

# Parallel Scientific Computation

*A structured approach using BSP and MPI*

ROB H. BISSELING

*Utrecht University*

**OXFORD**  
UNIVERSITY PRESS

# CONTENTS

<b>1 Introduction</b>	1
1.1 Wanted: a gentle parallel programming model	1
1.2 The BSP model	3
1.3 BSP algorithm for inner product computation	9
1.4 Starting with BSPlib: example program <code>bspinprod</code>	13
1.5 BSP benchmarking	24
1.6 Example program <code>bspbench</code>	27
1.7 Benchmark results	31
1.8 Bibliographic notes	38
1.8.1 BSP-related models of parallel computation	38
1.8.2 BSP libraries	40
1.8.3 The non-BSP world: message passing	42
1.8.4 Benchmarking	43
1.9 Exercises	44
<b>2 LU decomposition</b>	50
2.1 The problem	50
2.2 Sequential LU decomposition	51
2.3 Basic parallel algorithm	57
2.4 Two-phase broadcasting and other improvements	64
2.5 Example function <code>bsplu</code>	72
2.6 Experimental results on a Cray T3E	79
2.7 Bibliographic notes	85
2.7.1 Matrix distributions	85
2.7.2 Collective communication	87
2.7.3 Parallel matrix computations	87
2.8 Exercises	88
<b>3 The fast Fourier transform</b>	100
3.1 The problem	100
3.2 Sequential recursive fast Fourier transform	103
3.3 Sequential nonrecursive algorithm	105
3.4 Parallel algorithm	113
3.5 Weight reduction	120
3.6 Example function <code>bspfft</code>	127
3.7 Experimental results on an SGI Origin 3800	136
3.8 Bibliographic notes	145
3.8.1 Sequential FFT algorithms	145

3.8.2	Parallel FFT algorithms with $\log_2 p$ or more supersteps	147
3.8.3	Parallel FFT algorithms with $\mathcal{O}(1)$ supersteps	148
3.8.4	Applications	151
3.9	Exercises	152
<b>4</b>	<b>Sparse matrix–vector multiplication</b>	<b>163</b>
4.1	The problem	163
4.2	Sparse matrices and their data structures	167
4.3	Parallel algorithm	173
4.4	Cartesian distribution	179
4.5	Mondriaan distribution for general sparse matrices	186
4.6	Vector distribution	197
4.7	Random sparse matrices	203
4.8	Laplacian matrices	210
4.9	Remainder of BSPlib: example function <code>bspmv</code>	222
4.10	Experimental results on a Beowulf cluster	231
4.11	Bibliographic notes	235
4.11.1	Sparse matrix computations	235
4.11.2	Parallel sparse matrix–vector multiplication algorithms	237
4.11.3	Parallel iterative solvers for linear systems	239
4.11.4	Partitioning methods	240
4.12	Exercises	243
<b>A</b>	<b>Auxiliary BSPedupack functions</b>	<b>251</b>
A.1	Header file <code>bspedupack.h</code>	251
A.2	Utility file <code>bspedupack.c</code>	251
<b>B</b>	<b>A quick reference guide to BSPlib</b>	<b>254</b>
<b>C</b>	<b>Programming in BSP style using MPI</b>	<b>256</b>
C.1	The message-passing interface	256
C.2	Converting BSPedupack to MPIedupack	258
C.2.1	Program <code>mpiinprod</code>	258
C.2.2	Program <code>mpibench</code>	261
C.2.3	Function <code>mpilu</code>	265
C.2.4	Function <code>mpiffit</code>	270
C.2.5	Function <code>mpimv</code>	273
C.3	Performance comparison on an SGI Origin 3800	278
C.4	Where BSP meets MPI	280
	<b>References</b>	<b>283</b>
	<b>Index</b>	<b>299</b>