FEniCS Form Compiler (FFC) Documentation

Release 2017.1.0.post2

FEniCS Project

Sep 12, 2017

Contents

FFC is a compiler for finite element variational forms. From a high-level description of the form, it generates efficient low-level C++ code that can be used to assemble the corresponding discrete operator (tensor). In particular, a bilinear form may be assembled into a matrix and a linear form may be assembled into a vector. FFC may be used either from the command line (by invoking the `ffc` command) or as a Python module (`import ffc`).

FFC is part of the FEniCS Project.

For more information, visit http://www.fenicsproject.org

CHAPTER 1

Documentation

Installation

FFC is normally installed as part of an installation of FEniCS. If you are using FFC as part of the FEniCS software suite, it is recommended that you follow the installation instructions for FEniCS.

To install FFC itself, read on below for a list of requirements and installation instructions.

Requirements and dependencies

FFC requires Python version 2.7 or later and depends on the following Python packages:

- NumPy
- six

FFC also depends on the following FEniCS Python packages:

- FIAT
- UFL
- dijitso

These packages will be automatically installed as part of the installation of FFC, if not already present on your system.

TSFC requirements

To use experimental `tsfc` representation, additional dependencies are needed:

- TSFC[1]
- COFFEE[1]
- FInAT[1]

[1] These are forks of the original packages tested to be compatible with FFC and updated frequently from upstream.

and in turn their additional dependencies:

- singledispatch[2]
- networkx[2]
- PuLP[2],[4]
- GLPK[3],[4]

Note: TSFC requirements are not installed in FEniCS Docker images by default yet but they can be easilly installed on demand:

```
docker pull quay.io/fenicsproject/stable:2017.1.0
docker run -ti --rm quay.io/fenicsproject/stable:2017.1.0
sudo apt-get update && sudo apt-get -y install glpk-utils && \
  sudo pip2 install --prefix=${FENICS_PREFIX} --no-cache-dir \
  git+https://github.com/blechta/tsfc.git@2017.1.0 \
  git+https://github.com/blechta/COFFEE.git@2017.1.0 \
  git+https://github.com/blechta/FInAT.git@2017.1.0 \
  singledispatch networkx pulp && \
  sudo pip3 install --prefix=${FENICS_PREFIX} --no-cache-dir \
  git+https://github.com/blechta/tsfc.git@2017.1.0 \
  git+https://github.com/blechta/COFFEE.git@2017.1.0 \
  git+https://github.com/blechta/FInAT.git@2017.1.0 \
  singledispatch networkx pulp && \
  sudo apt-get clean && \
  sudo rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
```

The first two commands (or their modification, or `fenicsproject` helper script) are to be run on a host, while the last command, to be run in the container, actually installs all the TSFC requirements. For further reading, see FEniCS Docker reference.

Installation instructions

To install FFC, download the source code from the FFC Bitbucket repository, and run the following command:

```
pip install .
```

To install to a specific location, add the `--prefix` flag to the installation command:

```
pip install --prefix=<some directory> .
```

User manual

Note: This page is work in progress.

[2] Pip-installable.
[4] Needed for certain COFFEE optimizations.
[3] Binary package; `glpsol` executable needed. Version `GLPSOL: GLPK LP/MIP Solver, v4.57` from Ubuntu 16.04 `glpk-utils` package is known to produce the same references as our test system.

ffc package

Subpackages

ffc.backends package

Subpackages

ffc.backends.dolfin package

Submodules

ffc.backends.dolfin.capsules module

ffc.backends.dolfin.form module

ffc.backends.dolfin.functionspace module

ffc.backends.dolfin.goalfunctional module

ffc.backends.dolfin.includes module

ffc.backends.dolfin.wrappers module

Module contents

ffc.backends.ufc package

Submodules

ffc.backends.ufc.coordinate_mapping module

ffc.backends.ufc.dofmap module

ffc.backends.ufc.finite_element module

ffc.backends.ufc.form module

ffc.backends.ufc.function module

ffc.backends.ufc.integrals module

Module contents

Module contents

Module contents

Submodules

ffc.analysis module

ffc.codegeneration module

ffc.codesnippets module

ffc.compiler module

ffc.cpp module

ffc.enrichedelement module

ffc.evaluatebasis module

ffc.evaluatebasisderivatives module

ffc.evaluatedof module

ffc.extras module

ffc.fiatinterface module

ffc.formatting module

ffc.interpolatevertexvalues module

ffc.jitcompiler module

ffc.log module

ffc.main module

ffc.mixedelement module

ffc.optimization module

ffc.parameters module

ffc.plot module

ffc.quadratureelement module

ffc.representation module

ffc.representationutils module

ffc.restrictedelement module

ffc.utils module

ffc.wrappers module

Summary of changes

Note: Developers should use this page to track and list changes during development. At the time of release, this page should be published (and renamed) to list the most important changes in the new release.

Detailed changes

Note: At the time of release, make a verbatim copy of the ChangeLog here (and remove this note).

Changes in version 2017.1.0

FFC 2017.1.0.post2 was released on 2017-09-12.

Summary of changes

- Change PyPI package name to fenics-ffc.

Changes in version 2017.1.0

FFC 22017.1.0 was released on 2017-05-09.

Summary of changes

- Add experimental `tsfc` representation; for installation see *TSFC requirements*

Detailed changes

- Let ffc -O parameter take an optional integer level like -O2, -O0
- Implement blockwise optimizations in uflacs code generation
- Expose uflacs optimization parameters through parameter system

Changes in version 2016.2.0

FFC 2016.2.0 was released on 2016-11-30.

Summary of changes

- Generalize ufc interface to non-affine parameterized coordinates
- Add `ufc::coordinate_mapping` class
- Make ufc interface depend on C++11 features requiring gcc version >= 4.8

- Change the mapping `pullback as metric` to `double covariant piola` (this preserves tangential-tangential trace).
- Added Hellan-Herrmann-Johnson element as supported element
- Add mapping `double contravariant piola` (this preserves normal-normal trace).
- Include comment with effective representation and integral metadata to generated `tabulate_tensor` code

Detailed changes

- Jit compiler now compiles elements separately from forms to avoid duplicate work
- Add parameter max_signature_length to optionally shorten signatures in the jit cache
- Move uflacs module into ffc.uflacs
- Remove installation of pkg-config and CMake files (UFC path and compiler flags are available from ffc module)
- Add dependency on dijitso and remove dependency on instant
- Add experimental Bitbucket pipelines
- Tidy the repo after UFC and UFLACS merge, and general spring cleanup. This includes removal of instructions how to merge two repos, commit hash c8389032268041fe94682790cb773663bdf27286.

Changes in version 2016.1.0

FFC 2016.1.0 was released on 2016-06-23.

- Add function get_ufc_include to get path to ufc.h
- Merge UFLACS into FFC
- Generalize ufc interface to non-affine parameterized coordinates
- Add ufc::coordinate_mapping class
- Make ufc interface depend on C++11 features requiring gcc version >= 4.8
- Add function ufc_signature() to the form compiler interface
- Add function git_commit_hash()

Changes in version 1.6.0

FFC 1.6.0 was released on 2015-07-28.

- Rename and modify a number of UFC interface functions. See docstrings in ufc.h for details.
- Bump required SWIG version to 3.0.3
- Disable dual basis (`tabulate_coordinates` and `evaluate_dofs`) for enriched elements until correct implementation is brought up

[FIXME: These links don't belong here, should go under API reference somehow.]

- genindex
- modindex