

References

- [1] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified Form Language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, 40(2), 2014. doi:10.1145/2566630, arXiv:1211.4047.
- [2] Douglas N. Arnold and Anders Logg. Periodic table of the finite elements. *SIAM News*, 2014.
- [3] W. B. Bickford. *A First Course in the Finite Element Method*. Irwin, 2nd edition, 1994.
- [4] D. Braess. *Finite Elements*. Cambridge University Press, Cambridge, third edition, 2007.
- [5] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008.
- [6] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.
- [7] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*, volume 40 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 2002. Reprint of the 1978 original [North-Holland, Amsterdam; MR0520174 (58 #25001)].
- [8] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Wiley Press, 2003.
- [9] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations*. Cambridge University Press, 1996.
- [10] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. Springer, 2004.
- [11] Python Software Foundation. The Python tutorial. <http://docs.python.org/2/tutorial>.
- [12] M. Gockenbach. *Understanding and Implementing the Finite Element Method*. SIAM, 2006.

- [13] K. Goda. A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows. *Journal of Computational Physics*, 30(1):76–95, 1979.
- [14] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall, 1987.
- [15] J. M. Kinder and P. Nelson. *A Student's Guide to Python for Physical Modeling*. Princeton University Press, 2015.
- [16] Robert C. Kirby. Fiat, a new paradigm for computing finite element basis functions. *ACM Transactions on Mathematical Software*, 30(4):502–516, 2004.
- [17] Robert C. Kirby and Anders Logg. A compiler for variational forms. *ACM Transactions on Mathematical Software*, 32(3):417–444, 2006.
- [18] J. Kiusalaas. *Numerical Methods in Engineering With Python*. Cambridge University Press, 2005.
- [19] R. H. Landau, M. J. Paez, and C. C. Bordeianu. *Computational Physics: Problem Solving with Python*. Wiley, third edition, 2015.
- [20] H. P. Langtangen. *Python Scripting for Computational Science*. Springer, third edition, 2009.
- [21] H. P. Langtangen. *A Primer on Scientific Programming With Python*. Texts in Computational Science and Engineering. Springer, fifth edition, 2016.
- [22] H. P. Langtangen and L. R. Hellevik. Brief tutorials on scientific Python, 2016. <http://hplgit.github.io/bumpy/doc/web/index.html>.
- [23] H. P. Langtangen and A. Logg. *Solving PDEs in Hours – The FEniCS Tutorial Volume II*. Springer, 2016.
- [24] H. P. Langtangen and K.-A. Mardal. *Introduction to Numerical Methods for Variational Problems*. 2016. <http://hplgit.github.io/fem-book/doc/web/>.
- [25] M. G. Larson and F. Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*. Texts in Computational Science and Engineering. Springer, 2013.
- [26] A. Logg, K.-A. Mardal, and G. N. Wells. *Automated Solution of Partial Differential Equations by the Finite Element Method*. Springer, 2012.
- [27] Anders Logg and Garth N. Wells. DOLFIN: Automated finite element computing. *ACM Transactions on Mathematical Software*, 37(2), 2010.
- [28] M. Pilgrim. *Dive into Python*. Apress, 2004. <http://www.diveintopython.net>.
- [29] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer Series in Computational Mathematics. Springer, 1994.
- [30] A. Henderson Squillacote. *The ParaView Guide*. Kitware, 2007.
- [31] R. Temam. Sur l'approximation de la solution des équations de Navier-Stokes. *Arc. Ration. Mech. Anal.*, 32:377–385, 1969.

Index

- .hdf5 file, 69
- .pvd file, 45
- .vtu file, 45
- .xdmf file, 69

- abstract variational formulation, 15
- advection–diffusion–reaction, 73
- `assemble`, 64, 119, 120
- `assemble_system`, 120
- assembly, 119

- backward difference, 38
- boundary conditions, 92
- boundary markers, 88
- boundary specification (class), 88
- boundary specification (function), 21, 22
- `BoxField`, 136

- C++ expression syntax, 22
- `CellFunction`, 88
- CFD, 56
- channel flow, 60
- chemical reactions, 73
- Chorin’s method, 57
- `Circle`, 102
- code, 8
- `CompiledSubDomain`, 91
- components, 81

- constructive solid geometry, 67
- contour plot, 137
- convergence rate, 132
- coordinates, 123
- coupled systems, 73
- CSG, 67
- curve plots, 34
- cylinder flow, 65

- Debian, 7
- DEBUG log level, 67
- deep copy, 81
- degrees of freedom, 25, 29, 30, 129
- degrees of freedom, 123
- dimension-independent code, 114
- Dirichlet boundary condition, 92
- Dirichlet boundary conditions, 20
- `DirichletBC`, 20
- Docker, 6
- dof to vertex map, 124
- dofs, 29
- DOLFIN, 3

- editor, 7
- Eigen, 115
- elasticity, 50
- Emacs, 7
- energy functional, 130
- error, 28, 133

- error functional, 131
- `errornorm`, 131
- exact solution, 16
- exporting solutions, 27
- `Expression`, 21, 32
- expression syntax (C++), 22
- `FacetFunction`, 88
- `fenicsproject`, 6
- FFC, 3
- FIAT, 3
- finite element method, 9
- finite element space, 20
- flat program, 109
- flux functional, 131
- `ft01_poisson.py`, 18
- `ft02_poisson_membrane.py`, 34
- `ft03_heat.py`, 44
- `ft04_heat_gaussian.py`, 46
- `ft05_poisson_nonlinear.py`, 50
- `ft06_elasticity.py`, 54
- `ft07_navier_stokes_channel.py`, 65
- `ft08_navier_stokes_cylinder.py`, 73
- `ft09_reaction_system.py`, 80
- `ft10_poisson_extended.py`, 97, 117, 127
- `ft11_magnetostatics.py`, 107
- `ft12_poisson_solver.py`, 111
- function evaluation, 126
- function space, 20
- functionals, 130
- `FunctionSpace`, 20
- `generate_mesh`, 67, 102
- GNU/Linux, 5
- `grad`, 58
- HDF5 format, 69
- heat equation, 37
- heterogeneous media, 87
- implicit Euler, 38
- incremental pressure correction scheme, 57
- infinite domain, 101
- `info`, 116
- initial condition, 80
- installation, 5
- integration by parts, 13
- `interpolate`, 40
- interpolation, 41
- Jacobian, 50
- Jupyter, 19
- Krylov solver, 68, 115, 117
- `KrylovSolver`, 121
- Lagrange finite element, 20
- `lhs`, 41, 97
- linear algebra backend, 115
- linear solver, 115, 117
- linear system, 64, 120
- `LinearVariationalProblem`, 118
- `LinearVariationalSolver`, 118
- Linux, 5
- Mac OS X, 5
- magnetostatics, 100
- MATLAB, 122
- Maxwell's equations, 100
- `Measure`, 104
- `Mesh`, 19
- mesh, 19
- mesh generation, 102
- `MeshFunction`, 88
- method of manufactured solutions, 16, 48
- mixed finite element, 76
- mixed function space, 76
- `MixedElement`, 76
- `mshr`, 3, 67, 102
- multi-material domain, 87
- `nabla_grad`, 54, 58
- Navier-Stokes equations, 56
- `near`, 62, 85
- Neumann boundary condition, 83, 92
- Newton's method, 50
- nodal values, 29, 30, 129
- nonlinear problem, 46
- norm, 133

- numbering
 - cell vertices, 30
 - degrees of freedom, 30
- P1 element, 20
- packages, 7
- parameters, 116
- ParaView, 27
- performance, 6
- Periodic Table of the Finite Elements, 20
- PETSc, 3, 115
- plot, 25
- plotting, 25
- Poisson's equation, 11
- postprocessing, 27, 127
- preconditioner, 68, 115, 117
- PROGRESS log level, 67
- Progress, 67
- project, 40, 128
- projection, 41
- Python, 8
- Python module, 111

- reaction system, 73
- Reynolds number, 60
- rhs, 41, 97
- Robin boundary condition, 92
- rounding errors, 23

- scaling, 55, 60
- scitools, 136
- set_log_level, 67
- SLEPc, 122
- source, 5
- space dimensions, 114
- split, 81
- splitting method, 57
- Spyder, 19
- strain-rate tensor, 57
- stress tensor, 51, 56
- structured mesh, 136
- subdomains, 83
- surface plot (structured mesh), 137
- SymPy, 48
- sympy, 139

- system of PDEs, 50

- tensor, 51
- terminal window, 18
- test function, 13
- test problem, 16
- TestFunction, 20
- time step, 38
- time-dependent expression, 41
- time-dependent problem, 37
- TimeSeries, 69, 76
- tolerance, 23
- trial function, 13
- TrialFunction, 20

- Ubuntu, 7
- UFL, 3, 24
- unit testing, 111

- variational formulation, 12
- vector-valued functions, 54
- VectorElement, 77
- VectorFunctionSpace, 54, 62
- verification, 16, 65, 111
- vertex to dof map, 124
- vertex values, 29, 123, 124
- Vim, 7
- visualization, structured mesh, 136
- VTK format, 45

- Windows, 5

- XDMF format, 69

© The Editor(s) (if applicable) and The Author(s) 2016. This book is published open access.

Open Access This chapter is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, duplication, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the work's Creative Commons license, unless indicated otherwise in the credit line; if such material is not included in the work's Creative Commons license and the respective action is not permitted by statutory regulation, users will need to obtain permission from the license holder to duplicate, adapt or reproduce the material.

