NTNU

Kunnskap for en bedre verden

Department of Mathematics

TMA4220 - Numerical Solution of Partial Differential Equations Using Element Methods

Project 1

Author:
Anne Asklund
Jasper Steinberg
Paulius Cebatarauskas

29th April 2025

1 Gaussian quadrature

1.1 1D quadrature

For the code which implements `quadrature1D(a, b, Nq, g)` see the Jupyter Notebook. For verification, we have compared our results to the exact solution of

$$\int_1^2 e^x dx = e^2 - e. \tag{1}$$

The absolute error of our method for an increasing number of quadrature nodes ($N_q = 1, 2, 3, 4$) can be seen in Fig. 1.

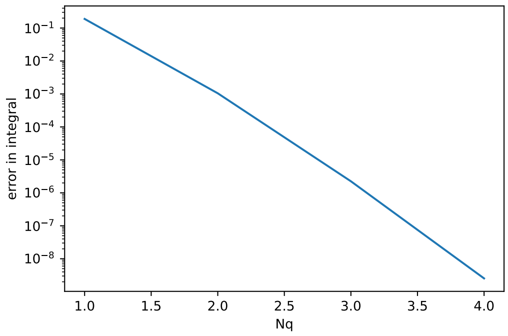

Figure 1: Absolute value of error in 1D quadrature when compared to the exact solution (1).

1.2 2D quadrature

The code which implements `quadrature2D(p1, p2, p3, Nq, g)` can also be seen in the Jupyter Notebook. For verification, we have compared our results to the exact solution of

$$\int_T \log(x+y)dxdy = -\frac{3}{2} - \frac{32\log(2)}{3} + \frac{25\log(5)}{4} \tag{2}$$

where T is the triangle with vertices $(1, 0)$, $(3, 1)$ and $(3, 2)$. The error of our method for an increasing number of quadrature nodes ($N_q = 1, 3, 4$) can be seen in Fig. 2.

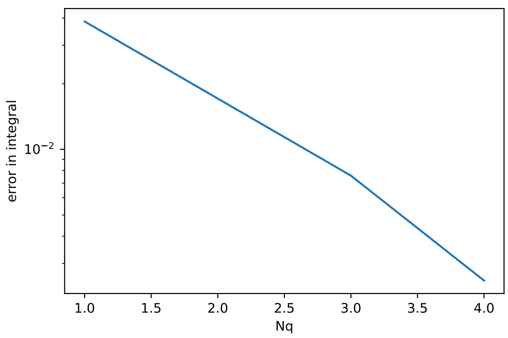

Figure 2: Absolute value of error in 2D quadrature when compared to the exact solution (2).

2 Poisson in 2 dimensions

We consider the Poisson problem in 2 dimensions

$$\begin{cases} \nabla^2 u(x,y) = -f(x,y), \quad (x,y) \in \Omega, \\ u(x,y) = 0, \quad (x,y) \in \partial\Omega \end{cases} \tag{3}$$

where $\Omega = \{(x,y) \in \mathbb{R}^2 \ : \ x^2 + y^2 \leq 1\}$ is the unit disc. We solve this problem with the following right-hand side

$$f(x,y) = -8\pi \cos(2\pi(x^2+y^2)) + 16\pi^2(x^2+y^2)\sin(2\pi(x^2+y^2)).$$

2.1 Analytical solution

We want to show that

$$u(x,y) = \sin\left(2\pi(x^2+y^2)\right) \tag{4}$$

is a solution to (3). We start by computing the Laplacian of u.

$$\begin{aligned} \frac{\partial u}{\partial x} &= \frac{\partial}{\partial x} \sin\left(2\pi(x^2+y^2)\right) = 4\pi x \cos\left(2\pi(x^2+y^2)\right) \\ \frac{\partial^2 u}{\partial x^2} &= 4\pi \frac{\partial}{\partial x} \left\{ x \cos\left(2\pi(x^2+y^2)\right) \right\} = 4\pi \left[\cos\left(2\pi(x^2+y^2)\right) - 4\pi x^2 \sin\left(2\pi(x^2+y^2)\right) \right] \end{aligned}$$

Now by symmetry the partial derivatives with respect to y are given by

$$\begin{aligned} \frac{\partial u}{\partial y} &= 4\pi y \cos\left(2\pi(x^2+y^2)\right) \\ \frac{\partial^2 u}{\partial y^2} &= 4\pi \left[\cos\left(2\pi(x^2+y^2)\right) - 4\pi y^2 \sin\left(2\pi(x^2+y^2)\right) \right]. \end{aligned}$$

Finally we see that

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 8\pi \cos\left(2\pi(x^2+y^2)\right) - 16\pi^2(x^2+y^2)\sin\left(2\pi(x^2+y^2)\right) = -f(x,y)$$

as desired. Now for the boundary condition, we get that $x^2 + y^2 = 1$ for $(x,y) \in \partial\Omega$. Thus

$$u(x,y) = \sin(2\pi(x^2+y^2)) = \sin(2\pi) = 0$$

for $(x,y) \in \partial\Omega$. Hence, u solves (3).

2.2 Weak formulation

We multiply the differential equation in (3) by $v \in H_0^1(\Omega)$, where $H_0^1(\Omega) = \{v \in H^1(\Omega) : v|_{\partial\Omega} = 0\}$, and integrate over Ω. This yields

$$\int_\Omega v \nabla^2 u \ d\Omega = -\int_\Omega f v \ d\Omega.$$

Now we use that $\nabla \cdot (v \nabla u) = v \nabla^2 u + \nabla u \cdot \nabla v$ to obtain,

$$\int_\Omega \nabla \cdot (v \nabla u) - \nabla u \cdot \nabla v \ d\Omega = -\int_\Omega f v \ d\Omega.$$

By the divergence theorem, we get that

$$\int_\Omega \nabla \cdot (v \nabla u) \ d\Omega = \int_{\partial\Omega} v \frac{\partial u}{\partial n} \ d\partial\Omega = 0,$$

where $\frac{\partial u}{\partial n}$ is the normal derivative of u and n is the outward unit normal. The last equality follows from the fact that $v|_{\partial\Omega} = 0$. We are thus left with

$$\int_\Omega \nabla u \cdot \nabla v \; d\Omega = \int_\Omega fv \; d\Omega.$$

This holds for all $v \in H_0^1(\Omega)$. This gives the weak formulation:

Find $u \in H_0^1(\Omega)$ such that for all $v \in H_0^1(\Omega)$,

$$a(u, v) = l(v),$$

where

$$\begin{aligned} a(u, v) &= \int_\Omega \nabla u \cdot \nabla v \; d\Omega, \\ l(v) &= \int_\Omega fv \; d\Omega. \end{aligned}$$

2.3 Galerkin projection

We are now restricting our search to functions in $X_h = \{v \in H_0^1 : v|_{K_k} \in \mathcal{P}_1(K_k), 1 \leq k \leq M\}$, where $\{K_k\}_{k=1}^M$ is the partition of Ω into triangles. We are given that $X_h = \text{span}\{\phi_i\}_{i=1}^n$. The Galerkin formulation then becomes "Find $u_h \in X_h$ such that $a(u_h, v_h) = l(v_h)$, $\forall v_h \in X_h$". It is natural to write $u_h = \sum_{i=1}^n u_h^i \phi_i$, $v_h = \sum_{j=1}^n v^j \phi_j$ in terms of the basis functions. Inserting this into the Galerkin formulation and reordering terms we get

$$\sum_{i,j} v^j a(\phi_i, \phi_j) u_h^i = \sum_j v^j l(\phi_j),$$

which we recognize as the matrix equation $v^T A u = v^T f$. This holds for all $v \in X_h$, so we choose $v_k^T = [0, \ldots, 0, 1, 0, \ldots, 0]$, which is 0 for all values except the k-th index for which it is 1, for $k = 1, \ldots, n$. This gives the equivalent formulation:
Find u such that

$$Au = f$$

where

$$\begin{aligned} &(A)_{ij} = a(\phi_i, \phi_j) \\ &u = [u_h^1, \ldots, u_h^n] \\ &(f)_i = l(\phi_i). \end{aligned}$$

2.4 Implementation

Figure 3 shows the triangulation on our domain Ω for number of triangulation points $N \in \{11, 21, 101\}$.

2.5 Stiffness matrix

The stiffness matrix A is implemented in the Jupyter notebook as the function `StiffnessMatrix(N)` where N is the number of nodes in the triangulation.

As observed in the Jupyter notebook, the rank of our stiffness matrix A appears to be one less than number of nodes in the triangulation. When one does not explicitly impose any boundary conditions in the implementation, by default homogeneous Neumann boundary conditions are imposed. Thus there is a degeneracy as our solution can be shifted by a constant and it is not uniquely defined. However, as only one point is required to "anchor" our solution, it makes sense that the rank is just diminished by one.

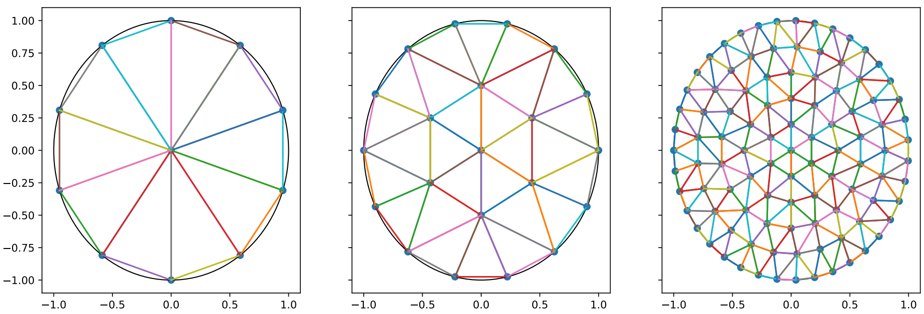

Figure 3: From left to right the figure shows the triangulation made from GetDisc with $N = \{11, 21, 101\}$ respectively.

2.6 Right hand side

The load vector f is implemented in the Jupyter notebook as the function `LoadVector(N, f)` where N is the number of nodes in the triangulation and f is the right hand side. The procedure is very similar to what we did for the stiffness matrix, the only difference is that we need quadrature to evaluate the resulting integrals.

2.7 Boundary conditions

Since the boundary points are last in our ordering, we can enforce the homogeneous Dirichlet boundary condition by slicing away the the last B columns and rows, where B is the number of boundary nodes.

2.8 Verification

Figures 4, 5, and 6 show the numerical solution, the exact solution and their difference respectively. As can been seen in the figures, the numerical solution is quite close to the exact solution.

Figures 7 and 8 show how the error changes for more nodes in the triangulation for the max-norm and the energy-norm respectively.

3 Neumann boundary conditions

We consider the Poisson problem in 2 dimensions with Neumann boundary conditions

$$\begin{cases} \nabla^2 u(x, y) = -f(x, y), \quad (x, y) \in \Omega, \\ u(x, y) = 0, \quad (x, y) \in \partial\Omega_D, \\ \frac{\partial u(x,y)}{\partial n} = g(x, y), \quad (x, y) \in \partial\Omega_N \end{cases} \tag{5}$$

where $\Omega = \{(x, y) \in \mathbb{R}^2 \ : \ x^2 + y^2 \leq 1\}$ is the unit disc, $\partial\Omega_D = \{(x, y) \in \mathbb{R}^2 \ : \ x^2 + y^2 = 1, \ y < 0\}$, and $\partial\Omega_N = \{(x, y) \in \mathbb{R}^2 \ : \ x^2 + y^2 = 1, \ y > 0\}$. We solve this problem with the following right-hand side

$$f(x, y) = -8\pi \cos(2\pi(x^2 + y^2)) + 16\pi^2(x^2 + y^2)\sin(2\pi(x^2 + y^2))$$

and for

$$g(x, y) = 4\pi\sqrt{x^2 + y^2}\cos(2\pi(x^2 + y^2)) + 16\pi^2(x^2 + y^2)\sin(2\pi(x^2 + y^2)).$$

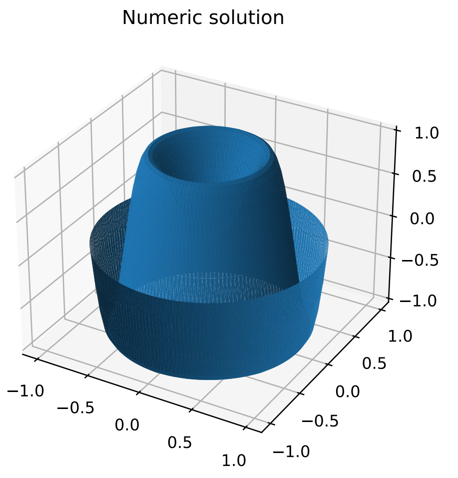

Figure 4: Numerical solution of (3).

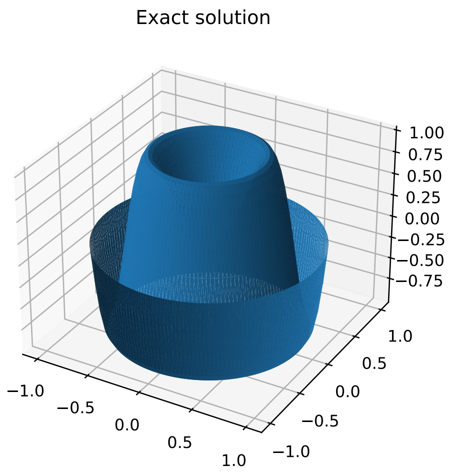

Figure 5: Exact solution of (3), i.e. (4).

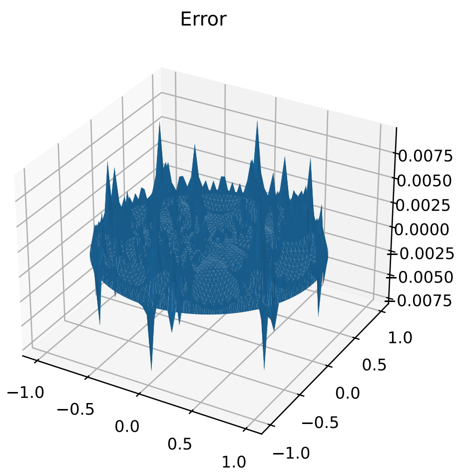

Figure 6: The difference between the exact solution and the numerical solution to (3).

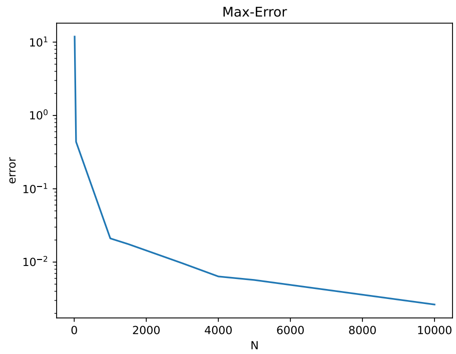

Figure 7: Error of numerical solution to (3) in max-norm.

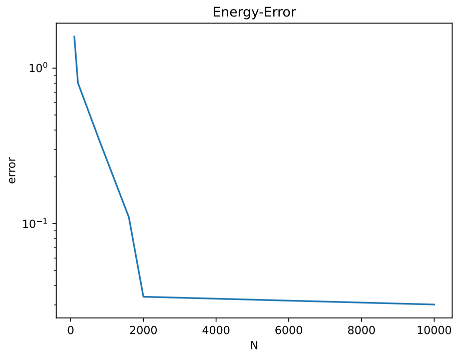

Figure 8: Error of numerical solution to (3) in energy-norm.

3.1 Boundary condition

Writing in polar coordinates where $x = \rho\cos(\theta)$ and $y = \rho\sin(\theta)$, the solution from 2.1 is $u(x, y) = u(\rho, \theta) = \sin(2\pi\rho^2)$. Moreover, for our domain

$$\frac{\partial u}{\partial n}\bigg|_{\partial\Omega_N} = \frac{1}{|\rho|}\frac{\partial u}{\partial \rho}\bigg|_{|\rho|=1} = 4\pi\rho\cos(2\pi\rho^2) = 4\pi\sqrt{x^2+y^2}\cos(2\pi(x^2+y^2)) = g(x, y)$$

as given, and thus verified.

3.2 Variational formulation

As we now do not only have homogeneous Dirichlet boundary conditions, we can no longer assume that $v \in H^1_0(\Omega)$. However, as we have homogeneous Dirichlet along the lower half-circle we can let $v \in \{v \in H^1(\Omega) : v|_{\partial\Omega_D} = 0\}$. Following the same procedure as in 2.2,

$$-\int_{\partial\Omega} v\nabla u \cdot n \; d\partial\Omega + \int_{\Omega} \nabla u \cdot \nabla v \; d\Omega = \int_{\Omega} fv \; d\Omega$$

and as $v|_{\partial\Omega_D} = 0$ we get that

$$-\int_{\partial\Omega} v\nabla u \cdot n \; d\partial\Omega = -\int_{\partial\Omega_N} v\nabla u \cdot n \; d\partial\Omega_N = -\int_{\partial\Omega_N} gv \; d\partial\Omega_N.$$

Since this integral is independent of u, we move it to the side of $l(\cdot)$ and our $a(\cdot,\cdot)$ remains the same as in section 2.2 while

$$l(v) = \int_{\Omega} fv \; d\Omega + \int_{\partial\Omega_N} gv \; d\partial\Omega_N.$$

3.3 Gauss quadrature

We use the parametrization of the line ℓ between two points (x_0, y_0) and (x_1, y_1) given by

$$x(t) = \frac{1}{2}\left((1-t)x_0 + (t+1)x_1\right)$$
$$y(t) = \frac{1}{2}\left((1-t)y_0 + (t+1)y_0\right),$$

with $t \in [-1, 1]$, to transform the line-integral into something simple:

$$\int_{\ell} f(x, y)\, dS = \int_{-1}^{1} f(x(t), y(t))\frac{1}{2}\sqrt{(x_1-x_0)^2 + (y_1-y_0)^2}\, dt.$$

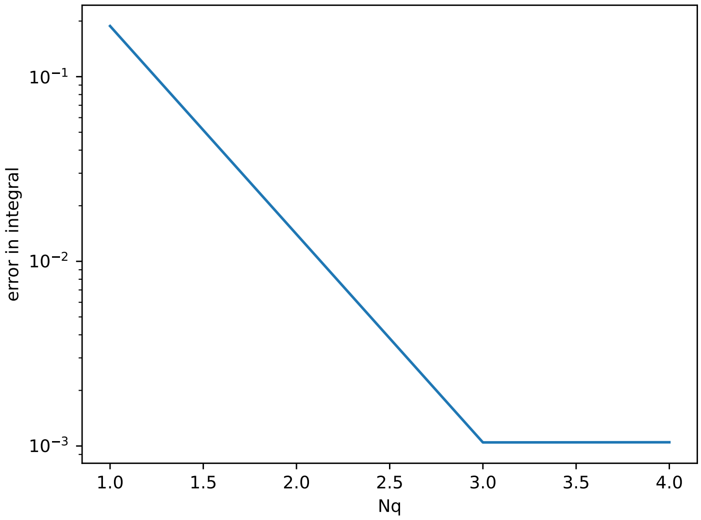

Figure 9: Absolute value of error in error, for $N_q = 1, 3, 4$.

Now the integral is solvable by the method used in 1.1, which we used use to implement the current task. For verification, we have compared our results to the exact solution of the line integral of e^x from $(1, 0)$ to $(2, 0)$. Figure 9 shows the absolute error of our method for an increasing number of quadrature nodes ($N_q = 1, 2, 3, 4$).

3.4 Implementation

Figures 10 and 11 show the numerical solution and difference between the numerical solution and the exact solution, respectively. When comparing our approximation to the exact solution, it is clear that we get the largest difference on the boundary where $y = 0$, as we change the boundary conditions from Neumann to Dirichlet. Moreover, under closer inspection, the Neumann boundary has a more general noise in the error than the Dirichlet boundary. In the interior the solutions are seemingly as good as for Dirichlet boundary conditions.

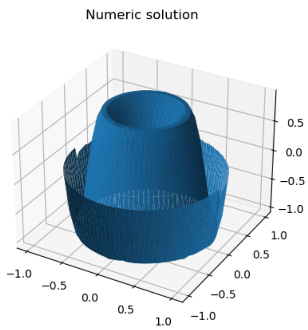

Figure 10: Numerical solution to (5).

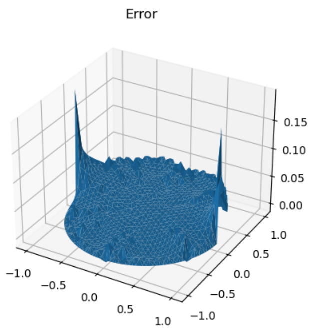

Figure 11: Difference between exact solution and numerical solution to (5).