The FEniCS Project (SOFA Talk)

Michal Habera, **Jack S. Hale**, Chris Richardson, Johannes Ring, Marie E. Rognes, Nathan Sime, Garth N. Wells and the FEniCS Project contributors.

https://fenicsproject.org

SOFA Technical Committee, 23rd February 2021.

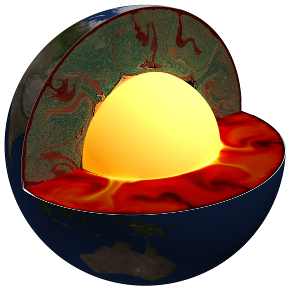
This talk takes place in the context of the DRIVEN project funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 811099.

Outline

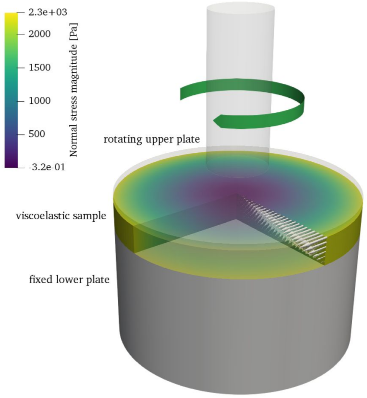
1. What is the FEniCS Project?
2. Key features
3. Tour of applications
4. Live demo
5. Summary

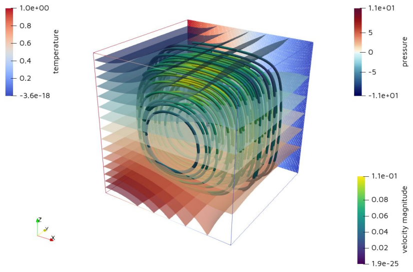
What is the FEniCS Project?

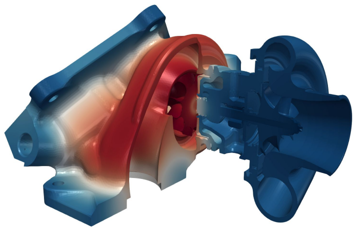
What is the FEniCS Project?



- A computing platform for translating *scientific models* into *finite element simulations*.
- 15 year history.
- Open Source (LGPLv3 and MIT).







What is the FEniCS Project for?

- For rapidly prototyping and testing novel finite element methods.
- For rapidly prototyping numerical solvers to test scientific hypotheses.
- For writing robust, scalable and maintainable finite element solvers in the context of larger engineering/scientific software projects.
- For teaching finite element methods to students.

Who is aimed at?

- Using the FEniCS Project requires domain specific knowledge of:
  - finite element methods,
  - partial differential equations,
  - the particulars of the problem you are trying to solve!
- Students, academics, engineers and scientists have all used the FEniCS Project successfully to solve problems of real-world interest.

- It is *not* an end-to-end finite element analysis package in the style of products from e.g. Ansys, Dassault or Comsol.
- The FEniCS Project has been used to write finite element solvers within end-to-end analysis packages aimed at users *without* domain specific knowledge.

Who is currently involved?

Legal

A Fiscally Sponsored Project of
NUMFOCUS
OPEN CODE = BETTER SCIENCE

Institutions

simula

UNIVERSITY OF CAMBRIDGE

uni.lu
UNIVERSITÉ DU LUXEMBOURG

CARNEGIE SCIENCE

Community

How can I try it out?

- Docker containers (x86-64 and ARM64)

```
docker run -ti dolfinx/dolfinx
```

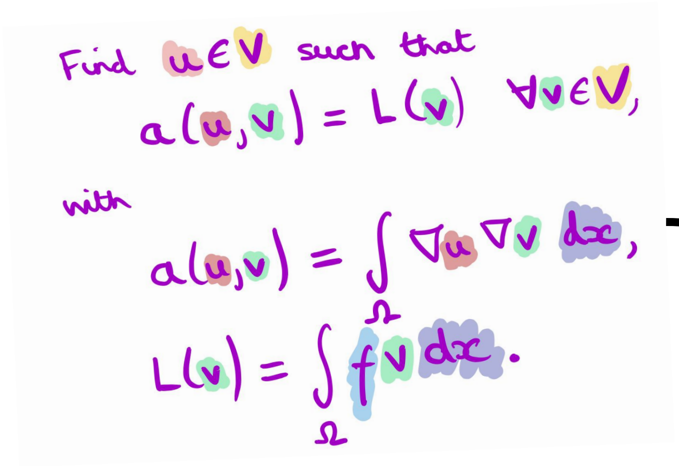
- Spack package manager (HPC)

```
spack add py-fenics-dolfinx ^petsc+mumps+hypre cflags="-O3"
fflags="-O3"

spack install
```

Key features re:
SOFA

Write mathematics as code




```
u = TrialFunction(V)
v = TestFunction(V)

f = Coefficient(V)

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

Automatic differentiation

- Automatic differentiation (AD) capabilities remove tedious and error-prone manual differentiation steps.
- AD operates at the variational level (c.f. TensorFlow, JAX etc.).

$$\psi(\underset{\sim}{u}) = \frac{\mu}{2}(I_c - 3) - \mu \ln(J) + \frac{\lambda}{2}\ln(J)$$

$$\min_{u \in V} \int_\Omega \psi(\underset{\sim}{u})\, dx - \int f \cdot \underset{\sim}{u}\, dx := \Pi(\underset{\sim}{u})$$

$$D_{\underset{\sim}{u}}[\Pi(\underset{\sim}{u})][\underset{\sim}{v}] = 0 \quad \forall \underset{\sim}{v} \in V$$

```
u = Function(V)

psi = ...

Pi = psi*dx - f*u*dx

v = TestFunction(V)
F = derivative(Pi, u, v)

u_bar = TrialFunction(V)
J = derivative(F, u, u_bar)
```

Automatic code generation

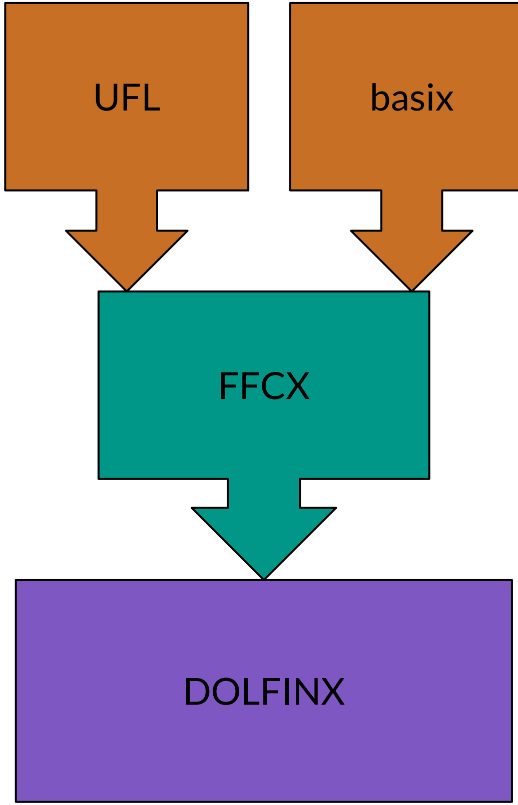
```
u = TrialFunction(V)
v = TestFunction(V)

f = Coefficient(V)

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

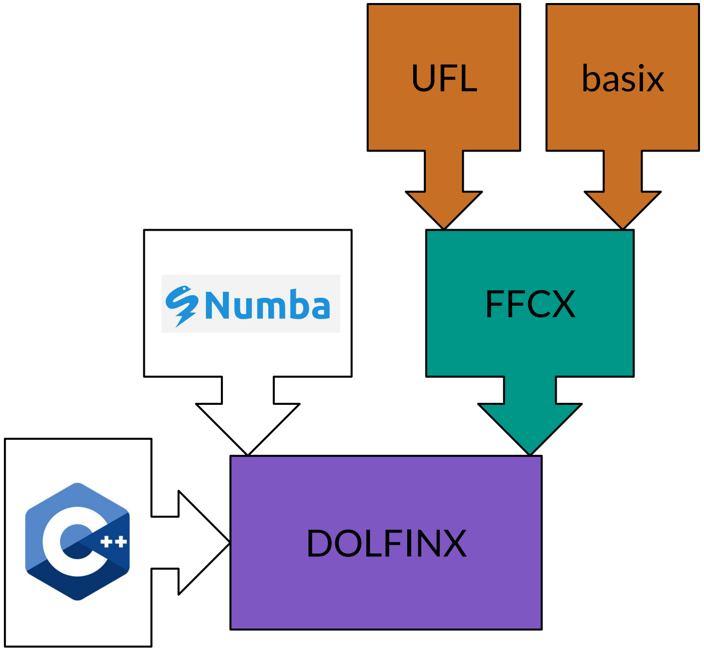
```
void tabulate_tensor_poisson_cell_integral_43165b9e21c850b7e46d14f843b
                                      const ufc_scalar_t* c,
                                      const double* restrict coordinate_
                                      const int* unused_local_index,
                                      const int* cell_orientation)
{
    // Precomputed values of basis functions and precomputations
    // FE* dimensions: [entities][points][dofs]
    // PI* dimensions: [entities][dofs][dofs] or [entities][dofs]
    // PM* dimensions: [entities][dofs][dofs]
    alignas(32) static const ufc_scalar_t FE3_C0_D01_Q1[1][1][2] = { {
    // Unstructured piecewise computations
    const double J_c0 = coordinate_dofs[0] * FE3_C0_D01_Q1[0][0][0] +
    const double J_c3 = coordinate_dofs[1] * FE3_C0_D01_Q1[0][0][0] +
    const double J_c1 = coordinate_dofs[0] * FE3_C0_D01_Q1[0][0][0] +
    const double J_c2 = coordinate_dofs[1] * FE3_C0_D01_Q1[0][0][0] +
```

Modular



Python/C++.	Abstract.
Python. Generates C code.	Transform.
C++ and Python.	Orchestrate/Execute.

Extendable



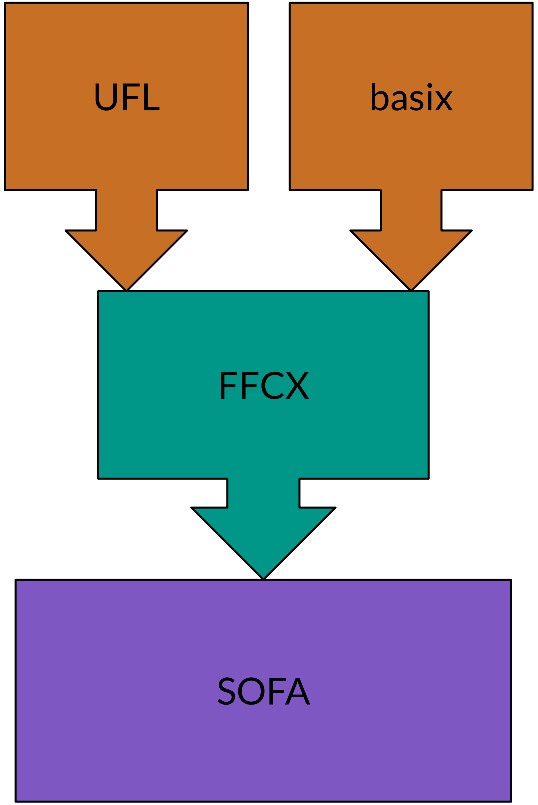
- The FEniCS Project has been designed to be extendable.
- Use automated tools 90% of the time.
- 10% of the time automated tools may not meet needs. Allows fully custom implementations of:
  - Finite element kernels.
  - Assemblers.
  - Input/Output.
  - Linear algebra backends.

Modular. SOFA?

Python.

Python. Generates C code.

C++ and Python.



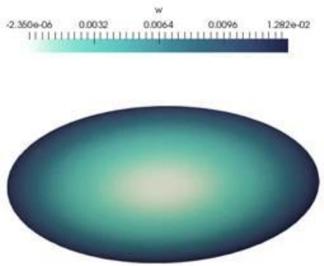
Abstract.

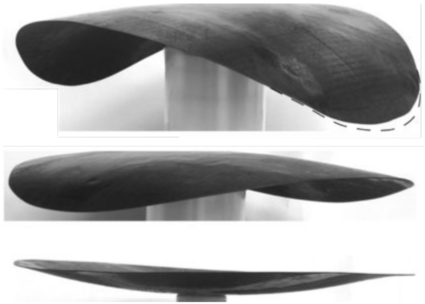
Transform.

Orchestrate/Execute.

Tour of applications

Thin Structures





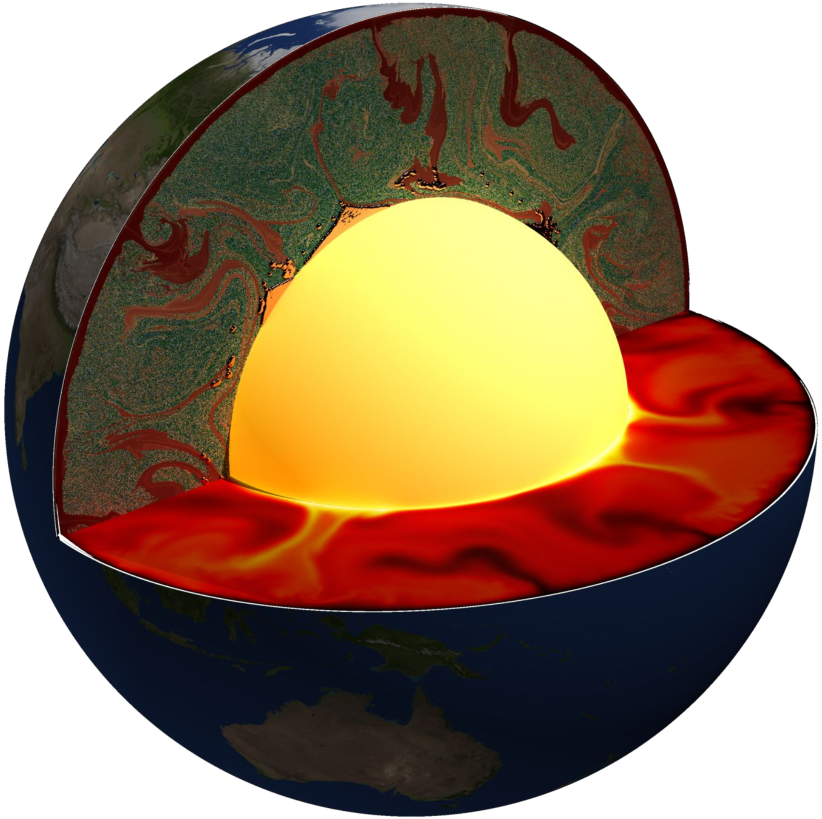
- FEniCS-Shells is a FEniCS based library for simulating thin structures.
- Bi-stable shell structure.
- Simulation.
    - Lenticular plate is heated.
    - Deforms with initially spherical curvature.
    - Snaps through to developable surface.
- SIGGRAPH 2021/ACM TOG: Top Engineering validation code for thin structures (https://hal.inria.fr/hal-03217459).

Hale, Brunetti, Bordas, Maurini,

Simple and extensible plate and shell finite element models through automatic code generation capabilities,

Computers and Structures, Volume 209, October 2018, 163-181.

Geodynamics



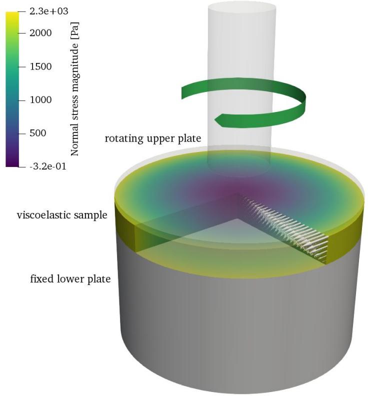
- Transient Boussinesq approximation of the Earth's compositional evolution.
- Tracer advection provided by LEoPart add-on to FEniCS.
- Age of compositional material (top), mantle temperature (bottom).
- Primordial material collects into persistent "piles" at the core-mantle boundary (specular highlight).

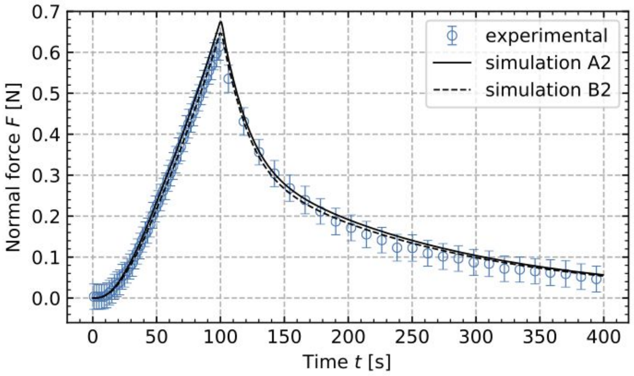
T. D. Jones, N. Sime and P. E. van Keken,

Burying Earth's primitive mantle in the slab graveyard

Geochemistry, Geophysics, Geosystems (accepted).

Viscoelastic flow characterisation





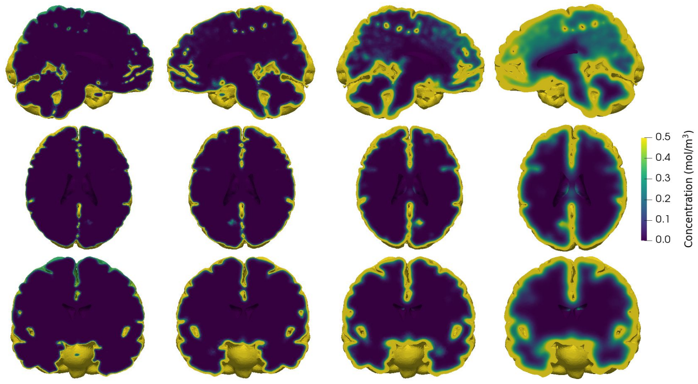
- Model of heated styrene butadiene rubber flowing in a rotational rheometer.
- Proposed three two time-scale viscoelastic constitutive models.
- Used for parameter estimation and model selection.

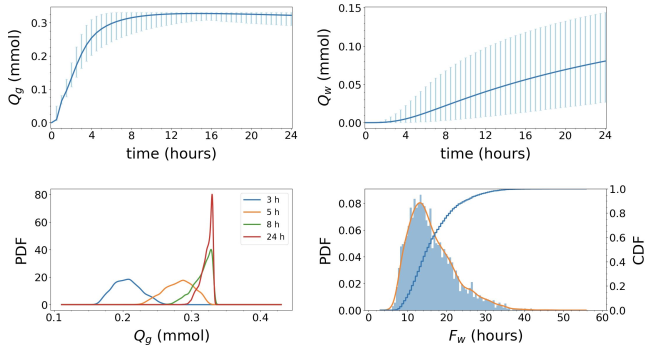
Řehoř, Gansen, Sill, Polińska, Westermann, Dheur, Baller, Hale,

A comparison of constitutive models for describing the flow of styrene-butadiene rubber,

Journal of Non-Newtonian Fluid Mechanics, Volume 286, 104398, (2020).

Biomechanics of the Human Brain





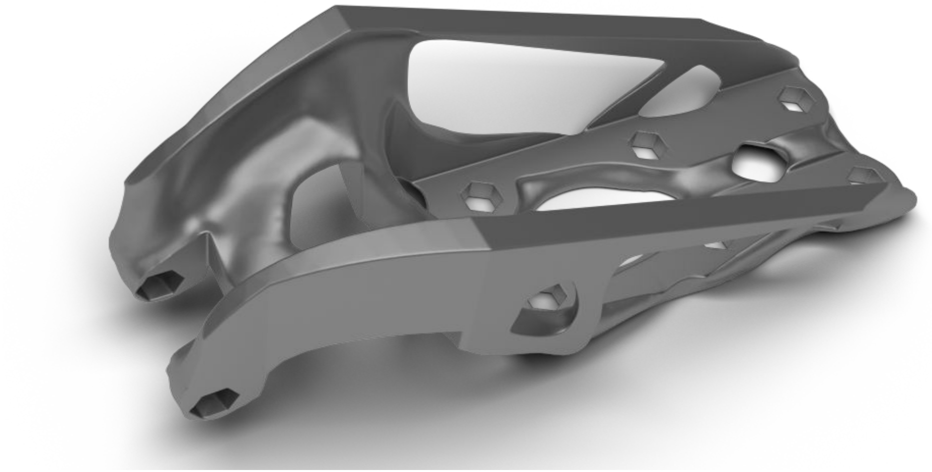
- The human brain contains a complex lymphatic system for draining unwanted substances.
- The precise mechanisms are poorly understood.
- Simulation of injected tracers in three-dimensional geometry.
- Quantified competing effects of advection and diffusion.

Croci, Vinje, Rognes,

Uncertainty quantification of parenchymal tracer distribution using random diffusion and convective velocity fields

Fluids and Barriers of the CNS, Volume 16, 32 (2019).

Stochastic Topology Optimisation



- Rafinex's *Stochastic Topology Optimisation Tool* can automatically produce robust designs that behave safely even when subjected to rare loading conditions.
- Use FEniCS Project as scalable and efficient solid mechanics solver at core of engineering-ready product.

rafinex

Luxembourg National Research Fund

Rafinex Sarl

https://www.rafinex.com/

Live demo

Summary

Summary

- The FEniCS Project allows the quick, efficient and flexible translation of scientific models into finite element simulators.
- Key features include:
  - **Writing mathematics as code.**
  - **Automatic differentiation.**
  - HPC capable.
  - Modular and extendable.
- It can be used to write simulators for a huge range of problems.
- It is Open Source and available for use today.
- What are the possibilities for automatic code generation in SOFA?

Questions?