

# Contents

<b>1</b>	<b>Preliminaries</b>	<b>5</b>
1.1	Notation . . . . .	5
1.2	Matrix Operations . . . . .	6
1.3	Basic matrix decomposition . . . . .	8
1.4	Determinants . . . . .	10
1.5	Sparse matrices . . . . .	11
1.5.1	Storage schemes . . . . .	11
<b>2</b>	<b>Iterative methods for linear systems of equations</b>	<b>15</b>
2.1	Why not use the direct methods? . . . . .	15
2.2	Linear iterative methods . . . . .	17
2.2.1	Definition . . . . .	17
2.2.2	Jacobi method . . . . .	20
2.2.3	Gauss-Seidel method . . . . .	21
2.2.4	Convergence of Jacobi and Gauss-Seidel methods . . . . .	22
2.2.5	Stationary Richardson method . . . . .	24
2.3	Stopping Criteria . . . . .	27
2.4	Preconditioning techniques . . . . .	29
2.4.1	Preconditioned Richardson method . . . . .	30
2.5	Gradient method . . . . .	31
2.6	Conjugate Gradient method . . . . .	32
2.7	Krylov-space . . . . .	35
2.7.1	BiConjugate Gradient (BiCG) and BiCGSTAB method . . . . .	37
2.7.2	Generalized Minimum Residual (GMRES) method . . . . .	39
<b>3</b>	<b>Solving large scale eigenvalue problems</b>	<b>41</b>
3.1	Eigenvalue problems . . . . .	41
3.2	Power method . . . . .	43
3.2.1	Deflation method . . . . .	45
3.3	Inverse power method . . . . .	46
3.3.1	Inverse power method with shift . . . . .	47
3.4	QR Factorization . . . . .	48
3.4.1	Schur decomposition applied to QR algorithm . . . . .	51
3.4.2	Hessenberg applied to QR algorithm . . . . .	54
3.5	Lanczos method . . . . .	56
<b>4</b>	<b>Numerical methods for overdetermined linear systems and SVD</b>	<b>59</b>
4.1	Overdetermined systems and Least Squares . . . . .	59
4.2	Singular Value Decomposition (SVD) . . . . .	61
<b>5</b>	<b>Multigrid methods</b>	<b>64</b>
5.1	Idea of MG methods . . . . .	64
5.2	How it works . . . . .	65
5.2.1	Coarse Grids . . . . .	66
5.2.2	Correction . . . . .	69
5.2.3	Interpolation Operator . . . . .	70
5.2.4	Restriction Operator . . . . .	74
5.2.5	Two-Grid Scheme . . . . .	76

5.2.6	V-Cycle Scheme . . . . .	78
5.3	Classical Algebraic Multigrid (AMG) . . . . .	81
<b>6</b>	<b>Domain Decomposition Methods</b>	<b>86</b>
6.1	Introduction . . . . .	86
	<b>Index</b>	<b>87</b>

## 6 Domain Decomposition Methods

### 6.1 Introduction

**Domain Decomposition Methods (DDM)** are numerical techniques used to solve large-scale computational problems by breaking them into smaller, more manageable subproblems. These methods are essential in scientific computing, engineering simulations, and various other fields that require solving extensive linear systems or partial differential equations (PDEs).

#### 🔍 What Are Domain Decomposition Methods?

DDM involves dividing a large **computational domain into smaller subdomains**. These subdomains are then **solved independently**, often **in parallel**, and their **solutions are combined to form the overall solution to the original problem**. This approach is particularly useful for problems that are too large to be solved as a single system due to computational limitations.

#### 🚩 Importance of Domain Decomposition Methods

1. **Parallelism**: By solving subdomains in parallel, DDM significantly reduces the computation time, making it **feasible to tackle massive problems** that would otherwise be intractable.
2. **Scalability**: These methods can handle extremely large problems, ensuring that **computational resources are used efficiently**, and allowing for the **solution of problems on supercomputers or distributed computing systems**.
3. **Modularity**: Breaking down a complex problem into smaller subproblems makes it **easier to manage, understand, and solve**. This modularity also facilitates debugging and improving algorithms.
4. **Flexibility**: DDM can be **applied to various types of problems** across different disciplines, including fluid dynamics, structural mechanics, and electromagnetic simulations. This versatility makes them a powerful tool in the computational scientist's toolkit.
5. **Improved Convergence**: With appropriate preconditioners and iterative methods, DDM can enhance the convergence rates of solving systems, leading to **faster and more accurate solutions**.