The Fundamental Theorem of Numerical Analysis

by Alec Johnson

Presented February 1, 2006

1 Introduction.

The **Fundamental Theorem of Numerical Analysis (FTNA)** states that for a numerical method, *consistency plus stability implies convergence.* These terms are defined, and the statement is proved, per context. As an abstract statement, it seems to be a principle rather than a theorem. (Generalized versions of the theorem shift the work into demonstrating that the hypotheses are satisfied.)

This exposition will define these terms and explain why this theorem is true, for a range of contexts.

2 FTNA notions for generic and initial value problems.

A **problem** consists of data and an equation that must be solved for an unknown. An **initial value problem (IVP)** consists of initial data (and possibly boundary data) and a differential equation which determines the evolution of the solution over time. For initial value problems, the Fundamental Theorem of Numerical Analysis is known as the **Lax-Richtmyer theorem**.

A **numerical method** for a (continuum) problem is a discrete problem (more properly a family of discrete problems indexed by a parameter) whose solution is intended to approximate the solution of the problem.

A *numerical algorithm* is an algorithm for computing the solution of a numerical method. [1]

Solutions to differential equations are generally computed on a discretized domain called a **mesh**. Numerical methods for initial value problems compute a solution at a sequence of discrete points in time. We refer to computing the solution at a point in time based on a value at the previous point in time as applying a **time step** to the value.

To discuss whether a numerical solution approximates the solution of a problem, we need (1) a measure of the distance between a numerical solution and the exact solution to the problem, and (2) a method parameter which we can use to vary the numerical method. For differential equations this parameter is typically some measure of the **mesh size**. The finer the mesh, the greater the potential of the numerical method to accurately represent the exact solution.

A numerical method is said to **converge** to a solution if the distance between the numerical solution and the exact solution goes to zero as the method parameter approaches some limit (e.g. the mesh size goes to zero). Convergence is the desired property of a numerical method.

To have any hope of convergence, the *problem* itself must be stable, or **well-posed**. Perturbing the data of a problem produces a resulting perturbation in the solution of the problem. Assume that there is a measure defined on these perturbations. Refer to the ratio of the magnitude of the perturbation in the solution divided by the magnitude of the perturbation in the data as **the error growth factor**. If the error growth factor is bounded independent of the perturbation (sufficiently small) in the initial data, then we say the problem is *well-posed.* In particular, an initial value problem is *well-posed* (over a finite time interval) if the factor by which an initial error can grow is bounded.

To have any hope of convergence, the *numerical method* must also be **stable**. This means that the error growth factor is bounded independent of the method parameter or perturbation of the initial data. For a discretized initial value problem, stability means that the factor by which an initial error can grow is bounded independent of the mesh size (for any allowed mesh).

Stability is purely a property of the numerical method and is independent of the problem. Likewise, well-posedness is purely a property of the problem and is independent of the numerical method. To have any hope of establishing convergence, it is necessary to establish some kind of connection between the problem and the numerical method. This connection is called consistency.

Roughly speaking, a numerical method is said to be **consistent** with a problem if the exact solution to the problem approximately satisfies the discretized problem. This is *not* the same as saying that the exact solution to the problem approximately equals the exact solution to the discretized problem. For a differential equation, consistency means that a solution to the initial value problem approximately satisfies the discretized equation as the mesh size goes to zero. For an initial value problem, consistency means that the error committed by the numerical algorithm over a sin-

[1] Wikipedia: An algorithm is a finite set of instructions for accomplishing some task which, given an initial state, will terminate in a corresponding recognizable end-state.

gle time step is small. We will make the notion of consistency more precise below.

The beauty of consistency is that it is a local property, and hence easy to verify, whereas convergence is a global property.

The fundamental theorem of numerical analysis says that consistency plus stability implies convergence.

For an initial value problem, the fundamental theorem simply says that if the error committed on each time step is small enough, and if the rate of error growth is bounded, than the error in the solution will remain small. Intuitively this is obvious. The rest of this exposition attempts to make these ideas more precise for this case.

3 Initial Value Problems: The Lax-Richtmyer Theorem

Consider the initial value problem

$$(P) \quad \begin{cases} y' = Ly & 0 \le t \le T \\ y(0) = y_0 \end{cases}$$

and the associated family of numerical methods indexed by the number of time steps $N \in \mathbf{N}$ or by the size of each time step $k = T/N$,

$$\text{(M)} \quad \begin{cases} Y_{n+1} = L_k Y_n & 0 \le n \le N \\ Y_0 = y_0. \end{cases}$$

Let t_n denote the nth time point: $t_n = nk$. Let y_n denote the value of the exact solution at time t_n: $y_n = y(t_n)$.

We will assume that the method (M) is **stable**. This means that there is a bound B (independent of k) on the factor by which error can grow over the duration of the time interval T. If the operator L_k is linear, to demonstrate stability it is sufficient to show that $(\exists B < \infty)$ $(\forall k)$

$$\|L_k\| \ \le e^{Bk}.$$

(Equivalently $\|L_k\| \ \le 1 + Bk$.) For then $\|Y_N\|/\|Y_0\| \le \|L_k^N\| \le \|L_k\|^N \le e^{BkN} \le e^{BT} =: S$.

We will assume also that (P) and (M) are **consistent of order m**. This means that the error committed by the numerical algorithm over a single time step is small: $\|y_{n+1} - L_k y_n\| \le k^{m+1}C$ (for some $C < \infty$), where C is *independent* of n (i.e. time).

We will show that Y_N converges to y_N as $k \to 0$. Define the **local truncation error** d_n to be the difference between the value predicted by applying a time step to the exact solution and the value of the exact solution at the incremented time point t_{n+1}:

$$d_n = L_k y_n - y_{n+1}.$$

The **(accumulated) error** e_n is simply the difference between the exact solution and the numerical solution:

$$e_n = Y_n - y_n$$

If L_k is a linear operator, then the error at the incremented time point t_{n+1} equals the local truncation error (which is limited by consistency) plus the application of a time step to the accumulated error: $e_{n+1} = Y_{n+1} - y_{n+1} = L_k Y_n - y_{n+1} = (L_k Y_n - L_k y_n) + (L_k y_n - y_{n+1})$. That is:

$$e_{n+1} = L_k e_n + d_n.$$

This equation is the essense of the proof. It is a linear difference equation. The truncation error d_n is the forcing function. By linearity, the error introduced by the forcing function at each time step grows indepently. By stability, the growth in the error introduced by d_0 is bounded by $S < \infty$. By consistency, the error introduced by d_0 is bounded by Ck^{m+1}. So after all N time steps, the error introduced by d_0 is still bounded by SCk^{m+1}. For any $0 < n \le N$, the error introduced by d_n has less time to grow than the error introduced by d_0. Since there are N time steps, the total accumulated error e_N is therefore bounded by $SCNk^{m+1} = SC(T/k)k^{m+1} = (SCT)k^m$. So on the time interval $0 \le t \le T$ the error never exceeds $(SCT)k^m$. i.e. the global (truncation) error is of order k^m. This is what it means for the method to have convergence of order m.

Exercise: The above proof unsharply assumes that $Y_0 = y_0$. Modify it to work under the weaker assumption that $\|Y_0(k) - y_0\| \le k^m C$ (some C).

4 Extension to nonlinear operators.

The fundamental theorem of numerical analysis can be extended and applied to operators L which are nonlinear but sufficiently smooth by local linearization. An operator L (say a differential operator on a Banach space of functions) is smooth if it can be locally approximated by a linear operator DL, called its derivative:
$L(y + \Delta y) = L(y) + DL(\Delta y) + O(\|\Delta y\|^2)$.

In this case the error growth equation becomes:
$e_{n+1} = d_n + L_k(y_n + e_n) - L_k(y_n)$
$= d_n + (DL_k|_{y_n})(e_n) + (1/2)(D^2 L_k)|_{y_n + te_n})(e_n \otimes e_n)$.

5 Bibliography

- Richtmyer and Morton. Difference Methods for Initial Value Problems, second edition (©1967).
- R. LeVeque. Finite Volume Methods for Hyperbolic Problems (2002), §8.2 - §8.3.