Numerical solution of ordinary differential equations: One step methods

André Massing

Oct 3, 2021

The Python codes for this note are given in `ode.py`.

1 One Step Methods

In the last lecture, we introduced the explicit Euler method and Heun's method. Both methods only need to know f, τ_k and the solution y_k at the *current* point t_k, but not at earlier points $t_{k-1}, t_{k-2}, \ldots$. This motivates the following definition.

Definition 1.1. *One step methods.*

A one step method defines an approximation to the IVP in the form of a discrete function $\boldsymbol{y}_\Delta : \{t_0, \ldots, t_N\} \to \mathbb{R}^n$ given by

$$\boldsymbol{y}_{k+1} := \boldsymbol{y}_k + \tau_k \Phi(t_k, \boldsymbol{y}_k, \boldsymbol{y}_{k+1}, \tau_k) \tag{1}$$

for some **increment function**

$$\Phi : [t_0, T] \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^n.$$

The OSM is called **explicit** if the increment function Φ does not depend on $\boldsymbol{y}_{k+1}$, otherwise it is called **implicit**.

Example 1.1. *Increment functions for Euler and Heun.*

The increment functions for Euler's and Heun's method are defined by respectively

$$\Phi(t_k, y_k, y_{k+1}, \tau_k) = f(t_k, y_k), \qquad \Phi(t_k, y_k, y_{k+1}, \tau_k) = \tfrac{1}{2}\left(f(t_k, y_k) + f\big(t_{k+1}, y_k + \tau_k f(t_k, y_k)\big)\right).$$

1.1 Local and global truncation error of OSM

Definition 1.2. *Local truncation error.*

The **local truncation error** $\eta(t,\tau)$ is defined by

$$\eta(t,\tau) = y(t) + \tau\Phi(t, y(t), y(t+\tau), \tau) - y(t+\tau). \tag{2}$$

$\eta(t,\tau)$ is often also called the **local discretization** or **consistency error**.

A one step method is called **consistent of order** $p \in \mathbb{N}$ if there is a constant $C > 0$ such that

$$|\eta(t,\tau)| \leqslant C\tau^{p+1} \quad \text{for } \tau \to 0. \tag{3}$$

A short-hand notation for this is to write $\eta(t,\tau) = \mathcal{O}(\tau^{p+1})$ for $\tau \to 0$.

Example 1.2. .

Euler's method has consistency order $p = 1$.

Definition 1.3. *Global truncation error.*

For a numerical solution $y_\Delta : \{t_0, \ldots, t_N\} \to \mathbb{R}$ the **global truncation error** is defined by

$$e_k(t_k, \tau_k) = y(t_k) - y_k \quad \text{for } k = 0, \ldots, N. \tag{4}$$

A one step method is called **convergent with order** $p \in \mathbb{N}$ if

$$\max_{k \in \{0,1,\ldots,N_t\}} |e_k(t_k, \tau_k)| = \mathcal{O}(\tau^p) \tag{5}$$

with $\tau = \max_k \tau_k$.

Discussion. If a one step method has convergence order equal to p, the maximum error $e(\tau) = \max_k |e(t_k, \tau)|$ can be thought as a function of the step size τ is of the form

$$e(\tau) = O(\tau^p) \leqslant C\tau^p.$$

This implies that if we change the time step size from τ to $\frac{\tau}{2}$, we can expect that the error decreases from $C\tau^p$ to $C(\frac{\tau}{2})^p$, that is, the error will be reduced by a factor 2^{-p}.

How can we determine the convergence rate by means of numerical experiments? Starting from $e(\tau) = O(\tau^p) \leqslant C\tau^p$ and taking the logarithm gives

$$\log(e(\tau)) \leqslant p \log(\tau) + \log(C).$$

Thus $\log(e(\tau))$ is a linear function of $\log(\tau)$ and the slope of this linear function corresponds to the order of convergence p.

So if you have an *exact solution* at your disposal, you can for an increasing sequence `Nmax_list` defining a descreasing sequence of *maximum* time-steps $\{\tau_0, \ldots, \tau_M\}$ and solve your problem numerically and then compute the resulting exact error $e(\tau_i)$ and plot it against τ_i in a $\log - \log$ plot to determine the convergence order.

In addition you can also compute the experimentally observed convergence rate EOC for $i = 1, \ldots M$ defined by

$$\mathrm{EOC}(i) = \frac{\log(e(\tau_i)) - \log(e(\tau_{i-1}))}{\log(\tau_i) - \log(\tau_{i-1})} = \frac{\log(e(\tau_i)/e(\tau_{i-1}))}{\log(\tau_i/\tau_{i-1})}$$

Ideally, $\mathrm{EOC}(i)$ is close to p.

This is implemented in the following `python` function.

```
def compute_eoc(y0, t0, T, f, Nmax_list, solver, y_ex):
    errs = [ ]
    for Nmax in Nmax_list:
        ts, ys = solver(y0, t0, T, f, Nmax)
        ys_ex = y_ex(ts)
        errs.append(np.abs(ys - ys_ex).max())
        print("For Nmax = {:3}, max ||y(t_i) - y_i||= {:.3e}".format(Nmax,errs[-1]))

    errs = np.array(errs)
    Nmax_list = np.array(Nmax_list)
    dts = (T-t0)/Nmax_list

    eocs = np.log(errs[1:]/errs[:-1])/np.log(dts[1:]/dts[:-1])

    # Insert inf at beginning of eoc such that errs and eoc have same length
    eocs = np.insert(eocs, 0, np.Inf)
    return errs, eocs
```

Exercise 1: Convergence order for Euler and Heun

Use the `compute_eoc` function and any of the examples with a known analytical solution from the previous lecture to determine convergence order for Euler's and Heun's method.

Solution. The solution to this exercise is implemented in the `exercise_eoc_study` function as part of the `ode.py` file. You can also find it in the jupyter notebook version of the this lecture note.

1.2 A general convergence result for one step methods

Theorem 1.1. *Convergence of one-step methods.*

Assume that there exist positive constants M and D such that the increment function satisfies

$$\|\mathbf{\Phi}(t, \mathbf{y}; \tau) - \mathbf{\Phi}(t, \mathbf{z}; \tau)\| \leq M \|\mathbf{y} - \mathbf{z}\|$$

and the local trunctation error satisfies

$$\|\boldsymbol{\eta}(t, \tau)\| = \|\mathbf{y}(t+\tau) - (\mathbf{y}(t) + \tau \mathbf{\Phi}(t, \mathbf{y}(t), \tau))\| \leqslant D\tau^{p+1}$$

for all t, $\mathbf{y}$ and $\mathbf{z}$ in the neighbourhood of the solution. In that case, the global error satisfies

$$\max_{k \in \{0,1,\ldots,N_t\}} \|e_k(t_k, \tau_k)\| \leqslant C\tau^p, \qquad C = \frac{e^{M(T-t_0)} - 1}{M} D,$$

where $\tau = \max_{k \in \{0,1,\ldots,N_t\}} \tau_k$.

Proof. We omit the proof here.

It can be proved that the first of these conditions are satisfied for all the methods that will be considered here.

Summary.

The convergence theorem for one step methods can be summarized as

"local truncation error behaves like $\mathcal{O}(\tau^{p+1})$" + "Increment function satisfies a Lipschitz condition" $\Rightarrow$ "global truncation error behaves like $\mathcal{O}(\tau^p)$"

or equivalently,

"consistency order p" + "Lipschitz condition for the Increment function" $\Rightarrow$ "convergence order p.

1.3 Convergence properties of Heun's method

Thanks to Theorem 1.1, we need to show two things to prove convergence and find the corresponding convergence of a given one step methods:

- determine the local truncation error, expressed as a power series in the step size τ
- the condition $\|\mathbf{\Phi}(t, \boldsymbol{y}, \tau) - \mathbf{\Phi}(t, \boldsymbol{z}, \tau)\| \leqslant M \|\boldsymbol{y} - \boldsymbol{z}\|$

Determining the consistency order. The local truncation error is found by making Taylor expansions of the exact and the numerical solutions starting from the same point, and compare. In practice, this is not trivial. For simplicity, we will here do this for a scalar equation $y'(t) = f(t, y(t))$. The result is valid for systems as well

In the following, we will use the notation

$$f_t = \frac{\partial f}{\partial t}, \qquad f_y = \frac{\partial f}{\partial y}, \qquad f_{tt} = \frac{\partial^2 f}{\partial t^2} \qquad f_{ty} = \frac{\partial^2 f}{\partial t \partial y} \qquad \text{etc.}$$

Further, we will surpress the arguments of the function f and its derivatives. So f is to be understood as $f(t, y(t))$ although it is not explicitly written.

The Taylor expansion of the exact solution $y(t+\tau)$ is given by

$$y(t+\tau) = y(t) + \tau y'(t) + \frac{\tau^2}{2} y''(t) + \frac{\tau^3}{6} y'''(t) + \cdots .$$

Higher derivatives of $y(t)$ can be expressed in terms of the function f by using the chain rule and the product rule for differentiation.

$$\begin{aligned}
y'(t) &= f, \\
y''(t) &= f_t + f_y y' = f_t + f_y f, \\
y'''(t) &= f_{tt} + f_{ty} y' + f_{yt} f + f_{yy} y' f + f_y f_t + f_y f_y y' = f_{tt} + 2 f_{ty} f + f_{yy} f^2 + f_y f_t + (f_y)^2 f.
\end{aligned}$$

Find the series of the exact and the numerical solution around x_0, y_0 (any other point will do equally well). From the discussion above, the series for the exact solution becomes

$$y(t_0+\tau) = y_0 + \tau f + \frac{\tau^2}{2}(f_t + f_y f) + \frac{\tau^3}{6}(f_{tt} + 2 f_{ty} f + f_{yy} f f + f_y f_x f + f_y f_t + (f_y)^2 f) + \cdots ,$$

where f and all its derivatives are evaluated in (t_0, y_0). For the numerical solution we get

$$\begin{aligned}
k_1 &= f(t_0, y_0) = f, \\
k_2 &= f(t_0 + \tau, y_0 + \tau k_1) \\
&= f + \tau f_t + f_y \tau k_1 + \frac{1}{2} f_{tt} \tau^2 + f_{ty} \tau \tau k_1 + \frac{1}{2} f_{yy} \tau^2 k_1^2 + \cdots \\
&= f + \tau (f_t + f_y f) + \frac{\tau^2}{2}(f_{tt} + 2 f_{ty} f + f_{yy} f^2) + \cdots , \\
y_1 &= y_0 + \frac{\tau}{2}(k_1 + k_2) = y_0 + \frac{\tau}{2}(f + f + \tau(f_t + f_y f) + \frac{\tau^2}{2}(f_{tt} + 2 f_{ty} k_1 + f_{yy} f^2)) + \cdots \\
&= y_0 + \tau f + \frac{\tau^2}{2}(f_t + f_y f) + \frac{\tau^3}{4}(f_{tt} + 2 f_{ty} f + f_{yy} f^2) + \cdots
\end{aligned}$$

and the local truncation error will be

$$\eta(t_0, \tau) = y(t_0 + \tau) - y_1 = \frac{\tau^3}{12}(-f_{tt} - 2 f_{ty} f - f_{yy} f^2 + 2 f_y f_t + 2 (f_y)^2 f) + \cdots$$

The first nonzero term in the local truncation error series is called *the principal error term*. For τ sufficiently small this is the term dominating the error, and this fact will be used later.

Although the series has been developed around the initial point, series around $x_n, y(t_n)$ will give similar results, and it is possible to conclude that, given sufficient differentiability of f there is a constant D such that

$$\max_i |\eta(t_i, \tau)| \le D \tau^3.$$

Consequently, Heun's method is of consistency order 2.

Lipschitz condition for Φ. Further, we have to prove the condition on the increment function $\Phi(t, y)$. For f differentiable, there is for all y, z some ξ between x and y such that $f(t, y) - f(t, z) = f_y(t, \xi)(y - z)$. Let L be a constant such that $|f_y| < L$, and for all x, y, z of interest we get

$$|f(t, y) - f(t, z)| \le L |y - z|.$$

The increment function for Heun's method is given by

$$\Phi(t, y) = \frac{1}{2}(f(t, y) + f(t + \tau, y + \tau f(t, y))).$$

By repeated use of the condition above and the triangle inequalitiy for absolute values we get

$$\begin{aligned}
|\Phi(t,y)-\Phi(t,z)| &= \frac{1}{2}|f(t,y)+f(t+\tau,y+f(t,y))-f(t,z)-\tau f(t+\tau,z+f(t,z)| \\
&\leq \frac{1}{2}\big(|f(t,y)-f(t,z)|+|f(t+\tau,y+\tau f(t,y))-f(t+\tau,z+\tau f(t,z)|\big) \\
&\leq \frac{1}{2}\big(L|y-z|+L|y+\tau f(t,y)-z-\tau f(t,z)|\big) \\
&\leq \frac{1}{2}\big(2L|y-z|+\tau L^2|y-z|\big) \\
&= (L+\frac{\tau}{2}L^2)|y-z|.
\end{aligned}$$

Assuming that the step size τ is bounded upward by some τ_0, we can conclude that

$$|\Phi(t,y)-\Phi(t,z)| \leq M|y-z|, \qquad M = L+\frac{\tau_0}{2}L^2.$$

Thanks to Theorem 1.1, we can conclude that Heun's method is convergent of order 2.