```
//######################################################################################
//
// Copyright, 2013, Wang Xian, Xi'an Jiaotong university
// -- stability.c (Feb.2009)                                                                   ↙

// -- LAX & FTCS for one-order wave equation

//######################################################################################


     #include <stdio.h>
     #include <stdlib.h>
     #include <malloc.h>
     #include <assert.h> //what is this?
     #include <math.h>
     #include <string.h>



int main(int argc, char *argv[])
{

//------------------------------ define varaibles -----------------------------------------------

      int   nx;                   /* number of nodes in the profile */  //波形部分的网格数
      int   nxx;                  /* total number of grid */            //求解区域总网格数

      int   i;                    /* variable for loop */

      int   method;               /* method=1: LAX; method=2: FTCS */
      int   ifun;                 /* ifun=1: square wave; ifun=2: smooth wave */

      double dt;                  /* time step */
      double dx;                  /* cell dimension */
      double time_max;            /* maxium time  */                           //input data    ↙
    (time_max)
      double t;

      double aspeed;              /* speed of wave */                                  //input data
      double CFL;                 /* CFL number = aspeed*dt/dx  */                     //input data

      double *x;                  /* x[i] coordinate */
      double *u;                  /* u[i] at t */
      double *un;                 /* un[i] at t+dt */


//---------------------------  read input data -----------------------------------------------------
    FILE *in_data;
    FILE  * file_dat;
   char input[20];

     file_dat=fopen("result.plt","w");
     in_data=fopen("INPUT.dat","r");

     fscanf(in_data,"%s%d\n",input,&nx);
     fscanf(in_data,"%s%lf\n",input,&time_max);
     fscanf(in_data,"%s%lf\n",input,&CFL);
     fscanf(in_data,"%s%lf\n",input,&aspeed);
     fscanf(in_data,"%s%d\n",input,&method);
     fscanf(in_data,"%s%d\n",input,&ifun);


     fclose(in_data);
     nxx=4*nx;      //求解总区域

//----------------------------------------------------------------------------------
    u = (double *) malloc(nxx*sizeof(double));
    un = (double *) malloc(nxx*sizeof(double));
    x = (double *) malloc(nxx*sizeof(double));
  //----------------------------------------------------

    x[10]=0.0;      //波形原始起点  注意！！
    x[nx-1]=1.0;   //波形原始终点


    dx=1.0/(double)(nx-1-10);  // 注意！！！

    for(i=11;i<nxx;i++)
    {
       x[i]=x[i-1]+dx;
       u[i]=0.0;
       un[i]=0.0;
    }

    for(i=9;i>=0;i--)
    {
       x[i]=x[i+1]-dx;
       u[i]=0.0;
       un[i]=0.0;
    }




    //-------------initialize--------------------------------------

    for(i=0;i<nx;i++)
    {

       if(ifun==1)
        {

            if( x[i]>=0.0 && x[i]<=1.0)
            {
              u[i]=1.0;
            }
            else
            {
              u[i]=0.0;
            }
       }
        else
        {
            u[i]=-64.0*x[i]*x[i]*x[i]*(x[i]-1.0)*(x[i]-1.0)*(x[i]-1.0)*exp(-16.0*(x[i]-0.5)*(x[i]-0. ↙
    5));
        }
    }




    //-------------start computation-------------------------------
        dt = CFL * dx / aspeed ;
        printf("---------dt = %f-------------\n",dt);

        for(t=0.0;t<time_max;t=t+dt)
        {
            for(i=1;i<nxx-1;i++)
            {
              if (method ==1 ) // Lax
              {
                  un[i] = ( u[i+1] + u[i-1] )/2.0 - CFL * ( u[i+1] - u[i-1] )/2.0;
              }

              else   // FTCS
              {
                  un[i] = u[i] - CFL * ( u[i+1] - u[i-1] )/2.0;
              }

            }

           for(i=1;i<nxx-1;i++)
           {
             u[i]=un[i];
           }

        }

//-----------------output---------------------------------------------



        fputs("VARIABLES=X,U\n",file_dat);
    fprintf(file_dat, "ZONE I=%6d, F=POINT\n",nxx-1);

         for (i=0; i<nxx; i++)
          {
            fprintf(file_dat,"%15.9f %15.9f\n",x[i],u[i]);
          }

    fclose(file_dat);




    return 0;


}/***** end main *****/
```