Part 4: Optimization (V)

A multi-dimensional constrained optimization is one that, in mathematical terms,

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad for\ \boldsymbol{x} \in \boldsymbol{R^n}$$

$$\begin{aligned} subject\ to\ F_j(\boldsymbol{x}) &= a_j, \quad j = 1, 2, \dots, m_1 \\ G_k(\boldsymbol{x}) &\le b_k, \quad k = 1, 2, \dots, m_2 \\ and\ \boldsymbol{U_L} &\le \boldsymbol{x} \le \boldsymbol{U_P} \end{aligned}$$

Where $f(\boldsymbol{x})$ is a continuous real-valued function.

The bounds can be expressed as inequalities such that the constraints are either equality-type or inequality-type.

Linear Programming:
If f, F_j and G_k are linear functions, that is,

$$f(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x}$$

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{Ax} - \{a_j\} = \boldsymbol{0}$$
$$\boldsymbol{G}(\boldsymbol{x}) = \boldsymbol{Bx} - \{b_k\} \le \boldsymbol{0}$$

$$\boldsymbol{x} \ge \boldsymbol{0}$$

Where **c** is a vector, and **A** and **B** are matrices), the optimization problem can and should be solved by linear programming as it is the most effective method for such optimizations.

Quadratic Programming:
If f is a quadratic function, while F_j and G_k remain linear, that is:

$$f(\boldsymbol{x}) = \boldsymbol{c}^T\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T\boldsymbol{Qx}$$

$$F(\boldsymbol{x}) = \boldsymbol{Ax} - \{a_j\} = 0$$
$$G(\boldsymbol{x}) = \boldsymbol{Bx} - \{b_k\} \le 0$$

$$\boldsymbol{x} \ge \boldsymbol{0}$$

Where $\boldsymbol{c}$ is a vector, $\boldsymbol{Q}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ are matrices, and $\boldsymbol{Q}$ is positive definite o rnegative definite), the optimization problem can and should be solved by quadratic programming as it is the most effective method for such optimizations.

General multi-dimensional nonlinear constrained optimization:

- Method of Lagrange multipliers
- Method of penalty functions
- Exterior penalty
- Interior penalty

Method of Lagrange multipliers:

Construct the Lagrange function as follows:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{j=1}^{m_1} \lambda_j \left(F_j(\boldsymbol{x}) - a_j\right) + \sum_{k=1}^{m_2} \mu_k (G_k(\boldsymbol{x}) - b_k)$$

Where $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ contain the λ_j and μ_k, respectively. $\boldsymbol{x}$ is known as the primal variables, while $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are the dual variables.

The Lagrange function transforms the constrained optimization problem into an unconstrained one but increases the dimension to $n + m_1 + m_2$.

Mathematically, the duality theorem stipulates the conditions on the optimal solution.

For not-too vigorous take at the theorem:

1. Zero gradient: $\nabla\mathcal{L} = \mathbf{0}$, or $\frac{\partial\mathcal{L}}{\partial\boldsymbol{x}} = \mathbf{0}$, $\frac{\partial\mathcal{L}}{\partial\boldsymbol{\lambda}} = \mathbf{0}$ and $\frac{\partial\mathcal{L}}{\partial\boldsymbol{\mu}} = \mathbf{0}$
2. Constraints are met.
3. $\boldsymbol{\lambda}^T(\boldsymbol{Ax} - \boldsymbol{a}) = 0$, $\boldsymbol{\mu}^{\boldsymbol{T}}(\boldsymbol{Bx} - \boldsymbol{b}) = 0$, with $\boldsymbol{\lambda} \geq \mathbf{0}$, and $\boldsymbol{\mu} \geq \mathbf{0}$

Example: minimizing the following:

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$

Subject to $x_1 = 1$ and $x_1 - x_2 \geq 0$

The Lagrange is:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3) + \lambda(x_1 - 1) + \mu(x_2 - x_1)$$

Applying Condition 1:

$$x_1 = 1, x_2 = 1, \lambda = 0.4, \mu = 0.4$$

Check with Condition 2:

$$\begin{aligned} &x_1 = 1,\ \ true \\ &x_1 - x_2 \geq 0,\ \ true \end{aligned}$$

Check with Condition 3:

$$\begin{aligned} &\boldsymbol{\lambda}^T(\boldsymbol{Ax} - \boldsymbol{a}) = 0,\ \ true \\ &\boldsymbol{\mu}^T(\boldsymbol{Bx} - \boldsymbol{b}) = 0,\ \ true \\ &\boldsymbol{\lambda} \geq \boldsymbol{0},\ \ true \\ &\boldsymbol{\mu} \geq \boldsymbol{0},\ \ true \end{aligned}$$

Steepest-descent with Newton's method yields:

$$\boldsymbol{x}^* = [\,1, 1, 0.4, 4]^T, \qquad f(\boldsymbol{x}^*) = 4$$

Method of exterior penalty functions:

Feasible region means the region, within the n −dimensional space, where all constraints are met. Constraints define the boundaries of the feasible region.

The exterior penalty functions method is applicable when the iteration points $\boldsymbol{x}_i$ are outside the feasible region.

The method works well with both the equality-type and inequality-type of constraints.

As to what penalty functions to use, it is heuristic.

Example: Minimizing the following:

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$

Subject to $x_1 = 1$, and $x_1 - x_2 \geq 0$.

The penalty functions may be:

For $x_1 - x_2 \geq 0$: $\Phi(\boldsymbol{x}) = (x_1 - x_2)^p$

For $x_1 = 1$: $\psi(\boldsymbol{x}) = (x_1 - 1)^q$

With $p = 2, 4, \ldots$ and $q = 2, 4, \ldots$

Then a Lagrange function is formed, say,

$$\mathcal{L}(\boldsymbol{x};\ \lambda, \mu) = f(\boldsymbol{x}) + \lambda\psi(\boldsymbol{x}) + \mu\phi(\boldsymbol{x})$$

Which is optimized, treating λ, μ as parameters of increasing values.

Setting $p = 4,\ q = 2,\ \lambda = 1,\ \mu = 100,\ \boldsymbol{x_0} = [0, 1]^T,\ \Delta = 0.1$

Using steepest descent + Golden-section search.

$\boldsymbol{x}^* =$
1.149322837308608
1.356474098102844

$\boldsymbol{\mathcal{L}}^* =$
3.371661647463247

$\boldsymbol{f}^* =$
3.165223411499403

Now, $\lambda = 10000,\ \mu = 10000$

$\boldsymbol{x}^* =$
1.000018513959687
1.046050470120337

$\boldsymbol{\mathcal{L}}^* =$
3.862886413499447

$\boldsymbol{f}^* =$
3.817983881276812

Method of interior penalty functions:
The interior penalty functions method is applicable if and only if the solutions points x_i are within the feasible region.

The method works better with inequality-type of constraints.

The penalty functions are to force the points to move away from the boundaries. They are gence known as the barrier functions.

Again, the choices of penalty functions are heuristic.

Example: Minimizing the following:

$$f(x) = (x_1 - 1)^2 + (x_2 - 3)^2 - 1.8(x_1 - 1)(x_2 - 3)$$

Subject to $x_1 - x_2 \geq 0$.

The interior penalty functions may be:

$$\Phi(x) = \frac{1}{(x_1 - x_2)^2}$$

Or

$$\Phi(x) = -\ln(x_1 - x_2)$$

Note that both functions approach $+\infty$ when x_1 approaches x_2 while meeting the constraint.

The Lagrange function is then formed,

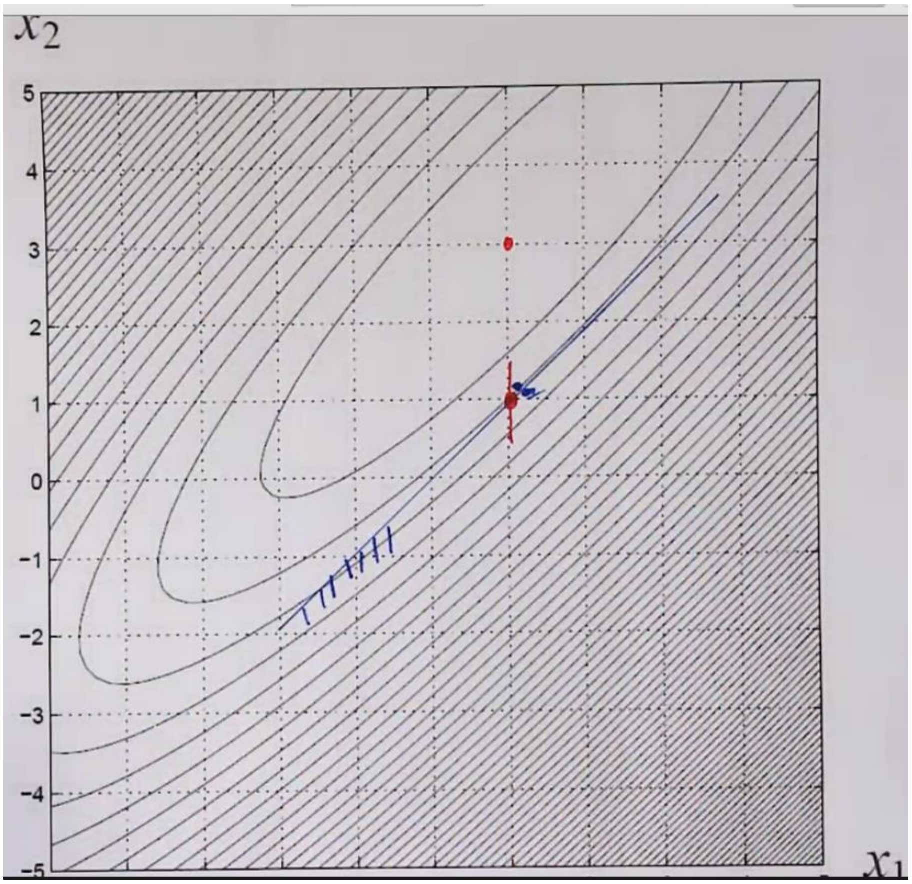
$$\mathcal{L}(x; \mu) = f(x) + \mu\Phi(x)$$

Which is optimized, treating μ as a parameter of decreasing values.

Summary:

- The method of penalty functions does not yield exact solutions.
- Optimization performance is heavily dependent on the choices of penalty functions and penalty parameters.
- Hessians may become ill-conditioned due to large penalty parameters.
- The method of Lagrange multipliers gives rise to exact results (or as close to exact as possible). The dimension of the problem is increased from n to $n + m_1 + m_2$.

Visual Explanation:



Part 5: Finite Difference Method

This part concerns itself with finite difference method as a numerical tool for solving differential equations (DEs).

The Big O Notation

In mathematics, the big O notation, such as O(δ^n), is used to indicate the order of accuracy or order of error. For example, if $n = 2$, one says that it is second order accurate.

Overview

Finite difference method comes with explicit and implicit versions, and the combinations of as well.

Explicit schemes are easy to use but the stability conditions must be adhered to. Explicit schemes are in general less accurate than the implicit ones.

Incorporating boundary conditions may be tedious but is the key to success.

Finite Difference Method for One-Dimensional DEs

Here, dimensions refer to spatial dimension. For one-dimensional DEs, the spatial coordinator is x. The temporal "coordinate" may come into the picture, depending on the DE.

Finite Difference for first-order derivatives

Forward difference:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} + O(\Delta x)$$

Backward difference:

$$f'(x) = \frac{f(x) - f(x + \Delta x)}{\Delta x} + O(\Delta x)$$

Central difference (first order):

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + O(\Delta x^2)$$

The central difference is one-order more accurate than the forward or backward difference.

Finite Difference (FD) for second-order derivatives

Central difference (second order):

$$f''(x) = \frac{f(x + \Delta x) - 2f(x) + f(x + \Delta x)}{\Delta x^2} + O(\Delta x^2)$$

That is, the error is of the order Δx^2.

The Hear Equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, L], t \geq 0$$

Assume forward difference for the temporal domain and central difference for the spatial domain, then the heat equation is discretized as:

$$\frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} = \kappa \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} + O(\Delta t, \Delta x^2)$$

The spatial domain is divided into N even intervals such that $x_n = n\Delta x$ with $n = 0, 1, 2, \dots, N$

The temporal domain is discretized by Δt such that $t_k = k\Delta t$, where $k = 0, 1, 2, \dots, K$.

Denoting $u(x_n, t_k)$ by u_n^k, the above equation becomes, neglecting the big O,

$$\frac{u_n^{k+1} - u_n^k}{\Delta t} = \kappa \frac{u_{n+1}^k - 2u_n^k + u_{n-1}^k}{\Delta x^2}$$

Solving for u_n^{k+1} ,

$$u_n^{k+1} = u_n^k + \kappa \frac{\Delta t}{\Delta x^2} \left(u_{n+1}^k - 2u_n^k + u_{n-1}^k \right)$$

This is the iteration scheme to go from time step k to time step $k + 1$.

Stability condition of the scheme:

$$\frac{\kappa \Delta t}{\Delta x^2} \leq \frac{1}{2}$$

Dirichlet Boundary Conditions

$$u(0,t) = A, \quad u(L,t) = B$$

After initial condition: $u(x,0) = f(x)$

Initial condition: $u_n^0 = f(x_n)$ where $n = 0, 1, 2, \dots, N$.

Boundary conditions: $u_0^k = A,\ \ u_N^k = B$ for all $k > 0$.

The iteration steps:

```
Assign initial condition $u_n^0$

for $k = 0, \dots, K-1$

  $u_0^{k+1} \leftarrow A$
  $u_N^{k+1} \leftarrow B$

  for $n = 1, \dots, N-1$

    $u_n^{k+1} \leftarrow u_n^k + \kappa \frac{\Delta t}{\Delta x^2} \left(u_{n+1}^k - 2u_n^k + u_{n-1}^k\right)$
  end

end
```

Neumann Boundary Conditions

$$\frac{\partial u}{\partial x}(0,t) = C, \quad \frac{\partial u}{\partial x}(L,t) = D$$

And initial condition: $u(x,0) = f(x)$

Using central difference on $\frac{\partial u}{\partial x}$

$$\frac{\partial u}{\partial x}(0,t) = \frac{u(\Delta x,t) - u(\Delta x,t)}{2\Delta x} = \frac{u_1^k - u_{-1}^k}{2\Delta x} = C$$

The mesh point $x = -\Delta x$ does not exist. However, u_{-1}^k can be determined as follows

$$u_{-1}^k = u_1^k - 2\Delta x C$$

$$\therefore u_0^{k+1} = u_0^k + \kappa \frac{\Delta t}{\Delta x^2}\left(-2u_0^k + 2u_1^k - 2\Delta x C\right)$$

By the same token, $x = L + \Delta x$ does not exist but

$$u_{N+1}^k = u_{N-1}^k + 2\Delta x D$$

$$\therefore u_N^{k+1} = u_N^k + \kappa \frac{\Delta t}{\Delta x^2}\left(2u_{N-1}^k - 2u_N^k + 2\Delta x D\right)$$

The iteration scheme remains the same as with Dirichlet boundary conditions.

Mixed boundary conditions; Robin boundary conditions

Apply the principles shown above.

Example: $\kappa = 0.835, L = 10, N = 10; \Delta t = 0.5\ s, t \in [0, 10]$

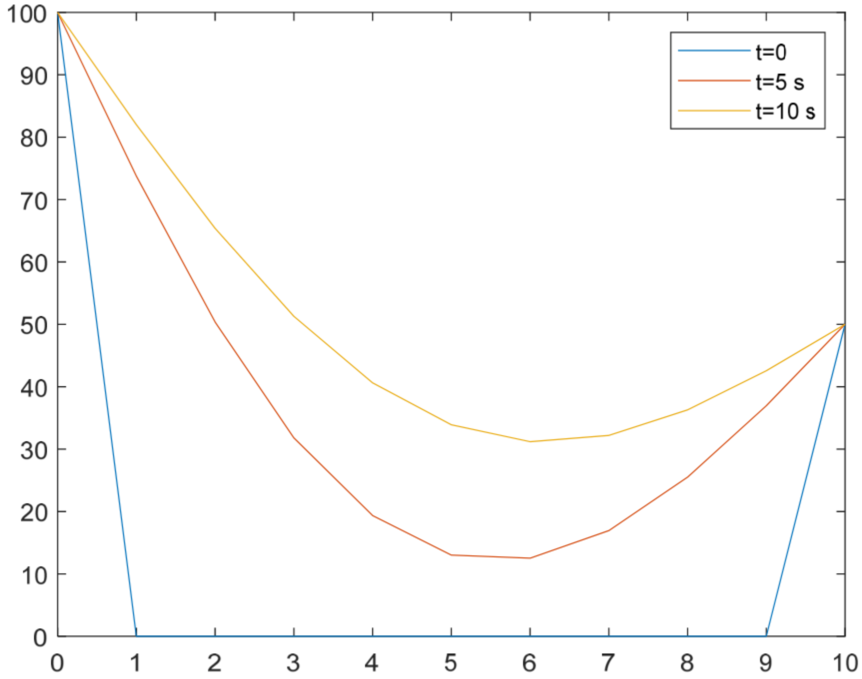
$$u(0,t) = 100, \quad u(L,t) = 50, \quad and\ u(x,0) = 0$$

Note these boundary conditions are maintained at all times.

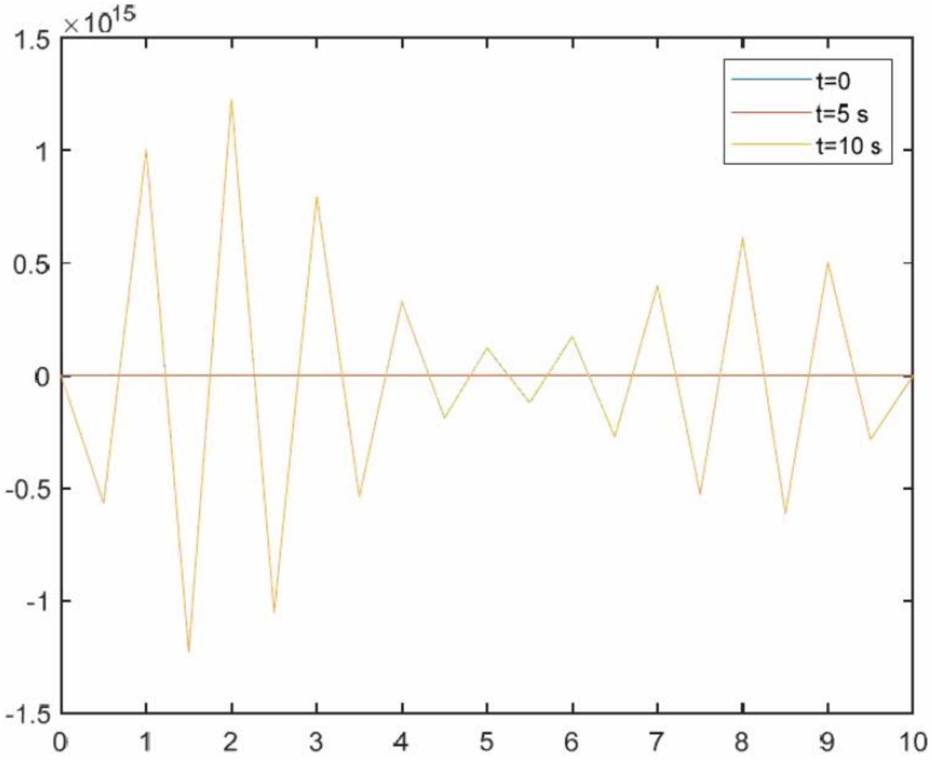
This example is available from “Numerical Methods for Engineers”, Chapter 30. An excel sheet will accompany this file. The sheet has results computed with $\Delta x = 2,\ \Delta t = 0.1$ for 5 time steps.

Stability condition:

$$\frac{\kappa \Delta t}{\Delta x^2} = \frac{(0.835)(0.5)}{1^2} = 0.4175 \leq \frac{1}{2}$$


100
90
80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
9
10
t=0
t=5 s
t=10 s

Δx = 0.5:

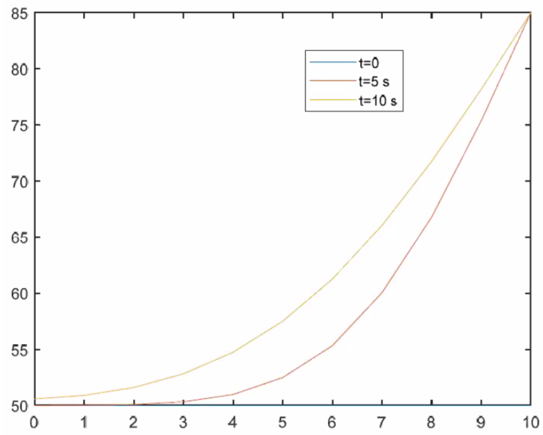

×10^15
1.5
1
0.5
0
-0.5
-1
-1.5
0
1
2
3
4
5
6
7
8
9
10
t=0
t=5 s
t=10 s

Example: $\kappa = 0.835, L = 10, N = 10; \Delta t = 0.5\ s, t \in (0, 10]$

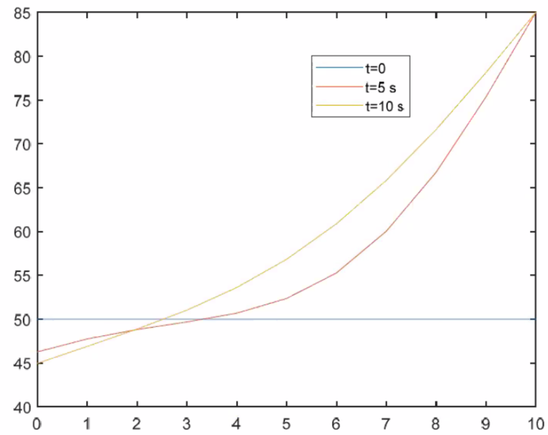
$$\frac{\partial u}{\partial x}(0,t) = 0\ or\ 1, \quad u(L,t) = 85, \quad and\ u(x,0) = 50$$

Stability condition:

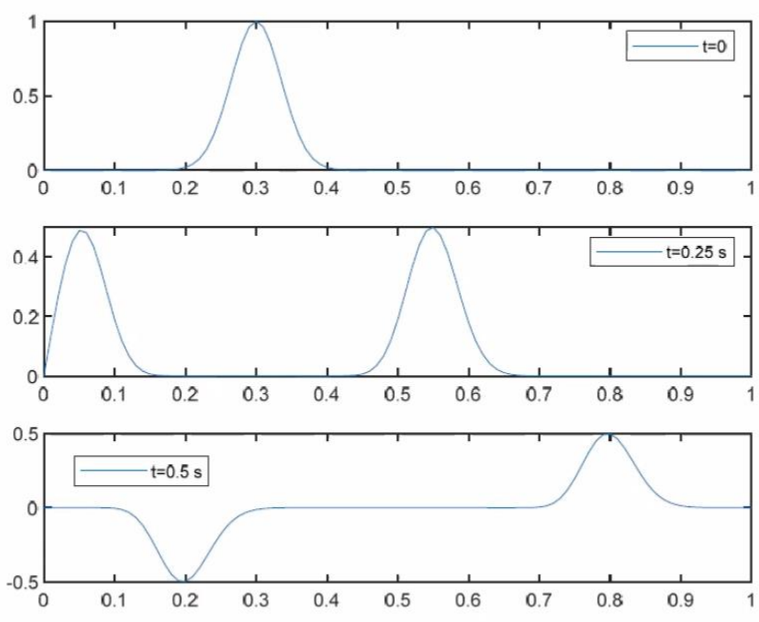
$$\frac{\kappa \Delta t}{\Delta x^2} = \frac{(0.835)(0.5)}{1^2} = 0.4175 \leq \frac{1}{2}$$



$$\frac{\partial u}{\partial x}(0,t) = 1$$



For assignment (wave equation):


t=0
1
0.5
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
t=0.25 s
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
t=0.5 s
0.5
0
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1