<https://github.com/Cave-man/NHT.git>

FTCS:

#include<iostream>

#include<cmath>

#include<stdlib.h>

#include<fstream>

using namespace std;

int main()

    {

        double c=0.1;

        double deltaX=0.01;

        double t=0.01;

        double deltaT=c\*deltaX\*deltaX;

        int const M=100;

        int N=t/deltaT;

        double T[M+1];

        double result[M+1];

        int i,j;

        for(i=1;i<M;i++)

        {

            if(i\*deltaX<0.3)

            {

                T[i]=0;

            }

            else if(i\*deltaX<0.7)

            {

                T[i]=1;

            }

            else

            {

                T[i]=10/3\*(1-i\*deltaX);

            }

        T[0]=0;

        T[M]=0;

        }

        for(i=1;i<=N+1;i++)

        {

            T[0]=0;

            T[M]=0;

            for(j=0;j<=M;j++)

            {

                result[j]=T[j];

            }

        std::ofstream outfile("FTCS.txt");//创建输出流对象

        for(j=1;j<100;j++)

        {

            T[j]=result[j]+deltaT\*(result[j+1]-2\*result[j]+result[j-1])/deltaX/deltaX;

            outfile << T[j] << std::endl;//向文件输出当前元素并换行

        }

            outfile.close();//关闭文件

        if(i\*deltaX==t)

        {

            cout<<"FTCS"<<" "<<endl;

            cout<<"Time="<< t <<endl;

            cout<<"sigma="<< c <<endl<<endl;

        for(j=0;j<=M;j++)

        {

                cout<< T[j] <<endl;

         }

            cout<<endl<<endl;

        }

        }

               system("pause");

                return 0;

BTCS:

#include<iostream>//输入输出流头文件

#include<cmath>

#include<stdlib.h>

#include<fstream>

using namespace std;

int main()

{

    double c=0.1;//参数设定

    double deltaX=0.01;//空间步长

    double t=0.01;//选定时刻

    double deltaT=c\*deltaX\*deltaX;//时间步长

    int const M=100;//空间网格数

    int N=t/deltaT;//时间网格数

    double T[M+1],P[M],Q[M],A[M],B[M],C[M],D[M];

    int j,i;//初始化

    for(i=1;i<M;i++)

    {

        if(i\*deltaX<0.3)

    {

        T[i]=0;

    }

    else if(i\*deltaX<0.7)

     {

        T[i]=1;

     }

    else

    {

        T[i]=10/3\*(1-i\*deltaX);

    }

        T[0]=0;

        T[M]=0;

    }

        //节点差分

    for(j=0;j<=N+1;j++)

    {

        T[0]=0;

        T[M]=0;

        for(i=1;i<=M-1;i++)

        {

            A[i]=1+2\*c;

}

        for(i=1;i<=M-2;i++)

        {

            B[i]=c;

}

        B[M-1]=0;

        for (i=2;i<=M-1;i++)

        {

        C[i]=c;

}

C[1]=0;

    for(i=2;i<=M-2;i++)

    {

    D[i]=T[i];

}

        D[1]=T[1];

        D[M-1]=T[M-1];

        P[1]=B[1]/A[1];

        Q[1]=D[1]/A[1];

    for(i=2;i<=M-1;i++)

    {

        P[i]=B[i]/(A[i]-C[i]\*P[i-1]);

        Q[i]=(D[i]+C[i]\*Q[i-1])/(A[i]-C[i]\*P[i-1]);

}

        std::ofstream outfile("BTCS.txt");

    for (i=1;i<100;i++)

    {

        T[i]=P[i]\*T[i+1]+Q[i];

        outfile << T[i] << std::endl;

}

        outfile.close();

    if(j\*deltaT==t)

    {

        cout<<"BTCS"<<""<<endl;

        cout<<"Time:"<< t <<endl;

        cout<<"c:"<< c  <<endl<<endl;

for(i=0;i<=M;i++)

    {

        cout<<T[i]<<endl;

}

        cout<<endl<<endl;

}

}

system("pause");

        return 0;

}

CNCS:

#include<iostream>//输入输出流头文件

#include<cmath>

#include<stdlib.h>

#include<fstream>

using namespace std;

int main()

{

double c=0.1;//参数设定

double deltaX=0.01;//空间步长

double t=0.01;//选定时刻

double deltaT=c\*deltaX\*deltaX;//时间步长

int const M=100;//空间网格数

int N=t/deltaT;//时间网格数

double T[M+1],P[M],Q[M],A[M],B[M],C[M],D[M];

int j,i;

for(i=1;i<M;i++)//初始化

{

    if(i\*deltaX<0.3)

    {

        T[i]=0;

    }

    else if(i\*deltaX<0.7)

     {

        T[i]=1;

     }

    else

    {

        T[i]=10/3\*(1-i\*deltaX);

    }

        T[0]=0;

        T[M]=0;

    }

        //节点差分

    for(j=0;j<=N+1;j++)

    {

        T[0]=0;

        T[M]=0;

        for(i=1;i<=M-1;i++)

        {

            A[i]=2/c+2;

        }

        for(i=1;i<=M-2;i++)

        {

            B[i]=1;

        }

        B[M-1]=0;

        for (i=2;i<=M-1;i++)

        {

        C[i]=1;

}

C[1]=0;

for(i=2;i<=M-2;i++)

{

    D[i]=T[i+1]+(2/c-2)\*T[i]+T[i-1];

}

D[1]=T[2]+(2/c-2)\*T[1]+T[0]+c\*0;

D[M-1]=T[M]+(2/c-2)\*T[M-1]+T[M-2]+c\*0;

P[1]=B[1]/A[1];

Q[1]=D[1]/A[1];

for(i=2;i<=M-1;i++)

{

    P[i]=B[i]/(A[i]-C[i]\*P[i-1]);

    Q[i]=(D[i]+C[i]\*Q[i-1])/(A[i]-C[i]\*P[i-1]);

}

std::ofstream outfile("CNCS.txt");

for (i=1;i<100;i++)

{

    T[i]=P[i]\*T[i+1]+Q[i];

    outfile << T[i] << std::endl;

}

outfile.close();

if(j\*deltaT==t)

{

        cout<<"CNCS"<<""<<endl;

        cout<<"Time:"<< t <<endl;

        cout<<"c:"<< c <<endl<<endl;

for(i=0;i<=M;i++)

{

        cout<<T[i]<<endl;

}

        cout<<endl<<endl;

}

}

system("pause");

        return 0;

}