poisson

November 3, 2022

1 Poisson Equation

Import Netgen/NGSolve Python modules:

[1]:
```
from ngsolve import *
# from netgen.geom2d import unit_square
from netgen.occ import unit_square
from ngsolve.webgui import Draw
import matplotlib as plt
```

```
importing NGSolve-6.2.2204
```

The unit_square is a predefined domain, use Netgen to generate a mesh:

[2]:
```
mesh = Mesh(unit_square.GenerateMesh(maxh=0.2))
Draw (mesh);
```

```
WebGuiWidget(layout=Layout(height='50vh', width='100%'), value={'gui_settings':␣
↪{}, 'ngsolve_version': '6.2.22…
```

Define a finite element space on that mesh.

[3]:
```
fes = H1(mesh, order=3, dirichlet="left|right|bottom|top")
print ("ndof =", fes.ndof)
```

```
ndof = 283
```

Define linear and bilinear-forms.

$$a(u,v) = \int \nabla u \nabla v \qquad \text{and} \qquad f(v) = \int fv$$

Forms are expressed in terms of trial and test-functions:

[4]:
```
u = fes.TrialFunction()
v = fes.TestFunction()

f = LinearForm(fes)
f += 32*(y*(1-y)+x*(1-x))*v*dx

a = BilinearForm(fes)
```

```
a += grad(u)*grad(v)*dx

a.Assemble()
f.Assemble();
```

[ ]:
```
print(f.vec)
print(a.mat)
```

Solve the problem:

[6]:
```
gfu = GridFunction(fes)
gfu.vec.data = a.mat.Inverse(fes.FreeDofs()) * f.vec
```

Plot the solution:

[7]:
```
Draw (gfu, mesh);
```

```
WebGuiWidget(layout=Layout(height='50vh', width='100%'), value={'gui_settings':␣
↪{}, 'ngsolve_version': '6.2.22…
```

[8]:
```
Draw (-grad(gfu), mesh, "Flux", vectors= { "grid_size" : 40});
```

```
WebGuiWidget(layout=Layout(height='50vh', width='100%'), value={'gui_settings':␣
↪{}, 'ngsolve_version': '6.2.22…
```

Calculate error:

[9]:
```
exact = 16*x*(1-x)*y*(1-y)
print ("L2-error:", sqrt(Integrate((gfu-exact)**2, mesh)))
```

```
L2-error: 5.579392445191467e-05
```

[10]:
```
Draw (exact, mesh);
```

```
WebGuiWidget(layout=Layout(height='50vh', width='100%'), value={'gui_settings':␣
↪{}, 'ngsolve_version': '6.2.22…
```