Formal Proof of a Wave Equation Resolution Scheme: the Method Error*

Sylvie Boldo[1,2], François Clément[3], Jean-Christophe Filliâtre[2,1], Micaela Mayero[4,5], Guillaume Melquiond[1,2], and Pierre Weis[3]

[1] INRIA Saclay – Île-de-France, ProVal, Orsay, F-91893
[2] LRI, Université Paris-Sud, CNRS, Orsay, F-91405
[3] INRIA Paris – Rocquencourt, Estime, Le Chesnay, F-78153
[4] LIPN, Université Paris 13, LCR, Villetaneuse, F-93430
[5] LIP, Arénaire (INRIA Grenoble - Rhône-Alpes, CNRS UMR 5668, UCBL, ENS Lyon), Lyon, F-69364

Abstract. Popular finite difference numerical schemes for the resolution of the one-dimensional acoustic wave equation are well-known to be convergent. We present a comprehensive formalization of the simplest scheme and formally prove its convergence in Coq. The main difficulties lie in the proper definition of asymptotic behaviors and the implicit way they are handled in the mathematical pen-and-paper proofs. To our knowledge, this is the first time this kind of mathematical proof is machine-checked.

Key words: partial differential equation, acoustic wave equation, numerical scheme, Coq formal proofs

1 Introduction

Ordinary differential equations (ODE) and partial differential equations (PDE) are ubiquitous in engineering and scientific computing. They show up in weather forecast, nuclear simulation, etc., and more generally in numerical simulation. Solutions to nontrivial problems are nonanalytic, hence approximated by numerical schemes over discrete grids.

Numerical analysis is mainly interested in proving the convergence of these schemes, that is, the approximation quality increases as the size of the discretization steps decreases. The approximation quality is characterized by the error defined as the difference between the exact continuous solution and the approximated discrete solution; this error must tend toward zero in order for the numerical scheme to be useful.

There is a wide literature on this topic, *e.g.* see [1, 2], but no article goes into all the details. These "details" may have been skipped for readability, but they could also be mandatory details that were omitted due to an oversight. The

* This research was supported by the ANR projects CerPAN (ANR-05-BLAN-0281-04) and F∮ST (ANR-08-BLAN-0246-01).

purpose of a mechanically-checked proof is to uncover these issues and check whether they could jeopardize the correctness of the schemes.

This work is a first step toward the development of formal tools for dealing with the convergence of numerical schemes. It would have been sensible to start with classical schemes for ODE, such as the Euler or Runge-Kutta methods. But we decided to directly validate the feasibility of our approach on the more complicated PDE. Moreover, this opens the door to a wide variety of applications, as they appear in many realistic problems from industry.

We chose the domain of wave propagation because it represents one of the most common physical phenomena one experiences in everyday life: directly through sight and hearing, but also via telecommunications, radar, medical imaging, etc. Industrial applications range from aeroacoustics to music acoustics (acoustic waves), from oil prospection to nondestructive testing (elastic waves), from optics to stealth technology (electromagnetic waves), and even include stabilization of ships and offshore platforms (surface gravity waves). We restrained ourselves to the simplest example of wave propagation models, the acoustic wave equation in a one-dimensional space domain, for it is a prototype model for all other kinds of wave. In this case, the equation describes the propagation of pressure variations (or sound waves) in a fluid medium; it also models the behavior of a vibrating string. For simplicity, we only consider homogeneous media, meaning that the propagation velocity is constant. Among the wide variety of numerical schemes for approximately solving the 1D acoustic wave equation, we chose the simplest one: the second order centered finite difference scheme, also known as the "three-point scheme". Again, for simplicity, we only consider regular grids with constant discretization steps for time and space.

To our knowledge, this is the first time this kind of mathematical proof is machine-checked.[6] Few works have been done on formalization and proofs on mathematical analysis inside proof assistants, and fewer on numerical analysis. Even real analysis developments are relatively new. The first developments on real numbers and real analysis are from the late 90's [3–7]. Some intuitionist formalizations have been realized by a team at Nijmegen [8, 9]. Analysis results are available in provers such as ACL2, Coq, HOL Light, Isabelle, Mizar, or PVS. Regarding numerical analysis, we can cite [10] which deals, more precisely, with the formal proof of an automatic differentiation algorithm. About $\mathbb{R}^n$ and the dot product, an extensive work has been done by Harrison [11]. About the big O operator for asymptotic comparison, a decision procedure has been developed in [12]; unfortunately, we needed a more powerful big O and those results were not applicable.

Section 2 presents the PDE, the numerical scheme, and their mathematical properties. Section 3 describes the basic blocks of the formalization: dot product, big O, and Taylor expansions. Section 4 is devoted to the formal proof of the convergence of the numerical scheme.

[6] The Coq sources of the formal development are available from `http://fost.saclay.inria.fr/wave_method_error.php`.

2 Wave Equation

A partial differential equation modeling an evolutionary problem is an equation involving partial derivatives of an unknown function of several independent space and time variables. The uniqueness of the solution is obtained by imposing additional conditions, typically the value of the function and the value of some of its derivatives at the initial time. The right-hand sides of such initial conditions are also called *Cauchy data*, making the whole problem a *Cauchy problem*, or an *initial-value problem*.

The mathematical theory is simpler when unbounded domains are considered [1]. When the space domain is bounded, the computation is simpler, but we have to take reflections at domain boundaries into account; this models a finite vibrating string fixed at both ends. Thanks to the nice property of finite velocity of propagation of the wave equation, we can build two Cauchy problems, one bounded and the other one unbounded, that coincide on the domain of the bounded one. Thus, we can benefit from the best of both worlds: the bounded problem makes computation simpler and the unbounded one avoids handling reflections. This section, as well as the steps taken at section 4 to conduct the proof of the convergence of the numerical scheme, is inspired by [13].

2.1 The continuous equation

The chosen PDE models the propagation of waves along an ideal vibrating elastic string, see [14, 15]. It is obtained from Newton's laws of motion [16].

The gravity is neglected, hence the string is supposed rectilinear when at rest. Let $u(x,t)$ be the transverse displacement of the point of the string of abscissa x at time t from its equilibrium position. It is a (signed) scalar. Let c be the constant propagation velocity. It is a positive number that depends on the section and density of the string. Let $s(x,t)$ be the external action on the point of abscissa x at time t; it is a source term, such that $t = 0 \Rightarrow s(x,t) = 0$. Finally, let $u_0(x)$ and $u_1(x)$ be the initial position and velocity of the point of abscissa x. We consider the Cauchy problem (*i.e.*, with conditions at $t = 0$)

$$(1) \quad \forall t \geq 0,\ \forall x \in \mathbb{R}, \quad (L(c)\,u)(x,t) \stackrel{\text{def}}{=} \frac{\partial^2 u}{\partial t^2}(x,t) + A(c)\,u(x,t) = s(x,t),$$

$$(2) \quad \forall x \in \mathbb{R}, \quad (L_1\,u)(x,0) \stackrel{\text{def}}{=} \frac{\partial u}{\partial t}(x,0) = u_1(x),$$

$$(3) \quad \forall x \in \mathbb{R}, \quad (L_0\,u)(x,0) \stackrel{\text{def}}{=} u(x,0) = u_0(x)$$

where the differential operator $A(c)$ is defined by

$$(4) \quad A(c) \stackrel{\text{def}}{=} -c^2 \frac{\partial^2}{\partial x^2}.$$

We admit that under reasonable conditions on the Cauchy data u_0 and u_1 and on the source term s, there exists a unique solution to the Cauchy problem (1)–(3) for each $c > 0$. This is a mathematical known fact (established for example from d'Alembert's formula (6)), that is left unproved here.

For such a solution u, it is natural to associate at each time t the positive definite quadratic quantity

$$E(c)(u)(t) \stackrel{\text{def}}{=} \frac{1}{2}\left\|x \mapsto \frac{\partial u}{\partial t}(x,t)\right\|^2 + \frac{1}{2}\left\|x \mapsto u(x,t)\right\|^2_{A(c)} \tag{5}$$

where $\langle v, w\rangle \stackrel{\text{def}}{=} \int_{\mathbb{R}} v(x)w(x)dx$, $\|v\|^2 \stackrel{\text{def}}{=} \langle v, v\rangle$ and $\|v\|^2_{A(c)} \stackrel{\text{def}}{=} \langle A(c)\, v, v\rangle$. The first term is interpreted as the kinetic energy, and the second term as the potential energy, making E the mechanical energy of the vibrating string.

This simple partial derivative equation happens to possess an analytical solution given by the so-called d'Alembert's formula [17], obtained from the method of characteristics [18], $\forall t \geq 0$, $\forall x \in \mathbb{R}$,

$$u(x,t) = \frac{1}{2}(u_0(x-ct) + u_0(x+ct)) + \frac{1}{2c}\int_{x-ct}^{x+ct} u_1(y)dy + \frac{1}{2c}\int_0^t \left(\int_{x-c(t-\sigma)}^{x+c(t-\sigma)} s(y,\sigma)dy\right) d\sigma. \tag{6}$$

One can deduce from formula (6) the useful property of finite velocity of propagation. Assuming that we are only interested in the resolution of the Cauchy problem on a compact time interval of the form $[0, t_{\max}]$ with $t_{\max} > 0$, we suppose that u_0, u_1 and s have a compact support. Then the property states that there exists $x_{\min}$ and $x_{\max}$ with $x_{\min} < x_{\max}$ such that the support of the solution is a subset of $\Omega \stackrel{\text{def}}{=} [x_{\min}, x_{\max}] \times [0, t_{\max}]$. Furthermore, since the boundaries do not have time to be reached by the signal, the Cauchy problem set on Ω by adding homogeneous Dirichlet boundary conditions (*i.e.* for all $t \in [0, t_{\max}]$, $u(x_{\min}, t) = u(x_{\max}, t) = 0$), admits the same solution. Hence, we will numerically solve the Cauchy problem on Ω, but with the assumption that the spatial boundaries are not reached.

Note that the implementation of the compact spatial domain $[x_{\min}, x_{\max}]$ will be abstracted by the notion of finite support (that is to say, being zero outside of an interval, see Section 4.2) and will not appear explicitly otherwise.

Note also that most properties of the continuous problem proved unnecessary in the formalization of the numerical scheme and the proof of its convergence. For instance, integration operators and d'Alembert's formula can be avoided as long as we suppose the existence and regularity of a solution to the PDE and that this solution has a finite support.

2.2 The discrete equations

Let $(\Delta x, \Delta t)$ be a point in the interior of Ω; define the discretization functions $j_{\Delta x}(x) \stackrel{\text{def}}{=} \left\lfloor \frac{x - x_{\min}}{\Delta x} \right\rfloor$ and $k_{\Delta t}(t) \stackrel{\text{def}}{=} \left\lfloor \frac{t}{\Delta t} \right\rfloor$; then set $j_{\max} \stackrel{\text{def}}{=} j_{\Delta x}(x_{\max})$ and $k_{\max} \stackrel{\text{def}}{=} k_{\Delta t}(t_{\max})$. Now, the compact domain Ω is approximated by the regular discrete grid defined by

$$\forall k \in [0..k_{\max}],\ \forall j \in [0..j_{\max}], \quad \mathbf{x}_j^k \stackrel{\text{def}}{=} (x_j, t^k) \stackrel{\text{def}}{=} (x_{\min} + j\Delta x, k\Delta t). \tag{7}$$

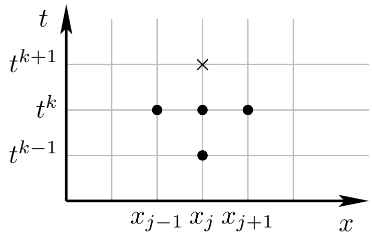

Fig. 1. Three-point scheme: u_j^{k+1} ($\times$) depends on u_{j-1}^k, u_j^k, u_{j+1}^k and u_j^{k-1} ($\bullet$).

Let v_h be a discrete function over $[0..j_{\max}] \times [0..k_{\max}]$. For all k in $[0..k_{\max}]$, we write $v_h^k = (v_j^k)_{0 \le j \le j_{\max}}$, then $v_h = ((v_h^k)_{0 \le k \le k_{\max}})$. A function v defined over Ω is approximated at the points of the grid by the discrete function v_h defined on $[0..j_{\max}] \times [0..k_{\max}]$ by $v_j^k \stackrel{\text{def}}{=} v(\mathbf{x}_j^k)$, except for u where we use the notation $\bar{u}_j^k \stackrel{\text{def}}{=} u(\mathbf{x}_j^k)$ to prevent notation clashes.

Let u_{0h} and u_{1h} be two discrete functions over $[0..j_{\max}]$; let s_h be a discrete function over $[0..j_{\max}] \times [0..k_{\max}]$. Then, the discrete function u_h over $[0..j_{\max}] \times [0..k_{\max}]$ is said to be the solution of the three-point[7] finite difference scheme, as illustrated in Figure 1, when the following set of equations holds:

(8) $\forall k \in [2..k_{\max}],\ \forall j \in [0..j_{\max}],$

$$(L_h(c)\,u_h)_j^k \stackrel{\text{def}}{=} \frac{u_j^k - 2u_j^{k-1} + u_j^{k-2}}{\Delta t^2} + (A_h(c)\,u_h^{k-1})_j = s_j^{k-1},$$

$$(9) \qquad \forall j \in [0..j_{\max}],\ (L_{1h}(c)\,u_h)_j \stackrel{\text{def}}{=} \frac{u_j^1 - u_j^0}{\Delta t} + \frac{\Delta t}{2}(A_h(c)\,u_h^0)_j = u_{1,j},$$

$$(10) \qquad \forall j \in [0..j_{\max}],\ (L_{0h}\,u_h)_j \stackrel{\text{def}}{=} u_j^0 = u_{0,j},$$

$$(11) \qquad \forall k \in [0..k_{\max}[,\ u_{-1}^k = u_{j_{\max}+1}^k = 0$$

where the matrix $A_h(c)$, discrete analog of $A(c)$, is defined, for any vector $v_h = ((v_j)_{0 \le j \le k_{\max}})$, by

$$(12) \qquad \forall j \in [0..j_{\max}], \quad (A_h(c)\,v_h)_j \stackrel{\text{def}}{=} -c^2\frac{v_{j+1} - 2v_j + v_{j-1}}{\Delta x^2}.$$

Note that defining u_h for artificial indexes $j = -1$ and $j = j_{\max} + 1$ is a trick to make the three-point spatial scheme valid for $j = 0$ and $j = j_{\max}$.

A discrete analog of the energy is also defined by[8]

$$(13) \qquad E_h(c)(u_h)^{k+\frac{1}{2}} \stackrel{\text{def}}{=} \frac{1}{2}\left\|\frac{u_h^{k+1} - u_h^k}{\Delta t}\right\|_{\Delta x}^2 + \frac{1}{2}\left\langle u_h^k, u_h^{k+1}\right\rangle_{A_h(c)}$$

[7] In the sense "three spatial points", for the definition of matrix $A_h(c)$.

[8] By convention, the energy is defined between steps k and $k+1$, thus the notation $k + \frac{1}{2}$.

where $\langle v_h, w_h \rangle_{\Delta x} \stackrel{\text{def}}{=} \sum_{j=0}^{j_{\max}} v_j w_j \Delta x$, $\|v_h\|^2_{\Delta x} \stackrel{\text{def}}{=} \langle v_h, v_h \rangle_{\Delta x}$,
and $\langle v_h, w_h \rangle_{A_h(c)} \stackrel{\text{def}}{=} \langle A_h(c)\, v_h, w_h \rangle_{\Delta x}$.

Note that the three-point scheme is parametrized by the discrete Cauchy data u_{0h} and u_{1h}, and by the discrete source term s_h. Of course, when u_{0h}, u_{1h}, and s_h are respectively approximations of u_0, u_1, f, then the discrete solution u_h is an approximation of the continuous solution u.

2.3 Convergence

Let ζ and ξ be in $]0, 1[$ with $\zeta \leq 1 - \xi$. The CFL(ζ, ξ) condition (for Courant-Friedrichs-Lewy, see [19]) states that the discretization steps satisfy the relation

$$\zeta \leq \frac{c\Delta t}{\Delta x} \leq 1 - \xi. \tag{14}$$

Note that the lower bound ζ may seem surprising from a numerical analysis point of view; the formalization has however shown that it was mandatory (see Section 4.3).

The convergence error e_h measures the distance between the continuous and discrete solutions. It is defined by

$$\forall k \in [0..k_{\max}], \forall j \in [0..j_{\max}], \quad e_j^k \stackrel{\text{def}}{=} \bar{u}_j^k - u_j^k. \tag{15}$$

The truncation error ε_h measures at which precision the continuous solution satisfies the numerical scheme. It is defined by

$$\forall k \in [2..k_{\max}], \forall j \in [0..j_{\max}], \ \varepsilon_j^{k-1} \stackrel{\text{def}}{=} (L_h(c)\, \bar{u}_h)_j^k - s_j^{k-1}, \tag{16}$$

$$\forall j \in [0..j_{\max}], \ \varepsilon_j^0 \stackrel{\text{def}}{=} (L_{1h}(c)\, \bar{u}_h)_j - u_{1,j}, \tag{17}$$

$$\forall j \in [0..j_{\max}], \ \varepsilon_j^{-1} \stackrel{\text{def}}{=} (L_{0h}\bar{u}_h)_j - u_{0,j}. \tag{18}$$

The numerical scheme is said to be convergent of order 2 if the convergence error tends toward zero at least as fast as $\Delta x^2 + \Delta t^2$ when both discretization steps tend toward 0. More precisely, the numerical scheme is said to be convergent of order (p,q) uniformly on the interval $[0, t_{\max}]$ if the convergence error satisfies (see Section 3.2 for the definition of the big O notation that will be uniform with respect to space and time)

$$\left\| e_h^{k_{\Delta t}(t)} \right\|_{\Delta x} = O_{[0,t_{\max}]}(\Delta x^p + \Delta t^q). \tag{19}$$

The numerical scheme is said to be consistent with the continuous problem at order 2 if the truncation error tends toward zero at least as fast as $\Delta x^2 + \Delta t^2$ when the discretization steps tend toward 0. More precisely, the numerical scheme is said to be consistent with the continuous problem at order (p, q) uniformly on interval $[0, t_{\max}]$ if the truncation error satisfies

$$\left\| \varepsilon_h^{k_{\Delta t}(t)} \right\|_{\Delta x} = O_{[0,t_{\max}]}(\Delta x^p + \Delta t^q). \tag{20}$$

The numerical scheme is said to be stable if the discrete solution of the associated homogeneous problem (*i.e.* without any source term, $s(x,t) = 0$) is bounded from above independently of the discretization steps. More precisely, the numerical scheme is said to be stable uniformly on interval $[0, t_{\max}]$ if the discrete solution of the problem without any source term satisfies

$$\text{(21)} \quad \exists \alpha, C_1, C_2 > 0,\ \forall t \in [0, t_{\max}],\ \forall \Delta x, \Delta t > 0, \quad \sqrt{\Delta x^2 + \Delta t^2} < \alpha \Rightarrow \left\| u_h^{k_{\Delta t}(t)} \right\|_{\Delta x} \leq (C_1 + C_2 t)(\|u_{0h}\|_{\Delta x} + \|u_{0h}\|_{A_h(c)} + \|u_{1h}\|_{\Delta x}).$$

The result to be formally proved at section 4 states that if the continuous solution u is regular enough on Ω and if the discretization steps satisfy the CFL(ζ, ξ) condition, then the three-point scheme is convergent of order (2, 2) uniformly on interval $[0, t_{\max}]$.

We do not admit (nor prove) the Lax equivalence theorem which stipulates that for a wide variety of problems and numerical schemes, consistency implies the equivalence between stability and convergence. Instead, we establish that consistency and stability implies convergence in the particular case of the one-dimensional acoustic wave equation.

3 The Coq Formalization: Basic Blocks

We decided to use the Coq proof assistant [20], as Coq was already used to prove the floating-point error [21] of this case study. All our developments use the Coq real standard (classical) library. Numerical equations, numerical schemes, numerical approximations deal with classical statements, and are not in the scope of intuitionist theory.

3.1 Dot product

The function space $\mathbb{Z} \to \mathbb{R}$ can be equipped with pointwise addition and multiplication by a scalar. The result is a vector space. In the following, we are only interested in functions with finite support, that is the subset

$$F \stackrel{\text{def}}{=} \{f : \mathbb{Z} \to \mathbb{R} \mid \exists a, b \in \mathbb{Z}, \forall i \in \mathbb{Z}, f(i) \neq 0 \Rightarrow a \leq i \leq b\},$$

which is also a vector space. Then it is possible to define a dot product on F, noted $\langle ., . \rangle$, as follows:

$$\text{(22)} \qquad \langle f, g \rangle \stackrel{\text{def}}{=} \sum_{i \in \mathbb{Z}} f(i) g(i)$$

and the corresponding norm $\|f\| \stackrel{\text{def}}{=} \sqrt{\langle f, f \rangle}$. The corresponding Coq formalization is not immediate, though. One could characterize F with a dependent type, but that would make operation $\langle ., . \rangle$ difficult to use (each time it is applied, proofs of finite support properties have to be passed as well). Instead, we define $\langle ., . \rangle$ on the full function space $\mathbb{Z} \to \mathbb{R}$ using Hilbert's ε-operator (provided in Coq standard library in module `Epsilon`), as follows:

$$\langle f, g \rangle \stackrel{\text{def}}{=} \varepsilon \left(\begin{array}{c} \lambda x. \exists a\, b,\, (\forall i, (f(i) \neq 0 \vee g(i) \neq 0) \Rightarrow a \leq i \leq b) \\ \wedge\ x = \sum_{i=a}^{b} f(i) g(i) \end{array} \right) \tag{23}$$

Said otherwise, we give $\langle f, g \rangle$ a definition as a finite sum whenever f and g both have finite support and we let $\langle f, g \rangle$ undefined otherwise.

To ease the manipulation of functions with finite support, we introduce the following predicate characterizing such functions

$$FS(f) \stackrel{\text{def}}{=} \exists a\, b, \forall i, f(i) \neq 0 \Rightarrow a \leq i \leq b$$

and we prove several lemmas about it, such as

$$\begin{aligned} &\forall f g, FS(f) \Rightarrow FS(g) \Rightarrow FS(f+g) \\ &\forall f c, FS(f) \Rightarrow FS(c \cdot f) \\ &\forall f k, FS(f) \Rightarrow FS(i \mapsto f(i+k)) \end{aligned}$$

We also provide a Coq tactic to automatically discharge most goals about $FS(.)$. Finally, we can establish lemmas about the dot product, provided functions have finite support. Here are some of these lemmas:

$$\begin{aligned} &\forall f\, g\, c, FS(f) \Rightarrow FS(g) \Rightarrow \langle c \cdot f, g \rangle = c \cdot \langle f, g \rangle \\ &\forall f_1\, f_2\, g, FS(f_1) \Rightarrow FS(f_2) \Rightarrow FS(g) \Rightarrow \langle f_1 + f_2, g \rangle = \langle f_1, g \rangle + \langle f_2, g \rangle \\ &\forall f\, g, FS(f) \Rightarrow FS(g) \Rightarrow |\langle f, g \rangle| \leq \|f\| \cdot \|g\| \\ &\forall f\, g, FS(f) \Rightarrow FS(g) \Rightarrow \|f + g\| \leq \|f\| + \|g\| \end{aligned}$$

These lemmas are proved by reduction to finite sums, thanks to Formula (23). Note that the value of $\langle f, g \rangle_{\Delta x}$ defined in Section 2.2 is equal to $\Delta x \cdot \langle f, g \rangle$.

3.2 Big O notation

For two functions f and g over $\mathbb{R}^n$, one usually writes $f(\boldsymbol{x}) = O_{\|\boldsymbol{x}\| \to 0}(g(\boldsymbol{x}))$ for

$$\exists \alpha, C > 0, \quad \forall \boldsymbol{x} \in \mathbb{R}^n, \quad \|\boldsymbol{x}\| \leq \alpha \Rightarrow |f(\boldsymbol{x})| \leq C \cdot |g(\boldsymbol{x})|.$$

Unfortunately, this definition is not sufficient for our formalism. Indeed, while $f(\mathbf{x}, \mathbf{\Delta x})$ will be defined over $\mathbb{R}^2 \times \mathbb{R}^2$, $g(\mathbf{\Delta x})$ will be defined over $\mathbb{R}^2$ only. So it begs the question: what to do about $\mathbf{x}$?

Our first approach was to use

$$\forall \mathbf{x}, \quad f(\mathbf{x}, \mathbf{\Delta x}) = O_{\|\mathbf{\Delta x}\| \to 0}(g(\mathbf{\Delta x}))$$

that is to say

$$\forall \mathbf{x}, \exists \alpha, C > 0, \quad \forall \mathbf{\Delta x} \in \mathbb{R}^2, \quad \|\mathbf{\Delta x}\| \le \alpha \Rightarrow |f(\mathbf{x}, \mathbf{\Delta x})| \le C \cdot |g(\mathbf{\Delta x})|$$

which means that α and C are functions of $\mathbf{x}$. So we would need to take the minimum of all the possible values of α, and the maximum for C. Potentially, they may be 0 and $+\infty$ respectively, making them useless.

In order to solve this issue, we had to define a notion of big O uniform with respect to the additional variable $\mathbf{x}$:

$$\exists \alpha, C > 0, \quad \forall \mathbf{x}, \mathbf{\Delta x}, \quad \|\mathbf{\Delta x}\| \le \alpha \Rightarrow |f(\mathbf{x}, \mathbf{\Delta x})| \le C \cdot |g(\mathbf{\Delta x})|.$$

Variables $\mathbf{x}$ and $\mathbf{\Delta x}$ are restricted to subsets S and P of $\mathbb{R}^2$. For instance, the big O that appears in Equation (19) uses

$$S = \mathbb{R} \times [0, t_{\max}],$$
$$P = \left\{ \mathbf{\Delta x} = (\Delta x, \Delta t) \mid 0 < \Delta x \;\wedge\; 0 < \Delta t \;\wedge\; \zeta \le \frac{c \cdot \Delta t}{\Delta x} \le 1 - \xi \right\}.$$

As often, the formal specification has allowed us to detect some flaws in usual mathematical pen-and-paper proofs, such as an erroneous switching of the universal and existential quantifiers hidden in the big O definition.

3.3 Taylor expansion

The formalization assumes that "sufficiently regular" functions can be uniformly approximated by multivariate Taylor series. More precisely, the development starts by assuming that there exists two operators `partial_derive_firstvar` and `_secondvar`. Given a real-valued function f defined on the 2D plane and a point of it, they respectively return the functions $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial t}$ for this point, if they exist.

Again, these operators are similar to the use of Hilbert's ε operator. For documentation purpose, one could add two axioms stating that the returned function computes the derivatives for derivable functions; they are not needed for the later development though. Indeed, none of our proofs depend on the actual properties of derivatives; they only care about the fact that differential operators appear in both the regularity definition below and the wave equation.

The two primitive operators $\frac{\partial}{\partial x}$ and $\frac{\partial}{\partial t}$ are encompassed in a generalized differential operator $\frac{\partial^{m+n}}{\partial x^m \partial t^n}$. This allows us to define the 2D Taylor expansion of order n of a function f:

$$\mathrm{DL}_n(f, \mathbf{x}) \stackrel{\text{def}}{=} (\Delta x, \Delta t) \mapsto \sum_{p=0}^{n} \frac{1}{p!} \left(\sum_{m=0}^{p} \binom{p}{m} \cdot \frac{\partial^p f}{\partial x^m \partial t^{p-m}}(\mathbf{x}) \cdot \Delta x^m \cdot \Delta t^{p-m} \right).$$

A function f is then said to be sufficiently regular of order n if

$$\forall m \le n, \quad \mathrm{DL}_{m-1}(f, \mathbf{x})(\mathbf{\Delta x}) - f(\mathbf{x} + \mathbf{\Delta x}) = O\left(\|\mathbf{\Delta x}\|^m\right). \tag{24}$$

4 The Coq Formalization: Convergence

4.1 Wave equation

As explained in Section 2, a solution of the wave equation with given u_0, u_1 and s verifies Equations (1)–(3). Its discrete approximation verifies Equations (8)–(10). Both are directly translated in Coq using the definitions of Section 3. Concerning the discretization, we choose that the space index is in $\mathbb{Z}$ (to be coherent with the dot product definition of Section 3.1) while the time index is in $\mathbb{N}$.

Our goal is to prove the uniform convergence of the scheme with order (2,2) on the interval $[0, t_{\max}]$:

$$\left\| e_h^{k_{\Delta t}(t)} \right\|_{\Delta x} = O_{\substack{t \in [0, t_{\max}] \\ (\Delta x, \Delta t) \to 0 \\ 0 < \Delta x \,\wedge\, 0 < \Delta t \,\wedge \\ \zeta \le c \frac{\Delta t}{\Delta x} \le 1 - \xi}} (\Delta x^2 + \Delta t^2).$$

4.2 Finite support

The proofs concerning the convergence of the scheme rely on the dot product. As explained in Section 3.1, the dot product requires the functions to have a finite support in order to apply any lemma. We therefore proved the finiteness of the support of many functions. We assume that the inputs u_0, u_1, and s of the wave equation have a finite support. More precisely, we assume that there exists χ_1 and χ_2 such that $u_0(x) = u_1(x) = 0$ for all x out of $[\chi_1, \chi_2]$ and $s(x, t) = 0$ for all x out of $[\chi_1 - c \cdot t, \chi_2 + c \cdot t]$ where c is the velocity of propagation of waves in Equation (1).

Figure 2 describes the nullity, that is to say the finite support, of the various functions. We needed to prove the finiteness of their support:

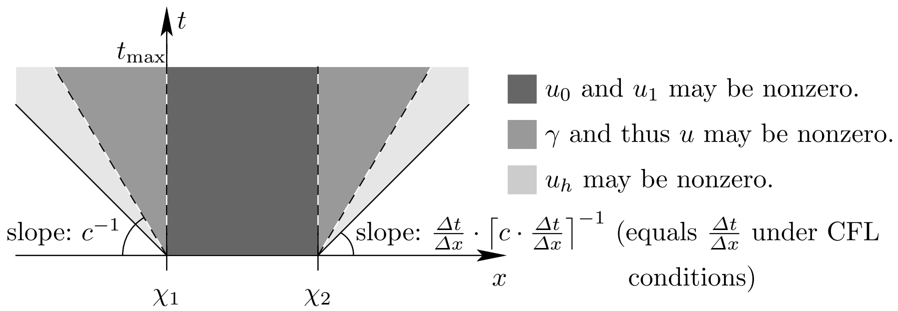

Fig. 2. Finite supports. The support of the Cauchy data u_0 and u_1 is included in the support of the continuous source term s, and of the continuous solution u. Which is in turn also included in the support of the discrete solution u_h, provided that the CFL condition holds. For a finite $t_{\max}$, all these supports are finite.

- u_0 and u_1 by hypothesis and therefore $u_{0,j}$ and $u_{1,j}$.
- s (for any value t) by hypothesis and therefore s_j^k is zero outside of a cone of slope c^{-1}.
- the scheme itself has a finite support: due to the definition of u_j^k and the nullity of $u_{0,j}$ and $u_{1,j}$ and s_j^k, we can prove that u_j^k is zero outside of a cone of slope $\frac{\Delta t}{\Delta x} \cdot \left\lceil c \cdot \frac{\Delta t}{\Delta x} \right\rceil^{-1}$. Under CFL$(\zeta, \xi)$ conditions, this slope will be $\frac{\Delta t}{\Delta x}$.
- the truncation and convergence errors also have finite support with the previous slope.

We need here an axiom about the nullity of the continuous solution. We assume that the continuous solution $u(x,t)$ is zero for x out of $[\chi_1 - c \cdot t, \chi_2 + c \cdot t]$ (same as s). This is mathematically correct, since it derives from d'Alembert's formula (6). But its proof is out of the scope of the current formalization and we therefore preferred to simply add the nullity axiom.

4.3 Consistency

We first prove that the truncation error is of order $\Delta x^2 + \Delta t^2$. The idea is to show that, for $\mathbf{\Delta x}$ small enough, the values of the scheme L_h are near the corresponding values of L. This is done using the properties of Taylor expansions. This involves long and complex expressions but the proof is straightforward.

We first prove that the truncation error in one point (j,k) is a $O(\Delta x^2 + \Delta t^2)$. This is proved for $k = 0$ and $k = 1$ by taking advantage of the initializations and Taylor expansions. For bigger k, the truncation error reduces to the sum of two Taylor expansions of degree 3 in time (this means $m = 4$ in Formula (24)) and two Taylor expansions of degree 3 in space that partially cancel (divided by something proportional to $\|\mathbf{\Delta x}\|^2$). Here, we take advantage of the generality of big O as we consider the sum of a Taylor expansion on Δx and of a Taylor expansion on $-\Delta x$. If we had required $0 < \Delta x$ (as a space grid step), we could not have done this proof.

The most interesting part is to go from pointwise consistency to uniform consistency. We want to prove that the norm of the truncation error (in the sense of the infinite dot product $\langle \cdot, \cdot \rangle_{\Delta x}$) is also $O(\Delta x^2 + \Delta t^2)$. We therefore need to bound the number of nonzero values of the truncation error. As explained in Section 4.2, the truncation error values at time $k \cdot \Delta t$ may be nonzero between ${\chi_1}'_k = \left\lfloor \frac{\chi_1}{\Delta x} \right\rfloor - \left\lceil c \cdot \frac{\Delta t}{\Delta x} \right\rceil k$ and ${\chi_2}'_k = \left\lceil \frac{\chi_2}{\Delta x} \right\rceil + \left\lceil c \cdot \frac{\Delta t}{\Delta x} \right\rceil k$. This gives a number of terms N roughly bounded by (all details are handled in the formal proof):

$$N \le \frac{{\chi_2}'_k - {\chi_1}'_k}{\Delta x} \le \frac{\chi_2 - \chi_1}{\Delta x^2} + 2 \cdot k_{\max} \cdot \frac{\left\lceil c \cdot \frac{\Delta t}{\Delta x} \right\rceil}{\Delta x}$$
$$\le \frac{\chi_2 - \chi_1}{\Delta x^2} + 2 \cdot \frac{t_{\max}}{\Delta t} \cdot \frac{c \cdot \frac{\Delta t}{\Delta x} + 1}{\Delta x}$$

As the norm is a Δx-norm, this reduces to bounding with a constant value the value $N \cdot \Delta x^2$ which is smaller than $\chi_2 - \chi_1 + 2 \cdot t_{\max} \cdot c + 2 \cdot t_{\max} \cdot \frac{\Delta x}{\Delta t}$. To bound this with a constant value, we require $c\frac{\Delta t}{\Delta x}$ to have a constant lower bound ζ (it already had an upper bound $1-\xi$). Then $N\cdot\Delta x^2 \le \chi_2 - \chi_1 + 2\cdot t_{\max}\cdot c + 2\cdot c\cdot t_{\max}\cdot\frac{1}{\zeta}$ which is constant.

Mathematically, this requirement comes as a surprise. The following scenario explains it. If $c\frac{\Delta t}{\Delta x}$ goes to zero, then Δt goes to zero much faster than Δx. It corresponds to Figure 3. The number of nonzero terms (for u_h and thus for the truncation error) goes to infinity as $\frac{\Delta t}{\Delta x}$ goes to zero.

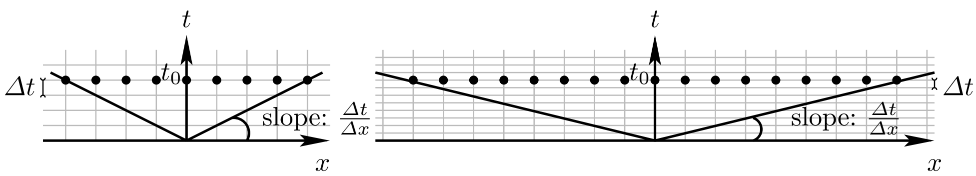

Fig. 3. For a given time t_0, the number of nonzero values increases when the slope $\frac{\Delta t}{\Delta x}$ goes to zero. From left to right, Δt is divided by 2 whereas Δx remains the same. We can see that the number of nonzero terms is almost doubled (from 9 to 17).

4.4 Stability

To prove stability, we use the discrete energy defined in Equation (13). From the properties of the scheme, we calculate the evolution of the energy. At each step, it increases by a known value. In particular, if s is zero, the discrete energy (as the continuous energy) is constant:

$$\forall k > 0, \quad E_h(c)(u_h)^{k+\frac{1}{2}} - E_h(c)(u_h)^{k-\frac{1}{2}} = \frac{1}{2}\left\langle u_h^{k+1} - u_h^{k-1}, s_h^k \right\rangle_{\Delta x}.$$

From this, we give an underestimation of the energy:

$$\forall k, \quad \frac{1}{2}\left(1 - \left(c\frac{\Delta t}{\Delta x}\right)^2\right)\left\|\frac{u_h^{k+1} - u_h^k}{\Delta t}\right\|_{\Delta x} \le E_h(c)(u_h)^{k+\frac{1}{2}}.$$

Therefore we have the nonnegativity of the energy under CFL(ζ, ξ) conditions. For convergence, the key result is the overestimation of the energy:

$$\sqrt{E_h(c)(u_h)^{k+\frac{1}{2}}} \le \sqrt{E_h(c)(u_h)^{\frac{1}{2}}} + \frac{\sqrt{2}}{2\sqrt{2\xi - \xi^2}}\cdot\Delta t\cdot\sum_{j=1}^{k}\|i \mapsto s_h(i,j)\|_{\Delta x}$$

for all time t, with $k = \left\lfloor \frac{t}{\Delta t} \right\rfloor - 1$.

This completes the stability proof. In the inequality above, the energy is bounded for u_h, but the bound is actually valid for all the solutions of the discrete scheme, for any initial conditions and source term.

Note that the formal proof of stability closely follows the mathematical pen-and-paper proof and no additional hypotheses were found to be necessary.

4.5 Convergence

We prove that the convergence error is the solution of a scheme and therefore the results of Section 4.4 apply to it. More precisely, for all $\mathbf{\Delta x}$, the convergence error is solution of a discrete scheme with inputs

$$u_{0,j} = 0, \qquad u_{1,j} = \frac{e_j^1}{\Delta t}, \qquad \text{and} \qquad s_j^k = \varepsilon_j^{k+1},$$

where the errors refer to the errors of the initial scheme of the wave equation with grid steps $\mathbf{\Delta x}$. (Actual Coq notations depend on many more variables.)

We have proved many lemmas about the initializations of our scheme and of the convergence error. The idea is to prove that the initializations of the scheme are precise enough to guarantee that the initial convergence errors (at step 0 and 1) are accurate enough.

We also bounded the energy of the convergence error. Using results of Section 4.4, the proof reduces to bounding the sum of the source terms, here the truncation errors. Using results of Section 4.3, we prove this sum to be $O(\Delta x^2 + \Delta t^2)$. A few more steps conclude the proof.

Once more, the formal proof follows the pen-and-paper proof and progresses smoothly under the required hypothesis, including all the conditions on $\frac{\Delta t}{\Delta x}$ of Equation (14).

5 Conclusion and perspectives

One of the goals of this work is to favor the use of formal methods in numerical analysis. It may seem to be just wishful thinking, but it is actually seen as needed by some applied mathematicians. An early case led to the certification of the O∂yssée tool [10]. This tool performs automatic differentiation, which is one of the basic blocks for *gradient*-based algorithms. Our work tackles the converse problem: instead of considering derivation-based algorithms, we have formalized and proved part of the mathematical background behind integration-based algorithms.

This work shows there may be a synergy between applied mathematicians and logicians. Both domains are required here: applied mathematics for an initial proof that could be enriched upon request and formal methods for machine-checking it. This may be the reason why such proofs are scarce as this kind of collaboration is uncommon.

Proof assistants seem to mainly deal with algebra, but we have demonstrated that formalizing numerical analysis is possible too. We can confirm that pen-and-paper proofs are sometimes sketchy: they may be fuzzy about the needed hypotheses, especially when switching quantifiers. We have also learned that filling the gaps may cause us to go back to the drawing board and to change the basic blocks of our formalization to make them more generic (a big O that needs to be uniform and also generic with respect to a property P).

The formal bound on the error method, while of mathematical interest, is not sufficient to guarantee the correction of numerical applications implementing the three-point scheme. Indeed, such applications usually perform approximated computations, *e.g.*, floating-point computations, for efficiency and simplicity reasons. As a consequence, the proof of the method error has to be combined with a proof on the rounding error, in order to get a full-fledged correction proof. Fortunately, the proof on the rounding error has already been achieved [21]. We are therefore close to having a formal proof of both the numerical scheme and its floating-point implementation.

An advantage of Coq with respect to most other proof assistants is the ability to *extract* programs from proofs [22]. For this work, it does not make much sense to extract the algorithm from the proofs: not only is the algorithm already well-known, but its floating-point implementation was also certified [21]. So, an extraction of the algorithm would not bring much. However, extraction gives access to the constant C hidden behind the big O notation. Indeed, the proof of the floating-point algorithm relies on the discrete solution being good enough, so that the computed result does not diverge. Precisely, the convergence error has to be smaller than 1, and an extracted computation would be able to ensure this property. Furthermore, having access to this constant can be useful to the applied mathematicians for the a posteriori estimations needed for adaptive mesh refinements. Extraction also gives access to the α constant. That way, we could check that the constant $\mathbf{\Delta x}$ chosen in the C program described in [21] verifies this requirement. Note that performing an extraction requires to modify the definition of `Oup` so that it lives in `Set` instead of `Prop`. But this formalization change happens to be straightforward and Coq then succeeds in extracting mathematical formulas for constants α and C. Only basic operators (*e.g.* $+$, $\sqrt{\cdot}$, min) and constants (*e.g.* $t_{\max}$, ξ, χ_1, Taylor constants) appear in them, so they should be usable in practice.

The formal development is about 4500-line long. Its dependency graph is detailed in Figure 4. About half of the development is a reusable library described in Section 3 and the other half is the proof of convergence of the numerical scheme described in Section 4. This may seem a long proof for a single scheme for a single PDE. To put it into perspective, usual pen-and-paper proofs are 10-page long and an in-depth proof can be 60-page long. (We wrote one to ensure that we were not getting sidetracked.) So, at least from a length point of view, the formal proof is comparable to a detailed pen-and-paper proof.

In the end, the whole development contains only two axioms: the ε operator for the infinite dot product (see Section 3.1) and the finite support of the continuous solution of the wave equation (see Section 4.2). So, except for this last axiom which is related to the chosen PDE, the full numerical analysis proof of convergence is machine-checked and all required hypotheses are made clear. There is no loss of confidence due to this axiom, since the kind of proof and the results it is based upon are completely different from the ones presented here. Indeed, this axiom is about continuous solutions and hence much less error-prone.

For this exploratory work, we only considered the simple three-point scheme for the one-dimensional wave equation. Further works involve generalizing our approach to other schemes and other PDEs. We are confident that it would

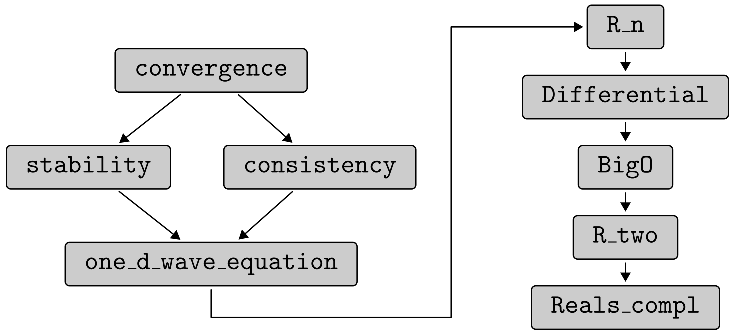

Fig. 4. Dependency graph of the Coq development. On the left are the files from the convergence proof. The other files correspond to the reusable library.

scale to higher-dimension and higher-order equations solved by discrete numerical schemes. However, the proofs of Section 4 are entangled with particulars of the presented problem, and would therefore have to be redone for other problems. So a more fruitful approach would be to prove once and for all the Lax equivalence theorem that states that consistency implies the equivalence between convergence and stability. This would considerably reduce the amount of work needed for tackling other schemes and equations.

This work also showed us that summations and finite support functions play a much more important role in the development than we first expected. We are therefore considering moving to the SSReflect interface and libraries for Coq [23], so as to simplify the manipulations of these objects in our forthcoming works.

References

1. Thomas, J.W.: Numerical Partial Differential Equations: Finite Difference Methods. Number 22 in Texts in Applied Mathematics. Springer (1995)
2. Zwillinger, D.: Handbook of Differential Equations. Academic Press (1998)
3. Dutertre, B.: Elements of Mathematical Analysis in PVS. In von Wright, J., Grundy, J., Harrison, J., eds.: Proceedings of the 9th International Conference on Theorem Proving in Higher Order Logics (TPHOLs'96). Volume 1125 of LNCS., Turku, Finland, Springer (1996) 141–156
4. Harrison, J.: Theorem Proving with the Real Numbers. Springer (1998)
5. Fleuriot, J.D.: On the Mechanization of Real Analysis in Isabelle/HOL. In Aagaard, M., Harrison, J., eds.: 13th International Conference on Theorem Proving and Higher-Order Logic (TPHOLs'00). Volume 1869 of LNCS., Springer (2000) 145–161
6. Mayero, M.: Formalisation et automatisation de preuves en analyses réelle et numérique. PhD thesis, Université Paris VI (2001)
7. Gamboa, R., Kaufmann, M.: Nonstandard Analysis in ACL2. Journal of Automated Reasoning **27**(4) (2001) 323–351
8. Geuvers, H., Niqui, M.: Constructive Reals in Coq: Axioms and Categoricity. In Callaghan, P., Luo, Z., McKinna, J., Pollack, R., eds.: Proceedings of the 1st

International Workshop on Types for Proofs and Programs (TYPES 2000). Volume 2277 of LNCS., Durham, United Kingdom, Springer (2002) 79–95
9. Cruz-Filipe, L.: A Constructive Formalization of the Fundamental Theorem of Calculus. In Geuvers, H., Wiedijk, F., eds.: Proceedings of the 2nd International Workshop on Types for Proofs and Programs (TYPES 2002). Volume 2646 of LNCS., Berg en Dal, Netherlands, Springer (2002)
10. Mayero, M.: Using Theorem Proving for Numerical Analysis (Correctness Proof of an Automatic Differentiation Algorithm). In Carreño, V., Muñoz, C., Tahar, S., eds.: 15th International Conference on Theorem Proving and Higher-Order Logic. Volume 2410 of LNCS., Hampton, VA, USA, Springer (2002) 246–262
11. Harrison, J.: A HOL Theory of Euclidean Space. In Hurd, J., Melham, T.F., eds.: 18th International Conference on Theorem Proving and Higher-Order Logic (TPHOLs'05). Volume 3603 of LNCS., Springer (2005) 114–129
12. Avigad, J., Donnelly, K.: A Decision Procedure for Linear "Big O" Equations. J. Autom. Reason. **38**(4) (2007) 353–373
13. Bécache, E.: Étude de schémas numériques pour la résolution de l'équation des ondes. Master Modélisation et simulation, Cours ENSTA, `http://www-rocq.inria.fr/~becache/COURS-ONDES/Poly-num-0209.pdf` (2009)
14. Achenbach, J.D.: Wave Propagation in Elastic Solids. North Holland, Amsterdam (1973)
15. Brekhovskikh, L.M., Goncharov, V.: Mechanics of Continua and Wave Dynamics. Springer (1994)
16. Newton, I.: Axiomata, sive Leges Motus. In: Philosophiae Naturalis Principia Mathematica. Volume 1., London (1687)
17. le Rond D'Alembert, J.: Recherches sur la courbe que forme une corde tendue mise en vibrations. In: Histoire de l'Académie Royale des Sciences et Belles Lettres (Année 1747). Volume 3. Haude et Spener, Berlin (1749) 214–249
18. John, F.: Partial Differential Equations. Springer (1986)
19. Courant, R., Friedrichs, K., Lewy, H.: On the partial difference equations of mathematical physics. IBM Journal of Research and Development **11**(2) (1967) 215–234
20. Bertot, Y., Castéran, P.: Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions. Texts in Theoretical Computer Science. Springer (2004)
21. Boldo, S.: Floats & Ropes: a case study for formal numerical program verification. In: Proceedings of the 36th International Colloquium on Automata, Languages and Programming. Volume 5556 of LNCS., Rhodos, Greece, Springer (2009) 91–102
22. Letouzey, P.: A New Extraction for Coq. In Geuvers, H., Wiedijk, F., eds.: Proceedings of the 2nd International Workshop on Types for Proofs and Programs (TYPES 2002). Volume 2646 of LNCS., Berg en Dal, Netherlands, Springer (2003)
23. Bertot, Y., Gonthier, G., Ould Biha, S., Pasca, I.: Canonical Big Operators. In: 21st International Conference on Theorem Proving in Higher Order Logics (TPHOLs'08). Volume 5170 of LNCS., Montreal, Canada, Springer (2008) 86–101