Chapter 7

Absolute Stability for Ordinary Differential Equations

7.1 Unstable computations with a zero-stable method

In the last chapter we investigated zero-stability, the form of stability needed to guarantee convergence of a numerical method as the grid is refined ($k \to 0$). In practice, however, we are not able to compute this limit. Instead we typically perform a single calculation with some particular nonzero time step k (or some particular sequence of time steps with a variable step size method). Since the expense of the computation increases as k decreases, we generally want to choose the time step as large as possible consistent with our accuracy requirements. How can we estimate the size of k required?

Recall that if the method is stable in an appropriate sense, then we expect the global error to be bounded in terms of the local truncation errors at each step, and so we can often use the local truncation error to estimate the time step needed, as illustrated below. But the form of stability now needed is something stronger than zero-stability. We need to know that the error is well behaved for the particular time step we are now using. It is little help to know that things will converge in the limit "for k sufficiently small." The potential difficulties are best illustrated with some examples.

Example 7.1. Consider the initial value problem (IVP)

$$u'(t) = -\sin t, \qquad u(0) = 1$$

with solution

$$u(t) = \cos t.$$

Suppose we wish to use Euler's method to solve this problem up to time $T = 2$. The local truncation error (LTE) is

$$\begin{aligned} \tau(t) &= \frac{1}{2} k u''(t) + O(k^2) \\ &= -\frac{1}{2} k \cos(t) + O(k^2). \end{aligned} \tag{7.1}$$

Since the function $f(t) = -\sin t$ is independent of u, it is Lipschitz continuous with Lipschitz constant $L = 0$, and so the error estimate (6.12) shows that

$$|E^n| \le T\|\tau\|_\infty = k \max_{0\le t\le T} |\cos t| = k.$$

Suppose we want to compute a solution with $|E| \le 10^{-3}$. Then we should be able to take $k = 10^{-3}$ and obtain a suitable solution after $T/k = 2000$ time steps. Indeed, calculating using $k = 10^{-3}$ gives a computed value $U^{2000} = -0.415692$ with an error $E^{2000} = U^{2000} - \cos(2) = 0.4548 \times 10^{-3}$.

Example 7.2. Now suppose we modify the above equation to

$$u'(t) = \lambda(u - \cos t) - \sin t, \tag{7.2}$$

where λ is some constant. If we take the same initial data as before, $u(0) = 1$, then the solution is also the same as before, $u(t) = \cos t$. As a concrete example, let's take $\lambda = -10$. Now how small do we need to take k to get an error that is 10^{-3}? Since the LTE (7.1) depends only on the true solution $u(t)$, which is unchanged from Example 7.1, we might hope that we could use the same k as in that example, $k = 10^{-3}$. Solving the problem using Euler's method with this step size now gives $U^{2000} = -0.416163$ with an error $E^{2000} = 0.161 \times 10^{-4}$. We are again successful. In fact, the error is considerably smaller in this case than in the previous example, for reasons that will become clear later.

Example 7.3. Now consider the problem (7.2) with $\lambda = -2100$ and the same data as before. Again the solution is unchanged and so is the LTE. But now if we compute with the same step size as before, $k = 10^{-3}$, we obtain $U^{2000} = -0.2453 \times 10^{77}$ with an error of magnitude 10^{77}. The computation behaves in an "unstable" manner, with an error that grows exponentially in time. Since the method is zero-stable and $f(u,t)$ is Lipschitz continuous in u (with Lipschitz constant $L = 2100$), we know that the method is convergent, and indeed with sufficiently small time steps we achieve very good results. Table 7.1 shows the error at time $T = 2$ when Euler's method is used with various values of k. Clearly something dramatic happens between the values $k = 0.000976$ and $k = 0.000952$. For smaller values of k we get very good results, whereas for larger values of k there is no accuracy whatsoever.

The equation (7.2) is a linear equation of the form (6.3) and so the analysis of Section 6.3.1 applies directly to this problem. From (6.7) we see that the global error E^n satisfies the recursion relation

$$E^{n+1} = (1 + k\lambda)E^n - k\tau^n, \tag{7.3}$$

where the local error $\tau^n = \tau(t_n)$ from (7.1). The expression (7.3) reveals the source of the exponential growth in the error—in each time step the previous error is multiplied by a factor of $(1 + k\lambda)$. For the case $\lambda = -2100$ and $k = 10^{-3}$, we have $1 + k\lambda = -1.1$ and so we expect the local error introduced in step m to grow by a factor of $(-1.1)^{n-m}$ by the end of n steps (recall (6.8)). After 2000 steps we expect the truncation error introduced in the first step to have grown by a factor of roughly $(-1.1)^{2000} \approx 10^{82}$, which is consistent with the error actually seen.

Note that in Example 7.2 with $\lambda = -10$, we have $1 + k\lambda = 0.99$, causing a *decay* in the effect of previous errors in each step. This explains why we got a reasonable result in Example 7.2 and in fact a better result than in Example 7.1, where $1 + k\lambda = 1$.

Table 7.1. *Errors in the computed solution using Euler's method for Example* 7.3, *for different values of the time step k. Note the dramatic change in behavior of the error for* $k < 0.000952$.

k	Error
0.001000	0.145252E+77
0.000976	0.588105E+36
0.000950	0.321089E-06
0.000800	0.792298E-07
0.000400	0.396033E-07

Returning to the case $\lambda = -2100$, we expect to observe exponential growth in the error for any value of k greater than $2/2100 = 0.00095238$, since for any k larger than this we have $|1 + k\lambda| > 1$. For smaller time steps $|1 + k\lambda| < 1$ and the effect of each local error decays exponentially with time rather than growing. This explains the dramatic change in the behavior of the error that we see as we cross the value $k = 0.00095238$ in Table 7.1.

Note that the exponential growth of errors does not contradict zero-stability or convergence of the method in any way. The method does converge as $k \to 0$. In fact the bound (6.12),

$$|E^n| \le e^{|\lambda|T} T \|\tau\|_\infty = O(k) \quad \text{as } k \to 0,$$

that we used to prove convergence allows the possibility of exponential growth with time. The bound is valid for all k, but since $Te^{|\lambda|T} = 2e^{4200} = 10^{1825}$ while $\|\tau\|_\infty = \frac{1}{2}k$, this bound does not guarantee any accuracy whatsoever in the solution until $k < 10^{-1825}$. This is a good example of the fact that a mathematical convergence proof may be a far cry from what is needed in practice.

7.2 Absolute stability

To determine whether a numerical method will produce reasonable results with a given value of $k > 0$, we need a notion of stability that is different from zero-stability. There are a wide variety of other forms of "stability" that have been studied in various contexts. The one that is most basic and suggests itself from the above examples is *absolute stability*. This notion is based on the linear test equation (6.3), although a study of the absolute stability of a method yields information that is typically directly useful in determining an appropriate time step in nonlinear problems as well; see Section 7.4.3.

We can look at the simplest case of the test problem in which $g(t) = 0$ and we have simply

$$u'(t) = \lambda u(t).$$

Euler's method applied to this problem gives

$$U^{n+1} = (1 + k\lambda)U^n$$

and we say that this method is *absolutely stable* when $|1+k\lambda| \leq 1$; otherwise it is unstable. Note that there are two parameters k and λ, but only their product $z \equiv k\lambda$ matters. The method is stable whenever $-2 \leq z \leq 0$, and we say that the *interval of absolute stability* for Euler's method is $[-2, 0]$.

It is more common to speak of the *region of absolute stability* as a region in the complex z plane, allowing the possibility that λ is complex (of course the time step k should be real and positive). The region of absolute stability (or simply the *stability region*) for Euler's method is the disk of radius 1 centered at the point -1, since within this disk we have $|1 + k\lambda| \leq 1$ (see Figure 7.1a). Allowing λ to be complex comes from the fact that in practice we are usually solving a system of ordinary differential equations (ODEs). In the linear case it is the eigenvalues of the coefficient matrix that are important in determining stability. In the nonlinear case we typically linearize (see Section 7.4.3) and consider the eigenvalues of the Jacobian matrix. Hence λ represents a typical eigenvalue and these may be complex even if the matrix is real. For some problems, looking at the eigenvalues is not sufficient (see Section 10.12.1, for example), but eigenanalysis is generally very revealing.

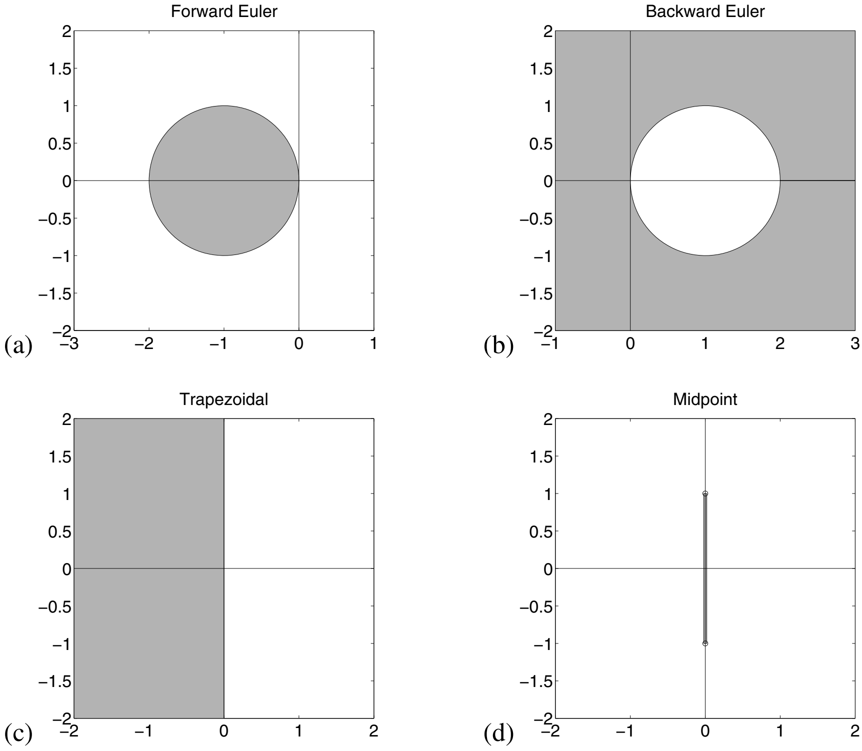

Figure 7.1. *Stability regions for* (a) *Euler,* (b) *backward Euler,* (c) *trapezoidal, and* (d) *midpoint (a segment on imaginary axis).*

7.3 Stability regions for linear multistep methods

For a general linear multistep method (LMM) of the form (5.44), the region of absolute stability is found by applying the method to $u' = \lambda u$, obtaining

$$\sum_{j=0}^{r} \alpha_j U^{n+j} = k \sum_{j=0}^{r} \beta_j \lambda U^{n+j},$$

which can be rewritten as

$$\sum_{j=0}^{r} (\alpha_j - z\beta_j) U^{n+j} = 0. \tag{7.4}$$

Note again that it is only the product $z = k\lambda$ that is important, not the values of k or λ separately, and that this is a dimensionless quantity since the decay rate λ has dimensions time^{-1}, while the time step has dimensions of time. This makes sense—if we change the units of time (say, from seconds to milliseconds), then the parameter λ will decrease by a factor of 1000 and we may be able to increase the numerical value of k by a factor of 1000 and still be stable. But then we also have to solve out to time $1000T$ instead of to time T, so we haven't really changed the numerical problem or the number of time steps required.

The recurrence (7.4) is a homogeneous linear difference equation of the same form considered in Section 6.4.1. The solution has the general form (6.26), where the ζ_j are now the roots of the characteristic polynomial $\sum_{j=0}^{r}(\alpha_j - z\beta_j)\zeta^j$. This polynomial is often called the *stability polynomial* and denoted by $\pi(\zeta; z)$. It is a polynomial in ζ but its coefficients depend on the value of z. The stability polynomial can be expressed in terms of the characteristic polynomials for the LMM as

$$\pi(\zeta; z) = \rho(\zeta) - z\sigma(\zeta). \tag{7.5}$$

The LMM is absolutely stable for a particular value of z if errors introduced in one time step do not grow in future time steps. According to the theory of Section 6.4.1, this requires that the polynomial $\pi(\zeta; z)$ satisfy the root condition (6.34).

Definition 7.1. *The* region of absolute stability *for the LMM* (5.44) *is the set of points z in the complex plane for which the polynomial $\pi(\zeta; z)$ satisfies the root condition* (6.34).

Note that an LMM is zero-stable if and only if the origin $z = 0$ lies in the stability region.

Example 7.4. For Euler's method,

$$\pi(\zeta; z) = \zeta - (1 + z)$$

with the single root $\zeta_1 = 1 + z$. We have already seen that the stability region is the disk in Figure 7.1(a).

Example 7.5. For the backward Euler method (5.21),

$$\pi(\zeta; z) = (1 - z)\zeta - 1$$

with root $\zeta_1 = (1-z)^{-1}$. We have

$$|(1-z)^{-1}| \le 1 \iff |1-z| \ge 1$$

so the stability region is the *exterior* of the disk of radius 1 centered at $z = 1$, as shown in Figure 7.1(b).

Example 7.6. For the trapezoidal method (5.22),

$$\pi(\zeta; z) = \left(1 - \frac{1}{2}z\right)\zeta - \left(1 + \frac{1}{2}z\right)$$

with root

$$\zeta_1 = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}.$$

This is a linear fractional transformation and it can be shown that

$$|\zeta_1| \le 1 \iff \text{Re}(z) \le 0,$$

where Re(z) is the real part. So the stability region is the left half-plane as shown in Figure 7.1(c).

Example 7.7. For the midpoint method (5.23),

$$\pi(\zeta; z) = \zeta^2 - 2z\zeta - 1.$$

The roots are $\zeta_{1,2} = z \pm \sqrt{z^2+1}$. It can be shown that if z is a pure imaginary number of the form $z = i\alpha$ with $|\alpha| < 1$, then $|\zeta_1| = |\zeta_2| = 1$ and $\zeta_1 \ne \zeta_2$, and hence the root condition is satisfied. For any other z the root condition is not satisfied. In particular, if $z = \pm i$, then $\zeta_1 = \zeta_2$ is a repeated root of modulus 1. So the stability region consists only of the open interval from $-i$ to i on the imaginary axis, as shown in Figure 7.1(d).

Since k is always real, this means the midpoint method is useful only on the test problem $u' = \lambda u$ if λ is pure imaginary. The method is not very useful for scalar problems where λ is typically real, but the method is of great interest in some applications with systems of equations. For example, if the matrix is real but skew symmetric ($A^T = -A$), then the eigenvalues are pure imaginary. This situation arises naturally in the discretization of hyperbolic partial differential equations (PDEs), as discussed in Chapter 10.

Example 7.8. Figures 7.2 and 7.3 show the stability regions for the r-step Adams–Bashforth and Adams–Moulton methods for various values of r. For an r-step method the polynomial $\pi(\zeta; z)$ has degree r and there are r roots. Determining the values of z for which the root condition is satisfied does not appear simple. However, there is a simple technique called the *boundary locus method* that makes it possible to determine the regions shown in the figures. This is briefly described in Section 7.6.1.

Note that for many methods the shape of the stability region near the origin $z = 0$ is directly related to the accuracy of the method. Recall that the stability polynomial $\rho(\zeta)$ for a consistent LMM always has a principal root $\zeta_1 = 1$. It can be shown that for z near 0 the polynomial $\pi(\zeta; z)$ has a corresponding principal root with behavior

$$\zeta_1(z) = e^z + O(z^{p+1}) \quad \text{as } z \to 0 \tag{7.6}$$

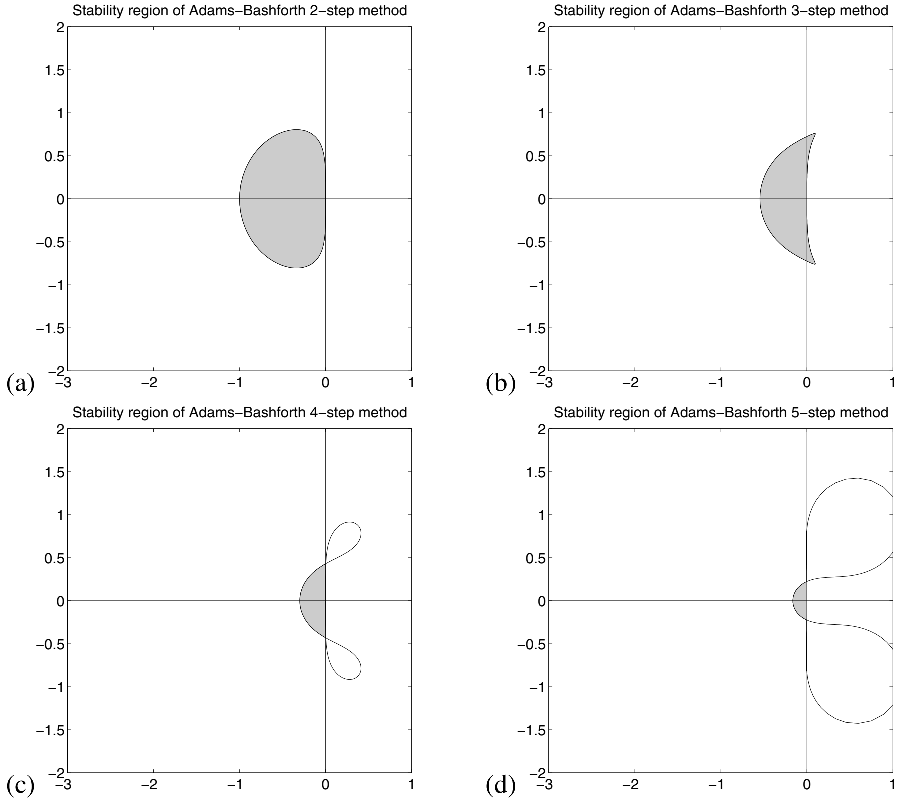

Figure 7.2. *Stability regions for some Adams–Bashforth methods. The shaded region just to the left of the origin is the region of absolute stability. See* Section 7.6.1 *for a discussion of the other loops seen in figures (c) and (d).*

if the method is pth order accurate. We can see this in the examples above for one-step methods, e.g., for Euler's method $\zeta_1(z) = 1 + z = e^z + O(z^2)$. It is this root that is giving the appropriate behavior $U^{n+1} \approx e^z U^n$ over a time step. Since this root is on the unit circle at the origin $z = 0$, and since $|e^z| < 1$ only when $\text{Re}(z) < 0$, we expect the principal root to move inside the unit circle for small z with $\text{Re}(z) < 0$ and outside the unit circle for small z with $\text{Re}(z) > 0$. This suggests that if we draw a small circle around the origin, then the left half of this circle will lie inside the stability region (unless some other root moves outside, as happens for the midpoint method), while the right half of the circle will lie outside the stability region. Looking at the stability regions in Figure 7.1 we see that this is indeed true for all the methods except the midpoint method. Moreover, the higher the order of accuracy in general, the larger a circle around the origin where this will approximately hold, and so the boundary of the stability region tends to align with the imaginary axis farther and farther from the origin as the order of the method increases, as observed in Figures 7.2 and 7.3. (The trapezoidal method is a bit of an anomaly, as its stability region exactly agrees with that of e^z for all z.)

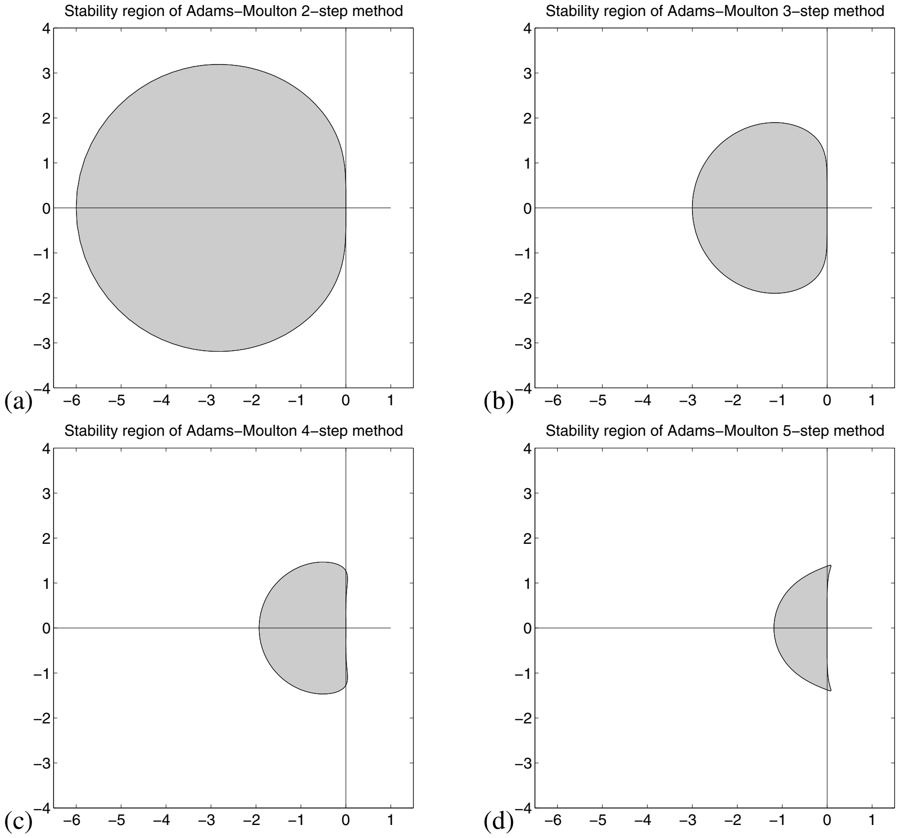

Figure 7.3. *Stability regions for some Adams–Moulton methods.*

See Section 7.6 for a discussion of ways in which stability regions can be determined and plotted.

7.4 Systems of ordinary differential equations

So far we have examined stability theory only in the context of a scalar differential equation $u'(t) = f(u(t))$ for a scalar function $u(t)$. In this section we will look at how this stability theory carries over to systems of m differential equations where $u(t) \in \mathbb{R}^m$. For a linear system $u' = Au$, where A is an $m \times m$ matrix, the solution can be written as $u(t) = e^{At}u(0)$ and the behavior is largely governed by the eigenvalues of A. A necessary condition for stability is that $k\lambda$ be in the stability region for each eigenvalue λ of A. For general nonlinear systems $u' = f(u)$, the theory is more complicated, but a good rule of thumb is that $k\lambda$ should be in the stability region for each eigenvalue λ of the Jacobian matrix $f'(u)$. This may not be true if the Jacobian is rapidly changing with time, or even for constant coefficient linear problems in some highly nonnormal cases (see [47] and Section 10.12.1 for an example), but most of the time eigenanalysis is surprisingly effective.

Before discussing this theory further we will review the theory of chemical kinetics, a field where the solution of systems of ODEs is very important, and where the eigenvalues of the Jacobian matrix often have a physical interpretation in terms of reaction rates.

7.4.1 Chemical kinetics

Let A and B represent chemical compounds and consider a reaction of the form

$$A \overset{K_1}{\to} B.$$

This represents a reaction in which A is transformed into B with rate $K_1 > 0$. If we let u_1 represent the concentration of A and u_2 represent the concentration of B (often denoted by $u_1 = [A]$, $u_2 = [B]$), then the ODEs for u_1 and u_2 are

$$\begin{aligned} u_1' &= -K_1 u_1, \\ u_2' &= K_1 u_1. \end{aligned}$$

If there is also a reverse reaction at rate K_2, we write

$$A \underset{K_2}{\overset{K_1}{\rightleftharpoons}} B$$

and the equations then become

$$\begin{aligned} u_1' &= -K_1 u_1 + K_2 u_2, \\ u_2' &= K_1 u_1 - K_2 u_2. \end{aligned} \tag{7.7}$$

More typically, reactions involve combinations of two or more compounds, e.g.,

$$A + B \underset{K_2}{\overset{K_1}{\rightleftharpoons}} AB.$$

Since A and B must combine to form AB, the rate of the forward reaction is proportional to the *product* of the concentrations u_1 and u_2, while the backward reaction is proportional to $u_3 = [AB]$. The equations become

$$\begin{aligned} u_1' &= -K_1 u_1 u_2 + K_2 u_3, \\ u_2' &= -K_1 u_1 u_2 + K_2 u_3, \\ u_3' &= K_1 u_1 u_2 - K_2 u_3. \end{aligned} \tag{7.8}$$

Note that this is a nonlinear system of equations, while (7.7) are linear.

Often several reactions take place simultaneously, e.g.,

$$\begin{aligned} A + B &\underset{K_2}{\overset{K_1}{\rightleftharpoons}} AB, \\ 2A + C &\underset{K_4}{\overset{K_3}{\rightleftharpoons}} A_2C. \end{aligned}$$

If we now let $u_4 = [C]$, $u_5 = [A_2C]$, then the equations are

$$\begin{aligned}
u_1' &= -K_1u_1u_2 + K_2u_3 - 2K_3u_1^2u_4 + 2K_4u_5,\\
u_2' &= -K_1u_1u_2 + K_2u_3,\\
u_3' &= K_1u_1u_2 - K_2u_3,\\
u_4' &= -K_3u_1^2u_4 + K_4u_5,\\
u_5' &= K_3u_1^2u_4 - K_4u_5.
\end{aligned} \tag{7.9}$$

Interesting kinetics problems can give rise to very large systems of ODEs. Frequently the rate constants $K_1, K_2, \ldots$ are of vastly different orders of magnitude. This leads to *stiff* systems of equations, as discussed in Chapter 8.

Example 7.9. One particularly simple system arises from the decay process

$$A \overset{K_1}{\to} B \overset{K_2}{\to} C.$$

Let $u_1 = [A]$, $u_2 = [B]$, $u_3 = [C]$. Then the system is linear and has the form $u' = Au$, where

$$A = \begin{bmatrix} -K_1 & 0 & 0\\ K_1 & -K_2 & 0\\ 0 & K_2 & 0 \end{bmatrix}. \tag{7.10}$$

Note that the eigenvalues are $-K_1$, $-K_2$, and 0. The general solution thus has the form (assuming $K_1 \neq K_2$)

$$u_j(t) = c_{j1}e^{-K_1t} + c_{j2}e^{-K_2t} + c_{j3}.$$

In fact, on physical grounds (since A decays into B which decays into C), we expect that u_1 simply decays to 0 exponentially,

$$u_1(t) = e^{-K_1t}u_1(0)$$

(which clearly satisfies the first ODE), and also that u_2 ultimately decays to 0 (although it may first grow if K_1 is larger than K_2), while u_3 grows and asymptotically approaches the value $u_1(0) + u_2(0) + u_3(0)$ as $t \to \infty$. A typical solution for $K_1 = 3$ and $K_2 = 1$ with $u_1(0) = 3$, $u_2(0) = 4$, and $u_3(0) = 2$ is shown in Figure 7.4.

7.4.2 Linear systems

Consider a linear system $u' = Au$, where A is a constant $m \times m$ matrix, and suppose for simplicity that A is diagonalizable, which means that it has a complete set of m linearly independent eigenvectors r_p satisfying $Ar_p = \lambda_p r_p$ for $p = 1, 2, \ldots, m$. Let $R = [r_1, r_2, \ldots, r_m]$ be the matrix of eigenvectors and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$ be the diagonal matrix of eigenvectors. Then we have

$$A = R\Lambda R^{-1} \quad \text{and} \quad \Lambda = R^{-1}AR.$$

Now let $v(t) = R^{-1}u(t)$. Multiplying $u' = Au$ by R^{-1} on both sides and introducing $I = RR^{-1}$ gives the equivalent equations

$$R^{-1}u'(t) = (R^{-1}AR)(R^{-1}u(t)),$$

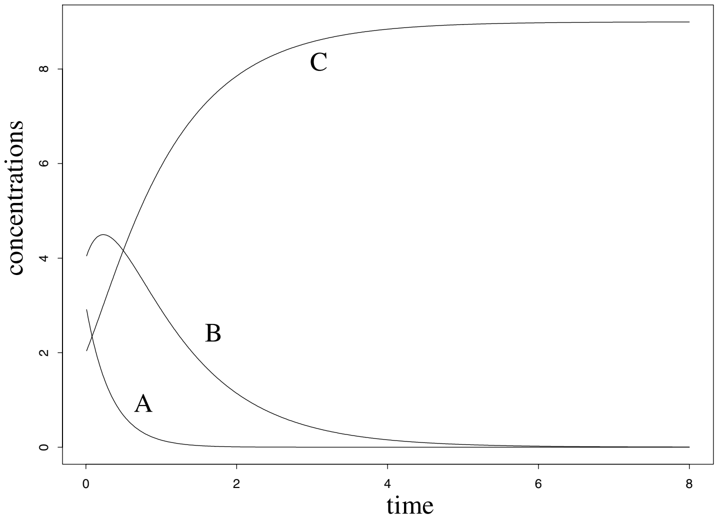

Figure 7.4. *Sample solution for the kinetics problem in Example* 7.9.

i.e.,

$$v'(t) = \Lambda v(t).$$

This is a diagonal system of equations that decouples into m independent scalar equations, one for each component of v. The pth such equation is

$$v'_p(t) = \lambda_p v_p(t).$$

A linear multistep method applied to the linear ODE can also be decoupled in the same way. For example, if we apply Euler's method, we have

$$U^{n+1} = U^n + kAU^n,$$

which, by the same transformation, can be rewritten as

$$V^{n+1} = V^n + k\Lambda V^n,$$

where $V^n = R^{-1}U^n$. This decouples into m independent numerical methods, one for each component of V^n. These take the form

$$V_p^{n+1} = (1 + k\lambda_p)V_p^n.$$

We can recover U^n from V^n using $U^n = RV^n$.

For the overall method to be stable, each of the scalar problems must be stable, and this clearly requires that $k\lambda_p$ be in the stability region of Euler's method for all values of p.

The same technique can be used more generally to show that an LMM can be absolutely stable only if $k\lambda_p$ is in the stability region of the method for each eigenvalue λ_p of the matrix A.

Example 7.10. Consider the linear kinetics problem with A given by (7.10). Since this matrix is upper triangular, the eigenvalues are the diagonal elements $\lambda_1 = -K_1, \lambda_2 = -K_2$, and $\lambda_3 = 0$. The eigenvalues are all real and we expect Euler's method to be stable provided $k \max(K_1, K_2) \le 2$. Numerical experiments easily confirm that this is exactly correct: when this condition is satisfied the numerical solution behaves well, and if k is slightly larger there is explosive growth of the error.

Example 7.11. Consider a linearized model for a swinging pendulum, this time with frictional forces added,

$$\theta''(t) = -a\theta(t) - b\theta'(t),$$

which is valid for small values of θ. If we introduce $u_1 = \theta$ and $u_2 = \theta'$ then we obtain a first order system $u' = Au$ with

$$A = \begin{bmatrix} 0 & 1 \\ -a & -b \end{bmatrix}. \tag{7.11}$$

The eigenvalues of this matrix are $\lambda = \frac{1}{2}\big(-b \pm \sqrt{b^2 - 4a}\big)$. Note in particular that if $b = 0$ (no damping), then $\lambda = \pm\sqrt{-a}$ are pure imaginary. For $b > 0$ the eigenvalues shift into the left half-plane. In the undamped case the midpoint method would be a reasonable choice, whereas Euler's method might be expected to have difficulties. In the damped case the opposite is true.

7.4.3 Nonlinear systems

Now consider a nonlinear system $u' = f(u)$. The stability analysis we have developed for the linear problem does not apply directly to this system. However, if the solution is slowly varying relative to the time step, then over a small time interval we would expect a linearized approximation to give a good indication of what is happening. Suppose the solution is near some value $\bar{u}$, and let $v(t) = u(t) - \bar{u}$. Then

$$v'(t) = u'(t) = f(u(t)) = f(v(t) + \bar{u}).$$

Taylor series expansion about $\bar{u}$ (assuming v is small) gives

$$v'(t) = f(\bar{u}) + f'(\bar{u})v(t) + O(\|v\|^2).$$

Dropping the $O(\|v\|^2)$ terms gives a linear system

$$v'(t) = Av(t) + b,$$

where $A = f'(\bar{u})$ is the Jacobian matrix evaluated at $\bar{u}$ and $b = f(\bar{u})$. Examining how the numerical method behaves on this linear system (for each relevant value of $\bar{u}$) gives a good indication of how it will behave on the nonlinear system.

Example 7.12. Consider the kinetics problem (7.8). The Jacobian matrix is

$$A = \begin{bmatrix} -K_1u_2 & -K_1u_1 & K_2 \\ -K_1u_2 & -K_1u_1 & K_2 \\ K_1u_2 & K_1u_1 & -K_2 \end{bmatrix}$$

with eigenvalues $\lambda_1 = -K_1(u_1 + u_2) - K_2$ and $\lambda_2 = \lambda_3 = 0$. Since $u_1 + u_2$ is simply the total quantity of species A and B present, this can be bounded for all time in terms of the initial data. (For example, we certainly have $u_1(t) + u_2(t) \leq u_1(0) + u_2(0) + 2u_3(0)$.) So we can determine the possible range of λ_1 along the negative real axis and hence how small k must be chosen so that $k\lambda_1$ stays within the region of absolute stability.

7.5 Practical choice of step size

As the examples at the beginning of this chapter illustrated, obtaining computed results that are within some error tolerance requires two conditions:

1. The time step k must be small enough that the local truncation error is acceptably small. This gives a constraint of the form $k \leq k_{\text{acc}}$, where k_{acc} depends on several things:

   - What method is being used, which determines the expansion for the local truncation error;
   - How smooth the solution is, which determines how large the high order derivatives occurring in this expansion are; and
   - What accuracy is required.

2. The time step k must be small enough that the method is absolutely stable on this particular problem. This gives a constraint of the form $k \leq k_{\text{stab}}$ that depends on the magnitude and location of the eigenvalues of the Jacobian matrix $f'(u)$.

Typically we would like to choose our time step based on accuracy considerations, so we hope $k_{\text{stab}} > k_{\text{acc}}$. For a given method and problem, we would like to choose k so that the local error in each step is sufficiently small that the accumulated error will satisfy our error tolerance, assuming some “reasonable” growth of errors. If the errors grow exponentially with time because the method is not absolutely stable, however, then we would have to use a smaller time step to get useful results.

If stability considerations force us to use a *much* smaller time step than the local truncation error indicates should be needed, then this particular method is probably not optimal for this problem. This happens, for example, if we try to use an explicit method on a “stiff” problem as discussed in Chapter 8, for which special methods have been developed.

As already noted in Chapter 5, most software for solving initial value problems does a very good job of choosing time steps dynamically as the computation proceeds, based on the observed behavior of the solution and estimates of the local error. If a time step is chosen for which the method is unstable, then the local error estimate will typically indicate a large error and the step size will be automatically reduced. Details of the shape of the stability region and estimates of the eigenvalues are typically *not* used in the course of a computation to choose time steps.

However, the considerations of this chapter play a big role in determining whether a given method or class of methods is suitable for a particular problem. We will also see in Chapters 9 and 10 that a knowledge of the stability regions of ODE methods is necessary in order to develop effective methods for solving time-dependent PDEs.

7.6 Plotting stability regions

7.6.1 The boundary locus method for linear multistep methods

A point $z \in \mathbb{C}$ is in the stability region $\mathcal{S}$ of an LMM if the stability polynomial $\pi(\zeta; z)$ satisfies the root condition for this value of z. It follows that if z is on the *boundary* of the stability region, then $\pi(\zeta; z)$ must have at least one root ζ_j with magnitude exactly equal to 1. This ζ_j is of the form

$$\zeta_j = e^{i\theta}$$

for some value of θ in the interval $[0, 2\pi]$. (Beware of the two different uses of π.) Since ζ_j is a root of $\pi(\zeta; z)$, we have

$$\pi(e^{i\theta}; z) = 0$$

for this particular combination of z and θ. Recalling the definition of π, this gives

$$\rho(e^{i\theta}) - z\sigma(e^{i\theta}) = 0 \tag{7.12}$$

and hence

$$z = \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})}.$$

If we know θ, then we can find z from this.

Since every point z on the boundary of $\mathcal{S}$ must be of this form for some value of θ in $[0, 2\pi]$, we can simply plot the parametrized curve

$$\tilde{z}(\theta) \equiv \frac{\rho(e^{i\theta})}{\sigma(e^{i\theta})} \tag{7.13}$$

for $0 \le \theta \le 2\pi$ to find the locus of all points which are *potentially* on the boundary of $\mathcal{S}$. For simple methods this yields the region $\mathcal{S}$ directly.

Example 7.13. For Euler's method we have $\rho(\zeta) = \zeta - 1$ and $\sigma(\zeta) = 1$, and so

$$\tilde{z}(\theta) = e^{i\theta} - 1.$$

This function maps $[0, 2\pi]$ to the unit circle centered at $z = -1$, which is exactly the boundary of $\mathcal{S}$ as shown in Figure 7.1(a).

To determine which side of this curve is the *interior* of $\mathcal{S}$, we need only evaluate the roots of $\pi(\zeta; z)$ at some random point z on one side or the other and see if the polynomial satisfies the root condition.

Alternatively, as noted on page 155, most methods are stable just to the left of the origin on the negative real axis and unstable just to the right of the origin on the positive real axis. This is often enough information to determine where the stability region lies relative to the boundary locus.

For some methods the boundary locus may cross itself. In this case we typically find that at most one of the regions cut out of the plane corresponds to the stability region. We can determine which region is $\mathcal{S}$ by evaluating the roots at some convenient point z within each region.

Example 7.14. The five-step Adams–Bashforth method gives the boundary locus seen in Figure 7.2(d). The stability region is the small semicircular region to the left of the origin where all roots are inside the unit circle. As we cross the boundary of this region one root moves outside. As we cross the boundary locus again into one of the loops in the right half-plane another root moves outside and the method is still unstable in these regions (two roots are outside the unit circle).

7.6.2 Plotting stability regions of one-step methods

If we apply a one-step method to the test problem $u' = \lambda u$, we typically obtain an expression of the form

$$U^{n+1} = R(z)U^n, \tag{7.14}$$

where $R(z)$ is some function of $z = k\lambda$ (typically a polynomial for an explicit method or a rational function for an implicit method). If the method is consistent, then $R(z)$ will be an approximation to e^z near $z = 0$, and if it is pth order accurate, then

$$R(z) - e^z = O(z^{p+1}) \quad \text{as } z \to 0. \tag{7.15}$$

Example 7.15. The pth order Taylor series method, when applied to $u' = \lambda u$, gives (since the jth derivative of u is $u^{(j)} = \lambda^j u$)

$$\begin{aligned} U^{n+1} &= U^n + k\lambda U^n + \frac{1}{2}k^2\lambda^2 U^n + \cdots + \frac{1}{p!}k^p\lambda^p U^n \\ &= \left(1 + z + \frac{1}{2}z^2 + \cdots + \frac{1}{p!}z^p\right) U^n. \end{aligned} \tag{7.16}$$

In this case $R(z)$ is the polynomial obtained from the first $p + 1$ terms of the Taylor series for e^z.

Example 7.16. If the fourth order Runge–Kutta method (5.33) is applied to $u' = \lambda u$, we find that

$$R(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4, \tag{7.17}$$

which agrees with $R(z)$ for the fourth order Taylor series method.

Example 7.17. For the trapezoidal method (5.22),

$$R(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \tag{7.18}$$

is a rational approximation to e^z with error $O(z^3)$ (the method is second order accurate). Note that this is also the root of the linear stability polynomial that we found by viewing this as an LMM in Example 7.6.

Example 7.18. The TR-BDF2 method (5.37) has

$$R(z) = \frac{1 + \frac{5}{12}z}{1 - \frac{7}{12}z + \frac{1}{12}z^2}. \tag{7.19}$$

This agrees with e^z to $O(z^3)$ near $z = 0$.

From the definition of absolute stability given at the beginning of this chapter, we see that the region of absolute stability for a one-step method is simply

$$\mathcal{S} = \{z \in \mathbb{C} : |R(z)| \le 1\}. \tag{7.20}$$

This follows from the fact that iterating a one-step method on $u' = \lambda u$ gives $|U^n| = |R(z)|^n |U^0|$ and this will be uniformly bounded in n if z lies in $\mathcal{S}$.

One way to attempt to compute $\mathcal{S}$ would be to compute the boundary locus as described in Section 7.6.1 by setting $R(z) = e^{i\theta}$ and solving for z as θ varies. This would give the set of z for which $|R(z)| = 1$, the boundary of $\mathcal{S}$. There's a problem with this, however: when $R(z)$ is a higher order polynomial or rational function there will be several solutions z for each θ and it is not clear how to connect these to generate the proper curve.

Another approach can be taken graphically that is more brute force, but effective. If we have a reasonable idea of what region of the complex z-plane contains the boundary of $\mathcal{S}$, we can sample $|R(z)|$ on a fine grid of points in this region and approximate the level set where this function has the value 1 and plot this as the boundary of $\mathcal{S}$. This is easily done with a contour plotter, for example, using the `contour` command in MATLAB. Or we can simply color each point depending on whether it is inside $\mathcal{S}$ or outside.

For example, Figure 7.5 shows the stability regions for the Taylor series methods of orders 2 and 4, for which

$$\begin{aligned} R(z) &= 1 + z + \frac{1}{2}z^2, \\ R(z) &= 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{24}z^4, \end{aligned} \tag{7.21}$$

respectively. These are also the stability regions of the second order Runge–Kutta method (5.30) and the fourth order accurate Runge–Kutta method (5.33), which are easily seen to have the same stability functions.

Note that for a one-step method of order p, the rational function $R(z)$ must agree with e^z to $O(z^{p+1})$. As for LMMs, we thus expect that points very close to the origin will lie in the stability region $\mathcal{S}$ for $\text{Re}(z) < 0$ and outside of $\mathcal{S}$ for $\text{Re}(z) > 0$.

7.7 Relative stability regions and order stars

Recall that for a one-step method the stability region $\mathcal{S}$ (more properly called the region of absolute stability) is the region $\mathcal{S} = \{z \in \mathbb{C} : |R(z)| \le 1\}$, where $U^{n+1} = R(z)U^n$ is the relation between U^n and U^{n+1} when the method is applied to the test problem $u' = \lambda u$. For $z = \lambda k$ in the stability region the numerical solution does not grow, and hence the method is absolutely stable in the sense that past errors will not grow in later time steps.

On the other hand, the true solution to this problem, $u(t) = e^{\lambda t}u(0)$, is itself exponentially growing or decaying. One might argue that if $u(t)$ is itself decaying, then it isn't good enough to simply have the past errors decaying, too—they should be decaying at a faster rate. Or conversely, if the true solution is growing exponentially, then perhaps it is fine for the error also to be growing, as long as it is not growing faster.

This suggests defining the *region of relative stability* as the set of $z \in \mathbb{C}$ for which $|R(z)| \le |e^z|$. In fact this idea has not proved to be particularly useful in terms of judging

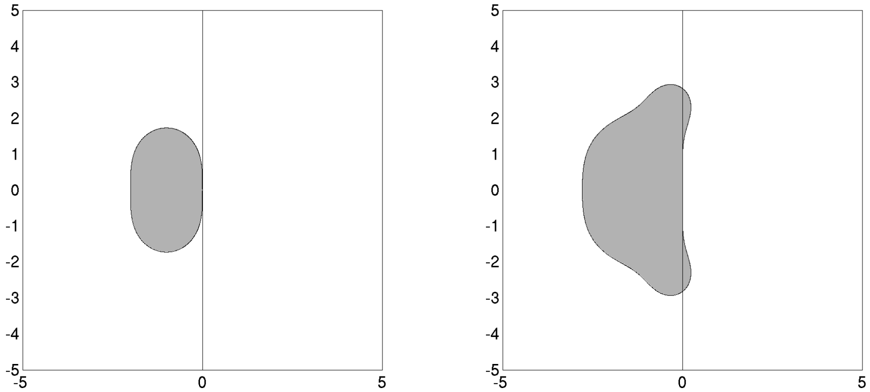

Figure 7.5. *Stability regions for the Taylor series methods of order 2 (left) and 4 (right).*

the practical stability of a method for finite-size time steps; absolute stability is the more useful concept in this regard.

Relative stability regions also proved hard to plot in the days before good computer graphics, and so they were not studied extensively. However, a pivotal 1978 paper by Wanner, Hairer, and Nørsett [99] showed that these regions are very useful in proving certain types of theorems about the relation between stability and the attainable order of accuracy for broad classes of methods. Rather than speaking in terms of regions of relative stability, the modern terminology concerns the *order star* of a rational function $R(z)$, which is the set of three regions $(\mathcal{A}_-,\ \mathcal{A}_0,\ \mathcal{A}_+)$:

$$\begin{aligned}\mathcal{A}_- &= \{z \in \mathbb{C} : \ |R(z)| < |e^z|\} = \{z \in \mathbb{C} : \ |e^{-z}R(z)| < 1\},\\ \mathcal{A}_0 &= \{z \in \mathbb{C} : \ |R(z)| = |e^z|\} = \{z \in \mathbb{C} : \ |e^{-z}R(z)| = 1\},\\ \mathcal{A}_+ &= \{z \in \mathbb{C} : \ |R(z)| > |e^z|\} = \{z \in \mathbb{C} : \ |e^{-z}R(z)| > 1\}.\end{aligned} \tag{7.22}$$

These sets turn out to be much more strange looking than regions of absolute stability. As their name implies, they have a star-like quality, as seen, for example, in Figure 7.6, which shows the order stars for the same two Taylor polynomials (7.21), and Figure 7.7, which shows the order stars for two implicit methods. In each case the shaded region is $\mathcal{A}_+$, while the white region is $\mathcal{A}_-$ and the boundary between them is $\mathcal{A}_0$. Their behavior near the origin is directly tied to the order of accuracy of the method, i.e., the degree to which $R(z)$ matches e^z at the origin. If $R(z) = e^z + Cz^{p+1} +$ higher order terms, then since $e^{-z} \approx 1$ near the origin,

$$e^{-z}R(z) \approx 1 + Cz^{p+1}. \tag{7.23}$$

As z traces out a small circle around the origin (say, $z = \delta e^{2\pi i\theta}$ for some small δ), the function $z^{p+1} = \delta^{p+1}e^{2(p+1)\pi i\theta}$ goes around a smaller circle about the origin $p+1$ times and hence crosses the imaginary axis $2(p+1)$ times. Each of these crossings corresponds to z moving across $\mathcal{A}_0$. So in a disk very close to the origin the order star must consist of $p+1$ wedgelike sectors of $\mathcal{A}_+$ separated by $p+1$ sectors of $\mathcal{A}_-$. This is apparent in Figures 7.6 and 7.7.

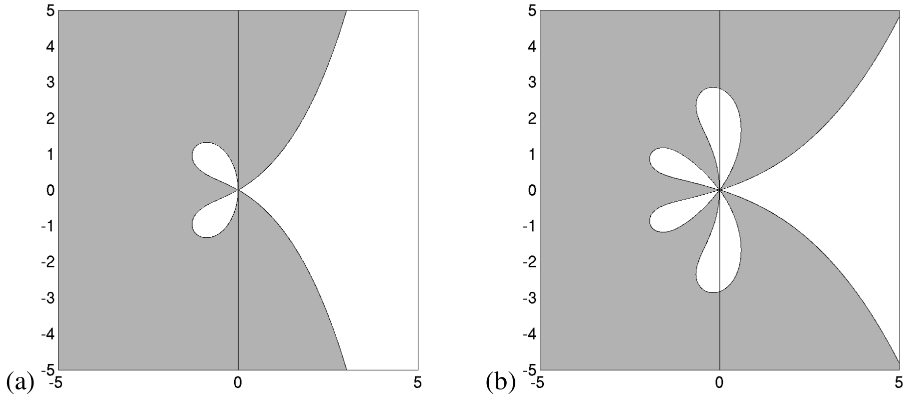

Figure 7.6. *Order stars for the Taylor series methods of order* (a) 2 *and* (b) 4.

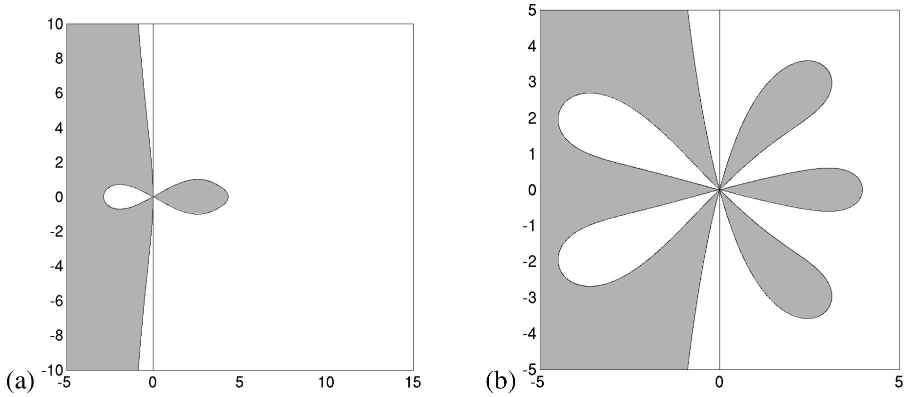

Figure 7.7. *Order stars for two A-stable implicit methods,* (a) *the TR-BDF2 method* (5.37) *with* $R(z)$ *given by* (7.19)*, and* (b) *the fifth-order accurate Radau*5 *method* [44]*, for which* $R(z)$ *is a rational function with degree* 2 *in the numerator and* 3 *in the denominator.*

It can also be shown that each bounded finger of $\mathcal{A}_-$ contains at least one root of the rational function $R(z)$ and each bounded finger of $\mathcal{A}_+$ contains at least one pole. (There are no poles for an explicit method; see Figure 7.6.) Moreover, certain stability properties of the method can be related to the geometry of the order star, facilitating the proof of some "barrier theorems" on the possible accuracy that might be obtained.

This is just a hint of the sort of question that can be tackled with order stars. For a better introduction to their power and beauty, see, for example, [44], [51], [98].