

Contents

Preface	1
1 Preliminaries	3
1.1 The FEniCS Project	3
1.2 What you will learn	4
1.3 Working with this tutorial	4
1.4 Obtaining the software	5
1.4.1 Installation using Docker containers	6
1.4.2 Installation using Ubuntu packages	7
1.4.3 Testing your installation	8
1.5 Obtaining the tutorial examples	8
1.6 Background knowledge	8
1.6.1 Programming in Python	8
1.6.2 The finite element method	9
2 Fundamentals: Solving the Poisson equation	11
2.1 Mathematical problem formulation	11
2.1.1 Finite element variational formulation	12
2.1.2 Abstract finite element variational formulation	15
2.1.3 Choosing a test problem	16
2.2 FEniCS implementation	17
2.2.1 The complete program	17
2.2.2 Running the program	18
2.3 Dissection of the program	19
2.3.1 The important first line	19
2.3.2 Generating simple meshes	20
2.3.3 Defining the finite element function space	20
2.3.4 Defining the trial and test functions	20
2.3.5 Defining the boundary conditions	21
2.3.6 Defining the source term	24
2.3.7 Defining the variational problem	24

2.3.8	Forming and solving the linear system	25
2.3.9	Plotting the solution using the <code>plot</code> command	25
2.3.10	Plotting the solution using ParaView	27
2.3.11	Computing the error	28
2.3.12	Examining degrees of freedom and vertex values	29
2.4	Deflection of a membrane	30
2.4.1	Scaling the equation	31
2.4.2	Defining the mesh	32
2.4.3	Defining the load	32
2.4.4	Defining the variational problem	33
2.4.5	Plotting the solution	33
2.4.6	Making curve plots through the domain	34
3	A Gallery of finite element solvers	37
3.1	The heat equation	37
3.1.1	PDE problem	37
3.1.2	Variational formulation	38
3.1.3	FEniCS implementation	40
3.2	A nonlinear Poisson equation	46
3.2.1	PDE problem	46
3.2.2	Variational formulation	47
3.2.3	FEniCS implementation	47
3.3	The equations of linear elasticity	50
3.3.1	PDE problem	51
3.3.2	Variational formulation	51
3.3.3	FEniCS implementation	52
3.4	The Navier–Stokes equations	56
3.4.1	PDE problem	56
3.4.2	Variational formulation	57
3.4.3	FEniCS implementation	60
3.5	A system of advection–diffusion–reaction equations	73
3.5.1	PDE problem	73
3.5.2	Variational formulation	75
3.5.3	FEniCS implementation	75
4	Subdomains and boundary conditions	83
4.1	Combining Dirichlet and Neumann conditions	83
4.1.1	PDE problem	83
4.1.2	Variational formulation	84
4.1.3	FEniCS implementation	85
4.2	Setting multiple Dirichlet conditions	86
4.3	Defining subdomains for different materials	87
4.3.1	Using expressions to define subdomains	88
4.3.2	Using mesh functions to define subdomains	88
4.3.3	Using C++ code snippets to define subdomains	91

- 4.4 Setting multiple Dirichlet, Neumann, and Robin conditions . . . 92
 - 4.4.1 Three types of boundary conditions 93
 - 4.4.2 PDE problem 93
 - 4.4.3 Variational formulation 94
 - 4.4.4 FEniCS implementation 95
 - 4.4.5 Test problem 97
 - 4.4.6 Debugging boundary conditions 98
- 4.5 Generating meshes with subdomains 99
 - 4.5.1 PDE problem 100
 - 4.5.2 Variational formulation 102
 - 4.5.3 FEniCS implementation 102
- 5 Extensions: Improving the Poisson solver 109**
 - 5.1 Refactoring the Poisson solver 109
 - 5.1.1 A more general solver function 110
 - 5.1.2 Writing the solver as a Python module 111
 - 5.1.3 Verification and unit tests 111
 - 5.1.4 Parameterizing the number of space dimensions 114
 - 5.2 Working with linear solvers 115
 - 5.2.1 Choosing a linear solver and preconditioner 115
 - 5.2.2 Choosing a linear algebra backend 115
 - 5.2.3 Setting solver parameters 116
 - 5.2.4 An extended solver function 117
 - 5.2.5 A remark regarding unit tests 117
 - 5.2.6 List of linear solver methods and preconditioners 117
 - 5.3 High-level and low-level solver interfaces 118
 - 5.3.1 Linear variational problem and solver objects 118
 - 5.3.2 Explicit assembly and solve 119
 - 5.3.3 Examining matrix and vector values 122
 - 5.4 Degrees of freedom and function evaluation 123
 - 5.4.1 Examining the degrees of freedom 123
 - 5.4.2 Setting the degrees of freedom 125
 - 5.4.3 Function evaluation 126
 - 5.5 Postprocessing computations 127
 - 5.5.1 Test problem 127
 - 5.5.2 Flux computations 128
 - 5.5.3 Computing functionals 130
 - 5.5.4 Computing convergence rates 132
 - 5.5.5 Taking advantage of structured mesh data 136
 - 5.6 Taking the next step 141
- References 143**
- Index 145**