AE 342 : Modeling and Analysis Lab, Session 5

Gaurav Gupta, SC21B026, Aerospace Engineering

November 24, 2023

1 Problem 1

A simple 1D transient heat conduction is simulated. The governing equations andboundary conditions are provided as follows.

$$\frac{du}{dt} = \frac{d^2u}{dx^2},\ 0 < x < 1,\ t > 0 \tag{1}$$

$$u(x=0) = x - x^2,\ u(0,t) = u(1,t) = 0 \tag{2}$$

The problem is solved in FreeFEM++ using the given code for a 1D domain.

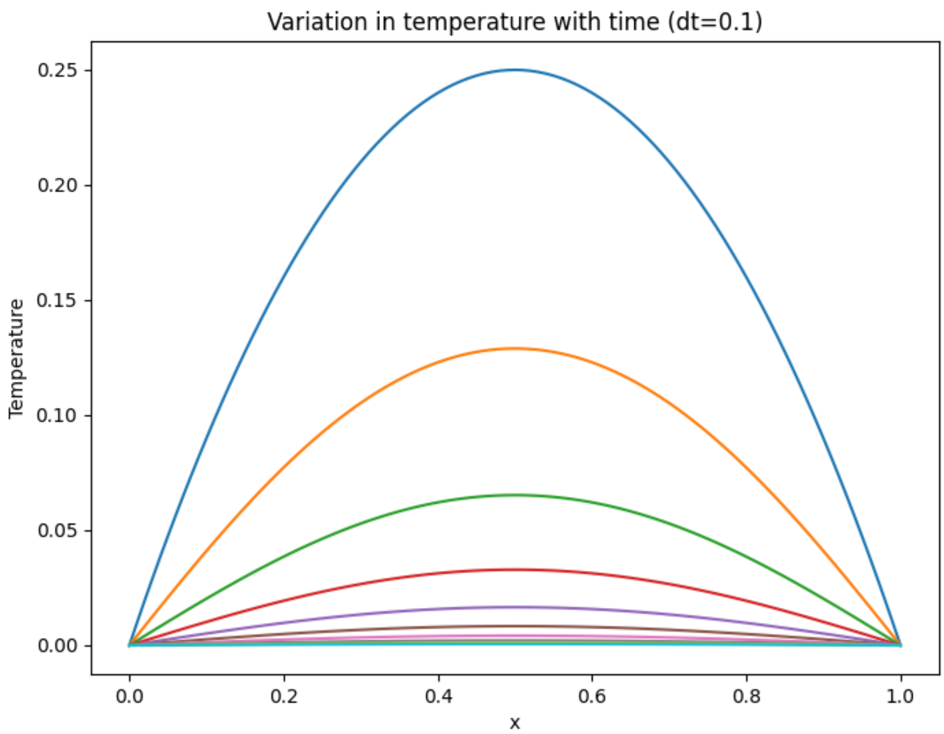

Figure 1: FreeFEM transient solution of the problem. As the time progresses the temperature across the whole domain reaches 0.

```
load "msh3"
real m = 100;
int l=1;
real dt = 0.001;
real T = 1;
meshL Th = segment(m,[x*l]);
func T0 = x - x^2;

ofstream ff("result1.csv");

fespace Vh(Th, P1);
Vh t=T0, v, told;

problem heateqn(t,v)= int1d(Th)(dx(t)*dx(v))
        +int1d(Th)(v*t/dt)
        -int1d(Th)(v*told/dt)
        +on(1,2, t=0); //comment the boundary condition to apply the neumann
     b.c. erquired in the section 1.2

//ff<<"t"<<",";
for(int j=0; j<=m; j++){
    if(j==m){
      ff <<"x"<<j;
    }else{
      ff <<"x"<<j<<",";
    }
  }
ff<<endl;
for(real i=0; i<=T; i+=dt){
  told = t;
  heateqn;
  //ff<<"t"<<i/dt<<",";
  for(int j=0; j<=m; j++){
    if(j==m){
      ff <<told[][j];
    }else{
      ff <<told[][j]<<",";
    }
  }
  ff<<endl;
}

//plot(t, wait=true, value=true);
```

Listing 1: Problem 1 Code

1.1 Comparision with Analytical Solution

The analytical solution for the problem is given as,

$$u(x,t) = \frac{4}{\pi^3}\sum_{n=1}^{\infty}\frac{1-(-1)^n}{n^3}e^{-n^2\pi^2 t}sin(n\pi x) \tag{3}$$

The solution of FreeFEM is compared to the analytical solution at time $t = 0.2s$,

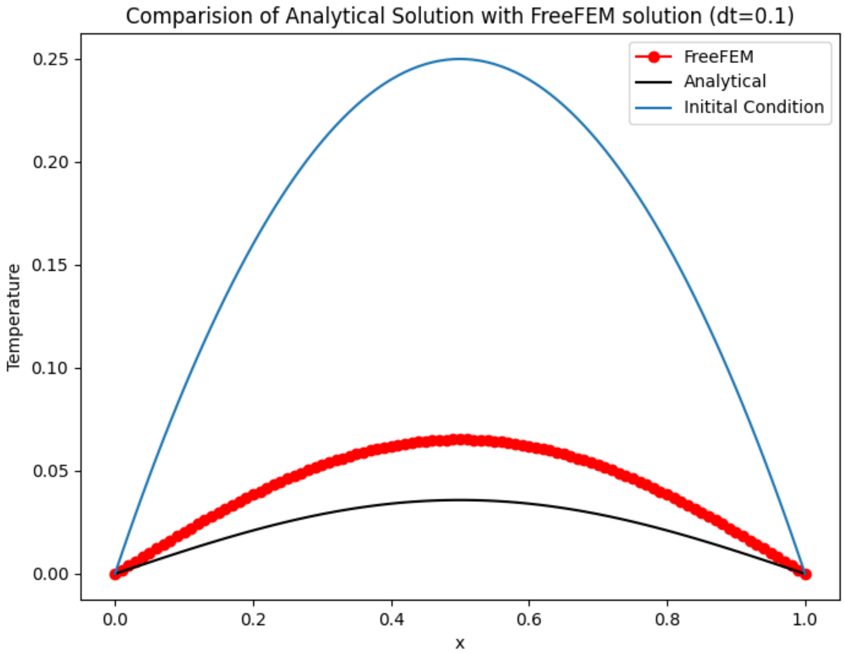

Figure 2: Comparision of the FreeFEM Solution with analytical solution for Problem 1.

1.2 Neumann Boundary Conditions

In the current problem, both the ends are insulated instead of the Dirchlet boundary condition of $U = 0$. This gives the solution of the problem in FreeFEM as,

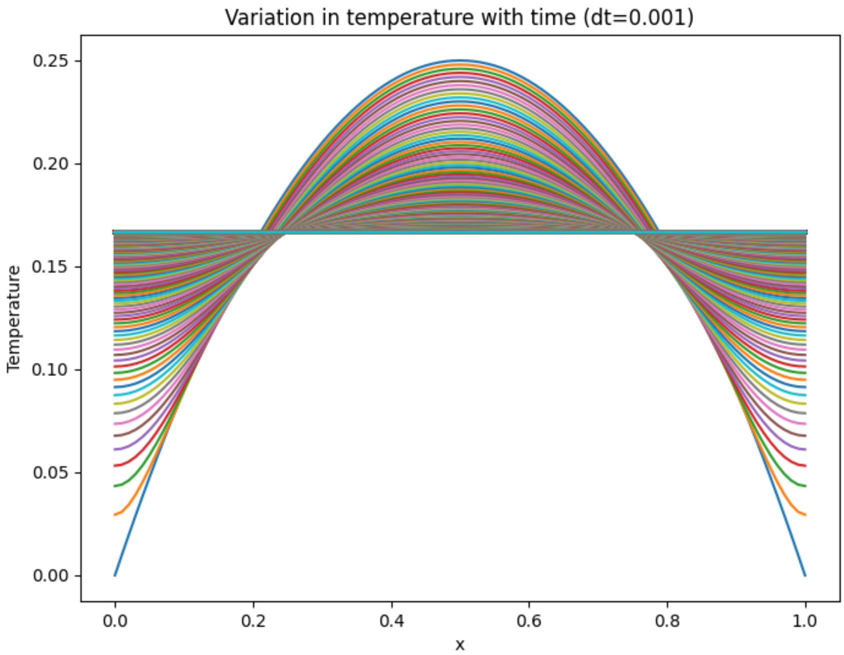

Figure 3: FreeFEM transient solution of the problem. As the time progresses the temperature across the whole domain reaches a constant temperature.

1.3 Time Convergence Study

The time step (dt) used to solve the problem is varied between 0.1, 0.01 and 0.001 while keeping the grid size ($dx = 0.01$) constant. As we reduce dt, the error in the FreeFEM solution reduces significantly as compared to the analytical solution.

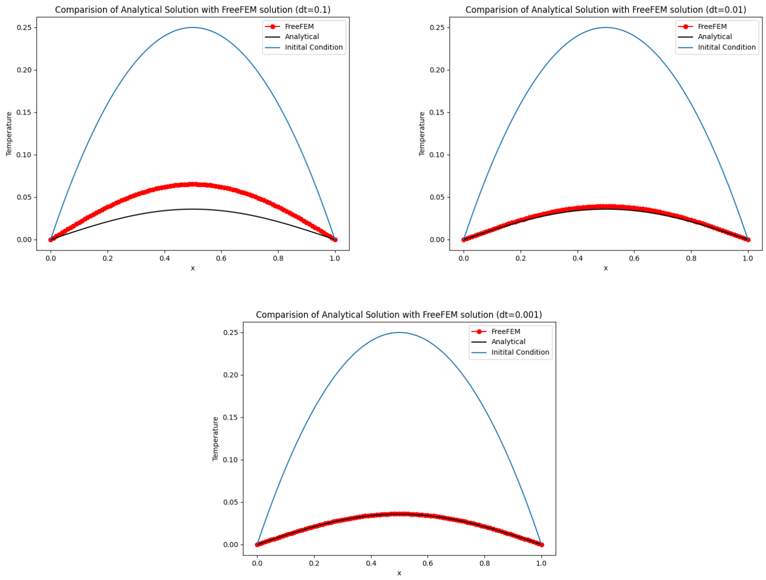

Figure 4: Time step convergence study of FreeFEM Solution for Problem 1.

1.4 Grid Convergence Study

The grid size (dx) used to solve the problem is varied between 0.1, 0.01 and 0.001 while keeping the time step ($dt = 0.001$) constant. No significant change in the error in the FreeFEM solution was observed for the tested grid size as compared to the analytical solution.

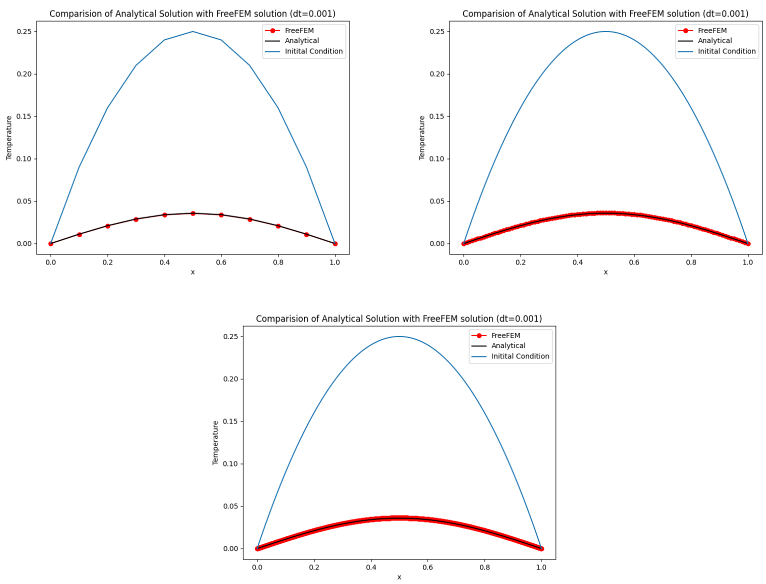

Figure 5: Time step convergence study of FreeFEM Solution for Problem 1.

2 Problem 2

A combination of Dirichlet and Neumann boundary condition is considered for this problem. The governing equation, boundary conditions, and initial conditions are stated below

$$u_t = u_{xx} \tag{4}$$

$$\text{for } 0 < x < 2,\ t > 0$$

$$u(0,x) = \begin{cases} x & x \leq 1 \\ 2 - x & 1 < x \leq 2 \end{cases} \tag{5}$$

$$u(0,t) = u_x(2,t) = 0$$

The problem is solved in FreeFEM++ using the given code for a 1D domain.

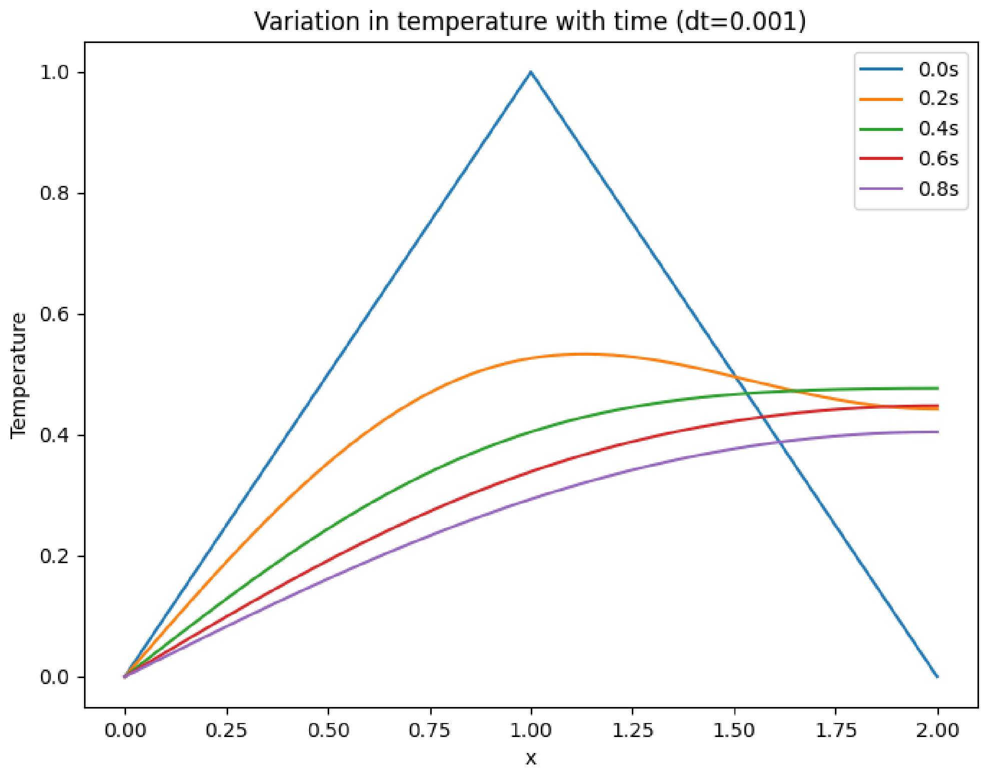

Figure 6: FreeFEM transient solution of the problem. As the time passes the temperature at the right boundary with insulation b.c. increases until it reaches steady state.

```
load "msh3"
real m = 200;
int l=2;
real dt = 0.001;
real T = 1;
meshL Th = segment(m,[x*l]);
func real T0(){
  if(x<=1){
    return x;
  }else{
    return 2-x;
  }
}


ofstream ff("result2.csv");

fespace Vh(Th, P1);
Vh t=T0(), v, told;

problem heateqn(t,v)= int1d(Th)(dx(t)*dx(v))
        +int1d(Th)(v*t/dt)
        -int1d(Th)(v*told/dt)
        +on(1, t=0);


//ff<<"t"<<",";
for(int j=0; j<=m; j++){
    if(j==m){
      ff <<"x"<<j;
    }else{
      ff <<"x"<<j<<",";
    }
  }
ff<<endl;
for(real i=0; i<=T; i+=dt){
  told = t;
  heateqn;
  //ff<<"t"<<i/dt<<",";
  for(int j=0; j<=m; j++){
    if(j==m){
      ff <<told[][j];
    }else{
      ff <<told[][j]<<",";
    }
  }
  ff<<endl;
}

//plot(t, wait=true, value=true);
```

Listing 2: Problem 2 Code

2.1 Comparision with Analytical Solution

The analytical solution for the problem is given as,

$$u(x,t) = \frac{32}{\pi^3}\sum_{n=1}^{\infty} sin\left(\frac{(2n-1)\pi}{4}\right)\frac{1-cos\left(\frac{(2n-1)\pi}{4}\right)}{(2n-1)^2}e^{\frac{(2n-1)^2}{16}\pi^2 t}sin\left(\frac{(2n-1)\pi}{4}x\right) \quad (6)$$

The solution of FreeFEM is compared to the analytical solution at time $t = 0.2s$,

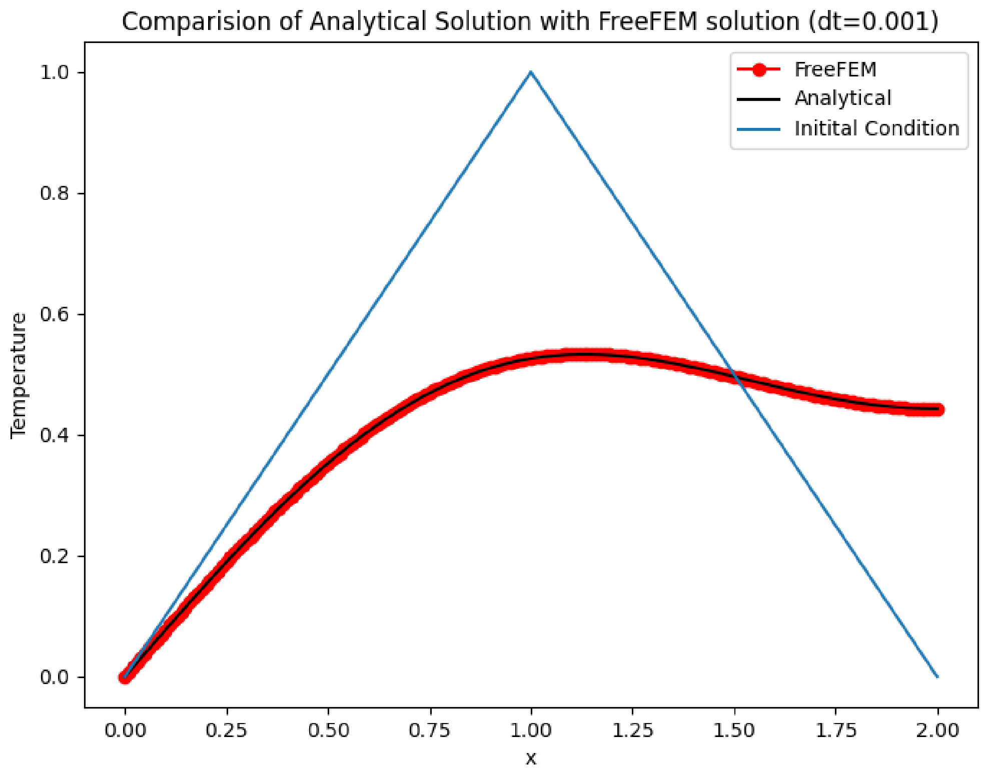

Figure 7: Comparision of the FreeFEM Solution with analytical solution for Problem 2.

3 Problem 3

In practical problems, there can be instances when we encounter non-standard or non-homogeneous boundaries. Here, a non-homogeneous boundary condition problem is discussed,

$$\frac{du}{dt} = \frac{d^2u}{dx^2},\ 0 < x < 3,\ t > 0 \tag{7}$$

$$u(x=0) = 4x - x^2,\ u(0,t) = 0,\ u(3,t) = 3 \tag{8}$$

The problem is solved in FreeFEM++ using the given code for a 1D domain.

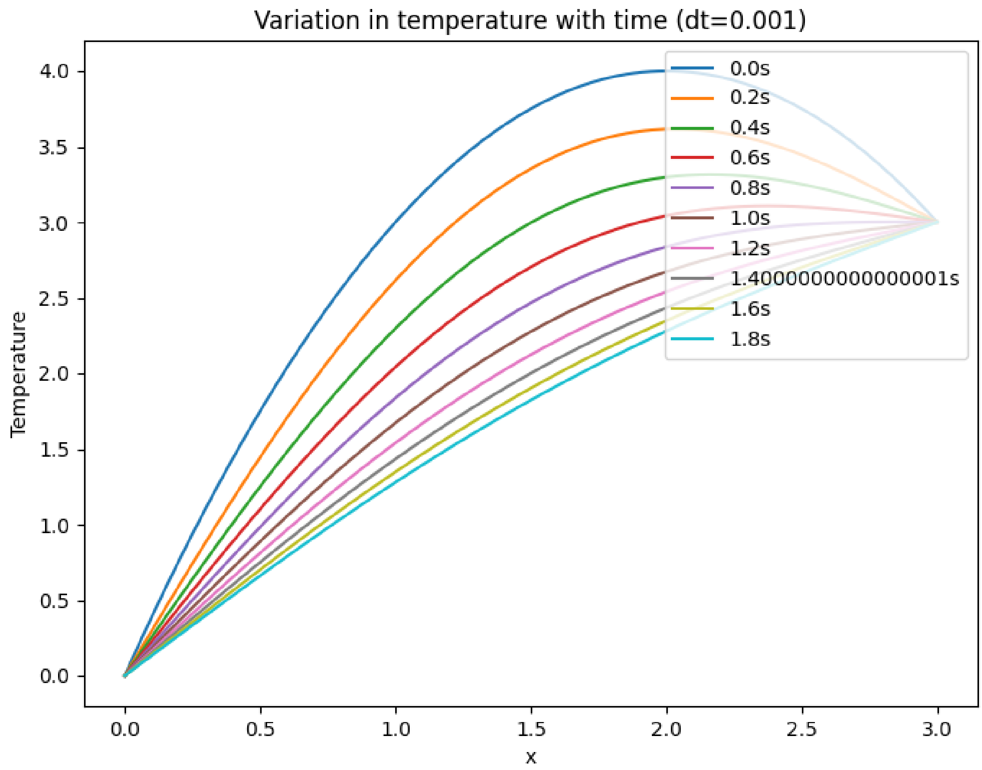

Figure 8: FreeFEM transient solution of the problem. As the time passes the temperature distribution in the domain linearizes and a linear distribution exists at steady state.

```
load "msh3"
real m = 300;
int l=3;
real dt = 0.001;
real T = 2;
meshL Th = segment(m,[x*l]);
func T0 = 4*x - x^2;

ofstream ff("result3.csv");

fespace Vh(Th, P1);
Vh t=T0, v, told;

problem heateqn(t,v)= int1d(Th)(dx(t)*dx(v))
        +int1d(Th)(v*t/dt)
        -int1d(Th)(v*told/dt)
        +on(1, t=0)
        +on(2, t=3);


//ff<<"t"<<",";
for(int j=0; j<=m; j++){
    if(j==m){
      ff <<"x"<<j;
    }else{
      ff <<"x"<<j<<",";
    }
  }
ff<<endl;
for(real i=0; i<=T; i+=dt){
  told = t;
  heateqn;
  //ff<<"t"<<i/dt<<",";
  for(int j=0; j<=m; j++){
    if(j==m){
      ff <<told[][j];
    }else{
      ff <<told[][j]<<",";
    }
  }
  ff<<endl;
}

//plot(t, wait=true, value=true);
```
Listing 3: Problem 3 Code

3.1 Comparision with Analytical Solution

The analytical solution for the problem is given as,

$$u(x,t) = \frac{32}{\pi^3}\sum_{n=1}^{\infty} sin\left(\frac{(2n-1)\pi}{4}\right)\frac{1-cos\left(\frac{(2n-1)\pi}{4}\right)}{(2n-1)^2}e^{\frac{(2n-1)^2}{16}\pi^2 t} sin\left(\frac{(2n-1)\pi}{4}x\right) \tag{9}$$

The solution of FreeFEM is compared to the analytical solution at different times,

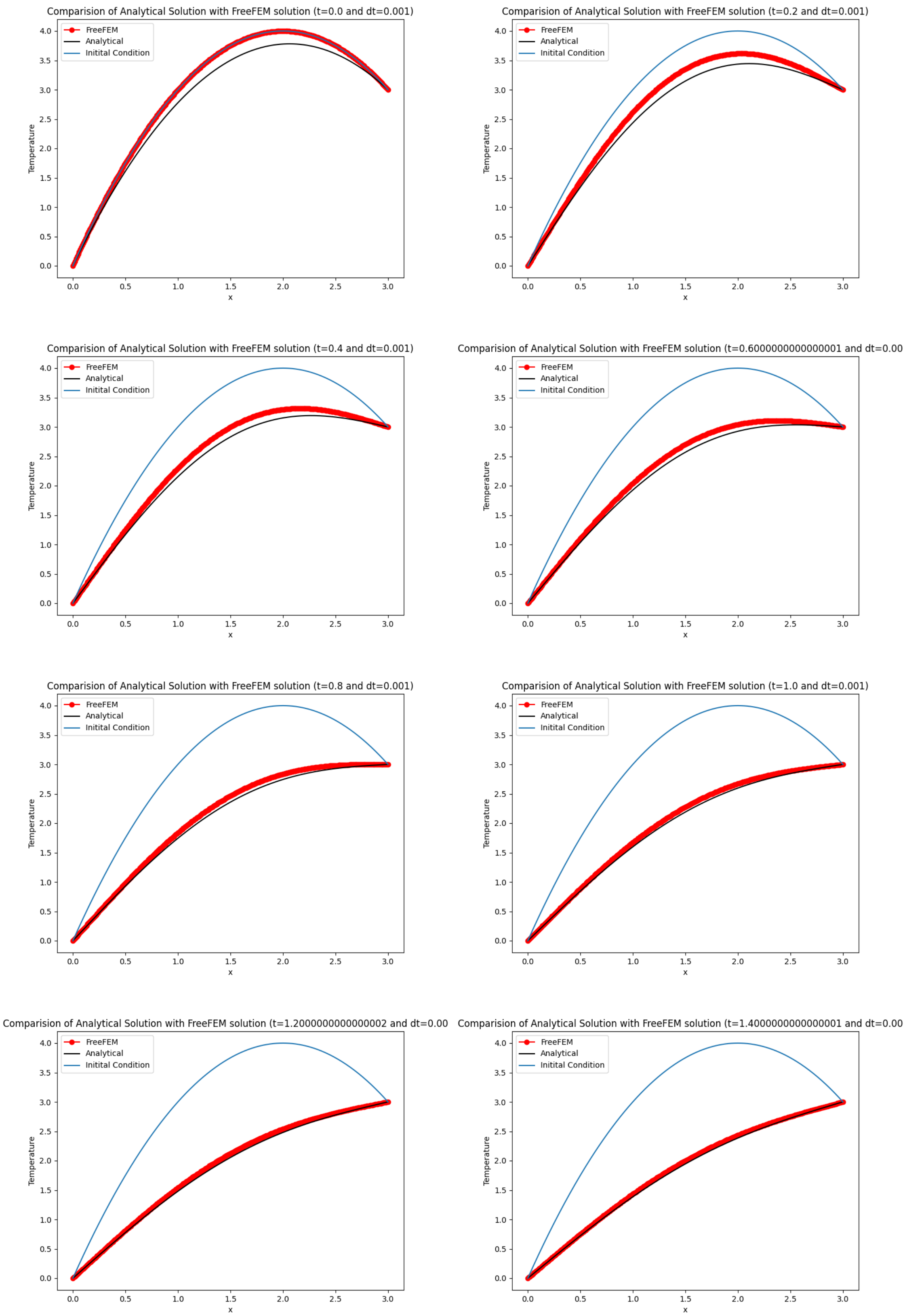

Figure 9: Comparision of the FreeFEM Solution with analytical solution for Problem 3. The error in the FreeFEM solution reduces as time passes and reaches steady state as compared to the analytical solution.

4 Problem 4

A problem similar to the previous one attempted. However in this problem a neumann condition of a constant gradient is given at the left boundary. The governing equations initial conditions and the boundary conditions are summarised below.

$$\frac{du}{dt} = \frac{d^2u}{dx^2},\ 0 < x < 1,\ t > 0 \tag{10}$$

$$u(x=0) = 0,\ u_x(0,t) = -1,\ u_x(1,t) = 0 \tag{11}$$

The problem is solved in FreeFEM++ using the given code in a 2D domain.

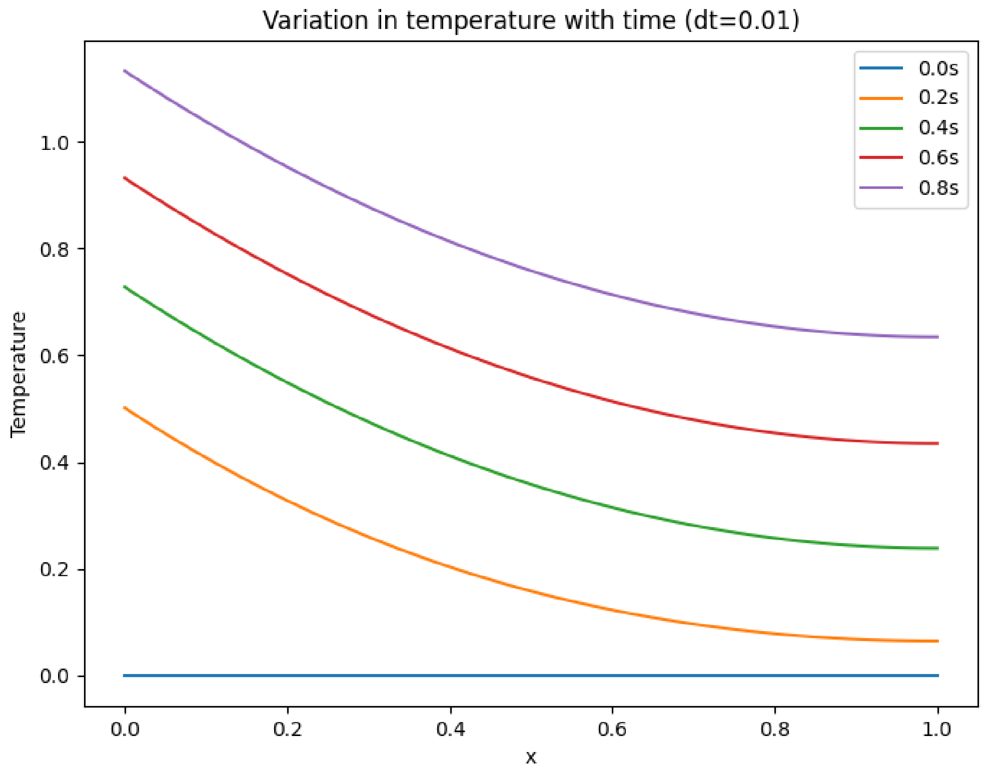

Figure 10: FreeFEM transient solution of the problem. As the time passes the temperature distribution in the domain increases throughout the domain.

```
load "msh3"
real m = 100;
int l=1;
real dt = 0.01;
real T = 2;
mesh Th = square(m,m,[x*l, y*l]);
real T0 = 0;

ofstream ff("result4.csv");

fespace Vh(Th, P1);
Vh t=T0, v, told;

problem heateqn(t,v)= int2d(Th)(dx(t)*dx(v))
        +int2d(Th)(v*t/dt)
        -int2d(Th)(v*told/dt)
        +int1d(Th,4)(v*(-1));


//ff<<"t"<<",";
for(int j=0; j<=m; j++){
    if(j==m){
      ff <<"x"<<j;
    }else{
      ff <<"x"<<j<<",";
    }
  }
ff<<endl;
for(real i=0; i<=T; i+=dt){
  told = t;
  heateqn;
  //ff<<"t"<<i/dt<<",";
  for(int j=0; j<=m; j++){
    if(j==m){
      ff <<told[][j];
    }else{
      ff <<told[][j]<<",";
    }
  }
  ff<<endl;
}

//plot(t, wait=true, value=true);
```
Listing 4: Problem 4 Code

4.1 Comparision with Analytical Solution

The analytical solution for the problem is given as,

$$u(x,t) = \frac{1}{2}(x^2 + 2t) - x + \frac{1}{3} - \frac{2}{\pi^2}\sum_{n=1}^{\infty}\frac{1}{n^2}e^{-n^2\pi^2 t}cos(n\pi x) \tag{12}$$

The solution of FreeFEM is compared to the analytical solution at different times,

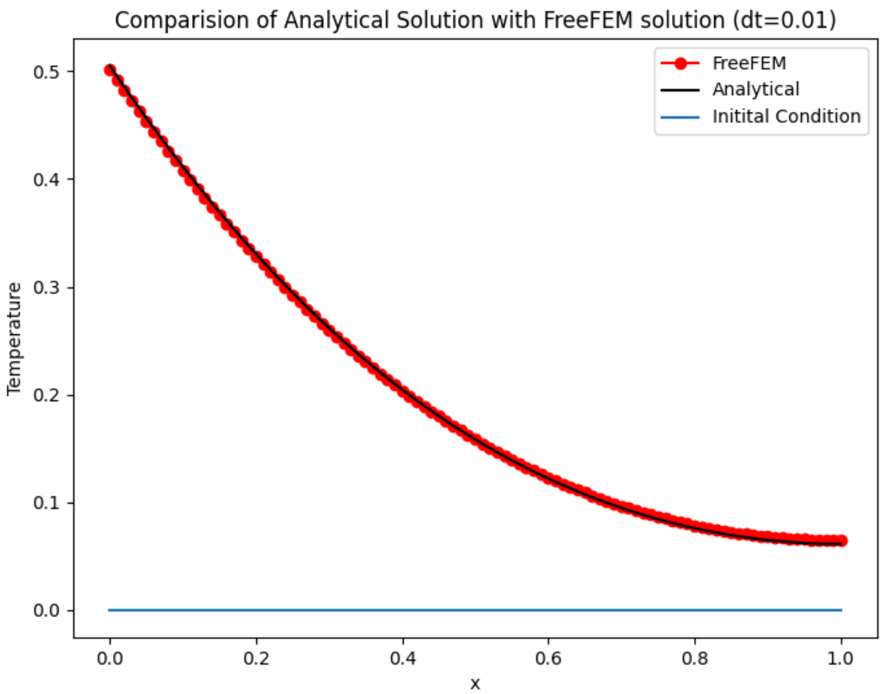

Figure 11: Comparision of the FreeFEM Solution with analytical solution for Problem 4.

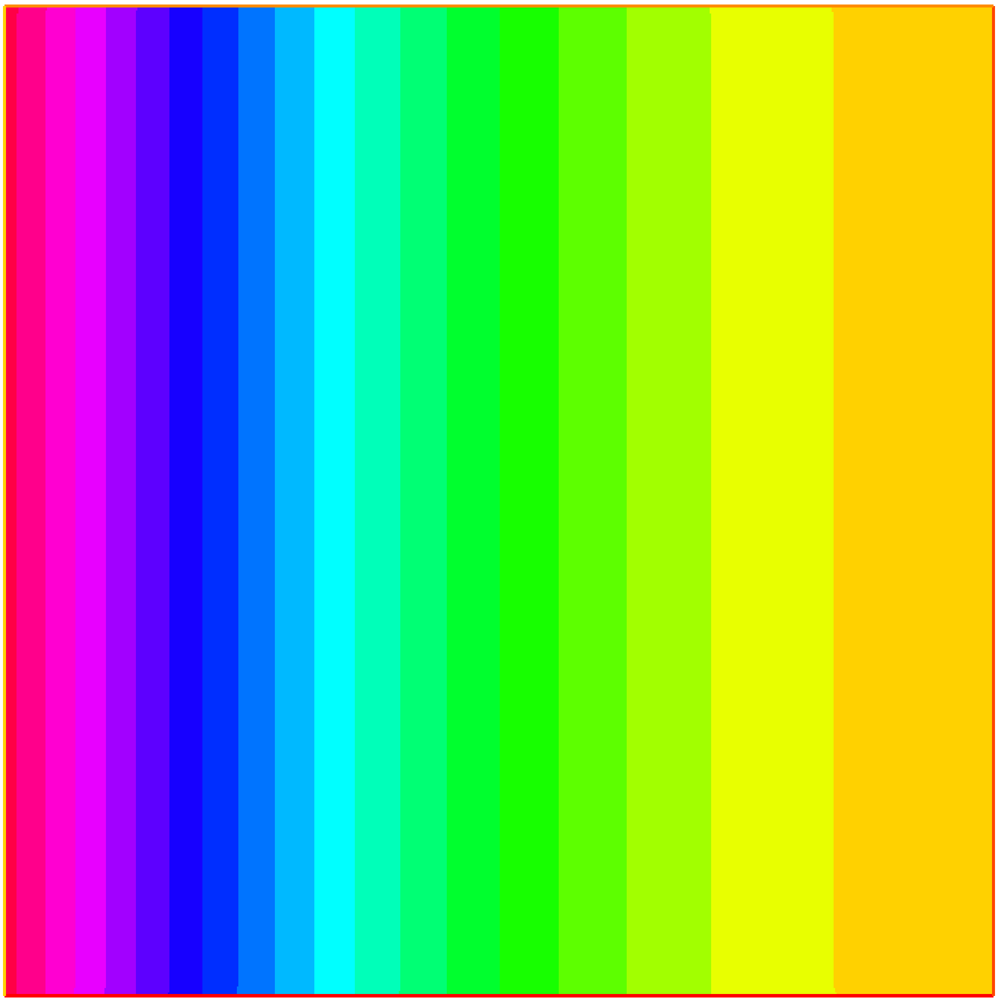

Figure 12: FreeFEM transient solution of the problem at time t=0.18s