Introduction to Firedrake

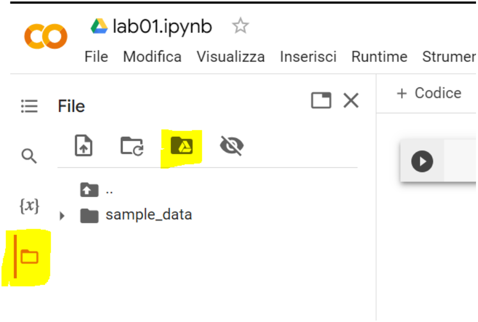
Ivan Fumagalli

Computational Fluid Dynamics
A.Y. 2023/2024

Firedrake on Colab

Requirements: a browser and a Google account

1. Go to https://colab.research.google.com and create a new notebook.
2. Mount your Google Drive space (pop-up window could appear; after, you may need to **reload** the notebook).



Firedrake on Colab

Requirements: a browser and a Google account

1. Go to `https://colab.research.google.com` and create a new notebook.
2. Mount your Google Drive space (pop-up window could appear; after, you may need to **reload** the notebook).
3. Install **Firedrake** as in `https://fem-on-colab.github.io/packages.html` **takes a while ...**; you may need to run the cell **twice**).

Firedrake on Colab

Requirements: a browser and a Google account

1. Go to https://colab.research.google.com and create a new notebook.
2. Mount your Google Drive space (pop-up window could appear; after, you may need to **reload** the notebook).
3. Install **Firedrake** as in https://fem-on-colab.github.io/packages.html **takes a while ...**; you may need to run the cell **twice**).
4. Have you obtained this? You are ready to go!

```
##################################################################################
#     This installation is offered by FEM on Colab, an open-source project      #
#       developed and maintained at Università Cattolica del Sacro Cuore        #
#    by Dr. Francesco Ballarin. Please see https://fem-on-colab.github.io/      #
#       for more details, including a list of further available packages       #
#       and how to sponsor the development or contribute to the project.        #
#                                                                                #
#   We are conducting an informal survey on FEM on Colab usage by our users.    #
#   The survey is anonymous, and its compilation will typically only require    #
#   a couple of minutes of your time. If you wish, give us your feedback at     #
#                     https://forms.gle/36sZZWNvPpUv8XWr7                        #
##################################################################################
```

Firedrake on Colab

Requirements: a browser and a Google account

1. Go to https://colab.research.google.com and create a new notebook.
2. Mount your Google Drive space (pop-up window could appear; after, you may need to **reload** the notebook).
3. Install **Firedrake** as in https://fem-on-colab.github.io/packages.html **takes a while ...**; you may need to run the cell **twice**).
4. Have you obtained this? You are ready to go!

```
################################################################################
#     This installation is offered by FEM on Colab, an open-source project     #
#       developed and maintained at Università Cattolica del Sacro Cuore       #
#    by Dr. Francesco Ballarin. Please see https://fem-on-colab.github.io/     #
#       for more details, including a list of further available packages       #
#       and how to sponsor the development or contribute to the project.       #
#                                                                              #
#   We are conducting an informal survey on FEM on Colab usage by our users.   #
#   The survey is anonymous, and its compilation will typically only require   #
#   a couple of minutes of your time. If you wish, give us your feedback at   #
#                       https://forms.gle/36sZZWNvPpUv8XWr7                    #
################################################################################
```

5.

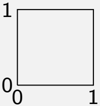
```python
from firedrake import *
import numpy as np
import matplotlib.pyplot as plt
```

Steps of FEM implementation

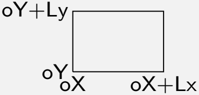
1. mesh generation
2. creation of finite element spaces
3. assembling matrices and vectors
4. solving the algebraic problem
5. save or visualize the solution

Mesh generation

Rectangular domains




```
mesh = UnitSquareMesh(nx, ny,
       diagonal='left',
       quadrilateral=False)
```

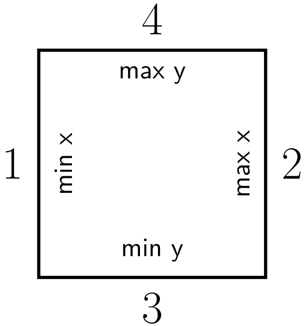


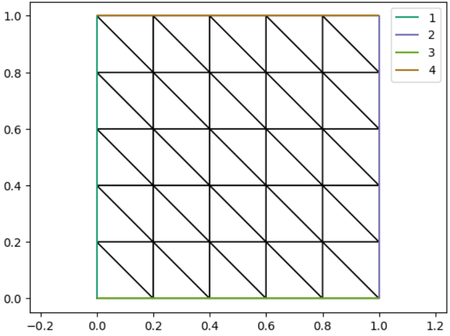

```
mesh = RectangleMesh(nx, ny,
       Lx, Ly,
       originX=0.0, originY=0.0,
       diagonal='left',
       quadrilateral=False)
```

`nx`, `ny`: subdivisions along x and y axis

Mesh generation

Boundary labels






```
fig, ax = plt.subplots()
triplot(mesh, axes=ax)
ax.legend()
```

Finite element spaces

```
V = FunctionSpace(mesh, family,
    degree)
```

`family` = 'P'/'CG'/Lagrange' (continuous FE),
'DG'/'Discontinuous Lagrange', ...
(full list here)

`degree` = 0 (NOT for 'CG'), 1, 2, ...

N.B. We use the FunctionSpace by Firedrake (documented here), not the one in UFL (see here).

The diffusion-reaction problem

Weak formulation

$$\begin{cases} -\Delta u + u = f, & \text{in } \Omega := [0,1]^2, \\ \nabla u \cdot \mathbf{n} = 0, & \text{on } \partial\Omega. \end{cases}$$

The diffusion-reaction problem

Weak formulation

$$\begin{cases} -\Delta u + u = f, & \text{in } \Omega := [0,1]^2, \\ \nabla u \cdot \mathbf{n} = 0, & \text{on } \partial\Omega. \end{cases}$$

Find $u \in V := H^1(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla v + uv \, d\mathbf{x} = \int_\Omega fv \, d\mathbf{x} \qquad \forall v \in V.$$

The diffusion-reaction problem

Galerkin projection - Finite Element Method

mesh $\mathcal{T}_h$ of Ω

$X_h^p = \{u \in C^0(\overline{\Omega}) \colon u|_K \in \mathbb{P}^p(K) \quad \forall K \in \mathcal{T}_h\}$

$V_h = V \cap X_h^1$

Find $u_h \in V_h$ such that

$$\int_\Omega \nabla u_h \cdot \nabla v_h + u_h v_h \, d\mathbf{x} = \int_\Omega f v_h \, d\mathbf{x} \quad \forall v_h \in V_h$$

The diffusion-reaction problem

Firedrake code

```
u = TrialFunction(V) # placeholders
v = TestFunction(V)  # not func's

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

u_h = Function(V) # actual function
solve(a == L, u_h)
```

The diffusion-reaction problem

Non-homogeneous Neumann BC - Weak formulation

$$\begin{cases} -\Delta u + u = 0, & \text{in } \Omega := [0,1]^2, \\ \nabla u \cdot \mathbf{n} = d, & \text{on } \partial\Omega. \end{cases}$$

The diffusion-reaction problem

Non-homogeneous Neumann BC - Weak formulation

$$\begin{cases} -\Delta u + u = 0, & \text{in } \Omega := [0,1]^2, \\ \nabla u \cdot \mathbf{n} = d, & \text{on } \partial\Omega. \end{cases}$$

Find $u \in V := H^1(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla v + uv \, d\mathbf{x} = \int_{\partial\Omega} d \, v \, ds \qquad \forall v \in V.$$

The diffusion-reaction problem

Non-homogeneous Neumann BC - Firedrake code

```
a = dot(grad(u), grad(v)) * dx
    + u * v * dx
L = d * v * ds
# L = d*v * ds(1) + d*v * ds(2)
#       + d*v * ds(3) + d*v * ds(4)

u_h = Function(V)
solve(a == L, u_h)
```

Visualization

matplotlib

```
fig, ax = plt.subplots()
q = tripcolor(u_h, axes=ax)
fig.colorbar(q) # show the colorbar
```

ParaView

Software for enhanced visualization
(later in the course)
https://www.paraview.org/download

Algebraic formulation of FEM

Let $V_h \subset V$ be a finite element space with dimension N_h and $\{\varphi_i\}_{i=1}^{N_h}$ its basis. Then, the FE approximation of the problem is:
Find $u_h = \sum_{j=1}^{N_h} u_j \varphi_j \in V_h$ such that

$$\int_\Omega (\nabla u_h \cdot \nabla \varphi_i + u_h \varphi_i) d\mathbf{x} = \int_{\partial\Omega} d\,\varphi_i\, ds$$

$$\forall i = 0, \ldots, N_h$$

Algebraic formulation of FEM

$$\int_\Omega \sum_{j=0}^{N_h} u_j(\nabla \phi_j \cdot \nabla \varphi_i + \varphi_j \varphi_i) d\mathbf{x} = \int_{\partial\Omega} d\,\varphi_i\, ds$$

$$\forall i = 0, \ldots, N_h$$

$$\Updownarrow$$

$$A\mathbf{U} = \mathbf{b} \text{ with}$$

$$A_{ij} = \int_\Omega (\nabla \varphi_j \cdot \nabla \varphi_i + \varphi_j \varphi_i) d\mathbf{x},$$

$$b_i = \int_{\partial\Omega} d\,\varphi_i\, ds$$

Algebraic formulation of FEM

Firedrake code

```
A = assemble(a)
rhs = assemble(L)

solver = LinearSolver(A,
             solver_parameters={...})

u_h = Function(V)
solver.solve(u_h, rhs)
```

The diffusion-reaction problem

Dirichlet boundary conditions

$$\begin{cases} -\Delta u + u = 0, & \text{in } \Omega := [0,1]^2, \\ u = g, & \text{on } \partial\Omega. \end{cases}$$

The diffusion-reaction problem

Dirichlet boundary conditions

$$\begin{cases} -\Delta u + u = 0, & \text{in } \Omega := [0,1]^2, \\ u = g, & \text{on } \partial\Omega. \end{cases}$$

$$V_0 = H_0^1(\Omega), \; V_{\mathrm{D}} = \{v \in H^1(\Omega) \colon v|_{\partial\Omega} = g\},$$

lifting operator $\mathcal{R} : H^{1/2}(\Omega) \to V_{\mathrm{D}}$

The diffusion-reaction problem

Dirichlet boundary conditions

$$\begin{cases} -\Delta u + u = 0, & \text{in } \Omega := [0,1]^2, \\ u = g, & \text{on } \partial\Omega. \end{cases}$$

$$V_0 = H^1_0(\Omega), \;\; V_\mathrm{D} = \{v \in H^1(\Omega) \colon v|_{\partial\Omega} = g\},$$

lifting operator $\mathcal{R} : H^{1/2}(\Omega) \to V_\mathrm{D}$

Find $u = \check{u} + \mathcal{R}g \in V_\mathrm{D}$ with $\check{u} \in V_0$ s.t.

$$a(\check{u}, v) = F(v) - a(\mathcal{R}g, v) \quad \forall v \in V_0$$

The diffusion-reaction problem

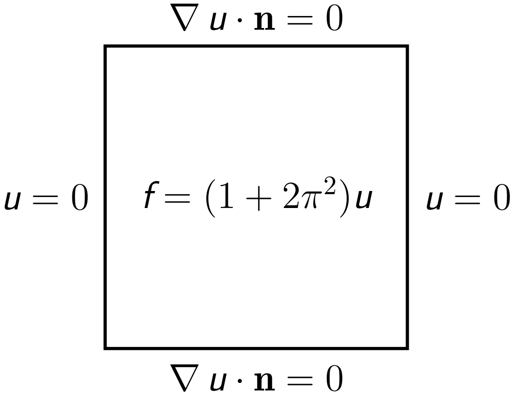
Dirichlet boundary conditions - Firedrake code

```python
x = SpatialCoordinate(mesh)
g = x[0] * sin(x[1]) # example

bc = DirichletBC(V, g, [1,2,3,4])
bcs = list(bc)
solve(a == L, u_h, bcs=bcs,
      solver_parameters={...})
```

Exercise

$$-\Delta u + u = f \quad \text{in } \Omega = [0,1]^2$$



$$u_{\text{exact}} = \sin(\pi x)\cos(\pi y)$$