FLORIDA INSTITUTE OF TECHNOLOGY
MECHANICAL AND AEROSPACE ENGINEERING DEPARTMENT

MAE 5150-E1: Computational Fluid Dynamics
Fall 2017

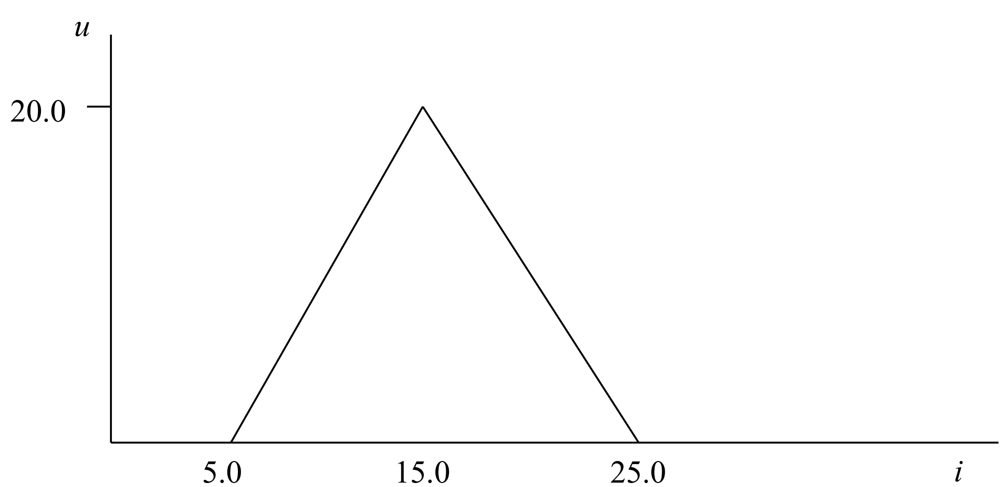
Max Le
Coding Project 3

Due October 31, 2017

1. A wave is propagating in a closed-end tube. Compute the wave propagation up to $t = 0.15$ sec. by solving the first-order wave equation. Assume the speed of sound to be 200 m/s. The wave has a triangular shape (see figure) which is to be used as the initial condition at $t = 0.0$. You will need to determine the initial u values on the slopes of the triangle and specify them in your code. Solve the problem by the following methods:

FTBS Explicit
Lax-Wendroff
Euler's BTCS Implicit

Use a $\Delta x = 1.0$ and $IM = 71$ $(1 \le i \le 71)$. For each case, use a $\Delta t = 0.005, 0.0025, 0.00125$ and compare solutions. Run the solution to $t = 0.15$ sec. Print and plot the solution at the initial condition, the end condition, and at the half-way point.



Assigned 10/19/2017

2. Use MacCormack's explicit method to solve the Burgers equation. The initial condition is specified as

$$u(x,0) = 5.0 \qquad 0.0 \le x \le 20.0$$

$$u(x,0) = 0.0 \qquad 20.0 < x \le 40.0$$

Print the solution at intervals of 0.4 seconds up to t = 2.4 sec. Run the code twice, once for each of the following conditions:

$$\Delta x = 1.0 \qquad \Delta t = 0.1$$

$$\Delta x = 1.0 \qquad \Delta t = 0.2$$

Plot each solution at each print interval.

```
//FTBS EXPLICIT

#include <iostream>
#include <stdio.h>
#include <cmath>
#include <fstream>
#include <vector>
using namespace std;

//DECLARE STRUCTURE

struct Hyperbolic
{
    double dx;
    double dt;
    int imax;
    double tmax; //max time
    double nmax; //max time steps
    double alpha; //speed of sound
    double c;
    vector <double> u0; //initial u //USED IN LOOPS
    vector <double> u; //u at n+1 level, USED IN LOOPS
    vector <double> uhalf; //half way point, for print
    vector <double> ufull; //end solution, for print
    vector <double> uInitial; //initial sol, for print

};

void InitializeU(struct Hyperbolic *wave){

    //BASIC PROPERTIES
    wave->dx = 1.0;
    wave->imax = 71;
    wave->alpha = 200.00;
    wave->tmax = 0.15;
    wave->nmax = (wave->tmax/wave->dt);
    wave->c = (wave->alpha*wave->dt)/(wave->dx);

    //SETTING U AND UN
    wave->u0.resize(wave->imax+1);
    wave->u.resize(wave->imax+1);
    wave->uhalf.resize(wave->imax+1);
    wave->ufull.resize(wave->imax+1);
    wave->uInitial.resize(wave->imax+1);



    //SETTING UP INITAL CONDITIONS, U0
    for (int i = 1; i<=wave->imax;i++){

        if (i == 15){
            wave->u0[i]= 20.00;
```

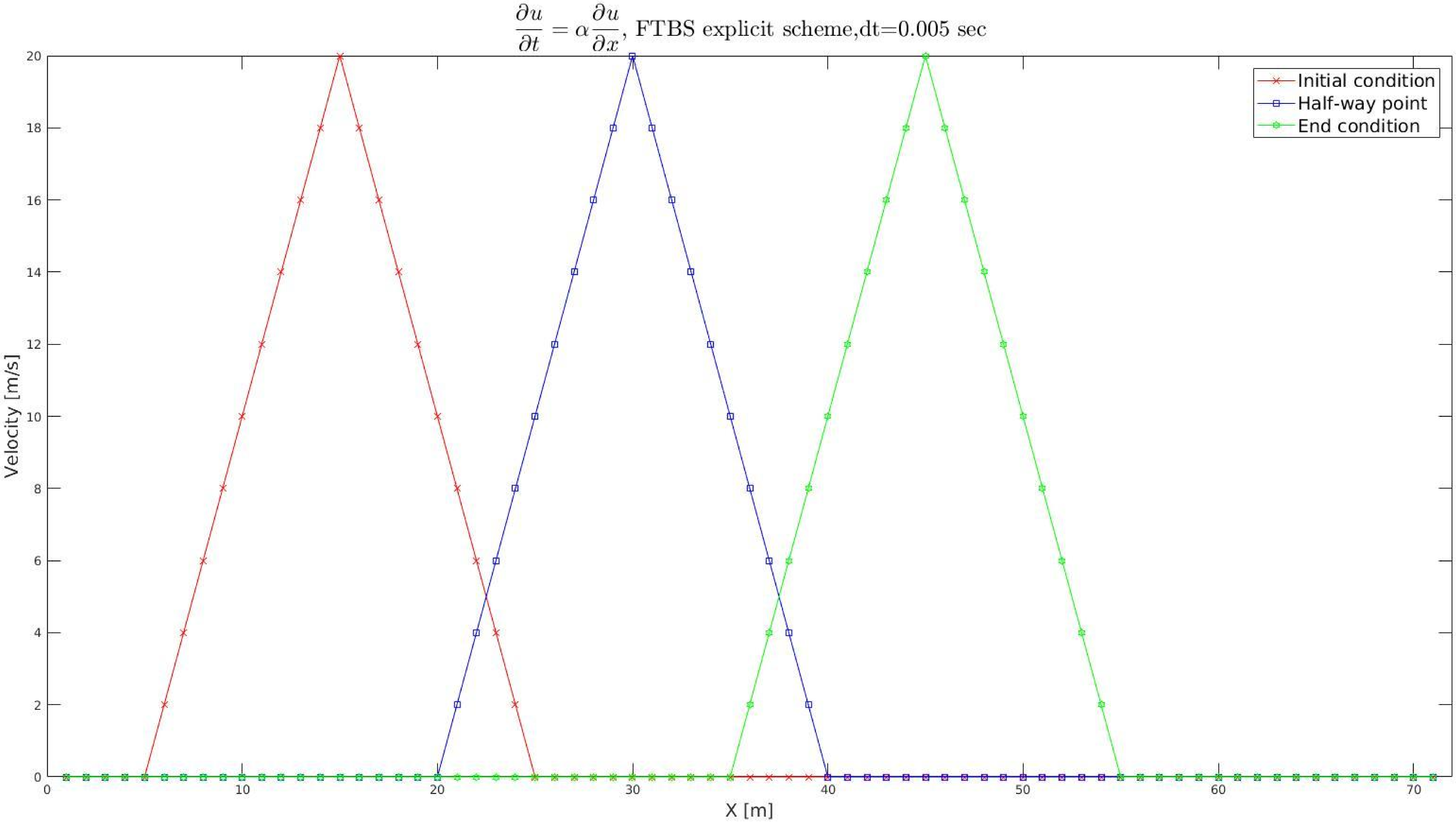
```
        }

        else if (i>=5 && i<15){
            wave->u0[i] = (2*i)-10;
        }
        else if (i>15 && i<=25){
            wave->u0[i] = (-2*i) + 50;
        }
    }
}


void FTBSExplicit(struct Hyperbolic *wave){

    //Finite Difference Loop

    //Time loop, FULL TIME
    for (int n = 0; n<=wave->nmax;n++){
        //copy loop
        for (int i = 1; i<=wave->imax;i++){
            wave->u[i] = wave->u0[i];
        }

        //FDE
        for (int i = 2; i<=wave->imax;i++){
            wave->u0[i] = wave->u[i] - (wave->c)*(wave->u[i]-wave->u
[i-1]);
        }

        if (n == 0){
            for (int i = 1; i<=wave->imax;i++){
                wave->uInitial[i] = wave->u[i];
            }

        }

        else if (n == (wave->nmax/2)){
            for (int i = 1; i<=wave->imax;i++){
                wave->uhalf[i] = wave->u[i];
            }
        }
        else if (n == wave->nmax){
            for (int i = 1; i<=wave->imax;i++){
                wave->ufull[i] = wave->u[i];
            }
        }
    }

    if (wave->dt == 0.005){
        FILE* outfile1;
        outfile1 = fopen("005FTBS.dat","w");
        fprintf(outfile1,"Initial Condition       Halfway Point        Final
```

```
Condition\n");
        fprintf
(outfile1,"---------------------------------------------------------------\n");
        for (int i = 1; i<=wave->imax;i++){
            // printf("%f\t\t%f\t\t%f\n",wave.uInitial[i],wave.uhalf
[i],wave.ufull[i]);
            fprintf(outfile1,"%10.7f\t\t\t\t%10.7f\t\t\t\t%10.7f\n",wave-
>uInitial[i],wave->uhalf[i],wave->ufull[i]);
        }
    }


    else if (wave->dt == 0.0025){
        FILE* outfile2;
        outfile2 = fopen("0025FTBS.dat","w");
        fprintf(outfile2,"Initial Condition       Halfway Point        Final
Condition\n");
        fprintf
(outfile2,"---------------------------------------------------------------\n");
        for (int i = 1; i<=wave->imax;i++){
            // printf("%f\t\t%f\t\t%f\n",wave.uInitial[i],wave.uhalf
[i],wave.ufull[i]);
            fprintf(outfile2,"%10.7f\t\t\t\t%10.7f\t\t\t\t%10.7f\n",wave-
>uInitial[i],wave->uhalf[i],wave->ufull[i]);
        }
    }

    else if (wave->dt == 0.00125){
        FILE* outfile3;
        outfile3 = fopen("0125FTBS.dat","w");
        fprintf(outfile3,"Initial Condition       Halfway Point        Final
Condition\n");
        fprintf
(outfile3,"---------------------------------------------------------------\n");
        for (int i = 1; i<=wave->imax;i++){
            printf("%f\t\t%f\t\t%f\n",wave->uInitial[i],wave->uhalf[i],wave-
>ufull[i]);
            fprintf(outfile3,"%10.7f\t\t\t\t%10.7f\t\t\t\t%10.7f\n",wave-
>uInitial[i],wave->uhalf[i],wave->ufull[i]);
        }
    }
}


void runFTBSCode(){
    Hyperbolic wave;
    wave.dt = 0.005;
    InitializeU(&wave);
    FTBSExplicit(&wave);
}
```

```
int main(){
    runFTBSCode();

}
```

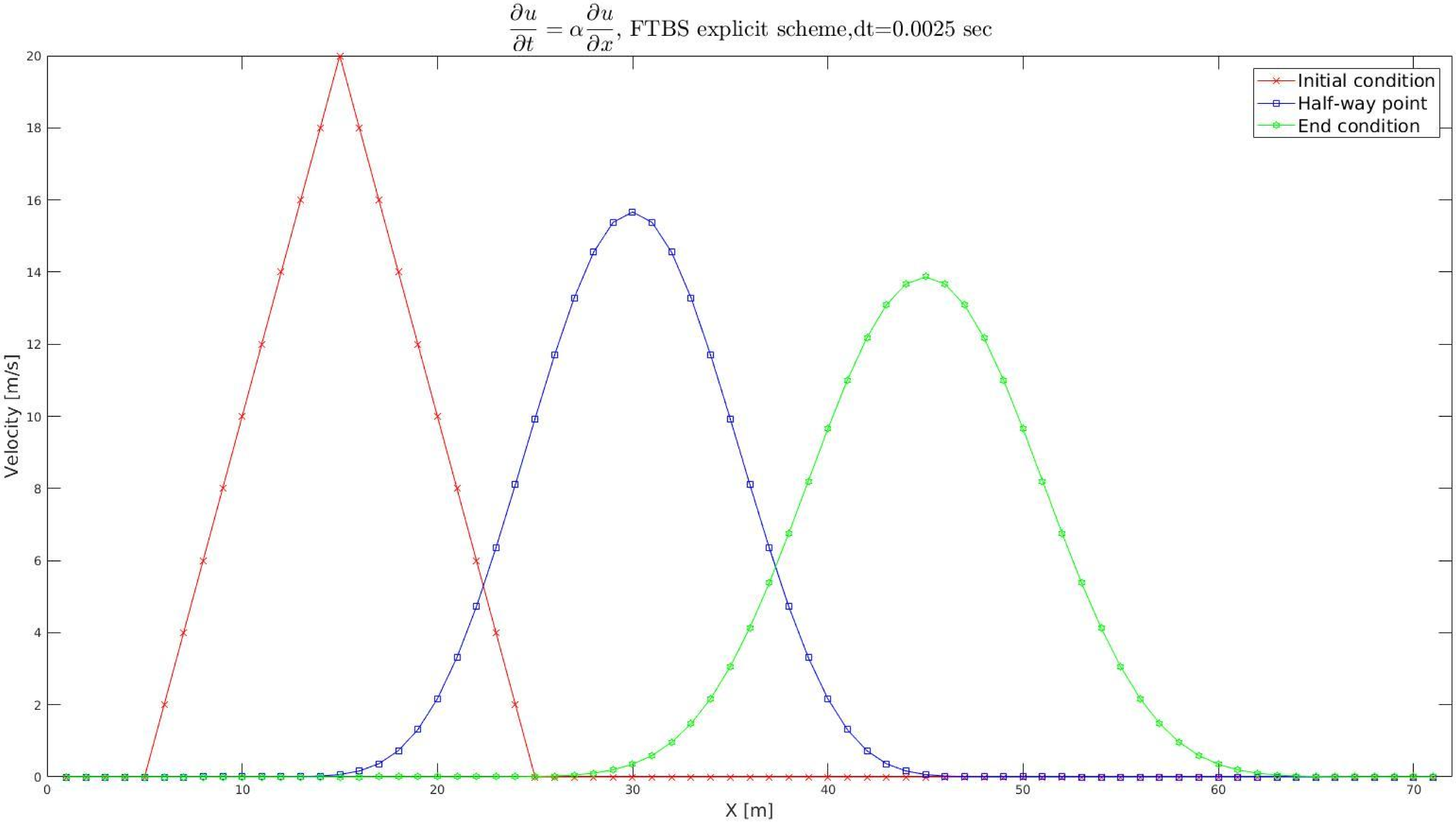
Initial Condition	Halfway Point	Final Condition
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
2.0000000	0.0000000	0.0000000
4.0000000	0.0000000	0.0000000
6.0000000	0.0000000	0.0000000
8.0000000	0.0000000	0.0000000
10.0000000	0.0000000	0.0000000
12.0000000	0.0000000	0.0000000
14.0000000	0.0000000	0.0000000
16.0000000	0.0000000	0.0000000
18.0000000	0.0000000	0.0000000
20.0000000	0.0000000	0.0000000
18.0000000	0.0000000	0.0000000
16.0000000	0.0000000	0.0000000
14.0000000	0.0000000	0.0000000
12.0000000	0.0000000	0.0000000
10.0000000	0.0000000	0.0000000
8.0000000	2.0000000	0.0000000
6.0000000	4.0000000	0.0000000
4.0000000	6.0000000	0.0000000
2.0000000	8.0000000	0.0000000
0.0000000	10.0000000	0.0000000
0.0000000	12.0000000	0.0000000
0.0000000	14.0000000	0.0000000
0.0000000	16.0000000	0.0000000
0.0000000	18.0000000	0.0000000
0.0000000	20.0000000	0.0000000
0.0000000	18.0000000	0.0000000
0.0000000	16.0000000	0.0000000
0.0000000	14.0000000	0.0000000
0.0000000	12.0000000	0.0000000
0.0000000	10.0000000	0.0000000
0.0000000	8.0000000	2.0000000
0.0000000	6.0000000	4.0000000
0.0000000	4.0000000	6.0000000
0.0000000	2.0000000	8.0000000
0.0000000	0.0000000	10.0000000
0.0000000	0.0000000	12.0000000
0.0000000	0.0000000	14.0000000
0.0000000	0.0000000	16.0000000
0.0000000	0.0000000	18.0000000
0.0000000	0.0000000	20.0000000
0.0000000	0.0000000	18.0000000
0.0000000	0.0000000	16.0000000
0.0000000	0.0000000	14.0000000
0.0000000	0.0000000	12.0000000
0.0000000	0.0000000	10.0000000

0.0000000	0.0000000	8.0000000
0.0000000	0.0000000	6.0000000
0.0000000	0.0000000	4.0000000
0.0000000	0.0000000	2.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000


$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x}$, FTBS explicit scheme,dt=0.005 sec
Initial condition
Half-way point
End condition
Velocity [m/s]
X [m]
0
10
20
30
40
50
60
70
0
2
4
6
8
10
12
14
16
18
20

Initial Condition	Halfway Point	Final Condition
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
2.0000000	0.0000000	0.0000000
4.0000000	0.0000001	0.0000000
6.0000000	0.0000009	0.0000000
8.0000000	0.0000094	0.0000000
10.0000000	0.0000688	0.0000000
12.0000000	0.0003937	0.0000000
14.0000000	0.0018247	0.0000000
16.0000000	0.0070475	0.0000000
18.0000000	0.0231723	0.0000000
20.0000000	0.0659463	0.0000000
18.0000000	0.1646834	0.0000002
16.0000000	0.3651717	0.0000010
14.0000000	0.7267646	0.0000041
12.0000000	1.3114125	0.0000163
10.0000000	2.1668291	0.0000585
8.0000000	3.3106437	0.0001930
6.0000000	4.7231171	0.0005871
4.0000000	6.3510768	0.0016528
2.0000000	8.1183388	0.0043233
0.0000000	9.9340537	0.0105408
0.0000000	11.6938055	0.0240298
0.0000000	13.2767039	0.0513684
0.0000000	14.5482927	0.1032559
0.0000000	15.3775408	0.1956926
0.0000000	15.6662042	0.3506100
0.0000000	15.3775408	0.5954019
0.0000000	14.5482927	0.9609078
0.0000000	13.2767039	1.4777342
0.0000000	11.6938055	2.1712768
0.0000000	9.9340537	3.0562635
0.0000000	8.1183388	4.1318632
0.0000000	6.3510768	5.3783003
0.0000000	4.7231171	6.7555577
0.0000000	3.3106437	8.2043538
0.0000000	2.1668291	9.6493314
0.0000000	1.3114125	11.0043424
0.0000000	0.7267646	12.1796874
0.0000000	0.3651717	13.0909433
0.0000000	0.1646834	13.6685067
0.0000000	0.0659463	13.8663914
0.0000000	0.0231723	13.6685067
0.0000000	0.0070475	13.0909433
0.0000000	0.0018247	12.1796874
0.0000000	0.0003937	11.0043424
0.0000000	0.0000688	9.6493314

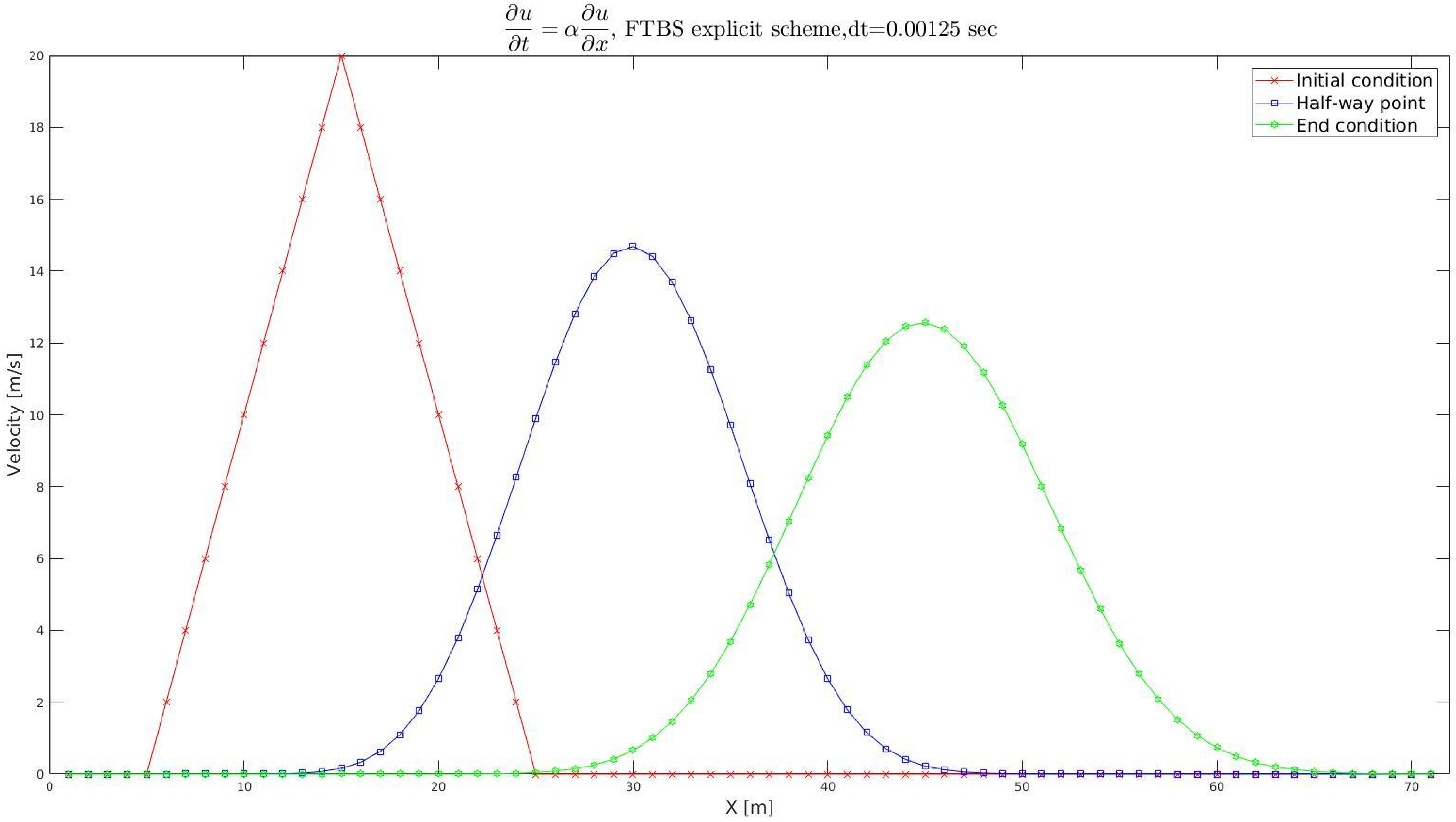
0.0000000	0.0000094	8.2043538
0.0000000	0.0000009	6.7555577
0.0000000	0.0000001	5.3783003
0.0000000	0.0000000	4.1318632
0.0000000	0.0000000	3.0562635
0.0000000	0.0000000	2.1712768
0.0000000	0.0000000	1.4777342
0.0000000	0.0000000	0.9609078
0.0000000	0.0000000	0.5954019
0.0000000	0.0000000	0.3506100
0.0000000	0.0000000	0.1956926
0.0000000	0.0000000	0.1032559
0.0000000	0.0000000	0.0513684
0.0000000	0.0000000	0.0240298
0.0000000	0.0000000	0.0105408
0.0000000	0.0000000	0.0043233
0.0000000	0.0000000	0.0016528
0.0000000	0.0000000	0.0005871
0.0000000	0.0000000	0.0001930
0.0000000	0.0000000	0.0000585
0.0000000	0.0000000	0.0000163



$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x}$, FTBS explicit scheme,dt=0.0025 sec
Initial condition
Half-way point
End condition
Velocity [m/s]
X [m]
0
2
4
6
8
10
12
14
16
18
20
0
10
20
30
40
50
60
70

Initial Condition	Halfway Point	Final Condition
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
2.0000000	0.0000001	0.0000000
4.0000000	0.0000014	0.0000000
6.0000000	0.0000153	0.0000000
8.0000000	0.0001100	0.0000000
10.0000000	0.0005887	0.0000000
12.0000000	0.0025010	0.0000000
14.0000000	0.0087935	0.0000000
16.0000000	0.0263497	0.0000001
18.0000000	0.0687796	0.0000004
20.0000000	0.1591145	0.0000018
18.0000000	0.3308880	0.0000072
16.0000000	0.6260506	0.0000260
14.0000000	1.0891381	0.0000850
12.0000000	1.7587715	0.0002559
10.0000000	2.6589527	0.0007114
8.0000000	3.7927227	0.0018376
6.0000000	5.1394097	0.0044306
4.0000000	6.6548206	0.0100151
2.0000000	8.2727767	0.0213049
0.0000000	9.9071382	0.0428032
0.0000000	11.4553033	0.0814814
0.0000000	12.8053722	0.1474153
0.0000000	13.8483229	0.2541919
0.0000000	14.4937831	0.4188683
0.0000000	14.6851378	0.6612802
0.0000000	14.4089223	1.0025812
0.0000000	13.6954857	1.4630236
0.0000000	12.6115464	2.0591508
0.0000000	11.2481849	2.8007129
0.0000000	9.7084035	3.6877152
0.0000000	8.0967361	4.7080493
0.0000000	6.5110999	5.8361257
0.0000000	5.0358792	7.0328552
0.0000000	3.7356793	8.2471919
0.0000000	2.6505115	9.4192775
0.0000000	1.7939834	10.4850052
0.0000000	1.1556255	11.3815837
0.0000000	0.7070466	12.0534660
0.0000000	0.4101820	12.4578770
0.0000000	0.2253202	12.5691729
0.0000000	0.1170659	12.3814168
0.0000000	0.0574740	11.9088220
0.0000000	0.0266441	11.1840474
0.0000000	0.0116560	10.2546382
0.0000000	0.0048093	9.1781426

0.0000000	0.0018706	8.0165652
0.0000000	0.0006856	6.8308376
0.0000000	0.0002367	5.6759250
0.0000000	0.0000769	4.5970522
0.0000000	0.0000235	3.6273590
0.0000000	0.0000068	2.7870874
0.0000000	0.0000018	2.0842006
0.0000000	0.0000005	1.5161518
0.0000000	0.0000001	1.0723973
0.0000000	0.0000000	0.7372051
0.0000000	0.0000000	0.4923416
0.0000000	0.0000000	0.3193259
0.0000000	0.0000000	0.2010727
0.0000000	0.0000000	0.1228862
0.0000000	0.0000000	0.0728753
0.0000000	0.0000000	0.0419273
0.0000000	0.0000000	0.0233979
0.0000000	0.0000000	0.0126637
0.0000000	0.0000000	0.0066465
0.0000000	0.0000000	0.0033825
0.0000000	0.0000000	0.0016690



$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x}$, FTBS explicit scheme,dt=0.00125 sec
Initial condition
Half-way point
End condition
Velocity [m/s]
X [m]
0
2
4
6
8
10
12
14
16
18
20
0
10
20
30
40
50
60
70

```cpp
//LAX WENDROFF

#include <iostream>
#include <stdio.h>
#include <cmath>
#include <fstream>
#include <vector>
using namespace std;

//DECLARE STRUCTURE

struct Hyperbolic
{
    double dx;
    double dt;
    int imax;
    double tmax; //max time
    double nmax; //max time steps
    double alpha; //speed of sound
    double c;
    double c_square;
    vector <double> u0; //initial u //USED IN LOOPS
    vector <double> u; //u at n+1 level, USED IN LOOPS
    vector <double> uhalf; //half way point, for print
    vector <double> ufull; //end solution, for print
    vector <double> uInitial; //initial sol, for print

};

void InitializeU(struct Hyperbolic *wave){

    //BASIC PROPERTIES
    wave->dx = 1.0;
    wave->imax = 71;
    wave->alpha = 200.00;
    wave->tmax = 0.15;
    wave->nmax = (wave->tmax/wave->dt);
    wave->c = (wave->alpha*wave->dt)/(wave->dx);
    wave->c_square = pow(wave->c,2.);

    //SETTING U AND UN
    wave->u0.resize(wave->imax+1);
    wave->u.resize(wave->imax+1);
    wave->uhalf.resize(wave->imax+1);
    wave->ufull.resize(wave->imax+1);
    wave->uInitial.resize(wave->imax+1);



    //SETTING UP INITAL CONDITIONS, U0
    for (int i = 1; i<=wave->imax;i++){
```

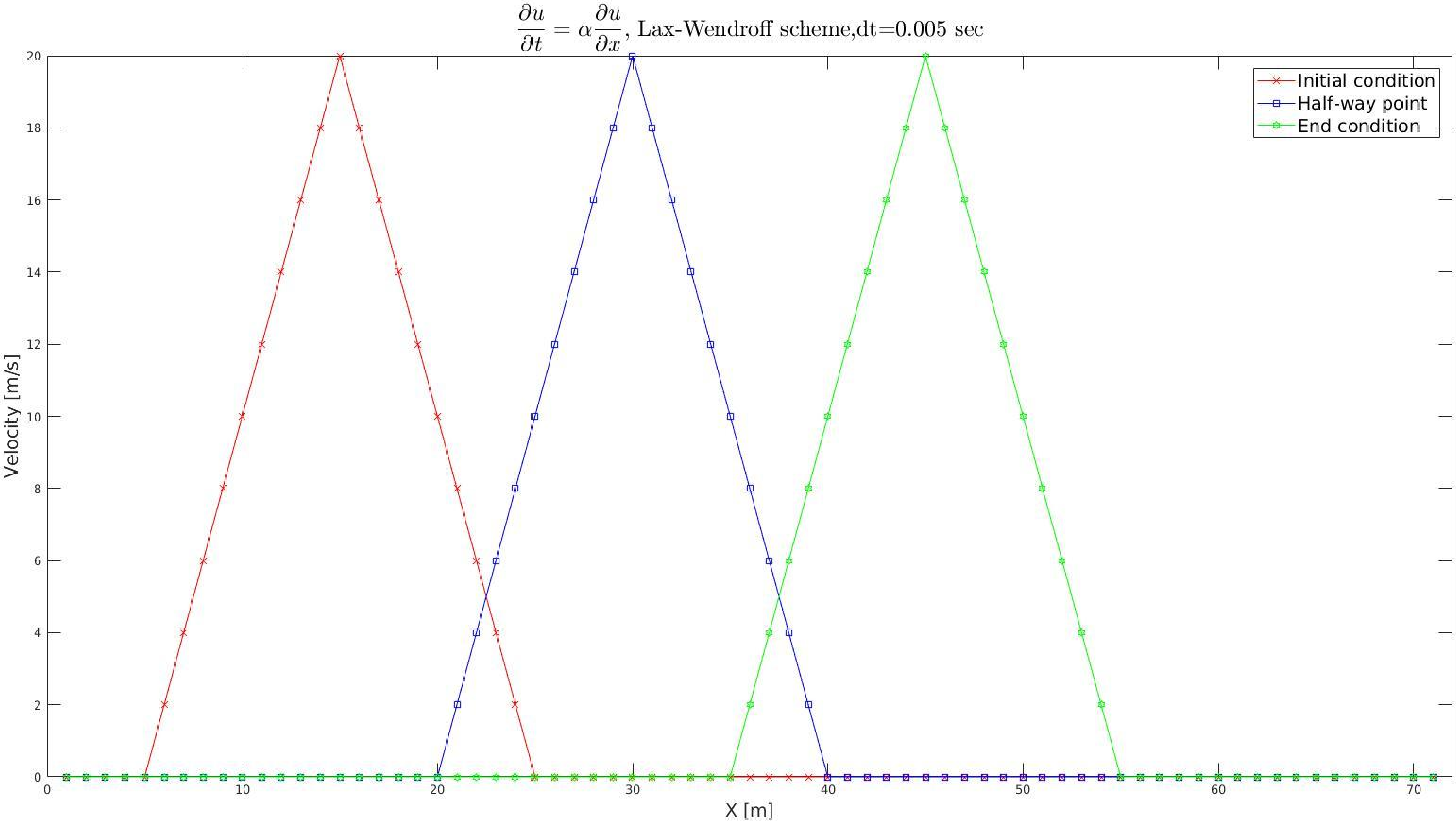
```
        if (i == 15){
            wave->u0[i]= 20.00;
        }

        else if (i>=5 && i<15){
            wave->u0[i] = (2*i)-10;
        }
        else if (i>15 && i<=25){
            wave->u0[i] = (-2*i) + 50;
        }
    }
}


void LaxWendroff(struct Hyperbolic *wave){

    //Finite Difference Loop

    //Time loop, FULL TIME
    for (int n = 0; n<=wave->nmax;n++){
        //copy loop
        for (int i = 1; i<=wave->imax;i++){
            wave->u[i] = wave->u0[i];
        }

        //FDE
        for (int i = 2; i<=wave->imax;i++){
            wave->u0[i] = wave->u[i] - ((wave->c/2)*(wave->u[i+1]-wave->u
[i-1]))+((wave->c_square/2)*(wave->u[i+1]-(2*wave->u[i])+wave->u
[i-1]));
        }

        if (n == 0){
            for (int i = 1; i<=wave->imax;i++){
                wave->uInitial[i] = wave->u[i];
            }

        }

        else if (n == (wave->nmax/2)){
            for (int i = 1; i<=wave->imax;i++){
                wave->uhalf[i] = wave->u[i];
            }
        }
        else if (n == wave->nmax){
            for (int i = 1; i<=wave->imax;i++){
                wave->ufull[i] = wave->u[i];
            }
        }
    }

    if (wave->dt == 0.005){
```

```
        FILE* outfile1;
        outfile1 = fopen("005Lax.dat","w");
        fprintf(outfile1,"Initial Condition       Halfway Point        Final 
Condition\n");
        fprintf
(outfile1,"---------------------------------------------------------------\n");
        for (int i = 1; i<=wave->imax;i++){
            // printf("%f\t\t%f\t\t%f\n",wave.uInitial[i],wave.uhalf
[i],wave.ufull[i]);
            fprintf(outfile1,"%10.7f\t\t\t\t%10.7f\t\t\t\t%10.7f\n",wave-
>uInitial[i],wave->uhalf[i],wave->ufull[i]);
        }
    }
    else if (wave->dt == 0.0025){
        FILE* outfile2;
        outfile2 = fopen("0025Lax.dat","w");
        fprintf(outfile2,"Initial Condition       Halfway Point        Final 
Condition\n");
        fprintf
(outfile2,"---------------------------------------------------------------\n");
        for (int i = 1; i<=wave->imax;i++){
            // printf("%f\t\t%f\t\t%f\n",wave.uInitial[i],wave.uhalf
[i],wave.ufull[i]);
            fprintf(outfile2,"%10.7f\t\t\t\t%10.7f\t\t\t\t%10.7f\n",wave-
>uInitial[i],wave->uhalf[i],wave->ufull[i]);
        }
    }

    else if (wave->dt == 0.00125){

        FILE* outfile3;
        outfile3 = fopen("0125Lax.dat","w");
        fprintf(outfile3,"Initial Condition       Halfway Point        Final 
Condition\n");
        fprintf
(outfile3,"---------------------------------------------------------------\n");
        for (int i = 1; i<=wave->imax;i++){
            // printf("%f\t\t%f\t\t%f\n",wave.uInitial[i],wave.uhalf
[i],wave.ufull[i]);
            fprintf(outfile3,"%10.7f\t\t\t\t%10.7f\t\t\t\t%10.7f\n",wave-
>uInitial[i],wave->uhalf[i],wave->ufull[i]);
        }
    }
}


void runLaxWendroff(){
    Hyperbolic wave;
    wave.dt = 0.0025;
    InitializeU(&wave);
    LaxWendroff(&wave);
}
```

```
int main(){
    runLaxWendroff();
}
```

Initial Condition	Halfway Point	Final Condition
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
2.0000000	0.0000000	0.0000000
4.0000000	0.0000000	0.0000000
6.0000000	0.0000000	0.0000000
8.0000000	0.0000000	0.0000000
10.0000000	0.0000000	0.0000000
12.0000000	0.0000000	0.0000000
14.0000000	0.0000000	0.0000000
16.0000000	0.0000000	0.0000000
18.0000000	0.0000000	0.0000000
20.0000000	0.0000000	0.0000000
18.0000000	0.0000000	0.0000000
16.0000000	0.0000000	0.0000000
14.0000000	0.0000000	0.0000000
12.0000000	0.0000000	0.0000000
10.0000000	0.0000000	0.0000000
8.0000000	2.0000000	0.0000000
6.0000000	4.0000000	0.0000000
4.0000000	6.0000000	0.0000000
2.0000000	8.0000000	0.0000000
0.0000000	10.0000000	0.0000000
0.0000000	12.0000000	0.0000000
0.0000000	14.0000000	0.0000000
0.0000000	16.0000000	0.0000000
0.0000000	18.0000000	0.0000000
0.0000000	20.0000000	0.0000000
0.0000000	18.0000000	0.0000000
0.0000000	16.0000000	0.0000000
0.0000000	14.0000000	0.0000000
0.0000000	12.0000000	0.0000000
0.0000000	10.0000000	0.0000000
0.0000000	8.0000000	2.0000000
0.0000000	6.0000000	4.0000000
0.0000000	4.0000000	6.0000000
0.0000000	2.0000000	8.0000000
0.0000000	0.0000000	10.0000000
0.0000000	0.0000000	12.0000000
0.0000000	0.0000000	14.0000000
0.0000000	0.0000000	16.0000000
0.0000000	0.0000000	18.0000000
0.0000000	0.0000000	20.0000000
0.0000000	0.0000000	18.0000000
0.0000000	0.0000000	16.0000000
0.0000000	0.0000000	14.0000000
0.0000000	0.0000000	12.0000000
0.0000000	0.0000000	10.0000000

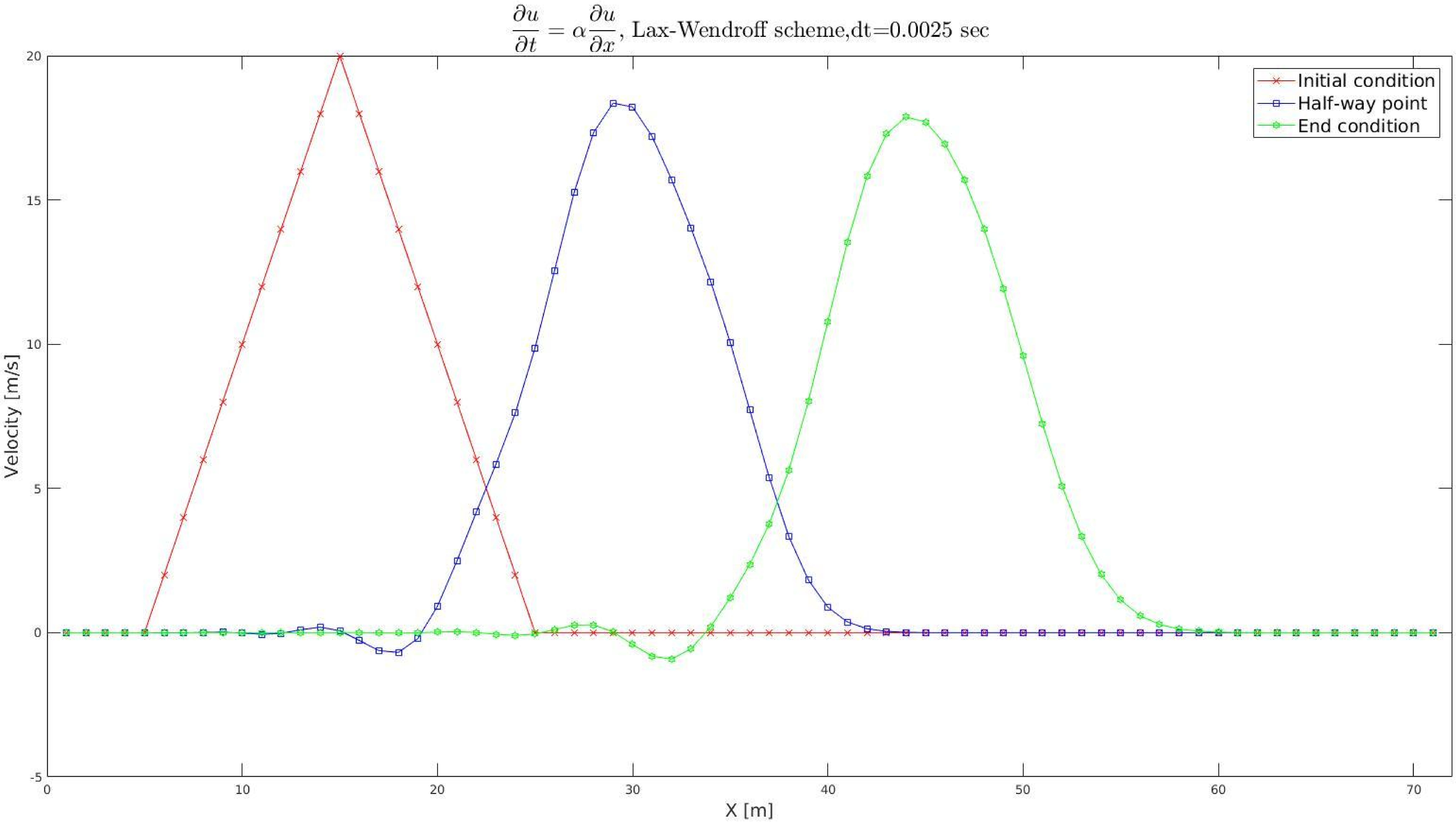
0.0000000	0.0000000	8.0000000
0.0000000	0.0000000	6.0000000
0.0000000	0.0000000	4.0000000
0.0000000	0.0000000	2.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000



$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x}$, Lax-Wendroff scheme,dt=0.005 sec
Initial condition
Half-way point
End condition
Velocity [m/s]
X [m]
0
10
20
30
40
50
60
70
2
4
6
8
10
12
14
16
18
20

Initial Condition	Halfway Point	Final Condition
0.0000000	0.0000000	0.0000000
0.0000000	-0.0001627	0.0000010
0.0000000	-0.0000428	-0.0000004
0.0000000	0.0012509	-0.0000083
0.0000000	0.0006766	-0.0000002
2.0000000	-0.0044176	0.0000364
4.0000000	-0.0044096	0.0000157
6.0000000	0.0101860	-0.0001182
8.0000000	0.0181120	-0.0001124
10.0000000	-0.0129662	0.0002860
12.0000000	-0.0553810	0.0004969
14.0000000	-0.0229774	-0.0004203
16.0000000	0.1029012	-0.0015721
18.0000000	0.1892467	-0.0002163
20.0000000	0.0675285	0.0035023
18.0000000	-0.2695090	0.0036320
16.0000000	-0.6199689	-0.0042078
14.0000000	-0.6845094	-0.0120409
12.0000000	-0.2038266	-0.0043809
10.0000000	0.9110296	0.0193756
8.0000000	2.4851865	0.0328664
6.0000000	4.1854171	0.0056884
4.0000000	5.8400588	-0.0555238
2.0000000	7.6308179	-0.0940905
0.0000000	9.8648929	-0.0433901
0.0000000	12.5508224	0.1047097
0.0000000	15.2520569	0.2598452
0.0000000	17.3396196	0.2692175
0.0000000	18.3568290	0.0275474
0.0000000	18.2209291	-0.4103195
0.0000000	17.1988793	-0.8163320
0.0000000	15.7014911	-0.9142820
0.0000000	14.0139427	-0.5556143
0.0000000	12.1668825	0.2055151
0.0000000	10.0637160	1.2086495
0.0000000	7.7218391	2.3686321
0.0000000	5.3718192	3.7763392
0.0000000	3.3350730	5.6236752
0.0000000	1.8300423	8.0091464
0.0000000	0.8823790	10.7799029
0.0000000	0.3723950	13.5385443
0.0000000	0.1371208	15.8160159
0.0000000	0.0438990	17.2865453
0.0000000	0.0121675	17.8753740
0.0000000	0.0029034	17.7028983
0.0000000	0.0005920	16.9378301
0.0000000	0.0001021	15.6870568
0.0000000	0.0000147	13.9941513
0.0000000	0.0000017	11.9159440
0.0000000	0.0000002	9.5907212

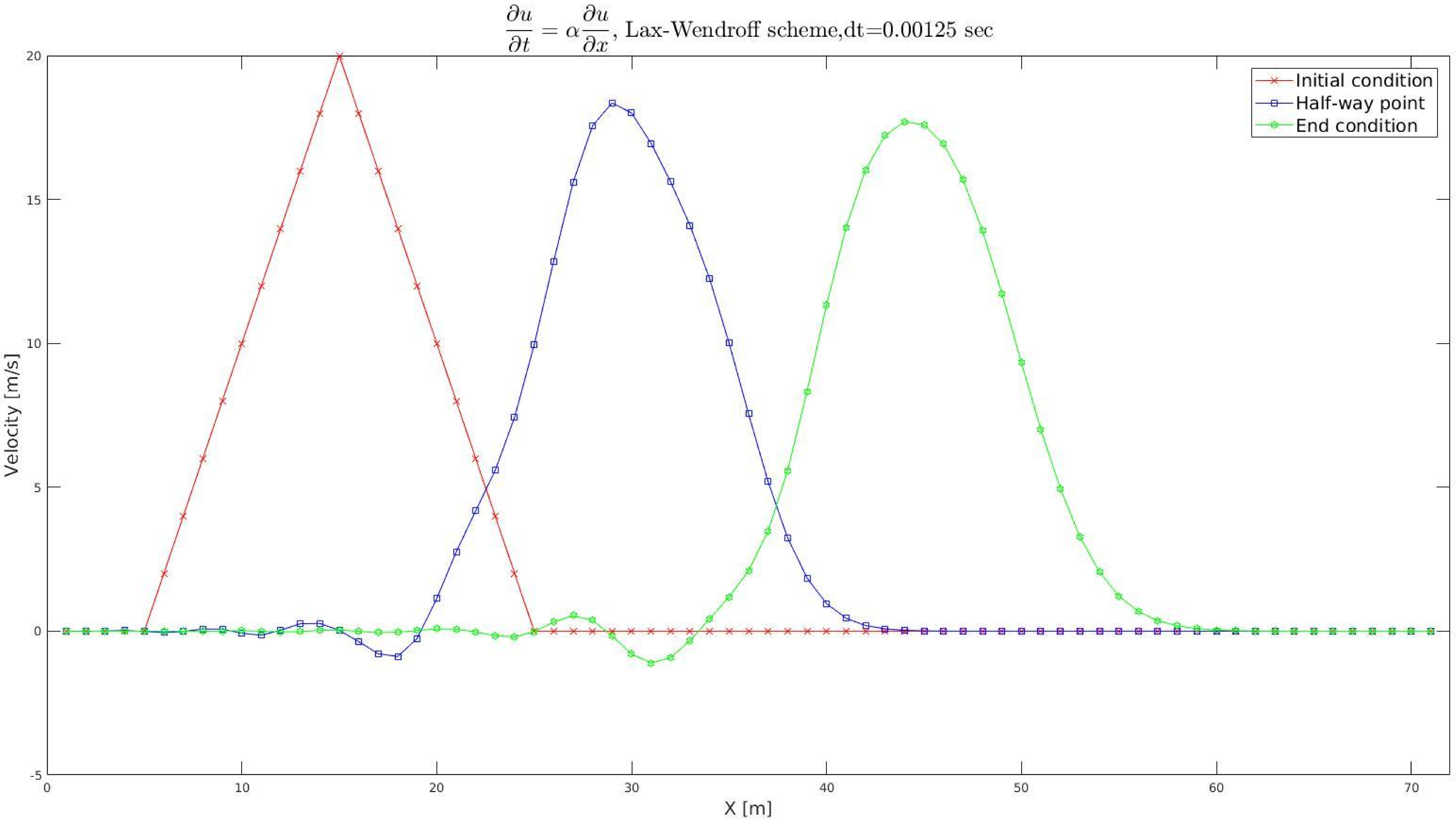
0.0000000	0.0000000	7.2373342
0.0000000	0.0000000	5.0909587
0.0000000	0.0000000	3.3261231
0.0000000	0.0000000	2.0142584
0.0000000	0.0000000	1.1295550
0.0000000	0.0000000	0.5863565
0.0000000	0.0000000	0.2817493
0.0000000	0.0000000	0.1253273
0.0000000	0.0000000	0.0516106
0.0000000	0.0000000	0.0196756
0.0000000	0.0000000	0.0069428
0.0000000	0.0000000	0.0022667
0.0000000	0.0000000	0.0006843
0.0000000	0.0000000	0.0001909
0.0000000	0.0000000	0.0000491
0.0000000	0.0000000	0.0000117
0.0000000	0.0000000	0.0000025
0.0000000	0.0000000	0.0000005
0.0000000	0.0000000	0.0000001
0.0000000	0.0000000	0.0000000
0.0000000	0.0000000	0.0000000



$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x}$, Lax-Wendroff scheme,dt=0.0025 sec
Initial condition
Half-way point
End condition
Velocity [m/s]
X [m]
20
15
10
5
0
-5
0
10
20
30
40
50
60
70

Initial Condition	Halfway Point	Final Condition
0.0000000	0.0000000	0.0000000
0.0000000	-0.0049369	0.0000818
0.0000000	0.0067162	-0.0006414
0.0000000	0.0212887	-0.0005580
0.0000000	-0.0086612	0.0020210
2.0000000	-0.0489392	0.0022477
4.0000000	-0.0153399	-0.0040811
6.0000000	0.0733127	-0.0067654
8.0000000	0.0686099	0.0050449
10.0000000	-0.0741890	0.0152554
12.0000000	-0.1407405	-0.0003719
14.0000000	0.0321247	-0.0248977
16.0000000	0.2568378	-0.0147085
18.0000000	0.2672814	0.0260726
20.0000000	0.0294881	0.0379110
18.0000000	-0.3676946	-0.0057190
16.0000000	-0.7845994	-0.0519972
14.0000000	-0.8832673	-0.0396092
12.0000000	-0.2497596	0.0239744
10.0000000	1.1298188	0.0786769
8.0000000	2.7505117	0.0681389
6.0000000	4.1810687	-0.0280611
4.0000000	5.5838227	-0.1615749
2.0000000	7.4396820	-0.1995751
0.0000000	9.9503760	-0.0201145
0.0000000	12.8533148	0.3169534
0.0000000	15.5972572	0.5389685
0.0000000	17.5665705	0.3765077
0.0000000	18.3531326	-0.1703161
0.0000000	18.0039582	-0.7943025
0.0000000	16.9504376	-1.1081920
0.0000000	15.6123648	-0.9194065
0.0000000	14.0907542	-0.3257445
0.0000000	12.2479368	0.4244930
0.0000000	10.0111261	1.1877938
0.0000000	7.5496136	2.0920449
0.0000000	5.1960335	3.4658599
0.0000000	3.2498618	5.5669606
0.0000000	1.8466858	8.3266229
0.0000000	0.9551136	11.3260470
0.0000000	0.4508681	14.0208001
0.0000000	0.1948486	16.0217404
0.0000000	0.0773214	17.2199231
0.0000000	0.0282535	17.7044583
0.0000000	0.0095305	17.5929272
0.0000000	0.0029745	16.9277409
0.0000000	0.0008606	15.6962687
0.0000000	0.0002313	13.9205234
0.0000000	0.0000578	11.7250356
0.0000000	0.0000134	9.3325448

0.0000000	0.0000029	7.0002442
0.0000000	0.0000006	4.9434810
0.0000000	0.0000001	3.2875450
0.0000000	0.0000000	2.0609567
0.0000000	0.0000000	1.2196712
0.0000000	0.0000000	0.6824880
0.0000000	0.0000000	0.3617024
0.0000000	0.0000000	0.1818557
0.0000000	0.0000000	0.0868774
0.0000000	0.0000000	0.0394945
0.0000000	0.0000000	0.0171088
0.0000000	0.0000000	0.0070715
0.0000000	0.0000000	0.0027921
0.0000000	0.0000000	0.0010543
0.0000000	0.0000000	0.0003811
0.0000000	0.0000000	0.0001320
0.0000000	0.0000000	0.0000438
0.0000000	0.0000000	0.0000140
0.0000000	0.0000000	0.0000043
0.0000000	0.0000000	0.0000013
0.0000000	0.0000000	0.0000004



$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x}$, Lax-Wendroff scheme,dt=0.00125 sec
Initial condition
Half-way point
End condition
Velocity [m/s]
X [m]
20
15
10
5
0
-5
0
10
20
30
40
50
60
70

```
//BTCS IMPLICIT

#include <iostream>
#include <stdio.h>
#include <cmath>
#include <fstream>
#include <vector>
using namespace std;

//DECLARE STRUCTURE

struct Hyperbolic
{
    double dx;
    double dt;
    int imax;
    double tmax; //max time
    double nmax; //max time steps
    double alpha; //speed of sound
    double c;
    double c_square;
    vector <double> u0; //initial u //USED IN LOOPS
    vector <double> u; //u at n+1 level, USED IN LOOPS
    vector <double> uhalf; //half way point, for print
    vector <double> ufull; //end solution, for print
    vector <double> uInitial; //initial sol, for print

};

void InitializeU(struct Hyperbolic *wave){

    //BASIC PROPERTIES
    wave->dx = 1.0;
    wave->imax = 71;
    wave->alpha = 200.00;
    wave->tmax = 0.15;
    wave->nmax = (wave->tmax/wave->dt)+1.;

    wave->c = (wave->alpha*wave->dt)/(wave->dx);
    wave->c_square = pow(wave->c,2.);

    //SETTING U AND UN
    wave->u0.resize(wave->imax+1);
    wave->u.resize(wave->imax+1);
    wave->uhalf.resize(wave->imax+1);
    wave->ufull.resize(wave->imax+1);
    wave->uInitial.resize(wave->imax+1);



    //SETTING UP INITAL CONDITIONS, U0
    for (int i = 1; i<=wave->imax;i++){
```

```
        if (i == 15){
            wave->u0[i]= 20.00;
        }

        else if (i>=5 && i<15){
            wave->u0[i] = (2*i)-10;
        }
        else if (i>15 && i<=25){
            wave->u0[i] = (-2*i) + 50;
        }
    }
}


void thomasTriDiagonal(int nmax, double a[],double b[],double c[], double d
[],struct Hyperbolic *wave){


    int imax = wave->imax-1;
    double dprime[imax+1];
    double cprime[imax+1];
    dprime[1] = d[1];
    cprime[1] = c[1];

    //FORWARD LOOP
    for (int i = 2; i<=imax;i++){
        dprime[i] = d[i] - ((b[i]*a[i-1])/(dprime[i-1]));
        cprime[i] = c[i] - ((cprime[i-1]*b[i])/(dprime[i-1]));
    }



    wave->u0[imax] = cprime[imax]/dprime[imax];

    // u[imax] = cprime[imax]/dprime[imax];


    //BACKWARD LOOP
    for (int i = imax-1;i>=2;i--){
        wave->u0[i] = (cprime[i]-(a[i]*wave->u0[i+1]))/(dprime[i]);

        // u[i] = (cprime[i]-(a[i]*u[i+1]))/(dprime[i]);
    }
}

void BTCS_implicit(struct Hyperbolic *wave){



    //THOMAS + TRIDIAGONAL PARAMETERS
```

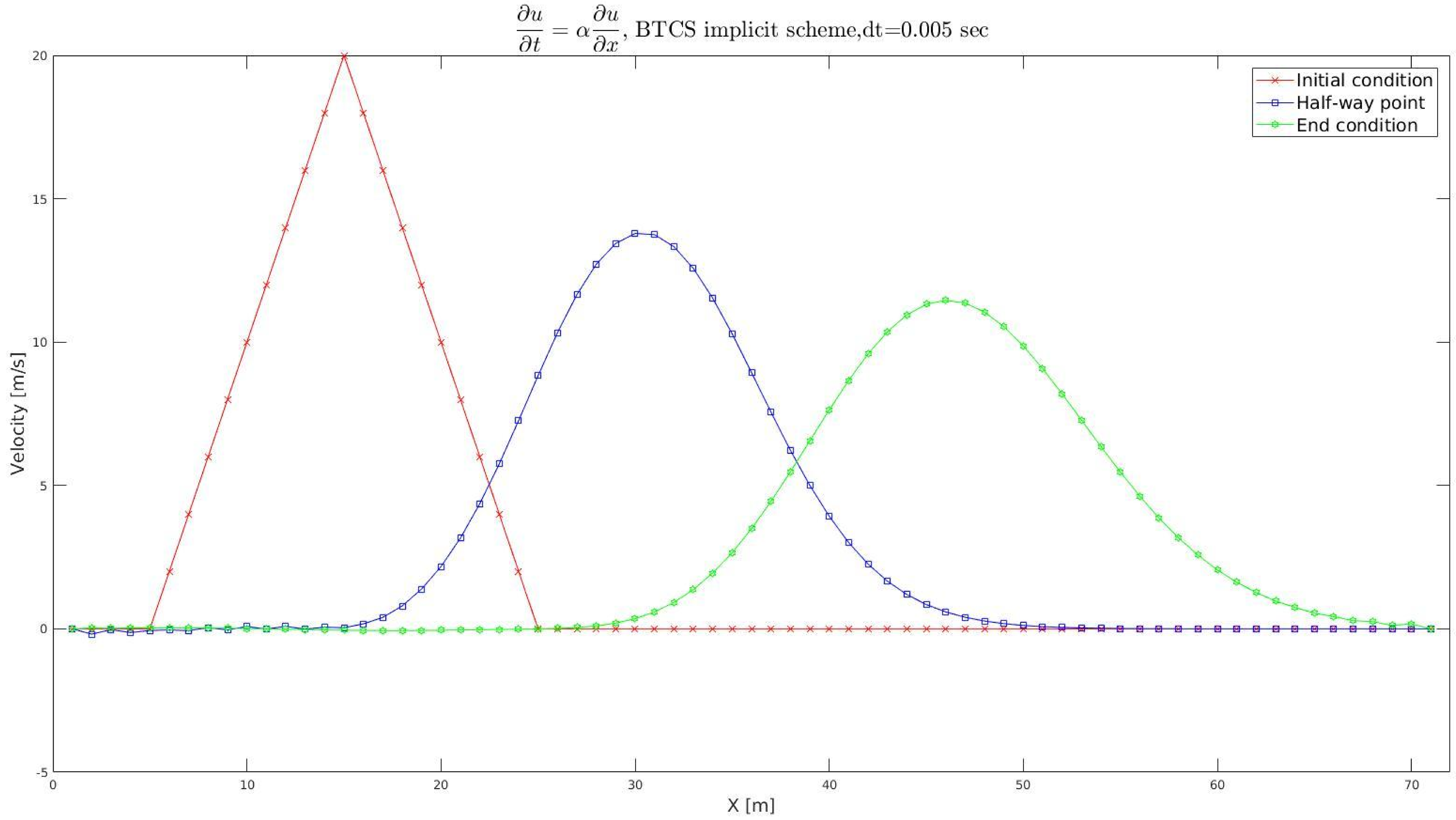
```
int imax = wave->imax;
int nmax = wave->nmax;

double halfC = 0.5*wave->c;

double a[imax+1] = {0.0}; //above
double b[imax+1] = {0.0}; //below
double c[imax+1]; //rhs
double d[imax+1] = {0.0};//diagonal

for (int i=2; i<=imax-1;i++){
    a[i] = -1.*halfC;
    b[i] = halfC;
    d[i] = -1.;
}

//Special values


a[imax-1] = 0.0;
a[1] = 0.0;

b[1] = 0.0;
b[imax] = 0.0;

d[1] = 1.;
d[imax] = 1.;


c[1] = wave->u[1];
c[imax] = wave->u[imax];

for (int i = 1; i<=imax-1;i++){
    c[i] = -1.*wave->u0[i];
}




//Create a U initial

for (int i = 1; i<=imax;i++){
    wave->uInitial[i] = wave->u0[i];
}

// thomasTriDiagonal(nmax,a,b,c,d,wave);

//Finite Difference Loop

for (int n = 0; n<=nmax;n++){
    thomasTriDiagonal(nmax,a,b,c,d,wave);
    for (int i = 1;i<=imax;i++){
```

```
            c[i] = -1.*wave->u0[i];


            if (n == nmax/2){
                wave->uhalf[i] = wave->u0[i];
            }

            else if (n == nmax){
                wave->ufull[i] = wave->u0[i];
            }
        }
    }

    if (wave->dt == 0.005){
        FILE* outfile1;
        outfile1 = fopen("05implicit.dat","w");
        fprintf(outfile1,"Initial Condition       Halfway Point        Final
Condition\n");
        fprintf
(outfile1,"---------------------------------------------------------------\n");
        for (int i = 1; i<=wave->imax;i++){
            printf("%f\t%f\t%f\n",wave->uInitial[i],wave->uhalf[i],wave->ufull
[i]);
            fprintf(outfile1,"%10.7f\t\t\t\t%10.7f\t\t\t\t%10.7f\n",wave-
>uInitial[i],wave->uhalf[i],wave->ufull[i]);
        }
    }

    else if (wave->dt == 0.0025){
        FILE* outfile2;
        outfile2 = fopen("025implicit.dat","w");
        fprintf(outfile2,"Initial Condition       Halfway Point        Final
Condition\n");
        fprintf
(outfile2,"---------------------------------------------------------------\n");
        for (int i = 1; i<=wave->imax;i++){
            printf("%f\t%f\t%f\n",wave->uInitial[i],wave->uhalf[i],wave->ufull
[i]);
            fprintf(outfile2,"%10.7f\t\t\t\t%10.7f\t\t\t\t%10.7f\n",wave-
>uInitial[i],wave->uhalf[i],wave->ufull[i]);
        }
    }
    else if (wave->dt == 0.00125){
        FILE* outfile3;
        outfile3 = fopen("0125implicit.dat","w");
        fprintf(outfile3,"Initial Condition       Halfway Point        Final
Condition\n");
        fprintf
(outfile3,"---------------------------------------------------------------
\n");
        for (int i = 1; i<=wave->imax;i++){
            printf("%f\t%f\t%f\n",wave->uInitial[i],wave->uhalf[i],wave->ufull
```

```
[i]);
            fprintf(outfile3,"%10.7f\t\t\t\t%10.7f\t\t\t\t%10.7f\n",wave-
>uInitial[i],wave->uhalf[i],wave->ufull[i]);
        }
    }
}

void runBTCSCode(){
    Hyperbolic wave;
    wave.dt = 0.00125;
    InitializeU(&wave);
    BTCS_implicit(&wave);
}

int main(){
    runBTCSCode();
}
```

Initial Condition	Halfway Point	Final Condition
0.0000000	0.0000000	0.0000000
0.0000000	-0.1848484	0.0303970
0.0000000	-0.0268878	0.0182988
0.0000000	-0.1309387	0.0330365
0.0000000	-0.0566288	0.0310222
2.0000000	-0.0374172	0.0343926
4.0000000	-0.0622950	0.0328864
6.0000000	0.0408233	0.0284698
8.0000000	-0.0402468	0.0216875
10.0000000	0.0826010	0.0118314
12.0000000	-0.0089482	0.0000391
14.0000000	0.0867680	-0.0128216
16.0000000	-0.0132461	-0.0258514
18.0000000	0.0571551	-0.0378541
20.0000000	0.0379892	-0.0478392
18.0000000	0.1747678	-0.0550541
16.0000000	0.3993221	-0.0589114
14.0000000	0.7977381	-0.0591262
12.0000000	1.3884252	-0.0557906
10.0000000	2.1722687	-0.0493662
8.0000000	3.1659002	-0.0407125
6.0000000	4.3688466	-0.0310105
4.0000000	5.7535355	-0.0213525
2.0000000	7.2670778	-0.0119585
0.0000000	8.8275157	-0.0011837
0.0000000	10.3282139	0.0154371
0.0000000	11.6569159	0.0458751
0.0000000	12.7173823	0.1018569
0.0000000	13.4412001	0.1986255
0.0000000	13.7889899	0.3542345
0.0000000	13.7476789	0.5883841
0.0000000	13.3290867	0.9207523
0.0000000	12.5699963	1.3687861
0.0000000	11.5308055	1.9450611
0.0000000	10.2903490	2.6545419
0.0000000	8.9368775	3.4922374
0.0000000	7.5572765	4.4417622
0.0000000	6.2272845	5.4751626
0.0000000	5.0048165	6.5541278
0.0000000	3.9272203	7.6324742
0.0000000	3.0121050	8.6596118
0.0000000	2.2606532	9.5846053
0.0000000	1.6621344	10.3603886
0.0000000	1.1985185	10.9476988
0.0000000	0.8484578	11.3182900
0.0000000	0.5902851	11.4571126
0.0000000	0.4039755	11.3631659
0.0000000	0.2722082	11.0490529
0.0000000	0.1807455	10.5391770
0.0000000	0.1183579	9.8671302

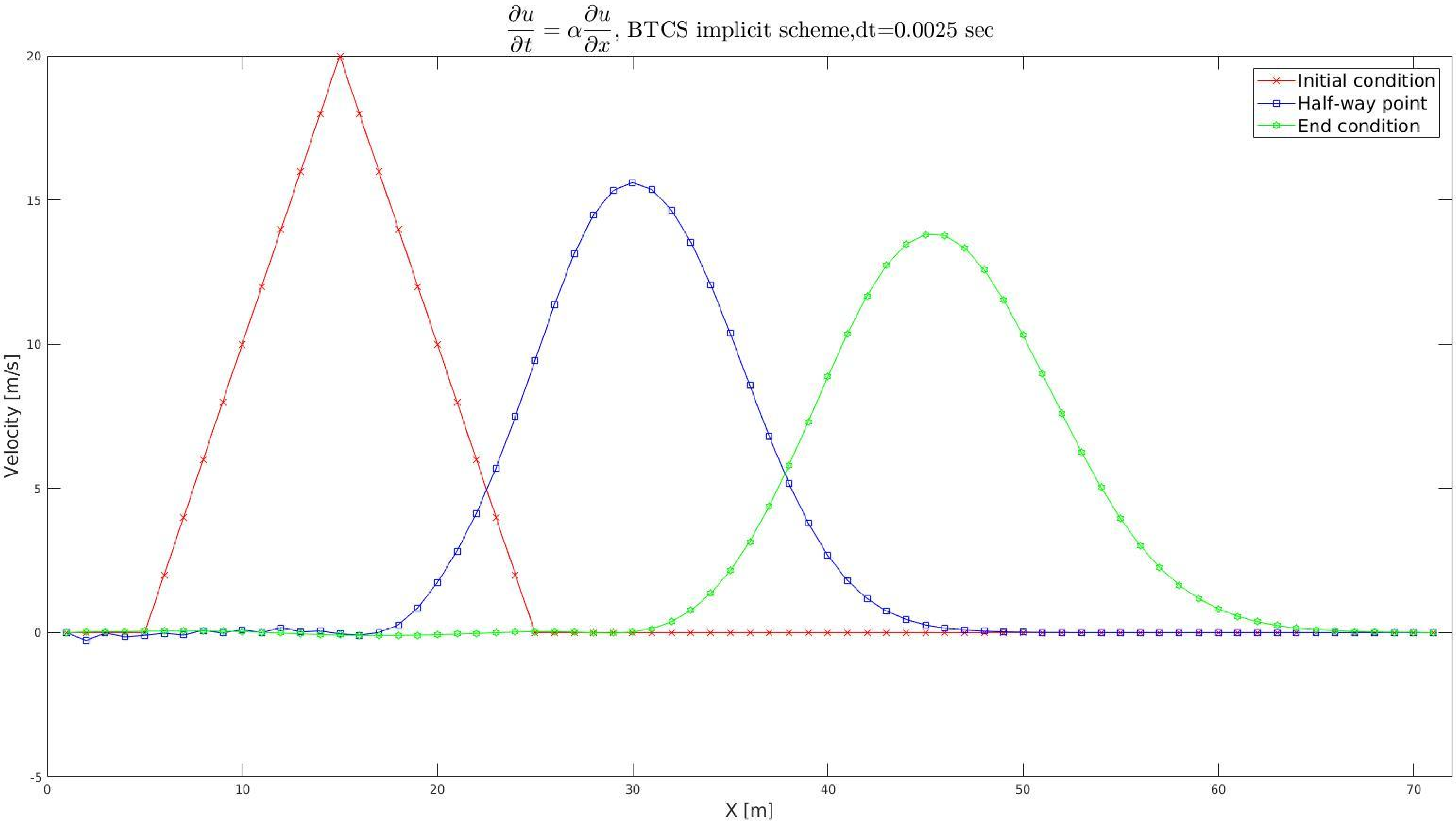
0.0000000	0.0764911	9.0721950
0.0000000	0.0488211	8.1961484
0.0000000	0.0307941	7.2795294
0.0000000	0.0192065	6.3597085
0.0000000	0.0118521	5.4676833
0.0000000	0.0072399	4.6289454
0.0000000	0.0043800	3.8599229
0.0000000	0.0026255	3.1733435
0.0000000	0.0015601	2.5710018
0.0000000	0.0009192	2.0573692
0.0000000	0.0005373	1.6212215
0.0000000	0.0003117	1.2671008
0.0000000	0.0001795	0.9697470
0.0000000	0.0001026	0.7462558
0.0000000	0.0000582	0.5485207
0.0000000	0.0000330	0.4269291
0.0000000	0.0000183	0.2863340
0.0000000	0.0000106	0.2492924
0.0000000	0.0000052	0.1207170
0.0000000	0.0000041	0.1711550
0.0000000	0.0000000	0.0000000



$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x}$, BTCS implicit scheme,dt=0.005 sec
Initial condition
Half-way point
End condition
Velocity [m/s]
X [m]

Initial Condition	Halfway Point	Final Condition
0.0000000	0.0000000	0.0000000
0.0000000	-0.2704862	0.0250349
0.0000000	-0.0121406	0.0324868
0.0000000	-0.1492419	0.0345976
0.0000000	-0.0937159	0.0490854
2.0000000	-0.0207015	0.0519139
4.0000000	-0.0851507	0.0554964
6.0000000	0.0766124	0.0567201
8.0000000	-0.0175737	0.0472068
10.0000000	0.1017377	0.0307578
12.0000000	-0.0022229	0.0082930
14.0000000	0.1713875	-0.0178777
16.0000000	0.0238662	-0.0430484
18.0000000	0.0551033	-0.0659095
20.0000000	-0.0422714	-0.0846919
18.0000000	-0.0853547	-0.0960161
16.0000000	0.0019749	-0.0993031
14.0000000	0.2734113	-0.0959114
12.0000000	0.8610860	-0.0862349
10.0000000	1.7380124	-0.0709230
8.0000000	2.8272861	-0.0517796
6.0000000	4.1411435	-0.0297881
4.0000000	5.7063608	-0.0051152
2.0000000	7.4955637	0.0193942
0.0000000	9.4307045	0.0364087
0.0000000	11.3748476	0.0377184
0.0000000	13.1295997	0.0211070
0.0000000	14.4906322	-0.0039978
0.0000000	15.3254570	-0.0149683
0.0000000	15.6018171	0.0210585
0.0000000	15.3551141	0.1414426
0.0000000	14.6417703	0.3827319
0.0000000	13.5206047	0.7782198
0.0000000	12.0636857	1.3576557
0.0000000	10.3717372	2.1456353
0.0000000	8.5731876	3.1558261
0.0000000	6.8047426	4.3819433
0.0000000	5.1851499	5.7899222
0.0000000	3.7953799	7.3159954
0.0000000	2.6718626	8.8723733
0.0000000	1.8118363	10.3584881
0.0000000	1.1855959	11.6738689
0.0000000	0.7500111	12.7293350
0.0000000	0.4595271	13.4551316
0.0000000	0.2731763	13.8062426
0.0000000	0.1578355	13.7655368
0.0000000	0.0887759	13.3449068
0.0000000	0.0486822	12.5839572
0.0000000	0.0260639	11.5457339
0.0000000	0.0136418	10.3095614

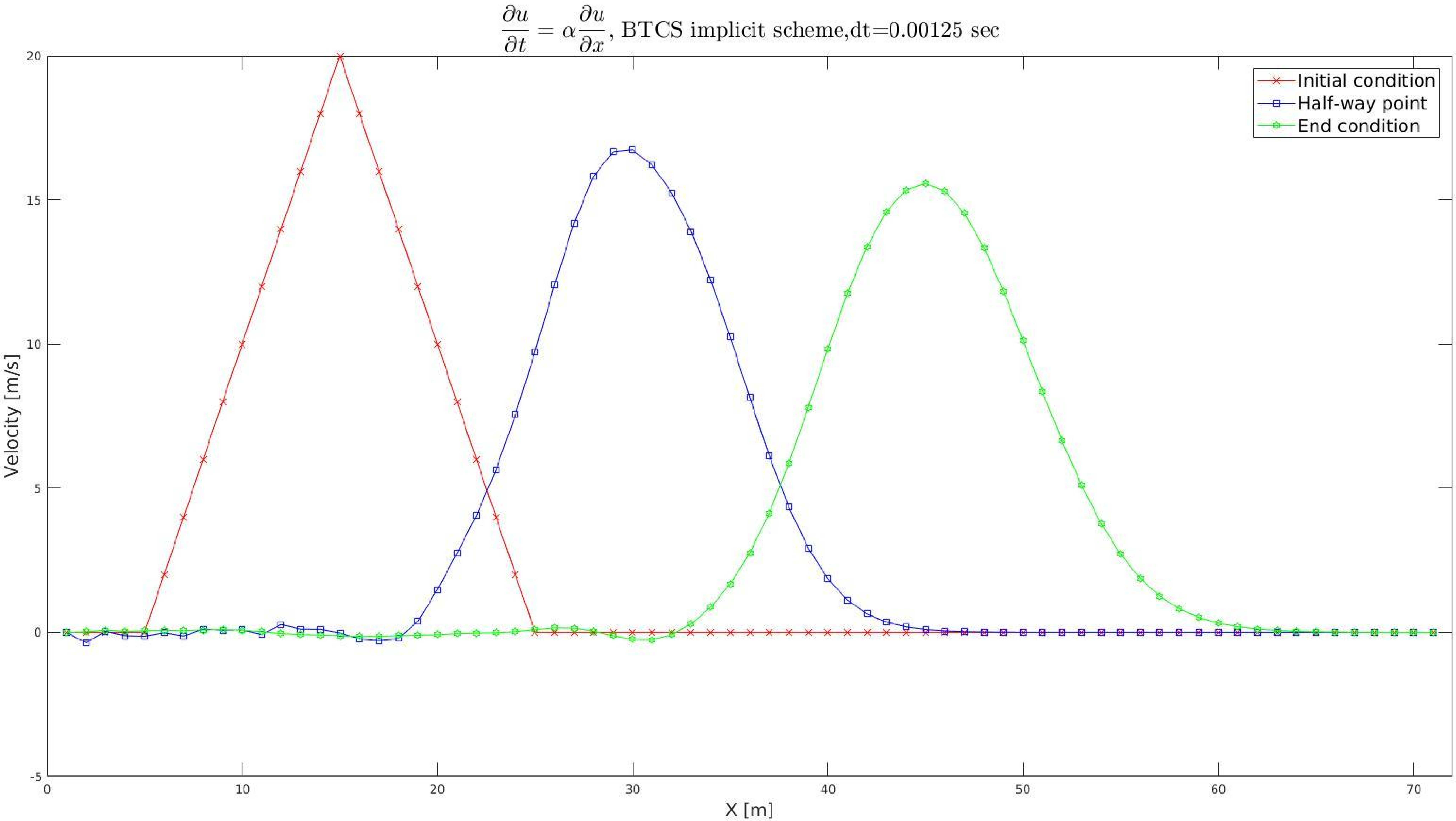
0.0000000	0.0069887	8.9619070
0.0000000	0.0035083	7.5868275
0.0000000	0.0017276	6.2576851
0.0000000	0.0008353	5.0314182
0.0000000	0.0003969	3.9459384
0.0000000	0.0001855	3.0204907
0.0000000	0.0000853	2.2582545
0.0000000	0.0000387	1.6502478
0.0000000	0.0000173	1.1795629
0.0000000	0.0000076	0.8253017
0.0000000	0.0000033	0.5656140
0.0000000	0.0000014	0.3800191
0.0000000	0.0000006	0.2503882
0.0000000	0.0000003	0.1620670
0.0000000	0.0000001	0.1027952
0.0000000	0.0000000	0.0645302
0.0000000	0.0000000	0.0390089
0.0000000	0.0000000	0.0247074
0.0000000	0.0000000	0.0126557
0.0000000	0.0000000	0.0110539
0.0000000	0.0000000	0.0000000



$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x}$, BTCS implicit scheme,dt=0.0025 sec
Initial condition
Half-way point
End condition
Velocity [m/s]
X [m]
20
15
10
5
0
-5
0
10
20
30
40
50
60
70

Initial Condition	Halfway Point	Final Condition
0.0000000	0.0000000	0.0000000
0.0000000	-0.3624306	0.0151217
0.0000000	0.0410833	0.0740911
0.0000000	-0.1183540	0.0338040
0.0000000	-0.1351541	0.0482200
2.0000000	-0.0114128	0.0681920
4.0000000	-0.1270261	0.0554586
6.0000000	0.1078493	0.0785344
8.0000000	0.0720009	0.0944560
10.0000000	0.0800488	0.0684249
12.0000000	-0.0772083	0.0181659
14.0000000	0.2672706	-0.0394243
16.0000000	0.1030460	-0.0750526
18.0000000	0.0990161	-0.0915072
20.0000000	-0.0198052	-0.1189489
18.0000000	-0.2196948	-0.1426061
16.0000000	-0.2976073	-0.1357636
14.0000000	-0.2119058	-0.1165894
12.0000000	0.3775132	-0.1024025
10.0000000	1.4852739	-0.0783677
8.0000000	2.7372731	-0.0443100
6.0000000	4.0542319	-0.0241200
4.0000000	5.6337789	-0.0144241
2.0000000	7.5521329	0.0204536
0.0000000	9.7359528	0.0902584
0.0000000	12.0419070	0.1501861
0.0000000	14.1936788	0.1402891
0.0000000	15.8179537	0.0394310
0.0000000	16.6615531	-0.1151925
0.0000000	16.7319365	-0.2431287
0.0000000	16.2052404	-0.2538499
0.0000000	15.2410406	-0.0826389
0.0000000	13.9017253	0.2931345
0.0000000	12.2095510	0.8734598
0.0000000	10.2381416	1.6744440
0.0000000	8.1434580	2.7450457
0.0000000	6.1227510	4.1377851
0.0000000	4.3477059	5.8536966
0.0000000	2.9185381	7.8065361
0.0000000	1.8558665	9.8339368
0.0000000	1.1207973	11.7450265
0.0000000	0.6446452	13.3709935
0.0000000	0.3541175	14.5919855
0.0000000	0.1862878	15.3372230
0.0000000	0.0940900	15.5730721
0.0000000	0.0457363	15.2951193
0.0000000	0.0214435	14.5297513
0.0000000	0.0097170	13.3398121
0.0000000	0.0042637	11.8252874
0.0000000	0.0018147	10.1138208

0.0000000	0.0007504	8.3424996
0.0000000	0.0003019	6.6369812
0.0000000	0.0001183	5.0946559
0.0000000	0.0000452	3.7759957
0.0000000	0.0000169	2.7046625
0.0000000	0.0000062	1.8741769
0.0000000	0.0000022	1.2577903
0.0000000	0.0000008	0.8184733
0.0000000	0.0000003	0.5170097
0.0000000	0.0000001	0.3173816
0.0000000	0.0000000	0.1895527
0.0000000	0.0000000	0.1102561
0.0000000	0.0000000	0.0625220
0.0000000	0.0000000	0.0345989
0.0000000	0.0000000	0.0186974
0.0000000	0.0000000	0.0098861
0.0000000	0.0000000	0.0050947
0.0000000	0.0000000	0.0026143
0.0000000	0.0000000	0.0012180
0.0000000	0.0000000	0.0007629
0.0000000	0.0000000	0.0000000



$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x}$, BTCS implicit scheme,dt=0.00125 sec
Initial condition
Half-way point
End condition
Velocity [m/s]
X [m]
20
15
10
5
0
-5
0
10
20
30
40
50
60
70

```
//MAC CORMACK

#include <iostream>
#include <stdio.h>
#include <cmath>
#include <fstream>
#include <vector>
using namespace std;


struct Hyperbolic
{
    double dx;
    double dt;
    int imax;
    double tmax; //max time
    int nmax; //max time steps
    double dtbydx; //dt/dx quantity
    vector <double> u_star; //intermediate u value
    vector <double> un; //u at nth level, for corrector method
    vector <double> u; // future u, final solution (n+1)
    vector <double> e_vector; //E = u square /2
    vector <double> u_initial;
    vector <double> x_vector_MCM;
    double interval;
};


void InitializeU(struct Hyperbolic *burgers){

    //BASIC PROPERTIES
    burgers->dx = 1.0;
    burgers->imax = 41;
    burgers->tmax = 2.4;
    burgers->nmax = (burgers->tmax/burgers->dt)+1;
    burgers->dtbydx = (burgers->dt)/(burgers->dx);
    burgers->interval = 0.4;

    //SETTING UP AND Es
    burgers->u_star.resize(burgers->imax+1);
    burgers->un.resize(burgers->imax+1);
    burgers->e_vector.resize(burgers->imax+1);
    burgers->u.resize(burgers->imax+1);
    burgers->u_initial.resize(burgers->imax+1);
    burgers->x_vector_MCM.resize(burgers->imax+1);


    burgers->x_vector_MCM[1] = 0;
    for (int i = 1; i<=burgers->imax;i++){
        burgers->x_vector_MCM[i+1] = burgers->x_vector_MCM[i]+1.0;
    }
```

```
    //Zeros out the vectors

    for (int i = 1; i<=burgers->imax;i++){
        burgers->u[i] = 0.0;

    }





    //SETTING UP INITIAL CONDITIONS

    for (int i = 1; i<=burgers->imax;i++){
        if (i>=1 && i<=20){
            burgers->u[i] = 5.0;
        }

        else if (i>20 && i<=burgers->imax){
            burgers->u[i] = 0.0;
        }
    }

    //Zeros out Un

    for (int i = 1; i<=burgers->imax;i++){
        burgers->un[i] = 0.0;
    }

}


double FluxVariable(double u){
    u = 0.5*pow(u,2.);
    return u;
}

double MacCormack(struct Hyperbolic *burgers){


    InitializeU(burgers);
    int imax = burgers->imax;
    int nmax = burgers->nmax;

    double u_combined[100][100]={0.0};

    //Generating nmax_vector;

    double nmax_vector[7];

    double factor = burgers->interval/burgers->dt;
```

```
    for (int i = 2; i<=7;i++){
        nmax_vector[i] = (factor)*double(i)-factor;
    }

    //Put into Initial

    for (int i = 1; i<=imax;i++){
        burgers->u_initial[i] = burgers->u[i];
    }

    for (int k = 1; k<=imax;k++){
        u_combined[k][1] = burgers->u_initial[k];
    }

    burgers->u_star[1] = burgers->u[1]-(burgers->dtbydx)*(FluxVariable(burgers-
>u[2])-FluxVariable(burgers->u[1]));

    //Time loop

    for (int n = 1; n<=nmax;n++){


        //Predictor step
        for (int i = 2; i<=imax-1;i++){

            burgers->u_star[i] = burgers->u[i]-(burgers->dtbydx)*0.5*(pow
(burgers->u[i+1],2.)-pow(burgers->u[i],2.));
        }

        //Corrector step

        for (int k = 2; k<=imax-1;k++){
            burgers->un[k] = 0.5*(burgers->u[k]+burgers->u_star[k])-
(0.25*burgers->dtbydx*(pow(burgers->u_star[k],2.)-pow(burgers->u_star
[k-1],2.)));

        }

        //Update U
        for (int p = 2; p<=imax-1;p++){

            burgers->u[p] = burgers->un[p];
        }

        if (n == nmax_vector[2]){
            for (int i = 1; i<=imax;i++){
                u_combined[i][2] = burgers->u[i];
            }
        }

        if (n == nmax_vector[3]){
```

```
            for (int i = 1; i<=imax;i++){
                u_combined[i][3] = burgers->u[i];
            }
        }

        if (n == nmax_vector[4]){
            for (int i = 1; i<=imax;i++){
                u_combined[i][4] = burgers->u[i];
            }
        }

        if (n == nmax_vector[5]){
            for (int i = 1; i<=imax;i++){
                u_combined[i][5] = burgers->u[i];
            }
        }

        if (n == nmax_vector[6]){
            for (int i = 1; i<=imax;i++){
                u_combined[i][6] = burgers->u[i];
            }
        }

        if (n == nmax_vector[7]){
            for (int i = 1; i<=imax;i++){
                u_combined[i][7] = burgers->u[i];
            }
        }

    }

    for (int i = 1; i<=imax;i++){
        u_combined[i][8] = burgers->x_vector_MCM[i];
    }





    if (burgers->dt == 0.1){
        FILE *outfile1;
        outfile1 = fopen("dt01burgers.dat","w");
        fprintf(outfile1,"Table shows solution at dt = 0.1 sec\n\n");
        fprintf(outfile1,"0.0 sec\t\t0.4 sec\t\t0.8 sec\t\t1.2 sec\t\t1.6 sec\t\t2.0 sec\t\t2.4 sec\t\tX POSITION\n");
        fprintf(outfile1,"-------------------------------------------------------------------------------\n");
        for (int i = 1; i<=41;i++){
            for (int j = 1;j<=8;j++){
                if ( j == 8){
                    fprintf(outfile1,"%4.2f\n",u_combined[i][j]);
```

```
                }
                else{
                fprintf(outfile1,"%f\t", u_combined[i][j]);
                }
            }
        fprintf(outfile1,"\n");
        }
    }


    else if (burgers->dt == 0.2){
        FILE *outfile2;
        outfile2 = fopen("dt02burgers.dat","w");
        fprintf(outfile2,"Table shows solution at dt = 0.2 sec\n\n");
        fprintf(outfile2,"0.0 sec\t\t0.4 sec\t\t0.8 sec\t\t1.2 sec\t\t1.6 sec\t
\t2.0 sec\t\t2.4 sec\t\tX POSITION\n");
        fprintf
(outfile2,"-------------------------------------------------------------------------------
\n");
        for (int i = 1; i<=41;i++){
            for (int j = 1;j<=8;j++){
                if ( j == 8){
                    fprintf(outfile2,"%4.2f\n",u_combined[i][j]);
                }
                else{
                fprintf(outfile2,"%f\t", u_combined[i][j]);
                }
            }
        fprintf(outfile2,"\n");
        }

    }
}

void runMacCorMack(){
    Hyperbolic burgers;
    burgers.dt = 0.2;
    InitializeU(&burgers);
    MacCormack(&burgers);
}


int main(){
    runMacCorMack();
}
```

Table shows solution at dt = 0.1 sec

0.0 sec	0.4 sec	0.8 sec	1.2 sec	1.6 sec	2.0 sec	2.4 sec	X POSITION
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	0.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	1.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	2.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	3.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	4.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	6.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	7.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	4.999999	8.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000002	5.000005	9.00
5.000000	5.000000	5.000000	5.000000	4.999997	4.999988	4.999985	10.00
5.000000	5.000000	5.000000	5.000002	5.000021	5.000050	5.000025	11.00
5.000000	5.000000	5.000000	4.999979	4.999881	4.999859	5.000017	12.00
5.000000	5.000000	5.000007	5.000188	5.000491	5.000191	4.999774	13.00
5.000000	5.000000	4.999859	4.998826	4.998707	5.000335	5.000528	14.00
5.000000	5.000000	5.001676	5.004870	5.001227	4.997780	5.000048	15.00
5.000000	4.999446	4.988040	4.988381	5.004904	5.003470	4.997004	16.00
5.000000	5.013947	5.049284	5.003973	4.980145	5.004925	5.005208	17.00
5.000000	4.866771	4.900338	5.060803	5.013412	4.973644	5.005845	18.00
5.000000	5.479994	4.930438	4.850228	5.073624	5.016653	4.968710	19.00
0.000000	4.145834	5.566253	4.921725	4.825632	5.086386	5.015221	20.00
0.000000	0.493369	4.084889	5.604734	4.894889	4.819047	5.095760	21.00
0.000000	0.000639	0.478479	4.097645	5.609318	4.870858	4.823407	22.00
0.000000	0.000000	0.000737	0.467975	4.124086	5.598410	4.857077	23.00
0.000000	0.000000	0.000000	0.000671	0.472995	4.144515	5.585103	24.00
0.000000	0.000000	0.000000	0.000000	0.000671	0.483181	4.154017	25.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000707	0.491522	26.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000745	27.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	28.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	29.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	30.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	31.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	32.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	33.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	34.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	35.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	36.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	37.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	38.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	39.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	40.00

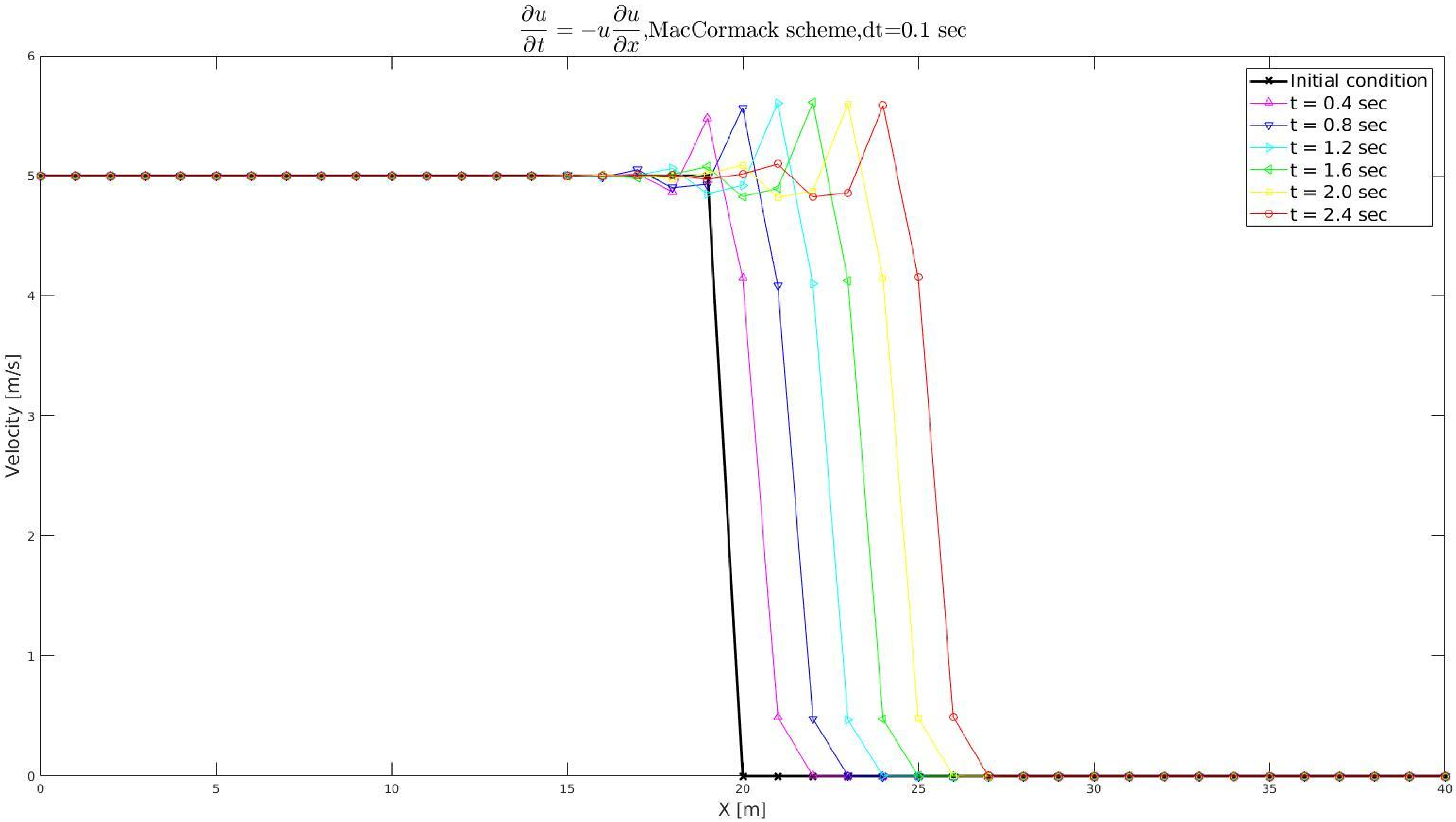
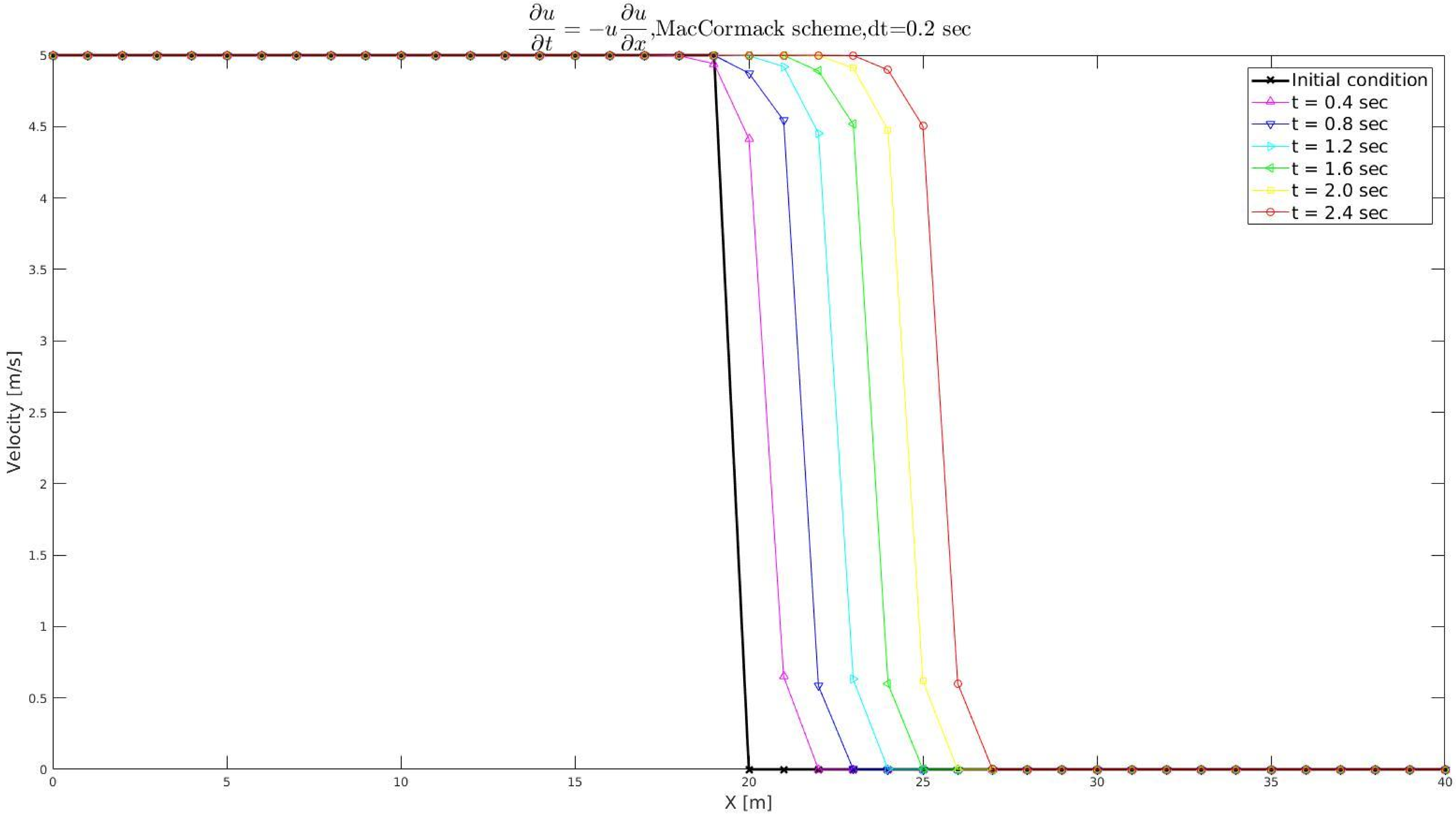

$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x}$,MacCormack scheme,dt=0.1 sec
Initial condition
t = 0.4 sec
t = 0.8 sec
t = 1.2 sec
t = 1.6 sec
t = 2.0 sec
t = 2.4 sec
Velocity [m/s]
X [m]
0
1
2
3
4
5
6
0
5
10
15
20
25
30
35
40

Table shows solution at dt = 0.2 sec

0.0 sec	0.4 sec	0.8 sec	1.2 sec	1.6 sec	2.0 sec	2.4 sec	X POSITION
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	0.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	1.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	2.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	3.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	4.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	6.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	7.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	8.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	9.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	10.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	11.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	12.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	13.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	14.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	15.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	16.00
5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	17.00
5.000000	4.995418	4.999990	5.000000	5.000000	5.000000	5.000000	18.00
5.000000	4.939885	4.999090	5.000000	5.000000	5.000000	5.000000	19.00
0.000000	4.415431	4.874366	4.996234	4.999995	5.000000	5.000000	20.00
0.000000	0.649266	4.543129	4.920080	4.998427	4.999999	5.000000	21.00
0.000000	0.000000	0.583397	4.454438	4.889093	4.997006	4.999997	22.00
0.000000	0.000000	0.000029	0.629230	4.515792	4.911090	4.998043	23.00
0.000000	0.000000	0.000000	0.000018	0.596669	4.473559	4.896024	24.00
0.000000	0.000000	0.000000	0.000000	0.000025	0.618326	4.503093	25.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000020	0.602819	26.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000023	27.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	28.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	29.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	30.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	31.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	32.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	33.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	34.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	35.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	36.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	37.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	38.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	39.00
0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	40.00


$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x}$,MacCormack scheme,dt=0.2 sec
Velocity [m/s]
X [m]
Initial condition
t = 0.4 sec
t = 0.8 sec
t = 1.2 sec
t = 1.6 sec
t = 2.0 sec
t = 2.4 sec