



HAL
open science

How to overcome the Courant-Friedrichs-Lewy condition of explicit discretizations?

Denys Dutykh

► **To cite this version:**

Denys Dutykh. How to overcome the Courant-Friedrichs-Lewy condition of explicit discretizations?. [Research Report] CNRS - LAMA UMR 5127, University Savoie Mont Blanc. 2016. hal-01401125

HAL Id: hal-01401125

<https://hal.science/hal-01401125>

Submitted on 24 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

Denys DUTYKH

CNRS–Université Savoie Mont Blanc, France

HOW TO OVERCOME THE
COURANT–FRIEDRICHS–LEWY
CONDITION OF EXPLICIT
DISCRETIZATIONS?

LAST MODIFIED: November 24, 2016

HOW TO OVERCOME THE COURANT–FRIEDRICHS–LEWY CONDITION OF EXPLICIT DISCRETIZATIONS?

DENYS DUTYKH*

ABSTRACT. This manuscript contains some thoughts on the discretization of the classical heat equation. Namely, we discuss the advantages and disadvantages of explicit and implicit schemes. Then, we show how to overcome some disadvantages while preserving some advantages. However, since there is no free lunch, there is a price to pay for any improvement in the numerical scheme. This price will be thoroughly discussed below.

In particular, we like explicit discretizations for the ease of their implementation even for nonlinear problems. Unfortunately, when these schemes are applied to parabolic equations, severe stability limits appear for the time step magnitude making the explicit simulations prohibitively expensive. Implicit schemes remove the stability limit, but each time step requires now the solution of linear (at best) or even nonlinear systems of equations. However, there exists a number of tricks to overcome (or at least to relax) severe stability limitations of explicit schemes without going into the trouble of fully implicit ones. The purpose of this manuscript is just to inform the readers about these alternative techniques to extend the stability limits. It was not written for classical scientific publication purposes.

Key words and phrases: Heat equation; Finite differences; Explicit schemes; Implicit schemes; CFL condition

MSC: [2010] 65M70, 65N35 (primary), 80M22, 76M22 (secondary)

PACS: [2010] 47.11.Kb (primary), 44.35.+c (secondary)

Key words and phrases. Heat equation; Finite differences; Explicit schemes; Implicit schemes; CFL condition.

* Corresponding author.

CONTENTS

1	Introduction	4
1.1	Some healthy criticism	5
2	Classical numerical schemes	6
2.1	The Explicit scheme	6
2.2	The Implicit scheme	7
2.3	The Leap-frog scheme	8
2.4	The Crank–Nicholson scheme	8
	Some nonlinear extensions	10
2.5	Information propagation speed	11
3	Improved explicit schemes	11
3.1	Dufort–Frankel method	12
3.2	Saul’yev method	13
	Resolution procedure	14
3.3	Hyperbolization method	16
	Dispersion relation analysis	16
	Discretization	17
	Error estimate	18
4	Discussion	18
	Acknowledgments	19
	References	19

1. Introduction

In this text we consider the classical linear heat equation:

$$u_t = \nu \nabla^2 u, \quad (1.1)$$

where $u(\mathbf{x}, t)$ is a quantity being diffused in some domain $\Omega \subseteq \mathbb{R}^d$. In physical applications $u(\mathbf{x}, t)$, $\mathbf{x} \in \Omega$, $t > 0$ may represent the temperature field, moisture content, vapor concentration, *etc.* and $\nu > 0$ is the diffusion coefficient. The subscripts denote partial derivatives, *i.e.* $u_t \stackrel{\text{def}}{=} \frac{\partial u(\mathbf{x}, t)}{\partial t}$. Finally, $\nabla^2 \equiv \nabla \cdot \nabla$ is the classical d -dimensional LAPLACE operator:

$$\nabla^2 \stackrel{\text{def}}{=} \sum_{i=1}^d \frac{\partial^2}{\partial x_i^2}.$$

The derivation of this equation for BROWNIAN motion process was given by A. EINSTEIN [6] in 1905.

From now on we shall restrict our ambitions on the 1-dimensional case where $\Omega \equiv [0, \ell] \subseteq \mathbb{R}^1$ and equation (1.1) correspondingly becomes:

$$u_t = \nu u_{xx}. \quad (1.2)$$

This equation has to be supplemented by one initial

$$u|_{t=0} = u_0(x),$$

and two boundary conditions:

$$\Phi_1(t, u(t, 0), u_x(t, 0)) = 0, \quad (1.3)$$

$$\Phi_r(t, u(t, \ell), u_x(t, \ell)) = 0. \quad (1.4)$$

The functions $\Phi_{1,r}(\bullet)$ have to be specified depending on the practical situation in hands. For example, if we have the DIRICHLET-type condition on the left boundary then

$$\Phi_1(t, u(t, 0), u_x(t, 0)) \equiv u(t, 0) - u^\circ(t) = 0,$$

where $u^\circ(t)$ is a prescribed function of time. Often, it is assumed that $u^\circ(t) \equiv \text{const}$. The homogeneous NEUMANN-type condition on the right looks like

$$\Phi_r(t, u(t, \ell), u_x(t, \ell)) \equiv u_x(t, \ell) = 0.$$

1.1. Some healthy criticism

Obviously, the heat equation (1.1) is a simplified model obtained after a series of idealizations and simplifications. As a result, equation (1.1) is linear and its GREEN function* can be computed analytically†

$$G(x, t) = \frac{1}{\sqrt{4\pi\nu t}} e^{-\frac{x^2}{4\nu t}}$$

In particular, one can see that for any sufficiently small $t > 0$ the function $G(x, t)$ is not of *compact support*. In other words, the information about a point source initially localized at $x = 0$ spreads instantly over the whole domain. Of course, the infinite speed of information propagation is physically forbidden. This non-physical feature of heat equation's solutions is a consequence of simplifying assumptions made during the derivation. We just mention here that some nonlinear versions of the heat equation do have fundamental solutions with compact support.

However, in practice the solutions to Partial Differential Equations (PDEs) such as (1.1) are computed numerically than constructed analytically. That is why we can hope to correct some non-physical features of the heat equation solution at the discrete level. This is the main topic of the present manuscript.

Manuscript organization. Below we shall review some classical numerical schemes for the heat equation (1.1) in one spatial dimension in Section 2:

- The explicit scheme in Section 2.1
- The implicit scheme in Section 2.2
- The leap-frog scheme in Section 2.3
- The CRANK–NICHOLSON scheme in Section 2.4

In that Section we justify our preference for explicit schemes in time. However, explicit schemes are known to have an important CFL-type stability restrictions on the time step [5]. That is why in Section 3 we review some not so widely known schemes which allow to overcome the stability limit while still being explicit in time. In particular, we consider the following alternatives:

- DUFORT–FRANKEL method in Section 3.1
- SAULYEV method in Section 3.2
- Hyperbolization method in Section 3.3

*The GREEN function is also known as the fundamental solution. More precisely, it solves the following problem:

$$\begin{aligned} G_t &= \nu G_{xx}, & x &\in \mathbb{R}, \\ G|_{t=0} &= \delta(x), \end{aligned}$$

where $\delta(x)$ is DIRAC delta function.

†This expression is false. To be checked later!

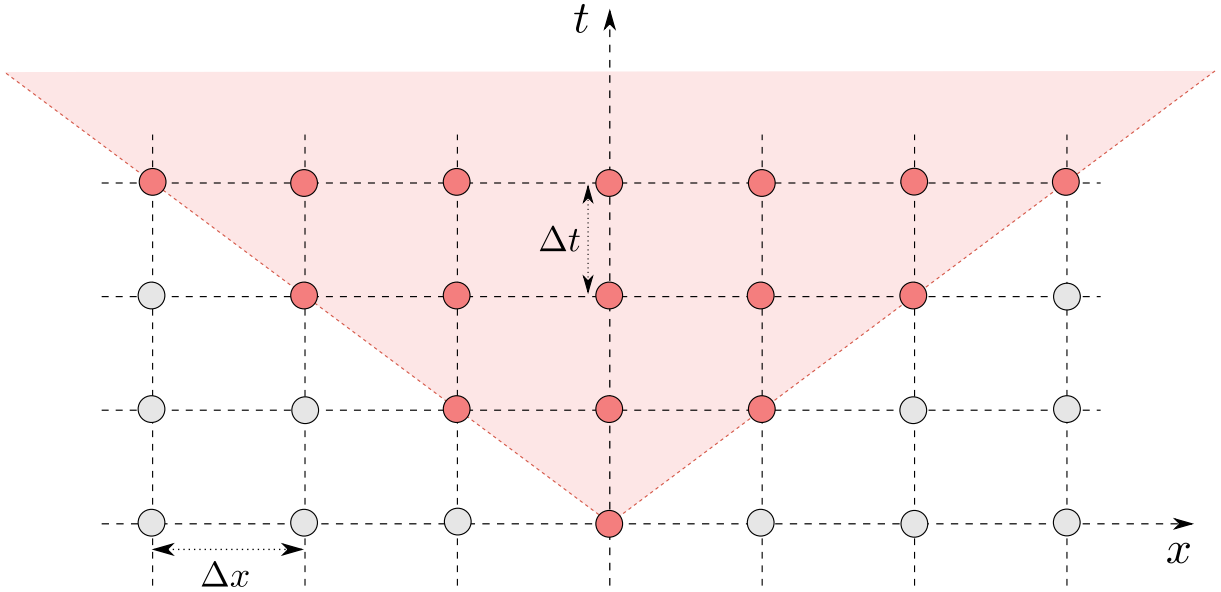


Figure 1. A schematic representation of the uniform discretization in space and time. Red nodes correspond to non-zero values of the discrete solution. Grey nodes correspond to $u_j^n \equiv 0$. The shaded area is an equivalent of the ‘light cone’ for the initially activated node.

Finally, the main conclusions and perspectives of the present study are discussed in Section 4.

2. Classical numerical schemes

In order to describe numerical schemes in simple terms, consider a uniform discretization of the interval $\Omega \rightsquigarrow \Omega_h$:

$$\Omega_h = \bigcup_{j=0}^{N-1} [x_j, x_{j+1}], \quad x_{j+1} - x_j \equiv \Delta x, \quad \forall j \in \{0, 1, \dots, N-1\}.$$

The time layers are uniformly spaced as well $t^n = n \Delta t$, $\Delta t = \text{const} > 0$, $n = 0, 1, 2, \dots$. The values of function $u(x, t)$ in discrete nodes will be denoted by $u_j^n \stackrel{\text{def}}{=} u(x_j, t^n)$. The space-time grid is schematically depicted in Figure 1.

2.1. The Explicit scheme

The standard explicit scheme for the linear heat equation (1.2) can be written as

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2}, \quad j = 1, \dots, N-1, \quad n \geq 0. \quad (2.1)$$

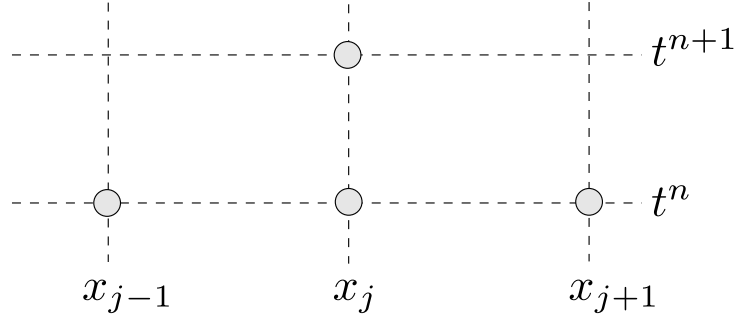


Figure 2. Stencil of the explicit finite difference scheme (2.1).

The stencil of this scheme is depicted in Figure 2. In order to complete this discretization we have to find from boundary conditions (1.3), (1.4) the boundary values:

$$u_0^{n+1} = \psi_l(t^{n+1}, u_1^{n+1}, \dots), \quad u_N^{n+1} = \psi_r(t^{n+1}, u_{N-1}^{n+1}, \dots), \quad (2.2)$$

where functions $\psi_{l,r}(\bullet)$ may depend on adjacent values of the solution whose number depends on the approximation order of the scheme (here we use the second order in space). For example, if the temperature is prescribed on the right boundary, then we simply have

$$u_N^{n+1} = \psi_r(t^{n+1}) \equiv \phi_r(t^{n+1}),$$

where $\phi_r(t)$ is a given function of time. On the other hand, if the heat flux is prescribed on the left boundary $\nu \frac{\partial u}{\partial x} = \phi_l(t)$ then it can be discretized as

$$\nu \frac{-3u_0^{n+1} + 4u_1^{n+1} - u_2^{n+1}}{2\Delta x} = \phi_l(t^{n+1}).$$

By solving (2.1) with respect to u_j^{n+1} we obtain a discrete dynamical system

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

whose starting value is directly obtained from the initial condition:

$$u_j^0 = u_0(x_j), \quad j = 0, 1, \dots, N.$$

It is well-known that scheme (2.1) approximates the continuous operator to order $\mathcal{O}(\Delta t + \Delta x^2)$. The explicit scheme is conditionally stable under the following CFL-type condition:

$$\Delta t \leq \frac{1}{2\nu} \Delta x^2. \quad (2.3)$$

Unfortunately, this condition is too restrictive for sufficiently fine discretizations.

2.2. The Implicit scheme

The implicit scheme for the 1D heat equation (1.2) is given by the following relations:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{\Delta x^2}, \quad j = 1, \dots, N-1, \quad n \geq 0. \quad (2.4)$$

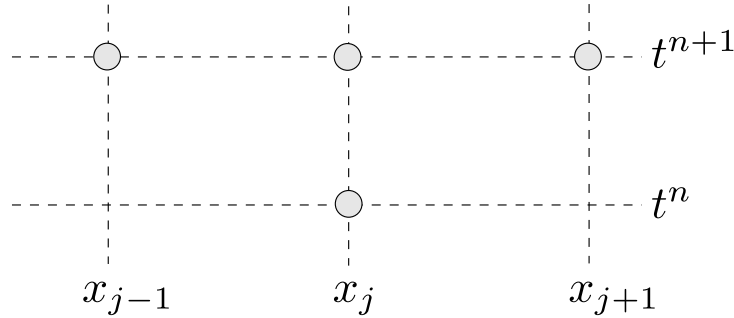


Figure 3. Stencil of the implicit finite difference scheme (2.4).

The finite difference stencil of this scheme is depicted in Figure 3. These relations have to be properly initialized and supplemented with numerical boundary conditions (2.2). In the following Sections we shall not return to the question of initial and boundary conditions in order to focus on the discretization. The scheme (2.4) has the same order of accuracy as the explicit scheme (2.1), *i.e.* $\mathcal{O}(\Delta t + \Delta x^2)$. However, the implicit scheme (2.4) is unconditionally stable, which constitutes its major advantage. It could be interesting to have also the second order in time as well. This issue will be addressed in the following Sections.

The most important difference with the explicit scheme (2.1) is that we have to solve a tridiagonal system of linear algebraic equations to determine the numerical solution values $\{u_j^{n+1}\}_{j=0}^N$ on the following time layer $t = t^{n+1}$. It determines the algorithm complexity — a tridiagonal system of equations can be solved in $\mathcal{O}(N)$ operations (using the simple THOMAS algorithm, for example) and it has to be done at every time step.

2.3. The Leap-frog scheme

The leap-frog scheme* is obtained by replacing in (2.1) the forward difference in time by the symmetric one, *i.e.*

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = \nu \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta x^2}, \quad j = 1, \dots, N-1, \quad n \geq 0. \quad (2.5)$$

This scheme is second order accurate in space and in time, *i.e.* $\mathcal{O}(\Delta t^2 + \Delta x^2)$. Unfortunately, the leap-frog scheme is unconditionally unstable. It makes it un-exploitable in practice. However, we shall use some modifications of this scheme below.

2.4. The Crank–Nicholson scheme

We saw above that the first tentative to obtain a scheme with second order accuracy in space *and* in time was unsuccessful (see Section 2.3). However, a very useful method was

*This scheme is called in French as ‘*le schéma saute-mouton*’.

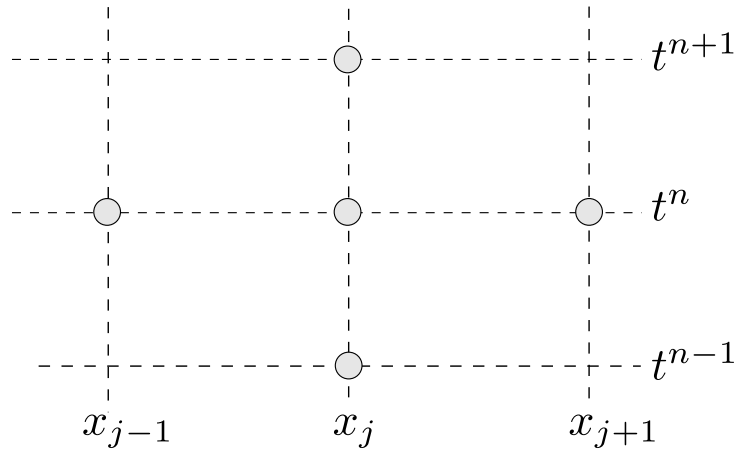


Figure 4. Stencil of the leap-frog (2.5) and hyperbolic (3.9) finite difference schemes.

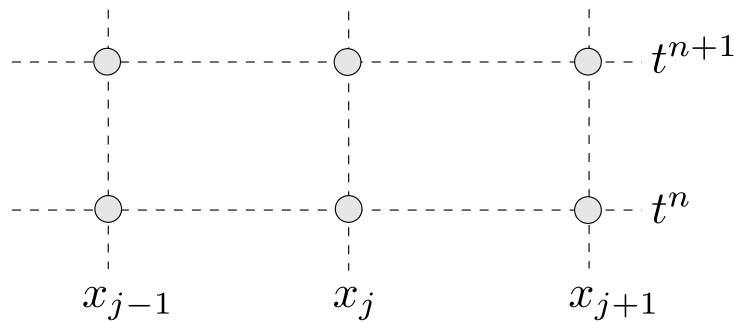


Figure 5. Stencil of the CRANK-NICHOLSON (CN) finite difference scheme (2.6).

proposed by CRANK & NICHOLSON (CN) and it can be successfully applied to the heat equation (1.2) as well:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{2\Delta x^2} + \nu \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{2\Delta x^2},$$

$$j = 1, \dots, N-1, \quad n \geq 0. \quad (2.6)$$

This scheme is $\mathcal{O}(\Delta t^2 + \Delta x^2)$ accurate and unconditionally stable (similarly to (2.4)). That is why numerical results obtained with the CN scheme will be more accurate than implicit scheme (2.4) predictions. The stencil of this scheme is depicted in Figure 5. The CN scheme has all advantages and disadvantages (except for the order of accuracy in time) of the implicit scheme (2.4). At every time step one has to use a tridiagonal solver to invert the linear system of equations to determine solution value at the following time layer $t = t^{n+1}$.

2.4.1 Some nonlinear extensions

Since most of real-world heat conduction models used in building physics are nonlinear, it is worth to discuss some nonlinear extensions of the CRANK–NICHOLSON (CN) scheme. For linear problems CN scheme turns out to be the same as the mid-point and trapezoidal rules for Ordinary Differential Equations (ODEs). Indeed, consider a nonlinear ODE:

$$\dot{u} = f(u), \quad u(0) = u_0. \quad (2.7)$$

The mid-point and trapezoidal rules consist correspondingly in discretizing (2.7) as follows:

$$\begin{aligned} \frac{u^{n+1} - u^n}{\Delta t} &= f\left(\frac{u^n + u^{n+1}}{2}\right), \\ \frac{u^{n+1} - u^n}{\Delta t} &= \frac{f(u^n) + f(u^{n+1})}{2}. \end{aligned}$$

Now, if we set in formulas above $f(u) = \nu \mathcal{L} \cdot u$, where $\mathcal{L} \simeq \partial_{xx}$ is the linear operator which represents the second central finite difference, we recover the CN scheme (2.6).

Consider a non-conservative nonlinear heat equation:

$$u_t = k(u) u_{xx}. \quad (2.8)$$

The straightforward application of the CN scheme to equation (1.1) yields the following scheme:

$$\begin{aligned} \frac{u_j^{n+1} - u_j^n}{\Delta t} &= k(u_j^n) \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{2\Delta x^2} + k(u_j^{n+1}) \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{2\Delta x^2}, \\ & \quad j = 1, \dots, N-1, \quad n \geq 0. \end{aligned} \quad (2.9)$$

However, it is less known that one can apply also the cross-CRANK–NICHOLSON (cCN) scheme:

$$\begin{aligned} \frac{u_j^{n+1} - u_j^n}{\Delta t} &= k(u_j^{n+1}) \frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{2\Delta x^2} + k(u_j^n) \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{2\Delta x^2}, \\ & \quad j = 1, \dots, N-1, \quad n \geq 0. \end{aligned} \quad (2.10)$$

We underline that both schemes (2.9) and (2.10) are second order accurate in space *and* in time, *i.e.* the consistency error is $\mathcal{O}(\Delta t^2 + \Delta x^2)$. However, there is a major advantage of the cCN scheme (2.10) over the classical CN scheme (2.9) in the fact that cCN is linear with respect to quantities evaluated at the upcoming time layer $t = t^{n+1}$ provided that $k(u)$ is an affine function of u . This fact can be used to simplify the resolution procedure without destroying the accuracy of the CN scheme. Otherwise, for more general diffusion coefficients $k(u)$ the success of operation depends on the easiness to solve nonlinear equation (2.10). It goes without saying that information propagates instantaneously in both CN and cCN schemes.

2.5. Information propagation speed

Let us discuss now an important issue of the information propagation speed in the discretized heat equation (1.2). As the initial condition we take the following grid function:

$$u_j^0 = \begin{cases} 1, & j = 0, \\ 0, & j \neq 0, \end{cases}$$

which corresponds to the discrete DIRAC function. In all fully implicit schemes (such as (2.4) and (2.6)) the grid function $\{u_j^1\}_{j=0}^N$ will generally have non-zero values in all nodes (modulo perhaps homogeneous boundary conditions). Thus, we can conclude that information has spreaded instantaneously. On the other hand, as it is illustrated in Figure 1 with grey and red circles, in explicit discretizations the information propagates one cell to the left and one cell to the right in one time step. Thus, its speed c_s can be estimated as

$$c_s = \frac{\Delta x}{\Delta t} \underbrace{\geq}_{\text{CFL}} \frac{2\nu}{\Delta x}.$$

Thus, the value of c_s is finite. Of course, in the limit $\Delta x \rightarrow 0$ we recover the infinite propagation speed, but let us not forget that computations are always run for a *finite* value of Δx .

We arrived to an interesting conclusion. Even if the continuous heat equation (1.1) possesses an unphysical property, it can be corrected if we use a judicious (in this case *explicit*) discretization. This is the main reason why we privilege explicit schemes in time. However, these schemes are subject to severe stability restrictions. The rest of the manuscript is devoted to the question how to overcome the stability limit?

There is another computational advantage of explicit schemes over the implicit ones. Namely, explicit methods can be easily parallelized and they allow to achieve almost perfect scaling on HPC systems [3]. Indeed, the computational domain can be split into sub-domains, each sub-domain being handled by a separate processor. Since the stencil is local, only direct neighbours are involved in individual computations. The communication among various processes is almost minimal since only boundary nodes have to be shared. This is another good reason to privilege explicit schemes over the implicit ones.

3. Improved explicit schemes

Below we present several alternative methods which were specifically designed to overcome the stability limitation of the standard explicit scheme (2.1).

3.1. Dufort–Frankel method

Let us take the unconditionally unstable leap-frog scheme (2.5) and slightly modify it to obtain the so-called DUFORT–FRANKEL method:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2 \Delta t} = \nu \frac{u_{j-1}^n - (u_j^{n-1} + u_j^{n+1}) + u_{j+1}^n}{\Delta x^2}, \quad j = 1, \dots, N-1, \quad n \geq 0, \quad (3.1)$$

where we made a replacement

$$2 u_j^n \leftrightarrow u_j^{n-1} + u_j^{n+1}.$$

The scheme (3.1) has the stencil depicted in Figure 6. At the first glance the scheme (3.1) looks like an implicit scheme, however, it is not truly the case. Equation (3.1) can be easily solved for u_j^{n+1} to give the following discrete dynamical system:

$$u_j^{n+1} = \frac{1 - \lambda}{1 + \lambda} u_j^{n-1} + \frac{\lambda}{1 + \lambda} (u_{j+1}^n + u_{j-1}^n),$$

where

$$\lambda \stackrel{\text{def}}{=} 2 \nu \frac{\Delta t}{\Delta x^2}.$$

The standard VON NEUMANN stability analysis shows that the DUFORT–FRANKEL scheme is *unconditionally stable*.

The consistency error analysis of the scheme (3.1) shows the following interesting result:

$$\begin{aligned} \mathcal{L}_j^n &= \underbrace{\nu \frac{\Delta t^2}{\Delta x^2}}_{\equiv \tau} u_{tt} + \underbrace{u_t - \nu u_{xx}}_{(1.2)} + \\ &\quad \frac{1}{6} \Delta t^2 u_{ttt} - \frac{1}{12} \nu \Delta x^2 u_{xxxx} - \frac{1}{12} \nu \Delta t^2 \Delta x u_{xxx} + \mathcal{O}\left(\frac{\Delta t^4}{\Delta x^2}\right), \end{aligned}$$

where

$$\mathcal{L}_j^n \stackrel{\text{def}}{=} \frac{u_j^{n+1} - u_j^{n-1}}{2 \Delta t} - \nu \frac{u_{j-1}^n - (u_j^{n-1} + u_j^{n+1}) + u_{j+1}^n}{\Delta x^2}.$$

So, from the asymptotic expansion for \mathcal{L}_j^n we obtain that the DUFORT–FRANKEL scheme is second order accurate in time *and*

- First order accurate in space if $\Delta t \propto \Delta x^{3/2}$
- Second order accurate in space if $\Delta t \propto \Delta x^2$

However, the DUFORT–FRANKEL scheme is unconditionally consistent with the so-called *hyperbolic heat conduction equation*:

$$\tau u_{tt} + u_t - \nu u_{xx} = 0.$$

We shall return to this equation below. At this stage we only mention that information propagates with the finite speed in hyperbolic models.

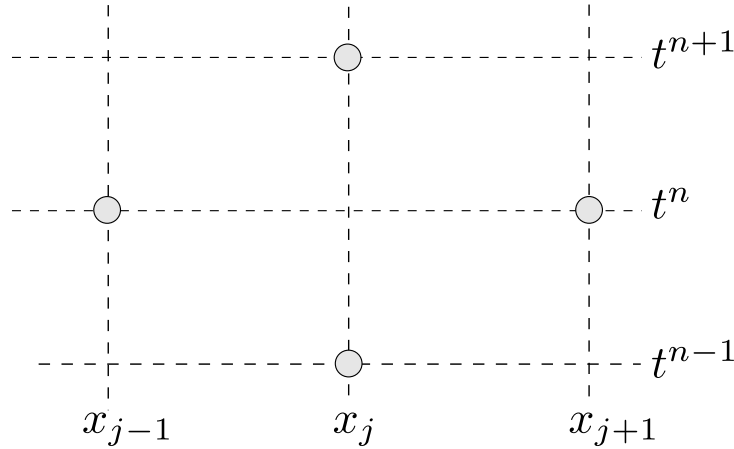


Figure 6. Stencil of the DUFORT–FRANKEL (3.1) finite difference scheme.

3.2. Saulyev method

In this Section we describe a not so widely known method proposed by SAULYEV in [8] to integrate parabolic equations. For simplicity we focus on the 1–dimensional heat equation (1.2). The first idea of this method consists in rewriting the discrete second (spatial) derivative as

$$u_{xx}|_{x=x_j} \approx \frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta x^2} \equiv \frac{\frac{u_{j+1} - u_j}{\Delta x} - \frac{u_j - u_{j-1}}{\Delta x}}{\Delta x}.$$

In the finite difference formula above we do not specify intentionally the time layer number. The next trick consists in writing the following asymmetric finite difference approximation:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \frac{\frac{u_{j+1}^n - u_j^n}{\Delta x} - \frac{u_j^{n+1} - u_{j-1}^{n+1}}{\Delta x}}{\Delta x},$$

or after simplifications we obtain

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \frac{u_{j+1}^n - (u_j^n + u_j^{n+1}) + u_{j-1}^{n+1}}{\Delta x^2}. \quad (3.2)$$

The last difference relation is slightly different from the DUFORT–FRANKEL method presented above. Moreover, the relation written above is not consistent with the original equation (1.2). That is why we consider the next time layer $t = t^{n+2}$ and we apply symmetrically the same tricks, *i.e.*

$$\frac{u_j^{n+2} - u_j^{n+1}}{\Delta t} = \nu \frac{\frac{u_{j+1}^{n+2} - u_j^{n+2}}{\Delta x} - \frac{u_j^{n+1} - u_{j-1}^{n+1}}{\Delta x}}{\Delta x},$$

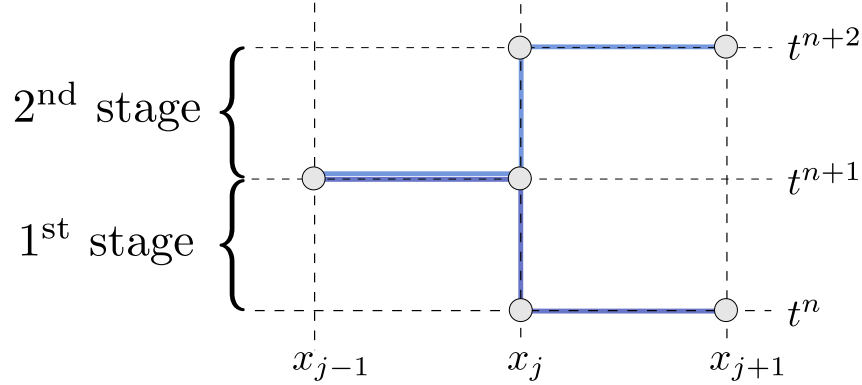


Figure 7. Stencil of the SAULYEV finite difference scheme, which consists of two stages (3.2) and (3.3).

and after simplifications we have

$$\frac{u_j^{n+2} - u_j^{n+1}}{\Delta t} = \nu \frac{u_{j+1}^{n+2} - (u_j^{n+2} + u_j^{n+1}) + u_{j-1}^{n+1}}{\Delta x^2}. \quad (3.3)$$

Without any surprise the relation (3.3) does not approximate equation (1.2) either. However, both relations (3.2) and (3.3) constitute the so-called SAULYEV scheme and are called the first and second stages of SAULYEV method correspondingly.

In order to perform the approximation error analysis we take the sum of (3.2), (3.3) and we divide it by two:

$$\mathcal{L}_j^n = \frac{u_j^{n+2} - u_j^n}{2 \Delta t} = \nu \frac{u_j^n - u_{j+1}^n - 2u_{j-1}^{n+1} + 2u_j^{n+1} + u_j^{n+2} - u_{j+1}^{n+2}}{2 \Delta x^2}.$$

After applying local TAYLOR expansions we obtain

$$\begin{aligned} \mathcal{L}_j^n &= u_t - \nu u_{xx} - \frac{1}{12} \nu \Delta x^2 u_{xxxx} + \\ &\quad \left[\frac{2}{3} u_{ttt} - \frac{3}{4} \nu u_{xxtt} \right] \Delta t^2 - \frac{1}{2} \nu \frac{\Delta t^2}{\Delta x} u_{xtt} + \mathcal{O}(\Delta t \Delta x^2 + \Delta t^2 \Delta x). \end{aligned}$$

From the last asymptotic expansion we arrive at an important result: SAULYEV scheme is second order accurate in space if $\Delta t = \mathcal{O}(\Delta x^{3/2})$. We underline the fact that this condition coming from the accuracy requirements is weaker than the usual CFL restriction (2.3). For instance, if the user is ready to sacrifice the spatial accuracy to the first order $\mathcal{O}(\Delta x)$, then it is sufficient to take $\Delta t = \mathcal{O}(\Delta x)$.

Without proof we report that SAULYEV's scheme is unconditionally stable (see [8] for more details). The stencil of SAULYEV's scheme is depicted in Figure 7.

3.2.1 Resolution procedure

At the first glance SAULYEV scheme appears as an implicit one since each relation (3.2) and (3.3) contains two terms from the following time layer ($t = t^{n+1}$ and $t = t^{n+2}$

correspondingly). However, this scheme can be recast in an almost explicit form using judicious recurrence relations.

Consider the first stage (3.2) of SAULYEV's scheme. Similarly to Section 3.1 we introduce for simplicity the parameter

$$\lambda \stackrel{\text{def}}{=} \nu \frac{\Delta t}{\Delta x^2}.$$

From difference relation (3.2) we find

$$u_j^{n+1} = \frac{1 - \lambda}{1 + \lambda} u_j^n + \frac{\lambda}{1 + \lambda} u_{j+1}^n + \frac{\lambda}{1 + \lambda} u_{j-1}^n. \quad (3.4)$$

The first stage of SAULYEV's scheme is computed in rightwards direction (increasing index $j \nearrow$). From the left boundary condition we find first the value

$$u_0^{n+1} = \psi_l(t^{n+1}).$$

This allows us to compute u_1^{n+1} , u_2^{n+1} , etc. thanks to the recurrence relation (3.4). At the final step, the value u_N^{n+1} is computed directly from the right boundary condition:

$$u_N^{n+1} = \psi_r(t^{n+1}).$$

This completes the first stage of computations.

Remark 1. For some types of boundary conditions (e.g. ROBIN-type) SAULYEV's scheme might require solution of a small dimensional (typically 2×2) linear system of algebraic equations to initiate the recurrence (3.4).

Let us make explicit now the second stage (3.3) of SAULYEV's method. For this purpose we solve relation (3.3) with respect to u_j^{n+2} :

$$u_j^{n+2} = \frac{1 - \lambda}{1 + \lambda} u_j^{n+1} + \frac{\lambda}{1 + \lambda} u_{j-1}^{n+1} + \frac{\lambda}{1 + \lambda} u_{j+1}^{n+1}. \quad (3.5)$$

Now it is getting clear that during the second stage of SAULYEV's scheme we proceed in the leftwards direction (decreasing $j \searrow$). From the right boundary condition we find first

$$u_N^{n+2} = \psi_r(t^{n+2}).$$

It allows us to compute u_{N-1}^{n+2} , u_{N-2}^{n+2} , etc. using the recurrence relation (3.5). At the final step, the value u_0^{n+2} is computed directly from the left boundary condition:

$$u_0^{n+2} = \psi_l(t^{n+2}).$$

As a result we obtain a fully explicit resolution scheme without stability related limitations. We notice however that SAULYEV's scheme provides consistent results only every second time step or, in other words, after the successive completion of both stages (3.4) and (3.5). The intermediate result is not consistent with the equation (1.2).

3.3. Hyperbolization method

We saw above that the DUFORT–FRANKEL scheme is a hidden way to add a small amount of ‘hyperbolicity’ into the model (1.2). In this Section we shall invert the order of operations: first, we perturb the equation (1.2) in an ad-hoc way and only after we discretize it with a suitable method.

Consider the 1–dimensional heat equation (1.2) that we are going to perturb by adding a small term containing the second derivative in time:

$$\tau u_{tt} + u_t - \nu u_{xx} = 0. \quad (3.6)$$

This is the *hyperbolic heat equation* already familiar to us since it appeared in the consistency analysis of the DUFORT–FRANKEL scheme. Here we perform a singular perturbation by assuming that

$$\|\tau u_{tt}\| \ll \|u_t\|.$$

The last condition physically means that the new term has only limited influence on the solution of equation (3.6). Here τ is a small ad-hoc parameter whose value is in general related to physical and discretization parameters $\tau = \tau(\nu, \Delta x, \Delta t)$.

Remark 2. *One can notice that equation (3.6) is second order in time, thus, it requires two initial conditions to obtain a well-posed initial value problem. However, the parabolic equation (1.2) is only first order in time and it only requires the knowledge of the initial temperature field distribution. When we solve the hyperbolic equation (3.6), the missing initial condition is simply chosen to be*

$$u_t|_{t=0} = 0.$$

3.3.1 Dispersion relation analysis

The classical dispersion relation analysis looks at plane wave solutions:

$$u(x, t) = u_0 e^{i(\kappa x - \omega t)}. \quad (3.7)$$

By substituting this solution ansatz into equation (1.2) we obtain the following relation between wave frequency ω and wavenumber k :

$$\omega(\kappa) = -i\nu\kappa^2. \quad (3.8)$$

The last relation is called the *dispersion relation* even if the heat equation (1.2) is not dispersive but dissipative. The real part of ω contains information about wave propagation properties (dispersive if $\frac{\text{Re}\omega(\kappa)}{\kappa} \neq \text{const}$ and non-dispersive otherwise) while the imaginary part describes how different modes κ dissipate (if $\text{Im}\omega(\kappa) < 0$) or grow (if $\text{Im}\omega(\kappa) > 0$). The dispersion relation (3.8) gives the damping rate of different modes.

The same plane wave ansatz (3.7) can be substituted into the hyperbolic heat equation (3.6) as well to give the following *implicit* relation for the wave frequency ω :

$$-\tau\omega^2 - i\omega + \nu\kappa^2 = 0.$$

By solving this quadratic equation with complex coefficients for ω , we obtain two branches:

$$\omega_{\pm}(\kappa) = \frac{-i \pm \sqrt{4\nu\kappa^2\tau - 1}}{2\tau}.$$

This dispersion relation will be analyzed asymptotically with $\tau \ll 1$ being the small parameter. The branch $\omega_{-}(\kappa)$ is not of much interest to us since it is constantly damped, *i.e.*

$$\omega_{-}(\kappa) = -\frac{i}{\tau} + \mathcal{O}(1).$$

It is much more instructive to look at the positive branch $\omega_{+}(\kappa)$:

$$\omega_{+}(\kappa) = -i\nu\kappa^2 [1 + \nu\kappa^2\tau + 2\nu^2\kappa^4\tau^2 + \mathcal{O}(\tau^3)].$$

The last asymptotic expansion shows that for small values of parameter τ we obtain a valid asymptotic approximation of the dispersion relation (3.8) for the heat equation (1.2).

3.3.2 Discretization

Equation (3.6) will be discretized on the same stencil as the leap-frog scheme (2.5) (see Figure 4):

$$\mathcal{L}_j^n \stackrel{\text{def}}{=} \tau \frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\Delta t^2} + \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} - \nu \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} = 0, \quad j = 1, \dots, N-1, \quad n \geq 0, \quad (3.9)$$

The last scheme is consistent with hyperbolic heat equation (3.6) to the second order in space and in time $\mathcal{O}(\Delta t^2 + \Delta x^2)$. Indeed, using the standard TAYLOR expansions we obtain

$$\begin{aligned} \mathcal{L}_j^n &= \tau u_{tt} + u_t - \nu u_{xx} \\ &\quad - \frac{\nu}{12} \Delta x^2 u_{xxxx} + \Delta t^2 \left[\frac{1}{6} u_{ttt} + \frac{1}{12} \tau u_{tttt} \right] + \mathcal{O}(\Delta t^4 + \Delta x^4). \end{aligned}$$

The stability of the scheme (3.9) was studied in [3] and the following stability condition was obtained:

$$\frac{\Delta t}{\Delta x} \leq \sqrt{\frac{\tau}{\nu}}.$$

By taking, for example, $\tau = \nu \Delta x$ we obtain the following stability condition

$$\Delta t \leq \Delta x^{\frac{3}{2}},$$

which is still weaker than the standard parabolic condition (2.3). However, it was reported in [2, 4] that stable computations (even in 3D) can be performed even with $\Delta t = \mathcal{O}(\Delta x)$. The authors explain informally this experimental observation by the fact that usual stability conditions are too ‘pessimistic’.

Remark 3. The ad-hoc parameter τ can be chosen in other ways as well. One popular choice consists in taking

$$\tau = \frac{\Delta x}{c_s},$$

where c_s is the real physical information speed.

3.3.3 Error estimate

It is legitimate to ask the question how far are solutions $u_h(x, t)$ to the hyperbolic equation (3.6) from the solutions $u_p(x, t)$ of the parabolic heat equation (1.1) (for the same initial condition). This question for the initial value problem was studied in [7] and we shall provide here only the obtained error estimate. Let us introduce the difference between two solutions:

$$\delta u(x, t) \stackrel{\text{def}}{=} u_h(x, t) - u_p(x, t).$$

Then, the following estimate holds

$$|\delta u(x, t)| \leq \tau \mathcal{M} \left(1 + \frac{2}{\sqrt{\pi}} \right) \left(8\sqrt{2}\tau + \frac{\sqrt[4]{2}\pi^2}{2} T \right),$$

where $T > 0$ is the time horizon and

$$\mathcal{M} \stackrel{\text{def}}{=} \sup_{\Omega_{\xi, \zeta}} \left| \frac{\partial^2 u_p}{\partial t^2}(\xi, \zeta) \right|,$$

and the domain $\Omega_{\xi, \zeta}$ is defined as

$$\Omega_{\xi, \zeta} \stackrel{\text{def}}{=} \left\{ (\xi, \zeta) : 0 \leq \zeta \leq t, \quad x - \frac{t - \zeta}{\sqrt{\tau}} \leq \xi \leq x + \frac{t - \zeta}{\sqrt{\tau}} \right\}.$$

4. Discussion

We saw above that only explicit schemes allow to have the finite speed of information propagation in the discretized version of the heat equation (1.1). The ease of parallelization along with excellent scaling properties of the codes obtained with explicit schemes constitute another important advantage to privilege explicit schemes over implicit ones [3]. However, explicit schemes for parabolic equations suffer from very stringent CFL-type conditions $\Delta t = \mathcal{O}(\Delta x^2)$ on the time step. That is why it is almost impossible to perform long time simulations (required in *e.g.* building physics applications) using fully explicit schemes such as (2.1). In order to keep explicit discretizations and overcome stringent CFL-type restrictions, a certain number of *hybrid* schemes were described. These hybrid schemes are based on different ideas. Some schemes rely on the information about the numerical solution on following time layers while keeping the overall scheme explicit using various tricks. Some hybrid schemes (*e.g.* DUFORT–FRANKEL method described in Section 3.1 and SAULYEV’s scheme from Section 3.2) can be even unconditionally stable.

Using the local error analysis we noticed that CFL-improved schemes gain the stability by introducing some weak hyperbolicity into the model (1.2). In particular, it is the case of the DUFORT–FRANKEL scheme, while SAULYEV method seems to be rather dispersive. This observation suggests that a new method can be derived by introducing this hyperbolicity in a controlled manner and to discretize the perturbed equation later. The method of hyperbolization deforms the equation operator to achieve desired properties* of the numerical solution. However, in all cases the gain in stability results in some loss of accuracy in representing the original continuous operator (1.1). The trade-off between the stability and accuracy has to be made by the end user. The second order accuracy is equivalent to the classical parabolic CFL-type condition. However, the user can choose to degrade intentionally the accuracy to relax the stability restriction up to hyperbolic-type conditions $\Delta t = \mathcal{O}(\Delta x)$ (originally found in [5]) and even beyond. It is not difficult to see that hybrid schemes presented in this study can be easily generalized to nonlinear cases with source terms in one and more spatial dimensions.

The main goal of this manuscript was to communicate and attract community's attention to these improved discretizations, which can be used in modern building physics simulations where typical time scales are measured rather in months or even years. Our preference goes perhaps to the method of hyperbolization since it has been successfully applied (and validated) even to compressible NAVIER–STOKES and MHD equations [2]. Another advantage of hyperbolization technique is that it can be mathematically derived for gas dynamics from the kinetic theory of BOLTZMANN.

While preparing this text, the Author discovered also another strategy to relax the time step limits. This idea can be traced back to the scientific school of R. TEMAM who proposed to separate the scales for the purposes of numerical simulations. Then, high frequencies, which impose severe stability limits, are treated separately. For a modern introduction to these approaches we refer to [1]. There exist also explicit time integration methods THE SO-CALLED RUNGE–KUTTA–TCHEBYSHEV METHODS, especially designed to have extended stability limits. This research direction was pioneered also by V. K. SAULYEV in 1960 [8].

Acknowledgments

The author would like to thank Professor B.N. CHETVERUSHKIN whose talks available on-line greatly inspired the preparation of this manuscript. I would like also to thank my collaborators J. BERGER, S. GASPARIN and N. MENDES for bringing my attention to heat and mass conduction problems.

*By desired properties we mean the finite speed of information propagation along with less stringent CFL-type restrictions in explicit finite difference discretizations.

References

- [1] M. Brachet and J.-P. Chehab. Stabilized Times Schemes for High Accurate Finite Differences Solutions of Nonlinear Parabolic Equations. *J. Sci. Comput.*, 69(3):946–982, dec 2016. [19](#)
- [2] B. N. Chetverushkin, N. D’Ascenzo, and V. I. Saveliev. Hyperbolic type explicit kinetic schemes of magneto gas dynamic for high performance computing systems. *Num. Anal. and Math. Model.*, 30(1), 2015. [17](#), [19](#)
- [3] B. N. Chetverushkin and A. V. Gulin. Explicit schemes and numerical simulation using ultrahigh-performance computer systems. *Doklady Mathematics*, 86(2):681–683, sep 2012. [11](#), [17](#), [18](#)
- [4] B. N. Chetverushkin, D. N. Morozov, M. A. Trapeznikova, N. G. Churbanova, and E. V. Shil’nikov. An explicit scheme for the solution of the filtration problems. *Math. Model. Comput. Simul.*, 2(6):669–677, 2010. [17](#)
- [5] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzgleichungen der mathematischen Physik. *Mathematische Annalen*, 100(1):32–74, 1928. [5](#), [19](#)
- [6] A. Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 322(8):549–560, 1905. [4](#)
- [7] E. E. Myshetskaya and V. F. Tishkin. Estimates of the hyperbolization effect on the heat equation. *Comp. Math. Math. Phys.*, 55(8):1270–1275, aug 2015. [18](#)
- [8] V. K. Saul’yev. *Integration of parabolic equations by the grid method*. Fizmatgiz, Moscow, 1960. [13](#), [14](#), [19](#)

LAMA, UMR 5127 CNRS, UNIVERSITÉ SAVOIE MONT BLANC, CAMPUS SCIENTIFIQUE, 73376 LE BOURGET-DU-LAC CEDEX, FRANCE

E-mail address: Denys.Dutykh@univ-savoie.fr

URL: <http://www.denys-dutykh.com/>