

Computational Science, Engineering and Technology Series: 24

**Substructuring Techniques
and
Domain Decomposition Methods**

Computational Science, Engineering and Technology Series:

Parallel, Distributed and Grid Computing for Engineering

Edited by: B.H.V. Topping and P. Iványi

Trends in Engineering Computational Technology

Edited by: M. Papadrakakis and B.H.V. Topping

Trends in Computational Structures Technology

Edited by: B.H.V. Topping and M. Papadrakakis

Computational Methods for Acoustics Problems

Edited by: F. Magoulès

Mesh Partitioning Techniques and Domain Decomposition Methods

Edited by: F. Magoulès

Computational Mechanics using High Performance Computing

Edited by: B.H.V. Topping

High Performance Computing for Computational Mechanics

Edited by: B.H.V. Topping, L. Lämmer

Parallel and Distributed Processing for Computational Mechanics: Systems and Tools

Edited by: B.H.V. Topping

Saxe-Coburg Publications:

Programming Distributed Finite Element Analysis:

An Object Oriented Approach

R.I. Mackie

Object Oriented Methods and Finite Element Analysis

R.I. Mackie

Domain Decomposition Methods for Distributed Computing

J. Kruis

Computer Aided Design of Cable-Membrane Structures

B.H.V. Topping and P. Iványi

**Substructuring Techniques
and
Domain Decomposition Methods**

Edited by
F. Magoulès



© Saxe-Coburg Publications, Kippen, Stirling, Scotland

published 2010 by

Saxe-Coburg Publications

Civil-Comp Ltd, Dun Eaglais, Station Brae

Kippen, Stirlingshire, FK8 3DY, Scotland

Saxe-Coburg Publications is an imprint of Civil-Comp Ltd

Computational Science, Engineering and Technology Series: 24

ISSN 1759-3158

ISBN: 978-1-874672-33-3

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

Front cover: Domain Decomposition for 50 subdomains by T. Miyamura. For more information see Chapter 7.

Printed in Great Britain by Bell & Bain Ltd, Glasgow

Contents

| | |
|---|-----------|
| Preface | ix |
| 1 Substructuring and Domain Decomposition Methods: An Overview | 1 |
| F. Magoulès and D. Rixen | |
| 1.1 Domain Decomposition and Substructuring: Where do we go? | 1 |
| 1.1.1 “ <i>divide et impera!</i> ” | 2 |
| 1.1.2 Substructuring or Domain Decomposition? | 2 |
| 1.1.3 Key Issues for the Computational Mechanics Community | 3 |
| 1.1.3.1 Mathematical Properties | 4 |
| 1.1.3.2 Heterogeneous Problems | 4 |
| 1.1.3.3 Multi-Physics and Multi-Models | 4 |
| 1.1.3.4 Multi-Point Constraints | 5 |
| 1.1.3.5 Non-Linearities and Contact | 5 |
| 1.2 The Contributions in this Book | 5 |
| 2 The ‘Parareal in Time’ Algorithm | 19 |
| Y. Maday | |
| 2.1 Introduction | 19 |
| 2.2 Presentation of the ‘Parareal in Time’ Algorithm | 21 |
| 2.2.1 Basics on solution for ODE’s | 21 |
| 2.2.2 The ‘Parareal in Time’ Algorithm | 22 |
| 2.2.3 Numerical Simulation | 23 |
| 2.2.4 Complexity and Parallelism Efficiency: Part 1 | 24 |
| 2.2.5 An Algebraic Interpretation | 26 |
| 2.3 Various Ways to Reduce the Coarse Simulation | 27 |
| 2.3.1 Convergence Analysis | 27 |
| 2.3.2 Model Reduction | 29 |
| 2.3.3 Coarse Spacial Grid | 30 |
| 2.4 More on Iterative Solvers | 32 |
| 2.4.1 Coupling with Domain Decomposition: The Overlapping Case | 33 |
| 2.4.1.1 The Problem and the Decomposition of the Domain | 33 |

| | | |
|---------|---|----|
| 2.4.1.2 | The Iterative Procedure | 33 |
| 2.4.1.3 | The Coarse Propagator | 34 |
| 2.4.1.4 | The Fine Propagator | 34 |
| 2.4.1.5 | The Numerical Results | 35 |
| 2.4.2 | Coupling with Control | 35 |
| 2.4.2.1 | The Virtual Control Paradigm | 35 |
| 2.4.2.2 | The Parareal Algorithm for Control Problem | 37 |
| 2.4.2.3 | Another Route to Couple the Parareal Algorithm with a Control Strategy | 39 |
| 2.4.3 | Complexity and Parallelism Efficiency: Part 2 | 39 |
| 2.5 | Stability Issues | 40 |
| 2.5.1 | Basic Results | 40 |
| 2.5.2 | A Possible Strategy for Convection Dominated Problems | 40 |
| 2.6 | Conclusions | 42 |

3 On Transformation Methods and the Induced Parallel Properties for the Temporal Domain 45

| | | |
|-----------|---|----|
| C.-H. Lai | | |
| 3.1 | Introduction | 46 |
| 3.2 | Temporal Integration Methods | 47 |
| 3.3 | Transformation Methods | 48 |
| 3.3.1 | Similarity Transforms | 49 |
| 3.3.2 | Laplace Transforms | 50 |
| 3.4 | Decoupled Parametric Systems | 51 |
| 3.4.1 | Similarity Transformation Methods | 51 |
| 3.4.2 | Laplace Transformation Methods | 53 |
| 3.5 | Some Numerical Tests | 54 |
| 3.5.1 | A Nonlinear Parabolic Problem | 54 |
| 3.5.2 | An European Option Example | 57 |
| 3.5.3 | Testing a Two-Level Time-Domain Algorithm | 65 |
| 3.6 | Conclusion | 68 |

4 Asynchronous Substructuring Methods 71

| | | |
|--------------------------|--|----|
| C. Venet and F. Magoulès | | |
| 4.1 | Introduction | 71 |
| 4.2 | Synchronous Iterative Methods | 73 |
| 4.2.1 | Iterative Methods | 73 |
| 4.2.2 | Implementation | 75 |
| 4.3 | Asynchronous Iterative Methods | 76 |
| 4.3.1 | Asynchronous Iteration | 76 |
| 4.3.2 | Mathematical Model | 78 |

| | | |
|-------|---|-----|
| 4.3.3 | Convergence Analysis | 79 |
| 4.3.4 | Implementation | 81 |
| 4.4 | Substructuring Methods | 81 |
| 4.4.1 | Principles | 81 |
| 4.4.2 | Asynchronous Substructuring | 84 |
| 4.5 | Programming Environment | 92 |
| 4.5.1 | Presentation of the JACK Architecture | 93 |
| 4.5.2 | Implementation | 94 |
| 4.5.3 | Convergence Detection | 96 |
| 4.6 | Numerical Results | 96 |
| 4.7 | Conclusion | 102 |

5 Asynchronous Multi-Splitting Methods 105

J.M. Bahi, R. Couturier and D. Laiymani

| | | |
|---------|---|-----|
| 5.1 | Introduction | 106 |
| 5.2 | Parallel Numerical Iterative Algorithms | 107 |
| 5.2.1 | Generalities | 107 |
| 5.2.2 | Parallel Architectures | 108 |
| 5.2.3 | A Classification of Parallel Iterative Algorithms | 109 |
| 5.3 | Multisplitting Algorithms | 113 |
| 5.3.1 | Generalities | 114 |
| 5.3.2 | Linear Case | 114 |
| 5.3.2.1 | Implementation | 116 |
| 5.3.3 | Non-Linear Case | 119 |
| 5.3.3.1 | Multisplitting-Newton method | 121 |
| 5.3.3.2 | Implementation | 121 |
| 5.4 | The Jace Environment | 122 |
| 5.4.1 | The Daemon | 123 |
| 5.4.2 | The Spawner | 124 |
| 5.4.3 | The Worker | 124 |
| 5.4.3.1 | The Application Layer | 125 |
| 5.4.3.2 | The Communication Layer | 125 |
| 5.4.4 | The User Application | 125 |
| 5.5 | Experiments | 126 |
| 5.5.1 | The Grid'5000 Testbed | 126 |
| 5.5.2 | The linear case | 126 |
| 5.5.2.1 | The Conjugate Gradient NAS Parallel Benchmark | 127 |
| 5.5.2.2 | Experiments Conditions | 127 |
| 5.5.2.3 | Local Context | 128 |
| 5.5.2.4 | Distant Context | 129 |
| 5.5.3 | The Non-Linear Case | 131 |
| 5.5.3.1 | The Application | 131 |

| | | |
|---------|--|-----|
| 5.5.3.2 | Experimental Results | 133 |
| 5.6 | Conclusions and Perspectives | 134 |

6 A Coarse Grid Conjugate Gradient Method and its Application to Realistic Problems: Fully Iterative Method with Coarse Grid 137

| | | |
|----------|--|-----|
| H. Akiba | | |
| 6.1 | Introduction | 137 |
| 6.2 | The Iterative Substructuring Method | 139 |
| 6.2.1 | Basic description | 139 |
| 6.2.2 | Domain Decomposition and the Iterative Substructuring Method | 140 |
| 6.2.3 | Preconditioning the Iterative Substructuring Method and the Schwarz Form | 143 |
| 6.2.4 | Formalization of the Iterative Substructuring Method | 144 |
| 6.2.4.1 | Discrete Harmonic Expansion | 144 |
| 6.2.4.2 | K -Orthogonal Decomposition of Global Space | 145 |
| 6.2.4.3 | Solution of the Stiffness Equations using K -Orthogonal Decomposition | 147 |
| 6.2.4.4 | The Schur Complement and the Conjugate Gradient Method | 148 |
| 6.3 | Preconditioning of the Iterative Substructuring Method | 149 |
| 6.3.1 | Equations of the Substructure | 149 |
| 6.3.2 | Application of the Schwarz Form | 152 |
| 6.3.3 | Separation of the Degrees of Freedom in the Subdomains | 152 |
| 6.3.4 | The Neumann Method | 155 |
| 6.3.5 | The Balancing Domain Decomposition Method | 157 |
| 6.3.5.1 | The Balancing Domain Algorithm | 157 |
| 6.3.5.2 | Construction of the Coarse Grid | 158 |
| 6.3.5.3 | The Procedure of the Balancing Domain Decomposition method | 160 |
| 6.3.5.4 | Preconditioning the Balancing Domain Decomposition Method | 161 |
| 6.3.6 | The Coarse Grid Conjugate Gradient Method | 162 |
| 6.3.6.1 | The Coarse Grid Conjugate Gradient Algorithm | 162 |
| 6.3.6.2 | Construction of the Coarse Grid | 162 |
| 6.3.6.3 | The Procedure for the Coarse Grid Conjugate Gradient Method | 163 |
| 6.3.6.4 | The Preconditioner for the Coarse Grid Conjugate Gradient Method | 164 |
| 6.4 | The Performance of the Coarse Grid Conjugate Gradient Method | 164 |
| 6.4.1 | The ADVC Code | 164 |
| 6.4.2 | Comparison of the BDD and CGCG methods | 165 |

| | | |
|-------|--|-----|
| 6.4.3 | Drop Impact Analysis of a Mobile Phone using the CGCG method | 166 |
| 6.5 | Conclusions | 168 |

7 Incorporation of Multipoint Constraints into Domain Decomposition Methods 171

| | | |
|-----|---|-----|
| | T. Miyamura | |
| 7.1 | Introduction | 171 |
| 7.2 | The Domain Decomposition Method | 173 |
| | 7.2.1 Overview | 173 |
| | 7.2.2 A Domain Decomposition Method based on the Primal Substructuring Method | 175 |
| 7.3 | Incorporation of Multipoint Constraints into the Primal Substructuring Method | 176 |
| | 7.3.1 Classification of Multipoint Constraints in a Domain Decomposed Mesh | 176 |
| | 7.3.2 Modification of the Interface Problem | 177 |
| | 7.3.3 The Preconditioned Conjugate Projected Gradient Method | 178 |
| | 7.3.4 Treatment of the Dirichlet Boundary Conditions on the Interface Nodes | 179 |
| 7.4 | Incorporation of Multipoint Constraints into the Balancing Domain Decomposition Method | 180 |
| | 7.4.1 The Balancing Domain Decomposition Method | 180 |
| | 7.4.2 Incorporation of Multipoint Constraints into the Coarse Grid Problem | 181 |
| | 7.4.3 Combination of the Preconditioned Conjugate Projected Gradient Method and the Balancing Domain Decomposition Method | 182 |
| 7.5 | Parallel Implementation | 182 |
| | 7.5.1 Programming Environment | 182 |
| | 7.5.2 Implementation of the Boundary Domain Decomposition Method | 182 |
| | 7.5.3 The I/O format of the Multipoint Constraint Data | 183 |
| | 7.5.4 Parallel Implementation | 184 |
| 7.6 | Illustrative Examples | 188 |
| | 7.6.1 Hexahedron with a Constrained Face | 189 |
| | 7.6.2 Mechanical Parts in Contact | 190 |
| | 7.6.3 Parallel Performance | 197 |
| | 7.6.4 A Large-Scale Problem | 197 |
| 7.7 | Concluding Remarks | 198 |

8 Efficient Approximate Inverse Preconditioning Techniques for Reduced Systems on Parallel Computers **203**

K. Moriya, L. Zhang and T. Nodera

- 8.1 Introduction 204
- 8.2 Reduced System with the Schur Complement 205
 - 8.2.1 Greedy Algorithm 205
 - 8.2.2 Making the Schur Complement 207
- 8.3 Preconditioning Techniques of Approximate Inverse 208
 - 8.3.1 Newton Scheme 208
 - 8.3.2 The Minimum Residual Scheme 210
 - 8.3.3 The Approximate Inverse with the Sherman-Morrison Formulae Scheme 211
- 8.4 Parallelization 214
 - 8.4.1 Implementation of the Reduced Systems 214
 - 8.4.2 Implementation of the Preconditioner 215
- 8.5 Numerical Experiments 217
 - 8.5.1 Example 1: A Partial Differential Equation Problem 217
 - 8.5.2 Example 2: Matrix Market Problems 218
- 8.6 Concluding Remarks 223

9 Finite Element Matrices in Congruent Subdomains and some Techniques for Practical Problems **229**

A. Suzuki and M. Tabata

- 9.1 Introduction 229
- 9.2 Finite Element Matrices under Orthogonal Transformations 231
 - 9.2.1 An Affine Transformation with an Orthogonal Matrix 231
 - 9.2.2 Finite Element Bases and Matrices 232
 - 9.2.2.1 Transformation of Scalar-Valued Finite Element Bases and Matrices 233
 - 9.2.2.2 Transformation of Vector-Valued Finite Element Bases and Matrices 234
- 9.3 Finite Element Matrices Obtained from Decompositions into Congruent Subdomains 236
 - 9.3.1 A Domain Decomposition into a Union of Congruent Subdomains 237
 - 9.3.2 Representation of Total Matrices by Sub-Matrices 238
- 9.4 Finite Element Matrices under a Class of Orthogonal Transformations 240
 - 9.4.1 Transformation of Vector-Valued Finite Element Bases and Matrices 240
 - 9.4.2 Representation of Total Matrices by Sub-Matrices 241
- 9.5 Algorithms of Matrix-vector Multiplication for the Iterative Solver . 242
 - 9.5.1 General Orthogonal Transformations 242

| | | |
|-------|--|------------|
| 9.5.2 | A Class of Orthogonal Transformations | 243 |
| 9.6 | Treatment of Boundary Conditions | 244 |
| 9.6.1 | Examples from Continuum Mechanics | 244 |
| 9.6.2 | Weak Formulations and Finite Element Equations | 246 |
| 9.6.3 | Orthogonal Projection for Essential Boundary Conditions | 248 |
| 9.6.4 | Orthogonal Projection for Periodic Boundary Conditions | 251 |
| 9.6.5 | An Algorithm of Matrix-Vector Multiplication for Periodic Boundary Conditions | 254 |
| 9.7 | Reduction of Memory Requirements by Congruent Subdomains | 255 |
| 9.7.1 | Congruent Decomposition of a Cylinder | 255 |
| 9.7.2 | Reduction of Memory Requirements with Compressed Row Storage Format | 256 |
| 9.8 | Parallel Computation | 259 |
| 9.8.1 | Parallel Algorithm of Matrix-vector Multiplication | 259 |
| 9.8.2 | Parallel Implementation with OpenMP | 262 |
| 9.8.3 | Numerical Results | 263 |
| 9.9 | Conclusions | 264 |
| | Index | 267 |

Preface

Substructuring and domain decomposition methods are well suited for parallel computations. Indeed, the division of a problem into smaller subproblems, through artificial subdivisions of the domain, is a means for introducing parallelism. Substructuring and domain decomposition strategies include in one way or another the following ingredients: a decomposer to split a mesh into subdomains using different heuristics; local solvers (direct or iterative, exact or approximate) to find solutions for the subdomains for specific boundary conditions on the interface; interface conditions (weak or strong) enforcing compatibility and equilibrium between overlapping or non-overlapping subdomains; and a solution strategy for the interface problem. The differences between the methods lie in how those ingredients are actually put to work and how they are combined to form an efficient solution strategy for the problem at hand.

The golden age of domain decomposition probably came with the emergence of parallel computing: in order to efficiently use the computational power of several processors simultaneously first one has attempted to use additive Schwarz iterations or iterative solvers on the entire domain, using domain decomposition to dispatch the cost of matrix multiplications on the processors. Then Schur complement approaches (primal and dual) were developed: combining efficient solution techniques for the local problems with robust iterative algorithms to solve the interface problem one could build strategies to use the full potential of multiprocessor machines. Those methods were further developed to improve their robustness for practical engineering problems. Even today, despite the many reported successful applications of domain decomposition, making the iterative solution of the interface problem robust and efficiently parallel remains a challenge.

This edited book continues the series on Domain Decomposition Methods for engineering problems and follows the two previous volumes respectively entitled:

- *Mesh Partitioning Techniques and Domain Decomposition Methods (2007)*, edited by F. Magoulès, published by Saxe-Coburg Publications, Stirling, United Kingdom; and
- *Domain Decomposition Methods: Theory and Applications (2006)*, edited by F. Magoulès and T. Kako, published by Gakkotosho, Tokyo, Japan.

The present volume presents in nine chapters a selection of some domain decomposition and substructuring methods including: parareal in time algorithm, transformation methods and induced parallel properties for the temporal domain, asynchronous it-

erative methods, asynchronous sub-structuring methods, asynchronous multi-splitting methods, parallel iterative methods, coarse grid conjugate gradient, multi-point constraints in domain decomposition methods, approximate inverse preconditioning techniques, congruent sub-domains, linear and non-linear problems. The topics covered in this book are wide ranging and demonstrate the extensive use of substructuring and domain decomposition methods in fluid mechanics, structural mechanics, and computational finance.

Frédéric Magoulès
Applied Mathematics and Systems Laboratory
Ecole Centrale Paris, Grande Voie des Vignes
92295 Châtenay-Malabry Cedex, France