Computational exercice 2: Error, Interpolation & Projection

Consider the Poisson problem on a computational domain $\Omega = [0,1] \times [0,1]$, source term f and thermal diffusivity κ: Find u satisfying

$$\begin{cases} -\nabla \cdot (\kappa(\mathbf{x})\, \nabla u(\mathbf{x})) &= f(\mathbf{x}) & \mathbf{x} \in \Omega \\ u(\mathbf{x}) &= \cos(2\pi x_1)\cos(2\pi x_2) & \mathbf{x} \in \partial\Omega \end{cases}$$

Taking $\kappa = 1$, $f = 8\pi^2 \cos(2\pi x_1)\cos(2\pi x_2)$, the exact solution reads

$$u(\mathbf{x}) = \cos(2\pi x_1)\cos(2\pi x_2)$$

The solution u_h is sought in a finite dimensional space $V_h \subset H^1(\Omega)$, made up of polynomial functions of degree k. The variational formulation reads: "Find $u_h \in V_h$ such that

$$\int_\Omega \kappa\, \nabla u_h \cdot \nabla v_h \, d\mathbf{x} = \int_\Omega f\, v_h \, d\mathbf{x}$$

forall $v_h \in V_{h0}$".

1. Compute the error in the $L^2(\Omega)$ and $H^1(\Omega)$-norms, i.e.,

$$\| u - u_h \|_{L^2(\Omega)} = \sqrt{\int_\Omega (u - u_h)^2 \, d\mathbf{x}},$$

$$\| u - u_h \|_{H^1(\Omega)} = \sqrt{\int_\Omega \left[(u - u_h)^2 + (\nabla(u - u_h))^2\right] d\mathbf{x}}$$

Perform a mesh refinement study and plot the error norms as a function of mesh size taking P_k spaces with $k = 1, 2, 3$. The error can be computed using the `assemble` function or the built-in function `errornorm()`, e.g.

```
.
myerror = errornorm(uexact, u, 'L2')
.
```

Notice that as you refine the mesh and the polynomial degree the default direct solver should be replaced by an interative one. This can be done in Fenics as follows:

```
.
problem = LinearVariationalProblem(a, L, w, bcall)
solver = LinearVariationalSolver(problem)
prm = solver.parameters
linear_solver = 'Krylov'
if linear_solver == 'Krylov':
   prm["linear_solver"]= "gmres"
   prm["preconditioner"]= "jacobi"
   prm["krylov_solver"]["absolute_tolerance"]= 1e-11
   prm["krylov_solver"]["relative_tolerance"]= 1e-10
   prm["krylov_solver"]["maximum_iterations"]= 10000
   prm["krylov_solver"]["monitor_convergence"]= True
   prm["krylov_solver"]["nonzero_initial_guess"]= True
 else:
    prm["linear_solver"]= "mumps"
```

Extract conclusions and discuss the results in terms of accuracy, computational cost, number of unknowns, and (eventually) iterations needed for convergence of the linear iterative solver.

2. Interpolate the exact solution into the space V_h and compute the interpolation error in the same norms:

```
.
Pk = FiniteElement("Lagrange", 'triangle', order)
W = FunctionSpace(mesh, Pk)
uinter = interpolate(uexact, W)
.
```

3. Repeat Item 1 but now using quadrilateral elements. A square mesh with `nx*ny` elements of quadrilateral type can be defined by using

```
.
Qk = FiniteElement("Lagrange", 'quadrilateral', order)
mesh = UnitSquareMesh.create(nx, ny, CellType.Type.quadrilateral)
W = FunctionSpace(mesh, Qk)
.
```

   Compare the errors to that of triangular meshes considering discretizations with the same number of nodes.

4. Consider the following variational problem (orthogonal projection): "Find $p_h \in V_h$ such that

$$a(p_h - u, v_h) = 0 \quad \forall v_h \in V_h\text{"}$$

where

$$a(u,v) = \int_\Omega (u\,v + \alpha\,\nabla u \cdot \nabla v)\,d\mathbf{x}$$

$\alpha = \{0,1\}$. Note that we are taking $v_h \in V_h$ (not V_{h0}). Solve the projection problem in Fenics and compute the error norms $\|u - p_h\|$. It can be implemented by hand or using the built-in function `project()`, i.e.,

```
.
uproj = project(uexact, W)
.
```

5. Solve the problem in a circular region of radius $R = 1$, with $\kappa = 1$, $f = 1$ and boundary condition $u = 0$, for which the exact solution is

$$u = \frac{1 - x_1^2 - x_2^2}{4}$$

To define the boundary condition write `Constant(0)` as the second argument of `DirichletBC()`, i.e.,

```
.
bcall  = DirichletBC(W, Constant(0), allb)
.
```

Also, test writting `uexact` as the second argument. Discuss the results.

6. **BONUS:** Noticing the exact solution has zero mean and satifies homogeneous Neumann conditions on $\partial\Omega$, i.e.,

$$\begin{aligned} \int_\Omega u\,d\mathbf{x} &= 0, \\ \nabla u \cdot \check{\mathbf{n}} &= 0, \quad \mathbf{x} \in \partial\Omega \end{aligned}$$

We may solve instead the following variational problem: "Find $(u_h, \lambda) \in V_h \times \mathbb{R}$ such that

$$\begin{aligned}
\int_\Omega \kappa \nabla u_h \cdot \nabla v_h \, d\mathbf{x} \; + \; \lambda \int_\Omega v_h \, d\mathbf{x} &= \int_\Omega f \, v_h \, d\mathbf{x} \\
r \int_\Omega u_h \, d\mathbf{x} &= 0
\end{aligned}$$

forall $(v_h, r) \in V_h \times \mathbb{R}$". Note in this case there are no restrictions on V_h and the Neumann condition appears naturally in the variational formulation. Implement in Fenics and perform a mesh refinement study on triangular meshes.