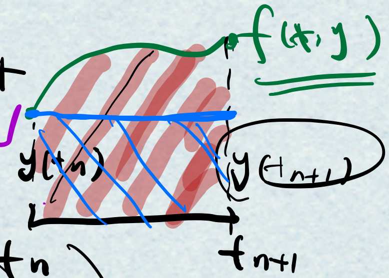
$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(t, y(t))\,dt$$



2.2 (Forward) Euler's Method

The simplest method to the integral in (2) is to use a constant for the $f(t, y(t))$, for example we let the constant to be $f(t_n, \underset{\text{know}}{\overset{y_n}{y(t_n)}})$.

we know $y_n = y(t_n)$

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

And this is the (forward or explicit) Euler's method.

If we use ϕ_n to denote the approximate solution of the IVP at t_n, then the Euler's method has formula note $\boxed{\phi_{n+1} \approx y(t_{n+1})}$

Forward

$$\phi_{n+1} = \phi_n + (t_{n+1} - t_n) f(t_n, \phi_n). \quad (3)$$

explicit method

We note this is one step method, and it is *explicit in time* method.

as $f(t, y) = (1-2t)y$

Example: Solve the IVP $\boxed{y' = (-2t + 1)y,}$ $y(0) = 1$ using forward Euler's method with $h = 0.2$. (fixed h means equally spaced time: $t_{n+1} - t_n = h$) $f(t, y) = (-2t+1)y$

Answer: We have $f(t_n, \phi_n) = (1 - 2t_n)\phi_n$, so

plug $t_n \to t$, $\phi_n \to y$ we get

$$\phi_{n+1} = \phi_n + h(1 - 2t_n)\phi_n = \phi_n(1 + h(1 - 2t_n)).$$

$t_0 = 0$, $t_1 = 0.2$, $t_2 = 0.4$, and $t_3 = 0.6$ if $h = 0.2$

$f(t_n, \phi_n) = (1-2t_n)\phi_n$, So,

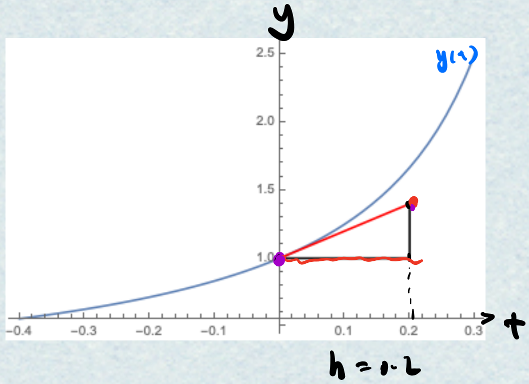
$\phi_{n+1} = \phi_n + (t_{n+1} - t_n) f(t_n, \phi_n)$

$t_{n+1} - t_n = h$

$\phi_{n+1} = \phi_n + h\boxed{(1-2t_n)\phi_n}$

$f(t_n, \phi_n) = y'(t_n)$

I.C. $y(0)=1$

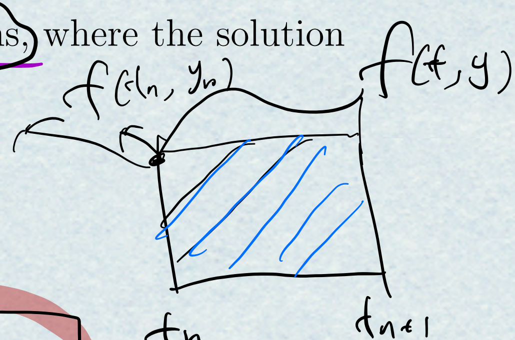


Euler's method

n	t_n	$f(t_n, \phi_n) = (1-2t_n)\phi_n$	$\phi_{n+1} = \phi_n + (t_{n+1} - t_n) f(t_n, \phi_n)$
0	0	1	–
1	0.2	$(1-2\cdot 0.2)\cdot 1.2 = 0.72$	$1 + 0.2(1) = 1.2$
2	0.4	$(1-2\cdot 0.4)\cdot 1.344 = 0.2688$	$1.2 + 0.2(0.72) = 1.344$

We should avoid using Euler's method for stiff problems, where the solution changes rapidly.

forward



2.3 Backward Euler's Method

y_{n+1} y_n

Recall the integral in (2), $y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y(t))dt$, we can approximate the integral by using the value of $f(t_{n+1}, y(t_{n+1}))$ instead of $f(t_n, y(t_n))$, and this leads to the (backward or implicit) Euler's method,

$\approx \int_{t_n}^{t_{n+1}} f(t, y(t))dt$

$$y(t_{n+1}) \approx y(t_n) + \boxed{(t_{n+1} - t_n) f(t_{n+1}, y(t_{n+1}))}. \tag{4}$$

If ϕ_n is approx to $y(t_n)$

$\phi_{n+1} \sim y(t_{n+1})$

$\phi_{n+1} = \phi_n + (t_{n+1} - t_n) f(t_{n+1}, \phi_{n+1})$

If we use ϕ_n to denote the approximate solution of the IVP at t_n, then the backwards Euler's method has formula is

Backward Euler

I.E.

$$\phi_{n+1} = \phi_n + (t_{n+1} - t_n) f(t_{n+1}, \phi_{n+1}).$$

this usually gives us non-linear eqn about ϕ_{n+1}

We note this is one step method, and it is *implicit in time* method. The implicit nature of the method makes it more stable than the explicit method. But we may need to solve a nonlinear equation to get the value ϕ_{n+1}.

$f(t, y) = (1-2t)y$ $\qquad$ $f(t_n, y_n) = (1-2t_n)y_n$

For example, if we have $y' = (1 - 2t)y$, $y(0) = 1$, then the backward Euler's method is (we get linear equation for ϕ_{n+1} in this case) $f(t_n, \phi_n) = (1-2t_n)\phi_n$

$f(t_{n+1}, \phi_{n+1}) =$ $\qquad$ $f(t_{n+1}, \phi_{n+1}) = (1-2t_{n+1})\phi_{n+1}$

$$\phi_{n+1} = \phi_n + h(1 - 2t_{n+1})\phi_{n+1} \Rightarrow \phi_{n+1} = \frac{\phi_n}{1 - h(1 - 2t_{n+1})}.$$

I.E.

$f(t_{n+1}, \phi_{n+1})$ $\qquad$ $f(t_{n+1}, \phi_{n+1})$ $\qquad$ B.E. method for ϕ_{n+1}

2.4 Error Analysis

We can define the **local truncation error** or the **one step error** at time t_n, as the error made in one step of the method, i.e., the error made in approximating $y(t_{n+1})$ by ϕ_{n+1}, as we know the true solution $y(t_n)$ to compute the one step approximation ϕ^*_{n+1},

$$\tau_{n+1} = y(t_{n+1}) - \phi^*_{n+1}.$$

If we use ϕ_n to denote the approximate solution of the IVP at t_n, then the backwards Euler's method has formula is

this is an eqn for ϕ_{n+1}

$$\boxed{\phi_{n+1} = \phi_n + (t_{n+1} - t_n) f(t_{n+1}, \phi_{n+1}).} \tag{5}$$

depend on $f(t, y)$ formula

We note this is one step method (the method only involves values at steps n and $n+1$), and it is *implicit in time* method. The implicit nature of the method makes it more stable than the explicit methods. But we **may need to solve a nonlinear equation** to get the value ϕ_{n+1}. If (5) is a nonlinear equation, we can use Newton's method to solve it: we introduce a function $F(\phi_{n+1}) = \phi_n + hf(t_{n+1}, \phi_{n+1}) - \phi_{n+1}$, and we solve $F(\phi_{n+1}) = 0$ using Newton's method. The iteration is $\phi_{n+1}^{(0)} = \phi_n$ and

$$\boxed{\phi_{n+1}^{(k+1)} = \phi_{n+1}^{(k)} - \frac{F(\phi_{n+1}^{(k)})}{F'(\phi_{n+1}^{(k)})},}$$

Newton's method

$$x_{n+1} = x_n - \frac{F(x_{n+1})}{F'(x_{n+1})}$$

where $F'(z)$ is obtained by taking the derivative of $F(z) = \phi_n + hf(t_{n+1}, z) - z$ with respect to z. This gives $F'(z) = h\frac{\partial f(t_{n+1}, z)}{\partial z} - 1$.

For simple example (this does not need to solve nonlinear equation to get ϕ_{n+1}), if we have $y' = (1-2t)y$, $y(0) = 1$, then the backward Euler's method is (we get linear equation for ϕ_{n+1} in this case)

$$\phi_{n+1} = \phi_n + h(1 - 2t_{n+1})\phi_{n+1} \Rightarrow \phi_{n+1} = \frac{\phi_n}{1 - h(1 - 2t_{n+1})}.$$

$\phi_n \quad \phi_{n+1} \simeq y(t_{n+1})$

$t_0 \quad t_n \quad t_{n+1}$

2.4 Error Analysis

We can define the **local truncation error** or the **one step error** at time t_n, as the error made in one step of the method, i.e., the error made in approximating $y(t_{n+1})$ by ϕ_{n+1}, as we know the true solution $y(t_n)$ to compute the one step approximation ϕ^*_{n+1},

analytical

$$\tau_{n+1} = y(t_{n+1}) - \phi^*_{n+1}.$$

one step approx using numerical method w/ starting true sol. $y(t_n)$

For example, if we consider the forward Euler's method in (3), then the ϕ^*_{n+1} is (note here $y(t_n)$ is the true solution value at t_n)

$\phi_{n+1} = \phi_n + hf(t_n, \phi_n)$

$$\phi^*_{n+1} = y(t_n) + hf(t_n, y(t_n)).$$

$\phi^*_{n+1} = y(t_n) + hf(t_n, y(t_n))$

The local truncation error is then ($y(t_{n+1})$ is the true solution value at t_{n+1})

analytical

f in the RHS of ODE $y' = f(t, y)$

$$\tau_{n+1} = y(t_{n+1}) - \phi^*_{n+1} = y(t_{n+1}) - \cancel{y(t_n)} - hf(t_n, y(t_n)).$$

Local Truncation Error Analysis for Forward Euler's Method

We can expand the true solution $y(t_{n+1})$ in a Taylor series about t_n,

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

Remainder

$= y(t_n) + h y'(t_n) + \frac{h^2}{2} y''(t_n) + \cdots$

We can subtract the two expansions to get the local truncation error,

For Forward Euler method

$$\tau_{n+1} = y(t_{n+1}) - \phi^*_{n+1} = \boxed{\frac{h^2}{2} y''(\xi)} \quad \text{L. T. E.}$$

This shows that the local truncation error of the forward Euler's method has the property

$$|\tau_{n+1}| \le \frac{1}{2} M_n h^2 \sim O(h^2).$$

We say the forward Euler's method is a **locally second order accurate** method, as the **local truncation error** (LTE) is of order $O(h^2)$.

We also want to introduce the **global error** for numerical solutions to ODEs. Suppose we have a sequence of approximations ϕ_n at t_n to the true solution $y(t_n)$ of the IVP (1), for time interval $[a, b]$, with $t_0 = a, t_1, \ldots, t_n = b$. Then the global discretization error E_n at time t_n is defined as

exact numerical

$$E_n = y(t_n) - \phi_n.$$

We note the error E_n at the end of the interval is usually called the **final global error**.

And particularly the global error is the maximum of the errors at all time points,

$$E = \max_{0 \le n \le N} |E_n|.$$

This error is a measure of how accurate ϕ_i agree with the true solution over the whole interval.

While one can derive the global error for a method, the local truncation error tends to be relatively easy to get. In general, the order of error for the global [redacted] error is one order lower than the local truncation error.

From the concept of global error, we can derive that the forward Euler's method is a first order method in terms of global error: $E_n = O(h)$. We can see this conclusion from the *error analysis.*

As for each n, we have the local error $\tau_{k+1} = y''(\xi_k)h^2/2$ derived above. **Roughly speaking**, we sum up the local errors for $k = 1, \ldots, n$, we get the accumulated error upto time t_n as

$\le M$ $\quad t_f \quad t_0$

$$\sum_{k=1}^{n} |\tau_k| \le \sum_{k=1}^{n} y''(\xi_k) h^2 \le n y''(\xi) h^2 \le nMh^2 \qquad n = \frac{b-a}{h}$$

$\approx (b-a)Mh$

where M is the maximum value of $y''(t)$ on $[a, b]$. We also note that $n \approx (b-a)/h$, so the **global error** is $O(h)$, which is first order accurate.

2.5 Convergence, consistency and stability

Definition: a numerical method is called **consistent** if the LTE at each step is at least $o(h)$ as $h \to 0$. This means we need

$$\lim_{h\to 0} \max_{0\le n\le N} \frac{\tau_n}{h} = 0.$$

$\lim_{h \to 0} \frac{o(h)}{h} = 0$

Definition: A numerical method is said to be **convergent** if the global error E goes to zero as the stepsize h goes to zero. (This means all E_n go to zero as $h \to 0$.)

Example: for IVP $y' = f(t, y), y(0) = y_0$, what is the local order of error of the following explicit two-step method, and is it consistent?

$$\phi_{n+1} + 9\phi_n - 10\phi_{n-1} = \frac{h}{2}(13f(t_n, \phi_n) + 9f(t_{n-1}, \phi_{n-1})).$$

You can derive it, and the error is 2 and it is consistent method.

Stability: A consistent method might not be convergent because of the way numerical errors accumulate over time steps. For example, consider the method above multi-step method. It is consistent with order 2. However, it is unstable. An easy way to see it is to apply it to solving the IVP $y' = 0, y(0) = a$. To initiate the method, we need two initial values $u(0) = u_0$ and $u(h) = u_1$. Since the right-hand side is zero, the method becomes the following linear recurrent relationship:

$$\phi_{n+1} + 9\phi_n - 10\phi_{n-1} = 0$$

The general solution to above is $\phi_n = Ar_1^n + Br_2^n$ where r_1 and r_2 are the roots to the characteristic equation $r^2 + 9r - 10 = 0$, i.e., $r_1 = 1, r_2 = -10$. In order to obtain the constant solution $\phi_n = a$, we need $\phi_1 = \phi_2 = a$. Then $A = a$ and $B = 0$. If either of these values ϕ_1 or ϕ_2, will be slightly perturbed, the coefficient B will be nonzero and hence the solution will blow up. Note that the smaller the time step h will be, the more the solution will blow up over a fixed interval of time.

This example shows that besides requiring that the errors committed at each time step be small, they also need to accumulate stably.

Indeed, we will need the so called zero-stability condition to ensure the stability of the method: for IVP $y' = f(t, y), y(t_0) = y_0$, if $\{\phi_n\}$ and $\{\psi_n\}$ are the numerical solutions to the IVP with the initial conditions ϕ_0 and ψ_0, if it holds that

$$\max_{0 \le n \le N} |\phi_n - \psi_n| \le C \max_{0 \le n \le N} |\phi_0 - \psi_0|,$$

where C is a constant independent of h (so independent of n), then the method is said to be zero-stable.

Convergence Theorem: A typical numerical method is convergent if it is zero-stable and consistent.