A formal proof of the Lax equivalence theorem for finite difference schemes

Mohit Tekriwal, Karthik Duraisamy, and Jean-Baptiste Jeannin

University of Michigan, Ann Arbor, MI 48109, USA
{tmohit,kdur,jeannin}@umich.edu

Abstract. The behavior of physical systems is typically modeled using differential equations which are too complex to solve analytically. In practical problems, these equations are discretized on a computational domain, and numerical solutions are computed. A numerical scheme is called convergent, if in the limit of infinitesimal discretization, the bounds on the discretization error is also infinitesimally small. The approximate solution converges to the "true solution" in this limit. The Lax equivalence theorem enables a proof of convergence given consistency and stability of the method.
In this work, we formally prove the Lax equivalence theorem using the Coq Proof Assistant. We assume a continuous linear differential operator between complete normed spaces, and define an equivalent mapping in the discretized space. Given that the numerical method is consistent (i.e., the discretization error tends to zero as the discretization step tends to zero), and the method is stable (i.e., the error is uniformly bounded), we formally prove that the approximate solution converges to the true solution. We then demonstrate convergence of the difference scheme on an example problem by proving both its consistency and stability, and then applying the Lax equivalence theorem. In order to prove consistency, we use the Taylor–Lagrange theorem by formally showing that the discretization error is bounded above by the n^{th} power of the discretization step, where n is the order of the truncated Taylor polynomial.

Keywords: Lax equivalence theorem · Finite difference scheme · Convergence · Taylor–Lagrange Theorem.

1 Introduction

Physical systems are typically modeled by differential equations. For instance, the aerodynamics of an airplane can be represented by the Navier–Stokes equations [1], which are too complex to solve analytically.

Since analytical solutions are intractable for most practical problems of interest, numerical solutions are sought in a discretized domain. The process of discretization in space and time results in approximate solutions to the governing equations. A numerical scheme is called *convergent*, if in the limit of infinitesimal discretization, the bound on the discretization error is also infinitesimally small. Under these conditions, the numerical solution converges or approaches

the analytic solution. This idea is formally articulated by the Lax equivalence theorem [26], which states that if a numerical method is *consistent* and *stable*, then it is *convergent*.

Proofs of consistency, stability, and convergence are typically performed by hand, making them prone to possible errors. Formal verification of mathematical proofs provides a much higher level of confidence of the correctness of manual proofs. Further, formal verification offers a pathway to leverage mathematical constructs therein, and to extend these proofs to more complex scenarios.

Recently, much effort has been dedicated to the definition of mathematical structures such as metric spaces, normed spaces, derivatives, limits etc. in a formal setting using proof assistants such as Coq [31,8,16,28]. Using automatic provers and proof assistants, a number of works have emerged in the formalization of numerical analysis [5]. Pasca has formalized the properties of the Newton method [32]. Mayero et al. [29] presented a formal proof, developed in the Coq system, of the correctness of an automatic differentiation algorithm. Besides Coq, numerical analysis of ordinary differential equations has also been done in Isabelle/ HOL [20]. Immler et al. [19,21,22], present a formalization of ordinary differential equations and the verification of rigorous (with guaranted error bounds) numerical algorithms in the interactive theorem prover Isabelle/HOL. The formalization comprises flow and Poincaré map of dynamical systems. Immler [18] implements a functional algorithm that computes enclosures of solutions of ODEs in the interactive theorem prover Isabelle/HOL. In [9], Brehard et al. present a library to verify rigorous approximations of univariate functions on real numbers, with the Coq proof assistant. Brehard [11], worked on rigorous numerics that aims at providing certified representations for solutions of various problems, notably in functional analysis. Work has also been done in formalizing real analysis for polynomials [12]. Boldo and co-workers [5,6,4] have made important contributions to formal verification of finite difference schemes. They proved consistency, stability and convergence of a second-order centered scheme for the wave equation. However, the Lax equivalence theorem – sometimes referred to as the fundamental theorem of numerical analysis – which is central to finite difference schemes, has not been formally proven in the general case.

In this paper, we present a formal proof of the Lax equivalence theorem for a general family of finite difference schemes. We use the definitions of consistency and stability and prove convergence. To prove the consistency of a second-order centered scheme for the wave equation, Boldo et al. [6] made assumptions on the regularity of the exact solution. This regularity is expressed as the existence of Taylor approximations of the exact solution up to some appropriate order. Our formalization instead takes the Taylor–Lagrange theorem of [28], to prove the consistency of a finite difference scheme of any order. It should be noted that the order of accuracy of an explicit finite difference scheme depends on the number of points in the discretized domain (called *stencils*) appearing in the numerical derivative. Our approach is to carry the Taylor series expansion for each of those stencils using the Taylor–Lagrange theorem, and appropriately instantiate the order of the truncated polynomial, to achieve the desired order of accuracy. By

incorporating the discretization error into the Lagrange remainder and proving an upper bound for the Lagrange remainder, we propose a rigorous method of proving consistency of a finite difference scheme.

Since the Lax equivalence theorem is an essential tool in the analysis of numerical schemes using finite differences, its formalization in the general case opens the door to the formalization and certification of finite difference-based numerical software. The present work will enable the formalization of convergence properties for a large class of finite difference numerical schemes, thereby providing formal proofs of convergence properties usually proved by hand, making explicit the underlying assumptions, and increasing the level of confidence in these proofs.

Overall this paper makes the following contributions:

- We provide a formalization in the Coq proof assistant of a general form of the Lax equivalence theorem.
- We prove consistency and stability of a second order accurate finite difference scheme for the example differential equation $\frac{d^2u}{dx^2} = 1$.
- We formally apply the Lax equivalence theorem on this finite difference scheme for the example differential equation, thereby formally proving convergence for this scheme.
- We also provide a generalized framework for a symmetric tri-diagonal (sparse) matrix in Coq. We define its eigen system and provide an explicit formulation of its inverse in Coq. We show that since the symmteric tri-diagonal matrix is normal, one can perform the stability analysis by just uniformly bounding the eigen values of the inverse. This is important because discretizations of mathematical model of physical systems are usually sparse [23].

This paper is structured as follows. In Section 2, we review the definitions of consistency, stability and convergence, state the Lax equivalence theorem [26,33], and discuss its formalization in the Coq proof assistant. In Section 3, we discuss the consistency of a finite difference scheme. In particular, we consider the central difference approximation of the second derivative and formally prove the order of accuracy using the Taylor–Lagrange theorem in the Coq proof assistant. We also relate the pointwise consistency of the finite difference scheme with the Lax equivalence theorem, by instantiating it with an example. In Section 4, we discuss the generalized formalization of a symmetric tri-diagonal matrix and later instantiate it with the scheme to prove stability of the scheme. In Section 5, we apply the Lax equivalence theorem to the concrete finite difference scheme that we are considering. In Section 6, we conclude by summarizing key takeaways from the paper, and discussing future work.

2 Lax equivalence theorem

In this section, we review the definitions of consistency, stability and convergence, discuss the problem set up and state the Lax equivalence theorem [26]. In this paper and for the formalization, we choose to follow the presentation of Sanz-Serna and Palencia [33]. We also discuss the proof of the Lax equivalence theorem which is then formalized in the Coq proof assistant.

2.1 Consistency, Stability and Convergence

Definition 1 (The Continuous Problem [33]). *Let X (the space of solutions) and Y (the space of data) be normed spaces, both real or both complex. We consider a linear operator A with domain $D \subset X$ and range $R \subset Y$. The problem to be solved is of the form*

$$Au = f, \qquad f \in Y \tag{1}$$

Here A is not assumed to be bounded, so that unbounded differential operators are included. The problem (1) is assumed to be well-posed, i.e., there exists a *bounded, linear operator*, $E \in B(Y, X)$, such that $EA = I$ in D, and that for $f \in Y$, equation (1) has a unique solution, $u = Ef$. Furthermore, the solution u depends continuously on the data.

Definition 2 (The Approximate Problem [33]). *Let H be a set of positive numbers such that 0 is the unique limit point of H. For each $h \in H$, let X_h, Y_h be normed spaces and consider the approximate or discretized problem*

$$A_h u_h = f_h, \qquad f_h \in Y_h \tag{2}$$

where A_h is a linear operator $A_h : X_h \longrightarrow Y_h$.

We assume that for each $h \in H$, problem (2) is well-posed and there exists a solution operator, $E_h = A_h^{-1}$, i.e. $u_h = E_h f_h$. The true solution u and the approximate solution u_h can be related with each other by defining a *bounded, linear operator*, $r_h : X \to X_h$ for each $h \in H$. Similarly, data $f \in Y$ can be related to data in a discrete space, $f_h \in Y_h$ by defining a restriction operator s_h. For each $h \in H$, $s_h : Y \to Y_h$ is also a *bounded, linear operator*. We assume that the operator norms can be uniformly bounded:

$$||r_h|| \leq C_1, \qquad ||s_h|| \leq C_2, \tag{3}$$

where the constants C_1, C_2 are independent of h. The true solution $u = Ef$ is compared with the discrete solution $u_h = E_h s_h f$ corresponding to the discretized datum f. The family $(X_h, Y_h, A_h, r_h, s_h)$ defines a *method* for the solution of (1) [33].

Definition 3 (Convergence [33]). *Let f be a given element in Y. The method $(X_h, Y_h, A_h, r_h, s_h)$ is convergent for the problem (1) if*

$$\lim_{h \to 0} ||r_h Ef - E_h s_h f||_{X_h} = 0 \tag{4}$$

We say that the method is convergent if it is convergent for each problem (1) for any f in Y.

Intuitively, this means that in the limit of the discretization step, h, tending to zero, the numerical solution $E_h s_h f$ approaches the analytical solution $r_h Ef$. The analytical solution $r_h Ef$ is the restriction of the true (analytical) solution, $u = Ef$, onto the grid of size $N = 1/h$, and $E_h s_h f$ is the discrete solution, $u_h = E_h f_h$ computed on the grid of size N.

Definition 4 (Consistency [33]). *Let u be a given element in D. The method is consistent at u if*

$$\lim_{h\to 0} ||A_h r_h u - s_h A u||_{Y_h} = 0 \tag{5}$$

A method is consistent if it is consistent at each u in a set D_o such that the image $A(D_o)$ is dense in Y.

Intuitively, this means that in the limit of the discretization step, h, tending to zero, the finite difference scheme $A_h u_h = f_h$ approaches the differential equation $Au = f$, i.e., we are discretizing the right differential equation.

Definition 5 (Stability [33]). *The method is stable if there exists a constant K such that*

$$||E_h||_{B(Y_h,X_h)} \leq K \tag{6}$$

Intuitively, stability of the numerical scheme means that a small numerical perturbation does not allow the solution to blow up. Uniform boundedness of the inverse $E_h = A_h^{-1}$ is a check on the conditioning of matrices (sensitivity to small perturbations), i.e., it ensures that the matrix A_h is not ill-conditioned. Thus, if the numerical problem (2) were unstable, even though we were trying to solve the right differential equation, we would never converge to the true solution. Hence, both stability and consistency are sufficient for proving convergence of the numerical scheme.

The quantities within the norms (4) and (5) are, respectively, the *global* and *local* discretization errors.

Theorem 1 (Lax equivalence theorem [33]). *Let $(X, Y, A, X_h, Y_h, A_h, r_h, s_h)$ be as above. If the method is consistent and stable, then it is convergent.*

Proof. We start with the definition of *convergence* in (4),

$$\begin{aligned}
&\lim_{h\to 0} ||r_h E f - E_h s_h f||_{X_h} \\
&= \lim_{h\to 0} ||r_h u - E_h s_h f||_{X_h} \quad (u \triangleq Ef) \\
&= \lim_{h\to 0} ||r_h u - E_h s_h A u||_{X_h} \quad (f \triangleq Au) \\
&= \lim_{h\to 0} ||I r_h u - E_h s_h A u||_{X_h} \quad (r_h u = I r_h u) \\
&= \lim_{h\to 0} ||E_h A_h r_h u - E_h s_h A u||_{X_h} \quad (E_h A_h \triangleq I) \\
&\leq \lim_{h\to 0} ||E_h||_{B(Y_h,X_h)} ||(A_h r_h u - s_h A u)||_{Y_h} \\
&\leq K \lim_{h\to 0} ||(A_h r_h u - s_h A u)||_{Y_h} \quad \text{(From stability: (6))} \\
&= 0 \quad \text{(From Consistency: (5))}
\end{aligned}$$

2.2 Formalization in the Coq Proof Assistant

In this Section we show how we formalized the proof of the Lax equivalence theorem [33] in the Coq proof assistant. All of the Coq formal proofs mentioned in this paper, containing the proofs of consistency, stability and convergence of

finite difference schemes, and of the Lax equivalence theorem, are available at `http://www-personal.umich.edu/~jeannin/papers/NFM21.zip`.

The `Coquelicot` library [8,7] defines mathematical structures required for implementing the proof. Since we use Coquelicot and standard reals library which are based on classical axiomatization of reals, our proofs are also non-constructive [8]. We define the *Banach spaces* (complete normed spaces, complete in the metric defined by the norm [25]) (X, Y, X_h, Y_h) using a canonical structure, `CompleteNormedModule`, in Coq [16].

The definitions of the true problem (1) and the approximate problem (2) require that the mappings $A : X \to Y$ and $A_h : X_h \to Y_h$ be linear, and the solution operators $E : Y \to X$ and $E_h : Y_h \to X_h$ be linear and bounded. The linear mappings A_h and E_h are defined as functions of $h \in \mathbb{R}$. Boldo et al. [3] have defined linear mapping in the context of a `ModuleSpace` and bounded linear mapping in the context of a `NormedModule` in their formalization of the *Lax Milgram Theorem* in Coq [2,15]. We extended these definitions in the context of `CompleteNormedModule`.

The definition of *consistency* (5) and *convergence* (4) hold in the limit of h tending to zero. Thus, an important step in the proof is to express these limits in Coq. Formally, the notion of f tending to l at the limit point x requires, for any $\epsilon > 0$, to find a neighborhood V of x such that any point u of V satisfies $|f(u) - l| < \epsilon$ [8]. This notion has been formalized in `Coquelicot` [7] using the concept of *filters*. In topology, a filter is a set of sets, which is nonempty, upward closed, and closed under intersection [13]. It is commonly used to express the notion of convergence in topology. We have used a filter, `locally x` [27] to denote an open neighborhood of x, and predicate `filterlim` [27] to formalize the notion of convergence (in the context of limits) of f towards l at limit point x, i.e. $\lim_{x \to a} f(x) = l$. Therefore, the definition of consistency (5) is expressed as:

```
(is_lim (fun h:R => norm (minus (Ah h (rh h u)) (sh h (A u)))) 0 0
```

where the limits of functions is expressed using `is_lim` [8].

We next discuss the formalization of the statement of convergence of a finite difference scheme in Coq. We note that from Theorem 1, *consistency* and *stability* imply *convergence*. This notion is expressed in Coq as follows:

```
(is_lim (fun h:R => norm (minus (Ah h (rh h u)) (sh h (A u))))  0 0
    (*Consistency*) /\
(exists K:R , forall (h:R), operator_norm(Eh h)<=K ) (* Stability*) ->
is_lim(fun h:R=>norm (minus (rh h (E(f))) (Eh h (sh h (f))))) 0 0)
    (*Convergence*).
```

where the *operator norm* is defined as $||f||_\phi = sup_{u \neq 0_E \wedge \phi(u)} \frac{||f(u)||_F}{||u||_E}$ and has been formally defined in [3].

The basic idea is that we bound the *global discretization error* ($||r_h E f - E_h s_h f||$) above using the stability criterion, i.e. $||r_h E f - E_h s_h f|| \leq K||A_h r_h u - s_h A u||$, and then prove that as the *local discretization error* ($||A_h r_h u - s_h A u||$) tends to zero in the limit of h tending to zero, the upper bound on the global discretization error tends to zero (using the property of limits). Using the property

of norm , i.e. $0 \leq ||r_h E f - E_h s_h f||$, we arrive at the inequality

$$0 \leq ||r_h E f - E_h s_h f|| \leq K||A_h r_h u - s_h A u||$$

In Coq, we define the lower bound of the inequality as a constant function with value 0 as: `fun _ => 0`. Since the limit of a constant function is the constant itself, i.e. $\lim_{h\to 0} 0 = 0$, and $lim_{h\to 0}||A_h r_h u - s_h A u|| = 0$ (Consistency), using the *sandwich theorem* for limits, $\lim_{h\to 0} ||r_h E f - E_h s_h f|| = 0$. The *sandwich theorem* states that if we have functions obeying the inequality: $f(x) \leq g(x) \leq h(x)$ and $\lim_{x\to a} f(x) = L \quad \wedge \quad \lim_{x\to a} h(x) = L$ on some open neighborhood of $x = a$, then $\lim_{x\to a} g(x) = L$. This proves the convergence of Definition 4 and completes the proof of the Lax equivalence theorem.

3 Proof of consistency of a sample finite difference scheme

A finite difference scheme (FD) approximates a differential equation with a difference equation. The derivatives are expressed in terms of function values at finite number of points in the dicretized domain. For instance, consider a simple differential equation, $\frac{d^2u}{dx^2} = 1$ on a domain $x \in (0, L)$ with boundary conditions $u(0) = 0$ and $u(L) = 0$, where L is the length of the domain. A second order accurate finite difference approximation would be $\frac{u(x+\Delta x)-2u(x)+u(x-\Delta x)}{\Delta x^2} = 1$, where Δx is the discretization step and x is the point at which the difference equation is evaluated. We will refer to this as numerical scheme $\mathcal{N}_h$. Since we are computing a numerical approximation to the actual derivatives, we are interested in knowing the order of the discretization error.

Definition 6 (Discretization error). *Let $D(u)$ denote the true derivative of a function $u : \mathbb{R} \to \mathbb{R}$ and $N(u)$ denote the finite difference approximation of the true derivative. The discretization error (commonly referred to as the truncation error) (τ) is then defined as:*

$$\tau \triangleq D(u) - N(u) \tag{7}$$

If the function u is *analytic*, it can be expressed as a *Taylor series expansion* at the point of evaluation. The truncation error is then evaluated by expressing the numerical derivatives in terms of a truncated Taylor polynomial and then taking a difference of the true derivative and the numerical derivative. This gives us an upper bound on the discretization error. If a numerical method is consistent, the truncation error can be expressed as:

$$\tau = \mathcal{O}(\Delta x^n)$$

when Δx tends to zero, and where n is the order of the truncated Taylor polynomial. We use this idea to formalize the proof of consistency of a finite difference scheme. This requires the use of an important theorem from calculus, the Taylor–Lagrange theorem.

Theorem 2 (Taylor–Lagrange theorem). *Suppose that f is $n + 1$ times differentiable on some interval containing the center of convergence c and x, and let $P_n(x) = f(c) + \frac{f^{(1)}(c)}{1!}(x-c) + \frac{f^2(c)}{2!}(x-c)^2 + .. + \frac{f^{(n)}(c)}{n!}(x-c)^n$ be the n^{th} order Taylor*

polynomial of f at $x = c$. Then $f(x) = P_n(x) + E_n(x)$ where $E_n(x)$ is the error term of $P_n(x)$ from $f(x)$. i.e. $E_n = f(x) - P_n(x)$, and for ξ between c and x, the Lagrange remainder form of the error E_n is given by the formula $E_n(x) = \frac{f^{n+1}(\xi)}{(n+1)!}(x-c)^{(n+1)}$.

Martin-Dorel et al. [28] proved the Taylor–Lagrange theorem formally in Coq, and it is available in the `Coq.Interval` library [30,10]. We used this formalization of the Taylor–Lagrange theorem to prove the consistency of a finite difference scheme.

We will specifically prove that for a central difference approximation of the second derivative, $\frac{d^2u}{dx^2}$, expressed as : $\frac{u(x+\Delta x)-2u(x)+u(x-\Delta x)}{(\Delta x)^2}$, the truncation error τ is quadratic in Δx:

$$\tau = \left| \frac{d^2u}{dx^2} - \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{(\Delta x)^2} \right| = \mathcal{O}(\Delta x^2)$$

3.1 Proof of consistency for the finite difference scheme

We want to prove that for a central difference approximation of the second derivative in the numerical scheme $\mathcal{N}_h$, the truncation error, $\tau = \mathcal{O}(\Delta x^2)$. By invoking the definition of Big-O notation, the theorem statement can be stated as:

$$\exists \gamma > 0, \Gamma > 0, \left| \frac{d^2u}{dx^2} - \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{(\Delta x)^2} \right| \leq \Gamma(\Delta x^2),\ 0 < |\Delta x| < \gamma. \tag{8}$$

The equation (8) is stated formally in Coq as:

```
Theorem taylor_FD (x:R): Oab x ->exists gamma:R, gamma >0 /\ exists G:R,
G>0/\ forall dx:R, dx>0 -> Oab (x+dx) -> Oab (x-dx)->(dx< gamma ->
Rabs((D 0 (x+dx)- 2*(D 0 x) + D 0 (x-dx))*/(dx * dx)- D 2 x)<= G*(dx^2)).
```

where `Oab x` mean $a < x < b$ and `D k x` denotes k^{th} derivative of u with respect to x.
We start by introducing the following lemmas required to complete the proof.

Lemma 1 **($|F(x)| \sim \mathcal{O}(\Delta x)^4$).** $\forall x \in (a,b), \exists\, \eta \in \mathbb{R}, \eta > 0 \wedge \exists\, M \in \mathbb{R}, M > 0 \wedge$
$\forall \Delta x \in \mathbb{R}, \Delta x > 0 \rightarrow (x + \Delta x) \in (a,b) \rightarrow \Delta x < \eta \rightarrow |F(x)| \leq M(\Delta x)^4$.

Here, $F(x)$ is the Lagrange remainder in the expansion of $u(x + \Delta x)$ up to degree 3 and is defined as:

$$F(x) \triangleq u(x+\Delta x) - u(x) - \Delta x \frac{du}{dx}\Big|_x - \frac{1}{2!}(\Delta x)^2 \frac{d^2u}{dx^2}\Big|_x - \frac{1}{3!}(\Delta x)^3 \frac{d^3u}{dx^3}\Big|_x \tag{9}$$

Thus, Lemma 1 states that the Lagrange remainder $F(x) = \frac{1}{4!}(\Delta x)^4 \frac{d^4u(\xi)}{dx^4}$ is of order $(\Delta x)^4$ for all $\xi \in (x, x + \Delta x)$.

Lemma 2 **($|G(x)| \sim \mathcal{O}(\Delta x)^4$).** $\forall x \in (a,b), \exists\, \delta \in \mathbb{R}, \delta > 0 \wedge \exists\, K \in \mathbb{R}, K > 0 \wedge$
$\forall \Delta x \in \mathbb{R}, \Delta x > 0 \rightarrow (x - \Delta x) \in (a,b) \rightarrow \Delta x < \delta \rightarrow |G(x)| \leq K(\Delta x)^4$.

Here, $G(x)$ is the Lagrange remainder in the expansion of $u(x-\Delta x)$ up to degree 3 and is defined as:

$$G(x) \triangleq u(x-\Delta x) - u(x) + \Delta x \frac{du}{dx}\Big|_x - \frac{1}{2!}(\Delta x)^2 \frac{d^2u}{dx^2}\Big|_x + \frac{1}{3!}(\Delta x)^3 \frac{d^3u}{dx^3}\Big|_x \tag{10}$$

Thus, Lemma 2 states that the Lagrange remainder $G(x) = \frac{1}{4!}(\Delta x)^4 \frac{d^4u(\xi)}{dx^4}$ is of order $(\Delta x)^4$ for all $\xi \in (x-\Delta x, x)$.

Both the lemmas are a straightforward application of the Taylor–Lagrange theorem (Theorem 2), and are crucial to the formalization of the proof of the consistency of the finite difference scheme.

Next, we present an informal proof of the theorem followed by a discussion on the formal proof of the consistency theorem.

Proof.

$$|F(x)| \leq M(\Delta x)^4 \quad \text{[From Lemma 1]} \tag{11}$$

$$|G(x)| \leq K(\Delta x)^4 \quad \text{[From Lemma 2]} \tag{12}$$

Adding equation (11) and (12), we get:

$$\begin{aligned} & |F(x)| + |G(x)| \leq (M+K)(\Delta x)^4 \\ \implies & |F(x) + G(x)| \leq (M+K)(\Delta x)^4 \\ & \text{[Using the triangle inequality, } (|F(x)+G(x)| \leq |F(x)| + |G(x)|) \text{]} \\ \implies & |F(x) + G(x)| \leq \Gamma(\Delta x)^4 \quad (\text{Instantiating} \Gamma := M+K) \end{aligned} \tag{13}$$

Unfolding the definitions $F(x)$ and $G(x)$, and doing the algebra we get:

$$\begin{aligned} & \Big|u(x+\Delta x) - 2u(x) + u(x-\Delta x) - (\Delta x)^2 \frac{d^2u}{dx^2}\Big| \leq \Gamma(\Delta x^4) \\ \implies & \Big|\frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{(\Delta x)^2} - \frac{d^2u}{dx^2}\Big| \leq \Gamma(\Delta x^2) \quad \textbf{[QED]} \end{aligned} \tag{14}$$

An important point to note is that the condition $|F(x)| + |G(x)| \leq M(\Delta x)^4 + K(\Delta x)^4$ holds when $0 < |\Delta x| < \gamma$, where γ is as defined in (8). We therefore choose, $\gamma = min(\eta, \delta)$, where η is such that, $|F(x)| \leq M(\Delta x)^4$ holds when $0 < |\Delta x| < \eta$, and δ is such that, $|G(x)| \leq K(\Delta x)^4$ holds when $0 < |\Delta x| < \delta$.

3.2 Formalization in the Coq Proof assistant

We followed the proof above and formalized it in the Coq proof assistant. To apply the Taylor–Lagrange theorem [28] to the consistency analysis of a central difference approximation, we broke down the theorem statement into two lemmas as discussed in the previous section. Therefore, in this section, we will discuss the proof of Lemma 1 and 2.

Proof of Lemma 1: Formally Lemma 1 is stated in Coq as:

```
Lemma taylor_uupper (x:R): Oab x-> exists eta: R, eta>0 /\
    exists M :R, M>0  /\ forall dx:R, dx>0 -> Oab (x+dx) ->
    (dx<eta -> Rabs(D 0 (x+dx)- Tsum 3 x (x+dx))<=M*(dx^4)).
```

In the proof of the Lemma, existential quantification associated with η and M has to be addressed. We chose η as $b - x$, since the interval in which we are studying Taylor–Lagrange for $u(x + \Delta x)$ is $[x, b]$. Since $\Delta x \in (x, b)$ and $\Delta x < \eta$, it seems logical to chose $\eta = b - x$. For the choice of M, we obtained extreme bounds in the interval. Since the function u and its derivatives are continuous in a compact set $[x, b]$, we are guaranteed to get maximum and minimum values. In Coq, we applied the lemma `continuity_ab_max` to obtain a maximum value, $\left(\frac{d^4u}{dx^4}\right)_{max} = \frac{d^4u(F)}{dx^4}$ such that $\frac{d^4u(\xi)}{dx^4} \leq \frac{d^4u(F)}{dx^4}, \forall \xi \in [x, b]$. Similarly, we apply the lemma `continuity_ab_min` to obtain a minimum value, $\left(\frac{d^4u}{dx^4}\right)_{min} = \frac{d^4u(G)}{dx^4}$ such that $\frac{d^4u(G)}{dx^4} \leq \frac{d^4u(\xi)}{dx^4}, \forall \xi \in [x, b]$.
Thus, M is chosen as $M = max\left(\left|\frac{d^4u(G)}{dx^4}\right|, \left|\frac{d^4u(F)}{dx^4}\right|\right)$. With this choice of M, we can bound the Lagrange remainder or the trunction error from above and thus prove Lemma 1.

Proof of Lemma 2: Formally Lemma 2 is stated in Coq as:

```
Lemma taylor_ulower (x:R): Oab x -> exists delta: R, delta>0 /\
    exists K :R, K>0 /\ forall dx:R, dx>0 ->Oab (x-dx) ->
    (dx<delta -> Rabs(D 0 (x-dx)-Tsum 3 x (x-dx))<=K*(dx^4)).
```

The proof of Lemma 2 follows the same approach as that of Lemma 1. Here, we chose δ as $x - a$, since the interval in which we are studying Taylor–Lagrange theorem for $u(x - \Delta x)$, $\Delta x \in (a, x)$, and $\Delta x < \delta$. We chose K in the same way as we chose M in Lemma 1 except that the interval in which we obtain maximum and minimum values for $\frac{d^4u}{dx^4}$ is $[a, x]$ in this case. Thus, $\left(\frac{d^4u}{dx^4}\right)_{min} = \frac{d^4u(G)}{dx^4}$, $\left(\frac{d^4u}{dx^4}\right)_{max} = \frac{d^4u(F)}{dx^4}$,and $K = max\left(\left|\frac{d^4u(G)}{dx^4}\right|, \left|\frac{d^4u(F)}{dx^4}\right|\right), \forall c \in [a, x]$.

To prove the main theorem statement on consistency, we break the statement into Lemma 1 and 2, by instantiating $\Gamma = M + K$, and $\gamma = \min(\eta, \delta)$, where (M, η) and (K, δ) have been defined as in Lemma 1 and 2 respectively, in the manner shown in section (3.1). To implement this instantiation, we have to carefully *destruct* the lemmas introduced in the theorem statement. Then, we simply apply lemma 1 and 2, to complete the main proof.

3.3 Relating pointwise consistency to the Lax equivalence theorem

In this section, we relate the proof of consistency from Section 3.1 with the Lax equivalence Theorem 1. The numerical discretization of the differential equation can be expressed in the discrete domain as:

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \vdots \\ 0 & \dots & 1 & -2 & 1 & 0 \\ 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & \dots & 0 & 0 & 0 & 1 \end{bmatrix}}_{A_h} \underbrace{\begin{bmatrix} u_o \\ u_1 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{bmatrix}}_{r_h u} = \underbrace{\begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 0 \end{bmatrix}}_{s_h Au} \tag{15}$$

Comparing with the statement of consistency (5), we have

$$\lim_{h\to 0}\left|\left|\frac{1}{h^2}\begin{bmatrix}1&0&0&0&\dots&0\\1&-2&1&0&\dots&0\\\vdots&\ddots&\ddots&\ddots&&\vdots\\0&\dots&1&-2&1&0\\0&\dots&0&1&-2&1\\0&\dots&0&0&0&1\end{bmatrix}\begin{bmatrix}u_o\\u_1\\\vdots\\u_{N-2}\\u_{N-1}\\u_N\end{bmatrix}-\begin{bmatrix}0\\1\\\vdots\\1\\1\\0\end{bmatrix}\right|\right| = \lim_{h\to 0}\left|\left|\begin{bmatrix}\frac{u_o}{h^2}\\\frac{u_o-2u_1+u_2}{h^2}-1\\\frac{u_1-2u_2+u_3}{h^2}-1\\\vdots\\\frac{u_{N-2}-2u_{N-1}+u_N}{h^2}-1\\\frac{u_N}{h^2}\end{bmatrix}\right|\right| = 0 \tag{16}$$

Taking the vector norm in the L_1 sense, $||.||_1$, equation (16) can be written as:

$$\lim_{h\to 0}\left[\left|\frac{u_o}{h^2}\right|+\left|\frac{u_o-2u_1+u_2}{h^2}-1\right|+..+\left|\frac{u_{N-2}-2u_{N-1}+u_N}{h^2}-1\right|+\left|\frac{u_N}{h^2}\right|\right]=0 \tag{17}$$

$\lim_{h\to 0}\frac{u_o}{h^2}=0$ and $\lim_{h\to 0}\frac{u_N}{h^2}=0$, trivially because of the boundary conditions we imposed, i.e. $u_o=0$ and $u_N=0$. The norm used in (16) are in the space Y_h, i.e., $||.||_{Y_h}$.
This reduces to proving:

$$\sum_{i=1}^{N-1}\lim_{h\to 0}\left|\frac{u_{i-1}-2u_i+u_{i+1}}{h^2}-1\right|=0 \tag{18}$$

But from the Taylor–Lagrange analysis discussed in section (3.1), we have

$$\left|\frac{u_{i-1}-2u_i+u_{i+1}}{h^2}-\frac{d^2u}{dx^2}\Big|_{x_i}\right|\le Ch^2 \tag{19}$$

where C is a constant, and $u_i=u(x_i), u_{i-1}=u(x_i-h), u_{i+1}=u(x_i+h)$. Substituting $\frac{d^2u}{dx^2}\Big|_{x_i}=1$, and using the inequality (19) and equation(18), we get

$$\sum_{i=1}^{N-1}0\le\sum_{i=1}^{N-1}\lim_{h\to 0}\left|\frac{u_{i-1}-2u_i+u_{i+1}}{h^2}-1\right|\le\sum_{i=1}^{N-1}\lim_{h\to 0}|Ch^2| \tag{20}$$

But, $\sum_{i=1}^{N-1}\lim_{h\to 0}|Ch^2|=0$. Hence, using the sandwich theorem, we prove that

$$\sum_{i=1}^{N-1}\lim_{h\to 0}\left|\frac{u_{i-1}-2u_i+u_{i+1}}{h^2}-1\right|=0 \qquad \textbf{[QED]}$$

3.4 Formalization in Coq

In order to represent, $x_i,\ i=0..N$, we define x of type: `nat → R`. The boundary conditions are imposed as hypothesis statements:

```
Hypothesis u_0 : (D 0 (x 0))= 0.
Hypothesis u_N: (D 0 (x N)) =0.
```

The differential equation is defined as:

```
Hypothesis u_2x: forall i:nat, (D 2 (x i)) =1.
```

Equation (18) is formalized as a lemma statement:

```
Lemma lim_sum:is_lim (fun h:R =>
  sum_n_m (fun i:nat =>Rabs (( D 0 (x i -h) -2* (D 0 (x i))
    + D 0 (x i +h))*/(h^2) -1)) 1%nat (pred N)) 0 0.
```

This is where we integrate the proof of pointwise consistency of the FD scheme from section (3).

The main theorem statement which is an application of the statement of consistency required in the proof of Lax equivalence theorem from section (2) is as follows:

```
Theorem consistency_inst: forall (U:X) (f:Y) (h:R) (uh: Xh h)
 (rh: forall (h:R), X -> (Xh h)) (sh: forall (h:R), Y->(Yh h))
 (E: Y->X) (Eh:forall (h:R),(Yh h)->(Xh h)),
 is_lim (fun h:R => norm (minus (Ah h (rh h U)) (sh h (A U)))) 0 0.
```

We note here that the above-mentioned formalization is not unique to the second order scheme that we discussed. The approach we discuss can easily be generalized to verify consistency of any finite difference scheme. The crucial step in such a generalization is the appropriate instantiation of the A_h matrix and the vectors $r_h u$ and $s_h A u$.

4 Stability of the scheme

In this section we discuss the stability of the scheme $\mathcal{N}_h$. From section 2, stability of a numerical scheme requires the solution operator $E_h = A_h^{-1}$ to be uniformly bounded. We prove this by bounding the eigenvalues of E_h uniformly. Eigenvalues of E_h are just inverse of the eigenvalues of A_h. A formal proof of this can be referred to in the Appendix B.2.

We will first discuss a generalized framework for the formalization of stability for a symmetric tri-diagonal matrix in Coq. We denote this matrix with $A_h(a, b, c)$ with $c = a$ for symmetry. This notation means that b is on the diagonal, c is on the upper diagonal and a is on the lower diagonal. All the other entries are zero. Since we are treating stability from a spectral viewpoint, we next discuss the formalization of the Eigen system for $A_h(a, b, a)$.

4.1 Lemma to verify that the eigenvalues and eigenvectors belong to the spectrum of $A_h(a, b, a)$

Analytical expressions for the eigenvalues and eigenvectors of $A_h(a, b, c)$ are given by:

$$\lambda_m = b + 2\sqrt{ac}\cos\left[\frac{m\pi}{N+1}\right]; \quad s_m = (s_j)_m = \left[\frac{a}{c}\right]^{j-1/2}\sqrt{\frac{2}{N+1}}\sin\left[j\frac{m\pi}{N+1}\right]$$

$\forall m, j = 1..N$. In Coq, we defined λ_m and s_m as follows:

```
Definition Eigen_vec (m N:nat) (a b c:R):= mk_matrix N 1%nat (fun i j =>
    sqrt ( 2 / INR (N+1))*(Rpower (a */c) (INR i +1 -1*/2))*
        sin(((INR i +1)*INR(m+1)*PI)*/INR (N+1))).

Definition Lambda (m N:nat) (a b c:R):= mk_matrix 1%nat 1%nat (fun i j =>
    b + 2* sqrt(a*c)* cos ( (INR (m+1) * PI)*/INR(N+1))).
```

Since naturals in Coq start with 0, we write `INR (m+1)` and `INR i+1`.

We then formally verify that the analytical expressions for the pair (λ_m, s_m) indeed belong to the spectrum of A_h. From now on, we will refer to $A_h(a, b, a)$ as A_h for the sake of brevity. In Coq, we state this formally as:

```
Lemma eigen_belongs (a b c:R): forall (m N:nat), (2 < N)%nat ->
    (0 <= m < N)%nat -> a=c /\ 0<c-> (LHS m N a b c) = (RHS m N a b c).
```

where, $LHS \triangleq A_h s_m$ and $RHS \triangleq s_m \lambda_m$. Here we used the definition of eigenvalue-eigenvector, i.e., $A_h s_m \triangleq \lambda_m s_m$. Formalizing the proof of the lemma `eigen_belongs` was challenging due to the structure of the matrix A_h. A_h is a tri-diagonal matrix with non-zero entries on the diagonal, sub-diagonal and super-diagonal. The other entries are zero and hence the matrix is sparse.

$$\therefore \underbrace{\sum_{j=0}^{N-1} A_h(i,j)s_m(i)}_{A_h(i,j)\neq 0} + \underbrace{\sum_{j=0}^{N-1} A_h(i,j)s_m(i)}_{A_h(i,j)=0} = \lambda_m s_m(i); \quad 0 \leq i \leq N-1 \tag{21}$$

In Coq, we have to carefully destruct the matrix A_h to separate the non-zero and zero sums in the LHS of equation (21). The idea is to do a case analysis on the row-index i, and has been illustrated in figure (1) in the Appendix B.4. Details on the formal proof of the zero and non-zero cases are presented in Appendix B.4.

Next, we discuss formalization of the boundedness of the matrix norm of $E_h = A_h^{-1}$. We have used an explicit formulation of A_h^{-1} [17] in our formalization and we verify this formally using the definition: $A_h^{-1} A_h = I \ \wedge \ A_h A_h^{-1} = I$. Details on the proof can be referred to in the Appendix B.1.

4.2 Lemma on the boundedness of the matrix norm for scheme $\mathcal{N}_h$

Here, we have used the definition of the spectral (2-norm): $||A||_2 = \rho(A)$, where $\rho(A)$ is the spectral radius of A and is defined as the maximum eigen-value of A, i.e. $\rho(A) = max_m|\lambda_m(A)|$. For the symmetric tri-diagonal matrix A_h, $A = E_h$ and $\lambda_m(E_h) = 1/\lambda_m(A_h)$. Since $\lambda_m(A_h) < 0$, $max_m|\lambda_m(E_h)| = 1/|\lambda_{min}(A_h)|$. Hence, we define the matrix norm in Coq as follows:

```
Definition matrix_norm (N:nat):= 1/ Rabs (Lambda_min N).
```

To show that the matrix norm is uniformly bounded, we need to show that $1/|\lambda_{min}(A_h)|$ is uniformly bounded. This is where we instantiate the tri-diagonal matrix A_h with the scheme $\mathcal{N}_h$. Thus, we prove the following lemma in Coq:

```
Lemma spectral: forall(N:nat),(2<N)%nat -> 1/Rabs(Lambda_min N) <= L^2/4.
```

where L is the length of the domain, independent of h, and is constant throughout. `Lambda_min` is the minimum eigenvalue for the instantiated matrix, $A'_h = A_h(\frac{1}{h^2}, \frac{-2}{h^2}, \frac{1}{h^2})$. We provide a paper proof of this bound in the Appendix A.

To show that all the eigenvalues have the same bound, we prove that $\frac{1}{\lambda_{min}(A'_h)}$ is the maximum eigenvalue of E'_h. The lemma statement is as follows:

```
Lemma eigen_relation: forall (i N:nat), (2<N)%nat ->(0<=i<N)%nat ->
    Rabs (lam i N) <= 1/ Rabs( Lambda_min N).
```

This completes the proof on the boundedness of the eigenvalues of E'_h. The lemma, `eigen_relation` also shows that the spectral radius of E'_h is $\frac{1}{|\lambda_{min}(A'_h)|}$, and justifies the defintion of `matrix_norm`.

We note that the definition of the matrix norm of A_h^{-1} is valid only if A_h^{-1} is a normal matrix . We therefore verify that A_h^{-1} is normal. The lemma statement is provided in the Appendix B.3.

We also provide the proof that A_h is diagonalizable in the Appendix C. This helps us to formally establish that the eigen vectors are orthogonal and hence the eigen space is complete.

4.3 Main stability theorem

In this section, we integrate all of the previous lemmas to prove the main stability theorem (6).

```
Theorem stability: forall (u:X) (f:Y) (h:R) (uh: Xh h)
    (rh: forall (h:R), X -> (Xh h))(sh: forall (h:R), Y->(Yh h))
    (E: Y->X) (Eh:forall (h:R), (Yh h)->(Xh h)),
    exists K:R , forall (h:R), operator_norm(Eh h)<=K.
```

where the operator norm is instantiated with the matrix norm using the following hypothesis:

```
Hypothesis mat_op_norm: forall (u:X) (f:Y) (h:R) (uh: Xh h)
    (rh: forall (h:R), X -> (Xh h))(sh: forall (h:R), Y->(Yh h))
    (E: Y->X) (Eh:forall (h:R),(Yh h)->(Xh h)),
    operator_norm (Eh h) = matrix_norm m.
```

5 Application of the Lax equivalence theorem to the example problem

In this section, we apply the Lax equivalence theorem that we proved in Section 2 to a concrete differential equation $\frac{d^2u}{dx^2} = 1$ and the numerical scheme $\mathcal{N}_h$ given by $\frac{u_{i+1}-2u_i+u_{i-1}}{\Delta x^2} = 1$. We recall that the proof of convergence using the Lax equivalence theorem requires that the difference scheme is consistent with respect to the differential equation and is stable. We discussed the proof of consistency of the scheme in Section 3 and the stability in Section 4. Thus, we apply these proofs to complete the proof of convergence for the scheme. We provide the theorem statement to verify convergence of the scheme in the Appendix D.

6 Conclusion and Future work

This work investigated the formalization of convergence, stability and consistency of a finite difference scheme in the Coq proof assistant. Any continuously differentiable function can be approximated by a Taylor polynomial. The Lagrange remainder of a Taylor series provides an estimate of the truncation error and we formally proved that this error can be bound by n^{th} power of the discretization step, Δx, where $n-1$ is the order of the Taylor polynomial. We implemented the proof of the consistency of a finite difference scheme by breaking down the theorem statement into lemmas, each corresponding to function values at points neighboring the point of evaluation. These lemmas were proved

individually by applying the Taylor–Lagrange theorem, the proof of which is already formalized in the `Coq.Interval` library [28]. Consistency and stability guarantees convergence as stated by the Lax equivalence theorem. Following the proof of the the Lax equivalence theorem, we formally proved convergence of a specific finite difference scheme. Specifically, we proved that the global discretization error could be bounded above by a constant times the local discretization error. Then, by applying the sandwich theorem for limits, we proved that the convergence condition is satisfied in the limit $\Delta x \to 0$. In the process of formalizing the proof of stability for the numerical scheme, we also developed tools for linear algebra and spectral theory, for the `Coquelicot` definition of matrices in Coq, which can be reused. As noted earlier, the approach we follow is not specific to the sample numerical scheme, but can be easily extended to other numerical schemes with appropriate instantiation of the matrix A_h, and vectors, $r_h u$, $s_h A u$. Formalization of the proof of orthogonality of the eigenvectors helped us report the missing constant $\sqrt{\frac{2}{N+1}}$ in s_m that occurs in most textbooks/literature on numerical analysis.

This work considered the impact of the discretization error on the convergence of a numerical method to the exact solution. In a practical setting, floating point errors have to be also accounted for, as an accumulation of such errors can lead to deviations from the true solution. In future work, we will extend our results to incorporate floating point errors and their impact on the convergence of finite difference numerical schemes. We also plan on working with iterative solvers, which would be an extension of our current work on direct solvers (explicit inversion of the matrix A_h). We also plan on working with the Frama-C toolkit [14] for verification of existing programs and be able to discharge the generated verification conditions using the Coq proofs we present in this paper.

6.1 Effort and challenges:

The total length of the Coq code and proofs is about 14,000 lines, of which about 1200 lines are specific to the scheme. The rest of the formalization can be reused for a generic symmetric tridiagonal matrix. It took us about 15 months for the entire formalization. Much of the effort was spent on destructing the matrices and developing required linear algebra tools to handle the matrix manipulation. Since we are treating stability from a spectral point of view, lack of spectral theory for numerical analysis for the `Coquelicot` definition of matrices has been challenging for us. For the proof of consistency, the primary challenge was the right placement of the quantifiers to bound the Lagrange remainder using the definition of big-O notation. To instantiate $\Gamma = M + K$, we had to carefully destruct the lemmas into the main theorem. We believe that a generic library with an automated implementation of the big-O definitions would save considerable effort here. We also encountered issues in selecting appropriate instantiations for other existential parameters. In the proof of convergence, we had to carefully construct the application of properties of limit with filters of neighborhoods.

References

1. Navier-stokes equations. https://www.grc.nasa.gov/WWW/k-12/airplane/nseqs.html, (Accessed on 9/20/2020)
2. Boldo, S., Clément, F., Faissole, F., Martin, V., Mayero, M.: Elfic Coq library for formalization of Lax-Milgram theorem. https://www.lri.fr/~sboldo/elfic/index.html, (Accessed on 9/20/2020)
3. Boldo, S., Clément, F., Faissole, F., Martin, V., Mayero, M.: A Coq formal proof of the Lax-Milgram theorem. In: Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs. pp. 79–89. ACM (2017)
4. Boldo, S., Clément, F., Filliâtre, J.C., Mayero, M., Melquiond, G., Weis, P.: Formal proof of a wave equation resolution scheme: the method error. In: International Conference on Interactive Theorem Proving. pp. 147–162. Springer (2010)
5. Boldo, S., Clément, F., Filliâtre, J.C., Mayero, M., Melquiond, G., Weis, P.: Wave equation numerical resolution: a comprehensive mechanized proof of a c program. Journal of Automated Reasoning **50**(4), 423–456 (2013)
6. Boldo, S., Clément, F., Filliâtre, J.C., Mayero, M., Melquiond, G., Weis, P.: Trusting computations: a mechanized proof from partial differential equations to actual program. Computers & Mathematics with Applications **68**(3), 325–352 (2014)
7. Boldo, S., Lelay, C., Melquiond, G.: Hierarchy Coq library. http://coquelicot.saclay.inria.fr/html/Coquelicot.Hierarchy.html, (Accessed on 9/20/2020)
8. Boldo, S., Lelay, C., Melquiond, G.: Coquelicot: A user-friendly library of real analysis for Coq. Mathematics in Computer Science **9**(1), 41–62 (2015)
9. Bréhard, F., Mahboubi, A., Pous, D.: A certificate-based approach to formally verified approximations. In: ITP 2019-Tenth International Conference on Interactive Theorem Proving. pp. 1–19 (2019)
10. Brisebarre, N., Joldeş, M., Martin-Dorel, É., Mayero, M., Muller, J.M., Paşca, I., Rideau, L., Théry, L.: Rigorous polynomial approximation using taylor models in Coq. In: NASA Formal Methods Symposium. pp. 85–99. Springer (2012)
11. Bréhard, F.: Numerical computation certified in functional spaces: A trilogue between rigorous polynomial approximations, symbolic computation and formal proof. Ph.D. thesis (2019)
12. Cohen, C.: Formalizing real analysis for polynomials (2010)
13. Cohen, C., Rouhling, D.: A formal proof in Coq of LaSalle's invariance principle. In: International Conference on Interactive Theorem Proving. pp. 148–163. Springer (2017)
14. Cuoq, P., Kirchner, F., Kosmatov, N., Prevosto, V., Signoles, J., Yakobowski, B.: Frama-c. In: Eleftherakis, G., Hinchey, M., Holcombe, M. (eds.) Software Engineering and Formal Methods. pp. 233–247. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
15. Faissole, F.: Library on lax-milgram theorem (coqlm). https://www.lri.fr/~faissole/these_coq.html, (Accessed on 12/30/2019)
16. Garillot, F., Gonthier, G., Mahboubi, A., Rideau, L.: Packaging mathematical structures. In: International Conference on Theorem Proving in Higher Order Logics. pp. 327–342. Springer (2009)
17. Hu, G., O'Connell, R.F.: Analytical inversion of symmetric tridiagonal matrices. Journal of Physics A: Mathematical and General **29**(7), 1511 (1996)
18. Immler, F.: Formally verified computation of enclosures of solutions of ordinary differential equations. In: Badger, J.M., Rozier, K.Y. (eds.) NASA Formal Methods. pp. 113–127. Springer International Publishing, Cham (2014)

19. Immler, F.: A Verified ODE Solver and Smale's 14th Problem. Dissertation, Technische Universität München, München (2018)
20. Immler, F., Hölzl, J.: Numerical analysis of ordinary differential equations in isabelle/hol. In: International Conference on Interactive Theorem Proving. pp. 377–392. Springer (2012)
21. Immler, F., Traut, C.: The flow of odes. In: International Conference on Interactive Theorem Proving. pp. 184–199. Springer (2016)
22. Immler, F., Traut, C.: The flow of odes: Formalization of variational equation and poincaré map. Journal of Automated Reasoning **62**(2), 215–236 (2019)
23. Kirk, D.B., mei W. Hwu, W.: Chapter 10 - parallel patterns: Sparse matrix–vector multiplication: An introduction to compaction and regularization in parallel algorithms. In: Kirk, D.B., mei W. Hwu, W. (eds.) Programming Massively Parallel Processors (Second Edition), pp. 217 – 234. Morgan Kaufmann, Boston, second edition edn. (2013). https://doi.org/https://doi.org/10.1016/B978-0-12-415992-1.00010-9, http://www.sciencedirect.com/science/article/pii/B9780124159921000109
24. Knapp, M.P.: Sines and cosines of angles in arithmetic progression. Mathematics magazine **82**(5), 371 (2009)
25. Kreyszig, E.: Introductory functional analysis with applications, vol. 1. wiley New York (1978)
26. Lax, P.D., Richtmyer, R.D.: Survey of the stability of linear finite difference equations. Communications on pure and applied mathematics **9**(2), 267–293 (1956)
27. Lelay, C.: How to express convergence for analysis in coq (2015)
28. Martin-Dorel, É., Rideau, L., Théry, L., Mayero, M., Pasca, I.: Certified, efficient and sharp univariate taylor models in coq. In: 2013 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. pp. 193–200. IEEE (2013)
29. Mayero, M.: Using theorem proving for numerical analysis correctness proof of an automatic differentiation algorithm. In: International Conference on Theorem Proving in Higher Order Logics. pp. 246–262. Springer (2002)
30. Melquiond, G., Érik Martin-Dorel, Mayero, M., Pasca, I., Rideau, L., Théry, L.: Interval Coq Library. http://coq-interval.gforge.inria.fr/, (Accessed on 9/20/2020)
31. O'Connor, R.: Certified exact transcendental real number computation in coq. In: International Conference on Theorem Proving in Higher Order Logics. pp. 246–261. Springer (2008)
32. Pasca, I.: Formal Verification for Numerical Methods. Ph.D. thesis, Université Nice Sophia Antipolis (2010)
33. Sanz-Serna, J., Palencia, C.: A general equivalence theorem in the theory of discretization methods. Mathematics of computation **45**(171), 143–152 (1985)

A Proof of the uniform bound on the eigen values of $A_h(1/h^2, -2/h^2, 1/h^2)$

In this section, we provide a paper proof of the uniform boundedness of the eigenvalues of the scheme $\mathcal{N}_h$.

Proof.

$\lambda_{min}(A'_h) = \frac{2}{h^2}\left[-1+\cos\left(\frac{\pi}{N+1}\right)\right]$ [For m=1 in the expression of λ_m]

Since all eigenvalues are negative, $\min|\lambda_m(A'_h)| = |\lambda_{min}(A'_h)|$,

$$\therefore \frac{1}{|\lambda_{min}(A_{h'})|} = \frac{1}{\left|\frac{2}{h^2}\left[-1+\cos\left(\frac{\pi}{N+1}\right)\right]\right|} \implies \frac{1}{|\lambda_{min}(A'_h)|} = \frac{h^2}{4\sin^2\left(\frac{\pi}{2(N+1)}\right)}$$

[Using the identity: $-1+\cos(2x) = -2\sin^2(x)$]

Using the definition, $h \triangleq \frac{L}{N+1}$, where L is the domain length,

$$\therefore \frac{1}{|\lambda_{min}(A'_h)|} = \frac{L^2}{4(N+1)^2\sin^2\left(\frac{\pi}{2(N+1)}\right)} = \frac{L^2}{\pi^2}\frac{\pi^2}{4(N+1)^2\sin^2\left(\frac{\pi}{2(N+1)}\right)} = \frac{L^2}{\pi^2}\frac{x^2}{\sin^2(x)}$$

where, $x = \frac{\pi}{2(N+1)}$

Using the relation, $\forall x \in (0,\pi/2], \quad \frac{2x}{\pi} \leq \sin(x),$ or, $\frac{x}{\sin(x)} \leq \frac{\pi}{2}$, we get : $\frac{x^2}{\sin^2(x)} \leq \frac{\pi^2}{4}$

$\therefore \frac{1}{|\lambda_{min}(A'_h)|} \leq \frac{L^2}{4}$ **[QED]**

We prove the relation $\forall x \in (0,\pi/2], \quad \frac{x}{\sin(x)} \leq \frac{\pi}{2}$, by using the concavity of $\sin(x)$ in $[0,\pi/2]$. We define a concave function $f : \mathbb{R} \to \mathbb{R}$ in Coq as follows:

```
Definition concave (f:R->R) (x y c:R):=
    0<=c<=1 -> f(c*x + (1-c) * y) >= c* f x + (1-c) * f y.
```

The proof for $\frac{x^2}{\sin^2(x)} \leq \frac{\pi^2}{4}, \ \forall x \in (0,\pi/2]$ is formalized as the following lemma statement in Coq:

```
Lemma spectral_intermed:forall(x:R),0<x<=PI/2 ->(x^2)/(sin x)^2 <=(PI^2)/4.
```

B Lemmas required to complete the proof of stability:

B.1 Lemma to verify the invertibility of A_h

In this subsection, we verify that the explicit form of the inverse [17] we use is indeed the inverse of A_h, i.e. $A_h A_h^{-1} = A_h^{-1} A_h = I$. In Coq, we state the following lemma to verify the invertibility of A_h:

```
Lemma invertible_check (a b:R) : forall (N:nat), (2<N)%nat -> 0<a ->
    Mk N (b/a) <> 0 -> invertible N (Ah N a b a ) (inverse_A N a b ).
```

Here, M_k is the determinant of A_h of size k. We used the recurrence relation [17]: $M_k = D \times M_{k-1} - M_{k-2}, \quad D = \frac{b}{a}$. Overall, the approach is similar to the proof of the lemma `eigen_belongs`, i.e. we exploit the tridiagonal structure of A_h. The proof required us to formalize some properties from combinatorics.

For the scheme that we are considering, $D = -2$. Two important steps that were required to complete the proof of $M_k \neq 0$ for the scheme $\mathcal{N}_h$ were:

1. Proving that $M_k = (-1)^k \times (k+1)$: We proved this using strong induction on k and the recurrence relation described above. To get an intuition of why it is true, we observe the values of M_k for initial values of k: $M_0 = 1, \quad M_1 = -2, \quad M_2 = 3, \quad M_3 = -4 \cdots M_k = (-1)^k \times (k+1)$
2. Proving that the determinant, $M_k \neq 0$

B.2 Lemmas on spectrum of E_h

In this subsection, we prove that the eigenvalues of E_h are just inverse of the eigenvalues of A_h, while the eigenvectors are the same. This follows from the following informal proof:

Proof. We start with the definition of Eigen-system (λ_m, s_m),

$$A_h s_m = \lambda_m s_m$$

Multiplying by A_h^{-1} on both sides and using the definition :$A_h^{-1}A_h = I$,

$$A_h^{-1}A_h s_m = A_h^{-1}\lambda_m s_m \implies s_m = \lambda_m A_h^{-1} s_m \implies \frac{s_m}{\lambda_m} = A_h^{-1} s_m$$

In Coq, we define the following lemma to formalize this proof:

```
Lemma inverse_eigen (m N:nat) (a b:R) : (2< N)%nat -> (0<=m<N)%nat ->
 0<a -> ((invertible N (Ah N a b a) (inverse_A N a b)) /\
 (LHS m N a b a= RHS m N a b a)) ->(Eigen_vec m N a b a) =
 Mmult (inverse_A N a b) (Mmult (Eigen_vec m N a b a) (Lambda m N a b a)).
```

B.3 Lemma to verify that the A_h^{-1} is normal

In this subsection, we verify that A_h^{-1} is normal. This lemma is stated as:

```
Lemma inverse_is_normal (a b:R): forall (N:nat),
 Mmult (inverse_A N a b ) (mat_transpose N (inverse_A N a b )) =
 Mmult (mat_transpose N (inverse_A N a b )) (inverse_A N a b ).
```

B.4 Intermediate lemmas to complete the proof of the lemma eigen_belongs:

This proof requires some intermediate lemmas which verify certain properties which are as follows:

Lemmas on structure of the matrix: In this subsection, we provide the lemmas that verify the structure of the matrix, i.e. the diagonal entries are b, the sub-diagonal entries are a and the super-diagonal entries are c. Mathematically, the lemmas say: $A_h(i,i) = b$, $A_h(i-1,i) = a$, $A_h(i,i+1) = c \quad \forall i = 1 \cdots N-2$. For the first and last rows, we have the structure as: $A_h(0,0) = b, A_h(0,1) = c, A_h(N-1,N-2) = a$ and $A_h(N-1,N-1) = b$. In Coq, we define the following lemmas to verify the above-mentioned structure:

```
Lemma coeff_prop_1 (a b c:R): forall (i N:nat), (2<N)%nat ->
  (0<i <N)%nat -> coeff_mat Hierarchy.zero (Ah N a b c) i (pred i) = a .

Lemma coeff_prop_2 (a b c:R): forall (i N:nat), (2<N)%nat ->
    (i <N)%nat ->coeff_mat Hierarchy.zero (Ah N a b c) i i = b.

Lemma coeff_prop_3 (a b c:R): forall (i N:nat), (2<N)%nat ->
  (i< pred N)%nat -> coeff_mat Hierarchy.zero (Ah N a b c) i (i + 1) = c.
```

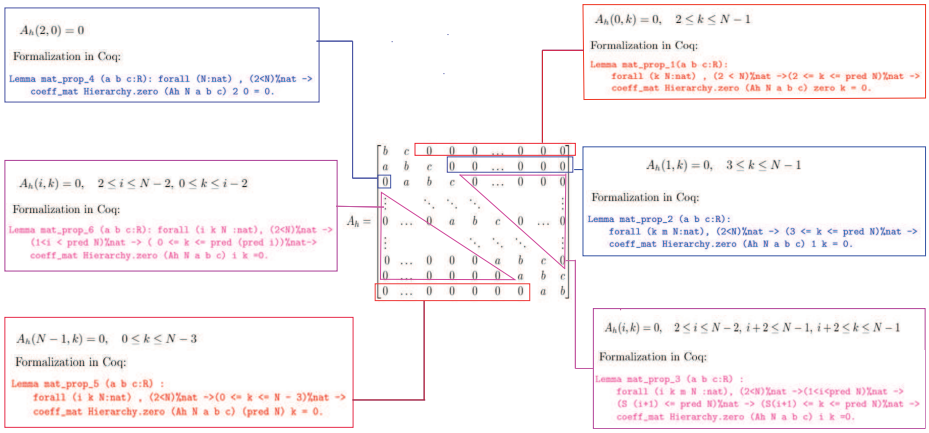

Fig. 1. Formalizing the tri-diagonal structure of the matrix. This formalization can be used for any tri-diagional system.

Lemmas to handle the zeros case: A good amount of effort was also required in extracting zero entries in the matrix A_h and proving that their sum equals zero. This again exploits the structure of the matrix, illustrated in figure (1). Two important lemmas that played a pivotal role in this proof are :

```
Lemma sum_const_zero:forall(n m:nat),(n<=m)%nat-> sum_n_m(fun _=>0)n m=0.
```

Mathematically this means: $\forall n, m : nat, (n \le m), \sum_n^m 0 = 0$

```
Lemma sum_n_m_zero(a:nat -> G)(n m:nat):(m<n)%nat -> sum_n_m a n m = zero.
```

Mathematically this means: $\forall (a : nat \to G), (n, m : nat),\ (m < n),\ \sum_n^m a = 0$, where, a is a function from naturals to an abelian group (G), in our case, it is reals. The first lemma was proved by us but the second one is already present in the `Coquelicot` library.

Lemmas to handle the non-zero case: The other part of the proof is to equate the sum of non-zero entries in LHS to a non-zero entry in RHS. i.e. $\vec{Ah}_i \cdot \vec{s_m} = \lambda_m s_{mi}$, where the Ah_i represents the i^{th} row of the matrix and s_{mi} denotes the i^{th} component of the Eigen-vector s_m and λ_m is a scalar. In Coq, considering $i = 0$, for example, would translate to the lemma statement:

```
Lemma i_0_j (a b c:R):
forall (m N:nat), (2<N)%nat -> (0<=m<N)%nat -> a=c /\ 0<c->
  mult (coeff_mat Hierarchy.zero (Ah N a b c ) zero 0)
    (coeff_mat Hierarchy.zero (Eigen_vec m N a b c ) 0 0) +
  mult (coeff_mat Hierarchy.zero (Ah N a b c ) zero 1)
    (coeff_mat Hierarchy.zero (Eigen_vec m N a b c ) 1 0) =
  mult (coeff_mat Hierarchy.zero (Eigen_vec m N a b c ) zero 0)
    (coeff_mat Hierarchy.zero (Lambda m N a b c ) 0 0).
```

We are not providing other lemmas here in the interest of space, but they can be referred in the attached code.

C Diagonalization of A_h

In this section, we discuss the lemmas required to prove that A_h is diagonalizable, i.e. $A_h = S\Lambda S^T$, where S is the matrix of eigenvectors and Λ is a diagonal matrix of Eigen-values of A_h. We first present an informal proof:

Proof. We start with the definition of an Eigensystem:

$$A_h S = S\Lambda \implies A_h S S^T = S\Lambda S^T \implies A_h = S\Lambda S^T \quad [SS^T = I]$$

Here, we use the fact that $S^{-1} = S^T$, since S is orthonormal. We verify this by using the definition of inverse of matrices, i.e. $SS^T = S^T S = I$. In Coq, we prove the following lemma:

```
Lemma Scond:forall (N:nat) (a b:R), (2<N)%nat -> 0<a ->
    Mmult (Sm N a b) (Stranspose N a b) = identity N /\
        Mmult (Stranspose N a b) (Sm N a b) = identity N.
```

To prove the lemma `Scond`, we split the proof into two sub-proofs:

1. i = j,
2. $i \neq j$

For the first case, we have the condition that $\vec{s_i} \cdot \vec{s_i} = 1$, i.e. $||\vec{s_i}||^2 = 1$. This reduces to proving that the sum of the following sine-squared series is 1.

$$\sum_{m=1}^{N} \frac{2}{N+1} \sin^2\left[j\frac{m\pi}{N+1}\right] = 1 \tag{22}$$

In Coq, we prove the following lemma to verify (22):

```
Lemma sin_sqr_sum: forall (i N:nat), (2<N)%nat /\ (0<=i<N)%nat ->
    sum_n_m (fun l:nat => (2/(INR(N+1)))*
        sin(((INR l+1)* INR (i+1)*PI)*/ INR (N+1)) ^2) 0 (pred N)=1.
```

Here, we make use of the following theorem from [24]:

Theorem 3. *If $a\ b \in \mathbb{R}$ and $d \neq 0$ and n is a positive integer,*
$\sum_{k=0}^{n-1} cos(a+kd) = \frac{\sin nd/2}{\sin d/2} cos\left(a + \frac{(n-1)d}{2}\right)$

where $\sin^2(\theta) = (1 - \cos(2\theta))/2$. We state the Theorem 3, using the following hypothesis statement in Coq:

```
Hypothesis cos_series_sum: forall (a d:R) (N:nat), d <>0->
    sum_n_m (fun l:nat => cos (a+(INR l)*d)) 0 (pred N)=
    sin(INR N*d/2)* cos(a+INR(N-1)*d/2)*/ sin(d/2).
```

We then use the hypothesis `cos_series_sum` to prove the lemma `sin_sqr_sum`. For the second case, we have the orthogonality condition $\vec{s}_i \cdot \vec{s}_j = 0,\ i \neq j$. This reduces to proving:

$$\sum_{k=0}^{N-1} \sin\left[(k+1)\frac{(i+1)\pi}{N+1}\right] \sin\left[(k+1)\frac{(j+1)\pi}{N+1}\right] = 0 \tag{23}$$

since, $\frac{2}{N+1}$ is a constant, it can be taken outside the summation.
Using the trigonometric identity,

$$\sin A \sin B = \frac{1}{2}\left[\cos\left(A-B\right) - \cos\left(A+B\right)\right]$$

we can reduce (23) into sums of cosines as follows:

$$\frac{1}{2}\sum_{k=0}^{N-1} \cos\left[(k+1)\frac{(i-j)\pi}{N+1}\right] - \frac{1}{2}\sum_{k=0}^{N-1} \cos\left[(k+1)\frac{(i+j+2)\pi}{N+1}\right] = 0 \tag{24}$$

Using Theorem (3), we can further reduce each sum in equation (24) into the product of sine and cosine. By doing some algebra, we prove that if $(i-j)$ and $(i+j+2)$ are simultaneously even or they are simultaneously odd, the sums in equation (24) cancel out. We further note that it is always the case that $(i-j)$ and $(i+j+2)$ are simultaneously even or they are simultaneously odd. We provide an informal proof of this fact as follows:

Proof. Case 1: $(i-j)$ is even:

$$\begin{aligned}
&\exists m : nat, (i-j) = 2m \\
&\implies i = 2m + j \\
&\implies i + j + 2 = 2m + j + j + 2 \\
&\implies i + j + 2 = 2 * (m + j + 1) \quad \therefore \text{ Even}
\end{aligned}$$

Case 2: $(i-j)$ is odd:

$$\begin{aligned}
&\exists m : nat, (i-j) = 2m + 1 \\
&\implies i = j + 2m + 1 \\
&\implies i + j + 2 = j + 2m + 1 + j + 2 \\
&\implies i + j + 2 = 2 * (j + m + 1) + 1 \quad \therefore \text{ Odd}
\end{aligned}$$

and vice-versa for each cases. This completes the proof of orthogonality of the Eigen vectors. In Coq, we prove the following lemma to verify (24):

```
Lemma cos_sqr_sum: forall (i j N:nat),
(2<N)%nat /\ (0<=i<N)%nat /\ (0<=j<N)%nat /\ (i<>j) ->
sum_n_m (fun l:nat => mult(/INR(N+1))
    (cos((INR(i) - INR(j)) * PI / INR (N + 1) +
        INR l * (INR(i) - INR(j)) * PI / INR (N + 1)) -
    cos(INR(i+j+2)*PI */ INR(N+1) +
        INR l * INR(i+j+2)*PI */ INR(N+1)))) 0 (pred N)=0.
```

D Application of the Lax–Equivalence theorem to the scheme $\mathcal{N}_h$

We define the following theorem statement to prove the convergence of the numerical scheme in Coq:

```
Theorem scheme_convergence: forall (U:X) (f:Y) (h:R) (uh: Xh h)
 (rh: forall (h:R), X -> (Xh h)) (sh: forall (h:R), Y->(Yh h)) (E: Y->X)
 (Eh:forall (h:R), (Yh h)->(Xh h)),
 is_linear_mapping X Y Aop ->  f=Aop U->
 (* Hypothesis that A is a linear mapping from X to Y*)
 (forall (h:R),is_linear_mapping (Xh h) (Yh h) (Ah_op h))->
 (* Hypothesis that Ah is a linear. mapping from Xh to Yh for each h*)
  (forall (h:R), is_bounded_linear X (Xh h) (rh h))->
  (* Hypothesis that rh is a bounded linear
  operatior (restriction) from X to Xh for each h*)
  (forall (h:R),is_bounded_linear Y (Yh h) (sh h))->
  (* Hypothesis that sh is a bounded linear
  operator (restriction) from Y to Yh*)
  is_bounded_linear Y X E ->
  (* Hypothesis that E is a bounded linear operator from Y to X*)
  U=E f->
  (* Defining solution in continuous space (true solution)*)
  (forall (h:R), is_bounded_linear (Yh h) (Xh h) (Eh h)) ->
  (* Hypothesis that Eh is a bounded linear operator from Yh to Xh for each h*)
  (forall h:R, is_finite (operator_norm(Eh h))) ->
   (* Hypothesis that ||Eh|| is finite*)
  (uh= Eh h (sh h f))->
   (* Defining a discrete solution uh*)
  ( Ah_op h uh = sh h f)-> (*f =fh*)
  (forall (h:R), rh h U= Eh h (Ah_op h (rh h U)))->
  (*uh =Eh *Ah *uh, where Eh*Ah=I*)
  (forall h:R, minus (Ah_op h (rh h U)) (sh h (Aop U)) <> Hierarchy.zero)->

  is_lim (fun h:R= norm (minus (rh h (E(f))) (Eh h (sh h (f))))) 0 0 .
    (*Convergence*)
```