Math 228B
Homework 1
Ahmad Zareei

Introduction

In this problem set, we want to solve 1-D heat equation which is written as:

$$u_t = \alpha u_{xx}$$

This equation describes heat (or temperature variation) in a region over time. α is a positive constant and is called thermal diffusivity of the medium. Actually this constant physically imply that how fast the temperature is moving in the region. For solving this 1-D heat equation we need two boundary conditions on x and one initial condition at $t = t_0$. This heat equation is same as diffusion equation for volumetric concentration of particles. In such equations, u is concentration and α will be diffusivity coefficient. The diffusion equation is usually written in the form

$$c_t = Dc_{xx}$$

In the above equation, c is volumetric concentration of particles and D is the diffusion coefficient. Generalized form of these equations (heat or diffusion equation) is usually written as

$$u_t = \nabla.(\alpha\nabla u) + f(x)$$

In this equation $f(x)$ is the source function and vector of $\alpha\nabla u$ is called flux of u in the domain. This flux is proportional to gradient of u times some coefficient. This equation for proportionality of flux and gradient when α is constant, is called Fourier's law in heat transfer.

The other very interesting are that uses this same equation is in quantum mechanics. Shrödinger equation of a free particle is described by the same equation. Shrödinger equation of a particle that has no potential around it (the Hamiltonian of particle just contains the momentum term) is as

$$\Psi_t = \frac{i\hbar}{2m} \triangle \Psi$$

The importance of these equations in physics motivates us to understand how to solve this equation numerically.

Problem

The heat equation that we are solving in the domain $[0, 1]$ is:

$$u_t = \alpha u_{xx}$$

$$\begin{cases} u(x = 0, t) = 0 \\ u(x = 1, t) = 0 \\ u(x, t = 0) = f(x) \end{cases}$$

In this problem set, we first set α to one and then discuss about schemes for different values of α, both when α is constant or is some function of x (either smooth or discontinuous).

The boundary conditions of $u(x = 0, t)$ and $u(x = 1, t)$, in here, are called Dirichlet boundary conditions. The boundary condition, when derivative of the function is known at the boundaries is called Neumann boundary condition. So we are solving 1-D heat equation, with Dirichlet boundary condition with constant conductivity. If α is not equal to one, we can rescale x and t to set values of α to one. In the last part of assignment we will solve this equation when α is some function of x.

Implementation

In this assignment, we will use finite difference method to solve this equation numerically and then analyze our method for convergence and stability conditions. The finite difference method used is as

$$D_0^t u = D_+^x D_-^x u$$

$$\frac{u_j^{n+1} - u_i^n}{k} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}$$

$$u_j^{n+1} = u_j^n + \frac{k}{h^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

We define $\lambda = \frac{k}{h^2}$, so we will have:

$$u_j^{n+1} = \lambda u_{j+1}^n + (1 - 2\lambda)u_j^n + \lambda u_{j-1}^n$$

The above equation is the main method that we will use to solve the heat equation. For analyzing this solution, we suppose that the exact solution of the heat equation, say $v(x, t)$, satisfy the numerical scheme that we have. We check to find how much error we will get if we use this numerical scheme. We should check two conditions of convergence and stability for our scheme.

(a) **Convergence**: We should check that if we put the exact solution in the numerical scheme, the error that we get goes to zero as h and k goes to zero.

For convergence we have put the exact solution into the numerical scheme in order to find the truncation error. we have:

$$\tau_j^n = \frac{v_j^{n+1} - v_i^n}{k} - \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{h^2}$$

Using Taylor's series, we will expand all these terms around v_j^n. This equation will become

$$\begin{aligned}\tau_j^n = \frac{1}{k}(v + kv_t + \frac{k^2}{2}v_{tt} + O(k^3)) - \frac{1}{h^2}(v + hv_x + \frac{h^2}{2}v_{xx} + \frac{h^3}{6}v_{xxx} + \frac{h^4}{4!}v_{xxxx} + \frac{h^5}{5!}v_{xxxxx} \\ - 2v \\ + v - hv_x + \frac{h^2}{2}v_{xx} - \frac{h^3}{6}v_{xxx} + \frac{h^4}{4!}v_{xxxx} - \frac{h^5}{5!}v_{xxxxx} + O(h^6))\end{aligned}$$

So we will have

$$\begin{aligned}\tau_j^n &= v_t - v_{xx} + \frac{k}{2}v_{tt} + O(k^2) - \frac{2h^2}{4!}v_{xxxx} + O(h^4) \\ &= h^2\{\frac{k}{2h^2}v_{tt} - \frac{1}{12}v_{xxxx}\} + O(k^2) + O(h^4)\end{aligned}$$

Considering that

$$v_t = v_{xx} \to v_{tt} = v_{xxt} = v_{txx} = (v_t)_{xx} = v_{xxxx}$$

so we will get $v_t = v_{xxxx}$. With this in mind and knowing that $\lambda = \frac{k}{h^2}$, we have

$$\tau_j^n = h^2 v_{tt}(\frac{\lambda}{2} - \frac{1}{12}) + O(k^2) + O(h^4)$$

In the above equation, we can see the convergence of the method. As $h \to 0$ and $k \to 0$, τ will goes to zero. More than this, we can see that for magical value of $\lambda = \frac{1}{6}$, the order of the method will rise from $O(h^2)$ to $O(h^4)$, which would be really useful.

(b) **Stability**: We should check that the round off error that we have, will always remains bounded, so the solution remains stable.

For stability condition, we use Von Neumann stability analysis. In this method, we consider that the error(ϵ_j^n) is absolute value difference between true solution (fine precision numerically calculated with the scheme) and numerical value with round of error computed at point x_j and time t_n, i.e.

$$\epsilon_j^n = u_j^n - v_j^n$$

We consider that this ϵ_j^n's spatial variation may be expanded in a finite Fourier series as

$$\epsilon_j^n = \sum_n A_n e^{i\xi_n x}$$

in which ξ_n is the n^{th} wave number in the domain ($\xi_n = \frac{\pi n}{L}$). We want to show that for any ξ this error remains bounded and does not grow to infinity. This way we have showed that round off error in initial data remains bounded and our method is stable. We try to show that the error does not grow to infinity for one arbitrary ξ_n. We have:

$$\epsilon_j^{n+1} = \lambda\epsilon_{j+1}^n + (1-2\lambda)\epsilon_j^n + \epsilon_{j-1}^n$$

We consider $\epsilon_j^n = e^{i\xi_n x}$, so we will have

$$\epsilon_j^{n+1} = \epsilon_j^n\{1 + \lambda(e^{i\xi k} + e^{-i\xi k} - 2)\}$$

$$\epsilon_j^{n+1} = \epsilon_j^n\{1 - (2i)^2\lambda(\frac{e^{i\xi k/2} - e^{-i\xi k/2}}{2i})^2\}$$

$$\frac{\epsilon_j^{n+1}}{\epsilon_j^n} = 1 - 4\lambda sin(\frac{k\xi}{2})^2$$

We set $|\frac{\epsilon_j^{n+1}}{\epsilon_j^n}| \leq 1$ for the error not to grow, we will have

$$\lambda \leq \frac{1}{2}$$

with the condition of $\lambda \leq 1$ our scheme will be stable.

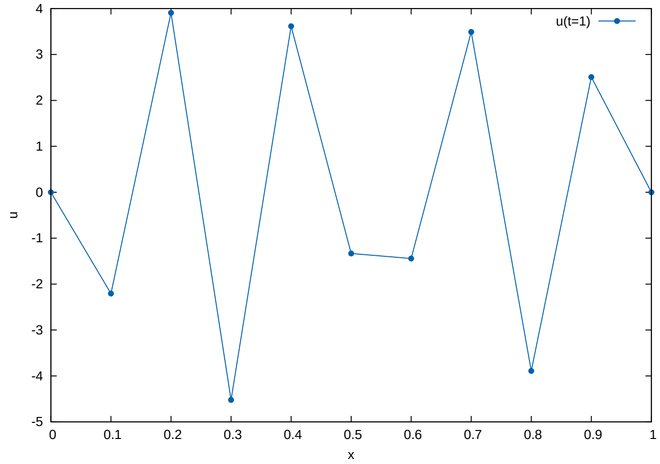

Figure 1: $u(x, t = 1)$ for $h = k = 1/10$

Problems

1. Find (write a program to find) the solution at time $t = 1$ with

   (a) $h = k = 1/10$ and show instability

   As it can be seen from the figure 1, for this method the solution gets unstable for $h = k = 1/10$. The solution oscillates and gets values higher than 1 and also negative values. We also showed in the previous part that for stability (errors not to grow) we should have $\lambda \leq 1$. So the method with $\lambda = 10$ will be unstable.

   (b) $h = 1/10$ and $k = 1/200$ and show stability

   As it can be seen in the figure 2, the solution is stable for this values of h and k, No strange oscillation or negative values is seen. In this case $\lambda = \frac{1}{2}$, so the method will be stable.

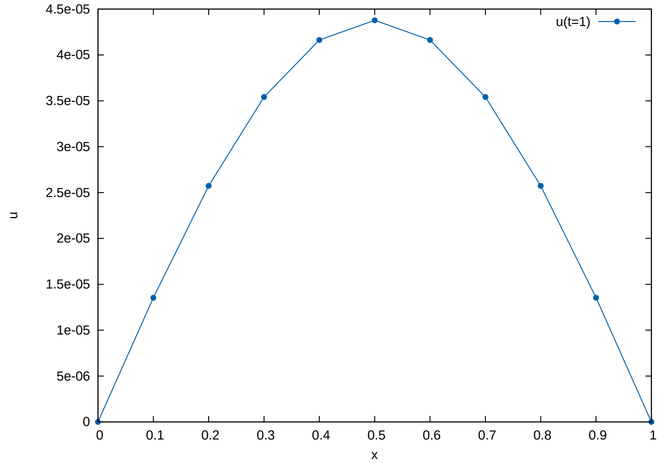

Figure 2: $u(x, t = 1)$ for $h = 1/10$ and $k = 1/200$

(c) Try the trick of $\lambda = 1/6$, and check for a higher order of convergence. That is, compare the convergence rate of the computed solution with $\lambda = 1/2$ to that with $\lambda = 1/6$. How do you do this without the exact solution?

In this part, as we saw in Implementation part, the order of convergence will differ by changing h. We saw that for $\lambda = \frac{1}{2}$ is $O(h^2)$ and for $\lambda = \frac{1}{6}$ is $O(h^4)$. In this part, we have computed the result u for different h and two cases of λ.

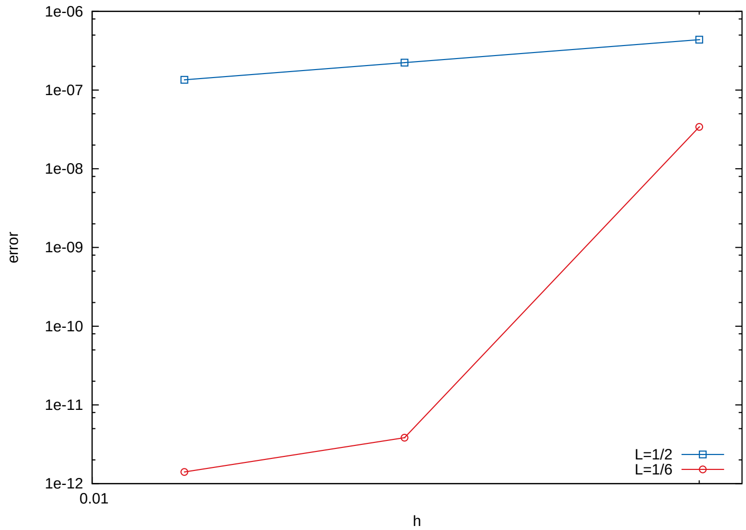

Figure 3: Error vs h for $\lambda = 1/6$ and $\lambda = 1/2$

We take the finest solution when $\lambda = 1/6$ and h is the smallest as the exact solution, then we computed errors (infinity norm) w.r.t. that solution. As the results suggests, for $\lambda = 1/6$ the rate of convergence is faster than $\lambda = 1/2$. We computed the slope of the error for different values of h for these two methods. This slope show the rate of convergence in the scheme. This slope for $\lambda = 1/2$ was 1.9 and for $\lambda = 1/6$ was 3.6. This values matches the order of convergence rate, 2 and 4, that we had found earlier in our convergence analysis.

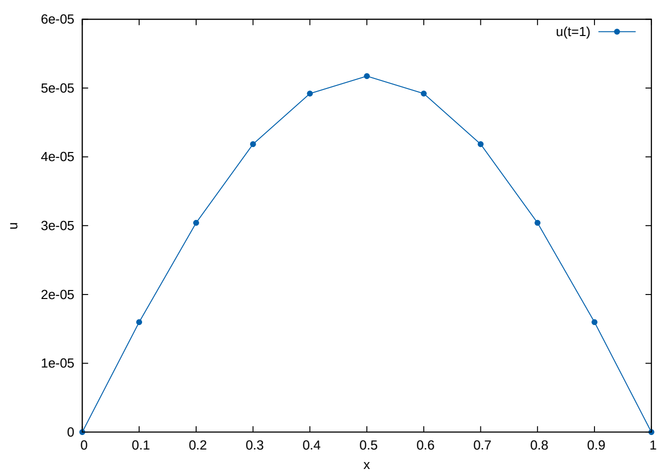

Figure 4: $u(x, t = 1)$ and $\lambda = 1/6$ and $h = 1/10$

2. Find the exact solution analytically (using Fourier Series). How long does it take to evaluate the solution to get the same degree of accuracy? How many terms of this expansion do you need to get the same accuracy as the numerical method? Compare the two methods, in terms of compute time and accuracy, at $T = 1$. Do some experiments.

   Seperation of Variables: suppose u can be written as $u = T(t).X(x)$. Given the differential equation $u_t = u_{xx}$, we have:

$$\frac{T'}{T} = \frac{X''}{X} = -k^2$$

$$X'' = -k^2 X \longrightarrow X = Asin(kx) + Bcos(kx)$$

$$T' = -k^2 T \longrightarrow T = Ae^{-k^2 t}$$

   using boundary conditions of $u(x = 0, t) = u(x = 1, t) = 0$, we will get $B = 0$ and $k = n\pi$. So the solution is

$$u = \sum_{n=1}^{\infty} A_n e^{-n^2\pi^2 t} sin(n\pi x)$$

   using the initial condition of $u(x, t = 0) = sin(\pi x)$, we have

$$u = \sum_{n=1}^{\infty} A_n sin(n\pi x) = sin(\pi x) \rightarrow A_n = \begin{cases} 1 & n = 1 \\ 0 & otherwise \end{cases}$$

   So the exact solution will be

$$u(x, t) = e^{-\pi^2 t} sin(\pi x)$$

   The exact solution has just one term. So only one term is needed to compute the exact analytic solution with no error.

   The computation time for different h, and different λ has been computed in the following tables.

h	0.1	0.1/4	0.1/16	0.1/32	0.1/64
Error	8.00E-007	5.23E-007	1.15E-007	4.28E-008	1.00E-08
Time (sec)	2.00E-005	0.00094	0.00996	0.06	0.955
Exact Sol. Time (sec)	8.00E-007	2.20E-006	4.00E-006	8.00E-006	2.00E-005

Table 1: Elapsing and Error for different values of h for $\lambda = 1/2$

h	0.1	0.1/4	0.1/16	0.1/32
Error	9.29E-009	3.60E-009	7.32E-010	1.31E-010
Time (sec)	5.00E-005	0.004	0.1793	1.478
Exact Sol. Time (sec)	1.00E-006	2.00E-006	1.00E-005	1.00E-005

Table 2: Elapsing and Error for different values of h for $\lambda = 1/6$

   As the above figure suggests, since only one term should be computed for the exact solution in the Fourier series, computation time of exact solution and accuracy of exact solution is much better than numerical method.

3. Take the initial data $u(x, 0) = x(1 - x)$, and do Problem 2 again.

   The proposed $u(x, t)$ which only satisfies Boundary conditions $u(x = 0, t) = u(x = 1, t) = 0$ is

$$u = \sum_{n=1}^{\infty} A_n e^{-n^2\pi^2 t} sin(n\pi x)$$

Satisfying the initial condition, we get

$$u = \sum_{n=1}^{\infty} A_n sin(n\pi x) = x(1-x)$$

using orthogonality of *sin* and *cos*, we have:

$$\sum_{n=1}^{\infty} \int_0^1 A_n sin(n\pi x) sin(m\pi x) dx = \int_0^1 x(1-x) sin(m\pi x) dx$$

$$\frac{1}{2m\pi} A_n \delta_{mn} = \int_0^1 x(1-x) sin(m\pi x) dx$$

So we just need to compute the RHS of the above equation. we have

$$\begin{aligned}\int_0^1 x(1-x) sin(m\pi x) dx &= \int_0^{m\pi} \frac{1}{m^2\pi^2} usin(u) du - \int_0^{m\pi} \frac{1}{m^3\pi^3} u^2 sin(u) du \\ &= \frac{(-1)^{m+1}}{m\pi} - \frac{(-1)^{m+1}}{m\pi} - 2\frac{(-1)^m - 1}{(m\pi)^3}\end{aligned}$$

simplifying above equation, we have

$$A_n = \begin{cases} 0 & n = 2k \\ \frac{8}{(n\pi)^2} & n = 2k+1 \end{cases}$$

So the exact solution is

$$u = \frac{8}{\pi^2} \sum_{k=0}^{+\infty} \frac{e^{-(2k+1)^2\pi^2 t}}{(2k+1)^2} sin((2k+1)\pi x)$$

As it is seen in the above equation, we need infinite number of terms in the Fourier series to compute the exact solution. However, the coefficient are decreasing exponentially. So for instance, the ratio of third term over first term is as

$$\frac{e^{-25\pi^2}}{e^{-9*pi^2}} \simeq 2.6E-69$$

So we just need first two terms of the series to capture the exact solution's trend. We computed elapsing time for different values of h, and the results are as follows

h	0.1	0.1/4	0.1/16
Error	2.04E-006	1.35E-007	1.10E-008
Time (sec)	2.00E-005	0.00096	0.0639
Exact Sol. Time (sec)	1.50E-006	4.00E-006	1.40E-005

Table 3: Elapsing and Error for different values of h for $\lambda = 1/2$

This shows again that analytic scheme is faster and more accurate. This is not true for all the cases. if the initial value of u has some discontinuity, the heat equation will smooth the discontinuity. However when we try to write this initial condition to find the coefficient, because of the discontinuity, the fourier series can fully capture the initial data (because of Gibb's Phenomena). In this case the numerical methods works better and gives more accurate answers, because no matter how many terms we keep

h	0.1	0.1/4	0.1/16
Error	1.85E-009	4.74E-008	8.57E-010
Time (sec)	0.0001	0.0156	0.1937
Exact Sol. Time (sec)	1.00E-006	2.00E-006	1.00E-005

Table 4: Elapsing and Error for different values of h for $\lambda = 1/6$

in the Fourier series, the initial value of u computed with Fourier series, will not be the same as what it should be.

For instance, if we take the initial condition to be

$$x = \begin{cases} 1 & 0 \leq x < 0.5 \\ 0 & 0.5 \leq x < 1 \end{cases}$$

The initial value, computed by Fourier series by 5 terms for the initial condition would look like the following figure

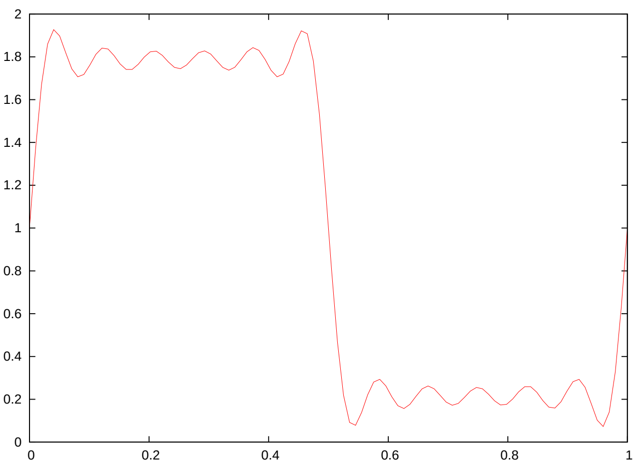

Figure 5: Initial condition of a discontinuous function computed by 5 terms of Fourier Series

No matter how many terms we pick in Fourier Series, we will still have overshooting at discontinuous points. For such cases, numerical method is more faster and accurate.

4. Consider $u_t = \alpha u_{xx}$ with $u(x,0) = sin(\pi x)$ on $[0,1]$, and boundary condition $u(0,t) = u(1,t) = 0$. Does this change the restriction on lambda for stability? How?

   Considering a constant α will change the stability restriction, since

$$u_t = \alpha u_{xx}$$

$$\frac{u_j^{n+1} - u_j^n}{k} = \alpha \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}$$

$$u_j^{n+1} - u_j^n = \alpha \frac{k}{h^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

$$u_j^{n+1} - u_j^n = \alpha\lambda(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

$\alpha\lambda$ found in the above equation is same as λ that we found in previous schemes that we had before. So for our scheme to be stable we should have this new λ to be less than $\frac{1}{2}$, so

$$\alpha\lambda \leq \frac{1}{2}$$

So, λ in our scheme should be $\lambda \leq \frac{1}{2\alpha}$. In the above procedure, we have supposed that α is positive, which is the physical case. if α was negative, there is no restriction on λ for the method to be stable. since $\alpha\lambda$ is already negative and less than $\frac{1}{2}$.

5. Consider $u_t = \alpha(x)u_{xx}$ with $u(x,0) = sin(\pi x)$ on $[0,1]$, and boundary condition $u(0,t) = u(1,t) = 0$. So now, α depends on x. See if you can get a scheme to work in two cases: first, when $\alpha(x)$ is smooth, and second, when $\alpha(x)$ is discontinuous. How are you checking that it works? What may go wrong?

   Considering that α could be some function of x in our shceme, we will have

$$u_t = \alpha(x)u_{xx}$$

$$D_0^t u = \alpha(x)D_+^x D_-^x u$$

$$\frac{u_j^{n+1} - u_j^n}{k} = \alpha_j \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}$$

   In the above equation $\alpha_j = \alpha(x_j)$. So we will have

$$u_j^{n+1} - u_j^n = \alpha_j \frac{k}{h^2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

$$u_j^{n+1} = u_j^n + \alpha_j\lambda(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

   For this method to be stable, we should have for all α_j the value of $\alpha_j\lambda$ to be less than $\frac{1}{2}$. This way we make sure that our scheme is stable in the whole domain. we can have two approaches in here

   (a) we can find maximum value of α in the domain, and then take some h and k such that $\alpha^{max}\lambda < \frac{1}{2}$. This way we are sure that for all points, value of $\alpha_j\lambda$ is less than $\frac{1}{2}$, so our method is stable.

   (b) Or we can, set different h and k for different part of domain, assuring that in all parts $\alpha_j\lambda$ is less than 1/2.

   In the first method, we are not using the fact that in some areas, we can integrate faster (larger h or k) with the same accuracy. If we use this fact, we are using method (b). However, in method (b), we will get into trouble of patching all different parts together, since h and k will not be the same for all regions. we should check depending on the behavior of α in the domain which method to pick.

   We tried to solve the heat equation with discontinuous α. α is defined as follows

$$\alpha(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \leq x < 1 \end{cases}$$

   The solution as shown in Figure 6, is stable and is decreasing with time.

   As it is also seen in the Figure 7, the solution is converging by decreasing h for a fixed value of *lambda*

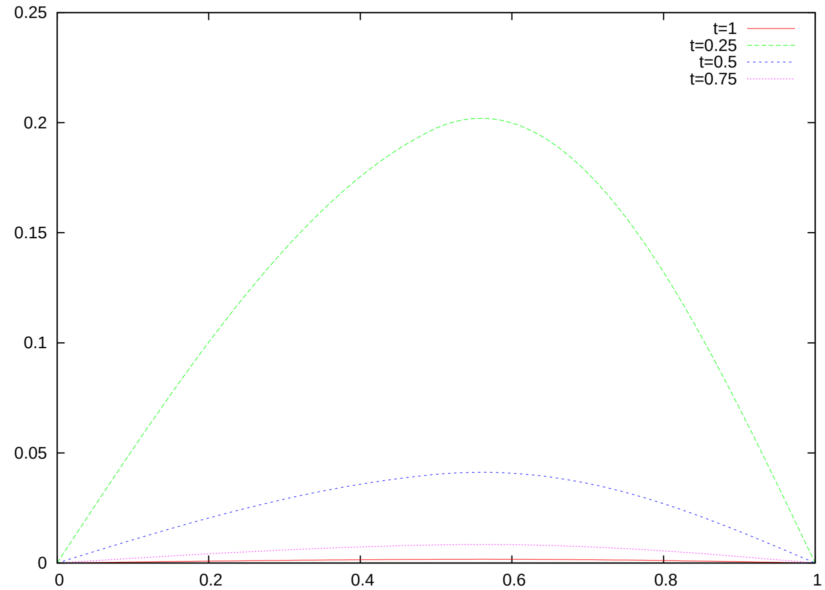

Figure 6: Solutions at different times for discontinuous $\alpha(x)$

Conclusion

In this problem set we discussed how to solve 1-D heat equation with different conductivity, either constant or with some function or discontinuous. we also talked about stability and convergence of the numerical scheme that we used. we found that for $\lambda > \frac{1}{2}$ our method will get unstable. we also saw that for certain ratio of $\lambda = \frac{1}{6}$ we will have higher order convergence.

We also found that for discontinuous initial condition, we will have trouble with analytical solution using Fourier Series.

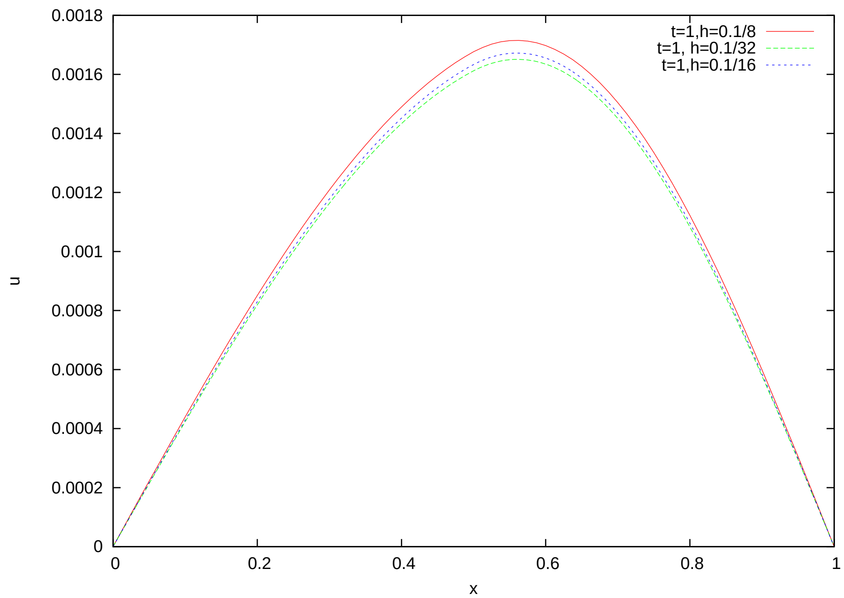

Figure 7: Solution with different h, at $t = 1$