

Numerical solution of the advection equation using finite difference methods

Simone Orioli

Academic Year 2013/2014

1 Introduction

Given the advection equation in 1D $u_t + u_x = 0$ build a numerical code to solve it on a grid with extent $x \in [0, 10]$ and with initial conditions given by

$$u(x, t = 0) = \exp[-(x - x_0)^2] \quad (1)$$

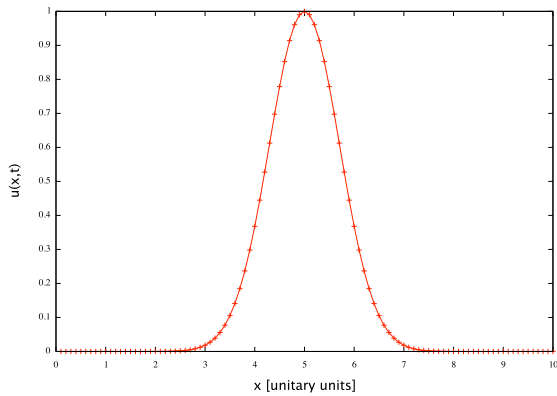
with $x_0 = 5$, or

$$u(x, t = 0) = \begin{cases} 1 & \text{if } x \in [4, 6] \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

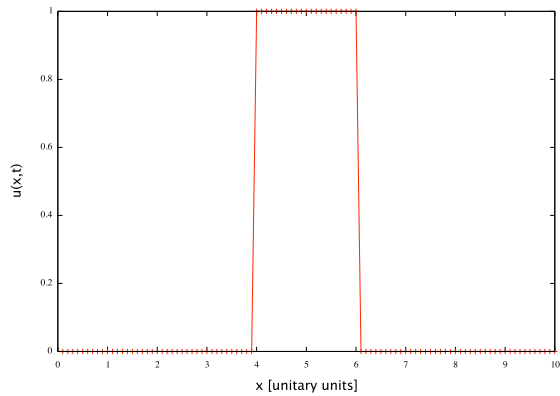
Perform the time integration using the following schemes:

1. FTCS;
2. Lax-Friedrichs;
3. Leapfrog;
4. Lax-Wendroff.

Use Courant factors $c_f = 0.5$ and $c_f = 1$ and compare the results obtained with the different methods, paying attention to their stability and dissipation properties (e.g., plot $u(x, t)$ at different times and the evolution of the L_2 -norm). Use at least $J = 101$ points in the x direction, so that the spacing δx is at least 0.1, and terminate your simulation at $t = 20$. Use periodic boundary conditions.



(a) Plot of (1)



(b) Plot of (2)

We give now some basic information and definitions that will be useful in the following sections. First of all, in discussing the algorithms used to solve the advection equation, the following notation will be used:

$$u_j^n \equiv u(x_j, t^n) \quad (3)$$

Secondary, the periodic boundary conditions are given by the following relations:

$$\begin{cases} u_0^n = u_J^n \\ u_{J+1}^n = u_1^n \end{cases} \quad (4)$$

while the L2-norm is the defined as

$$\|u_j^n\|_2 \equiv \sqrt{\frac{1}{J} \sum_{i=1}^J |u_i^n|^2} \quad (5)$$

Once chosen the space step Δx , the time step has always been obtained using the *CFL-condition*:

$$\Delta t = c_f \frac{\Delta x}{|a|} \quad (6)$$

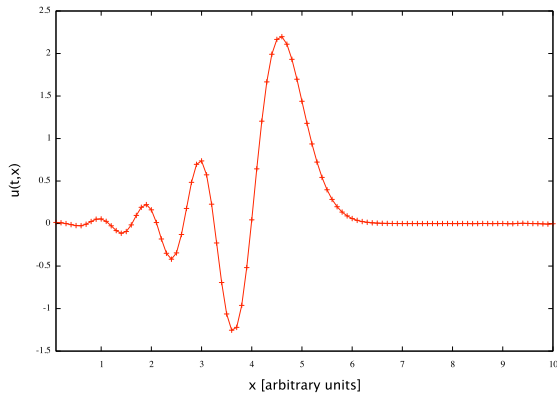
where, in our particular case, $|a| = 1$, and c_f is the Courant factor that has to obey the constraint $c_f \leq 1$.

2 The FTCS method

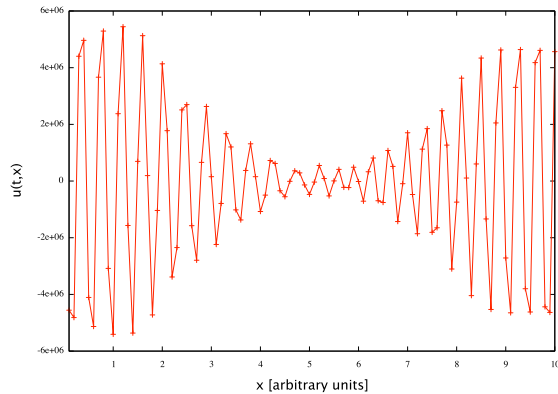
The FTCS method (*Forward in Time Centered in Space*) is a first order method in time and a second order method in space, based on the following algorithm:

$$u_j^{n+1} = u_j^n - a \frac{\Delta t}{\Delta x} (u_{j+1}^n - u_{j-1}^n) \quad (7)$$

However, this method is unstable and therefore is useless for simulations that last in time. With the initial condition defined in (1), the following results are obtained for the evolution of the wave in the case of a $J = 101$ points grid and $c_f = 0.5$:

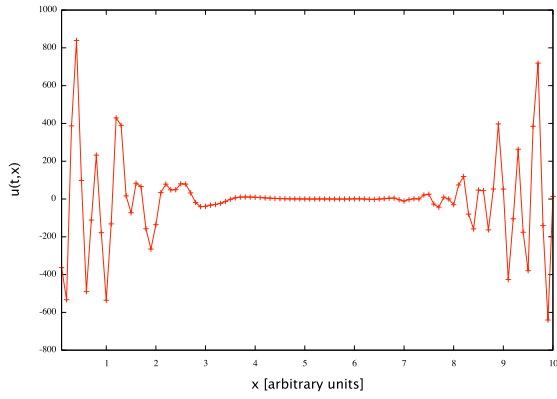


(c) $u(x, t = 10)$ with $c_f = 0.5$

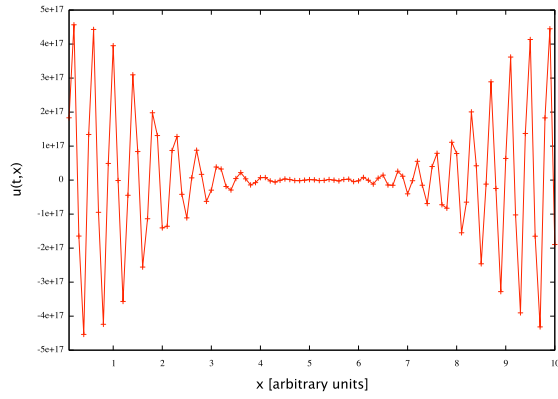


(d) $u(x, t = 20)$ with $c_f = 0.5$

In the case of $J = 101$ points in the grid and $c_f = 1$, instead, the following results are obtained:

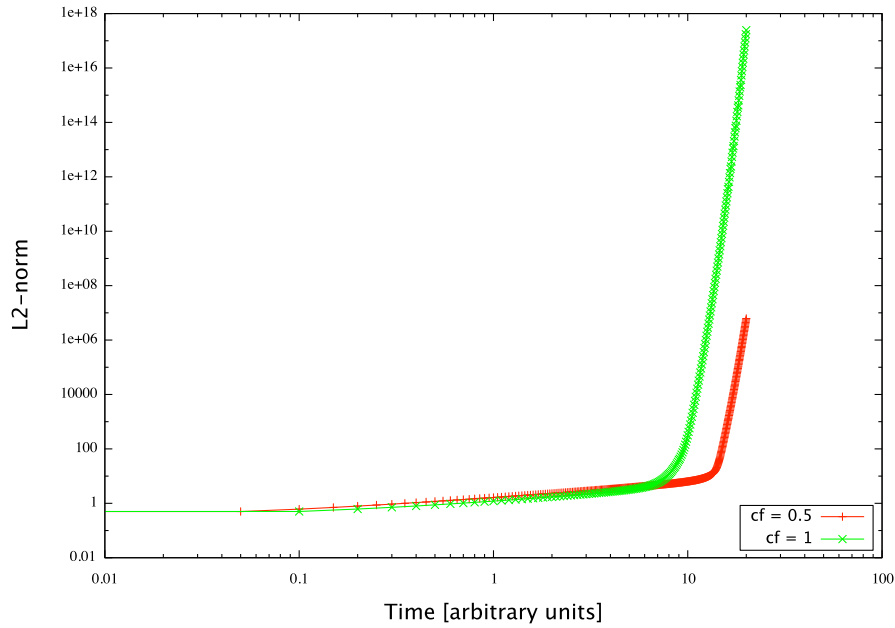


(e) $u(x, t = 10)$ with $c_f = 1$



(f) $u(x, t = 20)$ with $c_f = 1$

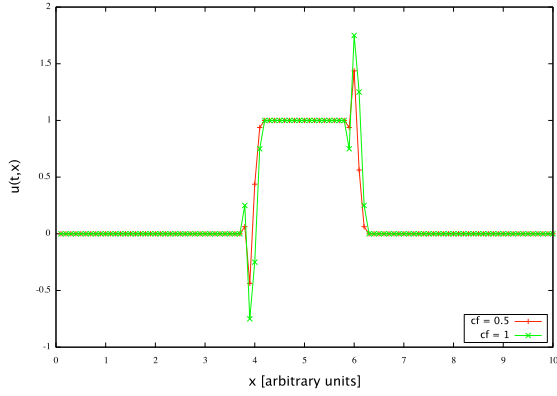
As one can easily figure out from the plots, the instability of the method completely destroys the solution even within $t = 10$, that means that the method is quite useless for long lasting time evolutions, as initially stated. Another confirmation of this fact comes from the behaviour of the L2-norm in terms of time. It can clearly be observed that the norm is



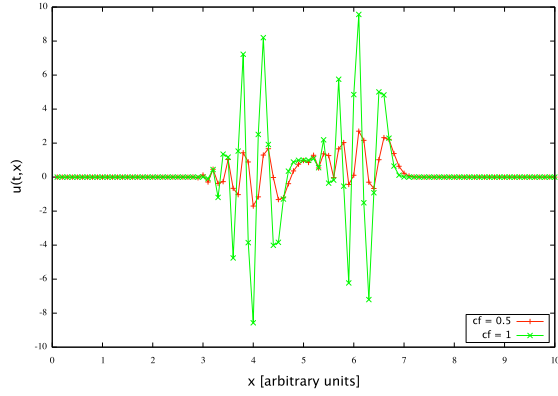
monotonously increasing and that it blows-up for big t 's (with both $c_f = 0.5$ and $c_f = 1$), that is the error on the solution becomes infinite within a finite time.

Different solutions of the advection equation can be obtained using as starting function (2). The computed results are shown in the following pictures. As can be clearly seen from the preceding plots, the solution is already destroyed within the very first seconds of the simulation.

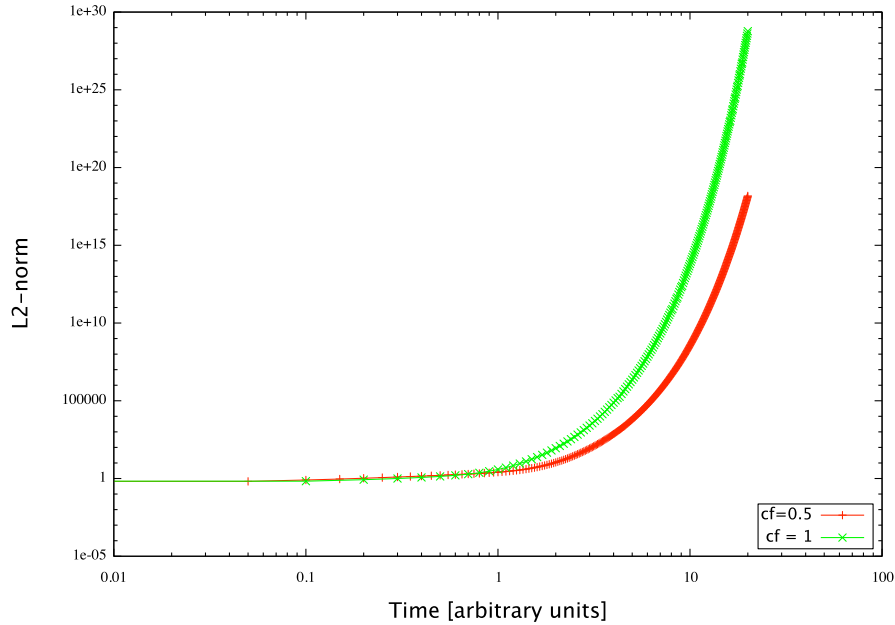
As a confirmation of this observation we can show the plot of the L2-norm in this case. The value reached by the L2-norm at $t = 20$ is even bigger then before, showing, definitively, that the FTCS is a non applicable method for the solution of the advection equation.



(g) $u(x, t = 0.05)$ with $c_f = 0.5, 1$



(h) $u(x, t = 1)$ with $c_f = 0.5, 1$

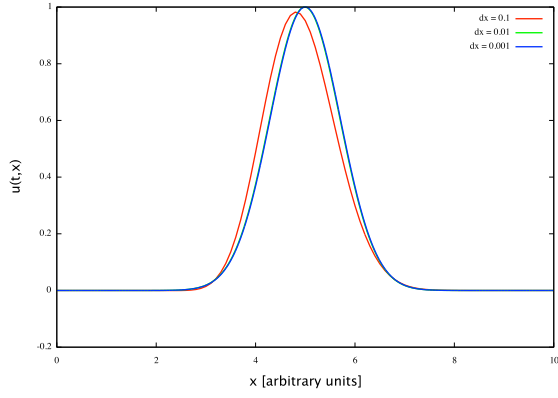


3 The Leapfrog method

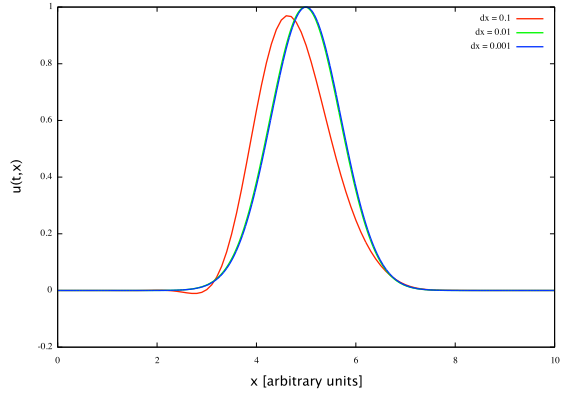
The Leapfrog algorithm states that the solution for the advection equation can be obtained as follows:

$$u_j^{n+1} = u_j^{n-1} - a \frac{\Delta t}{\Delta x} (u_{j+1}^n - u_{j-1}^n) \quad (8)$$

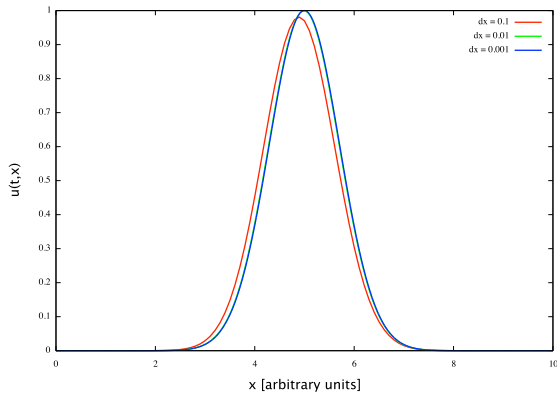
As can be easily seen, one of the fundamental differences between this and the former method lies in the fact that one needs to double the storage space in order to compute the solution, because the $n+1$ step depends on the n and the $n-1$ steps. Using the Von Neumann stability analysis one finds that leapfrog is a stable method as long as $|a| \frac{\Delta t}{\Delta x} = c_f < 1$. Therefore, let's look at the obtained solutions implementing the leapfrog algorithm in different conditions. One sees that the method is indeed stable, and the errors for $J = 101$ are due only to the fact that the spacing of the mesh is too big. About the square wave initial condition the following results have been obtained. It is clear that leapfrog is not a good method to integrate advection equation in case of square wave initial condition. The problem lies in the fact that the algorithm does not take care of the discontinuity in the derivative of the initial condition, therefore propagating the wave wrongly. Even with a different Courant factor the



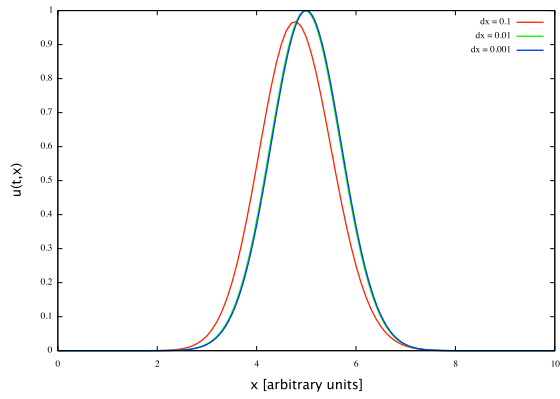
(i) $u(x, t = 10)$ with $c_f = 0.5$



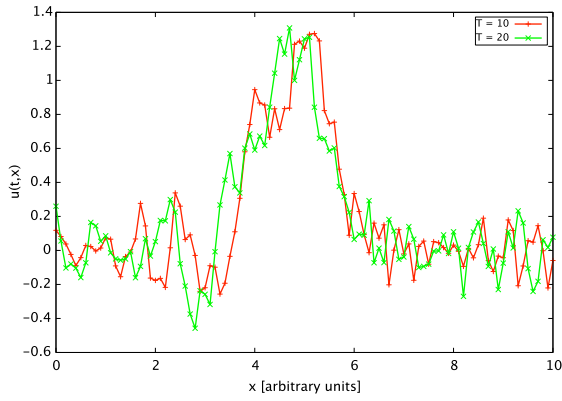
(j) $u(x, t = 20)$ with $c_f = 0.5$



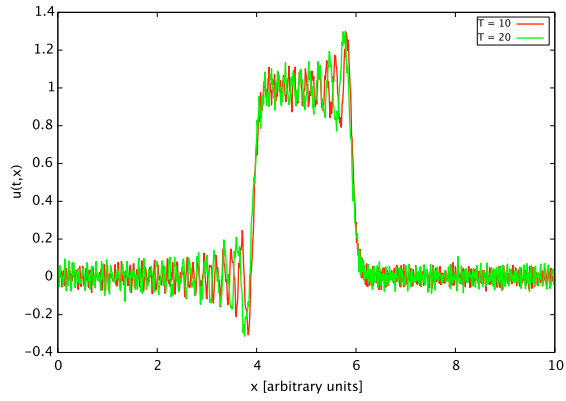
(k) $u(x, t = 10)$ with $c_f = 1$



(l) $u(x, t = 20)$ with $c_f = 1$

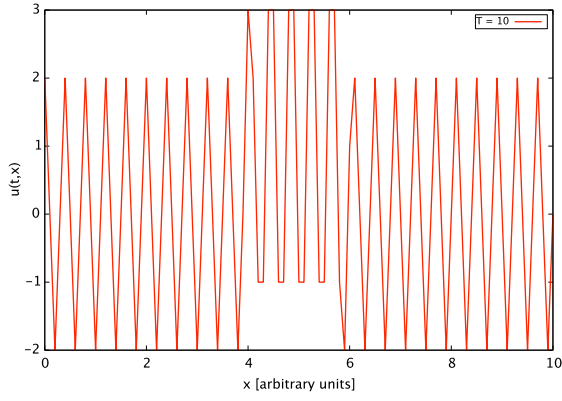


(m) $u(x, t)$ with $c_f = 0.5$ and $\Delta x = 0.1$

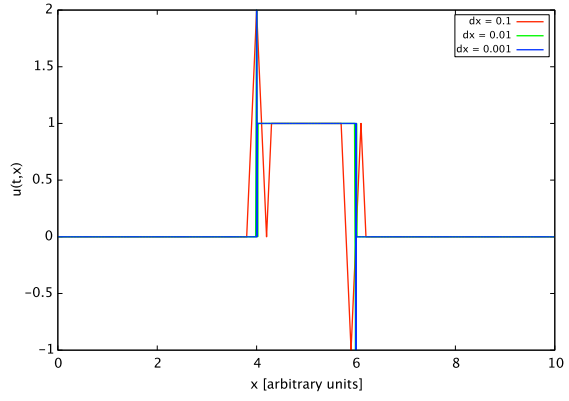


(n) $u(x, t = 10)$ with $c_f = 0.5$ and $\Delta x = 0.01$

problem is not solved. Therefore, the Leapfrog algorithm with $\Delta x \leq 0.01$ and gaussian initial condition is a reliable method to solve the advection equation, while with square wave initial condition it is not. This fact is underlined by analyzing the L2-norm of the algorithm: [da fare]



(o) $u(x, t = 10)$ with $c_f = 1$ and $\Delta x = 0.1$

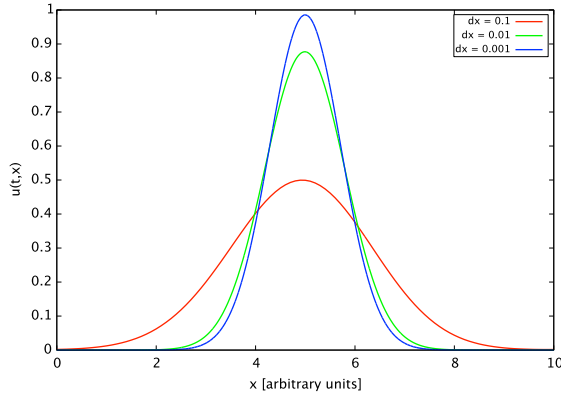


(p) $u(x, t = 20)$ with $c_f = 1$

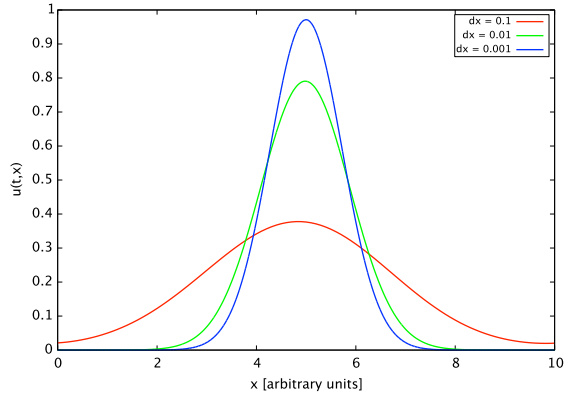
4 The Lax-Friedrichs method

The Lax-Friedrichs method is a stable method (under the condition $|a| \frac{\Delta t}{\Delta x} \leq 1$) to solve the advection equation described by the following algorithm:

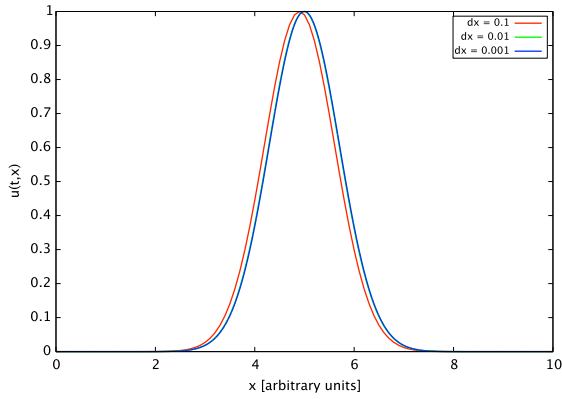
$$u_j^{n+1} = \frac{1}{2}(u_{j-1}^n + u_{j+1}^n) - a \frac{\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) \quad (9)$$



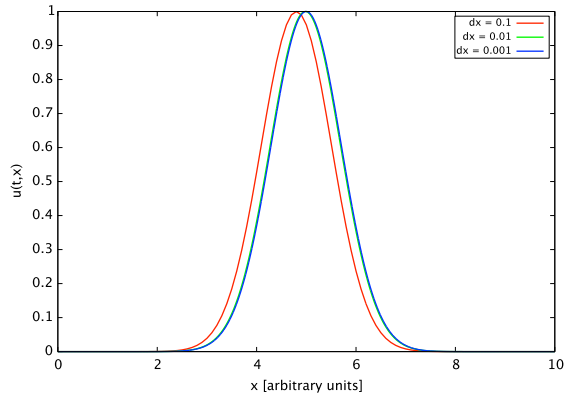
(q) $u(x, t = 10)$ with $c_f = 0.5$



(r) $u(x, t = 20)$ with $c_f = 0.5$



(s) $u(x, t = 10)$ with $c_f = 1$



(t) $u(x, t = 20)$ with $c_f = 1$

where the first factor on the right-hand side of (9) is obtained by substituting u_j^n with its mean

value. Let's look at some results obtained implementing the algorithm in different conditions (Fig. (q), (r), (s), (t)).