θ-Scheme of Finite Element Method for Heat Equation *

Wenqiang Feng †

Abstract

This is my MATH 574 course project report. In this report, I give some details for implementing the Finite Element Method (FEM) via Matlab and Python with FEniCs. This project mainly focuses on θ-Method for the initial boundary heat equation.

1 The Model Problem

Consider the initial boundary value problem for the heat equation:

$$\begin{cases} u_t - \Delta u & = f(x,t), \quad (x,t) \in \Omega \times (0,T] \\ u & = 0, \quad (x,t) \in \partial\Omega \times (0,T] \\ u(\cdot,0) & = u^0(x), \quad x \in \Omega, \end{cases} \tag{1}$$

where $\Omega = [0,1] \times [0,1]$. The initial data $u^0(x)$ and the function $f(x,t)$ are such that the solution is given by

$$u(x,y,t) = e^{-t}\sin(\pi x)\sin(\pi y).$$

2 A Full Discretization scheme: The θ-Method

$$\begin{cases} \left(\frac{u_h^{n+1}-u_h^n}{\tau}, v_h\right) + a\left(\theta u_h^{n+1} + (1-\theta)u_h^n, v_h\right) & = \left(\theta f(t^{n+1}) + (1-\theta)f(t^n), v_h\right) \\ u_h^0 & = \Pi u^0. \end{cases} \tag{2}$$

So, we get

$$\left(u_h^{n+1} - u_h^n, v_h\right) + \tau a\left(\theta u_h^{n+1} + (1-\theta)u_h^n, v_h\right) = \tau\left(\theta f(t^{n+1}) + (1-\theta)f(t^n), v_h\right).$$

Since

$$u_h^n = \sum_{j=1}^N \alpha_j^n \phi_j(x),$$

then

$$\begin{aligned} &\left(\sum_{j=1}^N \alpha_j^{n+1}\phi_j(x) - \sum_{j=1}^N \alpha_j^n \phi_j(x), \phi_i(x)\right) + \tau a\left(\theta \sum_{j=1}^N \alpha_j^{n+1}\phi_j(x) + (1-\theta)\sum_{j=1}^N \alpha_j^{n+1}\phi_j(x), \phi_i(x)\right) \\ = \quad & \tau\left(\theta f(t^{n+1}) + (1-\theta)f(t^n), \phi_i(x)\right), i = 1, 2, \cdots, N. \end{aligned}$$

---

*Key words: FEM, Pure Dirichlet boundary condition, Heat equation.

†Department of Mathematics,University of Tennessee, Knoxville, TN, 37909, wfeng@math.utk..edu

Therefore,

$$M\alpha^{n+1} - M\alpha^{n} + \theta\tau S\alpha^{n+1} + (1-\theta)\tau S\alpha^{n} = \theta\tau F^{n+1} + (1-\theta)\tau F^{n}$$

where

$$M = \sum_{j=1}^{N} \Big(\phi_j(x), \phi_i(x)\Big), M = \sum_{j=1}^{N} \Big(\nabla\phi_j(x), \nabla\phi_i(x)\Big), F_i^n = (f(t^n), \phi_i(x)).$$

Hence

$$(M + \theta\tau S)\,\alpha^{n+1} = (M - (1-\theta)\tau S)\,\alpha^{n} + \theta\tau F^{n+1} + (1-\theta)\tau F^{n}.$$

1. When $\theta = 0$, it is first-order explicit/forward Euler Scheme.
2. When $\theta = 1$, it is first-order implicit/backward Euler Scheme.
3. When $\theta = 1/2$, it is second-order Crank-Nicolson/ Trapezoidal Scheme.

3 Numerical Results

We use the the following test problem which has the exact solution

$$u(x,y,t) = e^{-t}\sin(\pi x)\sin(\pi y).$$

So,

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

and

$$f(x,y,t) = (2\pi^2 - 1)e^{-t}\sin(\pi x)\sin(\pi y).$$

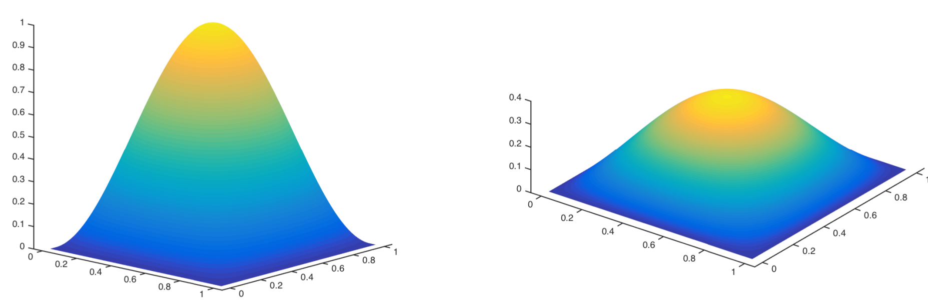

Figure 1: The initial data(left) and the exact solution at $T = 1$(right) .

Table 1: Errors of the computed solution (nx=ny=512)with implicit/backward Euler Scheme.

	$\#Tstep$	$\|u-u_h\|_{L^\infty}$	order
$q=1$	20	4.9128E-04	
	40	2.3917E-04	1.0271
	80	1.1481E-04	1.0416

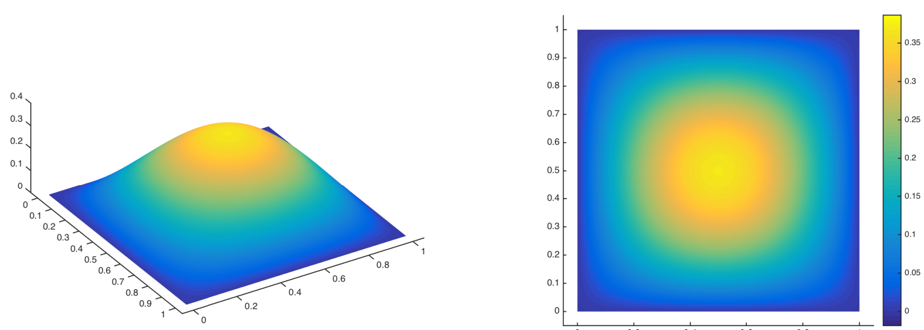

Figure 2: The numerical solution at $T=1$ with implicit/backward Euler Scheme.

Table 2: Errors of the computed solution ($nx = ny = 64$)with explicit/forward Euler Scheme.

	$\#Tstep$	$\|u-u_h\|_{L^\infty}$	order
$q=1$	20	8.6611E+67	divergence
	40	5.9872E+129	divergence
	80	1.0251E+242	divergence
	80	1.0251E+242	divergence
	160	4.5052E-17	convergence