Supplementary Notes 8

Tiago Salvador (tiago.saldanhasalvador@mail.mcgill.ca)

Abstract

We discuss numerical methods for initial value problems.

Contents

1 Numerical method for initial value problems

In the following we are interested in solving the initial value problem

$$\begin{cases} y'(t) = f(t, y(t)), \quad t \in [0, T], \\ y(0) = y_0, \end{cases} \tag{IVP}$$

where f and y_0 (initial value) are both given. In a ODE course, one studies how to find the exact solution of (IVP) for special forms of f. However, exact solutions might not be available or are impossible to compute. In such case, we look for approximate solutions. Here we will focus on obtaining approximations for y at various values, called mesh points or time steps, in the interval $[0, T]$, instead of obtaining a continuous approximation to the solution y. Once we know the approximation solution at the mesh points, we can use interpolation to get an approximation of the solution in the entire interval.

Usually the mesh points are equally spaced in the interval. Fixing the number of mesh points as $N + 1$, they are given by

$$t_n = nh \quad n = 0, 1, \ldots, N$$

where $h = \frac{T}{N} = t_{n+1} - t_n$ is called the step size. We will denote by y_n the approximation of $y(t_n)$.

To solve (IVP) numerically, we want to "march in time": at each time step t_n we solve an algebraic equation for y_n. The discrete set of algebraic equations is called a discretization and it can be seen as an approximation of the ODE in (IVP).

We will discuss two approaches to generate discretizations.

1.1 Numerical differentiation approach

To generate a discretization we can approximate y' by $D_h y$. We give two examples.

Example 1.1 (Forward Euler)**.** *The forward Euler method can be deduced by approximating y' with a forward difference, i.e.,*

$$y'(t_n) \approx D_h y(t_n) = \frac{y(t_{n+1}) - y(t_n)}{h},$$

Hence

$$\frac{y(t_{n+1}) - y(t_n)}{h} \approx f(t_n, y(t_n))$$

and so we obtain the discrete equation

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

which can be rewritten as

$$y_{n+1} = y_n + hf(t_n, y_n), \quad n = 0, \ldots, N-1.$$

This is an explicit 1*-step method.*

Example 1.2 (Backward Euler)**.** *The backward Euler method is similar to the forward Euler but we use a backward difference to approximate y', i.e.,*

$$y'(t_n) \approx D_h y(t_n) = \frac{y(t_n) - y(t_{n-1})}{h},$$

This leads to

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}), \quad n = 0, \ldots, N-1.$$

This is an implicit 1*-step method.*

Example 1.3 (Midpoint method)**.** *Using the centred difference to approximate y', i.e.,*

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

leads to

$$y_{n+1} = y_{n-1} + 2hf(t_n, y_n), \quad n = 0, \ldots, N-1.$$

This is an explicit 2*-step method.*

1.2 Numerical integration approach

From the fundamental theorem of calculus we can write (IVP) in an integral form as

$$y(t) = y_0 + \int_0^t f(s, y(s))\, ds$$

More generally,

$$y(t_{n+1}) = y(r) + \int_r^{t_{n+1}} f(s, y(s))\, ds$$

Hence, we can use numerical quadratures to obtain discretizations.

Example 1.4 (Trapezoid Method)**.** *The Trapezoid Rule tells us that*

$$\int_{t_n}^{t_{n+1}} f(s, y(s))\, ds \approx \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, y_{n+1})$$

and so

$$y(t_{n+1}) \approx y(t_n) + \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, y_{n+1}).$$

This leads to

$$y_{n+1} = y_n + \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, y_{n+1})), \quad n = 0, \ldots, N-1,$$

which is an implicit 1*-step method.*

Example 1.5 (Midpoint Method)**.** *The Midpoint rule tells us that*

$$\int_{t_{n-1}}^{t_{n+1}} f(s,y(s))\,ds \approx 2hf(t_n,y_n)$$

and therefore we get

$$y_{n+1} = y_{n-1} + 2hf(t_n,y_n), \quad n = 1,\dots,N-1.$$

This is an implicit 1*-step method.*

Example 1.6 (Milne-Simpson Method)**.** *The Simpson's quadrature tells us that*

$$\int_{t_{n-1}}^{t_{n+1}} f(s,y(s))\,ds \approx \frac{2h}{6}\left(f(t_{n-1},y(t_{n-1})) + 4f(t_n,y(t_n)) + f(t_{n+1},y(t_{n+1}))\right)$$

and therefore we get

$$y_{n+1} = y_{n-1} + \frac{h}{3}\left(f(t_{n-1},y_{n-1}) + 4f(t_n,y_n) + f(t_{n+1},y_{n+1})\right), \quad n = 1,\dots,N-1.$$

This is an implicit 1*-step method.*

Definition 1.1. *A discretization is called explicit if y_{n+1} can be solved explicitly. Otherwise, it is called implicit.*

Remark 1.2. *In general, implicit methods are more accurate and stable (a notion we will make precise later), but in order to apply them we need to solve the implicit equation. This is not always possible and so implicit methods usually require the use of a root find method, thus becoming more computationally expensive.*

Definition 1.3. *A discretization is called k-step if the method requires knowing k previous $y_n,\dots,y_{n-k+1}$ simultaneously to compute y_{n+1}.*

1.3 Local truncation error and convergence

Our goal in this section is to quantify the error of our numerical methods and discuss convergence, i.e., does the error go to 0 as h goes to 0.

For each $k - step$ discretization, we can rewrite the equation as

$$\Phi_h(t_n, y_{n+1}, y_n, \dots, y_{n-k+1}) = 0.$$

Example 1.7. *For the forward Euler method we have*

$$y_{n+1} = y_n + hf(t_n,y_n) \Longleftrightarrow y_{n+1} - y_n - hf(t_n,y_n) = 0$$

To analyze the error of each of the discretizations, we need a new concept of error. We start with the local truncation error.

Definition 1.4. *The local truncation error at t_n, $\tau_h(t_n)$, is*

$$\tau_h(t_n) = \Phi_h(t_n, y(t_{n+1}), y(t_n), \dots, y(t_{n-k+1})),$$

where $y(t)$ denotes the exact solution of (IVP)*.*

Remark 1.5. *This error is a local error since it measures the accuracy of the method at a specific time step, assuming we know the exact solution at the previous step. In other words, it measures the amount by which the exact solution fails the discretization.*

Definition 1.6. *Denote*

$$\tau_h := \max_{n=0,\dots,N-1} |\tau_h(t_n)|.$$

- $\frac{\tau_h}{h}$ *is called the consistency error or global truncation error.*
- *A method is of order p if*
$$\frac{\tau_h}{h} = \mathcal{O}(h^p).$$
- *A method is called consistent if*
$$\lim_{h\to 0} \frac{\tau_h}{h} = 0.$$

Example 1.8. *The forward Euler method gives us*

$$\Phi_h(t_n, y_n) = y_{n+1} - y_n - hf(t_n, y_n).$$

Therefore it has a local truncation error at t_n given by

$$\tau_h(t_n) = y(t_{n+1}) - y(t_n) - hf(t_n, y(t_n))$$

for each $i = 0, \ldots, N-1$. If we Taylor expand $y(t)$ about $t = t_n$ and evaluate at $t = t_{n+1}$, we get

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2} y''(\xi_n)$$

where $\xi_n \in (t_n, t_{n+1})$. Hence, since $f(t_n, y(t_n)) = y'(t_n)$, we get

$$\tau_h(t_n) = \frac{h^2}{2} y''(\xi_n).$$

Let M_2 be given by

$$M_2 = \max_{0\le t\le T} |y''(t)|$$

Then

$$|\tau_h(t_n)| \le \frac{M_2}{2} h^2$$

and so the local truncation error in the forward Euler method is $\mathcal{O}(h^2)$. Moreover, the forward Euler method is consistent and it has order 1.

Exercise 1.1. *Show that the local truncation error of the trapezoidal method is $\mathcal{O}(h^3)$.*

Solution: The Trapezoidal method is given by

$$y_{n+1} = y_n + \frac{h}{2}(f(t_n, y_n) + f(t_{n+1}, y_{n+1})), \quad n = 0, \ldots, N-1.$$

The local truncation error is then given by

$$\tau_{n+1}(h) = y(t_{n+1}) - y(t_n) - \frac{h}{2}\left(f(t_n, y(t_n)) + f(t_{n+1}, y(t_{n+1}))\right)$$

for $n = 1, 2 \ldots, N-1$. Since $y(\cdot)$ is a solution of (IVP), we can rewrite it as

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

We will use the Taylor expansion to simplify the above expression. We have

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2} y''(t_n) + \mathcal{O}(h^3)$$

and

$$y'(t_{n+1}) = y'(t_n) + hy''(t_n) + \mathcal{O}(h^2).$$

Hence

$$\begin{aligned}\tau_{n+1}(h) &= y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(t_n) + \mathcal{O}(h^3) - y(t_n) - \frac{h}{2}\left(y'(t_n) + y'(t_n) + hy''(t_n) + \mathcal{O}(h^2)\right) \\ &= \mathcal{O}(h^3)\end{aligned}$$

Definition 1.7. *Denote the error at t_n as $e_n = y(t_n) - y_n$. A method converges if*

$$\lim_{h\to 0} \max_{n=0,\ldots,N} |e_n| = 0.$$

Theorem 1.8. *If for all fixed t,*

$$\left|\frac{\partial f}{\partial z}(t,z)\right| \leq L$$

for all z, then the forward Euler method is convergent.

Remark 1.9. *The above result is also true for other 1-step methods. However, for k-step methods, consistency will not be enough in general to guarantee convergence, we will need our methods to be stable (a concept we will later).*

References

[1] R. L. Burden and J. D. Faires. *Numerical Analysis.* 9^{th} edition. Brookes/Cole, 2004.

[2] A. Quarteroni, R. Sacco and F. Saleri. *Numerical Mathematics.* 2^{nd} edition. Springer, 2006.

[3] Tutorial notes written by Jan Feys

[4] Class notes written by Andy Wan for Math 317 Fall 2014.