MATH 610

Programming Assignment #7

November 29, 2018

Robert Turner

MATH 610

Programming Assignment #7

Euler method for finite element approximations for parabolic problems

1 Specifications

Consider the following elliptic boundary value problem: find $u(x, y, t)$ such that

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + \gamma u = f(x, y, t) \quad (x, y) \in \Omega, t > 0, \tag{1}$$

$$\nabla u \cdot \mathbf{n} = g(x, y, t) \quad (x, y) \in \Gamma, t > 0, \tag{2}$$

$$u(x, y, 0) = u_0(x, y) \quad \in \Omega, \tag{3}$$

where Γ is the boundary of Ω and $\mathbf{n}$ is the outer unit normal vector to Γ.

Solve the given below problems by approximating the corresponding boundary value problem using triangular finite elements on a partition of the domain generated by TRIANGLE and **explicit and implicit** Euler methods in time with uniform time step k. Consider the meshes with $|\tau| \leq 1/n^2, n = 10, 20, 40,$ where $|\tau|$ is the maximal area of the triangular elements. Also, in time t use meshes with approximately 20, 40, and 80 mesh points for implicit Euler. For explicit Euler, use time-steps that give stability and also large enough time steps for which the scheme is not stable (this is an illustration of what happens when the method is unstable). Submit a report according to the given below specification. Use your program from the previous programming assignment.

1.1 Problem 1

Take $\Omega = (0, 1) \times (0, 1), 0 < t < 3, \gamma = 5, g = 0$, and $f(x, y, t)$ is such that the exact solution is $u(x, y, t) = te^{-t}\cos 3\pi x \cos \pi y$ and $u_0(x, y) = 0$. Present the L^2 and H^1 norms of the error $u(x, y, t_n) - u_h^n(x, y)$ in a table for $t = 1$ and $t = 3$.

1.2 Problem 2

Solve the problem with $\gamma = 1, g = 1, f(x, y) = 1$, where this time Ω is a polygonal domain with vertices $(0, 0), (0.5, 0), (1, 1)$, and $(0, 2)$. Also, $u_0(x, y) = 0$. Plot the solution for $t = 1$ and $t = 3$.

1.3 Problem 3

The domain $\Omega = (0, 1)^2 \neq \Omega_1$, where $\Omega_1 = \{|x - 0.5| < 0.25\}, \{|y - 0.5| < 0.25\}$. Take $\gamma = 1, g = 1, f(x, y) = xy$, and $u_0(x, y) = 0$. Plot the solution for $t = 1$ and $t = 3$.

2 Preliminaries

In order to solve this problem, we multiply by a test function v and integrate over the spatial domain to arrive at

$$(\dot{u}_h, v) + a(u_h(t), v) = (f(t), v).$$

This is the semidiscrete Galerkin method, but we wish to discretize the time-dependence of u_h as well. We can either use the backward/implicit Euler method or the forward/explicit Euler method. If we use the implicit method, our problem becomes the iterative method

$$\begin{aligned} D\frac{U^{n+1}-U^n}{k} + AU^{n+1} &= F^{n+1}, \\ U^0 &= u_0(x, y), \end{aligned}$$

with D being the global lumped mass matrix, U^n the solution at the discrete time t_n, A the sum of the global consistent mass matrix and the global stiffness matrix, k the time step, and F^n the global forcing vector at the discrete time t_n. The A matrix is computed identically to the development presented in Programming Assignment 5, so we will not reiterate the theory here. The F vector is also computed in the same fashion as before, only in Problem 1, the forcing function is time dependent. By substituting the exact solution for Problem 1 into our partial differential equation, we arrive at

$$f(x, y, t) = \left(e^{-t}\right)\cos(3\pi x)\cos(\pi y)\left[(1-t) + t\left(10\pi^2 + \gamma\right)\right].$$

It should also be noted that the elemental lumped mass matrix D_e is calculated as

$$D_e = \frac{\tau}{3}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where τ is the area of the triangular element. This backward or implicit Euler scheme is unconditionally stable, so large time steps k can be implemented without loss of stability. This scheme is also first-order accurate.

The forward or explicit iterative Euler method applied to this problem is

$$\begin{aligned} D\frac{U^{n+1}-U^n}{k} + AU^n &= F^n, \\ U^0 &= u_0(x, y). \end{aligned}$$

This can also be written as

$$U^{n+1} = U^n - kD^{-1}AU^n + kD^{-1}F^n.$$

While this scheme is also first order accurate, it is not unconditionally stable. To ensure stability, we must find k such that

$$k \leq \frac{2}{\mathbf{\Lambda}(D^{-1}A)},$$

where $\mathbf{\Lambda}$ is an operator that returns the greatest eigenvalue of the matrix $D^{-1}A$.

Finally, for the norms we have

$$||e||_{L^2} = \sqrt{\int_\Omega \left(u_h^n(x,y) - u(x,y,t_n)\right)^2 d\Omega},$$

where $u_h^n(x,y)$ is the Galerkin approximation to the solution at time t_n and $u(x,y,t_n)$ is the exact solution to the problem at time t_n. For the H^1 norm of the error, we have

$$||e||_{H^1} = \sqrt{\int_\Omega \left(\nabla u_h^n(x,y) - \nabla u(x,y,t_n)\right)^2 + \left(u_h^n(x,y) - u(x,y,t_n)\right)^2 d\Omega},$$

3 Problem 1 Results

3.1 Implicit

The error norms for each spatial and temporal discretization are shown in Table 1. It should be noted that, for any given time on the interior of the time domain, the error norm suffers from any inaccuracy associated with linear interpolation. This is due to the possibility of the reported time value missing from the set of discrete points. If this is the case, the solution at the reported discrete time value (ie $t = 0.002$ or $t = 1$) is linearly interpolated from the solution at the nearest lesser adjacent discrete time value. This may explain the decrease in the L^2 error norm with increasing temporal refinement at the interpolated $t = 1$ value. We see convergence of the error as a function of spatial refinement, which is as expected. We also see what seems like little dependence on the temporal refinement for the norms at time $t = 3$; larger differences might only be seen at much smaller time steps. The additional term in the H^1 norm "encompasses" whatever change might be present from one discretization to another; this might be why even less change is seen in the H^1 norms.

Table 1: Problem 1 error norms using the implicit Euler scheme for each time and space discretization.

Elements	k	N_{time}	t = 1		t = 3	
			L^2	H^1	L^2	H^1
154	1.579E-01	20	3.604E-02	7.367E-01	1.442E-02	2.991E-01
	7.692E-02	40	3.568E-02	7.368E-01	1.442E-02	2.991E-01
	3.797E-02	80	3.567E-02	7.368E-01	1.443E-02	2.991E-01
596	1.579E-01	20	1.333E-02	4.096E-01	5.234E-03	1.663E-01
	7.692E-02	40	1.296E-02	4.096E-01	5.239E-03	1.663E-01
	3.797E-02	80	1.296E-02	4.096E-01	5.242E-03	1.663E-01
2452	1.579E-01	20	2.402E-03	1.670E-01	7.390E-04	6.775E-02
	7.692E-02	40	1.919E-03	1.669E-01	7.464E-04	6.775E-02
	3.797E-02	80	1.912E-03	1.669E-01	7.501E-04	6.775E-02

3.2 Explicit

In Table 2, we use much smaller discrete time values, as the stability condition enforces the use of miniscule time steps. As a result, computation time increases drastically, as seen in the 1.1 hour run time for the finest spatial discretization. In the stable explicit case, we see greater norms at $t = 0.006$, as the forcing function, which drives the error norm along with the initial data, increases from $t = 0$ to $t = 1$.

Table 2: Error norms using the explicit Euler scheme with a time step that ensures stability for each spatial discretization.

Elements	k	Wall Clock Time (s)	N_{time}	t = 0.002		t = 0.006	
				L^2	H^1	L^2	H^1
154	8.571E-04	4	8	2.372E-04	4.544E-03	6.899E-04	1.254E-02
596	1.253E-05	703	480	7.328E-05	2.379E-03	2.172E-04	6.745E-03
2452	5.190E-06	3967	1157	1.419E-05	9.276E-04	3.976E-05	2.715E-03

Finally, we use an unstable time step size in Table 3 that results in quick computation times. The instability can be seen in the garbage norm values, as the values increase drastically from time $t = 1$ to time $t = 3$.

Table 3: Error norms using the explicit Euler scheme with a time step that results in instability for each spatial discretization.

Elements	k	Wall Clock Time (s)	N_{time}	t = 1		t = 3	
				L^2	H^1	L^2	H^1
154	0.3	3	10	2.206E+03	1.638E+05	3.495E+20	2.635E+22
596	0.3	11	10	1.201E+07	7.260E+09	2.405E+35	1.454E+38
2452	0.3	46	10	2.097E+06	1.835E+09	5.989E+36	5.503E+39

4 Problem 2 Results

4.1 Implicit

For the plots on the implicit solves, we see negligible change in the solution with increasing temporal discretization. This may mean only 20 time points are needed to maximize solution accuracy. From time $t = 1$ to time $t = 3$, the only difference we see is an increase in amplitude; this makes sense, as the solution must "develop" with each time iteration.

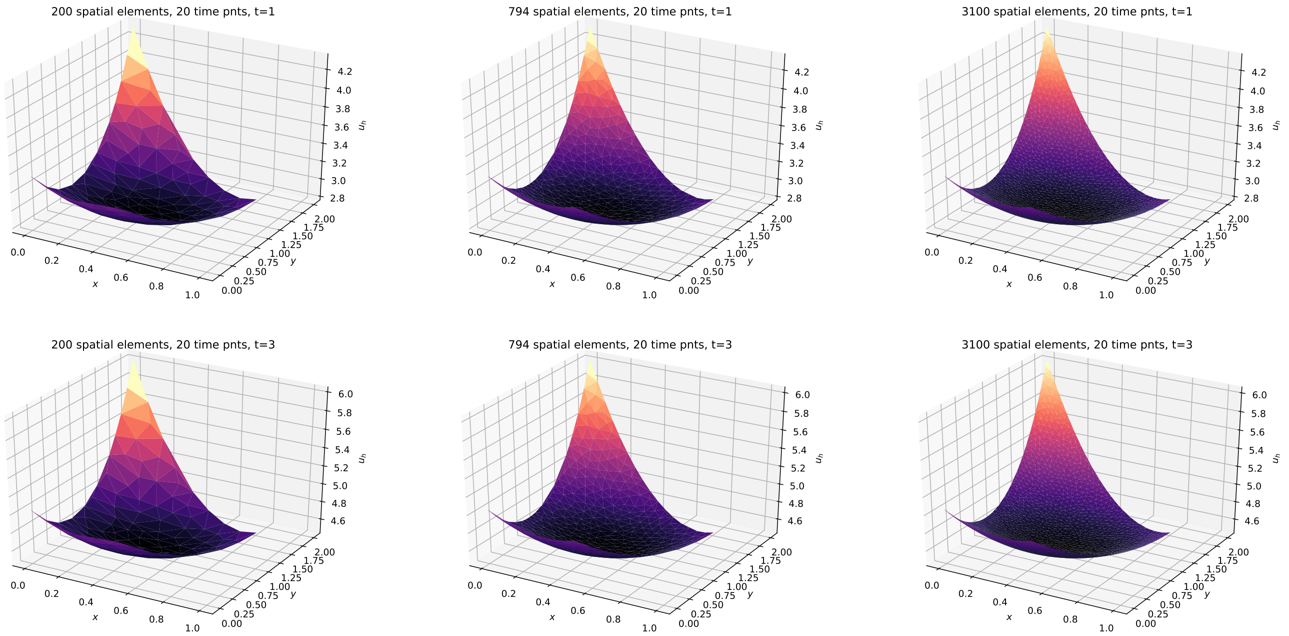

Figure 1: Solutions at times $t = 1$ (first row) and $t = 3$ (second row) with the implicit scheme with 20 time points for each spatial discretization.

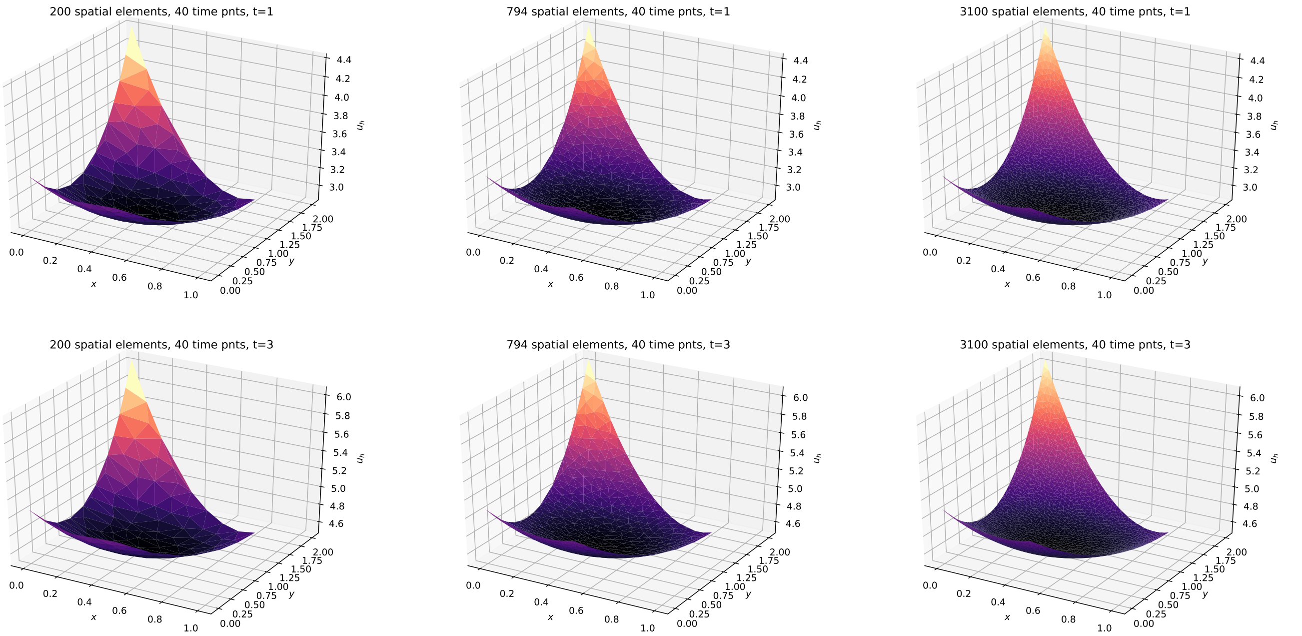

Figure 2: Solutions at times $t = 1$ (first row) and $t = 3$ (second row) with the implicit scheme with 40 time points for each spatial discretization.

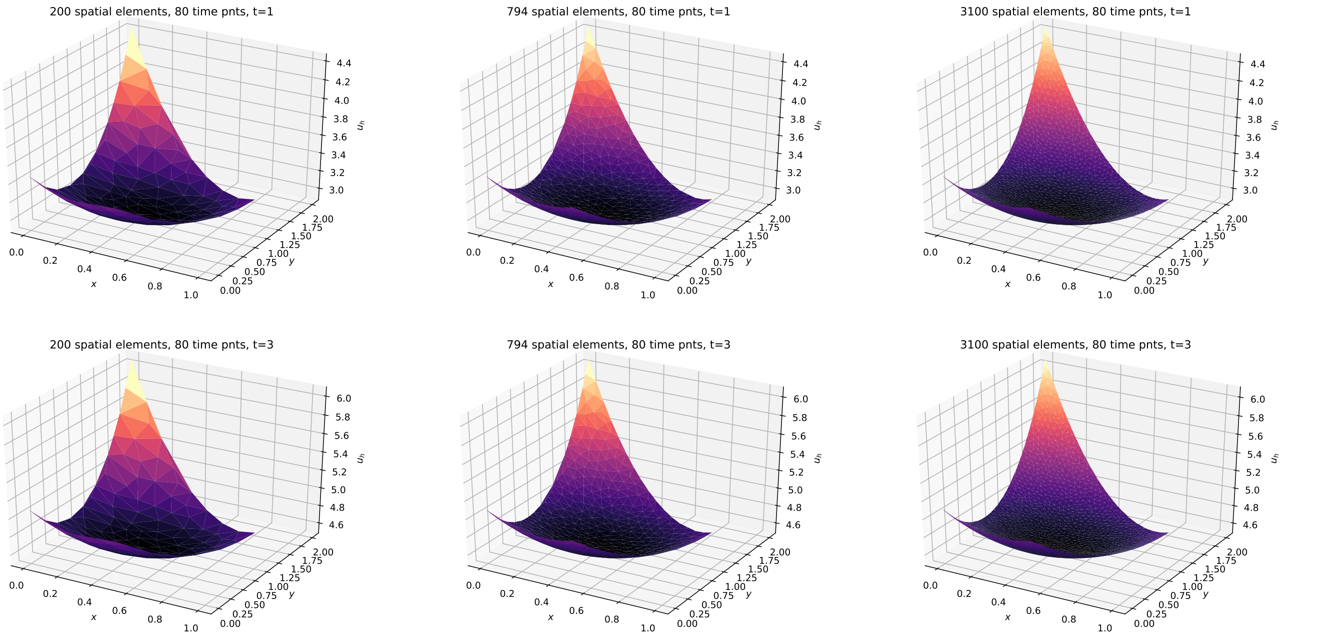

Figure 3: Solutions at times $t = 1$ (first row) and $t = 3$ (second row) with the implicit scheme with 80 time points for each spatial discretization.

4.2 Explicit

For the explicit solves, solutions were plotted at lesser discrete time values to show the evolution of the solution with time. Interestingly, the solution begins to assume the correct amplitude near the edges of the spatial domain first with a lagging of the solution on the interior of the domain. The unstable results show garbage solutions. The instability also seems to accumulate in the solution at the triangular element with the smallest area in the domain.

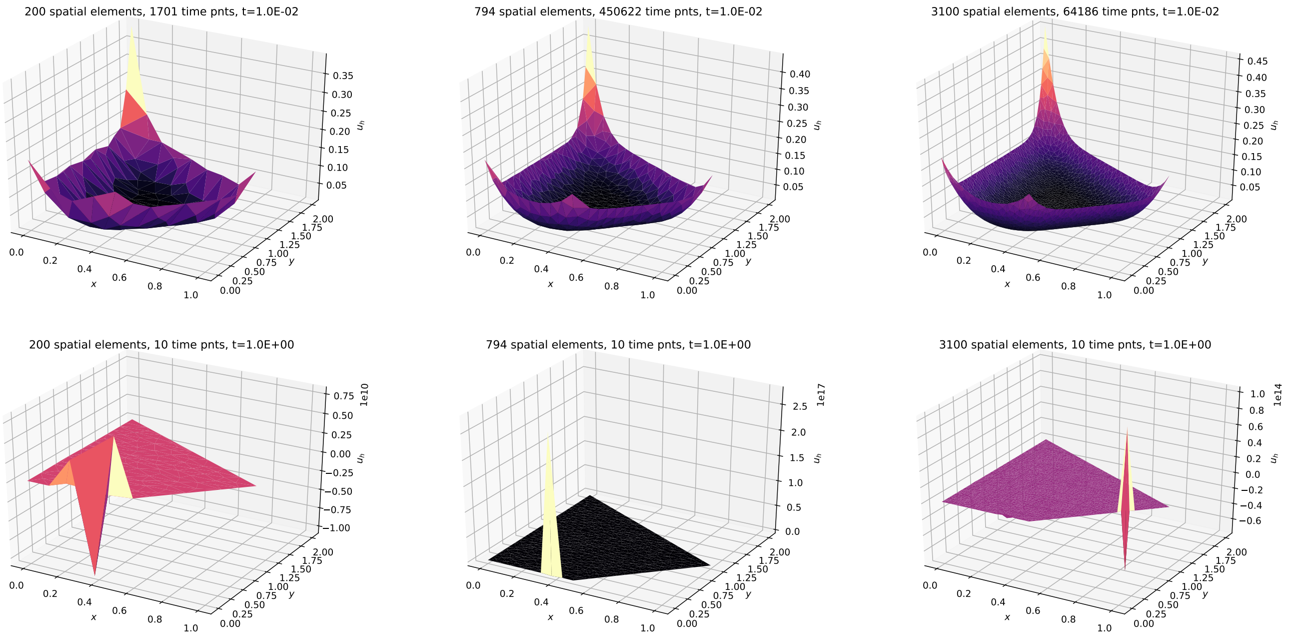

Figure 4: Solutions at times $t = 0.01$ that are stable (first row) and $t = 1$ that are unstable (second row) for each spatial discretization.

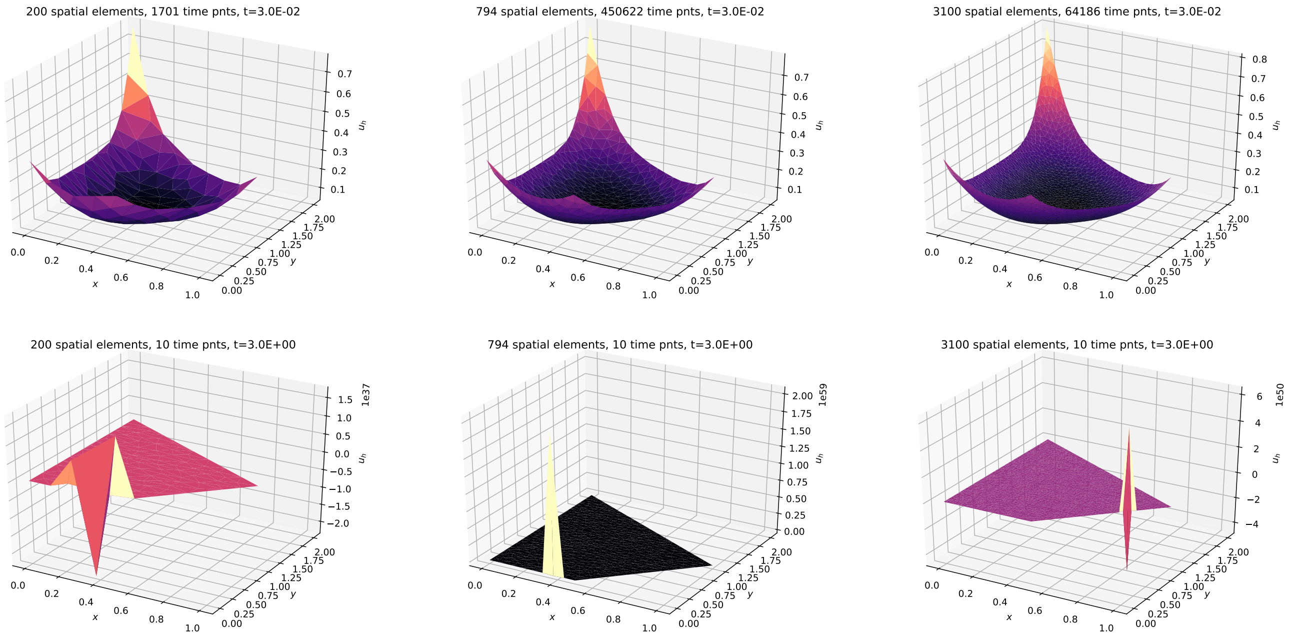

Figure 5: Solutions at times $t = 0.03$ that are stable (first row) and $t = 3$ that are unstable (second row) for each spatial discretization.

5 Problem 3 Results

5.1 Implicit

As in Problem 2, only the solution amplitude changes in time for each respective spatial discretization.

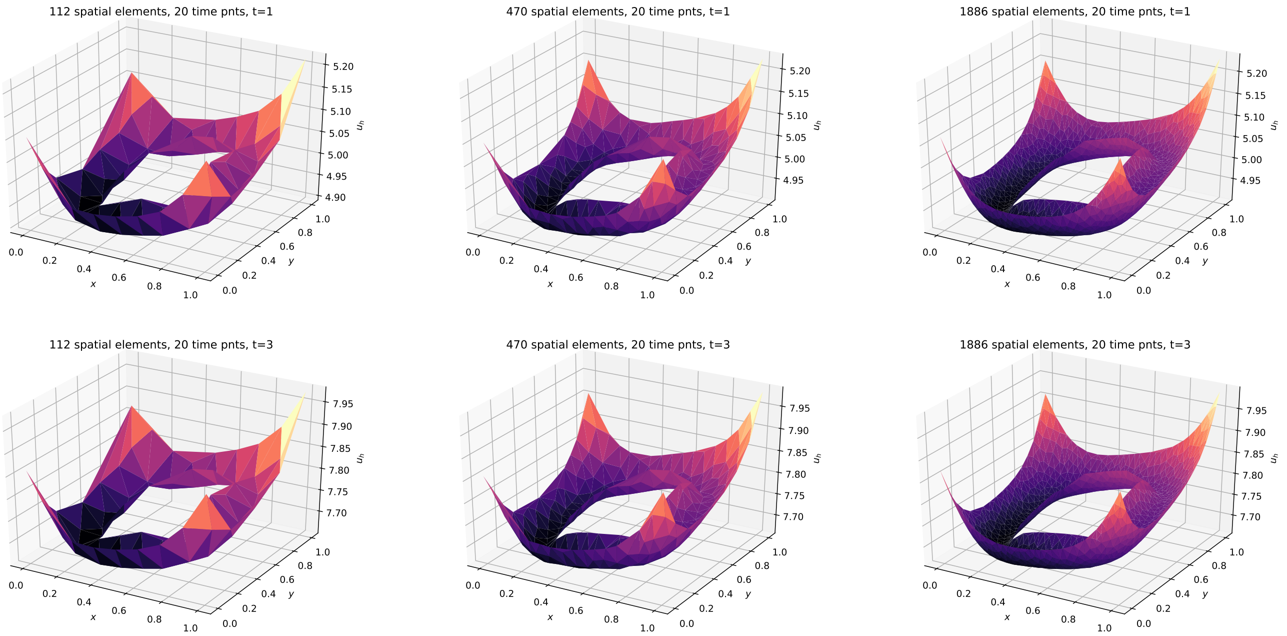

Figure 6: Solutions at times $t = 1$ (first row) and $t = 3$ (second row) with the implicit scheme with 20 time points for each spatial discretization.

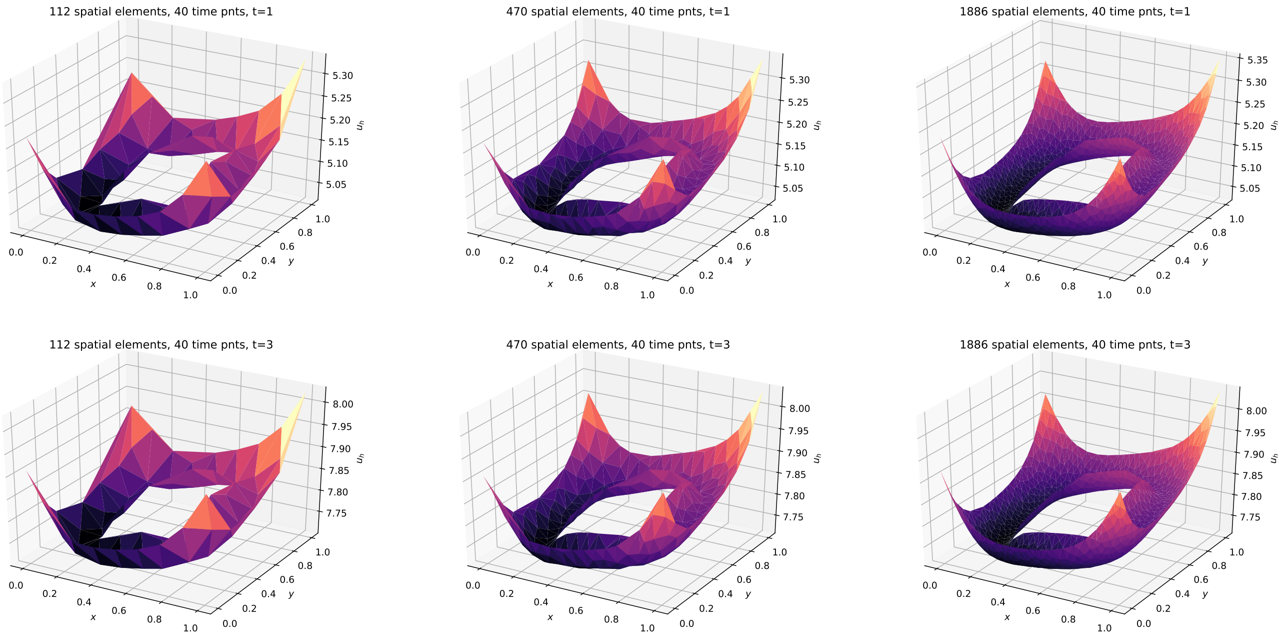

Figure 7: Solutions at times $t = 1$ (first row) and $t = 3$ (second row) with the implicit scheme with 40 time points for each spatial discretization.

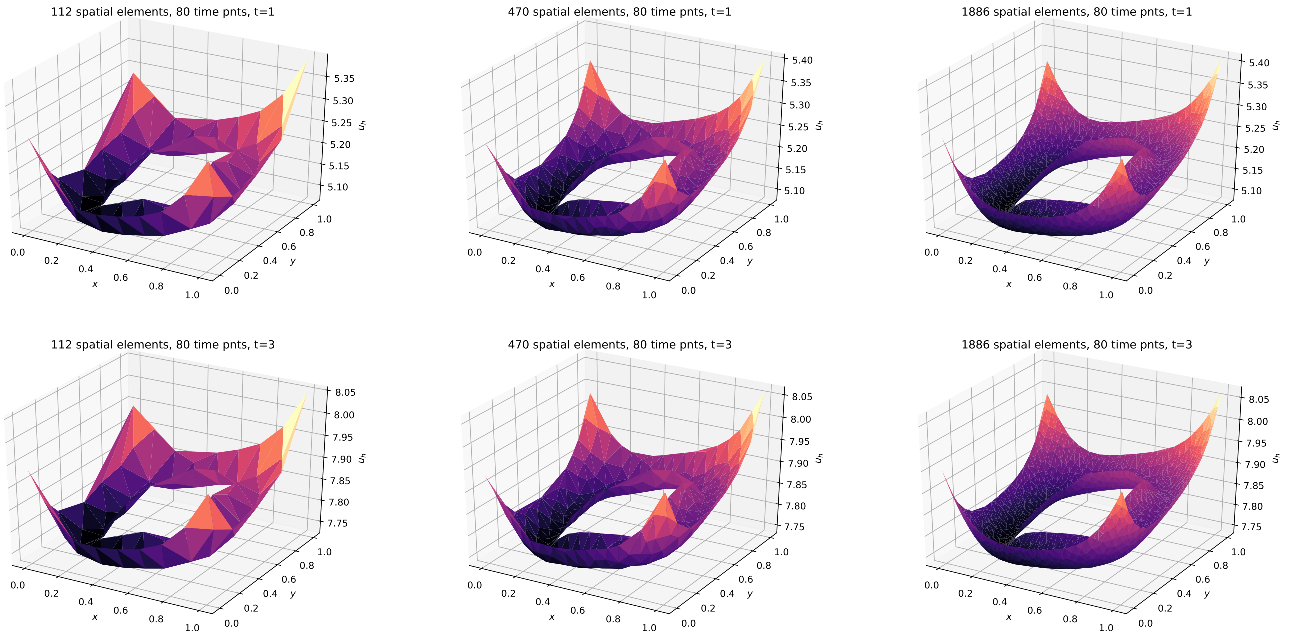

Figure 8: Solutions at times $t = 1$ (first row) and $t = 3$ (second row) with the implicit scheme with 80 time points for each spatial discretization.

5.2 Explicit

Again, solutions are plotted at lesser time values due to the long computation time of finding stable explicit solutions. The unstable solutions are still plotted at $t = 1$ and $t = 3$, as very large time steps can be used to illustrate the instability.

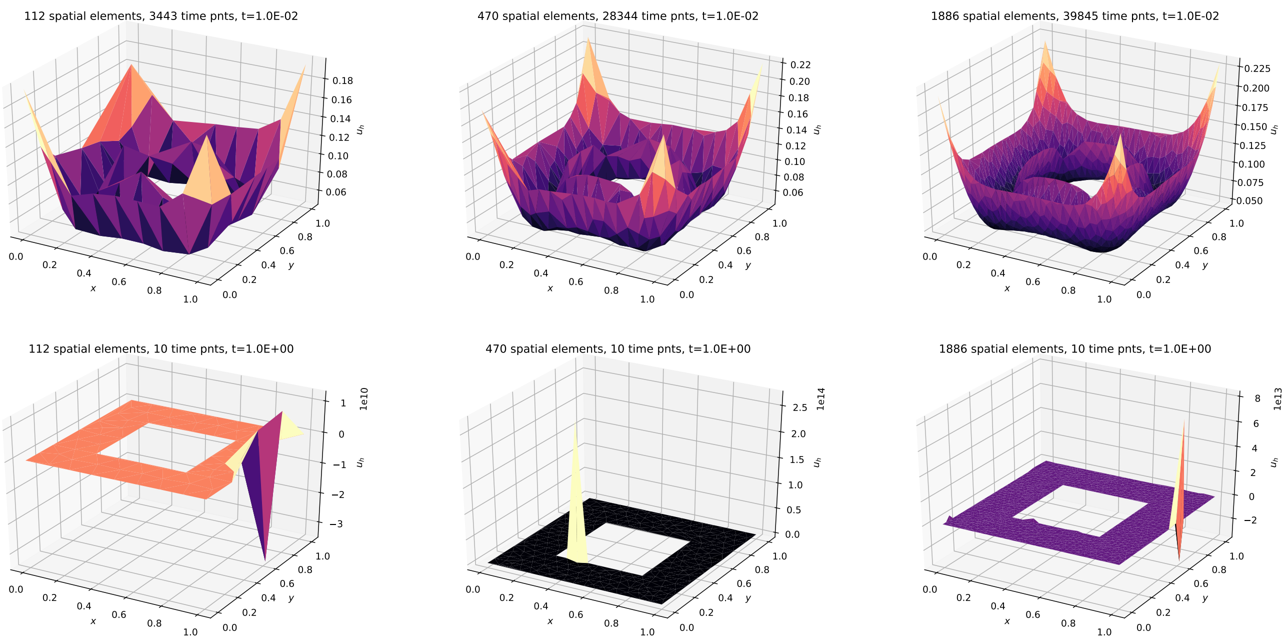

Figure 9: Solutions at times $t = 0.01$ that are stable (first row) and $t = 1$ that are unstable (second row) for each spatial discretization.

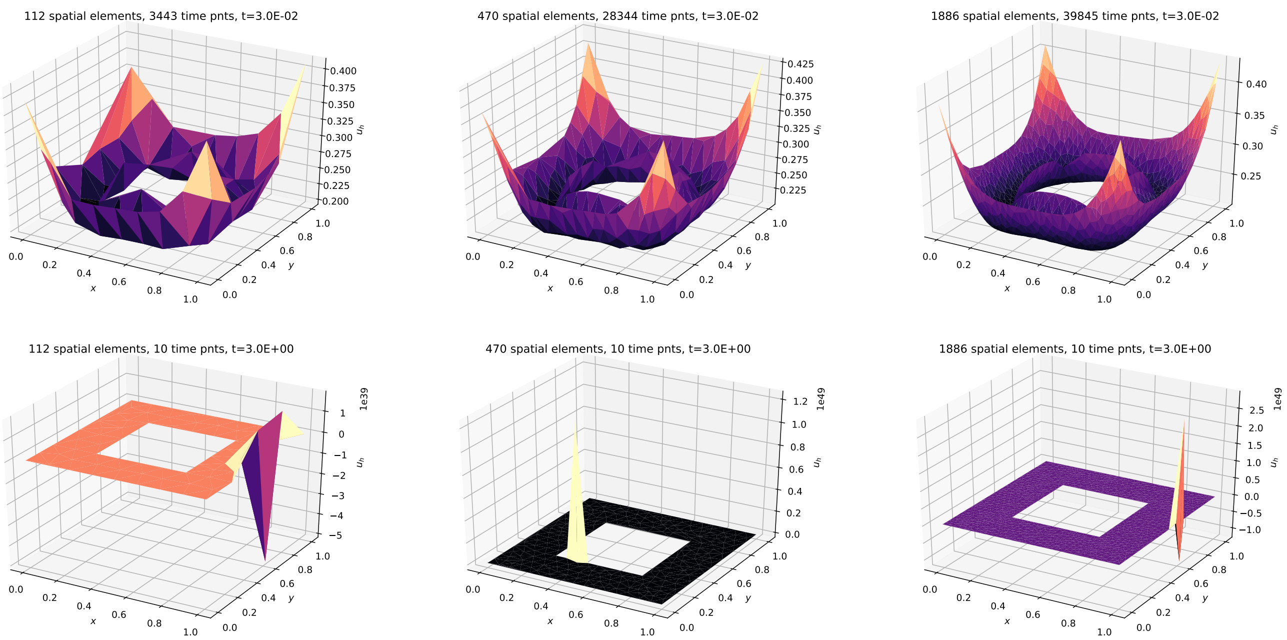

Figure 10: Solutions at times $t = 0.03$ that are stable (first row) and $t = 3$ that are unstable (second row) for each spatial discretization.