



# FEniCS Course

## Overview

---

*Contributors*

Martin Alnæs

# Course outline

- L00** Introduction to FEM
- L01** Introduction to 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

Lectures can be downloaded from  
<http://fenicsproject.org/pub/course/>



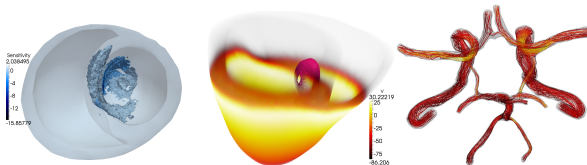
## Key distinguishing features

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

- 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, ...

# 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]", degree=2)

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

# 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](#)

<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](#)

The screenshot shows a web browser displaying the FEniCS Project documentation. The browser's address bar shows the URL `fenics.readthedocs.org/en/latest/`. The page has a dark blue sidebar on the left with the FEniCS Project logo and a search bar. The main content area has a light blue header with the text "Docs » FEniCS Project" and a link to "Edit on Bitbucket". The main heading is "FEniCS Project", followed by a paragraph stating that this is experimental documentation for the FEniCS Project, which is under development on Read the Docs. Below this, a paragraph states that FEniCS is a collection of inter-operating modules, with links to the documentation for each module listed below. A bulleted list follows, containing links to DOLFIN, UFL, FFC, FIAT, and Instant. The next section is "Installation", followed by "Containers/Docker", which states that a collection of Docker containers for FEniCS are available and provides a link to `http://fenics-containers.readthedocs.org/en/latest/` for how to run FEniCS inside a container. At the bottom, a footer contains copyright information for 2015, the FEniCS Project Team, revision a8a6ba83, and mentions that it was built with Sphinx using a theme provided by Read the Docs.

fenics.readthedocs.org/en/latest/

FEniCS Project  
latest

Search docs

FEniCS Project

Installation

Containers/Docker

WRITE THE DOCS

Love Documentation? Come to the Write the Docs 2016 conference in Portland.

Docs » FEniCS Project

Edit on Bitbucket

## FEniCS Project

This is experimental documentation for the FEniCS Project. This version of the documentation on Read the Docs is under development.

FEniCS is a collection of inter-operating modules. Links to the documentation for each module are listed below.

- DOLFIN
- UFL
- FFC
- FIAT
- Instant

## Installation

### Containers/Docker

A collection of Docker containers for FEniCS are available. See <http://fenics-containers.readthedocs.org/en/latest/> for how to run FEniCS inside a container.

© Copyright 2015, FEniCS Project Team. Revision a8a6ba83.  
Built with Sphinx using a theme provided by Read the Docs.

Read the Docs v:latest

# Development community is organized via bitbucket.org

The screenshot shows the Bitbucket web interface for the `fenics-project / DOLFIN` repository. The browser address bar shows the URL `https://bitbucket.org/fenics-project/dolfin`. The repository is a C++ project. The left sidebar contains navigation links: Overview (selected), Source, Commits, Branches, Pull requests (5), Issues (99), Wiki, Downloads (1), and Settings. The main content area is titled "Overview" and includes a table with repository statistics:

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

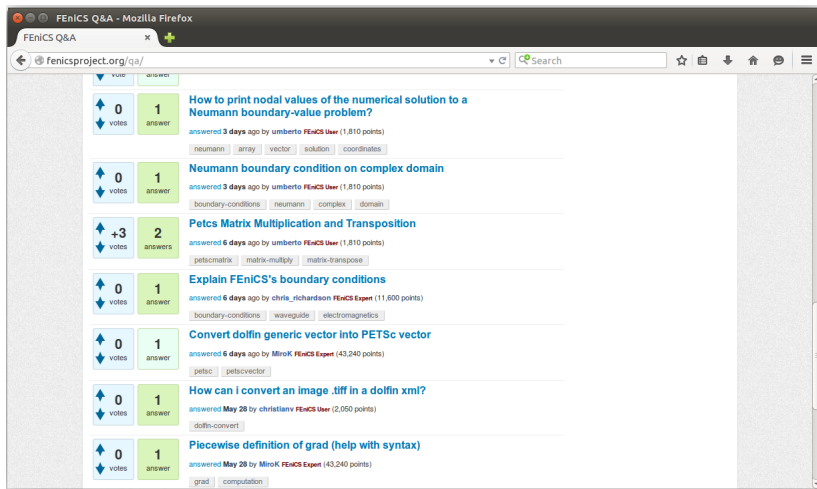
Below the table is a section for "DOLFIN" with a description: "DOLFIN is the C++/Python interface of FEniCS, providing a consistent PSE (Problem Solving Environment) for ordinary and partial differential equations." It also includes an "Installation" section with a code block:

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

For detailed instructions, see the file INSTALL. The "License" section states that DOLFIN is free software under the GNU Lesser General Public License. On the right, there is a "Recent activity" section showing three commits, each pushed by Garth Wells, including merge branches and replacing Boost lexical\_cast.

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

# Community help is available via QA forum



<https://fenicsproject.org/qa>

# Installation alternatives



☞ Nike Server (Simula internal)



☞ Docker images on Linux, Mac, Windows

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

# Installation using Docker

Follow instructions to install Docker on Linux, Mac, or Windows:

<https://docs.docker.com/linux/> or [mac/](https://docs.docker.com/mac/), [windows/](https://docs.docker.com/windows/)

Download and open a terminal in a clean FEniCS environment:

*Bash code*

```
$ curl -s http://get.fenicsproject.org | sh
```

*Bash code*

```
$ fenicsproject notebook suurph dev  
$ fenicsproject start suurph
```

More instructions on using FEniCS Docker images here:

<http://fenics-containers.readthedocs.org>

# Installation using Docker+fenicsproject script

Install Docker, then get the fenicsproject script:

*Bash code*

```
$ curl -s http://get.fenicsproject.org | sh
```

Now you can initialize and run in a clean FEniCS environment simply with:

*Bash code*

```
$ fenicsproject create myfenics dev  
$ fenicsproject start myfenics
```

Or start a Jupyter notebook in a clean environment with:

*Bash code*

```
$ fenicsproject notebook mynotebook dev-py3  
$ fenicsproject start mynotebook
```



# In this course we'll be running on a local Jupyter Notebook server

- Open `nike.simula.no` in a webbrowser
- Click “New”, “Python 3”
- Try entering some code:

*Python code*

```
from fenics import *

%matplotlib inline
parameters["plotting_backend"] = "matplotlib"

mesh = UnitCubeMesh(16, 16, 16)
plot(mesh)
```

# 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`

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

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

(Secret password needed!)