

FEniCS Course

Overview and Introduction

Lecturer

Anders Logg

Beijing, August 5–7 2015



FENICS
PROJECT

Course outline

★ Overview and Introduction

Wed **L01** Installation of FEniCS

L02 Static linear PDEs

L04 Time-dependent PDEs

Thu **L06** Static hyperelasticity

L07 Dynamic hyperelasticity

Fri **L08** The Stokes problem

L09 Incompressible Navier–Stokes

Full list of FEniCS lectures

★ Overview and Introduction

- L00** Introduction to FEM
- L01** Installation of FEniCS
- L02** Static linear PDEs
- L03** Static nonlinear PDEs
- L04** Time-dependent PDEs
- L05** Happy hacking: Tools, tips and coding practices
- L06** Static hyperelasticity
- L07** Dynamic hyperelasticity
- L08** The Stokes problem
- L09** Incompressible Navier–Stokes
- L10** Discontinuous Galerkin methods for elliptic equations
- L11** A posteriori error estimates and adaptivity
- L12** Computing sensitivities
- L13** Introduction to dolfin-adjoint
- L14** From sensitivities to optimisation
- L14** One-shot optimisation
- L16** Optimal control of the Navier-Stokes equations

All lectures can be downloaded from

<http://fenicsproject.org/pub/course/>



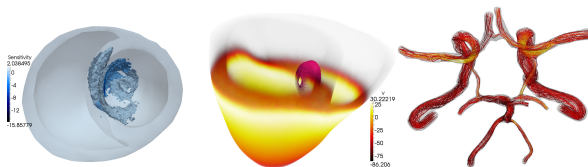
The FEniCS Project is a collection of open-source software components aimed at the numerical solution of partial differential equations using finite element methods

Key distinguishing features

- FEniCS (Python/C++) code is quick to write and easy to read
- ‘Any’ finite element formulation of ‘any’ partial differential equation can be coded
- Automated code generation is heavily used under the hood to create efficient, specialized, low-level code
- Performance – implicit problems with over 12 000 000 000 degrees of freedom can be solved in a couple of minutes

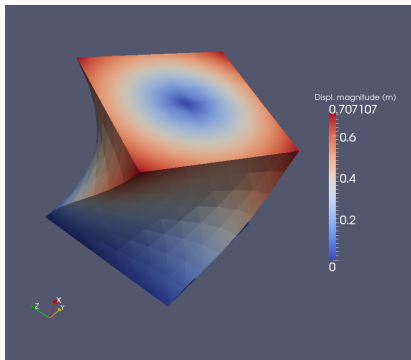
FEniCS has been used for a wide range of equations and applications

Reaction-diffusion equations; Stokes with or without nonlinear viscosity; compressible and incompressible Navier–Stokes; RANS turbulence models; shallow water equations; Bidomain equations; nonlinear and linear elasticity; nonlinear and linear viscoelasticity; Schrödinger; Biot's equations for porous media, fracture mechanics, electromagnetism, liquid crystals including liquid crystal elastomers, combustion, ... and coupled systems of the above, ...



for simulating blood flow, computing calcium release in cardiac tissue, computing the cardiac potential in the heart, simulating mantle convection, simulating melting ice sheets, computing the optimal placement of tidal turbines, simulating and reconstructing tsunamis, simulating the flow of cerebrospinal fluid and the deformation of the spinal cord, simulating waveguides, ...

Hyperelasticity



Python code

```
from fenics import *

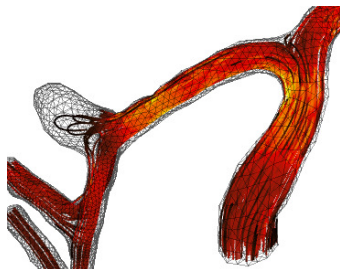
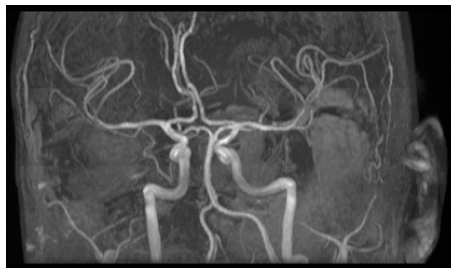
mesh = UnitCubeMesh(24, 16, 16)
V = VectorFunctionSpace(mesh, "Lagrange", 1)

left = CompiledSubDomain(("std::abs(x[0])
< DOLFIN_EPS) && on_boundary")
right = CompiledSubDomain(("std::abs(x[0] - 1.0)
< DOLFIN_EPS) && on_boundary")

c = Expression(("0.0", "0.0", "0.0"))
r = Expression(("0.0",
"0.5*(y0+(x[1]-y0)*cos(t)-(x[2]-z0)*sin(t)-x[1])",
"0.5*(z0+(x[1]-y0)*sin(t)+(x[2]-z0)*cos(t)-x[2])"),
y0=0.5, z0=0.5, t=pi/3)
bcl = DirichletBC(V, c, left)
bcr = DirichletBC(V, r, right)
bcs = [bcl, bcr]
v = TestFunction(V)
u = Function(V)
B = Constant((0.0, -0.5, 0.0))
T = Constant((0.1, 0.0, 0.0))
I = Identity(V.cell().d)
F = I + grad(u)
Ic = tr(F.T*F)
J = det(F)
E, nu = 10.0, 0.3
mu, lambda = Constant(E/(2*(1 + nu))),
Constant(E*nu/((1 + nu)*(1 - 2*nu)))
psi = (mu/2)*(Ic - 3) - mu*ln(J) +
(lambda/2)*(ln(J))**2
Pi = psi*dx - dot(B, u)*dx - dot(T, u)*ds
F = derivative(Pi, u, v)

solve(F == 0, u, bcs)
plot(u, interactive=True, mode="displacement")
```

Computational hemodynamics



- Low wall shear stress may trigger aneurysm growth
- Solve the incompressible Navier–Stokes equations on patient-specific geometries

$$\begin{aligned}\dot{u} + u \cdot \nabla u - \nabla \cdot \sigma(u, p) &= f \\ \nabla \cdot u &= 0\end{aligned}$$

Computational hemodynamics (contd.)



Python code

```
# Define Cauchy stress tensor
def sigma(v,w):
    return 2.0*mu*0.5*(grad(v) + grad(v).T) -
        w*Identity(v.cell().d)

# Define symmetric gradient
def epsilon(v):
    return 0.5*(grad(v) + grad(v).T)

# Tentative velocity step (sigma formulation)
U = 0.5*(u0 + u)
F1 = rho*(1/k)*inner(v, u - u0)*dx + rho*inner(v,
    grad(u0)*(u0 - w))*dx \
    + inner(epsilon(v), sigma(U, p0))*dx \
    + inner(v, p0*n)*ds - mu*inner(grad(U).T*n,
        v)*ds \
    - inner(v, f)*dx
a1 = lhs(F1)
L1 = rhs(F1)

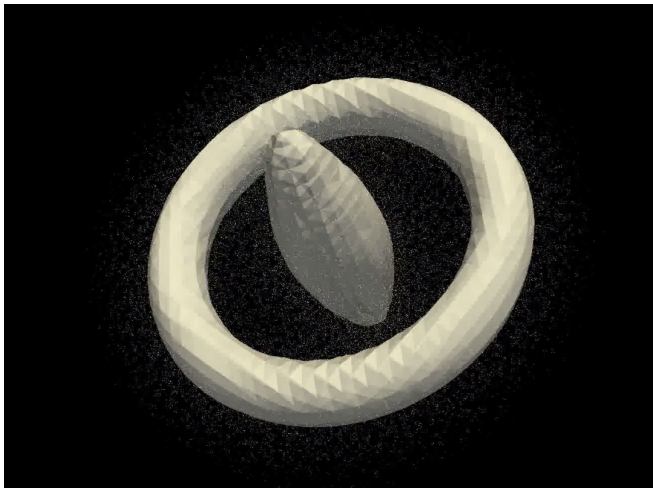
# Pressure correction
a2 = inner(grad(q), k*grad(p))*dx
L2 = inner(grad(q), k*grad(p0))*dx - q*div(u1)*dx

# Velocity correction
a3 = inner(v, u)*dx
L3 = inner(v, u1)*dx + inner(v, k*grad(p0 -
    p1))*dx
```

- The Navier–Stokes solver is implemented in Python/FEniCS
- FEniCS allows solvers to be implemented in a minimal amount of code

Numerical relativity

$$R_{ab} - \frac{1}{2}Rg_{ab} + g_{ab}\Lambda = 8\pi T_{ab}$$



Hello World in FEniCS: problem formulation

Poisson's equation

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega \\ u &= 0 && \text{on } \partial\Omega \end{aligned}$$

Finite element formulation

Find $u \in V$ such that

$$\underbrace{\int_{\Omega} \nabla u \cdot \nabla v \, dx}_{a(u,v)} = \underbrace{\int_{\Omega} f v \, dx}_{L(v)} \quad \forall v \in V$$

Hello World in FEniCS: implementation

Python code

```
from fenics import *

mesh = UnitSquareMesh(32, 32)

V = FunctionSpace(mesh, "Lagrange", 1)
u = TrialFunction(V)
v = TestFunction(V)
f = Expression("x[0]*x[1]")

a = dot(grad(u), grad(v))*dx
L = f*v*dx

bc = DirichletBC(V, 0.0, DomainBoundary())

u = Function(V)
solve(a == L, u, bc)
plot(u)
```

Basic API

- Mesh, Vertex, Edge, Face, Facet, Cell
 - FiniteElement, FunctionSpace
 - TrialFunction, TestFunction, Function
 - grad(), curl(), div(), ...
 - Matrix, Vector, KrylovSolver, LUSolver
 - assemble(), solve(), plot()
-
- Python interface generated semi-automatically by SWIG
 - C++ and Python interfaces almost identical

Sounds great, but how do I find my way through the jungle?



Three survival advices



Use the right Python
tools



Explore the
documentation



Ask, report and
request

Documentation for FEniCS 1.3.0

Our documentation includes a book, a collection of documented demo programs, and complete references for the FEniCS application programming interface (API). Note that the FEniCS API is documented separately for each FEniCS component. The most important interfaces are those of the C++/Python problem solving environment *DOLFIN* and the form language *UFL*.

(This page accesses the FEniCS 1.3.0 documentation. Not the version you are looking for? See [all versions](#).)

The FEniCS Tutorial

A good starting point for new users is the *FEniCS Tutorial*. The tutorial will help you get quickly up and running with solving differential equations in FEniCS. The tutorial focuses exclusively on the FEniCS Python interface, since this is the simplest approach to exploring FEniCS for beginners.

The FEniCS Book



The FEniCS Book, Automated Solution of Differential Equations by the Finite Element Method, is a comprehensive (700 pages) book documenting the mathematical methodology behind the FEniCS Project and the software developed as part of the FEniCS Project. The FEniCS Tutorial is included as the opening chapter

of the FEniCS Book.

The FEniCS Manual

The FEniCS Manual is a 200-page excerpt from the FEniCS Book, including the FEniCS Tutorial, an introduction to the finite element method and documentation of DOLFIN and UFL.

Additional Documentation

Mixing software with FEniCS is a tutorial on how to combine FEniCS applications in Python with software written in other languages.

Demos

A simple way to build your first FEniCS application is to copy and modify one of the existing demos:

Documented DOLFIN demos (Python)

Documented DOLFIN demos (C++)

The demos are *already installed on your system* or can be found in the demo directory of the DOLFIN source tree.

Quick Programmer's References

Some of the classes and functions in DOLFIN are more frequently used than others. To learn more about these, take a look at the

Basic classes and functions in DOLFIN (Python)

Basic classes and functions in DOLFIN (C++)

Complete Programmer's References

All classes and functions in DOLFIN (Python)

All classes and functions in DOLFIN (C++)

All classes and functions in UFL

Documentation for FEniCS 1.3.0

Our documentation includes a book, a collection of documented demo programs, and complete references for the FEniCS application programming interface (API). Note that the FEniCS API is documented separately for each FEniCS component. The most important interfaces are those of the C++/Python problem solving environment *DOLFIN* and the form language *UFL*.

(This page accesses the FEniCS 1.3.0 documentation. Not the version you are looking for? See [all versions](#).)

The FEniCS Tutorial

A good starting point for new users is the *FEniCS Tutorial*. The tutorial will help you get quickly up and running with solving differential equations in FEniCS. The tutorial focuses exclusively on the FEniCS Python interface, since this is the simplest approach to exploring FEniCS for beginners.

The FEniCS Book



The FEniCS Book, Automated Solution of Differential Equations by the Finite Element Method, is a comprehensive (700 pages) book documenting the mathematical methodology behind the FEniCS Project and the software developed as part of the FEniCS Project. The FEniCS Tutorial is included as the opening chapter of the FEniCS Book.

The FEniCS Manual

The FEniCS Manual is a 200-page excerpt from the FEniCS Book, including the FEniCS Tutorial, an introduction to the finite element method and documentation of DOLFIN and UFL.

Additional Documentation

Mixing software with FEniCS is a tutorial on how to combine FEniCS applications in Python with software written in other languages.

Demos

A simple way to build your first FEniCS application is to copy and modify one of the existing demos:

Documented DOLFIN demos (Python)

Documented DOLFIN demos (C++)

The demos are *already installed on your system* or can be found in the demo directory of the DOLFIN source tree.

Quick Programmer's References

Some of the classes and functions in DOLFIN are more frequently used than others. To learn more about these, take a look at the

Basic classes and functions in DOLFIN (Python)

Basic classes and functions in DOLFIN (C++)

Complete Programmer's References

All classes and functions in DOLFIN (Python)

All classes and functions in DOLFIN (C++)

All classes and functions in UFL

Documentation for FEniCS 1.3.0


Our documentation includes a book, a collection of documented demo programs, and complete references for the FEniCS application programming interface (API). Note that the FEniCS API is documented separately for each FEniCS component. The most important interfaces are those of the C++/Python problem solving environment *DOLFIN* and the form language *UFL*.

(This page accesses the FEniCS 1.3.0 documentation. Not the version you are looking for? See [all versions](#).)

The FEniCS Tutorial

A good starting point for new users is the *FEniCS Tutorial*. The tutorial will help you get quickly up and running with solving differential equations in FEniCS. The tutorial focuses exclusively on the FEniCS Python interface, since this is the simplest approach to exploring FEniCS for beginners.

The FEniCS Book



The FEniCS Book, Automated Solution of Differential Equations by the Finite Element Method, is a comprehensive (700 pages) book documenting the mathematical methodology behind the FEniCS Project and the software developed as part of the FEniCS Project. The FEniCS Tutorial is included as the opening chapter of the FEniCS Book.

The FEniCS Manual

The *FEniCS Manual* is a 200-page excerpt from the FEniCS Book, including the FEniCS Tutorial, an introduction to the finite element method and documentation of DOLFIN and UFL.

Additional Documentation

Mixing software with FEniCS is a tutorial on how to combine FEniCS applications in Python with software written in other languages.

Demos

A simple way to build your first FEniCS application is to copy and modify one of the existing demos:

[Documented DOLFIN demos \(Python\)](#)

[Documented DOLFIN demos \(C++\)](#)

The demos are *already installed on your system* or can be found in the demo directory of the DOLFIN source tree.

Quick Programmer's References

Some of the classes and functions in DOLFIN are more frequently used than others. To learn more about these, take a look at the

[Basic classes and functions in DOLFIN \(Python\)](#)

[Basic classes and functions in DOLFIN \(C++\)](#)

Complete Programmer's References

[All classes and functions in DOLFIN \(Python\)](#)

[All classes and functions in DOLFIN \(C++\)](#)

[All classes and functions in UFL](#)

Documentation for FEniCS 1.3.0

Our documentation includes a book, a collection of documented demo programs, and complete references for the FEniCS application programming interface (API). Note that the FEniCS API is documented separately for each FEniCS component. The most important interfaces are those of the C++/Python problem solving environment *DOLFIN* and the form language *UFL*.

(This page accesses the FEniCS 1.3.0 documentation. Not the version you are looking for? See [all versions](#).)

The FEniCS Tutorial

A good starting point for new users is the *FEniCS Tutorial*. The tutorial will help you get quickly up and running with solving differential equations in FEniCS. The tutorial focuses exclusively on the FEniCS Python interface, since this is the simplest approach to exploring FEniCS for beginners.

The FEniCS Book



The FEniCS Book, Automated Solution of Differential Equations by the Finite Element Method, is a comprehensive (700 pages) book documenting the mathematical methodology behind the FEniCS Project and the software developed as part of the FEniCS Project. The FEniCS Tutorial is included as the opening chapter of the FEniCS Book.

The FEniCS Manual

The *FEniCS Manual* is a 200-page excerpt from the FEniCS Book, including the FEniCS Tutorial, an introduction to the finite element method and documentation of DOLFIN and UFL.

Additional Documentation

Mixing software with FEniCS is a tutorial on how to combine FEniCS applications in Python with software written in other languages.

Demos

A simple way to build your first FEniCS application is to copy and modify one of the existing demos:

Documented DOLFIN demos (Python)

Documented DOLFIN demos (C++)

The demos are *already installed on your system* or can be found in the demo directory of the DOLFIN source tree.

Quick Programmer's References

Some of the classes and functions in DOLFIN are more frequently used than others. To learn more about these, take a look at the

Basic classes and functions in DOLFIN (Python)

Basic classes and functions in DOLFIN (C++)

Complete Programmer's References

All classes and functions in DOLFIN (Python)

All classes and functions in DOLFIN (C++)

All classes and functions in UFL

<http://fenicsproject.org/documentation/>

Documentation for FEniCS 1.3.0

Our documentation includes a book, a collection of documented demo programs, and complete references for the FEniCS application programming interface (API). Note that the FEniCS API is documented separately for each FEniCS component. The most important interfaces are those of the C++/Python problem solving environment *DOLFIN* and the form language *UFL*.

(This page accesses the FEniCS 1.3.0 documentation. Not the version you are looking for? See [all versions](#).)

The FEniCS Tutorial

A good starting point for new users is the *FEniCS Tutorial*. The tutorial will help you get quickly up and running with solving differential equations in FEniCS. The tutorial focuses exclusively on the FEniCS Python interface, since this is the simplest approach to exploring FEniCS for beginners.

The FEniCS Book



The FEniCS Book, Automated Solution of Differential Equations by the Finite Element Method, is a comprehensive (700 pages) book documenting the mathematical methodology behind the FEniCS Project and the software developed as part of the FEniCS Project. The FEniCS Tutorial is included as the opening chapter of the FEniCS Book.

The FEniCS Manual

The FEniCS Manual is a 200-page excerpt from the FEniCS Book, including the FEniCS Tutorial, an introduction to the finite element method and documentation of DOLFIN and UFL.

Additional Documentation

Mixing software with FEniCS is a tutorial on how to combine FEniCS applications in Python with software written in other languages.

Demos

A simple way to build your first FEniCS application is to copy and modify one of the existing demos:

[Documented DOLFIN demos \(Python\)](#)

[Documented DOLFIN demos \(C++\)](#)

The demos are *already installed on your system* or can be found in the demo directory of the DOLFIN source tree.

Quick Programmer's References

Some of the classes and functions in DOLFIN are more frequently used than others. To learn more about these, take a look at the

[Basic classes and functions in DOLFIN \(Python\)](#)

[Basic classes and functions in DOLFIN \(C++\)](#)

Complete Programmer's References

[All classes and functions in DOLFIN \(Python\)](#)

[All classes and functions in DOLFIN \(C++\)](#)

[All classes and functions in UFL](#)

Documentation for FEniCS 1.3.0

Our documentation includes a book, a collection of documented demo programs, and complete references for the FEniCS application programming interface (API). Note that the FEniCS API is documented separately for each FEniCS component. The most important interfaces are those of the C++/Python problem solving environment *DOLFIN* and the form language *UFL*.

(This page accesses the FEniCS 1.3.0 documentation. Not the version you are looking for? See [all versions](#).)

The FEniCS Tutorial

A good starting point for new users is the *FEniCS Tutorial*. The tutorial will help you get quickly up and running with solving differential equations in FEniCS. The tutorial focuses exclusively on the FEniCS Python interface, since this is the simplest approach to exploring FEniCS for beginners.

The FEniCS Book



The FEniCS Book, Automated Solution of Differential Equations by the Finite Element Method, is a comprehensive (700 pages) book documenting the mathematical methodology behind the FEniCS Project and the software developed as part of the FEniCS Project. The FEniCS Tutorial is included as the opening chapter of the FEniCS Book.

The FEniCS Manual

The FEniCS Manual is a 200-page excerpt from the FEniCS Book, including the FEniCS Tutorial, an introduction to the finite element method and documentation of DOLFIN and UFL.

Additional Documentation

Mixing software with FEniCS is a tutorial on how to combine FEniCS applications in Python with software written in other languages.

Demos

A simple way to build your first FEniCS application is to copy and modify one of the existing demos:

Documented DOLFIN demos (Python)

Documented DOLFIN demos (C++)

The demos are *already installed on your system* or can be found in the demo directory of the DOLFIN source tree.

Quick Programmer's References

Some of the classes and functions in DOLFIN are more frequently used than others. To learn more about these, take a look at the

Basic classes and functions in DOLFIN (Python)

Basic classes and functions in DOLFIN (C++)

Complete Programmer's References

All classes and functions in DOLFIN (Python)

All classes and functions in DOLFIN (C++)

All classes and functions in UFL

Development community is organized via bitbucket.org

The screenshot shows the Bitbucket web interface for the repository 'fenics-project/DOLFIN'. The browser address bar shows the URL 'https://bitbucket.org/fenics-project/dolfin'. The page layout includes a left sidebar with navigation options like 'Overview', 'Source', 'Commits', 'Branches', 'Pull requests', 'Issues', 'Wiki', 'Downloads', and 'Settings'. The main content area is titled 'Overview' and displays repository statistics: '99+ Branches', '3 Tags', '48 Forks', and '73 Watchers'. It also features an 'Invite users to this repo' section with a 'Send invitation' button and a 'Recent activity' section showing a list of commits with their authors and timestamps.

fenics-project / DOLFIN — Bitbucket - Mozilla Firefox

Atlassian, Inc. (US) | https://bitbucket.org/fenics-project/dolfin

Bitbucket Dashboard Teams Repositories Snippets Create Find a repository...

fenics-project DOLFIN

ACTIONS

- Clone
- Create branch
- Create pull request
- Compare
- Fork

NAVIGATION

- Overview
- Source
- Commits
- Branches
- Pull requests
- Issues
- Wiki
- Downloads
- Settings

Overview

SSH git@bitbucket.org:fenics-project/dolfin Share

Last updated	6 minutes ago	99+ Branches	3 Tags
Language	C++	48 Forks	73 Watchers
Access level	Admin (revoke)		

Invite users to this repo

Send invitation

Recent activity

- 1 commit Pushed to fenics-project/dolfin | 8bc2b98 Merge branch 'garth/replace-boo... Garth Wells · 7 minutes ago
- 1 commit Pushed to fenics-project/dolfin | 8db4a67 Merge branch 'garth/replace-lexi... Garth Wells · 10 minutes ago
- 1 commit Pushed to fenics-project/dolfin | ab59d44 Replace Boost lexical_cast with ... Garth Wells · 10 minutes ago
- 1 commit Pushed to fenics-project/dolfin | be6345d Merge branch 'garth/replace-boo... Garth Wells · 35 minutes ago

DOLFIN

DOLFIN is the C++/Python interface of FEniCS, providing a consistent PSE (Problem Solving Environment) for ordinary and partial differential equations.

Installation

To build DOLFIN, run:

```
mkdir build
cd build
cmake ..
make install
```

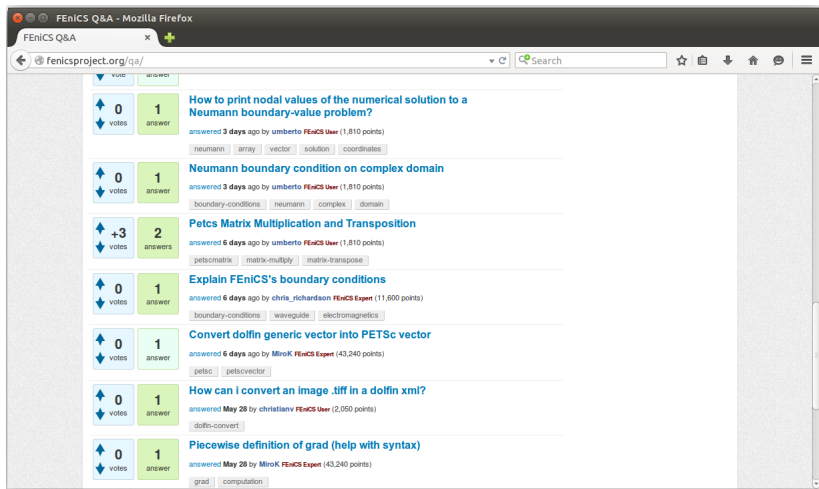
For detailed instructions, see the file INSTALL.

License

DOLFIN is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

<http://bitbucket.org/fenics-project/>

Community help is available via QA forum



The screenshot shows a web browser window titled "FEniCS Q&A - Mozilla Firefox" with the URL "fenicsproject.org/qa/". The page displays a list of questions and answers. Each entry includes a question title, the number of votes and answers, the user who answered, the time since the answer was posted, and the user's reputation points. Tags are also present for each question.

Question Title	Votes	Answers	Answered By	Points	Time Ago	Tags
How to print nodal values of the numerical solution to a Neumann boundary-value problem?	0	1	umberto FEniCS User	1,810	3 days ago	neumann, array, vector, solution, coordinates
Neumann boundary condition on complex domain	0	1	umberto FEniCS User	1,810	3 days ago	boundary-conditions, neumann, complex, domain
Petsc Matrix Multiplication and Transposition	+3	2	umberto FEniCS User	1,810	6 days ago	petscmatrix, matrix-multiply, matrix-transpose
Explain FEniCS's boundary conditions	0	1	chris_richardson FEniCS Expert	11,600	6 days ago	boundary-conditions, waveguide, electromagnetics
Convert dolfin generic vector into PETSc vector	0	1	MiroK FEniCS Expert	43,240	6 days ago	petsc, petscvector
How can i convert an image .tiff in a dolfin xml?	0	1	christianv FEniCS User	2,050	May 28	dolfin-convert
Piecewise definition of grad (help with syntax)	0	1	MiroK FEniCS Expert	43,240	May 28	grad, computation

<https://fenicsproject.org/qa>

Let's get started and remember:

- **Lectures** can be downloaded from

`http://fenicsproject.org/pub/course/lectures`

- **Data** for exercises can be downloaded from

`http://fenicsproject.org/pub/course/data`

(Or copy from the .../pub/data directory on alarik)

- **Solutions** for exercises can be downloaded from

`http://fenicsproject.org/pub/course/src`

(Secret password needed!)