4 Stability analysis of finite-difference methods for ODEs

4.1 Consistency, stability, and convergence of a numerical method; Main Theorem

In this Lecture and also in Lecture 8, we will need a notation for the *norm*. For any sequence of numbers $\{a_n\}$, let

$$||a||_\infty = \max_n |a_n| \,. \tag{4.1}$$

This norm is called the "L_∞-norm" of the sequence $\{a_n\}$. There exist other kinds of norms, each of which is useful in its own circumstances. In this course, we will only deal with the L_∞-norm, and therefore we will simply denote it by $||a||$, dropping the subscript "∞". The reason we are interested in this particular norm is that at the end of the day, we want to know that the *maximum* error of our numerical solution is bounded by some tolerance $\varepsilon_{\rm tol}$:

$$\max_n |\epsilon_n| \le \varepsilon_{\rm tol}, \qquad \text{or} \qquad ||\epsilon|| \le \varepsilon_{\rm tol} \,. \tag{4.2}$$

We now return to the main subject of this section.

Recall that the main goal of a numerical method when solving an IVP

$$y' = f(x,y), \qquad y(x_0) = y_0. \tag{4.3}$$

is to assure that the numerical solution Y_n closely matches the analytical solution $y(x_n)$. [15] In other words, the global error of the numerical solution must be acceptably small. Ideally, one also hopes that as one decreases the step size, the numerical solution approaches the analytical solution closer and closer. Consequently, one makes the following definition.

Definition 1: A numerical method is called *convergent* if its global error computed up to a given x satisfies:

$$\lim_{h\to 0} ||\epsilon|| \;\equiv\; \lim_{h\to 0} \|y - Y\| \;=\; \lim_{h\to 0} \max_n |y_n - Y_n| = 0 \,. \tag{4.4}$$

Note that when taking the limit $h \to 0$, the length $(x - x_0)$ of the computational interval is taken as fixed.

Let us note, in passing, that Definition 1 tacitly implies that the numerical solution Y_n is computed with no round-off (i.e., machine) error. Indeed, if a round-off error is present, the global error may increase, instead of decrease, as h gets too small: see the figure in Sec. 1.3 of Lecture 1.

Below we will show that **for a numerical method to be convergent, it needs to satisfy *two* conditions**. One of these conditions we know already:

- The numerical scheme must match the original ODE closer and closer as $h \to 0$.

Let us express this fact in a more formal way using the simple Euler method as an example. The numerical solution of (4.3) by that method satisfies:

$$\frac{Y_{n+1} - Y_n}{h} - f(x_n, Y_n) = 0 \,. \tag{4.5}$$

Denote the l.h.s. of (4.5) as $F[Y_n, h]$; then that equation can be rewritten as

$$F[Y_n, h] = 0 \,. \tag{4.6}$$

[15] In this and in all subsequent Lectures, we abandon the subscript i in favor of n, because we want to reserve i for the $\sqrt{-1}$.

(In general, any numerical method can be written in the form (4.6).) When one substitutes into (4.6) the exact solution $y(x_n)$ of the ODE, one obtains:

$$F[y_n, h] = \tau_n \,. \tag{4.7}$$

In Lecture 1, we called τ_n the discretization error and $h\tau_n$, the local truncation error. Recall that the local truncation error shows how close the numerical and exact solutions are after one step, provided that they start at the same initial value. On the other hand, the discretization error τ_n shows how closely the exact solution satisfies the numerical scheme. Equivalently, it shows how closely the numerical scheme matches the original differential equation (4.3). This motivates our next definition.

Definition 2: A numerical method $F[Y_n, h] = 0$ is called *consistent* if

$$\lim_{h\to 0} ||\tau|| = 0, \tag{4.8}$$

where τ_n is defined by Eq. (4.7). According to the interpretation of the discretization error stated after that equation, any consistent numerical method closely matches the original differential equation when the step size h is sufficiently small. Note that any method of order $l > 0$ is consistent because $\tau_n = O(h^l)$.

To motivate the *second condition* (of the two mentioned after Definition 1), we pose a **question:** What should one require of a consistent method in order for it to be convergent? We will answer it in Theorem 4.1 below, but let us first explain why a consistent method may fail to converge. Consider a numerical scheme of the form (4.6). Let its *ideal* solution, computed *without* the round-off error, be Y_n. Let its *actual* solution, computed *with* a round-off error, be U_n. That is, U_n satisfies

$$F[U_n, h] = \xi_n \,, \tag{4.9}$$

where ξ_n is a small number that arises due to the round-off error. Since the round-off error is small, then at early stages of the computation, Y_n and U_n are close to one another. Intuitively, we expect Y_n and U_n to remain close throughout the computation.[16] However, due to certain behavior of the numerical method, these initially close solutions may eventually move far apart. (*How* this can occur will be explained starting in Sec. 4.3; for now, just accept on faith that this *can* indeed occur.) Such divergence between the two numerical solutions is intuitively unsatisfactory. Indeed, one cannot guarantee the absence of tiny perturbations during calculation on a computer, and yet one desires that such perturbations would not affect the numerical solution in any significant way. These considerations motivate yet another definition.

Definition 3: Consider an IVP

$$y' = \lambda y, \quad \mathrm{Re}\lambda < 0; \qquad y(x_0) = y_0. \tag{4.10}$$

Let Y_n and U_n be its numerical solutions defined as in the previous paragraph. The numerical method is called *stable* if

$$||U - Y|| \le C||\xi|| \,, \tag{4.11}$$

where the constant C may depend on x (the length of the computational interval) but is required to be independent of h. That is, for a stable method, the deviation between two numerical solutions arising, e.g., due to the round-off error, does not grow with the number of steps.

[16] Otherwise, we would have extreme sensitivity of our computation to an unphysical cause, such as the round-off error, and such a sensitive computation without a physical reason cannot describe the "reality."

Definition 3 imposes an important restriction on the class of equations to which it can be applied. We will discuss this in detail in the next section. In the remainder of this section, we will state, and outline the proof of, the main theorem of numerical analysis.

Theorem 4.1 (P. Lax):
If for an IVP (4.10) a method is both consistent and stable, then it converges. In short:

$$\boxed{\text{Consistency} \;+\; \text{Stability} \;\Rightarrow\; \text{Convergence}}$$

Remark: Note that all three properties of the method: consistency, stability, and convergence, must be defined with respect to the same norm (in this course, we are using only one kind of norm, so that is not an issue anyway).

The idea of the Proof: Consistency of the method means that the local truncation error at each step, $h\tau_n$, is sufficiently small so that the accumulated (i.e., global) error, which is on the order of τ_n, tends to zero as h is decreased (see (4.8)). Thus:

$$\text{Consistency} \quad \Rightarrow \quad \|y - Y\| \quad \text{is small,} \tag{4.12}$$

where, as above, Y is the *ideal* solution of the numerical scheme (4.6) obtained in the absence of machine round-off errors and any errors in initial conditions.

Stability of the method means that if at any given step, the actual solution U_n slightly deviates from the ideal solution Y_n due to the round-off error, then this deviation remains small and does not grow as n increases. Thus:

$$\text{Stability} \quad \Rightarrow \quad \|Y - U\| \quad \text{is small.} \tag{4.13}$$

Equations (4.12) and (4.13) together imply that the maximum difference between the actual computed solution u_n and the exact solution y_n also remains small. Indeed:

$$\|y - U\| = \|(y - Y) + (Y - U)\| \le \|(y - Y)\| + \|(Y - U)\|, \tag{4.14}$$

which must be small because each term on the r.h.s. is small. The fact that the l.h.s. of the above equation is small means, by Definition 1, that the method is convergent. ***q.e.d.***

4.2 Setup for the stability analysis: the model equation

Let us first explain why we needed to restrict Definition 3 to IVPs (4.10). Suppose for a moment that in that equation, one takes $\lambda > 0$ instead of $\mathrm{Re}\lambda < 0$. Then any two analytical solutions of such an equation that initially differ by a small amount δ, will eventually be separated by $\delta \exp[\lambda x]$, which may no longer be small for a sufficiently large x. Consequently, one cannot expect that any two numerical solutions, which are supposed to follow the behavior of the analytical ones, will stay close as required by (4.11). On the other hand, for $\lambda < 0$, the "distance" between two analytical solutions of (4.10) decreases, and the same behavior is expected from its numerical solutions. Hence the requirement "$\mathrm{Re}\lambda < 0$" in (4.10).

The case $\mathrm{Re}\lambda = 0$ will be addressed separately in Lecture 5.

In the remainder of this Lecture and also in Lecture 5, we will study stability of numerical methods applied to the **model equation**

$$y' = \lambda y, \tag{4.15}$$

which differs from (4.10) only in that we have dropped the restriction on λ.

We will find that some of the methods are stable as per (4.11) even for $\mathrm{Re}\lambda > 0$ in (4.15). Let us note that this is not necessarily a good feature of the method; we will discuss this in more detail in Lecture 5. **A good method must have only one property: be convergent. This depends not only on the method itself, but also on the equation to which it is applied**. Again, more detail on this will follow in Lecture 5.

Let us now explain why the simple linear equation (4.15) is relevant for predicting stability of solutions of the ODE in (4.3) with a general, i.e., possibly nonlinear, function $f(x, y)$. Recall that we defined stability in regards to the deviation between two initially close numerical solutions. Since numerical solutions are supposed to match analytical ones, we may as well discuss stability of the latter. So, consider two analytical solutions $y(x)$ and $u(x)$ that are close to each other. Their difference satisfies:

$$(y-u)' \;=\; f(x,y) - f(x,u) \;\approx\; f_y(x,y)\,(y-u)\,. \tag{4.16}$$

Locally, i.e. near any given "point" (x, y), the coefficient f_y can be approximated by a constant, and then the equation for $(y - u)$ takes on the form (4.15). In other words, the model problem (4.15) is the local linear approximation (also known as a linearization) of (4.3). One performs stability analysis for the linear model problem (4.15) rather than for the ODE in Eq. (4.3) because the former equation is simpler.

It is worth restating the main point of the previous paragraph while making a slightly different emphasis: **Whether the solution obtained by a given numerical method will be stable or not, in the sense of condition** (4.11)**, depends on the problem which it is applied to.** Specifically, to decide whether the method at hand is going to be stable for a given IVP (4.3), one should first perform a linearization of the ODE as in (4.16). Then the range of values of λ in the model problem (4.15) is just the range of values of f_y in (4.16). Depending on the specific value of λ, the numerical method applied to (4.15) may be stable or unstable. We will discuss this further in Lecture 5.

Remark 1: Note that if in (4.3), $f(x, y) = a(x)y$, i.e. if the original ODE is *linear*, then the model equation (4.15) coincides with it. In such a case, the difference between any two of its solutions satisfies exactly the same ODE as the solution itself. We will reference this Remark in Lecture 5 and later in Lecture 12.

Remark 2: Finally, let us note that the (global) *numerical error* ϵ_n, even if it is small, does *not* satisfy the linear model equation (4.15). This is because an equation for the evolution of the global error is similar to (4.16) but has an additional driving term on the r.h.s., which arises from the local truncation error at each step:

$$\epsilon'(x) \;\text{ “}=\text{ ”}\; f_y(x,y)\cdot\epsilon(x) \;+\; \left\{ \begin{array}{c} \text{driving terms due to} \\ \text{local truncation error} \end{array} \right\}. \tag{4.17}$$

(Here the equation sign is taken in quotes because we have approximated the evolution of the discrete quantity ϵ_n with that of the continuous one, $\epsilon(x)$.) If the numerical method is unstable, both the solution of the model problem (4.15) and the numerical error will increase as the computation proceeds. However, for a stable numerical method, one can encounter situations where the numerical solution of (4.10) decays (as by definition it does for a stable method), but the global error tends to a constant. You will see such an example in HW 5.

4.3 Stability analyses of some familiar numerical methods

Below we present stability criteria for the numerical methods we have studied in the preceding Lectures for the model problem (4.15).

We begin with the simple Euler method. Applying scheme (4.5) to (4.15), we obtain:

$$Y_{n+1} = Y_n + \lambda h Y_n\,, \qquad \Rightarrow \qquad Y_n = Y_0(1+\lambda h)^n\,. \tag{4.18}$$

To get a better idea what can "go wrong" with this scheme, let us assume for the moment that $\lambda < 0$. (Recall that more generally, in Definition 3, we require $\mathrm{Re}\lambda < 0$.) For $\lambda < 0$, the true solution $y_0 e^{\lambda x}$ of (4.10) decreases, but the numerical solution (4.18) will decrease only if

$$|1+h\lambda| < 1 \quad \Rightarrow \quad -1 < 1+h\lambda < 1 \quad \Rightarrow \quad h < \frac{2}{|\lambda|}\,. \tag{4.19}$$

E.g., to solve $y' = -30y$ (with any initial condition), we must use $h < 2/30$ in order to guarantee that the round-off and truncation errors will decay.

Thus, for the model problem (4.15) with $\lambda < 0$, the simple Euler method is stable only when the step size satisfies Eq. (4.19). This conditional stability is referred to as *partial* stability; thus, the simple Euler method is partially stable.

For the general case where λ is a complex number, partial stability is defined as below.

Definition 4: A method is called *partially stable* if, when applied to the model problem (4.15) *with* $\mathrm{Re}\lambda < 0$, the corresponding numerical solution is stable only for *some* values of λh. The region in the λh-plane where the method is stable is called the *region of stability* of the method.

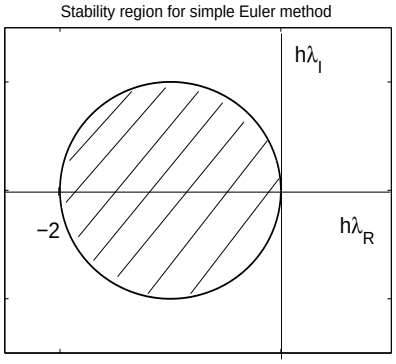
Let us find the region of stability of the simple Euler method. To this end, write λ as the sum of its real and imaginary parts: $\lambda = \lambda_R + i\lambda_I$ (note that here and below $i = \sqrt{-1}$). Then the first of the inequalities in (4.19) becomes

$$\begin{aligned} |1 + h\lambda_R + ih\lambda_I| \,&<\, 1 \qquad \Rightarrow \\ \sqrt{(1+h\lambda_R)^2 + (h\lambda_I)^2} \,&<\, 1\,. \end{aligned} \tag{4.20}$$

Thus, the region of stability of the simple Euler method is the inside of the circle

$$(1+h\lambda_R)^2 + (h\lambda_I)^2 = 1,$$

as shown in the figure on the right.



We now present brief details about the region of stability for the Modified Euler method. In a homework problem, you will be asked to supply the missing details.

Substituting the ODE from (4.15) into Eqs. (1.22) (see Lecture 1), we find that

$$Y_{n+1} = \left(1 + h\lambda + \frac{1}{2}(h\lambda)^2\right) Y_n\,. \tag{4.21}$$

Remark 3: Note that the factor on the r.h.s. of (4.21) is quite expected: Since the Modified Euler is the 2nd-order method, it makes sense that its solution of the model problem (4.15) turned out to be the 2nd-degree polynomial that approximates the exponential in the exact solution $y_{n+1} = y_n e^{\lambda h}$.

Remark 4: The factor relating solutions Y_n and Y_{n+1} at two consecutive steps is called the *amplification factor* of the scheme. We will denote it by ρ. Thus, for methods like simple and Modified Euler, and for RK methods of Lecture 2 in general, one has:

$$Y_{n+1} = \rho\big(h\lambda\big)\, Y_n\,. \tag{4.21$'$}$$

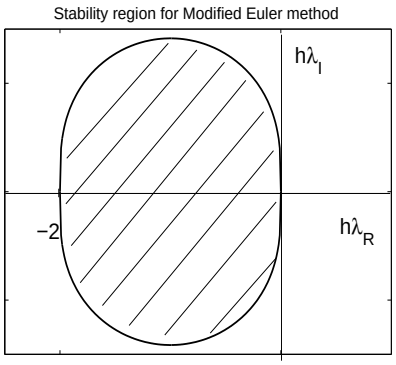
For example, for (4.21), $\rho = \big(1 + h\lambda + \frac{1}{2}(h\lambda)^2\big)$.

Remark 5: Returning to the point made in Remark 3, we note that the amplification factor of a RK method of order m does not have to be exactly equal to the m-order Taylor polynomial of $\exp[\lambda h]$. Rather, it has to coincide with such a polynomial up to terms of order $O\big((h\lambda)^{m+1}\big)$. A more general formula for $\rho\big(h\lambda\big)$ for RK methods is derived in Appendix 1.

The boundary of the stability region is obtained by setting the modulus of the factor on the r.h.s. of (4.21) to 1:

$$\left|1 + h\lambda + \frac{1}{2}(h\lambda)^2)\right| = 1.$$

Indeed, if the factor on the l.h.s. is less than 1, all errors will decay, and if it is greater than 1, they will grow, even though the exact solution may decay.



The above equation can be equivalently written as

$$\left(1 + h\lambda_R + \frac{1}{2}\left(\,(h\lambda_R)^2 - (h\lambda_I)^2\,\right)\right)^2 + (h\lambda_I + h^2\lambda_I\lambda_R)^2 = 1\,. \tag{4.22}$$

The corresponding region is shown above.

When the cRK method is applied to the model problem (4.15), the corresponding stability criterion becomes

$$\left|\sum_{k=0}^{4} \frac{(h\lambda)^k}{k!}\right| \le 1\,. \tag{4.23}$$

The expression on the l.h.s. is the fourth-degree polynomial approximating $e^{\lambda h}$; this is consistent with Remarks 3 and 5 above.

For real λ, criterion (4.23) reduces to

$$-2.79 \le h\lambda \le 0\,. \tag{4.24}$$

Note that the cRK method is not only more accurate than the simple and Modified Euler methods, but also has a greater stability region for negative real values of λ.

4.4 Stability analysis of multistep methods

We begin with the 2nd-order Adams–Bashforth method (3.5):

$$Y_{n+1} = Y_n + h\left(\frac{3}{2}f_n - \frac{1}{2}f_{n-1}\right)\,. \tag{3.5}$$

Substituting the model ODE (4.15) into that equation, one obtains

$$Y_{n+1} - \left(1 + \frac{3}{2}\lambda h\right) Y_n + \frac{1}{2}\lambda h Y_{n-1} = 0\,. \tag{4.25}$$

To solve this difference equation, we use the same procedure as we would use to solve a linear ODE. Namely, for the ODE

$$y'' + a_1 y' + a_0 y = 0$$

with constant coefficients a_1, a_0, we need to substitute the *ansatz*[17] $y = e^{rx}$, which yields the following polynomial equation for r:

$$r^2 + a_1 r + a_0 = 0\,.$$

Similarly, for the difference equation (4.25), we substitute $Y_n = \rho^n$. Here ρ has the same meaning of the amplification factor as defined in Remark 4 in Section 4.3. Upon this substitution and subsequent cancellation of all terms in (4.25) by the common factor ρ^{n-1}, one obtains:

$$\rho^2 - \left(1 + \frac{3}{2}\lambda h\right) \rho + \frac{1}{2}\lambda h = 0\,. \tag{4.26}$$

This quadratic equation has two roots:

$$\begin{aligned} \rho_1 &= \frac{1}{2}\left\{ \left(1 + \frac{3}{2}\lambda h\right) + \sqrt{\left(1 + \frac{3}{2}\lambda h\right)^2 - 2\lambda h} \right\}, \\ \rho_2 &= \frac{1}{2}\left\{ \left(1 + \frac{3}{2}\lambda h\right) - \sqrt{\left(1 + \frac{3}{2}\lambda h\right)^2 - 2\lambda h} \right\}. \end{aligned} \tag{4.27}$$

In the limit of $h \to 0$ (which is the limit where the difference method (4.25) reduces to the ODE (4.15)), one can use the Taylor expansion (and, in particular, the formula $\sqrt{1+\alpha} = 1 + \frac{1}{2}\alpha + O(\alpha^2)$), to obtain the asymptotic forms of ρ_1 and ρ_2:

$$\rho_1 \approx 1 + \lambda h, \qquad \rho_2 \approx \frac{1}{2}\lambda h\,. \tag{4.28}$$

The solution Y_n that corresponds to root ρ_1 turns, in the limit $h \to 0$, into the true solution of the ODE $y' = \lambda y$, because

$$\lim_{h\to 0} \rho_1^n \;=\; \lim_{h\to 0} (1 + \lambda h)^n \;=\; \lim_{h\to 0} (1 + \lambda h)^{x/h} \;=\; e^{\lambda x}\,; \tag{4.29}$$

see Sec. 0.5. However, the solution of the difference method (4.25) corresponding to root ρ_2 *does not correspond to any actual solution of the ODE* ! For that reason, root ρ_2 and the corresponding difference solution ρ_2^n are called *parasitic*.

The role of a parasitic root is that it gives rise to a contribution that contaminates the numerical solution in a way different than does the local truncation error, discussed in Lecture 1 and later. Namely, by the *linear superposition principle*, which applies to linear difference equations just like it does to linear differential equations, one can write the *general* solution of Eq. (4.25) as:

$$Y_n = c_1 \rho_1^n + c_2 \rho_2^n, \tag{4.30}$$

[17] This is a German word meaning, approximately, a template.

with some constants c_1 and c_2 (which depend on Y_0 and Y_1). In order for the numerical solution to remain close to the true solution (4.29), the parasitic-solution ρ_2-term must remain small compared to the true-solution ρ_1-term for all n.

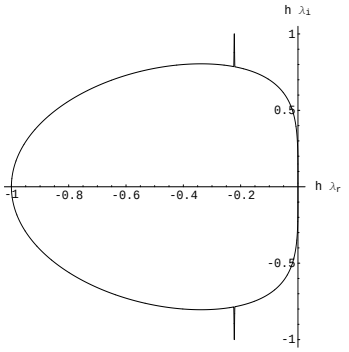
A good thing about the parasitic solution for the 2nd-order Adams–Bashforth method is that it does *not* grow for sufficiently small λh. In fact, since for sufficiently small h, $\rho_2 \approx \frac{1}{2}\lambda h < 1$, then that parasitic solution decays to zero rather rapidly and therefore does not contaminate the numerical solution.

To require that the 2nd-order Adams–Bashforth method be stable is equivalent to requiring that *both* ρ_1 and ρ_2 satisfy

$$|\rho_1| \le 1 \quad \underline{\text{and}} \quad |\rho_2| \le 1\,. \tag{4.31}$$

The stability region is inside the oval-shaped region shown on the right (the little "horns" are the plotting artifice). This figure is produced by *Mathematica*; in a homework problem, you will be asked to obtain this figure on your own.
A curious point to note is that the two requirements, $|\rho_1| \le 1$ and $|\rho_2| \le 1$, produce two non-overlapping parts of the stability region boundary (its right-hand and left-hand parts, respectively).

Stability region of 2nd-order Adams–Bashforth

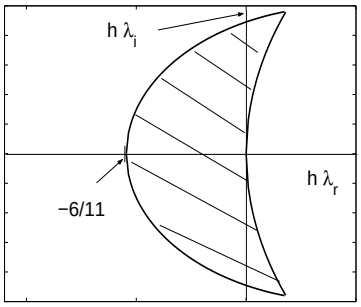


A similar analysis for the 3rd-order Adams–Bashforth method (3.11) shows that the corresponding difference equation has three roots, of which one (say, ρ_1) corresponds to the true solution of the ODE and the other two (ρ_2 and ρ_3) are the parasitic roots. Fortunately, these roots decay to zero as $O(h)$ for $h \to 0$, so they do not affect the numerical solution for sufficiently small h. For finite h, the requirement

$$|\rho_1| \le 1, \quad |\rho_2| \le 1, \quad |\rho_3| \le 1$$

results in the stability region whose shape is qualitatively shown on the right.

Stability region of 3rd-order Adams-Bashforth



From the above consideration of the 2nd- and 3rd-order Adams–Bashforth methods there follows an observation that is shared by some other families of methods: *the more accurate method has a smaller stability region.*

Let us now analyze the stability of the two methods considered in Sec. 3.3.

Leap-frog method

Substituting the model ODE into Eq. (3.20), one obtains

$$Y_{n+1} - Y_{n-1} = 2h\lambda Y_n\,. \tag{4.32}$$

For $Y_n = \rho^n$ we find:

$$\rho^2 - 2h\lambda\rho - 1 = 0 \quad \Rightarrow \quad \rho_{1,2} = h\lambda \pm \sqrt{1 + (h\lambda)^2}\,. \tag{4.33}$$

Considering the limit $h \to 0$, as before, we find:

$$\rho_1 \approx 1 + h\lambda, \qquad \rho_2 \approx -1 + h\lambda\,. \tag{4.34}$$

Again, as before, the solution of the difference equation with ρ_1 corresponds to the solution of the ODE: $\rho_1^n \approx (1 + h\lambda)^{x/h} \approx e^{\lambda x}$. The solution corresponding to root ρ_2 is parasitic. The general solution is a linear superposition of the true and parasitic solutions:

$$Y_n = c_1\rho_1^n + c_2\rho_2^n \approx c_1 e^{\lambda x} + c_2(-1)^n e^{-\lambda x}\,, \tag{4.35}$$

where constants c_1 and c_2 depend on the two initial points, Y_0 and Y_1. For a Y_1 obtained by a sufficiently accurate method, these constants are such that initially, the parasitic solution is much smaller than the true one: $|c_2| \ll |c_1|$, with $c_1 \approx Y_0$. As x increases, the relative size of the true and parasitic solutions may change. This depends on the sign of λ, as described below.

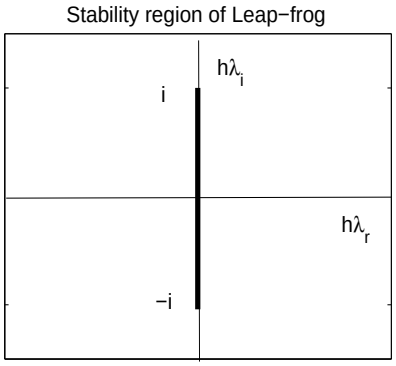
$\underline{\lambda > 0}$ Then $\rho_2 = -(1 - h\lambda)$, so that $|\rho_2| < 1$, and the parasitic solution decays, whereas the true solution, $(1 + h\lambda)^n \approx e^{\lambda x}$, grows. Thus, the method truthfully reproduces the actual solution of the differential equation. (Note that even though the numerical solution grows, one can*not* call the numerical method unstable, because by Definition 3 in Sec. 4.1, a method can be classified as stable or unstable only for $\lambda < 0$.)

$\underline{\lambda < 0}$ Then $\rho_2 = -(1 + h|\lambda|)$, so that $|\rho_2| > 1$. We see that in (4.35) the parasitic part (the second term) of the solution grows, whereas the true solution (the first term) decays; thus, for a sufficiently large x, the numerical solution will bear no resemblance to the true solution.

The stability region of the Leap-frog method, shown on the right, is disappointingly small: the method is stable only for

$$\lambda_R = 0 \quad \underline{\text{and}} \quad -1 \le h\lambda_I \le 1\,. \tag{4.36}$$

Moreover, since this region consists just of its own boundary, where $|\rho_1| = |\rho_2| = 1$, then the numerical error will *not decay but will maintain (approximately) constant magnitude* over time.



Note, however, that this occurs where $\lambda_R = 0$ and hence $|\exp[\lambda x]| \equiv |\exp[i\lambda_I x]| = 1$, i.e. where the true solution also does not decay. We will see in Lecture 5 that this coincidence of the behaviors of the numerical error and the true solution may be beneficial for certain problems.

For problems with $\lambda < 0$, the Leap-frog method is unstable. However, this numerical instability is weak, because the parasitic error $c_2(-1 + h\lambda)^n \approx c_2(-1)^n \exp[|\lambda|x]$ will require $|\lambda|x > O(1)$ to overtake the true solution $c_1(1 + h\lambda)^n \approx c_1 \exp[-|\lambda|x]$, given that $|c_2| \ll |c_1|$ (see the text after (4.35)). Thus, since the numerical solution will stay close to the true solution for $|\lambda x| \le O(1)$, the Leap-frog method is called *weakly* unstable.

Divergent 3rd-order method (3.22)

For the model problem (4.15), that method becomes

$$Y_{n+1} + \frac{3}{2}Y_n - 3Y_{n-1} + \frac{1}{2}Y_{n-2} = 3h\lambda Y_n \,. \tag{4.37}$$

Proceeding as before, we obtain the characteristic equation for the roots:

$$\rho^3 + \left(\frac{3}{2} - 3h\lambda\right)\rho^2 - 3\rho + \frac{1}{2} = 0 \,. \tag{4.38}$$

To consider the limit $h \to 0$, we can simply set $h = 0$ as the lowest-order approximation. The cubic equation (4.38) reduces to

$$\rho^3 + \frac{3}{2}\rho^2 - 3\rho + \frac{1}{2} = 0 \,, \tag{4.39}$$

which has the roots

$$\rho_1 = 1, \qquad \rho_2 \approx -2.69, \qquad \rho_3 \approx 0.19 \,. \tag{4.40}$$

Then for small h, the numerical solution is

$$\begin{array}{lccccc} Y_n = & c_1(1+h\lambda)^n & + & c_2(-2.69+O(h))^n & + & c_3(0.19+O(h))^n \\ & \text{(approximate true solution)} & & \text{(parasitic solution that explodes)} & & \text{(parasitic solution that decays)} \end{array} \tag{4.41}$$

The second term, corresponding to a parasitic solution, grows (in magnitude) much faster than the term approximating the true solution, and therefore the numerical solution very quickly becomes complete garbage. This happens much faster than for the Leap-frog method with $\lambda < 0$. Therefore, method (3.22) is called *strongly* unstable; obviously, it is useless for any computations.

The above considerations of multistep methods can be summarized as follows. Consider a multistep method of the general form (3.17). For the model problem (4.15), it becomes

$$Y_{n+1} - \sum_{k=0}^{M} a_k Y_{n-k} \;=\; h\sum_{k=0}^{N} b_k \lambda Y_{n-k} \,. \tag{4.42}$$

The first step of its stability analysis is to set $h = 0$, which will result in the following characteristic polynomial:

$$\rho^{M+1} - \sum_{k=0}^{M} a_k \rho^{M-k} = 0 \,. \tag{4.43}$$

This equation must always have a root $\rho_1 = 1$ which corresponds to the true solution of the ODE. If any of the other roots, i.e. $\{\rho_2, \rho_3, \ldots, \rho_{M+1}\}$, satisfies $|\rho_k| > 1$, then the method is strongly unstable. If any root with $k \geq 2$ satisfies $|\rho_k| = 1$, then the method may be weakly unstable (like the Leap-frog method). Finally, if all $|\rho_k| < 1$ for $k = 2, \ldots, M+1$, then the method is stable for $h \to 0$. It may be either partially stable, as the single-step methods and Adams–Bashforth methods,[18] or absolutely stable, as the implicit methods, that we will consider next.

An alternative form of the stability analysis of multistep methods is found in Appendix 2.

[18] You will be asked to explain why this is so in one of the QSAs.

4.5 Stability analysis of implicit methods

Consider the implicit Euler method (3.41). For the model problem (4.15) it yields

$$Y_{n+1} = Y_n + h\lambda Y_{n+1}, \qquad \Rightarrow \qquad Y_n = Y_0 \left(\frac{1}{1-h\lambda}\right)^n . \tag{4.44}$$

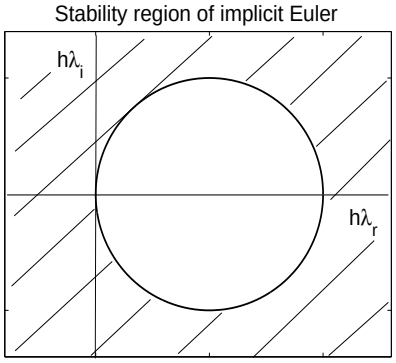
It can be verified (do it, following the lines of Sec. 0.5) that the r.h.s. of the last equation reduces for $h \to 0$ to the exact solution $y_0 e^{\lambda x}$, as it should. The stability condition is

$$\left|\frac{1}{1-h\lambda}\right| \le 1 \qquad \Rightarrow \qquad |1-h\lambda| \ge 1 . \tag{4.45}$$

The boundary of the stability region is the circle

$$(1-h\lambda_R)^2 + (h\lambda_I)^2 = 1 ; \tag{4.46}$$

the stability region is the *outside* of that circle (see the second of inequalities (4.45)).



Definition 5: If a numerical method, when applied to the model problem (4.15), is stable for all λ with $\lambda_R < 0$, such a method is called absolutely stable, or A-stable for short.

Thus, we have shown that the implicit Euler method is A-stable.

Similarly, one can show that the Modified implicit Euler method (3.44) is also A-stable (you will be asked to do so in a homework problem).

Theorem 4.2:

1) No explicit finite-difference method is A-stable.
2) No implicit method of order higher than 2 is A-stable.

Thus, according to Theorem 4.2, implicit methods of order 3 and higher are only partially stable; however, their regions of stability are usually *larger* than those of explicit methods of the same order.

An old but classic reference on stability of numerical methods is the book by P. Henrici, "Discrete variable methods in ordinary differential equations," (Wiley, 1968).

4.6 Appendix 1: Amplification factor of Runge–Kutta methods

As we noted in Sec. 2.3, RK methods can be (and in the professional literature are) described by Butcher tableau. In its notations, the first equation in (2.5) can be written in the form:

$$Y_{n+1} = Y_n + \mathbf{b}^T \mathbf{k}, \tag{4.47}$$

where $\mathbf{k} \equiv [k_1, k_2, \ldots, k_s]^T$ is the column vector, whose components are the stages, defined after (2.5). For a general function $f(x,y)$ on the r.h.s. of the ODE, one cannot write an

equation for the stages in a form more compact than done in (2.5). However, for the model equation (4.15), it *is* possible to write a compact equation for $\mathbf{k}$ using matrix $\mathbf{A}$ from the Butcher tableau.

To see this, let us first write down this equation for the specific form of (2.5) and then generalize. Let us denote

$$z \equiv h\lambda\,. \tag{4.48}$$

Given that $f(x,y) = \lambda y$, the lines in (2.5) starting with the second one become:

$$\begin{array}{l} k_1 = h\lambda\, Y_n \\ k_2 = h\lambda\,(Y_n + a_{21}k_1) \\ k_3 = h\lambda\,(Y_n + a_{31}k_1 + a_{32}k_2) \end{array} \qquad \Rightarrow \qquad \mathbf{k} = zY_n\hat{\mathbf{1}} + z\mathbf{A}\,\mathbf{k} \qquad \Rightarrow \qquad \mathbf{k} = zY_n\,(I - z\mathbf{A})^{-1}\hat{\mathbf{1}}, \tag{4.49}$$

where we have defined the column vector $\hat{\mathbf{1}} \equiv [1,\,1,\ldots,\,1]^T$, and I is the identity matrix of the same dimension as $\mathbf{A}$. Combining the last equation with (4.47) one arrives at:

$$Y_{n+1} = \left(1 + z\mathbf{b}^T\,(I - z\mathbf{A})^{-1}\hat{\mathbf{1}}\,\right) Y_n\,. \tag{4.50}$$

Thus, the amplification factor, defined in (4.21′), is:

$$\rho(z) = \left(1 + z\mathbf{b}^T\,(I - z\mathbf{A})^{-1}\hat{\mathbf{1}}\,\right), \tag{4.51}$$

where, again, matrix $\mathbf{A}$ and vector $\mathbf{b}$ of the RK coefficients are defined in Sec. 2.3.

4.7 Appendix 2: Alternative (matrix) form of stability analysis of multistep method

We will illustrate this alternative approach using the 2nd-order Adams–Bashforth method found at the beginning of Sec. 4.4. According to Appendix 2 of Lecture 3, that method can be recast in the form

$$\vec{\mathbf{y}}_{n+1} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \vec{\mathbf{y}}_n + \frac{h}{2} \begin{pmatrix} 3 & -1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix}, \qquad \vec{\mathbf{y}}_n \equiv \begin{pmatrix} Y_n \\ Y_{n-1} \end{pmatrix}. \tag{4.52}$$

Using $f = \lambda y$, as in the model equation (4.15), and seeking

$$\vec{\mathbf{y}}_{n+1} = \rho\,\vec{\mathbf{y}}_n \tag{4.53}$$

(see Remark 2 in Sec. 4.3, which now should be applied to a vector rather than a scalar), one finds:

$$\begin{pmatrix} 1 + (3h\lambda/2) - \rho & -h\lambda/2 \\ 1 & -\rho \end{pmatrix} \vec{\mathbf{y}}_n = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.54}$$

Since $\vec{\mathbf{y}}_n$ is not a zero vector (since otherwise we would get the exact zero for our numerical solution), the determinant of the matrix in (4.54) must vanish. This condition gives Eq. (4.26) for ρ.

Note that you may recognize ρ as being an eigenvalue of the matrix

$$\begin{pmatrix} 1 + (3h\lambda/2) & -h\lambda/2 \\ 1 & 0 \end{pmatrix}, \tag{4.55}$$

which relates $\vec{\mathbf{y}}_{n+1}$ to $\vec{\mathbf{y}}_n$ in the case of the model equation (i.e., when $f = \lambda y$).

4.8 Appendix 3: Justification of the coefficients in Nordsieck's method

Here, our goal is to establish the values for coefficients $\kappa_{y,a,b}$ defined right before the equations of Nordsick's 3-stage method (3.58). Our starting criterion is to require that the method with those coefficients be stable. It will turn out that one of those coefficients will remain undefined. This will give us the freedom to set it so as to restore the "lost" accuracy of the method, which (the lost accuracy) was described in the paragraph following Eqs. (3.58).

The calculations for the 3-stage method are quite cumbersome.[19] In order not to obscure the idea by technical details, below I will present the calculations for the 2-stage method, which will suffice to illustrate the general concept. The equations of this method with coefficients $\kappa_{y,a}$ (since for this method there is no b) yet undetermined are (compare with (3.59)):

$$Y_{n+1} = Y_n + h\, f_n + a_n + \kappa_y h\, (f_{n+1} - f^p)\,, \tag{4.56a}$$

$$a_{n+1} = a_n + \kappa_a h\, (f_{n+1} - f^p)\,, \tag{4.56b}$$

with f^p is still satisfying:

$$h\, f^p = h\, f_n + 2a_n. \tag{4.56c}$$

Using $f = \lambda y$, as in the model equation (4.15), and seeking

$$\begin{pmatrix} Y \\ a \end{pmatrix}_{n+1} = \rho \begin{pmatrix} Y \\ a \end{pmatrix}_n \tag{4.57}$$

(similarly to (4.53)), and denoting

$$\rho \equiv 1 + r \tag{4.58}$$

for convenience at the next step, one obtains:

$$\begin{pmatrix} -r(1-\kappa_y h\lambda) + h\lambda & 1 - 2\kappa_y \\ r\,\kappa_a h\lambda & -r - 2\kappa_a \end{pmatrix} \begin{pmatrix} Y \\ a \end{pmatrix}_n = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{4.59}$$

As in the reason given after (4.54), the determinant of the matrix must vanish, which yields:

$$(1 - \kappa_y h\lambda)\, r^2 + (2\kappa_a - (1+\kappa_a) h\lambda)\, r - 2\kappa_a h\lambda = 0. \tag{4.60}$$

Now, observe that as $h \to 0$, one of the two roots of this equation is 0, which is equivalent to $\rho|_{h=0} = 1$. Let us refer to this root as the principal root of (4.60). You may see that the principal root satisfies $\rho|_{h=0} = 1$ for all methods considered in Sec. 4.4. It is what one can require of the second root that is a *key* to Nordsieck's approach to the stability of his method. Namely, in order to maximize its chances of its staying within the unit circle $|\rho| = 1$ for $h > 0$, one can, most generally, require that it be as far as possible from the boundary of that circle for $h = 0$. In other words, Nordsieck required that

All roots ρ of (4.58), (4.60) except for the principal root must $= 0$ when $h = 0$. (4.61)

Condition (4.61) (along with notation (4.58)) yields the value for κ_a:

$$\kappa_a = 1/2. \tag{4.62}$$

[19] In this connection, recall that Nordsieck in his paper worked out the case of a 5-stage method, and did so without any help of symbolic software like Mathematica.

With the stability of the method having been optimized in the sense of (4.61), we now have a free coefficient κ_y, by adjusting which we can make sure that the LTE of this method is $O(h^4)$ (as it should be for the 3rd-order Adams–Moulton method (3.61), to which Nordsieck's 2-stage method is equivalent; see Appendix 3 in Lecture 3). This is done by requiring that the principal root of (4.58), (4.60) agree with the Taylor expansion of the exact solution up to $O(h^4)$. This principal root is $\rho_1 \equiv 1 + r_1$, where

$$\begin{aligned} r_1 &= \frac{-1 + (3/2)h\lambda + \sqrt{1 + h\lambda + (9/4 - 4\kappa_y)\,(h\lambda)^2}}{2(1 - \kappa_y h\lambda)} \\ &= h\lambda + \frac{1}{2}\,(h\lambda)^2 + \frac{1}{4}(4\kappa_y - 1)\,(h\lambda)^3 + \frac{1}{8}(4\kappa_y - 1)\,(h\lambda)^4 + O(h^5). \end{aligned} \tag{4.63}$$

The expansion in the last line was obtained by Mathematica.[20] It should be compared with the expansion of the exact solution (minus one) of the model equation (4.15),

$$e^{h\lambda} - 1 = h\lambda + \frac{1}{2}\,(h\lambda)^2 + \frac{1}{6}\,(h\lambda)^3 + \frac{1}{24}\,(h\lambda)^4 + O(h^5), \tag{4.64}$$

and required to match as many of its terms as possible. This yields that

$$\kappa_y = 5/12, \tag{4.65}$$

and the error of the resulting method, which can be verified to be (3.59), is $O(h^4)$.

For an m-stage Nordsieck method, which would include variables a_i, b_i, etc., the stability requirement (4.61) will constrain the coefficients $\kappa_{a,b,\ldots}$, and then the remaining coefficient κ_y can be chosen to make the LTE to be $O(h^{m+2})$.

4.9 Appendix 4: Effect of repeated application of the corrector equation on the stability of a P–C method

Note that in a P–C method, introduced in Sec. 3.5 of Lecture 3, one can apply the corrector equation more than once. For example, for the method (3.33), we will then have:

$$\begin{aligned} Y^p_{n+1} &= Y_n + \tfrac{1}{2}h(3f_n - f_{n-1}) \\ Y^{c,1}_{n+1} &= Y_n + \tfrac{1}{2}h\left(f_n + f(x_{n+1}, Y^p_{n+1})\right), \\ Y^{c,2}_{n+1} &= Y_n + \tfrac{1}{2}h\left(f_n + f(x_{n+1}, Y^{c,1}_{n+1})\right), \\ &\text{etc.} \\ Y_{n+1} &= Y^{c,k}_{n+1} \end{aligned} \tag{4.66}$$

Note that this repeated application of the corrector will *not* change the accuracy of the overall method, because it is limited by the accuracy of the corrector equation, which is still, in the case of (4.66), two.

What it *can* do, however, is, in some loose sense,[21] make the method "more implicit" (i.e., closer to the implicit equation represented by the corrector equation alone). Based on the material of this Lecture, one may expect that this can improve the stability region of the resulting P–[repeated C] method. You will be given the opportunity to explore this in a homework problem.

[20] See the previous footnote.

[21] Whether it may or may not be made more rigorous is beyond the expertise of this writer.

4.10 Questions for self-assessment

1. Explain the meanings of the concepts of consistency, stability, and convergence of a numerical method.
2. State the Lax Theorem.
3. Give the idea behind the proof of the Lax Theorem.
4. Why is the model problem (4.15) relevant to analyze stability of numerical methods?
5. Does the model problem (4.15) predict the behavior of the global error? Give a complete answer.
6. What is the general procedure of analyzing stability of a numerical method?
7. Obtain Eq. (4.26).
8. Obtain Eq. (4.28).[22]
9. Why does the characteristic polynomial for the 3rd-order Adams–Bashforth method (3.11) have exactly 3 roots?
10. Would you use the Leap-frog method to solve the ODE $y' = \sqrt{y} + x^2$?
11. Obtain (4.34) from (4.33).
12. Why is the Leap-frog method called *weakly* unstable?
13. Why is the method (3.22) called *strongly* unstable?
14. Are Adams–Bashforth methods always partially stable, or can they be weakly unstable?

    Hint: Look at Eq. (4.42) and determine what values a_0 through a_M are for these methods. You may review explicit forms of the the 2nd-, 3rd-, and 4th-order Adams-Bashforth methods in Lecture 3. (You may verify your guess by looking at (3.16).) Next, when $h\lambda = 0$, what are the roots $\rho_1, \rho_2, \ldots$ of Eq. (4.43)? Finally, make an educated guess what becomes of those roots when $0 < h|\lambda| \ll 1$.
15. Same question about Runge–Kutta methods.
16. Explain how the second equation in (4.44) follows from the first.
17. Present your opinion on the following issue. When programming any of the explicit methods, you wrote a Matlab function that could apply that method to an arbitrary function $f(x, y)$ in the ODE of (4.3). Would you write a Matlab function implementing the Implicit Euler method is a similar way?
18. Verify the statement made after Eq. (4.44).
19. Obtain (4.47) from (4.45).
20. Is the 3rd-order Adams–Moulton method that you obtained in Homework 3 (problem 3) A-stable?

[22] In the sentence above it, $(1 + \alpha)$ is the expression under the square root in (4.27).