

ASSIGNMENT 1

THE FINITE ELEMENT METHOD IN COMPUTATIONAL MECHANICS

Ivar Haugaløkken Stangeby

March 6, 2017

Exercise 1

In this assignment we start off by considering the following boundary value problem.

Boundary Value Problem 1. On the two dimensional rectangular domain $\Omega := (0, 1)^2$, consider the problem:

$$-\nabla u = f \text{ in } \Omega, \quad (1)$$

$$u = 0 \text{ for } x = 0 \text{ and } x = 1, \quad (2)$$

$$\frac{\partial u}{\partial n} = 0 \text{ for } y = 0 \text{ and } y = 1. \quad (3)$$

Analytical gobbledygook

We start by assuming $u = \sin(\pi kx) \cos(\pi ky)$ and compute the source term $f = -\Delta u = 2\pi^2 k^2 u$. We wish to compute analytically, the H^p norm. Recall that the H^p norm $\|\cdot\|_p$ is defined by

$$\|u\|_p = \left(\sum_{|\alpha| \leq p} \int_{\Omega} \left(\frac{\partial^{|\alpha|} u}{\partial \mathbf{x}^\alpha} \right)^2 d\mathbf{x} \right)^{1/2}$$

where $\alpha := (\alpha_1, \dots, \alpha_d)$ is a multi-index, and $|\alpha| := \alpha_1 + \dots + \alpha_d$. In the case where Ω is a subset of \mathbb{R}^2 , we have $\alpha = (i, j)$ and $|\alpha| = i + j$. Note that

the terms in the sum occur as the L^2 norm squared of the mixed partial derivatives, for instance:

$$\left\| \frac{\partial^{i+j} u}{\partial x^i \partial y^j} \right\|_{L^2}^2 = (k\pi)^{2(i+j)} \int_0^1 \int_0^1 \cos^2(k\pi x) \sin^2(k\pi y) \, dx dy.$$

Using the fact that both $\sin^2(\pi ky)$ and $\cos^2(\pi kx)$ integrate to $1/2$ over the unit interval, we have that this equals $(k\pi)^{2(i+j)}/4$. For $|\alpha| = n$, we have $n + 1$ partial derivatives of order n , hence $\|u\|_p$ can be computed as

$$\|u\|_p = \frac{1}{2} \left(\sum_{|\alpha| \leq p} (k\pi)^{2|\alpha|} \right)^{1/2}.$$

Numerical error estimates

We solve the system given in Boundary Value Problem 1 in the PYTHON-framework **FeniCS**. Our mesh is taken to be uniformly spaced with mesh size $h := 1/N$. We examine the error in both the L_2 and the H^1 norms for $k = 1, 10$ and for both first and second order Lagrangian elements, that is

$$\text{error} = \|u - u_h\|_q \quad \text{for } q = 0, 1.$$

Computed by the function **exercise_1_b()**, the numerical errors are listed in Table 1.

We now wish to verify the two following error estimates:

$$\|u - u_h\|_1 \leq C_\alpha h^\alpha, \quad (4)$$

and

$$\|u - u_h\|_0 \leq C_\beta h^\beta. \quad (5)$$

These error estimates can be rewritten as linear equations in h with slopes α, β and constant terms $\log(C_\alpha), \log(C_\beta)$, respectively. That is

$$\log(\|u - u_h\|_1) \leq \alpha h + \log(C_\alpha),$$

and similarly for the $\|\cdot\|_0$ error. Sampling the left hand side for several values of h , we can fit a linear function to the data, hence finding the unknown slope and constant terms. This has been done using the function **numpy.polyfit()**, and the full implementation can be seen in **estimate_error()**. The results are given in Table 2.

Table 1: The L_2 and H^1 errors for varying mesh size, using both first order and second order Lagrangian elements.

(a) L_2 error to the left, H^1 error to the right. Using first order elements.

N	k = 1	k = 10	N	k = 1	k = 10
8	0.032766	0.677496	8	0.436116	25.511516
16	0.008462	0.363384	16	0.218105	17.233579
32	0.002133	0.177866	32	0.109047	10.543850
64	0.000534	0.054880	64	0.054523	5.430879

(b) L_2 error to the left, H^1 error to the right. Using second order elements.

N	k = 1	k = 10	N	k = 1	k = 10
8	0.000569	0.424446	8	0.033141	17.666883
16	0.000069	0.088649	16	0.008387	6.717151
32	0.000009	0.010174	32	0.002105	1.961954
64	0.000001	0.001139	64	0.000527	0.517347

Table 2: The slopes and coefficients for the error estimates given in Equation (4), for both first and second order elements.

(a) First order elements.

k	α	C_α	β	C_β
1	1.980242	2.029814	1.991671	2.090703
10	1.190830	9.085196	1.705687	677.310236

(b) Second order elements.

k	α	C_α	β	C_β
1	1.980242	2.029814	0.999942	3.488758
10	1.190830	9.085196	0.740449	126.848072

Exercise 2

We now consider another boundary value problem:

Boundary Value Problem 2. On the two dimensional rectangular domain $\Omega := (0, 1)^2$, consider the second order problem:

$$-\mu\Delta u + u_x = 0 \text{ in } \Omega, \quad (6)$$

$$u = 0 \text{ for } x = 0, \quad (7)$$

$$u = 1 \text{ for } x = 1, \quad (8)$$

$$\frac{\partial u}{\partial n} = 0 \text{ for } y = 0 \text{ and } y = 1. \quad (9)$$

Analytical solution

It is possible to derive an analytical solution for the above boundary value problem using separation of variables. We make the ansatz that we can write $u(x, y) = f(x)g(y)$. Plugging this into Boundary Value Problem 2, and dividing by $-\mu u$ we arrive at the set of equations

$$f''(x) - \frac{1}{\mu}f'(x) - Cf(x) = 0, \quad (10)$$

$$g''(y) + Cg(y) = 0, \quad (11)$$

where C is some unknown constant. Solving for $g(y)$ first, we arrive at the solution

$$g(y) = A \sin(\sqrt{C}y) + B \cos(\sqrt{C}y).$$

Enforcing the Neumann boundary conditions given in Equation (9), we determine g to be constant (with respect to x) equal to

$$g(y) = B \cos(n\pi y),$$

with $n \in \mathbb{N}$. In particular, for $n = 0$, $C = 0$ so we have $g(y) = B$. Furthermore, with this choice of n , Equation (10) reduces to

$$f''(x) - \frac{1}{\mu}f'(x) = 0$$

which has solution $f(x) = D \exp(\frac{1}{\mu}x) + E$. Enforcing the Dirichlet boundary conditions given in Equations (7) and (8) we determine $E = -1$ and $D = (e^{\frac{1}{\mu}} - 1)^{-1}$, as well as $B = 1$, yielding the final solution

$$u(x, y) = f(x)g(y) = \frac{e^{\frac{1}{\mu}x} - 1}{e^{\frac{1}{\mu}} - 1}.$$

Streamline diffusion / Petrov-Galerkin.

The parameter μ causes problems for low values of μ , as later can be seen in the numerical errors. This can be remedied by introducing the test function

$$w = v + \beta \nabla v.$$

Setting $\beta = h/2$, where h is the size of the discretized mesh, corresponds to the upwinding scheme.

Numerical error estimates

Again, we look at the numerical errors, both L_2 and H^1 using both first and second order Lagrange elements. We first examine what happens with the regular weak formulation, and then we see how the SUPG-method may improve our results.

Without SUPG: Calling `exercise.t.b()` with the `SUPG`-flag set to false, we achieve the errors listed in Table 3. In all cases we see that the error convergence is slower for both norms for lower values of μ . Similarly to what was done in the previous boundary value problem, we estimate the values for C_α , C_β , α and β , using `estimate_error()`. The results can be seen in Table 4.

With SUPG: Setting the SUPG-flag to true, we achieve the values listed in Table 5 and Table 6. As we see from the values, the SUPG-method is superior for low values of μ and coarse meshes.

Table 3: The L_2 and H^1 errors of the new boundary value problem, using both first and second order elements for varying values of μ .

(a) L_2 error using first order elements.

N	$\mu = 1.00$	$\mu = 0.10$	$\mu = 0.01$
8	0.001402	0.023747	0.238965
16	0.000351	0.006177	0.103990
32	0.000088	0.001561	0.038142
64	0.000022	0.000391	0.011255

(b) H^1 error using first order elements.

N	$\mu = 1.00$	$\mu = 0.10$	$\mu = 0.01$
8	0.037522	0.769237	7.796998
16	0.018766	0.398389	7.008644
32	0.009383	0.201077	5.086480
64	0.004692	0.100781	2.982329

(c) L_2 error using second order elements.

N	$\mu = 1.00$	$\mu = 0.10$	$\mu = 0.01$
8	0.000012	0.002248	0.086719
16	0.000001	0.000304	0.030833
32	0.000000	0.000039	0.007649
64	0.000000	0.000005	0.001329

(d) H^1 error using second order elements.

N	$\mu = 1.00$	$\mu = 0.10$	$\mu = 0.01$
8	0.000597	0.118721	5.633591
16	0.000150	0.031667	3.801062
32	0.000038	0.008068	1.736641
64	0.000009	0.002028	0.569069

Table 4: Error estimates for both first and second order Lagrangian elements, for varying values of μ .

(a) First order elements.

μ	α	C_α	β	C_β
1.00	1.999763	0.089724	0.999856	0.300108
0.10	1.975224	1.458317	0.978303	5.936369
0.01	1.467166	5.552503	0.462191	22.684702

(b) Second order elements.

μ	α	C_α	β	C_β
1.00	2.994029	0.005828	1.994016	0.037794
0.10	2.950373	1.059090	1.958612	7.086985
0.01	2.009556	6.772810	1.105224	67.385481

Table 5: The L_2 and H^1 errors of the new boundary value problem, using both first and second order elements for varying values of μ , using the SUPG-method.

(a) L_2 error using first order elements.

N	$\mu = 1.00$	$\mu = 0.10$	$\mu = 0.01$
8	0.008173	0.116845	0.200561
16	0.003970	0.063322	0.131785
32	0.001956	0.033130	0.079917
64	0.000971	0.016982	0.045471

(b) H^1 error using first order elements.

N	$\mu = 1.00$	$\mu = 0.10$	$\mu = 0.01$
8	0.044519	1.008406	5.434785
16	0.022577	0.622238	5.792799
32	0.011370	0.351801	4.973273
64	0.005705	0.188202	3.626298

(c) L_2 error using second order elements.

N	$\mu = 1.00$	$\mu = 0.10$	$\mu = 0.01$
8	0.008173	0.116845	0.200561
16	0.003970	0.063322	0.131785
32	0.001956	0.033130	0.079917
64	0.000971	0.016982	0.045471

(d) H^1 error using second order elements.

N	$\mu = 1.00$	$\mu = 0.10$	$\mu = 0.01$
8	0.044519	1.008406	5.434785
16	0.022577	0.622238	5.792799
32	0.011370	0.351801	4.973273
64	0.005705	0.188202	3.626298

Table 6: Error estimates for both first and second order Lagrangian elements, for varying values of μ , using the SUPG-method.

(a) First order elements.

μ	α	C_α	β	C_β
1.00	1.024002	0.068333	0.988169	0.348477
0.10	0.928208	0.817010	0.808787	5.626513
0.01	0.714465	0.919312	0.197124	9.027664

(b) Second order elements.

μ	α	C_α	β	C_β
1.00	-5.295546	0.000071	-6.289647	0.000438
0.10	-7.960629	0.000000	-9.286106	0.000000
0.01	0.609334	0.719865	-0.245373	3.387804