

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

$$\|a\|_\infty = \max_n |a_n|. \quad (4.1)$$

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 ε_{tol} :

$$\max_n |\epsilon_n| \leq \varepsilon_{\text{tol}}, \quad \text{or} \quad \|\epsilon\| \leq \varepsilon_{\text{tol}}. \quad (4.2)$$

We now return to the main subject of this section.

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

$$y' = f(x, y), \quad y(x_0) = y_0. \quad (4.3)$$

is to assure that the numerical solution Y_n closely matches the analytical solution $y(x_n)$ ¹⁰. 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:

$$\lim_{h \rightarrow 0} \|\epsilon\| \equiv \lim_{h \rightarrow 0} \|y - Y\| = \lim_{h \rightarrow 0} \max_n |y_n - Y_n| = 0. \quad (4.4)$$

Note that when taking the limit $h \rightarrow 0$, 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 them we already know: It is just that the numerical scheme must match the original ODE closer and closer as $h \rightarrow 0$.

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:

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

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

$$F[Y_n, h] = 0. \quad (4.6)$$

¹⁰In 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:

$$F[y_n, h] = \tau_n. \quad (4.7)$$

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

$$\lim_{h \rightarrow 0} \|\tau\| = 0, \quad (4.8)$$

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 $\tau_n = O(h^l)$.

We now 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

$$F[U_n, h] = \xi_n, \quad (4.9)$$

where ξ_n is a small number that arises due to the round-off error¹¹. Intuitively, we expect Y_n and U_n to remain very close throughout the computation. 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.) 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. The considerations motivate yet another definition.

Definition 3: Consider an IVP

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

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

$$\|U - Y\| \leq C \|\xi\|, \quad (4.11)$$

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.

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.

¹¹In general, it may arise due to any other source.

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:

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

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:

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

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:

$$\|y - U\| = \|(y - Y) + (Y - U)\| \leq \|(y - Y)\| + \|(Y - U)\|, \quad (4.14)$$

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 $\text{Re}\lambda < 0$. 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 “ $\text{Re}\lambda < 0$ ” in (4.10).

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

$$y' = \lambda y, \quad (4.15)$$

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 $\text{Re}\lambda > 0$ 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:

$$(y - u)' = f(x, y) - f(x, u) \approx f_y(x, y) (y - u). \quad (4.16)$$

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: **The stability of a numerical method depends on the problem which it is applied to.** Indeed, to decide whether the method at hand is going to be stable for a given IVP (4.3), one should 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.

Incidentally, 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.

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 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:

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

(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:

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

For $\lambda < 0$ and, more generally, for $\operatorname{Re}\lambda < 0$, the true solution $y_0 e^{\lambda x}$ decreases, but the numerical solution (4.18) will decrease only if

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

E.g., to solve $y' = -30y$ (with any initial condition), we must use $h < \frac{2}{30}$ 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 $\operatorname{Re}\lambda < 0$, the corresponding numerical solution is stable only for *some* values of $h\lambda$. The region in the λh -plane where the method is stable is called the *region of stability* of the method.

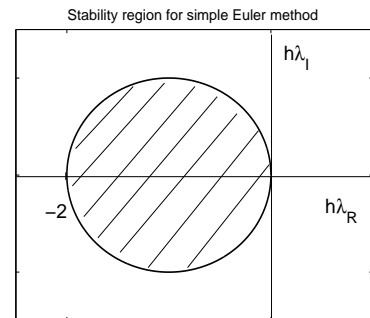
Let us find the region of stability of the simple Euler method. To this end, write λ as the sum of its real and imaginary parts: $\lambda = \lambda_R + i\lambda_I$ (note that here and below $i = \sqrt{-1}$). Then the first of the inequalities in (4.19) becomes

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

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

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

Remark: Note that the factor on the r.h.s. of (4.21) is quite expected: Since the Modified Euler is the 2nd-order method, its solution of the model problem (4.15) should be the 2nd-degree polynomial that approximates the exponential in the exact solution $y = y_0 e^{\lambda x}$.

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:

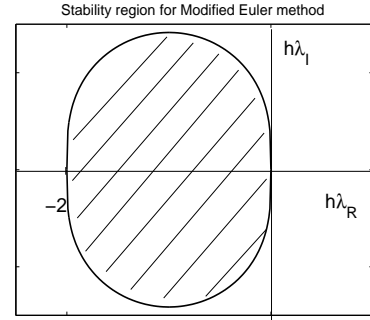
$$\left| 1 + h\lambda + \frac{1}{2}(h\lambda)^2 \right| = 1.$$

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

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

The corresponding region is shown above.



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

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

The expression on the l.h.s. is the fourth-degree polynomial approximating $e^{h\lambda}$; this is consistent with the Remark made after Eq. (4.21).

For real λ , criterion (4.23) reduces to

$$-2.79 \leq h\lambda \leq 0. \quad (4.24)$$

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):

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

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

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

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* $y = e^{\rho x}$, which yields the following polynomial equation for ρ :

$$\rho^2 + a_1 \rho + a_0 = 0.$$

Similarly, for the difference equation (4.25), we substitute $Y_n = \rho^n$ and, upon cancelling by the common factor ρ^{n-1} , obtain:

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

This quadratic equation has two roots:

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

In the limit of $h \rightarrow 0$ (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 :

$$\rho_1 \approx 1 + \lambda h, \quad \rho_2 \approx \frac{1}{2}\lambda h. \quad (4.28)$$

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

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

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*.

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.

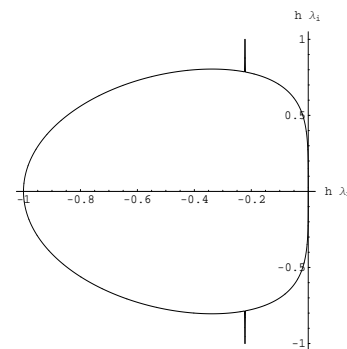
To require that the 2nd-order Adams–Bashforth method be stable is equivalent to requiring that *both* ρ_1 and ρ_2 satisfy

$$|\rho_1| \leq 1 \quad \text{and} \quad |\rho_2| \leq 1. \quad (4.30)$$

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, $|\rho_1| \leq 1$ and $|\rho_2| \leq 1$, 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 $h \rightarrow 0$, so they do not affect the numerical solution for sufficiently small h . For finite h , the requirement

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

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

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

$$Y_{n+1} - Y_{n-1} = 2h\lambda Y_n. \quad (4.31)$$

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

$$\rho^2 - 2h\lambda\rho - 1 = 0 \quad \Rightarrow \quad \rho_{1,2} = h\lambda \pm \sqrt{1 + (h\lambda)^2}. \quad (4.32)$$

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

$$\rho_1 \approx 1 + h\lambda, \quad \rho_2 \approx -1 + h\lambda. \quad (4.33)$$

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. Its behavior needs to be analyzed separately for $\lambda > 0$ and $\lambda < 0$.

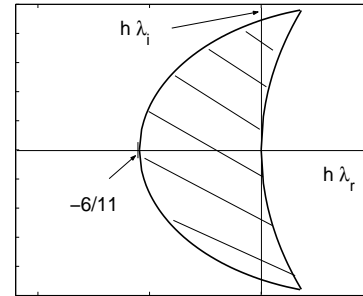
$\lambda > 0$ Then $\rho_2 = -(1 - h\lambda)$, so that $|\rho_2| < 1$, and the parasitic solution decays, whereas the true solution, $(1 + h\lambda)^n \approx e^{\lambda x}$, grows. Thus, the method is stable in this case.

$\lambda < 0$ Then $\rho_2 = -(1 + h|\lambda|)$, so that $|\rho_2| > 1$. Thus, the parasitic solution grows exponentially and makes the method unstable. Specifically, the difference solution will be the linear combination

$$Y_n = c_1\rho_1^n + c_2\rho_2^n \approx c_1e^{\lambda x} + c_2(-1)^ne^{-\lambda x}. \quad (4.34)$$

We see that the parasitic part (the 2nd term in (4.34) of this solution grows for $\lambda < 0$, whereas the true solution (the first term in (4.34)) decays; thus, for a sufficiently large x , the numerical solution will bear no resemblance to the true solution.

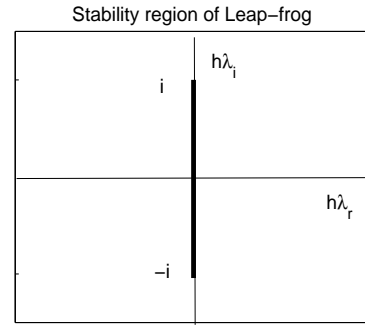
Stability region of 3rd-order
Adams-Bashforth



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

$$\lambda_R = 0 \quad \text{and} \quad -1 \leq h\lambda_I \leq 1. \quad (4.35)$$

However, since the numerical solution will still stay close to the true solution for $|\lambda x| \ll 1$, the Leap-frog method is called *weakly* unstable.



Divergent 3rd-order method (3.22)

For the model problem (4.15), that method becomes

$$Y_{n+1} + \frac{3}{2}Y_n - 3Y_{n-1} + \frac{1}{2}Y_{n-2} = 3h\lambda Y_n. \quad (4.36)$$

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

$$\rho^3 + \left(\frac{3}{2} - 3h\lambda\right)\rho^2 - 3\rho + \frac{1}{2} = 0. \quad (4.37)$$

To consider the limit $h \rightarrow 0$, we can simply set $h = 0$ as the lowest-order approximation. The cubic equation (4.37) reduces to

$$\rho^3 + \frac{3}{2}\rho^2 - 3\rho + \frac{1}{2} = 0, \quad (4.38)$$

which has the roots

$$\rho_1 = 1, \quad \rho_2 \approx -2.69, \quad \rho_3 \approx 0.19. \quad (4.39)$$

Then for small h , the numerical solution is

$$Y_n = \underbrace{c_1(1 + h\lambda)^n}_{\text{(approximate true solution)}} + \underbrace{c_2(-2.69 + O(h))^n}_{\text{(parasitic solution that explodes)}} + \underbrace{c_3(0.19 + O(h))^n}_{\text{(parasitic solution that decays)}} \quad (4.40)$$

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

$$Y_{n+1} - \sum_{k=0}^M a_k Y_{n-k} = h \sum_{k=0}^N b_k \lambda Y_{n-k}. \quad (4.41)$$

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

$$\rho^{M+1} - \sum_{k=0}^M a_k \rho^{M-k} = 0. \quad (4.42)$$

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. $\{\rho_2, \rho_3, \dots, \rho_{M+1}\}$, 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 $k = 2, \dots, M + 1$, then the method is stable for $h \rightarrow 0$. It may be either partially stable, as the single-step methods and Adams–Bashforth methods, or absolutely stable, as the implicit methods that we will consider next.

4.5 Stability analysis of implicit methods

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

$$Y_{n+1} = Y_n + h\lambda Y_{n+1}, \quad \Rightarrow \quad Y_n = Y_0 \left(\frac{1}{1 - h\lambda} \right)^n. \quad (4.43)$$

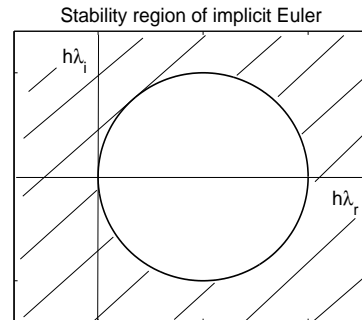
It can be verified (do it, following the lines of Sec. 0.5) that the r.h.s. of the last equation reduces for $h \rightarrow 0$ to the exact solution $y_0 e^{\lambda x}$, as it should. The stability condition is

$$\left| \frac{1}{1 - h\lambda} \right| \leq 1 \quad \Rightarrow \quad |1 - h\lambda| \geq 1. \quad (4.44)$$

The boundary of the stability region is the circle

$$(1 - h\lambda_R)^2 + (h\lambda_I)^2 = 1; \quad (4.45)$$

the stability region is the *outside* of that circle (see the second of inequalities (4.44)).



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 (3.43) 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 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).
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 (4.33) from (4.32).
12. Why is the Leap-frog method called weakly unstable?
13. Why is the method (3.22) called strongly unstable?
14. Are Adams–Bashforth and Runge–Kutta methods always partially stable, or can they be weakly unstable? (*Hint*: Look at Eqs. (4.41) and (4.42) and think of the roots ρ_1, ρ_2, \dots that these methods can have when $h\lambda \rightarrow 0$.)
15. Explain how the second equation in (4.43) follows from the first.
16. 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 in a similar way?
17. Verify the statement made after Eq. (4.43).
18. Obtain (4.45) from (4.44).
19. Is the 3rd-order Adams–Moulton method that you obtained in Homework 3 (problem 3) A-stable?