NTNU Norwegian University of Science and Technology

Project 1

Course: TMA4220 - Finite Element Method

Authors: Muhammad Hamza Khalid, Sindre Grøstad

Student Number: 495026, 759123

October 28, 2018

1 Problem Formulation

The equation that is considered is the Helmoltz equation

$$u(\boldsymbol{x}) - \Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), \qquad \forall \boldsymbol{x} = (x, y) \in \Omega \tag{1}$$

with $\Omega = (0,1)^2$, $f(\boldsymbol{x}) = (2\pi + 1)\cos(\pi x)\sin(\pi y)$ and the boundary conditions

$$u(\boldsymbol{x}) = u_D \quad , \quad \boldsymbol{x} \in \Gamma_D = \{\boldsymbol{x} \in \partial\Omega : y \in \{0,1\}\} \tag{2}$$

$$\nabla u(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = 0 \quad , \quad \boldsymbol{x} \in \Gamma_N = \{\boldsymbol{x} \in \partial\Omega : x \in \{0,1\}\} \tag{3}$$

with $\boldsymbol{n}$ as the outward unit normal to the boundary.

1.1 Exact Solution Verification

The project description gives us the analytic solution $\tilde{u}(\boldsymbol{x}) = \cos(\pi x)\sin(\pi y)$ to the problem. This can be verified, first by inserting $\tilde{u}$ into (1):

$$\begin{aligned}
\tilde{u}(\boldsymbol{x}) - \Delta\tilde{u}(\boldsymbol{x}) &= \cos(\pi x)\sin(\pi y) - \Delta\cos(\pi x)\sin(\pi y) \\
&= \cos(\pi x)\sin(\pi y) + \pi^2\cos(\pi x)\sin(\pi y) \\
&= (1 + 2\pi^2)\cos(\pi x)\sin(\pi y) \\
&= f(\boldsymbol{x}).
\end{aligned}$$

$\tilde{u}$ also has to satisfy the boundary conditions in (2) and (3):

$$\tilde{u}(x, 0) = \tilde{u}(x, 1) = 0,$$

hence the Dirichlet boundary conditions in (2) are satisfied. For the Neumann boundary conditions we get

$$\begin{aligned}
\boldsymbol{n} &= \pm \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
\nabla\tilde{u}(\boldsymbol{x}) \cdot \boldsymbol{n} &= \begin{bmatrix} -\pi\sin(\pi x)\sin(\pi y) \\ \pi\cos(\pi x)\cos(\pi y) \end{bmatrix} \cdot \boldsymbol{n} \\
&= \mp\pi\sin(\pi x)\sin(\pi y) \\
&= 0 \quad \text{ on } \Gamma_N
\end{aligned}$$

hence $\tilde{u}$ satisfies (1) and both boundary conditions (2) and (3).

1.2 Weak Formulation

To obtain the weak formulation of (1) we test the equation against any smooth function $v \in V$ where $V = \{v : v \in C^\infty(\Omega), v(\mathbf{x}) = 0 \text{ for } \mathbf{x} \in \Gamma_D, \nabla v(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0 \text{ for } \mathbf{x} \in \Gamma_N\}$. We multiply our problem with the test function $v \in V$ and integrate,

$$\int_\Omega u(\mathbf{x})\, v(\mathbf{x}) d\Omega \, - \int_\Omega \Delta u(\mathbf{x})\, v(\mathbf{x}) d\Omega \, = \int_\Omega f(\mathbf{x})\, v(\mathbf{x}) d\Omega.$$

Using Greens formula on the second term in the above expression, we get,

$$\int_\Omega u(\mathbf{x})\, v(\mathbf{x}) d\Omega \, - \left(\int_{\partial\Omega} \Big(\nabla u(\mathbf{x}).\, \mathbf{n}\Big)\, v(\mathbf{x}) ds - \int_\Omega \nabla u(\mathbf{x})\, \nabla v(\mathbf{x}) d\Omega \right) \, = \int_\Omega f(\mathbf{x})\, v(\mathbf{x}) d\Omega.$$

The integral on the boundary of square can be split into two parts,

$$\int_{\partial\Omega} \Big(\nabla u(\mathbf{x}).\, \mathbf{n}\Big)\, v(\mathbf{x}) ds = \int_{\Gamma_D} \Big(\nabla u(\mathbf{x}).\, \mathbf{n}\Big)\, \underbrace{v(\mathbf{x})}_{=0} ds + \int_{\Gamma_N} \Big(\underbrace{\nabla u(\mathbf{x}).\, \mathbf{n}}_{=0}\Big)\, v(\mathbf{x}) ds \, = \, 0.$$

Using this in the above formulation we are left with the following week formulation.
Find $u \in V$ such that for all $v \in V$,

$$\int_\Omega u(\mathbf{x})\, v(\mathbf{x}) d\Omega \, + \int_\Omega \nabla u(\mathbf{x})\, \nabla v(\mathbf{x}) d\Omega \, = \int_\Omega f(\mathbf{x})\, v(\mathbf{x}) d\Omega, \quad \Omega = (0,1)^2, \ \mathbf{x} = (x, y). \tag{4}$$

1.3 Uniqueness of Solution

The function $\tilde{u}$ is a unique solution. To prove this we show that it satisfies the Lax-Milgram Theorem that states:

The weak formulation is well posed and hence have an unique solution if a$(\cdot,\cdot)$ is a coercive continuous bilinear form on V×V and L$(\cdot)$ is a continuous linear form on V.

We start by observing that since both $f(\boldsymbol{x})$ and $v(\boldsymbol{x})$ are continuous on V, we know that L(v) is a continuous linear form on V. For a$(\cdot,\cdot)$ to be continuous we need that for any $u, v \in V$,

$$|a(u,v)| \leq M \parallel u \parallel_V \parallel v \parallel_V$$

for some real M > 0.

$$\begin{aligned} |a(u,v)| &= \Big| \int_\Omega uv d\Omega + \int_\Omega \nabla u \nabla v d\Omega \Big| \\ &= |\langle u, v \rangle_V| \\ &\leq \parallel u \parallel_V \parallel v \parallel_V \end{aligned}$$

where the Cauchy-Schwarz Theorem is used in the last step. Hence a$(\cdot,\cdot)$ is continuous with M $= 1$. For a$(\cdot,\cdot)$ to be coercive we need that for any $u \in V$,

$$a(u,u) \geq \alpha \parallel u \parallel_V^2$$

for some real $\alpha > 0$.

$$\begin{aligned} a(u,u) &= \int_\Omega u^2 d\Omega + \int_\Omega (\nabla u)^2 d\Omega \\ &= \parallel u \parallel_V^2 \end{aligned}$$

hence a$(\cdot,\cdot)$ is coercive with $\alpha = 1$ and a$(\cdot,\cdot)$ is a coercive continuous bilinear form. This proves that the weak formulation is well-posed and that $\tilde{u}$ is a unique solution.

2 Finite Element Space

2.1 Lagrange P1 on Reference Element

Linear Lagrange elements defined on the reference element are as given below,

$$\begin{aligned} \hat{\varphi}_1 &= 1 - \hat{x} - \hat{y}, \\ \hat{\varphi}_2 &= \hat{x}, \\ \hat{\varphi}_3 &= \hat{y}. \end{aligned}$$

2.2 Affine Mapping

In Figure 1, a test for mapping of reference element to a particular triangle has been plotted. The mapping is given by

$$\boldsymbol{x} = v_0 + J_{T_k}\hat{\boldsymbol{x}} = v_0 + [v_1 - v_0, v_2 - v_0]\hat{\boldsymbol{x}}$$

where v_0, v_1 and v_2 are column vectors representing the vertices of the cell, $\boldsymbol{x}$ is the mapped coordinate and $\hat{\boldsymbol{x}}$ is the coordinate in the reference cell. For the cell K_0 in figure 1 the determinant of J_{T_k} is positive. This is because the vertices is given anti-clockwise. If K_0 is defined as (1,0), (3,2), (3,1) the determinant of J_{T_k} will be positive.

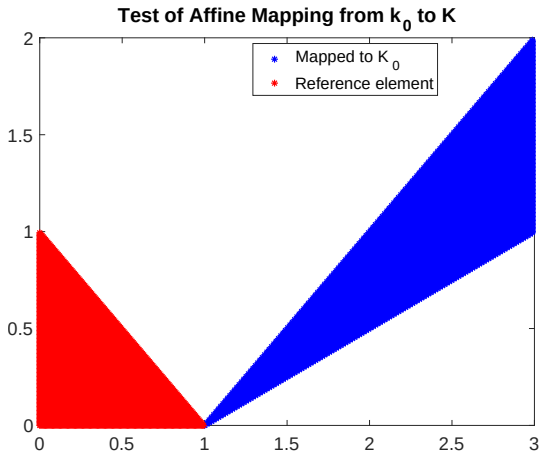


Figure 1: Test of affine mapping from the reference cell K = (0,0), (1,0), (0,1) to the cell K_0 = (1,0), (3,1), (3,2) and back to the reference cell again.

2.3 Galerkin Problem Formulation

Let $V_h \subseteq V$ be any finite dimensional space consisting of linear piece-wise functions. Let for $u, v \in V_h$ we can write v and u as a linear combination of it's basis i.e $v(\mathbf{x}) = \sum_{i=1}^{n} v_i(\mathbf{x})\varphi_i(\mathbf{x})$. Using similar expression for u and substituting in Equation 4 we get the following form

$$\int_\Omega \sum_{i,j=1}^{n} u_j\varphi_i(\mathbf{x})\, v_i\varphi_i(\mathbf{x})d\Omega + \int_\Omega \sum_{i,j=1}^{n} \nabla u_j\varphi_j(\mathbf{x})\, \nabla v_i\varphi_i(\mathbf{x})d\Omega = \int_\Omega \sum_{i=1}^{n} f(\mathbf{x})\, v_i\varphi_i(\mathbf{x})d\Omega,$$

Since we have continuous functions we can take the summations outside and write them in a vector form such that,

$$\begin{aligned} v^T A\, u &= v^T F, \quad \text{where } A{=}S{+}M, \\ \implies A\, u &= F \end{aligned}$$

The expressions for *S,M and F* are as given below,

$$S = \sum_K \int_{K_i} \nabla\varphi_j \nabla\varphi_i \, d\Omega, \quad M = \sum_K \int_{K_i} \varphi_j\varphi_i \, d\Omega, \quad F = \sum_K \int_{K_i} F_i\varphi_i \, d\Omega,$$

The approximation space for the Galerkin Problem V_h is hence the span of all Linear Lagrange elements on each cell.

3 Numerical Integration

To evaluate the integrals in the finite element method we used Gauss-Legendre quadratures. Firstly we implemented a method to integrate one-dimensional functions. To verify the correctness of the method, the approximation error to the integral $\int_1^2 \exp(x)dx$ was plotted as a function of N_q, the number of quadrature points, in figure 2.

Then we implemented a second method that used Gauss-Legendre quadratures to evaluate two-dimensional integrals on a triangle K. To verify the correctness of the method, the approximation error to the integral $\int_K \log(x+y)dxdy$ on the triangle with vertices K = (1,0), (3,1), (3,2) was plotted as a function of N_q in figure 3.

The choice of quadrature is important as we are approximating integrals and the better we approximate these integrals, the better result the finite element method will yield. When evaluating the left-hand side of Equation 4 we never have polynomials of higher degree than 2. As Gauss-Legendre quadratures gives an exact result for polynomials of degree 2n-1 when n is the degree of the method, there is no need to have any higher order than 2 in the approximation. For the right-hand side of Equation 4 the higher order on the approximation, the better, since f is not a polynomial.

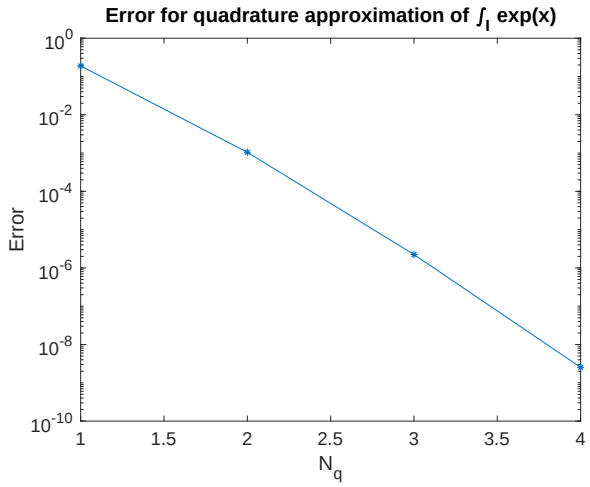


Figure 2: The approximation error of one-dimensional Gauss-Lengedre quadratures approximation for $\int_I \exp(x)dx$ on the interval I = (1,2).

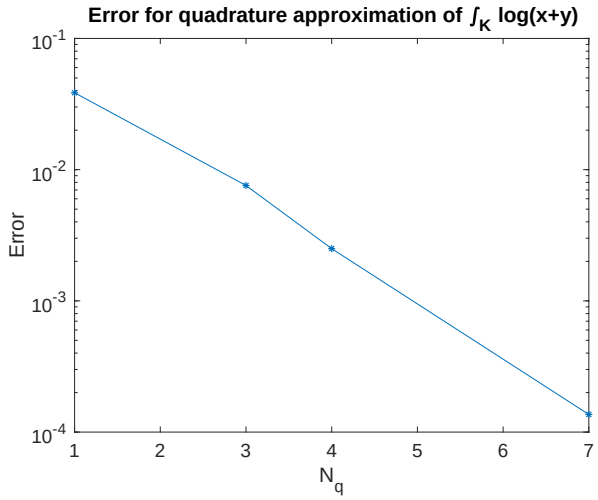


Figure 3: The approximation error of two-dimensional Gauss-Lengedre quadratures approximation for the integral $\int_K \log(x+y)dxdy$ on the triangle with vertices K = (1,0), (3,1), (3,2).

4 Numerical Implementation

4.1 Mesh Generation

Given square mesh, was distributed into smaller sub-squares and then the sub-squares were triangulated as shown in Figure 4. The assumptions used are as given below,

- Step length chosen for both axis is equal
- Vertices are named in the anti clockwise direction

4.1.1 Connectivity and Geometry Matrix

In the numerical implementation, *tri* stores row wise, the data of each triangle's vertices respective to their triangle and P is the geometry matrix that contains the data of points at vertices.

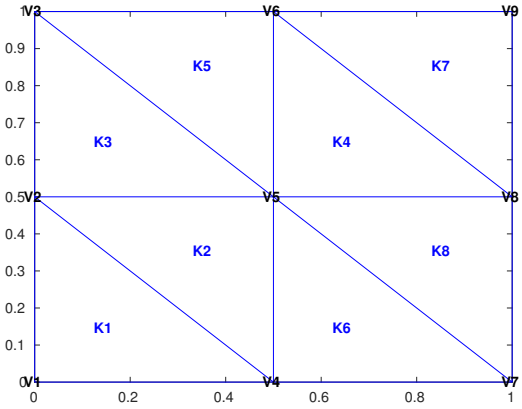


Figure 4: Triangulation of square mesh

4.2 Assembly

4.2.1 Stiffness Matrix

The stiffness matrix as formulated in the previous section is given as $S = \sum_K \int_{K_k} \nabla\varphi_i \nabla\varphi_j \, d\Omega$ For all $K \in \Omega$, the stiffness matrix was found by transporting it to reference element and solving it on the reference element. Using chain rule and substitution $x = x(\hat{x})$ and $\varphi = \hat{\varphi}(\hat{x})$ we get,

$$\begin{aligned}
\nabla\varphi_i \nabla\varphi_j &= \sum_K \frac{\partial\varphi_i}{\partial x_k} \frac{\partial\varphi_j}{\partial x_k}, \\
\nabla\varphi_i \nabla\varphi_j &= (\nabla_{\hat{x}}\hat{\varphi}\, J^{-1})\,(\nabla_{\hat{x}}\hat{\varphi}\, J^{-1}), \\
\nabla\varphi_i \nabla\varphi_j &= G\, G^T,
\end{aligned}$$

From this we observe that, $\nabla\varphi_i \nabla\varphi_j = G\, G^T$ is a constant on K_k, and area of the triangle is given by $det(J)/2!$. Using these results we can write an explicit expression for the stiffness Matrix.

$$S = \frac{G\, G^T\, |J|}{2}$$

After solving for each triangle, the local vertices were then transported to their exact positions on the global triangle.

4.2.2 Mass Matrix

The mass matrix was solved using Gaussian Quadrature order 4 locally on each triangle and then transported back to their respective position on the global triangle.

4.2.3 Load Vector

Load vector was also solved using Gaussian Quadrature order 4. Load vector as formulated in the previous section is written as,

$$F_j = \int_{K_k} f\, \varphi_j,$$

This was achieved locally on each triangle using,

$$F_j = \frac{|J|}{2} \omega_i\, \lambda_j^i\, f(x(\lambda^i)), \quad j = 1, 2, 3$$

where $\lambda^i = (\lambda_1^i, \lambda_2^i, \lambda_3^i)$ are quadrature nodes in barycentric coordinates and ω are respective quadrature weights. After evaluating load vector locally it was transported to it's respective global position in the mesh.

4.3 Boundary Conditions

Neumann boundary conditions don't play any role in solving the problem, however Homogeneous Dirichlet Boundary conditions were implemented on top and bottom of the problem. The first step was to find the vertices that are on the boundary. At boundary vertices number i, $A_{ii} = 1$ and $A_{ij} = 0$ for $i \neq j$. Likewise the load vector was set zero at the boundary vertices i.e $F_i = 0$.

4.4 Solution

After implementing the boundary conditions, the solution was found using backslash command of Matlab, and compared with the given exact solution.

4.5 Numerical Results

Figure 5 shows the comparison of the comparison of solutions of numerical and exact solutions.

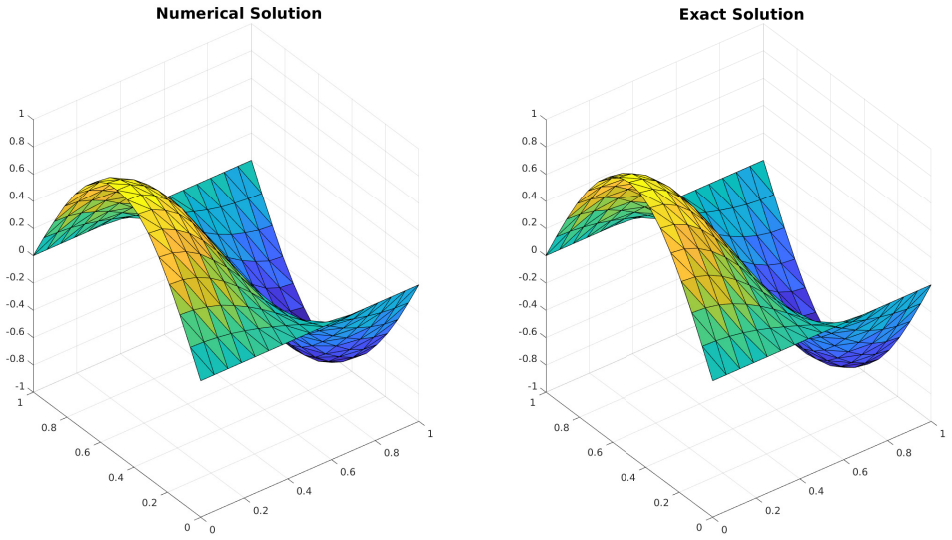


Figure 5: Numerical and Exact Solution

5 Covergence analysis

Let $e_h = U_{numerical} - U_{exact}$ on the interval $[0, 1]$ be a constant piece-wise function. Since we are approximating the function $U(x)$ which is solution of our equation,so we can think of error as a constant function rather than a vector. The error used to plot the error is,

$$e = \frac{||U_{numerical} - U_{exact}||}{M}$$

To evaluate the convergence of the method, we solved the problem for mesh sizes $h_\tau = 1/\mathrm{M}$ for M = 2,4 and 16 and the results can be seen in Figure 6.

6 Extension to an Evolution Problem

We now consider the evolution problem

$$\partial_t u(\boldsymbol{x}, t) - \nu \Delta u(\boldsymbol{x}, t) = f(\boldsymbol{x}, t), \qquad \forall (\boldsymbol{x}, t) \in \Omega \times (0, T) \tag{5}$$

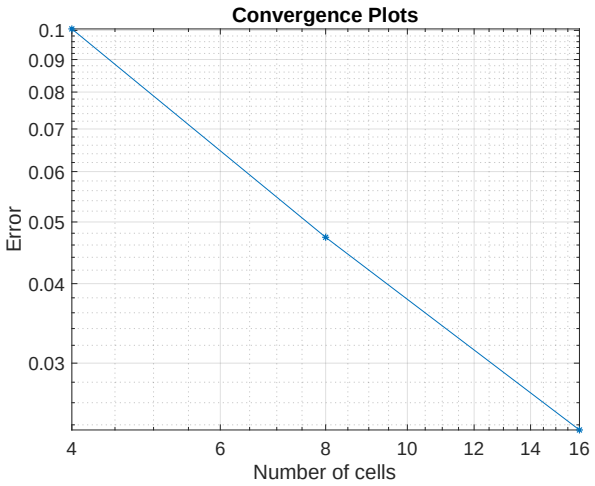


Figure 6: The approximation error at the vertices for different mesh sizes.

with initial condition $u(\boldsymbol{x}, 0) = u_0$ and ν as the diffusivity. To obtain the weak formulation of (5), we multiply the equation with $v \in V$ where

$$V = \{v : v \in C^{\infty}(\Omega, (0,T)), v(\mathbf{x},t) = 0 \text{ for } \mathbf{x} \in \Gamma_D, \nabla v(\mathbf{x},t) \cdot \mathbf{n}(\mathbf{x}) = 0 \text{ for } \mathbf{x} \in \Gamma_N \text{ and } u(\boldsymbol{x}, 0) = u_0\}$$

and integrate to obtain

$$\int_{\Omega} \partial_t uv - \nu \int_{\Omega} \Delta uv = \int_{\Omega} fv$$

Similar as in equation (4), integration by parts gives

$$\int_{\Omega} \partial_t uv + \nu \int_{\Omega} \nabla u \nabla v = \int_{\Omega} fv \tag{6}$$

As in the "Galerkin problem formulation section", let $V_h \subseteq V$ and write u as a linear combination of it's basis i.e. $u(\mathbf{x}, t) = \sum_{i=1}^{n} u_i(\mathbf{x}, t)\varphi_i(\mathbf{x})$. Substituting into equation (6) gives

$$\int_{\Omega} \sum_{i,j=1}^{n} \partial_t u_j \varphi_i(\mathbf{x})\, v_i \varphi_i(\mathbf{x}) d\Omega + \nu \int_{\Omega} \sum_{i,j=1}^{n} \nabla u_j \varphi_j(\mathbf{x})\, \nabla v_i \varphi_i(\mathbf{x}) d\Omega = \int_{\Omega} \sum_{i=1}^{n} f(\mathbf{x})\, v_i \varphi_i(\mathbf{x}) d\Omega,$$

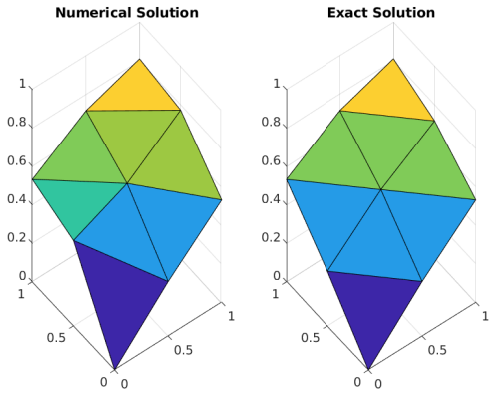
where u and v depends on t. since we have continuous functions we can take the summations outside and write the equations above in vector from such that,

$$\begin{aligned} v^T M \partial_t u + v^T \nu S u &= v^T F \\ \implies M \partial_t u &= -\nu S u + F. \end{aligned}$$

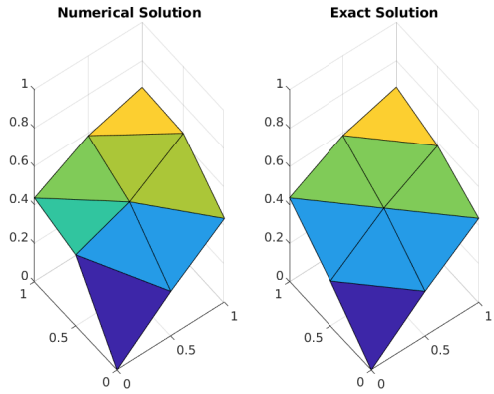
We end up with an ODE which can be solved by various methods, However two method were used to test the program, Forward Euler and Backward Euler. Results can be seen in Figure 7. By using backward Euler with $\frac{du}{dt} = \frac{u_n - u_{n-1}}{dt}$ we obtain,

$$\begin{aligned} M\left(\frac{u_n - u_{n-1}}{dt}\right) &= -\nu S u_n + F. \\ u_n &= u_{n-1} + dt\, M^{-1}(-\nu S u_n + F) \end{aligned}$$

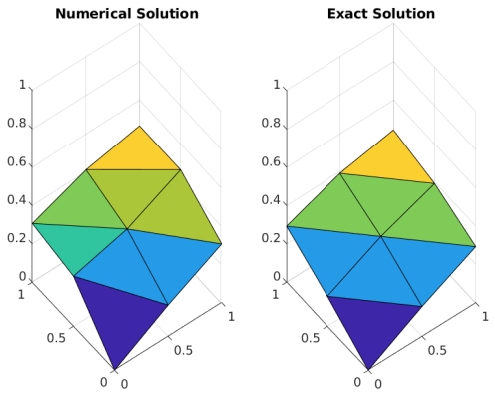
For our case we used $\nu = 1$ and $F = 0$. A noticeable point in the problem is the structure of M. For large stepsizes the condition number of Matrix was very large therefore the iterative scheme chosen should avoid taking inverse of M or some splitting methods for solving the problem can be considered. For our case, we considered a smaller mesh and using Backward Euler with Fixed Point iteration the obtained results can be seen in the Figure 7.



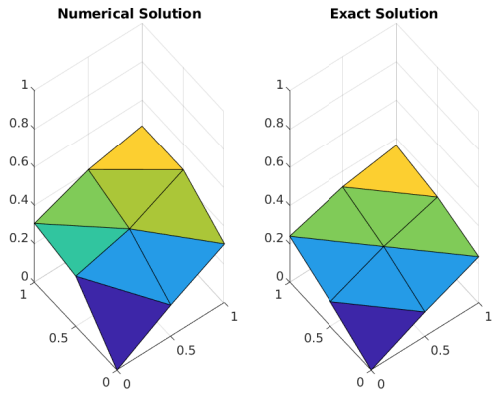
Time = 0.2



Time = 0.4



Time = 0.8



Time = 1.0

Figure 7: The Evolution Problem in equation (5) solved with finite element method and backward Euler for $\nu = 1$ and f = 0.