Berkeley
UNIVERSITY OF CALIFORNIA

Numerical Methods for Differential Equations

Notes By: Yuhang Cai

2023 Spring

CONTENTS

Part I

The Initial Value Problem

CHAPTER 1

IVP FOR ODE

In this chapter we begin a study of time-dependent differential equations, beginning with the initial value problem (IVP) for a time-dependent ordinary differential equation (ODE).

Problem 1.1 (Initial value problem for ODE).
The IVP takes the form

$$\begin{cases} u'(t) = f(u(t), t) & \forall t > t_0 \\ u(t_0) = \eta \end{cases} \tag{1.1}$$

For convenience, we often assume $t_0 = 0$. Here we call η the initial condition and $u(t)$ the state.

Note that here $u : [0, T] \to \mathbb{R}^d$ is a vector function. We will use the following notations:

Notation 1.2.
- Newton dot notation: $\frac{dx}{dt}(t) = \dot{x}(t)$,
- p-th order derivative: $\frac{d^p x}{dt^p}(t) = x^{(p)}(t)$.

1.1 Reduction to autonomous case

Definition 1.3 (Autonomous).
An first order ODE is autonomous if

$$u'(t) = f(u(t)).$$

In other words, f doesn't depend directly on t. Otherwise, we call the ODE non-autonomous.

We have the following proposition:

Proposition 1.4.
Any IVP problem (1.1) can be reduced to the autonomous case.

Proof Sketch.
We can do this by adding one dimension to the state variable. We let $v(t) = (u(t)^\top, t)^\top$. Then we have

$$\begin{cases} v'(t) = \begin{pmatrix} g(v(t)) \\ 1 \end{pmatrix} & \forall t > t_0 \\ v(t_0) = \begin{pmatrix} \eta \\ t_0 \end{pmatrix} \end{cases}$$

where $g(v(t)) = f(u(t), t)$. This ODE is autonomous. □

1.2 Reduction to first-order case

A more general ODE is of the form:

$$u^{(p)}(t) = F(u(t), u'(t), u''(t), \dots, u^{(p-1)}(t), t).$$

Proposition 1.5.
A more general ODE can be reduced to (1.1).

Proof sketch.
Just consider

$$v(t) = \begin{bmatrix} u(t) \\ u'(t) \\ \vdots \\ u^{(p-1)}(t) \end{bmatrix}.$$

□

1.3 Linear ODEs

Definition 1.6 (Linear ODEs).
The system (1.1) is linear if

$$f(u, t) = A(t)u + g(t), \tag{1.2}$$

where $A(t) \in \mathbb{R}^{d\times d}, g(t) \in \mathbb{R}^d$. A special case is the constant coefficient linear system

$$f(u, t) = Au(t) + g(t), \tag{1.3}$$

where A is a constant matrix. If $g(t) \equiv 0$, the equation is homogeneous.

The solution to a homogeneous system $u' = Au$ is

$$u(t) = e^{A(t-t_0)}\eta.$$

We have the Duhamel's principle for linear ODEs.

Theorem 1.7 (Duhamel's principle).
The solution to (1.3) can be written as

$$u(t) = e^{A(t-t_0)}\eta + \int_{t_0}^{t} e^{A(t-\tau)}g(\tau)\, d\tau. \tag{1.4}$$

Especially, if $g(t) \equiv g$, then this can be reduced to

$$u(t) = e^{A(t-t_0)}\eta + A^{-1}(e^{A(t-t_0)} - I)g. \tag{1.5}$$

CHAPTER 2

EXISTENCE AND UNIQUENESS

When does there exist a unique solution to (1.1)? There is a standard sufficient condition. The proof by Picard iteration is standard issue mathematics that you must know, and it conveys something important about life. Suppose for a moment that we wanted to solve instead

$$\begin{cases} x'(t) = h(t), \\ x(0) = x_0 \end{cases} \tag{2.1}$$

where h is some known function. This is easy! The fundamental theorem of calculus guarantees (under mild integrability conditions on h) the existence of a unique solution as

$$x(t) = x_0 + \int_0^t h(s)dx.$$

Now suppose we already knew the solution $x(t)$. Then we could define

$$h(t) := f(x(t), t)$$

and recover $x(t)$ again as the solution of (2.1). We define Φ to be the composition of the maps before, so Φ is a map $x \mapsto \Phi[x]$ from functions to functions defined by

$$\Phi[x](t) = x_0 + \int_0^t f(x(s), s)dx$$

The self-consistency discussion is precisely saying that we are looking for fixed points of this map, i.e., x such that $\Phi[x] = x$.

Definition 2.1 (Function space).
Let $C\left([0,T];\mathbb{R}^d\right)$ denote the space of continuous functions $[0,T] \to \mathbb{R}^d$. Let $\|\cdot\|_\infty$ denote the uniform (or sup) norm on $C\left([0,T];\mathbb{R}^d\right)$, defined by

$$\|g\|_\infty = \sup\{|g(t)| : t \in [0,T]\}.$$

Here $|\cdot$ denotes the Euclidean norm on $\mathbb{R}^d$, which will be our convention throughout.

Remark 2.2.
Recall that $C\left([0,T];\mathbb{R}^d\right)$ is a Banach space with respect to the uniform norm, i.e., it is a complete metric space with respect to the metric $d(g,h) = \|g-h\|_\infty$.

2.1 Banach fixed point theorem

The key theorem for proving the existence of fixed points is the Banach fixed point theorem. Note that the proof is constructive via an iterative scheme, which even comes with a rate of convergence.

Theorem 2.3 (Banach fixed point).
Let X denote a complete metric space with metric d. Suppose that $\Phi : X \to X$ is a contraction map, i.e., there exists $\alpha \in [0, 1)$ such that

$$d(\Phi(x), \Phi(y)) \leqslant \alpha d(x, y)$$

for all $x, y \in X$. Then there exists a unique fixed point for the map Φ, i.e., a unique $x^\star$ such that $\Phi(x^\star) = x^\star$. Moreover, for any $x \in X$, if we define the sequence $\{x_k\}_{k=0}^{\infty}$ via $x_0 = x$ and $x_k = \Phi(x_{k-1})$ for all $k \geqslant 1$, then $\lim_{k\to\infty} x_k = x^\star$. In other words, the result of repeated application of Φ converges to $x^\star$ for arbitrary initial input.

Remark 2.4.
In fact, the proof recovers a convergence rate for the limit $x_k \to x^\star$. To wit, we have $d(x_k, x^\star) = O(\alpha^k)$

2.2 Picard-Lindelöf theorem

The key sufficient condition establishing existence and uniqueness of solutions of (**??**) phrased in terms of Lipschitz functions.

Definition 2.5 (Lipschitz).
A function $f : \mathbb{R}^n \to \mathbb{R}$ is Lipschitz (or Lipschitz continuous) if there exists $L \geqslant 0$ such that $|f(u) - f(v)| \leqslant L|u - v|$ for all $u, v \in \mathbb{R}^n$. In this case we can say more specifically that f is L-Lipschitz, and L is a Lipschitz constant.

Theorem 2.6 (Picard-Lindelöf).
Suppose that f is Lipschitz. Then the system (1.1) i.e.,

$$\begin{cases} x'(t) = f(x(t), t), \\ x(0) = x_0, \end{cases}$$

admits a unique solution on $[0, T]$ for any $T > 0$.

Proof.
Note that

$$\begin{aligned}
\|\Phi[x] - \Phi[y]\|_\infty &= \sup_{t\in[0,T]} \left\{ \left| \int_0^t f(x(s), s) - f(y(s), s) dx \right| \right\} \\
&\leqslant \sup_{t\in[0,T]} \left\{ \int_0^t |f(x(s), s) - f(y(s), s)| dx \right\} \\
&\leqslant \sup_{t\in[0,T]} \left\{ L \int_0^t |x(s) - y(s)| dx \right\} \\
&\leqslant \sup_{t\in[0,T]} \left\{ L \int_0^t \|x - y\|_\infty dx \right\} \\
&\leqslant LT\|x - y\|_\infty.
\end{aligned}$$

If we have $LT < 1$, then we have a contraction map. Unfortunately this cannot be guaranteed a priori.

We sidestep the difficulty in the following way. Let $h = T/N$, where N is sufficiently large such that $Lh < 1$, and consider dividing the interval $[0, T]$ into the N subintervals

$$I_0 := [0, h], I_1 := [h, 2h], \ldots, I_{N-1} := [(N-1)h, T].$$

We are going to construct a solution for (1.1) on each of these intervals individually and then argue that together they constitute a solution on $[0, T]$. Indeed let us first define $\Phi_0 : C\left(I_0; \mathbb{R}^d\right) \to C\left(I_0; \mathbb{R}^d\right)$ via

$$\Phi_0[y](t) = x_0 + \int_0^h f(y(s), s)ds.$$

Our preceding calculation ensures that Φ_0 is a contraction mapping, hence admits a unique fixed point in $C\left(I_0; \mathbb{R}^d\right)$, which is therefore the unique solution of (1.1) on I_0. As this solution is continuous up to the boundary of $[0, h]$, the final state $x_1 := x(h)$ is well-defined.

Then we can consider x_1 as the initial condition of (1.1) on the interval $[h, 2h]$. By shifting the time variable appropriately, the same argument suggests that we can extend x continuously to the interval $[0, 2h]$ such that x solves (1.1) on both $[0, h]$ and $[h, 2h]$, and we define $x_2 := x(2h)$.

More concretely, given the preceding final state x_n, we inductively define $\Phi_n : C\left(I_n; \mathbb{R}^d\right) \to C\left(I_n; \mathbb{R}^d\right)$ via

$$\Phi_n[y](t) = x_n + \int_{nh}^t f(y(s), s)ds$$

This is a contraction mapping, and we can extend x continuously to $[0, (n+1)h]$ by appending its unique fixed point. Then we define $x_{n+1} = x((n+1)h)$ to complete the inductive procedure.

In summary the construction yields $x \in C\left([0, T]; \mathbb{R}^d\right)$ with $x_n = x(nh)$, solving (1.1) on each of the individual subintervals $I_n, n = 0, \ldots, N-1$ in the sense that the restriction $x|_{I_n}$ is the unique fixed point of Φ_n for each n. Then this solution must be unique on $[0, T]$. □

Note 2.7.
Note that the most famous counterexample to global-in-time existence, the scalar equation $x'(t) = x(t)^2$, does not satisfy the Lipschitz condition. Solutions of this ODE blow up in finite time, as can be checked by direct solution via separation of variables.

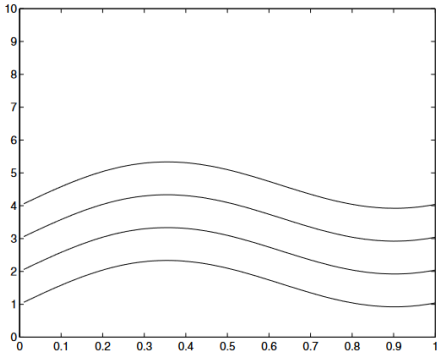
2.3 Significance of Lipschitz constant

The Lipschitz constant measures how much $f(u, t)$ changes if we perturb u (at some fixed time t). Since $f(u, t) = u'(t)$, the slope of the line tangent to the solution curve through the value u, this indicates how the slope of the solution curve will vary if we perturb u. The significance of this is best seen through some examples.

Example 2.8.
Consider $u'(t) = g(t)$ and the solution is

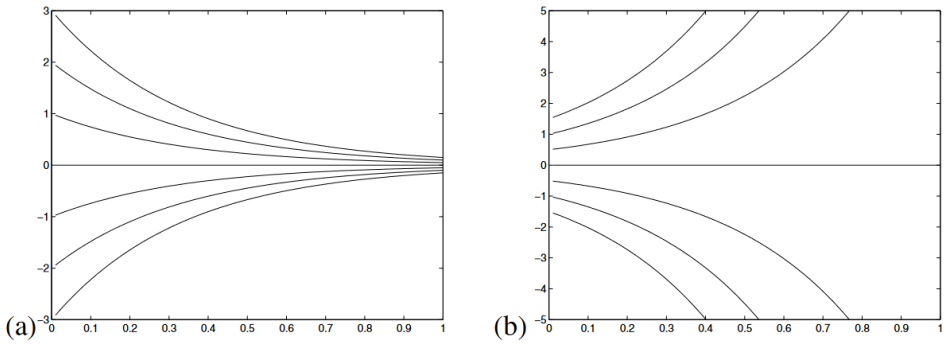
$$u(t) = u(t_0) + \int_{t_0}^t g(\tau)\, d\tau.$$

The Lipschitz constant is $L = 0$. And we have the following figure:



Note that all these curves are "parallel", since the tangent line at any particular time only depends on t.

Example 2.9.
Consider $u'(t) = \lambda u(t)$ with λ constant and $L = |\lambda|$. Then, $u(t) = u(0)\exp(\lambda t)$. Then, we have



Here $\lambda = -3$ for (a) and $\lambda = 3$ for (b). Here the slope of the solution curve does vary depending on u. The variation in the slope with u (at fixed t) gives an indication of how rapidly the solution curves are converging toward one another (in the case $\lambda < 0$) or diverging away from one another (in the case $\lambda > 0$). If the magnitude of λ were increased, the convergence or divergence would clearly be more rapid.

However, rapidly converging solution curves can also give serious numerical difficulties, which one might not expect at first glance. This is discussed in detail later, which covers stiff equations.

One should also keep in mind that a small value of the Lipschitz constant does not necessarily mean that two solution curves starting close together will stay close together forever.

Example 2.10 (Pendulum problem).
Consider the pendulum problem:

$$\theta''(t) = -\sin(\theta(t)),$$

which can be rewritten as a first order system of two equations by introducing $v(t) = \theta'(t)$:

$$u = \begin{bmatrix} \theta \\ v \end{bmatrix}, \quad \frac{d}{dt}\begin{bmatrix} \theta \\ v \end{bmatrix} = \begin{bmatrix} v \\ -\sin(\theta) \end{bmatrix}.$$

Consider the max-norm. We have

$$\|u - u^*\|_\infty = \max\left(|\theta - \theta^*|, |v - v^*|\right)$$

and

$$\|f(u) - f(u^*)\|_\infty = \max\left(|v - v^*|, |\sin(\theta) - \sin(\theta^*)|\right).$$

To bound $\|f(u) - f(u^*)\|_\infty$, first note that $|v - v^*| \leqslant \|u - u^*\|_\infty$. We also have

$$|\sin(\theta) - \sin(\theta^*)| \leqslant |\theta - \theta^*| \leqslant \|u - u^*\|_\infty$$

since the derivative of $\sin(\theta)$ is bounded by 1 . So we have Lipschitz continuity with $L = 1$:

$$\|f(u) - f(u^*)\|_\infty \leqslant \|u - u^*\|_\infty .$$

Consider two solutions to the pendulum problem one with initial data

$$\theta_1(0) = \pi - \epsilon, \quad v_1(0) = 0,$$

and the other with

$$\theta_2(0) = \pi + \epsilon, \quad v_2(0) = 0.$$

The Lipschitz constant is 1 and the data differ by 2ϵ, which can be arbitrarily small, and yet the solutions eventually diverge dramatically, as Solution 1 falls toward $\theta = 0$, while in Solution 2 the pendulum falls the other way, toward $\theta = 2\pi$.

In this case the IVP is very ill conditioned: small changes in the data can lead to order 1 changes in the solution. As always in numerical analysis, the solution of ill-conditioned problems can be very hard to compute accurately.

CHAPTER 3

EULER'S METHOD

Unfortunately, (1.1) can rarely be solved in closed form. (But when closed-form solutions exist, they're as good as gold!) The main point of numerical methods is to obtain approximate solutions in the generic unfortunate case.

3.1 Discretization

An approximate solution is represented via discretization. We will consider a step size $h = T/N$ and discrete times $0, h, 2h, \ldots, Nh$. We will represent the solution $x : [0, T] \to \mathbb{R}^d$ by its values x_n as these discrete times $t_n := nh$ for $n = 0, \ldots, N$. When N is not fixed, we use the notations $x_n^{(N)}$ and $t_n^{(N)}$ to disambiguate if the meaning is not clear from context.

A numerical scheme is a tractable computational recipe furnishing a collection of states $x_{0:N}^{(N)} = (x_0^{(N)}, \ldots, x_N^{(N)})$ that approximate the true solution states $(x(0), \ldots, x(Nh))$ at our discrete times, i.e., achieving $x_n^{(N)} \approx x(nh)$. Ideally we can control the approximation error, and to have much regard for a scheme at all, it must be the case that the error can be made arbitrarily small by advancing to the limit $N \to \infty$, in the sense that

$$\lim_{N\to\infty} \max_{n=0,\ldots,N} \left| x_n^{(N)} - x(nh) \right| = 0.$$

3.2 Explicit Euler method

Here we reproduce the IVP problem

$$\begin{cases} x'(t) = f(x(t), t) \\ x(0) = x_0. \end{cases} \tag{3.1}$$

Note that we have

$$f(x(t_n), t_n) = x'(t_n) \approx \frac{x(t_{n+1}) - x(t_n)}{h} \Rightarrow x(t_{n+1}) \approx x(t_n) + hf(x(t_n), t_n).$$

Definition 3.1 (Explicit Euler).
The discrete approximation for (1.1)

$$x_{n+1} = x_n + hf(x_n, t_n), \quad n = 0, \ldots, N-1 \tag{3.2}$$

is the explicit Euler method.

Theorem 3.2 (First order convergence of Euler).
Let f be Lipschitz continuous. Then Euler's method is convergent, and moreover

$$\max_{n=0,\dots,N} \left|x_n^{(N)} - x(nh)\right| = O(h) = O\left(N^{-1}\right).$$

Proof.
Let $E_n := |x_n - x(t_n)|$. Note that

$$x\left(t_{n+1}\right) = x\left(t_n\right) + \int_{t_n}^{t_{n+1}} f(x(t), t)dt,$$
$$x_{n+1} = x_n + hf(x_n, t_n).$$

Subtract them, we have

$$\begin{aligned} x_{n+1} - x\left(t_{n+1}\right) &= \left[x_n - x\left(t_n\right)\right] + \left[hf\left(x_n, t_n\right) - \int_{t_n}^{t_{n+1}} f(x(t), t)dt\right] \\ &= \left[x_n - x\left(t_n\right)\right] + \int_{t_n}^{t_{n+1}} \left[f\left(x_n, t_n\right) - f(x(t), t)\right] dt. \end{aligned}$$

Hence,

$$E_{n+1} \leqslant E_n + \int_{t_n}^{t_{n+1}} \left|f\left(x_n, t_n\right) - f(x(t), t)\right| dt.$$

Note that

$$\begin{aligned} \left|f\left(x_n, t_n\right) - f(x(t), t)\right| &= \left|f\left(x_n, t_n\right) - f\left(x_n, t\right) + f\left(x_n, t\right) - f(x(t), t)\right| \\ &\leqslant \left|f\left(x_n, t_n\right) - f\left(x_n, t\right)\right| + \left|f\left(x_n, t\right) - f(x(t), t)\right| \\ &\leqslant L\left|t - t_n\right| + L\left|x_n - x(t)\right| \\ &\leqslant Lh + L\left|x_n - x\left(t_n\right) + x\left(t_n\right) - x(t)\right| \\ &\leqslant Lh + L\left|x_n - x\left(t_n\right)\right| + L\left|x\left(t_n\right) - x(t)\right| \\ &= Lh + LE_n + L\left|x\left(t_n\right) - x(t)\right|. \end{aligned}$$

Besides,

$$x(t) = x\left(t_n\right) + \int_{t_n}^{t} f(x(s), s)ds,$$

so after subtracting $x\left(t_n\right)$ from both sides and taking norms, we may derive that

$$\left|x\left(t_n\right) - x(t)\right| \leqslant Bh,$$

where $B := \max_{t\in[0,T]} |f(x(t), t)|$. Hence,

$$\left|f\left(x_n, t_n\right) - f(x(t), t)\right| \leqslant LE_n + \tilde{B}h,$$

where $\tilde{B} := L(B + 1)$. Plug in this into the integral we obtain

$$E_{n+1} \leqslant (1 + Lh)E_n + \tilde{B}h^2 \Rightarrow E_{n+1} + \frac{\tilde{B}h}{L} \leqslant (1 + Lh)\left(E_n + \frac{\tilde{B}h}{L}\right).$$

Hence,

$$E_{n+1} + \frac{\tilde{B}h}{L} \leqslant (1 + Lh)^n \left(E_0 + \frac{\tilde{B}h}{L}\right) = (1 + Lh)^n \frac{\tilde{B}h}{L} \leqslant e^{Lhn} \frac{\tilde{B}h}{L}.$$

Recall that $h = T/N$, and this implies that

$$E_{n+1} \leqslant \frac{e^{Lhn} - 1}{L} \tilde{B}h \leqslant (B + 1)\frac{T}{N}(e^{LT} - 1).$$

□

Remark 3.3.
If we assume that a unique solution exists, for convergence we don't really need to additionally assume that f is Lipschitz. In this case it need only be locally Lipschitz (which follows in particular from being C^1). In the proof, wherever we use the Lipschitz constant for f, it is possible to replace with a local Lipschitz constant on a neighborhood of the true solution of the ODE. For simplicity, we just adopt the stronger assumption.

3.3 Implicit Euler method

Compare to the explicit Euler method, we can alternatively approximate the difference quotient by

$$\frac{x\left(t_{n+1}\right)-x\left(t_n\right)}{h} \approx f\left(x\left(t_{n+1}\right), t_{n+1}\right).$$

Then we will get the implicit Euler method.

Definition 3.4 (Implicit Euler).
The approximation scheme

$$x_{n+1}=x_n+hf\left(x_{n+1}, t_{n+1}\right), \quad n=0, \ldots, N-1 \tag{3.3}$$

is called the implicit Euler method.

The reason it is called implicit is that given $x_0, \ldots, x_n$, in order to determine x_{n+1} we must solve a system of (possibly nonlinear) equations, since the right-hand side depends on x_{n+1}.

Note that it is not even obvious a priori that a solution exists! However, concern about this obstacle vanishes as h goes to zero. The general intuition for why this concern vanishes comes from the implicit function theorem. Indeed, observe that if $h=0$, there exists a (unique) trivial solution $x_{n+1}=x_n$. The implicit function theorem guarantees precisely that if we perturb h by a sufficiently small amount, then we can perturb our solution accordingly to maintain satisfaction of (3.3).

More concretely, let $F(u, x)=x-uf(x, t_n+u)-x_n$. We must solve

$$F(h, x)=0. \tag{3.4}$$

Note that u is a dummy variable for the step size h. Note that $F(0, x_n)=0$. We want to say that as we perturb u away from 0, we can deform x to maintain $F(u, x)=0$.

Theorem 3.5 (Implicit function).
Let $f: \mathbb{R}^{n+m} \to \mathbb{R}^m$ be a continuous differentiable function, and (x, y) is the coordinate for $\mathbb{R}^{n+m}$, where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. Fix $(a, b) \in \mathbb{R}^{n+m}$, and $f(a, b)=0$. If the Jacobian matrix:

$$D_x f(a, b)=\left[\frac{\partial f_i}{\partial y_j}(a, b)\right]$$

is invertible, then there exists an open set $U \subset \mathbb{R}^n$ containing a such that there exists a unique continuous differentiable function $g: U \to \mathbb{R}^m$ such that $g(a)=b$ and $f(x, g(x))=0$ for all $x \in U$. Moreover,

$$\left[\frac{\partial g_i}{\partial x_j}(x)\right]_{m \times n}=-\left[\frac{\partial f_i}{\partial y_j}(x, g(x))\right]_{m \times m}^{-1}\left[\frac{\partial f_i}{\partial x_j}(x, g(x))\right]_{m \times n}.$$

Hence, our Jacobian here is

$$D_F(u, x)=I-uD_x f(x, t_n+u) \Rightarrow D_F(0, x)=I.$$

Hence, there exists $(-\delta, \delta)$ and a function $g : (-\delta, \delta) \to \mathbb{R}^d$ s.t. $F(u, g(u)) = 0$.

There is a catch, however! We don't know that we can use the same h for every n. Moreover, as we take h smaller, there infinitely many equations to solve as $N \to \infty$. In order to guarantee a uniform choice of h, one needs to sort through the innards of the implicit function theorem to determine what the choice of δ actually depends on. A sufficient (but certainly not necessary condition) that covers most cases of interest is that f is C^2.

If one adopts the simplifying assumption that f is L-Lipschitz, then the idea of Picard iteration quite easily guarantees the existence of a unique solution to the implicit Euler method for $h < 1/L$. To see this, observe that solving (3.3) for $x \in \mathbb{R}^d$ is equivalent to finding a fixed point of $\phi : \mathbb{R}^d \to \mathbb{R}^d$ defined by

$$\phi(x) := x_n + hf\left(x, t_n + h\right).$$

But

$$|\phi(x) - \phi(y)| = h\left|f\left(x, t_n + h\right) - f\left(y, t_n + h\right)\right| \leqslant Lh|x - y|,$$

so ϕ is a contraction map for $h < 1/L$, as claimed, and there exists a unique solution. This argument, together with a simple modification of our proof of Theorem 3.2 yields the following.

Theorem 3.6 (First order convergence of implicit Euler).
If f is Lipschitz, then for h sufficiently small the implicit Euler method admits a unique solution $x_{0:N}^{(N)}$, which is convergent with $\max_{n=0,\ldots,N}\left|x_n^{(N)} - x(nh)\right| = O(h) = O\left(N^{-1}\right)$.

Why consider an implicit method as opposed to an explicit one, given the hassle? A large part of the note seeks to address this question, and we defer it for now. We also remark that for any implicit method, considerations regarding the existence of solutions are quite similar to those made above.

CHAPTER 4

TAYLOR SERIES METHOD AND ERRORS

In this chapter, we will introduce Taylor series method, error estimation with Richardson extrapolation, and two types of errors: the local truncation error and the one-step error. Taylor series method will be the first attempt to systematically derive schemes of arbitrary orders of accuracy.

4.1 Taylor series methods

With Taylor series we have

$$x\left(t_{n+1}\right)=x\left(t_n\right)+hx'\left(t_n\right)+\frac{h^2}{2}x''\left(t_n\right)+O\left(h^3\right).$$

The key point is that we can get $x'(t)$ and $x''(t)$ in terms of $f, x(t_n)$ and t_n.

$$\begin{aligned}x'(t)&=f(x(t),t),\\ x''(t)&=\nabla_x f(x(t),t)\cdot x'(t)+\partial_t f(x(t),t)\\ &=\nabla_x f(x(t),t)\cdot f(x(t),t)+\partial_t f(x(t),t).\end{aligned}$$

Given an analytic expression for f, assuming it is differentiable, we may typically derive analytic expressions for $\nabla_x f$ and $\partial_t f$. Then we can write the Taylor series method of order 2, or $\mathbf{TS}(2)$ for short, as

$$x_{n+1}=x_n+hf\left(x_n,t_n\right)+\frac{h^2}{2}\left[D_x f\left(x_n,t_n\right)\cdot f\left(x_n,t_n\right)+\partial_t f\left(x_n,t_n\right)\right],$$

where D_x denotes the Jacobian matrix (with respect to the x variables) and the dot indicates matrix-vector multiplication. Note that this is an explicit method.Similarly, we can extend this to order p.

Definition 4.1 (Taylor series method).
The $\mathbf{TS}(p)$ scheme can be written as:

$$x_{n+1}=x_n+\sum_{k=1}^{p}\frac{h^k}{k!}f^{(k)}(x_n,t_n), \tag{4.1}$$

where $f^{(k+1)}(x(t),t):=\frac{d^k}{dt^k}[f(x(t),t)]$.

We have the following convergence theorem.

Theorem 4.2 (Convergence of Taylor series method).
Suppose that f is C^p and $f^{(k)}$ is Lipschitz for $k = 1, \ldots, p$. Then TS(p) is convergent, and moreover

$$\left\| \mathrm{x}^{(N)} - \mathbf{x}_{\text{true}}^{(N)} \right\|_\infty = O\left(h^p\right).$$

Proof.
Firstly,

$$x_{n+1} - x\left(t_{n+1}\right) = \left[x_n - x\left(t_n\right)\right] + \left[\sum_{k=1}^{p} \frac{h^k}{k!} f^{(k)}\left(x_n, t_n\right) - \int_{t_n}^{t_{n+1}} f(x(t), t) dt\right], \tag{4.2}$$

and we want to bound the second term by $(1 + B_1 h)E_n + B_2 h^{p+1}$. We expand $f(x(t), t)$,

$$f(x(t), t) = \sum_{k=0}^{p-1} \frac{(t - t_n)^k}{k!} \frac{d^k}{dt^k}[f(x(t), t)]\Big|_{t=t_n} + O\left(h^p\right) = \sum_{k=0}^{p-1} \frac{(t - t_n)^k}{k!} f^{(k+1)}\left(x\left(t_n\right), t_n\right) + O\left(h^p\right),$$

where we additionally define $f^{(1)} := f$. After performing the exact integration

$$\int_{t_n}^{t_{n+1}} (t - t_n)^k \, dt = \frac{\left(t_{n+1} - t_n\right)^{k+1}}{(k+1)} = \frac{h^{k+1}}{k+1}$$

and shifting the summation index we have

$$\int_{t_n}^{t_{n+1}} f(x(t), t) dt = \sum_{k=1}^{p} \frac{h^k}{k!} f^{(k)}\left(x\left(t_n\right), t_n\right) + O\left(h^{p+1}\right).$$

Then (4.2) can be rewritten as

$$x_{n+1} - x\left(t_{n+1}\right) = \left[x_n - x\left(t_n\right)\right] + \sum_{k=1}^{p} \frac{h^k}{k!} \left[f^{(k)}\left(x_n, t_n\right) - f^{(k)}\left(x\left(t_n\right), t_n\right)\right] + O\left(h^{p+1}\right).$$

Taking norms of both sides, using the triangle inequality and the Lipschitz property for each of the $f^{(k)}$, we obtain

$$\begin{aligned} E_{n+1} &\leqslant E_n + \sum_{k=1}^{p} \frac{h^k}{k!} L E_n + O\left(h^{p+1}\right) \\ &= \left(1 + L \sum_{k=1}^{p} \frac{h^k}{k!}\right) E_n + O\left(h^{p+1}\right). \end{aligned}$$

Note that for small h (say concretely for $h \leqslant 1$), there exists some universal $B_1 \geqslant 0$ such that

$$L \sum_{k=1}^{p} \frac{h^k}{k!} \leqslant B_1 h,$$

and there exists also some $B_2 \geqslant 0$ such that then

$$E_{n+1} \leqslant \left(1 + B_1 h\right) E_n + B_2 h^{p+1}$$

for all h sufficiently small. Similar to the proof of Theorem 3.2, we have

$$E_n \leqslant \frac{\left(1 + B_1 h\right)^{n+1} - 1}{B_1 h} \cdot B_2 h^{p+1} \leqslant C h^p$$

for an appropriate constant C independent of n and N. This completes the proof. □

Remark 4.3.
We don't really need to additionally assume that the $f^{(k)}$ are (globally) Lipschitz. Assuming that a unique solution exists, it suffices for them all to be merely locally Lipschitz, which is in particular guaranteed by the assumption that f is C^p. In the proof, wherever we use Lipschitz constants, it is possible to replace them with local Lipschitz constants on a neighborhood of the true solution of the ODE. For simplicity we just adopt the stronger assumption.

Taylor series methods have a drawback that is often fatal: they require us to compute all the p-th order partial derivatives of f, which, depending on the practical context may not be readily available or may be very expensive to evaluate. Still they serve as a useful starting point and inspiration for thinking about higher-order schemes.

4.2 Error estimation with Richardson extrapolation

Note that the proof of Theorem 3.2 and Theorem 4.2 furnishes an explicit error bound on the approximate solution $x^{(N)}_{0:N}$. *In practice this error bound can be extremely pessimistic, although though the order of convergence is sharp!* By order of convergence, we mean the largest exponent p such that the error is $O(h^p) = O(1/N^p)$, and we say for p so defined that a scheme is p-th order accurate. In the case of Euler's method, $p = 1$.

Unfortunately, the optimal preconstant C such that the error at some time t is bounded asymptotically by Ch^p may be very hard to estimate a priori. In order to save as much computation as possible, we want to take h only as small as necessary to achieve a desired error tolerance, so getting a sharper estimate, even if a posteriori, is quite desirable.

In the sequel we will want to compare our discrete solutions $\mathrm{x}^{(N)}$ solutions for different values of N. Note that the N-th discrete solution is only defined at grid points $(0, h, 2h, \dots, T)$, where $h = T/N$, so in general for $N' \neq N, x^{(N)}_{0:N}$ and $x^{(N')}_{0:N'}$ may not be directly comparable. They can be compared after suitable interpolation to the entire interval $[0, T]$. However, such interpolation is rarely performed in practice, and moreover, care must be taken so that the interpolation preserves the order of accuracy of the scheme! In the simple case of Euler's method, linear interpolation is sufficient to preserve first-order accuracy.

For the purposes of this discussion it is more elegant/convenient to assume that such an interpolation exists (though as we shall see, we will not need to construct it in practice), so for a general p-th order accurate scheme, we assume that we have an interpolant $x^{(N)} : [0, T] \to \mathbb{R}^d$ such that

$$\left\| x^{(N)} - x \right\|_\infty = O(1/N^p)$$

where $x(\cdot)$ is here the true solution. (In this section, the superscript does not indicate repeated differentiation!) In particular, we have for every time $t \in [0, T]$,

$$x^{(N)}(t) = x(t) + O(1/N^p).$$

We postulate the more detailed error expansion, consistent with p-th order accuracy:

$$x^{(N)}(t) = x(t) + \frac{C(t)}{N^p} + O\left(1/N^{p+1}\right).$$

Plugging in $2N$ in the place of N (in practice, solving the scheme on a grid that is twice as fine), we have

$$x^{(2N)}(t) = x(t) + \frac{C(t)}{2^p N^p} + O\left(1/N^{p+1}\right).$$

Then observe that be taking the linear combination

$$\tilde{x}^{(N)}(t) := \frac{2^p x^{(2N)}(t) - x^{(N)}(t)}{2^p - 1} = x(t) + O\left(1/N^{p+1}\right)$$

we cancel the leading-order contribution to the error. Then not only is $\tilde{x}^{(N)}$ a more accurate solution, but also we can use it to estimate the error of our original solution $x^{(N)}$. Indeed our error $E^{(N)}$ as a function of time satisfies

$$E^{(N)}(t) := \left|x^{(N)}(t) - x(t)\right| = \left|x^{(N)}(t) - \tilde{x}^{(N)}(t)\right| + O\left(1/N^{p+1}\right),$$

and in the last expression the first term is only $O\left(1/N^p\right)$, hence dominates the second term. Therefore the N-point scheme error at time t can be estimated as

$$E^{(N)}(t) \approx \left|x^{(N)}(t) - \tilde{x}^{(N)}(t)\right| = \left|\frac{2^p(x^{(N)}(t) - x^{(2N)}(t))}{2^p - 1}\right|.$$

Definition 4.4 (Richardson extrapolation).
In practice, let's fix some N and $h = T/N$ and say we are interested in estimating the accuracy of the solution on the fixed grid $t_n := t_n^{(N)}, n = 0, \ldots, N$, i.e., on $(0, h, 2h, \ldots, T)$. We can do this by also solving the ODE on the finer grid $t_n^{(2N)}, n = 0, \ldots, 2N$, which includes the original grid as a subgrid. Then we can simply perform the above procedure at the grid points t_n instead of the entire interval, defining an extrapolated solution

$$\tilde{x}_n^{(N)} := \frac{2^p x_{2n}^{(2N)} - x_n^{(N)}}{2^p - 1}$$

which is $(p+1)$-th order accurate. This procedure for determining the extrapolated solution is called Richardson extrapolation.

Theorem 4.5 (Error estimation).
Given a p-th order accuracy method and two approximation sequences $x_{0:N}^{(N)}$ and $x_{0:2N}^{(2N)}$, we can estimate the error of $x_N^{(N)}$ by

$$E^{(N)}(t) := \left|x^{(N)}(t) - x(t)\right| \approx \left|x^{(N)}(t) - \tilde{x}^{(N)}(t)\right| = \left|\frac{2^p\left(x^{(N)}(t) - x^{(2N)}(t)\right)}{2^p - 1}\right|,$$

where $\tilde{x}^{(N)}(t) := \frac{2^p x^{(2N)}(t) - x^{(N)}(t)}{2^p - 1}$ is the Richardson extrapolation.

4.3 Local truncation error and one-step error

The truncation error for all these methods is defined by writing the difference equation in the form that directly models the derivatives and inserting the true solution to the ODE.

Example 4.6 (LTE of Explicit Euler).
The local truncation error (LTE) of the explicit Euler method (3.2) is defined by

$$\begin{aligned}
\tau^n &= \frac{x(t_{n+1}) - x(t_n)}{h} - f(x(t_n)) \\
&= \left[x'(t_n) + \frac{1}{2}hx''(t_n) + O(h^2)\right] - f(x(t_n)) \\
&= \frac{1}{2}hx''(t_n) + O(h^2).
\end{aligned}$$

Hence, the LTE is $O(k)$ and we say the method is first order consistency.

We need some form of stability to guarantee that the global error will exhibit the same rate of convergence as the local truncation error. This will be discussed below.

In much of the literature concerning numerical methods for ODEs, a slightly different definition of the local truncation error. Denoting this value by $\mathcal{L}^n$, we have

$$\begin{aligned}\mathcal{L}^n &= x\left(t_{n+1}\right) - x\left(t_n\right) - hf\left(x\left(t_n\right)\right) \\ &= \frac{1}{2}h^2 x''(t_n) + O(h^3).\end{aligned}$$

Since $\mathcal{L}^n = h\tau^n$, this local error is $O\left(h^2\right)$ rather than $O\left(h\right)$, but of course the global error remains the same and will be $O\left(h\right)$. Using this alternative definition, many standard results in ODE theory say that a p th order accurate method should have an LTE that is $O\left(h^{p+1}\right)$. With the notation we are using, a pth order accurate method has an LTE that is $O\left(h^p\right)$. The notation used here is consistent with the standard practice for PDEs and leads to a more coherent theory, but one should be aware of this possible source of confusion.

In this note $\mathcal{L}^n$ will be called the **one-step error**, since this can be viewed as the error that would be introduced in one time step if the past values $x_n, x_{n-1}, \ldots$ were all taken to be the exact values from $x(t)$. In the explicit Euler method, in one step the error introduced is $O(h^2)$. This is consistent with first order accuracy in the global error if we think of trying to compute an approximation to the true solution $x(T)$ at some fixed time $T > 0$. To compute from time $t = 0$ up to time T, we need to take T/h time steps of length h. A rough estimate of the error at time T might be obtained by assuming that a new error of size $\mathcal{L}^n$ is introduced in the nth time step and is then simply carried along in later time steps without affecting the size of future local errors and without growing or diminishing itself. Then we would expect the resulting global error at time T to be simply the sum of all these local errors. Since each local error is $O\left(h^2\right)$ and we are adding up T/h of them, we end up with a global error that is $O\left(h\right)$.

This viewpoint is in fact exactly right for the simplest ODE, in which $f(x,t) = g(t)$ is independent of x and the solution is simply the integral of g, but it is a bit too simplistic for more interesting equations since the error at each time feeds back into the computation at the next step in the case where $f(x,t)$ depends on x. Nonetheless, it is essentially right in terms of the expected order of accuracy, provided the method is stable. In fact, it is useful to think of stability as exactly what is needed to make this naive analysis correct, by ensuring that the old errors from previous time steps do not grow too rapidly in future time steps. This will be investigated in detail in the following chapters.

Part II

Linear Multistep Methods

CHAPTER 5

LINEAR MULTISTEP METHODS

Given the solution $x(t)$ at time t of (1.1) we already encountered two different ways of approximately advancing the solution by one increment h of time. The first was

$$x(t+h) \approx x(t) + hf(x(t), t)$$

which led to the explicit Euler method, and the second was

$$x(t+h) \approx x(t) + hf(x(t+h), t+h)$$

which led to the implicit Euler method. One can readily imagine a mixture of the two, approaches:

$$x(t+h) \approx x(t) + \frac{1}{2}[f(x(t), t) + f(x(t+h), t+h)],$$

which in fact has one higher order of accuracy, as we shall see. The resulting numerical schemes were written compactly as

$$\begin{aligned} x_{n+1} - x_n &= hf_n && \text{(Explicit Euler)} \\ x_{n+1} - x_n &= hf_{n+1} && \text{(Implicit Euler)} \\ x_{n+1} - x_n &= h\left(f_n + f_{n+1}\right) && \text{(Trapezoidal rule)} \end{aligned}$$

where as always $f_n = f\left(x_n, t_n\right)$ implicitly depends on x_n. Linear multistep methods (LMMs) generalize these schemes considerably.

Definition 5.1 (Linear multistep methods).
The general form of an r-step LMM is

$$\sum_{j=0}^{r} \alpha_j x_{n+j} = h \sum_{j=0}^{r} \beta_j f_{n+j}. \tag{5.1}$$

Assuming that x_m is known for $m = 0, \ldots, n+r-1$, this equation can in principle be solved for the next value x_{n+r}.

Note that explicit Euler, implicit Euler and trapezoidal rule are special cases of LMM. For convenience, we always assume $\alpha_r = 1$. If $\beta_r = 0$, this method will be explicit. Otherwise, it will be implicit.

5.1 Local truncation error

In terms of Example 4.6, the LTE for LMM is:

$$\tau^n := \frac{1}{h} \sum_{j=0}^{r} \alpha_j x_{n+j} - \sum_{j=0}^{r} \beta_j f_{n+j}.$$

Besides, the one-step error is:

$$\mathcal{L}^n = \sum_{j=0}^{r} \alpha_j x_{n+j} - h \sum_{j=0}^{r} \beta_j f_{n+j}.$$

Note that we can also define a linear operator for a differential function z,

$$\mathscr{L}_h z(t) := \sum_{j=0}^{r} \alpha_j z(t+jh) - h \sum_{j=0}^{r} \beta_j z'(t+jh). \tag{5.2}$$

Proposition 5.2 (LTE for 3 methods).

- For Explicit Euler, $\mathscr{L}_h z(t) = \frac{h^2}{2} z''(t) + O(h^3)$;
- For Implicit Euler, $\mathscr{L}_h z(t) = -\frac{h^2}{2} z''(t) + O(h^3)$;
- For trapezoidal rule, $\mathscr{L}_h z(t) = \frac{h^3}{12} z'''(t) + O(h^4)$.

Hence, the trapezodial rule is second order accurate.

5.2 Consistency

Definition 5.3 (Consistency).
We say that an LMM is consistent of order p or order- p consistent if $\mathscr{L}_h z(t) = O\left(h^{p+1}\right)$ for every smooth z. In particular, we say that it is consistent if it is order- p consistent for some $p \geqslant 1$. Moreover, if

$$\mathscr{L}_h z(t) = C_{p+1} h^{p+1} z^{(p+1)}(t) + O\left(h^{p+2}\right),$$

where $C_{p+1} \neq 0$. This nonzero coefficient C_{p+1} is called the error constant of the LMM.

In general, the error constant can in principle be computed analytically, and it may be useful to do so! Indeed, Milne's device (to be discussed later) uses error constants to estimate the error of an LMM and possibly improve it.

It is not hard to derive necessary and sufficient conditions for the consistency of (5.1).

Theorem 5.4 (Consistency of LMM).
An LMM (5.1) is consistent if and only if

$$\sum_{j=0}^{r} \alpha_j = \sum_{j=0}^{r} (j\alpha_j - \beta_j) = 0.$$

Proof.
Note that

$$\begin{aligned}
\mathscr{L}_h z(t) &= \sum_{j=0}^{r} \alpha_j z(t+jh) - h \sum_{j=0}^{r} \beta_j z'(t+jh) \\
&= \sum_{j=0}^{r} \alpha_j \left[z(t) + jhz'(t)\right] - h \sum_{j=0}^{r} \beta_j \left[z'(t)\right] + O\left(h^2\right) \\
&= \left(\sum_{j=0}^{r} \alpha_j\right) z(t) + h \left(\sum_{j=0}^{r} j\alpha_j - \beta_j\right) z'(t) + O\left(h^2\right).
\end{aligned}$$

In fact, a more general expansion is

$$\mathscr{L}_h z(t) = \left(\sum_{j=0}^{r} \alpha_j\right) z(t) + \sum_{q=1}^{p-1} h^q \left(\sum_{j=0}^{r} \left(\frac{1}{q!} j^q \alpha_j - \frac{1}{(q-1)!} j^{q-1} \beta_j\right)\right) z^{(q)}(t) + O(h^p).$$

Hence, we can get the sufficient and necessary condition for any order p. □

Definition 5.5 (first and second characteristic polynomials of LMM).
The first and the second characteristic polynomials are:

$$\rho(w) = \sum_{j=0}^{r} \alpha_j w^j, \quad \sigma(w) = \sum_{j=0}^{r} \beta_j w^j.$$

Corollary 5.6.
An LMM is consistent if and only if

$$\rho(1) = 0, \rho'(1) = \sigma(1).$$

As we shall later see, consistency does not imply convergence! There is an additional stability requirement that we shall treat later.

5.3 Starting values

Note that to solve an r-step LMM, we need r starting values $x_0, \ldots, x_{r-1}$ in order to initialize the scheme. This is annoying! We are only given $x_0 = x(0)$ from the initial condition of (1.1).

The most naive option is to simply take $x_j = x_0$ for $j = 1, \ldots, r-1$. However, this might sacrifice the order of accuracy of our scheme. Let us try to get some heuristic understanding of what we require.

In general, assume we have $x(t_j) = x_0 + \Theta(h)$ for $j = 0, \ldots, r-1$, since $x(t)$ is differentiable at $t = 0$. Therefore under the naive approach, our error $e_j := x_j - x(t_j)$ is already $\Theta(h)$ by time step $j = r-1$. If our scheme is order- p consistent in the sense that the one-step error is $O(h^{p+1})$, we will accumulate an additional error that is $O(h^{p+1})$ at each of the remaining $O(h^{-1})$ time steps (cf. the proof of Theorem 11, conceptually), i.e., we will accumulate additional error of $O(h^p)$. However, this additional error accumulation is dominated by our $\Theta(h)$ initialization error, and the entire method is really only first-order consistent. If $p = 1$, this is no worse than expected, and we can adopt the naive approach without too much regret.

More generally, the preceding argument suggests that we need to initialize $x_j = x(t_j) + O(h^p)$ in order to fulfill the dream of order- p accuracy for an LMM with $O(h^{p+1})$ one-step error. We could determine these intitial values, for example, by running $r-1$ steps of the order- $(p-1)$ accurate Taylor series method TS $(p-1)$. Since the number of steps needed for initialization is independent of the step size h, despite the disadvantages of Taylor series methods this may be a reasonable practical approach. In order to solve the ODE up to time T, we will need to run $\sim h^{-1}$ steps of the LMM, so as $h \to 0$, the initialization cost should be negligible.

Still, it may be more practically efficient to use Runge-Kutta methods, which do not require starting values, for initialization. (Runge-Kutta methods will be the subject of the next chapter of the note.)

CHAPTER 6

FAMILIES OF LMMS

At this point we introduce several important families of LMMs that go beyond $r = 1$ to achieve higher orders of accuracy. Later on we will discuss the stability and convergence of these methods, but for now we focus only on the local truncation error.

6.1 Integral-based methods

Recall we are essentially trying to predict $x(t_{n+r})$ based on the values $x(t_{n+j})$ for $j = 0, \ldots, r-1$. Also recall that the solution of (1.1) satisfies

$$x(t_{n+r}) = x(t_{n+r-1}) + \int_{t_{n+r-1}}^{t_{n+r}} f(x(t), t)dt. \tag{6.1}$$

Can we use the preceding values $x(t_{n+j})$ to approximate the integral term more accurately? Note that explicit Euler, implicit Euler, and the trapezoidal rule follow, respectively, from applying the left endpoint, right endpoint, and trapezoidal rules for approximating the integral.

In particular, the trapezoidal rule can be interpreted as replacing $f(x(t), t)$) with a linear interpolation over the interval of integration. But to begin with, we will focus on developing a higher-order explicit approachg. As such we will first generalize the left endpoint rule. This can be seen as replacing $f(x(t), t)$ with the 'trivial polynomial interpolation' which matches the value at the left endpoint with a constant polynomial.

6.1.1 Adams-Bashforth methods

More generally, we can use all r of the values $f_{n+j} = f(x(t_{n+j}), t_{n+j}), j = 0, \ldots, r-1$, to construct a more accurate polynomial interpolation over the interval $[t_{n+r-1}, t_{n+r}]$. We expect our interpolation to then achieve $O(h^r)$ accuracy pointwise, so our LTE will be $O(h^r)$, and we will have a method that is order- p consistent. In fact as we shall see the resulting r-step **Adams-Bashforth method** will be a LMM, since the interpolation depends linearly on the f_{n+j}.

We will use the Lagrange basis to construct the polynomial interpolation. Since we already know $\alpha_r = 1, \alpha_{r-1} = -1$, we only need to determine the coefficients β_j for the method. Let's focus on the first r nodes:

$$\mathbf{t} = (t_0, t_1, \ldots, t_{r-1}) = (0, h, \ldots, (r-1)h)$$

and let $\ell_j(t; \mathbf{t})$ be the Lagrange basis polynomials as in Appendix A:

$$\ell_j(t; \mathbf{t}) = \prod_{i \in \{0,\ldots,r-1\} \setminus \{j\}} \frac{t - ih}{(j-i)h}, \quad j = 0, \ldots, r-1.$$

And our approximation for f is:

$$f(x(t),t) \approx \sum_{j=0}^{r-1} \ell_j(t;\mathbf{t}) f_j.$$

Hence, after the integration, the coefficient

$$\begin{aligned}
\beta_j &= \frac{1}{h}\int_{(r-1)h}^{rh} \ell_j(t;\mathbf{t})dt = \int_{r-1}^{r} \ell_j(ht;\mathbf{t})dt \\
&= \frac{(-1)^{r-1-j}}{j!(r-1-j)!}\int_{r-1}^{r} \prod_{i\in\{0,\dots,r-1\}\setminus\{j\}} (t-i)dt \\
&= \frac{(-1)^{r-1-j}}{j!(r-1-j)!}\int_{r-1}^{r} \prod_{i\in\{0,\dots,r-1\}\setminus\{r-1-j\}} (t-(r-1-i))dt \\
&= \frac{(-1)^{r-1-j}}{j!(r-1-j)!}\int_{0}^{1} (t+i)dt,
\end{aligned}$$

or equivalently:

$$\beta_{r-1-j} = \frac{(-1)^{j}}{j!(r-1-j)!}\int_{0}^{1} \prod_{i\in\{0,\dots,r-1\}\setminus\{j\}} (t+i)dt, \quad j=0,\dots,r-1. \tag{6.2}$$

The integrals in (6.2) can be evaluated analytically, but there does not seem to be a neat formula. Of course, once they are worked out a single time, they never need be computed again!

Theorem 6.1 (Consistency of Adams-Bashforth).
Given $\alpha = (1,-1,0,\dots,0)$ and β_j satisfying (6.2), the Adams-Bashforth is of order r consistency.

Here are some examples.

Example 6.2 (Explicit Adams-Bashforth methods).

- 1-step: $x^{n+1} = x^n + kf\left(x^n\right)$ (forward Euler)
- 2-step: $x^{n+2} = x^{n+1} + \frac{k}{2}\left(-f\left(x^n\right) + 3f\left(x^{n+1}\right)\right)$
- 3-step: $x^{n+3} = x^{n+2} + \frac{k}{12}\left(5f\left(x^n\right) - 16f\left(x^{n+1}\right) + 23f\left(x^{n+2}\right)\right)$
- 4-step: $x^{n+4} = x^{n+3} + \frac{k}{24}\left(-9f\left(x^n\right) + 37f\left(x^{n+1}\right) - 59f\left(x^{n+2}\right) + 55f\left(x^{n+3}\right)\right)$

6.1.2 Adams-Moulton methods

If we allow ourselves to additionally use the right endpoint of our interval of integration (as in the trapezoidal rule), then we can get a polynomial interpolation with one additional order of accuracy. Specifically, the LTE is $O\left(h^{r+2}\right)$, and the method is now of order $r+1$. However, beware that this so-called Adams-Moulton method is now implicit!

Now we have:

$$\beta_{r-j} = \frac{(-1)^{j}}{j!(r-j)!}\int_{0}^{1} \prod_{i\in\{0,\dots,r\}\setminus\{j\}} (t+i-1)dt, \quad j=0,\dots,r. \tag{6.3}$$

Theorem 6.3 (Consistency of Adams-Moulton).
Given $\alpha = (1,-1,0,\dots,0)$ and β_j satisfying (6.2), the Adams-Moulton is of order $r+1$ consistency.

Example 6.4 (Implicit Adams-Moulton methods).

- 1-step: $x^{n+1} = x^n + \frac{k}{2}\left(f\left(x^n\right) + f\left(x^{n+1}\right)\right)$ (trapezoidal method)
- 2-step: $x^{n+2} = x^{n+1} + \frac{k}{12}\left(-f\left(x^n\right) + 8f\left(x^{n+1}\right) + 5f\left(x^{n+2}\right)\right)$
- 3-step: $x^{n+3} = x^{n+2} + \frac{k}{24}\left(f\left(x^n\right) - 5f\left(x^{n+1}\right) + 19f\left(x^{n+2}\right) + 9f\left(x^{n+3}\right)\right)$
- 4-step: $x^{n+4} = x^{n+3} + \frac{k}{720}\left(-19f\left(x^n\right) + 106f\left(x^{n+1}\right) - 264f\left(x^{n+2}\right) + 646f\left(x^{n+3}\right) + 251f\left(x^{n+4}\right)\right)$

6.1.3 Nyström methods

The **(explicit) Nyström methods** are based on the identity

$$x\left(t_{n+r}\right) = x\left(t_{n+r-2}\right) + \int_{t_{n+r-2}}^{t_{n+r}} f(x(t), t)dt. \tag{6.4}$$

The data $\left(t_n, f_n\right), \ldots, \left(t_{n+r-1}, f_{n+r-1}\right)$ can be used to construct a Lagrange interpolating polynomial which may then be substituted for $f(x(t), t)$ in the preceding expression to derive the r-step Nyström method. The order of accuracy is r, same as that of the r-step Adams-Bashforth method. For such methods we always have

$$\alpha_r = 1, \alpha_{r-1} = 0, \alpha_{r-2} = -1, \alpha_{r-3} = \cdots = \alpha_0 = 0.$$

Further general details are left as an exercise. Let us illustrate the special case of the 2-step explicit Nyström method. In this case we insert into (6.4) the linear interpolation

$$f(x(t), t) \approx f_n\left(1 - \frac{t - t_n}{h}\right) + f_{n+1}\frac{t - t_n}{h},$$

yielding

$$f(x(t), t) \approx f_n\left(1 - \frac{t - t_n}{h}\right) + f_{n+1}\frac{t - t_n}{h}, \quad \beta_1 = \int_0^2 tdt = 2.$$

Hence, the 2-step explicit Nyström method is given by

$$x_{n+2} - x_n = 2hf_{n+1}.$$

This is called the midpoint method or leapfrog method. It can of course be derived more simply by substituting the second-order accurate difference quotient approximation

$$x'(t) = \frac{x(t + h) - x(t - h)}{2h} + O\left(h^2\right).$$

6.1.4 Milne-Simpson methods

The Milne-Simpson methods are to the Nyström methods as the Adams-Moulton methods are to the Adams-Bashforth methods, using the full data $\left(t_n, f_n\right), \ldots, \left(t_{n+r}, f_{n+r}\right)$ for polynomial interpolation. Sometimes these are called (implicit) Nyström methods, and the order of accuracy is $r + 1$ in general. Again

$$\alpha_r = 1, \alpha_{r-1} = 0, \alpha_{r-2} = -1, \alpha_{r-3} = \cdots = \alpha_0 = 0,$$

and further details are left as an exercise. We illustrate the special case of the 2-step Milne-Simpson method. Here the substitution of the interpolating polynomial is equivalent to approxmating the integral in (6.4) by Simpson's rule,

$$\int_a^b g(t)dt \approx \frac{g(a) + 4g\left(\frac{a+b}{2}\right) + g(b)}{6}(b - a),$$

which yields the method that we also call Simpson's rule:

$$x_{n+2} - x_n = \frac{2h}{6}\left(f_n + 4f_{n+1} + f_{n+2}\right).$$

In fact, Simpson's rule (due to it's high symmetry about the $(n + 1)$-th time step) enjoys one higher order of accuracy than is guaranteed a priori by its status as the 2-step Milne Simpson method, i.e., it is order- 4 consistent. The verification of this is left as an exercise.

6.2 Backward differentiation formulas

The LMMs of Adams and Nyström type considered above adopt the approach of taking only a few of the α_j nonzero and then guaranteeing higher orders of accuracy by integral approximation within an integral formulation of the ODE (1.1) as in (6.1).

An alternative approach is to take the differential formulation

$$x'(t) = f(x(t), t) \tag{6.5}$$

and simply plug higher-order accurate finite difference approximations into the left-hand side. In the resulting scheme, only one of the β_j will be nonzero, but perhaps many of the α_j will be nonzero, depending on the order of the method.

We already have some examples that can be interpreted this way: explicit Euler, implicit/backward Euler, and the midpoint method are based on substitution of the backward, forward, and central finite difference formulas

$$\frac{x(t) - x(t-h)}{h}, \quad \frac{x(t+h) - x(t)}{h}, \quad \frac{x(t+h) - x(t-h)}{2h}$$

into (6.5) at time t.

At the same time, we can also think of backward Euler as the result of substituting a backward finite difference formula into (6.5) at time $t + h$. This perspective is generalized by the backward differentiation formulas (BDFs), which are of interest for their stability properties to be considered later.

Concretely, we seek coefficients c_j such that

$$z'(t) \approx \frac{1}{h} \sum_{j=0}^{r} c_j z(t - jh) \tag{6.6}$$

for smooth z. In fact it will be possible for the approximation to hold with $O(h^r)$ error.

Then the resulting BDF is derived by substituting the approximation (6.6) into (6.5) at time $t + rh$, resulting in a LMM with coefficients $\alpha_j = c_{r-j}$ for $j = 0, \ldots, r$ and $\beta_r = 1, \beta_0 = \cdots = \beta_{r-1} = 0$. Note that by considering $\tilde{z}(t) = z(-t)$, it is equivalent to find v_j such that

$$z'(t) = \frac{1}{h} \sum_{j=0}^{r} v_j z(t + jh) + O(h^r) \tag{6.7}$$

for all smooth z, i.e., to determine a higher-order forward differentiation formula, and then set $c_j = -v_j$ for $j = 0, \ldots, r$. Simply expand

$$\begin{aligned} \sum_{j=0}^{r} v_j z(t + jh) &= \sum_{j=0}^{r} v_j \left(\sum_{k=0}^{r} z^{(k)}(t) \frac{j^k h^k}{k!} + O\left(h^{r+1}\right) \right) \\ &= \sum_{k=0}^{r} \left(\sum_{j=0}^{r} j^k v_j \right) z^{(k)}(t) \frac{h^k}{k!} + O\left(h^{r+1}\right), \end{aligned}$$

where we interpret $0^0 = 1$ for notational compactness. Note that to guarantee (6.7) we need

$$\sum_{j=0}^{r} j^k v_j = \begin{cases} 0, & k = 0, \\ 1, & k = 1, \\ 0, & k = 2, \ldots, r. \end{cases}$$

We can view this is a linear system of equations for the v_j. Define the $(r+1) \times (r+1)$ matrix $A = (A_{ij}) = (j^i)$, where we have adopted a zero-indexing convention for the indices $i, j = 0, \ldots, r$. We

can also write out

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 2 & 3 & \cdots & r \\ 0^2 & 1^2 & 2^2 & 3^2 & \cdots & r^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0^r & 1^r & 2^r & 3^r & \cdots & r^r \end{pmatrix}$$

If we let $\mathbf{v} = (v_0, \ldots, v_r)^\top \in \mathbb{R}^{r+1}$ be the vector of unknown coefficients, then we are seeking to solve the linear system

$$A\mathbf{v} = e_1$$

where $e_1 = (0, 1, 0, \ldots, 0)^\top$. In fact $V := A^\top$ is a Vandermonde matrix, i.e., a matrix of the form

$$V = \left(a_i^j\right) = \begin{pmatrix} 1 & a_0 & a_0^2 & a_0^3 & \cdots & a_0^r \\ 1 & a_1 & a_1^2 & a_1^3 & \cdots & a_1^r \\ 1 & a_2 & a_2^2 & a_2^3 & \cdots & a_2^r \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_r & a_r^2 & a_r & \cdots & a_r^r \end{pmatrix}.$$

The classical formula

$$\det(V) = \det(A) = \prod_{0 \leqslant i < j \leqslant r} (a_j - a_i)$$

guarantees in particular that A is invertible if and only if the a_i are all distinct. In our case $a_i = i$ for $i = 0, \ldots, r$, so A is invertible and $Av = e_1$ has a unique solution $\mathbf{v}$, which uniquely specifies a unique BDF scheme of order r via $\alpha_j = -v_{r-j}$ for $j = 0, \ldots, r$.

Example 6.5 (BDFs).

- BDF1: $x_{n+1} - x_n = hf_{n+1}$,
- BDF2: $x_{n+2} - \frac{4}{3}x_{n+1} + \frac{1}{3}x_n = \frac{2}{3}hf_{n+2}$,
- BDF3: $x_{n+3} - \frac{18}{11}x_{n+2} + \frac{9}{11}x_{n+1} - \frac{2}{11}x_n = \frac{12}{25}hf_{n+3}$.

CHAPTER 7

SOLVING IMPLICIT METHODS

For LMM of the form (5.1), which we reproduce here:

$$\sum_{j=0}^{r} \alpha_j x_{n+j} = h \sum_{j=0}^{r} \beta_j f_{n+j}.$$

Here we assume $\alpha_r = 1$ and $\beta_r \neq 0$. Then

$$x_{n+r} = \sum_{j=0}^{r-1} (-\alpha_j) x_{n+j} + h\beta_r f(x_{n+r}, t_{n+r}) + \sum_{j=0}^{r-1} h\beta_j f_{n+j}.$$

For fixed $x_n, \ldots, x_{n+r-1}$, define the function $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ by

$$\Phi(u) := \sum_{j=0}^{r-1} (-\alpha_j) x_{n+j} + h\beta_r f(u, t_{n+r}) + \sum_{j=0}^{r-1} h\beta_j f_{n+j}.$$

Then to solve the implicit scheme for x_{n+r}, given all preceding values $x_n, \ldots, x_{n+r-1}$, we must solve the (generally nonlinear) system of equations

$$u = \Phi(u). \tag{7.1}$$

Or, defining $F(u) := u - \Phi(u)$, we may write even more abstractly

$$F(u) = 0. \tag{7.2}$$

General approaches to solving such systems of equations are *iterative*, in that sense that a solution $u^\star$ is furnished as a limit

$$u^\star = \lim_{k \to \infty} u^{(k)},$$

where each successive iterate $u^{(k+1)}$ can be feasibly computed from the last iterate $u^{(k)}$.

Definition 7.1 (Linear convergence).
We say that an iterative converges linearly with rate $\alpha \in (0, 1)$ if

$$|u^{(k)} - u^*| = O(\alpha^{(k)}).$$

You might think that we should say that such a method converges 'exponentially,' but in fact it is much less classy to do so. It is the logarithm of the error that converges linearly, and it is important to stay humble, especially since, as we shall see it is always possible in principle to do much better (though perhaps at great cost). Indeed, if a method converges linearly, this means precisely that each successive digit of accuracy is just as costly to produce as the last digit. If a method converges superlinearly, then each successive digit comes more easily, and once we are close to a solution, we tend to get to machine precision in a hurry.

7.1 Picard iteration

Solving (7.2) is equivalent to looking for a fixed point Φ. And we can use the Picard iteration.

$$u^{(k+1)} = \Phi(u^{(k)}),$$

for arbitrary initialization $u^{(0)}$. Now if f is L-Lipschitz, then we have

$$\begin{aligned} |\Phi(u) - \Phi(v)| &= h\,|\beta_r|\,|f\,(u, t_{n+r}) - f\,(v, t_{n+r})| \\ &\leqslant L\,|\beta_r|\,h|u - v|, \end{aligned}$$

so if $h < \frac{1}{(L|\beta_r|)}$, then indeed Φ is a contraction mapping with a unique fixed point.

Part III

Advection Equations and Hyperbolic Systems

CHAPTER 8

ADVECTION EQUATIONS AND METHOD OF LINES

Hyperbolic partial differential equations (PDEs) arise in many physical problems, typically whenever wave motion is observed. Acoustic waves, electromagnetic waves, seismic waves, shock waves, and many other types of waves can be modeled by hyperbolic equations. Often these are modeled by linear hyperbolic equations (for the propagation of sufficiently small perturbations), but modeling large motions generally requires solving nonlinear hyperbolic equations. Hyperbolic equations also arise in advective transport, when a substance is carried along with a flow, giving rise to an advection equation. This is a scalar linear first order hyperbolic PDE, the simplest possible case.

8.1 Wave equation and advection

The wave equation is:

Example 8.1 (Wave equation in 1d).

$$\begin{cases} u_{tt} = c^2 u_{xx} \\ u(x,0) = g_0(x) \\ u_t(x,0) = g_1(x). \end{cases} \tag{8.1}$$

With the D'Alemlert's formula, we can obatain the exact solution on the real line:

$$u(x,t) = \frac{g_0(x-ct) + g_0(x+ct)}{2} + \frac{1}{2c}\int_{x-ct}^{x+ct} g_1(\xi)\, d\xi.$$

In fact, we can reduce this PDE to one first order system. Let

$$v = \begin{pmatrix} u_x \\ u_t \end{pmatrix}, \quad v_t = \begin{pmatrix} u_{xt} \\ u_{tt} \end{pmatrix} = \begin{pmatrix} u_{xt} \\ c^2 u_{xx} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ c^2 & 0 \end{pmatrix} \begin{pmatrix} u_x \\ u_t \end{pmatrix}_x = Av_x,$$

where $A = \begin{pmatrix} 0 & 1 \\ c^2 & 0 \end{pmatrix}$. We can diagonalize A:

$$A = \underbrace{\begin{pmatrix} 1 & 1 \\ c & -c \end{pmatrix}}_{U} \underbrace{\begin{pmatrix} c & -c \end{pmatrix}}_{\Lambda} \underbrace{\begin{pmatrix} \frac{1}{2} & \frac{1}{2c} \\ \frac{1}{2} & -\frac{1}{z_c} \end{pmatrix}}_{U^{-1}}.$$

Let $w = U^{-1}v$, we have

$$Av_x = v_t \Longleftrightarrow \begin{cases} (w_1)_t = c(w_1)x \\ (w_2)_t = -c(w_2)x \end{cases}$$

Hence, the components of w decouple and satisfy the advection wave equation

$$u_t + au_x = 0. \tag{8.2}$$

For the Cauchy problem we also need initial data

$$u(x, 0) = \eta(x).$$

This is the simplest example of a **hyperbolic** equation, and it is so simple that we can write down the exact solution,

$$u(x, t) = \eta(x - at).$$

The first approach we might consider is the analogue of the method (9.4) for the heat equation. Using the centered difference in space,

$$u_x(x, t) = \frac{u(x + h, t) - u(x - h, t)}{2h} + O\left(h^2\right)$$

and the forward difference in time results in the numerical method

$$\frac{U_j^{n+1} - U_j^n}{k} = -\frac{a}{2h}\left(U_{j+1}^n - U_{j-1}^n\right),$$

which can be rewritten as

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}\left(U_{j+1}^n - U_{j-1}^n\right). \tag{8.3}$$

In practice this method is not useful because of stability considerations, as we will see in the next section.

A minor modification gives a more useful method. If we replace U_j^n on the righthand side of (8.3) by the average $\frac{1}{2}\left(U_{j-1}^n + U_{j+1}^n\right)$, then we obtain the **Lax-Friedrichs method**,

$$U_j^{n+1} = \frac{1}{2}\left(U_{j-1}^n + U_{j+1}^n\right) - \frac{ak}{2h}\left(U_{j+1}^n - U_{j-1}^n\right). \tag{8.4}$$

Because of the low accuracy, this method is not commonly used in practice, but it serves to illustrate some stability issues and so we will study this method along with (8.3) before describing higher order methods, such as the well-known Lax-Wendroff method.

We will see in the next section that Lax-Friedrichs is Lax-Richtmyer stable and convergent provided

$$\left|\frac{ak}{h}\right| \leqslant 1. \tag{8.5}$$

Note that this stability restriction allows us to use a time step $k = O(h)$ although the method is explicit, unlike the case of the heat equation. The basic reason is that the advection equation involves only the first order derivative u_x rather than u_{xx} and so the difference equation involves $1/h$ rather than $1/h^2$.

The time step restriction (8.5) is consistent with what we would choose anyway based on accuracy considerations, and in this sense the advection equation is not stiff, unlike the heat equation. This is a fundamental difference between hyperbolic equations and parabolic equations more generally and accounts for the fact that hyperbolic equations are typically solved with explicit methods, while the efficient solution of parabolic equations generally requires implicit methods.

To see that (8.5) gives a reasonable time step, note that

$$u_x(x, t) = \eta'(x - at)$$

while

$$u_t(x, t) = -au_x(x, t) = -a\eta'(x - at).$$

The time derivative u_t is larger in magnitude than u_x by a factor of a, and so we would expect the time step required to achieve temporal resolution consistent with the spatial resolution h to be smaller by a factor of a. This suggests that the relation $k \approx h/a$ would be reasonable in practice. This is completely consistent with (8.5) .

8.2 Method of lines discretization

To obtain a system of equations with finite dimension we must solve the equation on some bounded domain rather than solving the Cauchy problem. However, in a bounded domain, say, $0 \leqslant x \leqslant 1$, the advection equation can have a boundary condition specified on only one of the two boundaries. If $a > 0$, then we need a boundary condition at $x = 0$, say,

$$u(0, t) = g_0(t) \tag{8.6}$$

which is the **inflow** boundary in this case. The boundary at $x = 1$ is the **outflow** boundary and the solution there is completely determined by what is advecting to the right from the interior. If $a < 0$, we instead need a boundary condition at $x = 1$, which is the **inflow** boundary in this case.

The symmetric 3-point methods defined above can still be used near the inflow boundary but not at the outflow boundary. Instead the discretization will have to be coupled with some "numerical boundary condition" at the outflow boundary, say, a one-sided discretization of the equation. This issue complicates the stability analysis and will be discussed later.

For analysis purposes we can obtain a nice MOL discretization if we consider the special case of **periodic boundary conditions**,

$$u(0, t) = u(1, t) \text{ for } t \geqslant 0.$$

Physically, whatever flows out at the outflow boundary flows back in at the inflow boundary. This also models the Cauchy problem in the case where the initial data is periodic with period 1 , in which case the solution remains periodic and we need to model only a single period $0 \leqslant x \leqslant 1$.

In this case the value $U_0(t) = U_{m+1}(t)$ along the boundaries is another unknown, and we must introduce one of these into the vector $U(t)$. If we introduce $U_{m+1}(t)$, then we have the vector of grid values

$$U(t) = \begin{bmatrix} U_1(t) \\ U_2(t) \\ \vdots \\ U_{m+1}(t) \end{bmatrix}$$

For $2 \leqslant j \leqslant m$ we have the ordinary differential equation (ODE)

$$U_j'(t) = -\frac{a}{2h}\left(U_{j+1}(t) - U_{j-1}(t)\right)$$

while the first and last equations are modified using the periodicity:

$$\begin{aligned} U_1'(t) &= -\frac{a}{2h}\left(U_2(t) - U_{m+1}(t)\right), \\ U_{m+1}'(t) &= -\frac{a}{2h}\left(U_1(t) - U_m(t)\right). \end{aligned}$$

This system can be written as

$$U'(t) = AU(t) \tag{8.7}$$

with

$$A = -\frac{a}{2h}\begin{bmatrix} 0 & 1 & & & & -1 \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ 1 & & & & -1 & 0 \end{bmatrix} \in \mathbb{R}^{(m+1)\times(m+1)}. \tag{8.8}$$

Note that this matrix is skew-symmetric $\left(A^T = -A\right)$ and so its eigenvalues must be pure imaginary. In fact, the eigenvalues are

$$\lambda_p = -\frac{ia}{h}\sin(2\pi ph) \text{ for } p = 1, 2, \ldots, m+1.$$

The corresponding eigenvector u^p has components

$$u_j^p = e^{2\pi ipjh} \quad \text{for } j = 1, 2, \ldots, m+1$$

The eigenvalues lie on the imaginary axis between $-ia/h$ and ia/h. For absolute stability of a time discretization we need the stability region $\mathcal{S}$ to include this interval. Any method that includes some interval $iy, |y| < b$ of the imaginary axis will lead to a stable method for the advection equation provided $|ak/h| \leqslant b$.

8.2.1 Forward Euler time discretizaion

The method (8.3) can be viewed as the forward Euler time discretization of the MOL system of ODEs (8.7). We found in Section 7.3 that this method is stable only if $|1 + k\lambda| \leqslant 1$ and the stability region $\mathcal{S}$ is the unit circle centered at -1 . No matter how small the ratio k/h is, since the eigenvalues λ_p are imaginary, the values $k\lambda_p$ will not lie in $\mathcal{S}$. Hence the method (8.3) is unstable for any fixed mesh ratio k/h; see Figure [**Will: Todo**].

The method (8.3) will be convergent if we let $k \to 0$ faster than h, since then $k\lambda_p \to 0$ for all p and the zero-stability of Euler's method is enough to guarantee convergence. Taking k much smaller than h is generally not desirable and the method is not used in practice. However, it is interesting to analyze this situation also in terms of LaxRichtmyer stability, since it shows an example where the Lax-Richtmyer stability uses a weaker bound, $\|B\| \leqslant 1 + \alpha k$, rather than $\|B\| \leqslant 1$. Here $B = I + kA$. Suppose we take $k = h^2$, for example. Then we have

$$|1 + k\lambda_p|^2 \leqslant 1 + (ka/h)^2$$

for each p (using the fact that λ_p is pure imaginary) and so

$$|1 + k\lambda_p|^2 \leqslant 1 + a^2h^2 = 1 + a^2k$$

Hence $\|I + kA\|_2^2 \leqslant 1 + a^2k$ and if $nk \leqslant T$, we have

$$\|(I + kA)^n\|_2 \leqslant \left(1 + a^2k\right)^{n/2} \leqslant e^{a^2T/2},$$

showing the uniform boundedness of $\|B^n\|$ (in the 2-norm) needed for Lax-Richtmyer stability.

8.2.2 Leapfrog

A better time discretization is to use the midpoint method,

$$U^{n+1} = U^{n-1} + 2kAU^n$$

which gives the leapfrog method for the advection equation,

$$U_j^{n+1} = U_j^{n-1} - \frac{ak}{h}\left(U_{j+1}^n - U_{j-1}^n\right). \tag{8.9}$$

This is a 3-level explicit method and is second order accurate in both space and time.

Recall that the stability region of the midpoint method is the interval $i\alpha$ for $-1 < \alpha < 1$ of the imaginary axis. This method is hence stable on the advection equation provided $|ak/h| < 1$ is satisfied.

On the other hand, note that the $k\lambda_p$ will always be on the boundary of the stability region (the stability region for midpoint has no interior). This means the method is only marginally stable - there

is no growth but also no decay of any eigenmode. The difference equation is said to be nondissipative. In some ways this is good-the true advection equation is also nondissipative, and any initial condition simply translates unchanged, no matter how oscillatory. Leapfrog captures this qualitative behavior well.

However, there are problems with this. All modes translate without decay, but they do not all propagate at the correct velocity, as will be explained in Example [**Will: Todo**]. As a result initial data that contains high wave number components (e.g., if the data contains steep gradients) will disperse and can result in highly oscillatory numerical approximations. The marginal stability of leapfrog can also turn into instability if a method of this sort is applied to a more complicated problem with variable coefficients or nonlinearities.

8.2.3 Lax-Friedrichs

Again consider the Lax-Friedrichs method (8.4). Note that we can rewrite (8.4) using the fact that

$$\frac{1}{2}\left(U_{j-1}^n + U_{j+1}^n\right) = U_j^n + \frac{1}{2}\left(U_{j-1}^n - 2U_j^n + U_{j+1}^n\right)$$

to obtain

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}\left(U_{j+1}^n - U_{j-1}^n\right) + \frac{1}{2}\left(U_{j-1}^n - 2U_j^n + U_{j+1}^n\right). \tag{8.10}$$

This can be rearranged to give

$$\frac{U_j^{n+1} - U_j^n}{k} + a\left(\frac{U_{j+1}^n - U_{j-1}^n}{2h}\right) = \frac{h^2}{2k}\left(\frac{U_{j-1}^n - 2U_j^n + U_{j+1}^n}{h^2}\right)$$

If we compute the local truncation error from this form we see, as expected, that it is consistent with the advection equation $u_t + au_x = 0$, since the term on the right-hand side vanishes as $k, h \to 0$ (assuming k/h is fixed). However, it looks more like a discretization of the advection-diffusion equation

$$u_t + au_x = \epsilon u_{xx}$$

where $\epsilon = h^2/2k$. Later in this chapter we will study the diffusive nature of many methods for the advection equation. For our present purposes, however, the crucial part is that we can now view (8.10) as resulting from a forward Euler discretization of the system of ODEs

$$U'(t) = A_\epsilon U(t)$$

with

$$A_\epsilon = -\frac{a}{2h}\begin{bmatrix} 0 & 1 & & & & -1 \\ -1 & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 0 & 1 \\ 1 & & & & -1 & 0 \end{bmatrix} + \frac{\epsilon}{h^2}\begin{bmatrix} -2 & 1 & & & & 1 \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ 1 & & & & 1 & -2 \end{bmatrix}, \tag{8.11}$$

(8.12)

where $\epsilon = h^2/2k$. The matrix A_ϵ differs from the matrix A of (8.8) by the addition of a small multiple of the second difference operator, which is symmetric rather than skew symmetric. As a result the eigenvalues of A_ϵ are shifted off the imaginary axis and now lie in the left half-plane. There is now some hope that each $k\lambda$ will lie in the stability region of Euler's method if k is small enough relative to h.

It can be verified that the eigenvectors (10.12) of the matrix A are also eigenvectors of the second difference operator (with periodic boundary conditions) that appears in (10.15), and hence these are also the eigenvectors of the full matrix A_ϵ. We can easily compute that the eigenvalues of A_ϵ are

$$\mu_p = -\frac{ia}{h}\sin(2\pi ph) - \frac{2\epsilon}{h^2}(1 - \cos(2\pi ph)) \tag{8.13}$$

The values $k\mu_p$ are plotted in the complex plane for various different values of ϵ in Figure 8.2.3. They lie on an ellipse centered at $-2k\epsilon/h^2$ with semi-axes of length $2k\epsilon/h^2$ in the x-direction and ak/h in the y-direction. For the special case $\epsilon = h^2/2k$ used in Lax-Friedrichs, we have $-2k\epsilon/h^2 = -1$ and this ellipse lies entirely inside the unit circle centered at -1 , provided that $|ak/h| \leqslant 1$. (If $|ak/h| > 1$, then the top and bottom of the ellipse would extend outside the circle.) The forward Euler method is stable as a timediscretization, and hence the Lax-Friedrichs method is Lax-Richtmyer stable, provided $|ak/h| \leqslant 1$.

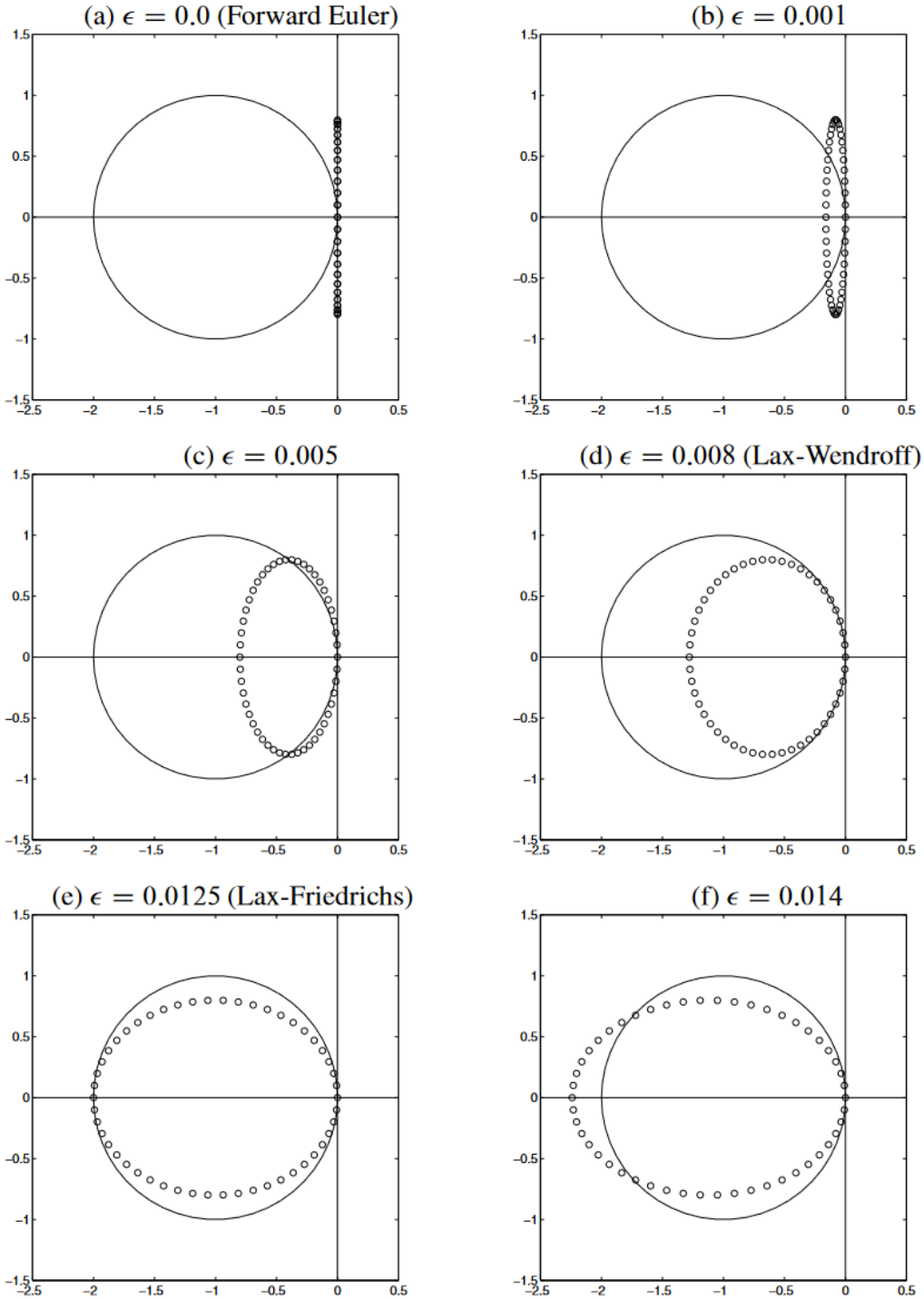


Figure 8.1: Eigenvalues of the matrix A_ϵ in , for various values of ϵ, in the case $h = 1/50$ and $k = 0.8h, a = 1$, so ak/h=0.8. (a) shows the case $\epsilon = 0$ which corresponds to the forward Euler method. (d) shows the case $\epsilon = a^2k/2$, the Lax-Wendroff method. (e) shows the case $\epsilon = h^2/2k$, the Lax-Friedrichs method. The method is stable for ϵ between $a^2k/2$ and $h^2/2k$, as in (d) through (e).

CHAPTER 9

LAX-WENDROFF METHOD AND UPWIND METHODS

One way to achieve second order accuracy on the advection equation is to use a second order temporal discretization of the system of ODEs (8.7) since this system is based on a second order spatial discretization. This can be done with the midpoint method, for example, which gives rise to the leapfrog scheme already discussed. However, this is a three-level method and for various reasons it is often much more convenient to use two-level methods for PDEs whenever possible-in more than one dimension the need to store several levels of data may be restrictive, boundary conditions can be harder to impose, and combining methods using fractional step procedures may require two-level methods for each step, to name a few reasons. Moreover, the leapfrog method is nondissipative, leading to potential stability problems if the method is extended to variable coefficient or nonlinear problems.

Another way to achieve second order accuracy in time would be to use the trapezoidal method to discretize the system, as was done to derive the Crank-Nicolson method for the heat equation. But this is an implicit method and for hyperbolic equations there is generally no need to introduce this complication and expense.

Another possibility is to use a two-stage Runge-Kutta method. This can be done, although some care must be exercised near boundaries, and the use of a multistage method again typically requires additional storage.

9.1 The Lax-Wendroff method

One simple way to achieve a two-level explicit method with higher accuracy is to use the idea of Taylor series methods. Applying this directly to the linear system of ODEs $U'(t) = AU(t)$ (and using $U'' = AU' = A^2U$) gives the second order method

$$U^{n+1} = U^n + kAU^n + \frac{1}{2}k^2A^2U^n.$$

Here A is the matrix(8.8) and computing A^2 and writing the method at the typical grid point then gives

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}\left(U_{j+1}^n - U_{j-1}^n\right) + \frac{a^2k^2}{8h^2}\left(U_{j-2}^n - 2U_j^n + U_{j+2}^n\right).$$

This method is second order accurate and explicit but has a 5-point stencil involving the points U_{j-2}^n and U_{j+2}^n. With periodic boundary conditions this is not a problem, but with other boundary conditions this method needs more numerical boundary conditions than a 3-point method. This makes it less convenient to use and potentially more prone to numerical instability.

Note that the last term is an approximation to $\frac{1}{2}a^2k^2u_{xx}$ using a centered difference based on step size $2h$. A simple way to achieve a second order accurate 3-point method is to replace this term by the

more standard 3-point formula. We then obtain the standard **Lax-Wendroff method**:

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}\left(U_{j+1}^n - U_{j-1}^n\right) + \frac{a^2k^2}{2h^2}\left(U_{j-1}^n - 2U_j^n + U_{j+1}^n\right). \tag{9.1}$$

A cleaner way to derive this method is to use Taylor series expansions directly on the PDE $u_t + au_x = 0$, to obtain

$$u(x, t+k) = u(x,t) + ku_t(x,t) + \frac{1}{2}k^2 u_{tt}(x,t) + \cdots.$$

Replacing u_t by $-au_x$ and u_{tt} by a^2u_{xx} gives

$$u(x, t+k) = u(x,t) - kau_x(x,t) + \frac{1}{2}k^2a^2 u_{xx}(x,t) + \cdots.$$

If we now use the standard centered approximations to u_x and u_{xx} and drop the higher order terms, we obtain the Lax-Wendroff method. It is also clear how we could obtain higher order accurate explicit two-level methods by this same approach, by retaining more terms in the series and approximating the spatial derivatives (including the higher order spatial derivatives that will then arise) by suitably high order accurate finite difference approximations. The same approach can also be used with other PDEs. The key is to replace the time derivatives arising in the Taylor series expansion with spatial derivatives, using expressions obtained by differentiating the original PDE.

We can analyze the stability of Lax-Wendroff following the same approach used for LaxFriedrichs. Note that with periodic boundary conditions, the Lax-Wendroff method can be viewed as Euler's method applied to the linear system of ODEs $U'(t) = A_\epsilon U(t)$, where A_ϵ is given by (8.8) with $\epsilon = a^2k/2$ (instead of the value $\epsilon = h^2/2k$ used in Lax-Friedrichs). The eigenvalues of A_ϵ are

$$k\mu_p = -i\left(\frac{ak}{h}\right)\sin(p\pi h) + \left(\frac{ak}{h}\right)^2(\cos(p\pi h) - 1).$$

These values all lie on an ellipse centered at $-(ak/h)^2$ with semi-axes of length $(ak/h)^2$ and $|ak/h|$. If $|ak/h| \leqslant 1$, then all of these values lie inside the stability region of Euler's method. Figure 8.2.3 shows an example in the case $ak/h = 0.8$. The Lax-Wendroff method is stable with exactly the same time step restriction as required for LaxFriedrichs. Later we will see that this is a very natural stability condition to expect for the advection equation and is the best we could hope for when a 3-point method is used.

A close look at Figure 8.2.3 shows that the values $k\mu_p$ near the origin lie much closer to the boundary of the stability region for the Lax-Wendroff method (Figure 8.2.3) than for the other methods illustrated in this figure. This is a reflection of the fact that LaxWendroff is second order accurate, while the others are only first order accurate. Note that a value $k\mu_p$ lying inside the stability region indicates that this eigenmode will be damped as the wave propagates, which is unphysical behavior since the true solution advects with no dissipation. For small values of μ_p (low wave numbers, smooth components) the LaxWendroff method has relatively little damping and the method is more accurate. Higher wave numbers are still damped with Lax-Wendroff (unless $|ak/h| = 1$, in which case all the $k\mu_p$ lie on the boundary of $\mathcal{S}$) and resolving the behavior of these modes properly would require a finer grid.

9.2 Upwind methods

So far we have considered methods based on symmetric approximations to derivatives. Alternatively, one might use a nonsymmetric approximation to u_x in the advection equation, e.g.,

$$u_x(x_j, t) \approx \frac{1}{h}(U_j - U_{j-1})$$

or

$$u_x(x_j, t) \approx \frac{1}{h}(U_{j+1} - U_j)$$

These are both one-sided approximations, since they use data only to one side or the other of the point x_j. Coupling one of these approximations with forward differencing in time gives the following methods for the advection equation:

$$U_j^{n+1} = U_j^n - \frac{ak}{h}\left(U_j^n - U_{j-1}^n\right) \tag{9.2}$$

or

$$U_j^{n+1} = U_j^n - \frac{ak}{h}\left(U_{j+1}^n - U_j^n\right). \tag{9.3}$$

These methods are first order accurate in both space and time. One might wonder why we would want to use such approximations, since centered approximations are more accurate.

For the advection equation, however, there is an asymmetry in the equations because the equation models translation at speed a. If $a > 0$, then the solution moves to the right, while if $a < 0$ it moves to the left. There are situations where it is best to acknowledge this asymmetry and use one-sided differences in the appropriate direction.

The choice between the two methods (9.2) and (9.3) should be dictated by the sign of a. Note that the true solution over one time step can be written as

$$u\left(x_j, t + k\right) = u\left(x_j - ak, t\right)$$

so that the solution at the point x_j at the next time level is given by data to the left of x_j if $a > 0$, whereas it is determined by data to the right of x_j if $a < 0$. This suggests that (9.2) might be a better choice for $a > 0$ and (9.3) for $a < 0$. In fact the stability analysis below shows that (9.2) is stable only if

$$0 \leqslant \frac{ak}{h} \leqslant 1$$

Since k and h are positive, we see that this method can be used only if $a > 0$. This method is called the **upwind** method when used on the advection equation with $a > 0$. If we view the equation as modeling the concentration of some tracer in air blowing past us at speed a, then we are looking in the correct upwind direction to judge how the concentration will change with time. (This is also referred to as an upstream differencing method in some literature.) Conversely, (9.3) is stable only if

$$-1 \leqslant \frac{ak}{h} \leqslant 0$$

and can be used only if $a < 0$. In this case (9.3) is the proper upwind method to use.

The method (9.2) can be written as

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}\left(U_{j+1}^n - U_{j-1}^n\right) + \frac{ak}{2h}\left(U_{j+1}^n - 2U_j^n + U_{j-1}^n\right),$$

which puts it in the form (8.11) with $\epsilon = ah/2$. We have seen previously that methods of this form are stable provided $|ak/h| \leqslant 1$ and also $-2 < -2\epsilon k/h^2 < 0$. Since $k, h > 0$, this requires in particular that $\epsilon > 0$. For Lax-Friedrichs and Lax-Wendroff, this condition was always satisfied, but for upwind the value of ϵ depends on a and we see that $\epsilon > 0$ only if $a > 0$. If $a < 0$, then the eigenvalues of the MOL matrix lie on a circle that lies entirely in the right half-plane, and the method will certainly be unstable. If $a > 0$, then the above requirements lead to the stability restriction.

The three methods, Lax-Wendroff, upwind, and Lax-Friedrichs, can all be written in the same form (8.11) with different values of ϵ. If we call these values $\epsilon_{LW}, \epsilon_{up}$, and ϵ_{LF}, respectively, then we have

$$\epsilon_{LW} = \frac{a^2k}{2} = \frac{ahv}{2}, \quad \epsilon_{up} = \frac{ah}{2}, \quad \epsilon_{LF} = \frac{h^2}{2k} = \frac{ah}{2v},$$

where $v = ak/h$. Note that

$$\epsilon_{LW} = v\epsilon_{up} \quad \text{and} \quad \epsilon_{up} = v\epsilon_{LF}$$

If $0 < v < 1$, then $\epsilon_{LW} < \epsilon_{up} < \epsilon_{LF}$ and the method is stable for any value of ϵ between ϵ_{LW} and ϵ_{LF}.

9.2.1 The Beam-Warming method

The upwind method is only first order accurate. A second order accurate method with the same one-sided character can be derived by following the derivation of the Lax-Wendroff method, but using one-sided approximations to the spatial derivatives. This results in the **Beam-Warming** method, which for $a > 0$ takes the form

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}\left(3U_j^n - 4U_{j-1}^n + U_{j-2}^n\right) + \frac{a^2k^2}{2h^2}\left(U_j^n - 2U_{j-1}^n + U_{j-2}^n\right).$$

For $a < 0$ the Beam-Warming method is one-sided in the other direction:

$$U_j^{n+1} = U_j^n - \frac{ak}{2h}\left(-3U_j^n + 4U_{j+1}^n - U_{j+2}^n\right) + \frac{a^2k^2}{2h^2}\left(U_j^n - 2U_{j+1}^n + U_{j+2}^n\right).$$

These methods are stable for $0 \leqslant v \leqslant 2$ and $-2 \leqslant v \leqslant 0$, respectively.