

Contents

1	A FEniCS tutorial	1
1.1	Fundamentals	1
1.2	Nonlinear problems	39
1.3	Time-dependent problems	48
1.4	Creating more complex domains	59
1.5	Handling domains with different materials	62
1.6	More examples	68
1.7	Miscellaneous topics	69
I	Methodology	75
2	The finite element method	77
2.1	A simple model problem	77
2.2	Finite element discretization	78
2.3	Finite element abstract formalism	80
2.4	Finite element function spaces	82
2.5	Finite element solvers	88
2.6	Finite element error estimation and adaptivity	89
2.7	Automating the finite element method	93
2.8	Historical notes	94
3	Common and unusual finite elements	95
3.1	The finite element definition	95
3.2	Notation	96
3.3	H^1 finite elements	98
3.4	$H(\text{div})$ finite elements	101
3.5	$H(\text{curl})$ finite elements	106
3.6	L^2 finite elements	108
3.7	H^2 finite elements	109
3.8	Enriching finite elements	112
3.9	Finite element exterior calculus	114
3.10	Summary	114
4	Constructing general reference finite elements	121
4.1	Background	121
4.2	Preliminaries	122
4.3	Mathematical framework	123

4.4	Examples of elements	127
4.5	Operations on the polynomial spaces	130
5	Finite element variational forms	133
5.1	Background	133
5.2	Multilinear forms	134
5.3	Discretizing multilinear forms	136
5.4	The action of a multilinear form	137
5.5	The derivative of a multilinear form	138
5.6	The adjoint of a bilinear form	139
5.7	A note on the order of trial and test functions	140
6	Finite element assembly	141
6.1	Assembly algorithm	141
6.2	Implementation	143
6.3	Symmetric application of boundary conditions	144
6.4	Parallel assembly	145
6.5	Matrix-free methods	146
7	Quadrature representation of finite element variational forms	147
7.1	Standard quadrature representation	147
7.2	Quadrature optimizations	148
7.3	Performance comparisons	153
7.4	Automatic selection of representation	158
8	Tensor representation of finite element variational forms	159
8.1	Tensor representation for Poisson's equation	159
8.2	A representation theorem	161
8.3	Extensions and limitations	162
9	Discrete optimization of finite element matrix evaluation	163
9.1	Optimization framework	163
9.2	Topological optimization	164
9.3	A graph problem	166
9.4	Geometric optimization	167
9.5	Optimization by dense linear algebra	168
9.6	Notes on implementation	169
II	Implementation	171
10	DOLFIN: a C++/Python finite element library	173
10.1	Overview	173
10.2	User interfaces	174
10.3	Functionality	176
10.4	Implementation notes	216
10.5	Historical notes	224

11 FFC: the FEniCS form compiler	227
11.1 Compilation of variational forms	227
11.2 Compiler interfaces	228
11.3 Parameters affecting code generation	232
11.4 Compiler design	232
11.5 Form representation	234
11.6 Optimization	235
11.7 Just-in-time compilation	236
11.8 Extending FFC	236
11.9 Historical notes	237
12 FErari: an optimizing compiler for variational forms	239
12.1 Optimized form compilation	239
12.2 Performance of optimizations	240
12.3 Conclusions	246
12.4 Historical notes	246
13 FIAT: numerical construction of finite element basis functions	247
13.1 Prime basis: collapsed-coordinate polynomials	247
13.2 Representing polynomials and functionals	248
13.3 Other polynomial spaces	251
13.4 Conveying topological information to clients	253
13.5 Functional evaluation	254
13.6 Overview of fundamental class structure	254
14 Instant: just-in-time compilation of C/C++ in Python	257
14.1 Brief overview of Instant and its role in FEniCS	257
14.2 Examples	257
14.3 Errors encountered when using Instant	262
14.4 Instant explained	263
14.5 Instant API	267
14.6 Related work	272
15 SyFi and SFC: symbolic finite elements and form compilation	273
15.1 GiNaC and Swiginac	273
15.2 SyFi: symbolic finite elements	275
15.3 SFC: SyFi form compiler	277
15.4 Code generation design	280
16 UFC: a finite element code generation interface	283
16.1 Overview	283
16.2 Finite element discretization and assembly	284
16.3 The UFC interface	285
16.4 Examples	290
16.5 Numbering conventions	294
16.6 Discussion	301
16.7 Historical notes	302

17 UFL: a finite element form language	303
17.1 Overview	304
17.2 Defining finite element spaces	306
17.3 Defining forms	307
17.4 Defining expressions	309
17.5 Form operators	315
17.6 Expression representation	319
17.7 Computing derivatives	324
17.8 Algorithms	329
17.9 Implementation issues	336
17.10 Conclusions and future directions	337
17.11 Acknowledgements	338
18 Unicorn: a unified continuum mechanics solver	339
18.1 Background	339
18.2 Unified continuum modeling	340
18.3 Space-time general Galerkin discretization	341
18.4 Implementation	343
18.5 Solving continuum mechanics problems	354
18.6 Acknowledgments	361
19 Lessons learned in mixed language programming	363
19.1 Background	363
19.2 Using SWIG	363
19.3 SWIG and the DOLFIN Python interface	368
19.4 JIT compilation of UFL forms, Expressions and SubDomains	380
19.5 Debugging mixed language applications	381
III Applications	383
20 Finite elements for incompressible fluids	385
20.1 Stokes equations	385
20.2 Finite element formulations for the mixed Stokes problem	386
20.3 A penalty approach: the Scott–Vogelius method	389
20.4 Numerical tests	390
20.5 Conclusions	397
21 A comparison of finite element schemes for the incompressible Navier–Stokes equations	399
21.1 Preliminaries	399
21.2 Implementation	400
21.3 Solvers	401
21.4 Test problems and results	408
21.5 Summary of results	416
21.6 Discussion	418
21.7 Conclusions	419

22 Simulation of transitional flows	421
22.1 Background	421
22.2 Numerical method and energy conservation	423
22.3 Numerical investigations	431
22.4 Conclusions	439
23 Computational hemodynamics	441
23.1 Medical background	441
23.2 Preliminaries	443
23.3 Gender differences in the intracranial vasculature	445
23.4 CFD versus 4D PC MRA in an experimental canine aneurysm	448
23.5 Patient-specific Circle of Willis	451
24 Cerebrospinal fluid flow	455
24.1 Medical background	455
24.2 Mathematical description	456
24.3 Numerical experiments	457
24.4 Conclusions	470
25 Improved Boussinesq equations for surface water waves	471
25.1 Overview	471
25.2 DOLFWAVE	473
25.3 Model derivation	475
25.4 Linear dispersion relation	478
25.5 Numerical methods	480
25.6 Wave generation	483
25.7 Reflective walls and sponge layers	484
25.8 Linear stability analysis	485
25.9 Model validation and numerical applications	489
25.10 Conclusions	501
26 Applications in solid mechanics	505
26.1 Background	505
26.2 Governing equations	506
26.3 Constitutive models	508
26.4 Time integration	510
26.5 Linearization issues for complex constitutive models	511
26.6 Implementation and examples	514
26.7 Future developments	524
27 A computational framework for nonlinear elasticity	525
27.1 Brief overview of nonlinear elasticity theory	526
27.2 Numerical methods and further implementation details	530
27.3 Examples of CBC.Twist usage	534
27.4 Conclusions	541

28 Turbulent flow and fluid–structure interaction	543
28.1 Background	543
28.2 Simulation of high Re turbulent flow	543
28.3 Turbulent boundary layers	544
28.4 Adaptivity and a posteriori error estimation	544
28.5 Robust fluid–structure coupling	545
28.6 Applications	546
28.7 Acknowledgments	551
29 An adaptive finite element solver for fluid–structure interaction problems	553
29.1 Fluid–structure interaction	553
29.2 FSI and the ALE computational framework	555
29.3 The FSI solver	556
29.4 Duality–based error control	560
29.5 Numerical examples	565
29.6 Conclusions	568
30 Modeling evolving discontinuities	571
30.1 Background	571
30.2 Partition-of-unity/extended finite element method	572
30.3 Software components	574
30.4 Design and implementation	576
30.5 Examples	577
30.6 Summary	583
31 Dynamic simulations of convection in the Earth’s mantle	585
31.1 Convection in the Earth’s mantle	585
31.2 Mathematical statement of the problem	586
31.3 Numerical method	588
31.4 Implementation	591
31.5 Results	594
31.6 Discussion and concluding remarks	597
32 Automatic calibration of depositional models	601
32.1 Issues in sedimentary deposition	601
32.2 A multidimensional sedimentation model	601
32.3 An inverse approach	602
32.4 The Landweber algorithm	602
32.5 Evaluation of gradients by duality arguments	603
32.6 Aspects of the implementation	605
32.7 Numerical experiments	606
32.8 Results and conclusions	609
33 A coupled stochastic and deterministic model of Ca^{2+} dynamics in the dyadic cleft	611
33.1 Biological background	611
33.2 Mathematical models	612
33.3 Numerical methods for the continuous system	615
33.4 diffsim: an event-driven simulator	623
33.5 Discussion	627

34 Electromagnetic waveguide analysis	629
34.1 Formulation	630
34.2 Implementation	634
34.3 Examples	637
34.4 Conclusions	642
35 Block preconditioning of systems of PDEs	643
35.1 Abstract framework for constructing preconditioners	643
35.2 Overview of CBC.Block	644
35.3 Numerical examples	645
35.4 Conclusions	654
36 Automated testing of saddle point stability conditions	657
36.1 Background	657
36.2 Why does discrete stability matter?	658
36.3 Discrete stability	659
36.4 Eigenvalue problems associated with saddle point stability	662
36.5 Automating the stability testing	663
36.6 Examples	667
36.7 Conclusions	671
List of authors	673
GNU Free Documentation License	681
0. Preamble	681
1. Applicability and definitions	681
2. Verbatim copying	682
3. Copying in quantity	683
4. Modifications	683
5. Combining documents	685
6. Collections of documents	685
7. Aggregation with independent works	685
8. Translation	686
9. Termination	686
10. Future revisions of this license	686
11. Relicensing	687
References	689
Index	717