencerás</i> en un cluster de Pentiums . . . . .	226
<b>A</b>	<b>Conclusiones y líneas futuras de trabajo</b>	<b>231</b>
	<b>Bibliografía</b>	<b>235</b>

# Índice de Tablas

2.1	Valores de los parámetros $s$ , $g$ , $l$ y $n_{1/2}$ . . . . .	44
3.1	Tiempos teóricos para los algoritmos 3.3, 3.4 y 3.5 medidos en un IBM SP2 dotado con un switch de alto rendimiento y conexión ethernet. . . . .	84
3.2	Diferencias de tiempos entre los algoritmos 3.4 y 3.5, medidos en porcentajes. . . . .	86
3.3	Tiempos teóricos para los algoritmos 3.3, 3.4 y 3.5 medidos en un CRAY T3D para 2, 4, 8, 16, 32, 64, 128 y 256 procesadores. . . . .	88
3.4	Número óptimo de procesadores y diferencias de tiempos para la ejecución de los algoritmos 3.4 y 3.5 en un CRAY T3D. . . . .	90
3.5	Tiempos teóricos y diferencias de tiempos para los algoritmos 3.3, 3.4 y 3.5 en un cluster de Pentiums. . . . .	92
3.6	Speedup y eficiencia de los algoritmos 3.3, 3.4 y 3.5 para $n = 2097152$ . . . . .	94
4.1	Tiempos teóricos para los algoritmos 4.2, 4.3, 4.4 y 4.5 en un IBM SP2 con switch. . . . .	147

---

4.2	Tiempos teóricos para los algoritmos 4.2, 4.3, 4.4 y 4.5 en un IBM SP2 con ethernet. . . . .	149
4.3	Tiempos teóricos para los algoritmos 4.2, 4.3, 4.4 y 4.5 en un CRAY T3D desde $n = 2048$ hasta $n = 65536$ . . . . .	150
4.4	Tiempos teóricos para los algoritmos 4.2, 4.3, 4.4 y 4.5 en un CRAY T3D desde $n = 131072$ hasta $n = 4194304$ . . . . .	151
4.5	Número óptimo de procesadores en un CRAY T3D para valores de $n$ comprendidos entre $n = 2048$ y $n = 4194304$ . . . . .	153
4.6	Tiempos teóricos para los algoritmos 4.2, 4.3, 4.4 y 4.5 en un cluster de Pentiums. . . . .	155
4.7	Speedup y eficiencia del algoritmo Alg. 4.2 para una matriz de tamaño $n = 2097152$ . . . . .	157
5.1	Tiempos teóricos para los algoritmos 5.1, 5.2 y 5.3 medidos en un IBM SP2 dotado con un switch de alto rendimiento. . . . .	194
5.2	Tiempos teóricos para los algoritmos 5.1, 5.2 y 5.3 medidos en un IBM SP2 utilizando conexión ethernet. . . . .	195
5.3	Tiempos teóricos para los algoritmos 5.1, 5.2 y 5.3 medidos en un CRAY T3D utilizando 2, 4, 8 y 16 procesadores. . . . .	197
5.4	Tiempos teóricos para los algoritmos 5.1, 5.2 y 5.3 medidos en un CRAY T3D para 32, 64, 128 y 256 procesadores. . . . .	198
5.5	Número óptimo de procesadores en un CRAY T3D para tamaños de matrices comprendidos entre $n = 2048$ y $n = 4194304$ . . . . .	200
5.6	Tiempos teóricos para los algoritmos 5.1, 5.2 y 5.3 medidos en un Cluster de Pentiums para 2, 4 y 8 procesadores. . . . .	201
5.7	Speedup y eficiencia de los algoritmos 5.1, 5.2 y 5.3 para $n = 2097152$ . . . . .	203

# Índice de Figuras

2.1	Algoritmo <i>fan-in</i> para la suma de ocho números. . . . .	34
3.1	Inicialización y actualización del algoritmo 3.2 para el ejemplo 3.2. . . . .	62
3.2	Esquema del algoritmo 3.4 para $n = 32$ y $p = 4$ . . . . .	70
3.3	Inicialización y actualizaciones que realiza el procesador $P_0$ cuando ejecuta el algoritmo 3.4 para el ejemplo 3.2. . . . .	72
3.4	Esquema del algoritmo 3.4 para el caso $n = 32$ y $p = 4$ . . . . .	80
3.5	Valores del <i>speedup</i> en un IBM SP2 . . . . .	96
3.6	Valores del <i>speedup</i> en un CRAY y un cluster de Pentiums . . . . .	97
3.7	Valores de la eficiencia en un IBM SP2 . . . . .	98
3.8	Valores del <i>speedup</i> en un CRAY y un cluster de Pentiums . . . . .	99

---

4.1	Esquema de ejecución del algoritmo 4.2, para 4 procesadores. . . . .	120
4.2	Esquema de comunicaciones del método divide y vencerás basado en el método recursive doubling, para $p = 16$ . . . . .	127
4.3	Esquema de ejecución del algoritmo 4.4, para 8 procesadores. . . . .	133
4.4	Esquema de ejecución del algoritmo (4.5), para 8 procesadores. . . . .	141
4.5	Valores del <i>speedup</i> en un IBM SP2 . . . . .	158
4.6	Valores del <i>speedup</i> en un CRAY T3D y un cluster de Pentiums . . . . .	159
4.7	Valores de la eficiencia en un IBM SP2 . . . . .	160
4.8	Valores de la eficiencia en un CRAY T3D y un cluster de Pentiums . . . . .	161
5.1	Esquema de ejecución del algoritmo 5.1, para 4 procesadores. . . . .	179
5.2	Esquema de ejecución del algoritmo 5.2, para 4 procesadores. . . . .	184
5.3	Valores del <i>speedup</i> en un IBM SP2 . . . . .	204
5.4	Valores del <i>speedup</i> en un CRAY T3D y un cluster de Pentiums . . . . .	205
5.5	Valores de la eficiencia en un IBM SP2 . . . . .	206
5.6	Valores de la eficiencia en un CRAY T3D y un cluster de Pentiums . . . . .	207

---

6.1	Tiempos en un IBM SP2 con switch para $n = 512, n = 1024, n = 2048$ y $n = 4096$ . . . . .	211
6.2	Tiempos en un IBM SP2 con switch para $n = 8192, n = 16384, n = 32768$ y $n = 65536$ . . . . .	212
6.3	Tiempos en un IBM SP2 con switch para $n = 8192, n = 16384, n = 32768$ y $n = 65536$ . . . . .	213
6.4	Tiempos en un IBM SP2 con switch para $n = 2097152$ y $n = 4194304$ . . . . .	214
6.5	Tiempos en un IBM SP2 con ethernet para $n = 512, n = 1024, n = 2048$ y $n = 4096$ . . . . .	215
6.6	Tiempos en un IBM SP2 con ethernet para $n = 8192, n = 16384, n = 32768$ y $n = 65536$ . . . . .	216
6.7	Tiempos en un IBM SP2 con ethernet para $n = 131072, n = 262144, n = 524288$ y $n = 1048576$ . . . . .	217
6.8	Tiempos en un IBM SP2 con ethernet para $n = 2097152$ y $n = 4194304$ . . . . .	218
6.9	Tiempos en un CRAY T3D para $n = 2048$ y $n = 4096$ . . . . .	220
6.10	Tiempos en un CRAY T3D para $n = 8192$ y $n = 16384$ . . . . .	221
6.11	Tiempos en un CRAY T3D para $n = 32768$ y $n = 65536$ . . . . .	222
6.12	Tiempos en un CRAY T3D para $n = 131072$ y $n = 262144$ . . . . .	223
6.13	Tiempos en un CRAY T3D para $n = 524288$ y $n = 1048576$ . . . . .	224
6.14	Tiempos en un CRAY T3D para $n = 2097152$ y $4194304$ . . . . .	225
6.15	Tiempos en un cluster de Pentiums para $n = 512, n = 1024, n = 2048$ y $n = 4096$ . . . . .	227

6.16 Tiempos en un cluster de Pentiums para  $n = 8192$ ,  $n = 16384$ ,  $n = 32768$  y  $n = 65536$  . . . . . 228

6.17 Tiempos en un cluster de Pentiums para  $n = 8192$ ,  $n = 16384$ ,  $n = 32768$  y  $n = 65536$  . . . . . 229

6.18 Tiempos en un cluster de Pentiums para  $n = 2097152$  y  $n = 4194304$  . . . . . 230