Math 563 Lecture Notes
ODE Initial value problems (Part I)

Spring 2020

The point: An introduction to numerical methods for solving initial value problems. The focus here is on the fundamental theory and concepts of stability and convergence. The theory here has important practical consequences; the stability of methods must be carefully considered when selecting which one to use on a given problem.

Related reading: Ascher & Petzold, *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*, Chapter 2.1-2.4 and 3.1-3.4 (the notes here follow the presentation in this chapter). Quarteroni, 11.3.2 and 11.3.3. Section 11.3.2 is a more technical version of what was done in class.

1 Review of ODEs

1.1 IVPs vs. BVPs

An initial value problem (IVP) in one dimension takes the form

$$y' = f(t, y), \quad y(t_0) = y_0.$$

Typically, we consider solving the ODE **forward** in 'time' (the independent variable), in which case the value $y(t)$ depends on the solution at previous times. One can equivalently consider solving backward in time; for the most part, this half will be omitted for simplicity.

The initial data $y(t_0) = y_0$ is carried by the ODE; in this way we can (theoretically and numerically) follows this data from the initial time t_0 to solve the ODE.

In contrast, a boundary value problem includes 'boundary conditions' at more than one point, like

$$y'' = f(x, y), \quad y(a) = y_1, \quad y(b) = y_2, \quad x \in [a, b]$$

We cannot just start at one point to solve, because the solution at all points depends on both boundary conditions. This added complexity makes boundary value problems much harder to solve than IVPs.

A **system** of first-order ODEs has the form

$$\mathbf{y}' = F(t, \mathbf{y}), \quad F : \mathbb{R} \to \mathbb{R}^n$$

for a function $\mathbf{y}(t) : \mathbb{R} \to \mathbb{R}^n$. Numerical methods for systems, for the most part, are straightforward extensions of the scalar version (up to some technical details and a few key points to be addressed).

Note that an n-th order ODE or system of ODEs can always be converted to a first order system, e.g.

$$y^{(n)} = f(t, y, y', \cdots, y^{(n-1)})$$

can be converted by setting$w_1 = y$ and $w_k = y^{(k-1)}$ for $k = 2 \cdots n$ so that

$$w_k' = w_{k+1}, \quad k = 1, \cdots n-1, \qquad w_n' = f(t, w_1, \cdots, w_n).$$

For this reason, it is sufficient to develop numerical methods only for first-order ODEs and first-order systems, unless there is reason to exploit the structure of the n-th order ODE.

1.2 Notation

We will use $f \in C([a,b])$ and $f \in C^k([a,b])$ to denote a continuous function in $[a,b]$ or k-continuously differentiable function. If the domain is implied, then $f \in C^k$ is used to mean continuous in that domain (e.g. the domain where the ODE solution is defined).

For the most part, $y(t)$ will denote an ODE solution, $y_n = y(t_n)$ its value at a point t_n and u_n will denote an approximation at t_n (the notation may vary as we run out of letters, so it is important to be careful when reading).

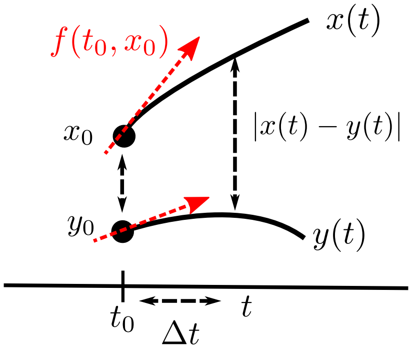
1.3 Important theorems

The results here are the fundamentals relevant to the numerics; some technical details are omitted. We will revisit sketches of proofs if they are useful analogies to the numerical verisons.

Definition (Lipschitz in a variable): Let $D \subset \mathbb{R}^2$ be a simply connected domain in the (t, y) plane. A function $f : \mathbb{R}^2 \to \mathbb{R}$ is **Lipschitz** in y if

$$|f(t, y_1) - f(t, y_2)| \leq L|y_1 - y_2| \text{ for all } (t, y_1), (t, y_2) \in D$$

for some constant $L > 0$ (the **Lipchitz constant**).



Main theorem (existence/stability): Consider the initial value problem

$$y'(t) = f(t, y(t)), \quad y(a) = y_0$$

and the 'strip' in the (t, y) plane

$$R = \{(t, y) : a \leq t \leq b, \ y \in \mathbb{R}\}.$$

If f is **Lipschitz in** y with constant L within R then:

i) The IVP has a unique solution $y(t)$ in $[a, b]$.

ii) If $z(t)$ is a solution starting with 'perturbed data' $z(a) = z_0$ then

$$|y(t) - z(t)| \leq e^{L(t-t_0)}|y_0 - z_0| \text{ in } R.$$

iii) If $z(t)$ solves the 'perturbed' problem

$$z'(t) = f(t, z) + h(t, z), \quad z(a) = z_0$$

and the perturbation satisfies $|h(t, z)| \leq M$ in R then

$$|y(t) - z(t)| \leq e^{L(t-t_0)}|y_0 - z_0| + M\frac{e^{L(t-t_0)} - 1}{L}.$$

In particular, $z(t) \to y(t)$ uniformly in $[a, b]$ as $M, |y_0 - z_0| \to 0$ (the perturbed solution approaches the original as the perturbation goes to zero).

Backwards case: The theorem can be applied 'backwards' too, to obtain some interval $[c, b]$ with $c < a < b$ on which the solution is defined. The bound (ii) then involves $|t - t_0|$ (always positive) instead of $t - t_0$.

Extension theorem: If the strip R is replaced by a domain D containing the initial point (a, y_0), then (i) is replaced by:

(i'): The solution exists in $[a, \epsilon]$ for some $\epsilon > 0$ and extends until it leaves D.

For instance, the IVP

$$y' = y^2, \quad y(0) = 1$$

does not have $f(t, y) = y^2$ Lipschitz in any strip (because $\partial f/\partial y = 2y$ is not bounded in y). However, it is Lipschitz in any finite range of y, e.g. $(t, y) \in [-1, 1] \times [-1000, 1000]$. Then the solution is guaranteed to exist until it leaves this region - either by t or y growing large. In this case, the actual solution $y(t) = 1/(1 - t)$ blows up as $t \nearrow 1$.

For systems: For an n-dimensional system of ODEs

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(a) = \mathbf{y}_0$$

where $\mathbf{y}(t) : \mathbb{R} \to \mathbb{R}^n$, the definitions extend in a straightforward way. 'Lipschitz in $\mathbf{y}$' means

$$\|f(t, \mathbf{y}_1) - f(t, \mathbf{y}_2)\| \leq L\|\mathbf{y}_1 - \mathbf{y}_2\|$$

where $\|y\|$ is any suitable norm in $\mathbb{R}^n$ (e.g. $\|y\| = \max_{1\leq i\leq n} |y_i|$) and the other absolute values are replaced by norms in the same way.

Stability (informal): Consider $\mathbf{y}(t)$ solving the initial value problem

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0.$$

Let $\mathbf{z}(t)$ denote the solution to the IVP with initial data $\mathbf{z}(0) = \mathbf{z}_0$. The solution is called

- **stable** (or 'Lyapunov stable') if, for each small $\delta > 0$ there is an $\epsilon > 0$ such that

$$\|\mathbf{y}_0 - \mathbf{z}_0\| < \delta \implies \|\mathbf{y}(t) - \mathbf{z}(t)\| < \epsilon \text{ for all } t.$$

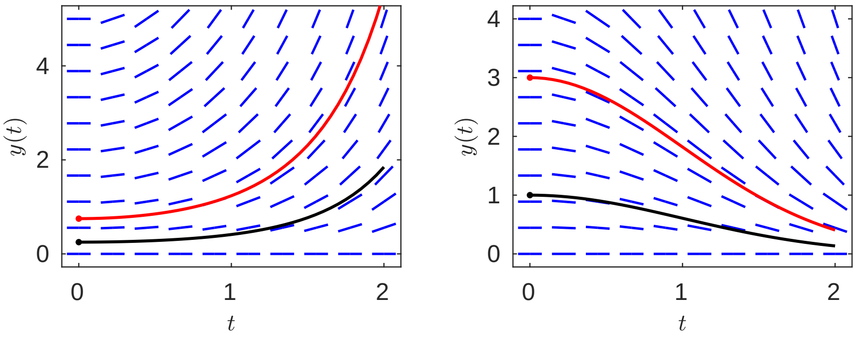
- **asymptotically stable** if there is a δ such that

$$\|\mathbf{y}_0 - \mathbf{z}_0\| < \delta \implies \lim_{t\to\infty} \|\mathbf{y}(t) - \mathbf{z}(t)\| = 0.$$

- **unstable** otherwise.

Technical point: In addition, one can also require that the same bounds must hold if $\mathbf{z}(t)$ instead solves a perturbed problem $\mathbf{y}' = (\mathbf{t}, \mathbf{y}) + \mathbf{h}(\mathbf{t}, \mathbf{y})$ and $\|\mathbf{h}\| \leq \delta$.

Below are sketches of the difference in two solutions that start at nearby points (t_0, x_0) and (t_0, y_0) and numerical examples for $y' = ty$ and $y' = -ty$. Solutions on the left are unstable; solutions on the right are stable.



2 Definitions; Euler's method

Consider the scalar first order IVP

$$y' = f(t, y), \quad y(0) = y_0$$

For a typical method, we seek a solution in an interval $[0, b]$ that is defined on **grid points**

$$0 = t_0 < t_1 < \cdots < t_N = b.$$

Let $y_n = y(t_n)$ denote the exact solution at each grid point (this shorthand will be used when convenient). The goal is to construct an approximation $\{u_n\}$ such that

$$u_n \approx y_n = y(t_n), \quad n = 0, \cdots, N$$

starting with $u_0 = y_0$ (the given value). The approximation $\{u_n\}$ is called a **grid function**, because it takes on values only at the grid points (but is approximating a function).

The spacing $h_n = t_{n+1} - t_n$ between successive points is the **step size** (at that point).

Convergence: The **global error** in the grid function u_n defined in $[0, b]$ is

$$\max_{0 \le n \le N} |u_n - y(t_n)|.$$

Let h be the maximum spacing between grid points, i.e.

$$h = \max_{0 \le n \le N-1} h_n, \quad h_n = t_{n+1} - t_n.$$

Then the method is said to be **convergent** with order p if

$$\max_{0 \le n \le N} |u_n - y(t_n)| = O(h^p) \text{ as } h \to 0.$$

Euler's method: A simple method can be obtained by use of Taylor series. Assume for simplicity that h is fixed (equal spacing). The exact solution $y(t)$ (recall $y_n = y(t_n)$) satisfies

$$y'(t_n) = f(t_n, y_n).$$

The point is that derivatives of $y(t)$ are known in terms of t and y. We cannot just solve for $y(t_n)$, but the data at t_n can be used to estimate the next value. To get y at the next grid point, Taylor expand around t_n:

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2} y''(\xi_{n+1}), \qquad \xi_{n+1} \in (t_n, t_{n+1}).$$

But $y'(t_n)$ is known from the ODE, so

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2} y''(\xi_n).$$

Dropping the $O(h^2)$ term gives a computable formula (a **finite difference equation**/FDE)

$$u_{n+1} = u_n + hf(t_n, u_n), \qquad u_0 = y_0.$$

This formula, **Euler's method**, generates an approximation u_n to the true solution.

Note that the FDE is an approximation to the ODE, and relates to the 'exact' resut relating y_{n+1} to y_n up to an error term:

$$\begin{aligned} \text{ODE:} \quad & y' = f(t,y) \\ \text{exact:} \quad & \frac{y_{n+1}-y_n}{h} = f(t_n, y_n) + \frac{h}{2}y''(\xi_{n+1}) \\ \text{FDE:} \quad & \frac{u_{n+1}-u_n}{h} = f(t_n, u_n) \end{aligned}$$

Definition (local truncation error): Let $A_h u$, acting on a grid function u_n, denote the 'finite difference operator' the defines the formula by

$$A_h u(t_n) = 0.$$

This is unique up to some extra factor, so the convention is that

$$A_h u(t_n) \approx y'(t_n) - f(t_n, y(t_n)).$$

The **local truncation error** (LTE) is the result of applying the finite difference operator to the exact solution:

$$\text{LTE} = \tau_{n+1} := A_h y(t_n)$$

The LTE is the left-over when plugging the exact solution into the finite difference formula.

For instance, for Euler's method, plugging $y(t)$ into the FDE gives

$$\tau_{n+1} = \frac{y_{n+1}-y_n}{h} - f(t_n, y_n)$$

which, from the earlier calculations, yields the LTE in the form

$$\tau_{n+1} = \frac{h}{2}y''(\xi_{n+1}).$$

Caution: Note that the LTE will typically be obtained naturally as part of deriving the FDE. One just has to be careful to **divide by** h appropriately. The actual error incurred at step n is more like $h^2 f''(\xi_n)/2$ which differs from the LTE by a factor of h.

Definition (consistency): A finite difference formula is **consistent** (with order p) if

$$\text{LTE} = O(h^p).$$

The reason for the dividing-by-h convention is so that the order of the local truncation error matches the global error.

2.1 Consistency and stability

The LTE does **not** describe $1/h$ times the error in the n-th step,

$$u_{n+1} = u_n + \text{error}_n,$$

because the LTE 'assumes' that the solution at time n is exact. In reality, there is some accumulated error that propagates from n to $n+1$. For this reason, consistency alone is far from enough to guarantee convergence.

To prove convergence, the strategy is to figure out what additional condition is needed to turn consistency into convergence, i.e. a result like

$$\text{consistency } + \text{ ???? } \implies \text{convergence.}$$

The key result is the **Lax equivalence theorem**, which says that an extra **stability** condition is required. This condition is analogous to 'stability' for ODEs (see subsection 1.3), and can take several forms depending on the problem.

Stability, v1:[1] For initial value problems, the fundamental condition is **zero stability**.

Definition (zero stability, informal): Suppose an FDE generates grid functions x_n (for $n = 0, \cdots, N$). The FDE is **zero stable** if, for small enough h, there is a constant M such that, for any two grid functions v_n and w_n,

$$|v_n - w_n| \leq M \left((|x_0 - z_0| + \max_{0 \leq j \leq N-1} |A_h v(t_j) - A_h w(t_j)| \right)$$

for $1 \leq n \leq N$. That is, small perturbations in the initial data ($x_0 \to z_0$) or in the formula at time j (the last term) lead to small changes in the result.

Note that here, $A_h v(t_j)$ is an 'error' (times $1/h$) in the j-th step, e.g. for Euler's method,

$$v_{j+1} = v_j + hf(t_j, v_j) + hA_h v(t_j).$$

If $\{u_j\}$ is the approximation to the ODE, then $A_h u(t_j)$ is either zero (ideally) or $1/h$ times whatever rounding error was introduced at that step.

It follows (by the way the definition is set up) that consistency and zero-stability implies convergence. Moroever, if the method is consistent with order p then it converges with order p since then

$$\max_{1 \leq n \leq N} |u_n - y(t_n)| \leq M \max_{0 \leq j \leq N-1} |\tau_j| = O(h^p).$$

[1] Definition from Ascher & Petzold, *Computer methods for ordinary differential equations*. Because there are equivalent conditions and variations, the definition may vary from source to source.

2.2 Convergence for Euler's method

The proof for Euler's method will illustrate the idea. We have already shown that Euler's method is consistent with order 1. Now let t_j, b etc. be as setup before and suppose that

$$f(t,y) \text{ is Lipschitz in } y \text{ for } t \in [0,b] \text{ with constant } L.$$

Now suppose that u_n and v_n are grid functions. From the definition of Euler's method, We have, from the definition of the LTE and Euler's method,

$$\begin{aligned} v_{n+1} &= v_n + hf(t_n, v_n) + hA_h v(t_n) \\ w_{n+1} &= w_n + hf(t_n, w_n) + hA_h w(t_n). \end{aligned}$$

Now let

$$\tau = \max_{1 \leq j \leq N} |A_h v(t_n) - A_h w(t_n)|.$$

Taking the difference gives four terms, comparing v and w at times t_n and t_{n+1}, f evaluated at t_n and the 'local errors' hA_h. The last term is just bounded by $h\tau$ by definition. For the f term, use the Lipschitz condition:

$$\begin{aligned} |v_{n+1} - w_{n+1}| &\leq |v_n - w_n| + h(f(t_n, v_n) - f(t_n, w_n)) + h\tau \\ &= |v_n - w_n| + hL|v_n - w_n| + h\tau. \end{aligned}$$

Let $d_n = v_n - w_n$. Then

$$|d_{n+1}| \leq (1 + hL)|d_n| + h\tau.$$

Iterating the bound yields

$$\begin{aligned} |d_n| &\leq (1+hL)^n |d_0| + h\tau \sum_{j=0}^{n-1} (1+hL)^j \\ &= (1+hL)^n |d_0| + h\tau \frac{(1+hL)^n - 1}{hL}. \end{aligned}$$

The $1 + hL$ is inconvenient. It would be better as an exponential, which can be achieved using the bound $1 + x \leq e^x$:

$$|d_n| \leq e^{nhL}|d_0| + \frac{\tau}{L}(e^{nhL} - 1).$$

Finally, the bound on the right must be **independent of** n (and h). But $h = b/n$ so $nh = b$ is independent of n and h, yielding the zero stability condition

$$|d_n| \leq e^{bL}|d_0| + \frac{e^{bL} - 1}{L}\tau.$$

This condition plus consistency, by the theorem, implies convergence.

In practice: In particular, if v_n is the approximation to the ODE (u_n) starting with $u_0 \approx y_0$

and $w_n = y(t_n)$ and u_n is computed with no rounding error (just truncation error) then τ is the maximum of the local truncation errors τ_n so the bound reads

$$|u_n - y_n| \leq e^{bL}|u_0 - y_0| + \frac{e^{bL}-1}{L} \max_{0\leq j\leq N-1} |\tau_j|.$$

For Euler's method,

$$|\tau_j| \leq \frac{h}{2} \max_{t\in[0,b]} |y''(t)| = O(h),$$

confirming that it is a first-order method (note that the error from the initial data is typically small, so the truncation error is the most significant part).

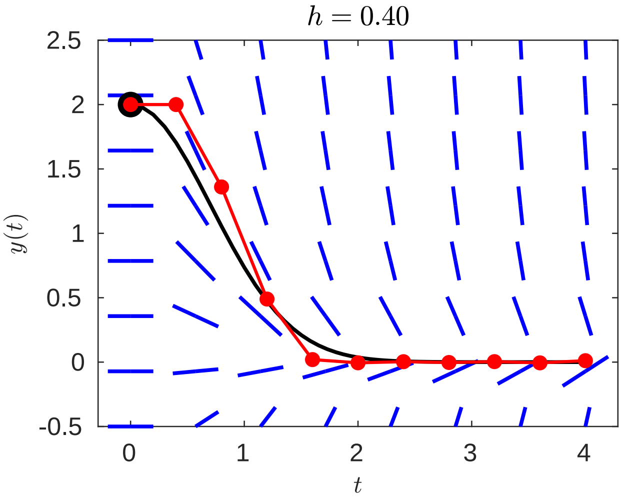
The bad news: The bound has the significant disadvantage that it grows exponentially in b, even if the ODE does not do the same. For example, consider

$$y' = -2ty, \quad y(0) = 1, \qquad t \in [0,4] \tag{2.1}$$

with exact solution $y(t) = e^{-t^2}$. Suppose Euler's method is used to approximate the solution, with $h = 0.1$. The Lipschitz constant in $[0,4]$ is 8, so the error bound says that

$$\text{max. error} \leq e^{32}(\cdots)$$

which is useless. The actual error is much better, as shown below. The issue here is that the Lipschitz bound does not distinguish between + and − exponentials, so the abs. value turns a decaying exponential into a growing one.



Above: solution (black) and Euler's method approximation (red) for (2.1).

2.3 A few more methods

Before proceeding, we pause to derive a few more methods useful for comparison.

The backward Euler method: From the scalar ODE

$$y' = f(t,y),$$

approximate y' at t_{n+1} by a backward difference (using t_n, t_{n+1}) to get

$$\frac{y(t_{n+1}) - y(t_n)}{h} = y'(t_{n+1}) + \frac{h}{2}y''(\xi_{n+1})$$

which yields the **Backward Euler formula**

$$u_{n+1} = u_n + hf(t_{n+1}, u_{n+1}).$$

The local truncation error is $O(h)$ so the formula is first order. However, it is **implicit**, meaning that the difference equation for the 'next' value (u_{n+1}) involves both previous known values (u_n) **and itself**. Thus, each step requires solving a non-linear equation of the form

$$0 = g(z) := z - u_n - hf(t_{n+1}, z)$$

for z, which requires a zero-finding routine (to be revisited!).

The trapezoidal method: Another approach is to integrate the ODE and use the fundamental theorem of calculus to obtain

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(s, y(s))\, ds.$$

The integral can then be estimated by some integration formula. Applying the trapezoidal rule in $[t_n, t_n + h]$ and letting $f_n = f(t_n, y_n)$, we get

$$y_{n+1} = y_n + \frac{1}{2}(f_n + f_{n+1}) - \frac{h^3}{12}y'''(\xi_{n+1}).$$

This yields the **trapezoidal method**

$$u_{n+1} = u_n + \frac{1}{2}(f_n + f_{n+1}).$$

Recalling the divide by h convention, the formula has the form

$$y_{n+1} = y_n + \frac{1}{2}(f_n + f_{n+1}) + h\tau_{n+1}, \qquad \tau_{n+1} := -\frac{h^2}{12}y'''(\xi_{n+1})$$

so the method should be second-order (to show convergence, one also has to prove zero stability; the proof is similar to Euler's method).

As with Backward Euler, the trapezoidal method is implicit.

3 Absolute stability and stifness

Consider the system of ODEs

$$\mathbf{y}' = F(t, \mathbf{y}).$$

The limitations of the error bound, plus the cumbersome definition of zero stability, suggest another approach is needed to better understand numerical methods.

One property of interest is that the method should ‘respect’ the behavior of solutions. Recalling the definitions of stability from subsection 1.3, we may want to have that

$$y(t) \text{ is a stable solution for the ODE } \implies \text{FDE for computing } y(t) \text{ is stable.}$$

Assessing this property is difficult for an ODE in general, as the ODE may be complicated. Instead, we can **approximate** and study a simpler ODE that captures the same behavior.

3.1 Review: LCC systems

?? The solution to the linear-constant coefficient system (LCC)

$$\mathbf{y}' = Ay, \qquad A \text{ an } m \times m \text{ matrix}$$

is a linear combination of m basis solutions.

Nice case: If A has eigenvalues $\lambda_1, \cdots, \lambda_m$ and a basis of eigenvectors $\mathbf{v}_1, \cdots, \mathbf{v}_m$ then the solutions $e^{\lambda_j}\mathbf{v}_j$ form this basis; the solution is

$$\mathbf{y} = \sum_{j=1}^{m} c_j e^{\lambda_j t}\mathbf{v}_j.$$

Notably, the system can be put in a more natural form by changing to the eigenvector basis. We have $A = VDV^{-1}$ where D is the diagonal matrix with entries $\lambda_1, \cdots, \lambda_m$ and V is the matrix whose columns are the eigenvectors.

Letting $\mathbf{x}(t) = V^{-1}\mathbf{y}(t)$ and plugging into the ODE gives

$$V\mathbf{x}'(t) = (VDV^{-1})V\mathbf{x}(t) \implies \mathbf{x}'(t) = D\mathbf{x}(t).$$

Thus, in the eigenvector basis, the components evolve independently, according to

$$x_j'(t) = \lambda_j x_j(t), \quad j = 1, \cdots m.$$

Less nice case: When there repeated eigenvalues and not enough eigenvectors to span $\mathbb{R}^m$, one instead gets solutions that involve $t^k e^{\lambda t}$. For details, see any ODE textbook.

The point (Long-term behavior): Let A be an $m \times m$ matrix with eigenvalues $\lambda_1, \cdots, \lambda_m$ and consider solutions to

$$\mathbf{y}'(t) = A\mathbf{y}(t).$$

Then all solutions involve only functions like $t^k e^{\lambda_j t}$ and

$$\|\mathbf{y}(t)\| \to \begin{cases} 0 & \text{Re}(\lambda_j) < 0 \text{ for all } j \\ \infty & \text{Re}(\lambda_j) > 0 \text{ for some } j \end{cases}. \tag{3.1}$$

The largest real part determines the growth/decay rate of the solution.
The marginal cases are more subtle (left as an exercise).

3.2 (ODE) Linear stability

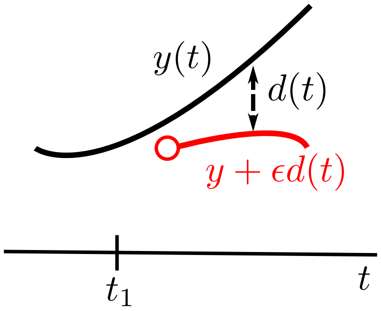
First, let's see what simple approximation gives us information about the stability of a general ODE. Consider the (non-linear) system

$$\mathbf{y}'(t) = F(t, \mathbf{y}(t)).$$

Let $y(t)$ be the 'base' solution and suppose the perturbed solution is

$$\mathbf{v}(t) = \mathbf{y}(t) + \epsilon \mathbf{d}(t)$$

which starts near a time t_1.



Plug this expression into the ODE:

$$\mathbf{y}'(t) + \epsilon \mathbf{d}'(t) = F(t, \mathbf{y}(t) + \epsilon \mathbf{d}(t)).$$

Now use a Taylor expansion around a point t_1:

$$\mathbf{y}'(t) + \epsilon \mathbf{d}'(t) = F(t, \mathbf{y}(t)) + \epsilon J \mathbf{d}(t) + O(\epsilon^2).$$

where J is the Jacobian of F at $(t_1, \mathbf{y}(t_1))$. Dropping $O(\epsilon^2)$ yields the **linearized** system

$$\mathbf{d}'(t) \approx J \mathbf{d}(t) \tag{3.2}$$

Whether the perturbation decays or grows then depends on the eigenvalues of J according to (3.1). In particular, we have that

- Perturbations grow (unstable) if some λ has a positive real part
- Perturbations decay to zero (asymptotically stable) if all λ's have negative real parts

For a scalar first-order ODE, the analysis is simpler; for

$$y' = f(t, y)$$

and a perturbation $v(t) + \epsilon d(t)$ around t_1 we get

$$d'(t) \approx \lambda d(t), \qquad \lambda = \frac{\partial f}{\partial y}(t_1, y_1).$$

Thus, perturbations grow/decay depending on the sign of $\partial f / \partial y$.

3.3 (Numerical method) Absolute stability

We can examine the stability of the numerical method in the same way, by studying **exactly** how it behaves on this simpler system, the scalar **test equation**

$$y' = \lambda y, \quad y(0) = y_0, \qquad \lambda \in \mathbb{C}$$

and the sequence $\{u_n\}$ generated by a numerical method. The solution is $y(t) = y_0 e^{\lambda t}$ so

- If $\mathrm{Re}(\lambda) < 0$ then $y(t) \to 0$ as $t \to \infty$ (asymptotic stability)
- If $\mathrm{Re}(\lambda) > 0$ then $y(t)$ grows exponentially. (instability)

Ideally, the numerical method should also be 'asymptotically' stable' when the ODE has this property - if the solution decays to zero, so should the approximation. Precisely,

$$\mathrm{Re}(\lambda) < 0 \implies u_n \to 0 \text{ as } n \to \infty.$$

Most importantly, it should not be unstable (u_n grows exponentially)!

For Euler's method: The test equation is simple enough that the FDE for numerical methods can be analyzed directly. For Euler's method,

$$u_{n+1} = u_n = hf(t_n, u_n)$$

we get

$$u_{n+1} = (1 + h\lambda)u_n \implies u_n = (1 + h\lambda)^n u_0.$$

It follows that, as $n \to \infty$,

- $u_n \to 0$ if $-2 < h\lambda < 0$
- $|u_n| \to \infty$ if $h\lambda < -2$ or $h\lambda > 0$

The behavior depends on the complex number $z = h\lambda$ (which includes h), so it may differ from the behavior of the ODE.

Definition: The **region of absolute stability** R (in the complex plane) is the set of complex numbers $z = h\lambda$ such that

$$|u_n| \to 0 \text{ as } n \to \infty \tag{AS}$$

where $\{u_n\}$ is the approximation with step size h applied to the test equation $y' = \lambda y$.

Note that even though the $\mathrm{Re}(z) < 0$ case is most important, the region is also defined for positive real part.

The **interval of absolute stability** is the part of R on the real axis, i.e. the set of real numbers $h\lambda$ such that the property holds.

For Euler's method, the property (AS) holds if and only if

$$|1 + h\lambda| < 1.$$

It follows that the region of absolute stability is

$$R = \{z \in \mathbb{C} : |1 + z| < 1\}.$$

This is a circle of radius 1 centered at $z = -1$.

What does the region tell us? Suppose $\lambda < 0$ is real, such as

$$y' = -20y.$$

The absolute stability condition says that

$$|u_n| \to 0 \text{ if and only if } h < \frac{2}{|\lambda|}. \tag{3.3}$$

Moreover, $|u_n|$ will **grow exponentially** (unstable!) if $h \geq 2/|\lambda|$. It follows that to get a reasonable solution at all, we need to take $h < 2/20 = 1/10$.

Compare the result here to **convergence**, which states that the error is $O(h)$ when h is small enough. It says nothing about **how small** h must be to have an error that decreases nicely. Thus, the dramatic failure for $h > 1/10$ does not violate the convergence theorem.

3.4 Stiffness

The effect of the step size is shown in Figure 3.4 for

$$y' = -20(y - \sin t) + \cos t, \quad y(0) = 1 \tag{3.4}$$

which has the solution

$$y(t) = e^{-20t} + \sin t.$$

At early times there is a rapidly decaying transient; then it settles down and looks like $\sin t$. A reasonable approximation needs to have a small enough h to resolve the transient, up to about $t = 1/10$.

(Informal) definition: An **accuracy constraint** is a restriction on the step size h to have a numerical solution that has the same qualitative features as the true solution.

Typically, such constraints arise if the solution changes rapidly or oscillates.

However, past the initial transient, the solution varies more slowly, so we would like to use a larger step size. Since

$$\frac{\partial f}{\partial y} = -20,$$

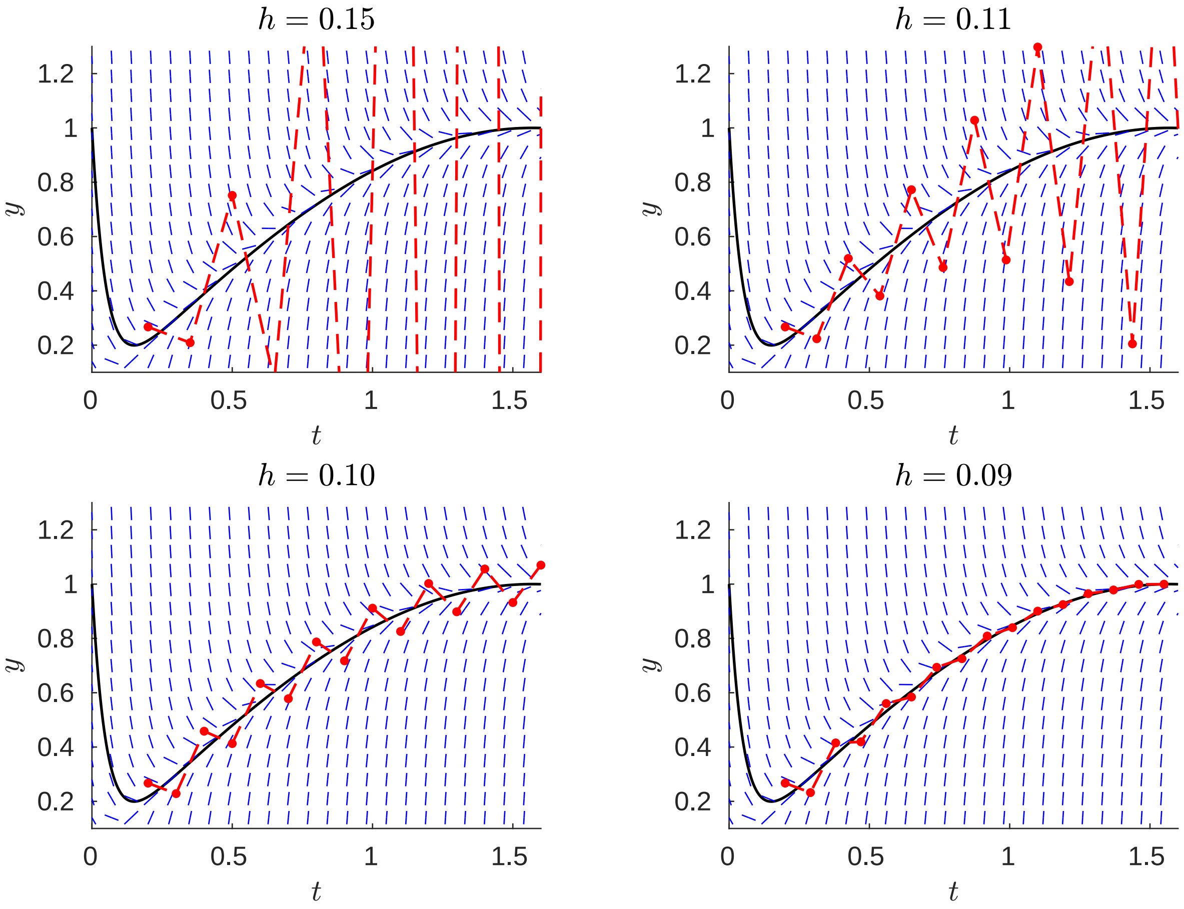


Figure 1: Euler's method applied to the stiff IVP (3.4) and the slope field (blue lines).

Euler's method requires that $h < 1/10$ to be stable. For small t (around $t < 1/10$), this requirement is not a problem because h must be this small to be accurate anyway (the **accuracy** condition is the limiting factor for the step size).

However, when t is larger, the accuracy constraint is much less than the stability constraint, and we must take $h < 1/10$ even though the solution $y(t) = \sin(t)$ does not require a step size that small to be reoslved.

One can think of the issue as follows: because of numerical error, the numerical method must be able to resolve not only the true solution, but **also nearby solutions**. In this case, nearby solutions $y(t) = \sin t + \epsilon d(t)$ satisfy

$$d'(t) = -20d(t)$$

i.e. they have a rapidly decaying behavior that requires a small step size to resolve. A system for which the stability constraint is an issue is called **stiff**. Formal definitions vary; what matters is the idea and its practical consequences.

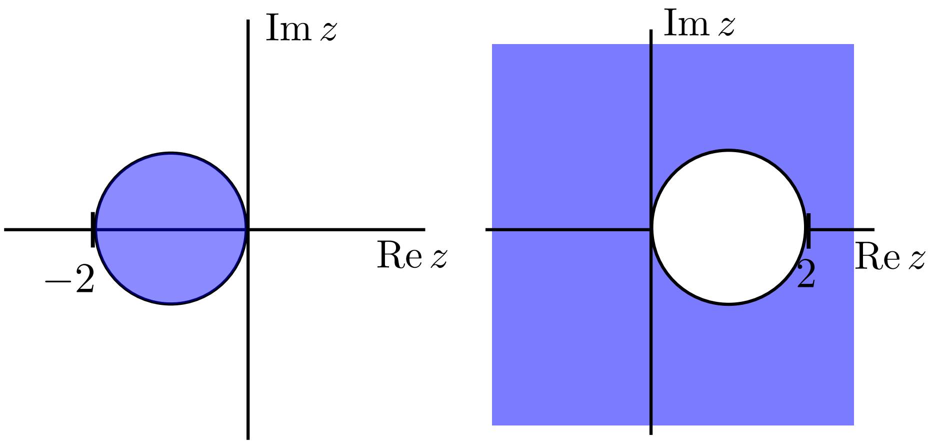


Figure 2: Sketches of the stability regions (shaded area) for Euler's method (left) and Backward Euler (right) in the complex plane.

Stiffness (definition I): An IVP in some interval $[a, b]$ is called **stiff** if Euler's method (or more generally, an explicit method) requires a much smaller time step h to be stable than it does to be accurate.

We say that the ODE (3.4) is **stiff** in $[0, 1.5]$ but not stiff in $[0, 0.1]$. The practical question is: can a method like Euler's method (i.e. any method with a significant absolute stability constraint) be used efficiently?

Stiffness (for first-order ODEs, definition II):) A first order ODE is stiff if $|\frac{\partial f}{\partial y}|$ is large in the sense that $1/|\frac{\partial f}{\partial y}|$ is much less than the typical width of a 'change' in $y(t)$.

Geometrically, this typically means that nearby trajectories to $y(t)$ converge rapidly to $y(t)$ compared to the variation in $y(t)$ itself.

3.5 Backward Euler

For a stiff system, the best method should be one that has no stability constraint for $\text{Re}(\lambda) < 0$. Recall that the **backward Euler method** is

$$u_{n+1} = u_n + hf(t_{n+1}, u_{n+1})$$

Applying the method to the test equation, we get

$$u_{n+1} = \frac{1}{(1 - h\lambda)} u_n.$$

Thus $u_n \to 0$ as $n \to \infty$ if and only if $|1 - h\lambda| > 1$. The region of absolute stability is

$$R = \{z \in \mathbb{C} : |1 - z| > 1\}$$

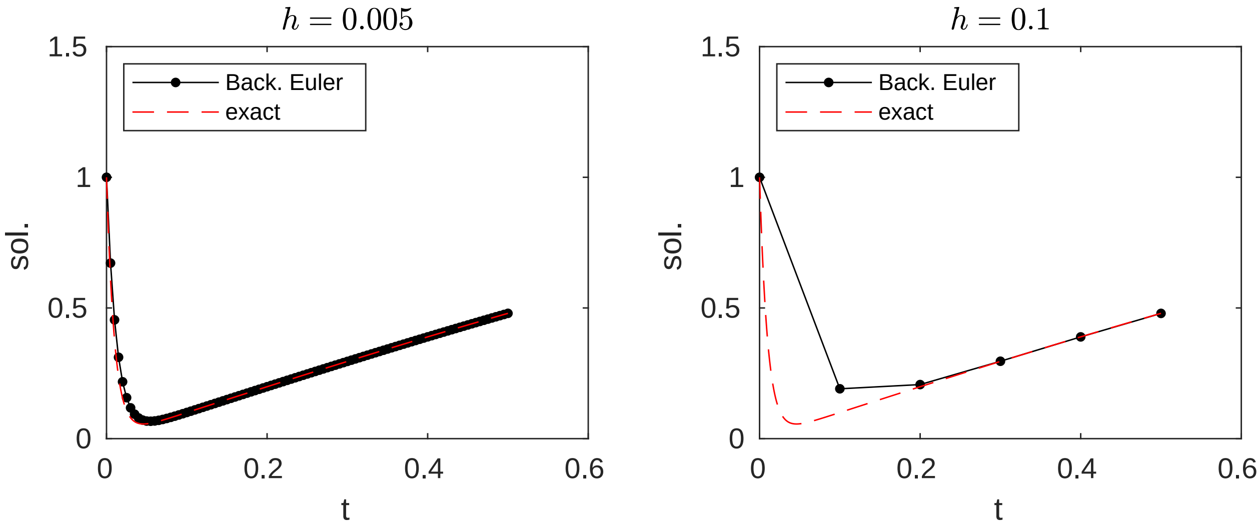
which is the complement of a circle centered at $z = 1$ of radius 1 (see Figure 2). In particular, R contains the entire half plane $\{z < 0\}$, which means that if $\text{Re}\lambda < 0$ then $|y_n| \to 0$ for any positive value of h.

Defintion: A method for which R contains all of the half-plane $\{\text{Re}(z) < 0\}$ is called **A-stable**. An A-stable method has no stability constraint when solving a stiff equation.

The backward Euler method is A-stable. Indeed, if it is used on the example problem, the approximation will always be reasonable, even when $h > 1/10$. Below, Backwards Euler is used to solve

$$y' = -100(y - \sin t) + \cos t, \quad y(0) = 1. \tag{3.5}$$

for $t \in [0, 0.5]$. Backwards Euler does fine in the stiff interval, but Euler's method would require $h < 1/50$. For the initial transient, $h = 0.1$ gives a poor approximation, but this is expected since the step size is so large it skips over that feature (the approximation is still stable, and fine after the transient is gone!).



3.6 Stiffness for systems

By the analysis for the system $\mathbf{y}' = F(t, \mathbf{y})$ (see (3.2)) we find that the appropriate test equation is the linear constant coefficient system

$$\mathbf{y}'(t) = A\mathbf{y}, \qquad A \in \mathbb{R}^{n\times n}.$$

For simplicity, suppose A is diagonalizable, with eigenvalues $\lambda_1, \cdots, \lambda_n$. From ODE theory, this system is equivalent to

$$\mathbf{x}'(t) = D\mathbf{x}$$

where $\mathbf{x} = V^{-1}\mathbf{y}$ and $D = \text{diag}(\lambda_1, \cdots, \lambda_n)$ and $A = VDV^{-1}$. That is, each component in the eigenvector basis evolves via the scalar equation

$$x_j' = \lambda_j x_j.$$

Thus $\mathbf{y} \to 0$ if and only if $\mathrm{Re}(\lambda_j) < 0$ for all j. The same holds for the non-diagonalizable case, except solutions may look like $t^k e^{\lambda t}$ instead of $e^{\lambda t}$.

Now suppose Euler's method is applied to the test equation. Then

$$\mathbf{u}_{n+1} = (I + hA)\mathbf{u}_n.$$

Diagonalizing by taking $\mathbf{v} = V^{-1}\mathbf{u}$ with components $\mathbf{v}^{(i)}$, we get

$$\mathbf{v}_{n+1} = (I + hD)\mathbf{v}_n \implies \mathbf{v}_{n+1}^{(i)} = (I + h\lambda_i)\mathbf{v}_n^{(i)}.$$

Thus, each component evolves like the scalar test equation. Let $R = \{z \in \mathbb{C} : |1+z| < 1\}$ be the stability region derived before. Applying the scalar result to each component, we obtain the stability condition

$$\mathbf{u}_n \to 0 \iff h\lambda_i \in R \text{ for } i = 1, \cdots n \tag{3.6}$$

since $h\lambda_i \in \mathbb{R}$ guarantees $\mathbf{v}_n^{(i)} \to 0$.

If all the λ's are real and the interval of absolute stability is $(-b, 0)$ (e.g. $(-2, 0)$ for Euler's method then the condition is simply

$$h < \frac{b}{\max_i |\lambda_i|}.$$

Example (real): Consider, for $b > 0$, the second order IVP

$$y'' + (b+1)y' + by = e^{2t}, \quad y(0) = a,\, y'(0) = b$$

which has solutions of the form

$$y(t) = Ce^{2t} + c_1 e^{-t} + c_2 e^{-bt}.$$

The ODE is equivalent to the system

$$x_1' = x_2, \quad x_2' = -bx_1 - (b+1)x_2 + e^{2t}$$

and the eigenvalues of the Jacobian $\partial F/\partial \mathbf{x}$ are $\lambda = -1$ and $\lambda = -b$. Both area real, so the stability constraint for Euler's method is

$$h < \frac{2}{\max\{1, b\}}.$$

That is, $h \cdot (-1)$ and $h \cdot (-b)$ must lie in the stability interval $(-2, 0)$.

Example (complex): Now consider the second order IVP

$$y'' + 2by' + (b^2 + \omega^2)y = e^{2t}, \quad y(0) = a,\, y'(0) = b$$

whose solution as the form

$$y(t) = e^{2t} + e^{-bt}(c_1 \sin \omega t + c_2 \cos \omega t).$$

The ODE is equivalent to the system

$$x_1' = x_2, \quad x_2' = -(b^2 + \omega^2)x_1 - 2bx_2 + e^{2t}$$

and the Jacobian $\partial F / \partial \mathbf{x}$ has eigenvalues

$$\lambda_1, \lambda_2 = -b \pm \omega i.$$

To apply Euler's method, we must have

$$|1 + h\lambda_i| < 1, \quad i = 1, 2.$$

The sets $\{h\lambda_i\}$ for $h > 0$ are rays extending out from zero in the complex plane. The method is stable for a given h if the points on these rays lie inside the stability region.

Compare to the real case, where both rays coincide with the negative real axis, so we only need to check if $h\lambda_i$ lies in the stability **interval** (along the real axis).

The non-zero ω makes the rays intersect with a smaller section of the circle, so there is a sharper stability constraint. We see, then, that a larger imaginary part means a sharper stability constraint, because the rays intersect with a smaller part of the circle. The significance of the 'imaginary' part of the stability region is that it tells us **how the method behaves on oscillating solutions**.

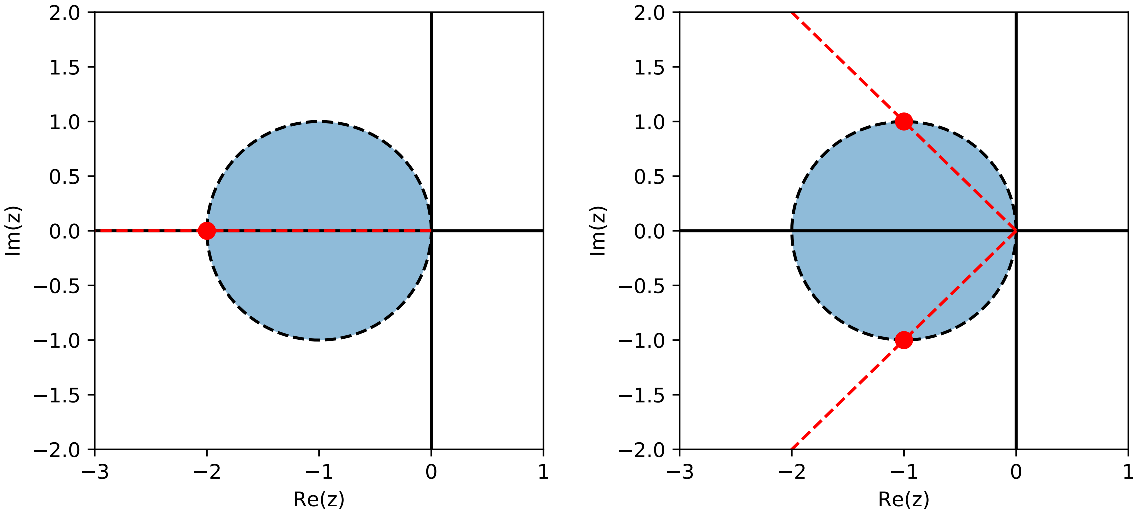


Figure 3: Stability region and rays $h\lambda$ for the two system examples. The constraint is more severe on the right, where λ has an imaginary part.

Definition (Stiffness ratio): Observe that for the test equation, if the solution goes to zero, its slowest component has a rate given by $\min|\text{Re}(\lambda_i)|$, but the stability constraint involves $\max|\text{Re}(\lambda)|$. One can then define the **stiffness ratio**

$$\sigma = \frac{\max_\lambda |\text{Re}(\lambda)|}{\min_\lambda |\text{Re}(\lambda)}$$

with the min/max taken over the eigenvalues of A. The value of σ measures, in some sense, the separation of time scales in the system (the ratio of rates between ‘fast’ and ’slow’ components), and tends to be large when a system has two widely separated time scales - such as a chemical reaction where the complete reaction is slow, but some component reacts very quickly at the start.

4 Multi-step methods

The Euler methods use only data from the previous step (t_n) to get to the next step (at t_{n+1}).

However, we also have access to the data from previous steps, which can be used to build methods with high accuracy. A **linear multi-step method** (with m steps) has the form

$$\frac{1}{h}\sum_{j=0}^{m} a_j u_{n-j} = \sum_{j=0}^{m} b_j f_{n-j} \tag{4.1}$$

where $f_n = f(t_n, u_n)$. We have already derived two examples:

$$\text{Euler's method:} \quad \frac{1}{h}(u_n - u_{n-1}) = f_{n-1},$$

$$\text{Trapezoidal rule:} \quad \frac{1}{h}(u_n - u_{n-1}) = \left(\frac{1}{2}f_n + \frac{1}{2}f_{n-1}\right).$$

There are several types of methods that can be derived.

4.1 Adams-Bashforth (explicit) methods

One popular class of methods are **Adams methods**, which start from the integrated form

$$y_n = y_{n-1} + \int_{t_{n-1}}^{t_n} f(s, y(s))\, ds.$$

For an **Adams-Bashforth** method, the integral is approximated with a Newton-Cotes formula using only the m **previous** steps at $t_{n-1}, \cdots, t_{n-m}$ (equivalently: the solution $y(t)$ is approximated in the integral using the polynomial interpolant through these points.)

With equally spaced points, the interpolant has an error $O(h^m)$, and the integral formula has no symmetry, so its error is $O(h^{m+1})$. This procedure then yields

$$\begin{aligned}\frac{y_n - y_{n-1}}{h} &= \frac{1}{h}\int_{t_{n-1}}^{t_n} f(s, y(s))\, ds \\ &= \frac{1}{h}\left(h\sum_{j=1}^{m} b_j f_{n-j} + O(h^{m+1})\right) \\ &= \sum_{j=1}^{m} b_j f_{n-j} + O(h^m).\end{aligned}$$

so the Adams-Bashforth formula is (with $f_{n-j} = f(t_{n-j}, u_{n-j})$)

$$u_n = u_{n-1} + h\sum_{j=1}^{m} b_j f_{n-j}.$$

Example ($m = 2$): One can use the standard methods for Newton-Cotes formulas to derive the method. However, we can use a shortcut by noting that $f(t, y(t)) = y'(t)$ is easy to expand in a Taylor series.

The desired formula has the form

$$\frac{1}{h}(y_n - y_{n-1}) = ay'_{n-1} + by'_{n-2} + \text{error}. \tag{4.2}$$

There are two points, so we expect an $O(h^2)$ error (the choice of a and b means that the first two terms, $O(1)$ and $O(h)$, in the error can be made to cancel, leaving $O(h^2)$).

First, expand the LHS of (4.2) in a Taylor series around t_n up to $O(h^2)$:

$$\text{LHS} = \frac{1}{h}(y_n - y_{n-1}) = y'_n - \frac{h}{2}y''_n + O(h^2).$$

Next, expand the formula in the RHS of (4.2) in the same way:

$$\text{RHS} = a(y'_n - hy''_n) + b(y'_n - 2hy''_n) + O(h^2).$$

Matching the two, we get $a + b = 1$ and $a + 2b = 1/2$ so

$$\frac{1}{h}(y_n - y_{n-1}) = \left(\frac{3}{2}y'_{n-1} - \frac{1}{2}y'_{n-2}\right) + O(h^2).$$

The difference equation for the method and local truncation error are then

$$u_n = u_{n-1} + h\left(\frac{3}{2}f_{n-1} - \frac{1}{2}f_{n-2}\right), \qquad \tau_n = O(h^2). \tag{4.3}$$

The order is one better than Euler's method.

Example ($m = 3$): The derivation is left as an exercise. The result is the formula/LTE

$$u_n = u_{n-1} + \frac{h}{12}\left(23f_{n-1} - 16f_{n-2} + 5f_{n-3}\right), \quad \tau_n = O(h^3).$$

4.2 Efficiency, starting values

This approach is popular because the formula is efficient to compute. To go from step n to $n+1$, only **one function evaluation** (at f_n) is required, because the function values at the previous steps have already been computed.

In implementation, one keeps track of the values $(f_{n-m}, f_{n-m+1} \cdots, f_{n-1})$ and updates the array from n to $n + 1$ by shifting each value left by one and adding the new value.

For instance, the AB2 method (4.3) requires two variables `fn` and `fn1` storing f_{n-1} and f_{n-2}, and at each step, `fn1` is overwritten with `fn` before updating `fn`

Starting values: A multistep method has the disadvantage that it requires more than one starting value - even though the ODE only requires one. This means that the first $m-1$ steps (for $u_1, \cdots, u_{m-1}$) must be computed by **some other method**. A good one step method (e.g. a Runge-Kutta method, to be covered) is typically used.

Note that the order requirements are less for a starter method. For the AB2 method (4.3),

$$u_n = u_{n-1} + h\left(\frac{3}{2}f_{n-1} - \frac{1}{2}f_{n-2}\right),$$

the starting values must be computed with an error of at most $O(h^2)$, recalling that errors in initial conditions lead to bounds like

$$\text{global error} \leq e^{Lb}|u_0 - y_0| + \cdots.$$

One step of Euler's method yields an $O(h^2)$ error (even though it is first order), since it is only taking a step of size h. Thus, Euler's method is enough to produce u_1 for (4.3).

4.3 Adams-Moulton (implicit) methods

When t_{n+1} is used in the in approximating the integral

$$\int_{t_{n-1}}^{t_n} f(s, y(s))\, ds$$

The resulting formula is implicit, and called an **Adams-Moulton method**. For example, the $m = 1$ Adams Moulton method is the trapezoidal rule:

$$u_n = u_{n-1} + \frac{h}{2}(f_n + f_{n-1}).$$

The formulas are derived in the same way as the Adams-Bashforth method, but the points $t_n, t_{n-1}, \cdots t_{n-m}$ are used:

$$\begin{aligned}\frac{1}{h}(y_n - y_{n-1}) &= \frac{1}{h}\int_{t_{n-1}}^{t_n} f(s, y(s))\, ds \\ &= \sum_{j=0}^{m} b_j f_{n-j} + O(h^{m+1}).\end{aligned}$$

Note that there are $m+1$ points in the approximation, so the local error is $O(h^{m+2})$ from the integral approximation; the order of the method is $m+1$.

Example ($m = 2$): The method has the form

$$\frac{1}{h}(y_n - y_{n-1}) = ay'_n + by'_{n-1} + cy'_{n-2} + \text{error}$$

With three points, we expect the LTE to be $O(h^3)$. Expanding the left hand side:

$$\text{LHS} = \frac{1}{h}(y_n - y_{n-1}) = y_n' - \frac{h}{2}y_n'' + \frac{h^2}{6}y_n''' + O(h^3).$$

Expanding the RHS gives

$$\begin{aligned}\text{RHS} &= ay_n' + b(y_n' - hy_n'' + \frac{h^2}{2}y_n''') + c(y_n' - 2hy_n'' + 2h^2y_n''') + O(h^3)\\ &= (a+b+c)y_n' - h(b+2c)y_n'' + \frac{h^2}{2}(b+4c)y_n''' + O(h^3).\end{aligned}$$

Matching the LHS and RHS yields the system

$$a+b+c = 1, \quad b+2c = \frac{1}{2}, \quad b+4c = \frac{1}{6}.$$

After solving the system, the resulting formula is

$$\frac{1}{h}(y_n - y_{n-1}) = \frac{1}{12}\left(5y_n' + 8y_{n-1}' - y_{n-2}'\right) + O(h^3),$$

which has order 3 (and is implicit). Some recycling of function evaluations of f can be done at each step, but one also needs to solve the implicit equation for y_n at each step.

4.4 Backward differentiation formulas

A third class of linear multi-step methods are the **backward differentiation formulas** (BDFs), which instead approximate y' in

$$y' = f(t, y)$$

using the points $t_n, t_{n-1}, \cdots t_{n-m}$ (a backwards difference) and the RHS as f_n. With equally spaced points, the result is a formula of the form

$$\frac{1}{h}\sum_{j=0}^{m} a_j y_{n-j} = f_n + \tau_n, \qquad \tau_n = O(h^m).$$

Since $m+1$ points are used, the derivative approximation (the LHS) has an $O(h^m)$ error, so the method has order m.

When $m = 1$, the method is just Backward Euler:

$$\frac{1}{h}(u_n - u_{n-1}) = f_n.$$

When $m = 2$, the method (BDF-2, or sometimes **Gears' method**) is

$$\frac{1}{2h}(4u_n - 3u_{n-1} + u_{n-2}) = f_n. \tag{4.4}$$

5 Stability for multi-step methods

To determine stability, it is necessary to know how to solve the difference equations for multi-step methods. Thankfully, one only has to do so in an easy special case.

The main result is the following:

Dahlquist equivalence theorem: A linear multistep method

$$\frac{1}{h}\sum_{j=0}^{m} a_j u_{n-j} = \sum_{j=0}^{m} b_j f_{n-j}$$

is zero stable if and only if all solutions u_n stay bounded as $n \to \infty$ when the method is applied to the **trivial ODE**

$$y' = 0.$$

That is, it is zero stable if and only if all solutions to

$$\sum_{j=0}^{m} a_j u_{n-j} = 0$$

stay bounded as $n \to \infty$.

To prove zero-stability, we need to be able to solve difference equations of this form.

5.1 Difference equations

A **linear, constant coefficient difference equation** has the form

$$c_m u_n + c_{m-1}u_{n-1} + \cdots + c_0 u_{n-m} = 0 \tag{5.1}$$

with given initial values $a_0, \cdots, a_{m-1}$. he theory for an LCC difference equation (5.1) is analogous to that for an n-th order ODE, a review of which is given in the box below.

Review (n-th order LCC ODEs) For the m-th order LCC ODE,

$$c_m y^{(m)} + c_{m-1}y^{(m-1)} + \cdots + c_1 y' + c_0 y = 0$$

look for solutions $y(x) = e^{rx}$; plug in to find that r must solve

$$0 = c_m r^m + \cdots + c_1 r + c_0 := p(r)$$

i.e. a zero of the characteristic polynomial $p(r)$.

A root repeated k times yields the k solutions

$$e^{rx}, xe^{rx}, \cdots x^{k-1}e^{rx}.$$

The full solution is then a linear combinations of the solutions above for each zero r of $p(r)$.

For an LCC difference equation, the general solution is a linear combination of m 'basis' solutions (due to linearity). First, look for solutions of the form

$$a_n = r^n.$$

Plug into the difference equation to find that r^n is a solution if

$$0 = c_m r^m + c_{m-1} r^{m-1} + \cdots + c_1 r + c_0 := p(r)$$

where $p(r)$ is the **characteristic polynomial**.

For zeros repeated k times, it is straightforward (but tedious) to show there are solutions[2]

$$r^n, nr^n, n^2 r^n, \cdots n^{k-1} r^n.$$

Stability (LCC difference equations): It follows, in particular, that

- $a_n \to 0$ as $n \to \infty$ for all initial values if and only if $|r_j| < 1$ for all zeros of $p(r)$
- a_n stays bounded as $n \to \infty$ for all initial values if and only if $|r_j| \leq 1$ or all zeros of $p(r)$ and no zero with $|r| = 1$ is repeated.

5.2 Convergence for multistep methods

It follows that convergence for a linear multi-step method can be established by proving consistency and then checking stability for the trivial ODE. One can also derive a global error bound similar to Euler's method, leading to the main theorem:

Theorem (convergence for multistep methods) A linear multistep method (4.1) is convergent with order p if and only if it is consistent with order p and all solutions to

$$\sum_{j=0}^{m} a_j u_{n-j} = 0$$

remain bounded as $n \to \infty$. The error in $[0, b]$ satisfies a bound of the form

$$\max_{1 \leq n \leq N} |u_n - y(t_n)| \leq K \left(\max_{0 \leq j \leq m-1} |u_j - y(t_j)| + \max_{1 \leq n \leq N} |\tau_n| \right).$$

[2] For the proof, let $\vec{u}_n = (u_n, \cdots, u_{n-m})$ and write the difference equation as $\vec{u}_{n+1} = A\vec{u}$ in matrix form; then A has a Jordan decomposition $A = VJV^{-1}$ and $\vec{u}$ can be obtained from J^n.

5.3 Example of an unstable method

Obviously, any method we propose should be consistent (i.e. the truncation error is small enough). As the theorem asserts, consistency is not enough for convergence! A simple example illustrates what can go wrong when a method is consistent but not stable.

Euler's method can be derived by replacing y' in the ODE with a forward difference:

$$\frac{y(t+h)-y(t)}{h} = y' = f(t,y).$$

One might hope, then, that a more accurate method can be obtained by using a second-order forward difference

$$y'(t) = \frac{-y(t+2h)+4y(t+h)-3y(t)}{2h} + O(h^2).$$

Plugging this in, we obtain the method

$$\frac{-u_{j+2}+4u_{j+1}-3u_j}{2h} = f(t_j,u_j) \tag{5.2}$$

which is consistent with an $O(h^2)$ LTE. However, this method is not zero stable!

For the trivial ODE $y'=0$, the iteration reduces to

$$u_{j+2} = 4u_{j+1} - 3u_j.$$

Plugging in $u_j = r^j$ we get a solution when

$$r^2 - 4r + 3 = 0 \implies r = 1,\, 3$$

so the general solution is

$$u_j = a + b\cdot 3^j.$$

If initial values are chosen so that

$$u_0 = u_1$$

then $y_j = y_0$ for all j with exact arithmetic. However, if there are any errors ($u_0 \neq u_1$) then $b \neq 0$ and $|u_j|$ willl grow exponentially. Thus, the method is unstable, and is not convergent.

Compare this result to the **backward differentiation formula** (4.4),

$$\frac{3u_n - 4u_{n-1} + u_{n-2}}{2h} = f(t_n,u_n).$$

Applied to $y'=0$, the difference equation reduces to

$$3u_n - 4u_{n-1} + u_{n-2} = 0$$

which has characteristic polynomial/roots

$$p(r) = 3r^2 - 4r + 1 \implies r = 1, 1/3.$$

The root $r=1$ is not repeated and the others have $|r|<1$, so solutions are bounded. By the theorem, the method is zero-stable so it is convergent.

6 Absolute stability for multistep methods

As before, the stability region R is the set of $z = h\lambda$ such that the result $\{u_n\}$ of applying the method to $y' = \lambda y$ has $u_n \to 0$ for any initial condition. We consider the method

$$\frac{1}{h}\sum_{j=0}^{m} a_j u_{n-j} = \sum_{j=0}^{m} b_j f_{n-j}.$$

Applying this method to the test equation yields the difference equation

$$\sum_{j=0}^{m} a_j u_{n-j} = z\sum_{j=0}^{m} b_j u_{n-j}, \qquad z = h\lambda. \tag{6.1}$$

The associated characteristic polynomial ψ for (6.1) has the form

$$\psi(r; z) = p(r) - zq(r), \qquad \text{with } p(r) = \sum_{j=0}^{m} a_j r^{m-j}, \quad q(r) = \sum_{j=0}^{m} b_j r^{m-j}. \tag{6.2}$$

Note that $p(r)$ is the same characteristic polynomial used for zero-stability (the $z = 0$ case). It follows, by applying the theory for difference equations, that

$$u_n \to 0 \text{ for all ICs} \iff |r_j| < 1 \text{ for all zeros } r_j \text{ of } \psi(r; z)$$

Then z is in the stability region R if the above holds for that value of z.
The algebra involved here is not convenient, because it is difficult to solve for the r_j's given a value of z. The more clever approach is to 'reverse' the order, since z is easy to find given r.

To start, we observe that:

- z is inside the stability region R if $|r_j| < 1$ for all zeros of ψ
- z is **outside** the stability region R if $|r_j| > 1$ for some zero of ψ

This leaves one last case:

(**) z is on the **boundary** of R if $|r_j| \le 1$ but some zero has magnitude exactly one.

Thus, we can find the boundary of R by looking for z-values corresponding to r's of magnitude one. To do so, suppose $r = e^{i\theta}$ is a zero and plug into the characteristic polynomial (6.2):

$$0 = \psi(e^{i\theta}; z) = p(e^{i\theta}) - zq(e^{i\theta}).$$

There is exactly one value of z such that this holds. Denoting it by $z^*(\theta)$, we obtain

$$z^*(\theta) = p(e^{i\theta})/q(e^{i\theta}), \qquad \theta \in [0, 2\pi].$$

It follows from condition (**) that the boundary of R is contained in the curve $z^*(\theta)$.

The last step is to identify which side is 'inside', which is straightforward to do by testing a value of z and finding the zeros of $\psi(r; z)$ directly, or checking a limit like $|z| \to \infty$.

Example 1 (BDF-2): We find the absolute stability region for the BDF-2 method

$$3u_n - 4u_{n-1} + u_{n-2} = 2hf_n.$$

Applying the method to the test equation gives (with $z = h\lambda$)

$$3u_n - 4u_{n-1} + u_{n-2} = 2zu_n.$$

Then z is in the region of absolute stability R if

$$\psi(r; z) = (3r^2 - 4r + 1) - 2zr^2 \text{ has zeros } r_1, r_2 \text{ with } |r_1|, |r_2| < 1.$$

If z is on the boundary of R, then ψ must have a zero of magnitude exactly one. Looking for values of z such that $r = e^{i\theta}$ is a zero, we obtain the solution

$$z^*(\theta) = \frac{3r^2 - 4r + 1}{2r^2}\Big|_{r=e^{i\theta}} = \frac{3}{2} - 2e^{-i\theta} + \frac{1}{2}e^{-2i\theta}.$$

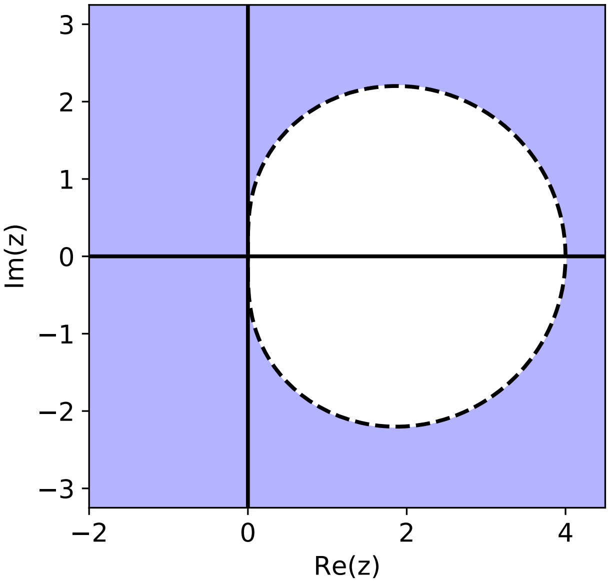
The stability region is shown below. We must check whether the inside or outside is R. By continuity, it suffices to check one point, the easiest of which is a point like $z = -1$:

$$\psi(r; -1) = 5r^2 - 4r + 1 \implies r = \frac{2 \pm \sqrt{4-5}}{5} = \frac{2}{5} \pm \frac{i}{5}$$

for which $|r| = 1/\sqrt{5} < 1$. From $-1 \in R$ and the plot, we conclude the outside of the curve is R. It can be checked that

$$\mathrm{Re}(z^*(\theta)) \geq 0 \text{ for all } \theta.$$

Thus the region is the complement of a blob in the right half-plane, so the method is A-stable.



Example 2 (Adams-Moulton): We find the stability region for

$$u_n = u_{n-1} + \frac{h}{12}\left(5f_n + 8f_{n-1} - f_{n-2}\right).$$

Plugging in the test equation,

$$u_n = u_{n-1} + \frac{z}{12}\left(5u_n + 8u_{n-1} - u_{n-2}\right)$$

which has the characteristic polynomial

$$\psi(r;z) = r^2 - r - \frac{z}{12}\left(5r^2 + 8r - 1\right).$$

If z is in the boundary of the stability region R then ψ has a zero $r = e^{i\theta}$ of magnitude one. Plugging this in and solving for z yields

$$z^*(\theta) = \frac{12(e^{2i\theta} - e^{i\theta})}{5e^{2i\theta} + 8e^{i\theta} - 1}$$

which gives the boundary of the stability region. The region is **inside** the boundary, which can be checked in several ways. One is to note that

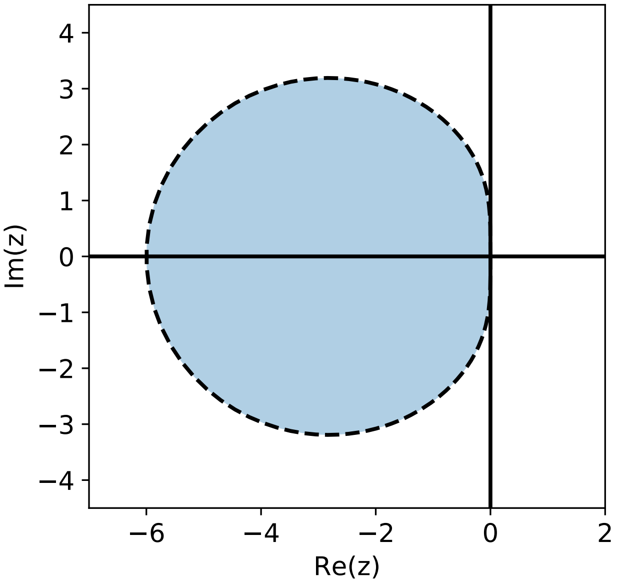
$$\text{as } z \to -\infty,\ \psi \approx -\frac{z}{12}(5r^2 + 8r - 1) \implies r \approx \frac{-8 \pm \sqrt{8^2 + 4\cdot 5}}{2\cdot 5}$$

one of which has magnitude > 1.

Thus,, the stability region is a bounded blob. It is useful to know the interval of absolute stability. This is easily found by noting that

$$z^*(\pi) = \frac{12(1+1)}{5-8-1} = -6$$

so the blob extends along the real axis down to -6, so the interval is $(-6, 0)$. Thus, there is a significant stability constraint (compare to the trapezoidal method, which has one order less accuracy but is A-stable).



6.1 Comparing stability regions

The stability regions for the first few BDF (Backwards differentiation), AB (Adams-Bashforth) and AM (Adams-Moulton) methods are shown in Figures Figure 6.1 and Figure 6.1.

Some observations:

- Only the AM/BDF methods of order up to two are A-stable (the trapezoidal method, backward Euler and BDF-2).
- The higher-order BDF methods (from order 3 to order 6) are not A-stable due to the lobes along the imaginary axis, but the region of absolute stability does contain $(-\infty, 0)$.[3]
- The AM methods have bounded stability regions for order > 2, unlike the BDF methods. For both the AB and AM methods, the stability region shrinks as the order increases.

A central result on this topic says that the failure to be A-stable is typical:

General rule (explicit methods and stability): All (reasonable) explicit methods have bounded stability regions (they cannot contain all of $(-\infty, 0)$).

Theorem (Dahlquist Barrier): A linear multi-step method of order greater than 2 cannot be A-stable.

Practically, the barrier means that one pays an efficiency price for the good stability of an A-stable method. The small stability regions for high order AB/AM methods suggests that such methods are best when high accuracy is needed for **non-stiff** problems, where the stability constraint doesn't matter.

Note that the General Rule does **not** mean that all implicit methods are good for stiff problems, just that any good method has to be implicit. The Adams-Moulton methods for third-order and above, for instance, are not well-suited to stiff problems.

[3] Note: We do not consider any higher order BDF methods because they are not zero-stable for order 7 and above!

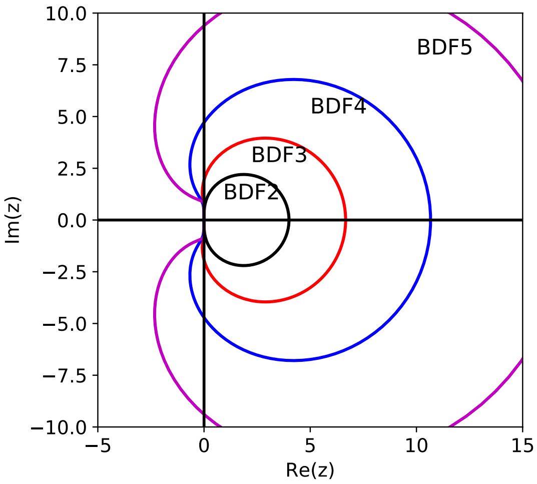


Figure 4: BDF stability regions for orders 2 to 5; the stability region is **outside** the curve. Note that only BDF-2 is A-stable.

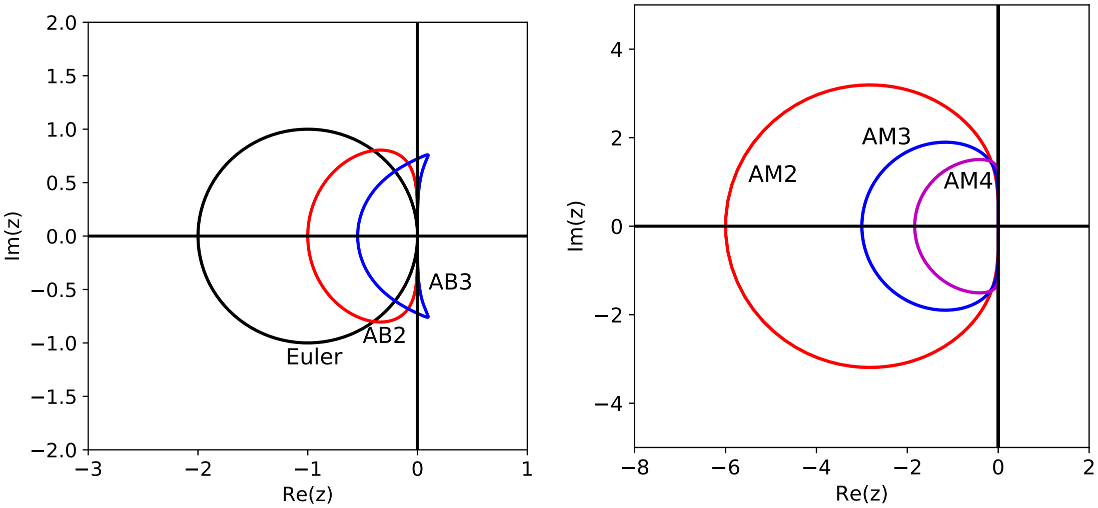


Figure 5: Stability regions for the Adams-Bashforth (AB) and Adams-Moulton (AM) methods with m previous steps. The region is **inside** the curve (Note that AB1 is Euler's method).