Nečas Center for Mathematical Modeling

Numerical solution of continuum physics problems with FEniCS or how to use FEM and not vary (too much) about programming ...

Jaroslav Hron

Mathematical Institute, Charles University in Prague

CHARLES UNIVERSITY PRAGUE

faculty of mathematics and physics

Numerical solution of continuum physics problems with FEniCS or how to use FEM and not vary (too much) about programming ...

✓	**lecture 1**	introduction to FEniCS/python, FEM and how to solve laplace equation
☞	**lecture 2**	boundary conditions, time discretization, convection-diffusion equation, (stabilization)
	lecture 3	Stokes, incompressible Navier-Stokes equations
	lecture 4	linear and non-linear elasticity
	lecture 5	ALE-method, level-set method

How to log in to the cluster ....

```
ssh -Y -C IName@10.4.8.14
```

Poisson's problem in FEniCS

$$-\operatorname{div}(\nabla u) = f \quad \text{in } \Omega = [0,1]^2, \qquad u = 0 \quad \text{on } \partial\Omega$$

- $\Omega = [0,1]^2$
- Lagrange FEM:

  $$V_h = \{v \in C^0(\Omega), v/T \in P_1(T)\}$$

- Dirichlet boundary condition
  $v \in V_h \to v = 0$ on $\partial\Omega$
- $f(x,y) = 1.0$
- $a(u,v) = \int_\Omega \nabla u \cdot \nabla v \, dx$
- $L(v) = \int_\Omega fv \, dx$
- Find $u \in V_h$ such that
  $a(u,v) = L(v)$ for all $v \in V_h$.

```
from dolfin import *

mesh = UnitSquare(10, 10)
V = FunctionSpace(mesh, "Lagrange", 1)

u0 = Constant(0.0)
bc = DirichletBC(V, u0, DomainBoundary())

u = TrialFunction(V)
v = TestFunction(V)
f = Expression("1.0")

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

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

Poisson's problem in FEniCS

- plot the mesh, solution

```
plot(mesh, interactive=True)
plot(u, interactive=True)

file=File("poisson.pvd")
u.rename("u","solution")
file << u
```

- interpolation, projection

```
f=Expression("sin(x[0])*sin(x[1])")
u=interpolate(f,V)
u=project(f,V)
plot(u)
```

- assemble integrals

```
uex=Expression("....",degree=2)
U=project(uex,FunctionSpace(mesh,"CG',2)
e=sqrt(assemble(pow(u-U,2)*dx))
err=errornorm(uex, u, norm_type='l2', degree_rise=1, mesh=mesh)
print "error:", e , err
```

Heat equation

$$
\begin{aligned}
\frac{\partial u}{\partial t} - \operatorname{div}(K \nabla u) &= f \qquad && \text{in } \Omega = [0,1]^2 \\
u &= u_D && \text{on } \Gamma_D \\
K \nabla u \cdot \mathbf{n} &= g && \text{on } \Gamma_N
\end{aligned}
$$

- time discretization:
  $\frac{\partial u}{\partial t} \approx \frac{u(t+dt)-u(t)}{dt} \qquad \Rightarrow \frac{u^n - u^{n-1}}{dt} + \theta A(u^n) + (1-\theta)A(u^{n-1}) = 0$
- weak form (let $\theta = 1$ for implicit Euler method): Given u^{n-1} find u^n such that

$$\int_\Omega \frac{u^n - u^{n-1}}{dt} v \,\mathrm{d}x + a(u^n, v) = L(v)$$

- with

$$a(u,v) = \int_\Omega K \nabla u \cdot \nabla v \,\mathrm{d}x, \qquad L(v) = \int_\Omega f v \,\mathrm{d}x + \int_{\Gamma_N} K \nabla u \cdot \mathbf{n} v \,\mathrm{d}s$$

  where we used the identity
  $-\int_\Omega \operatorname{div}(K \nabla u) v \,\mathrm{d}x = \int_\Omega K \nabla u \cdot \nabla v \,\mathrm{d}x - \int_{\Gamma_N} K \nabla u \cdot \mathbf{n} v \,\mathrm{d}s$

Task no.2 - Heat equation on moving medium

Task no.2: implement following problem with $\mathbf{b} = (-(x[1] - 0.5), x[0] - 0.5)$ (i.e. rigid body rotation) and right-hand side (volumetric heating, i.e. microwave)

1. add to previous formulation the term $\mathbf{b} \cdot \nabla u$ where $\mathbf{b}$ is given velocity vector of the underlining medium motion

$$\begin{aligned} \frac{\partial u}{\partial t} + \mathbf{b} \cdot \nabla u - \operatorname{div}(K \nabla u) &= f && \text{in } \Omega = [0,1]^2 \\ u &= u_D && \text{on } \Gamma_D \\ K \nabla u \cdot \mathbf{n} &= g && \text{on } \Gamma_N \end{aligned}$$

$$f(x) = \begin{cases} 1 & \text{if dist}(x, 0.75) < 0.2 \\ 0 & \text{otherwise} \end{cases}$$

```
heating=Expression("((pow(x[0]-0.75,2)+pow(x[1]-0.75,2))<0.2*0.2)?(1.0):(0.0)")
f=project(heating,V)
g=Constant(0.0)
b = Expression( ("-(x[1]-0.5)","(x[0]-0.5)") )
```