Example: Initial Value Problem

Consider scalar ODE

$$y' = y$$

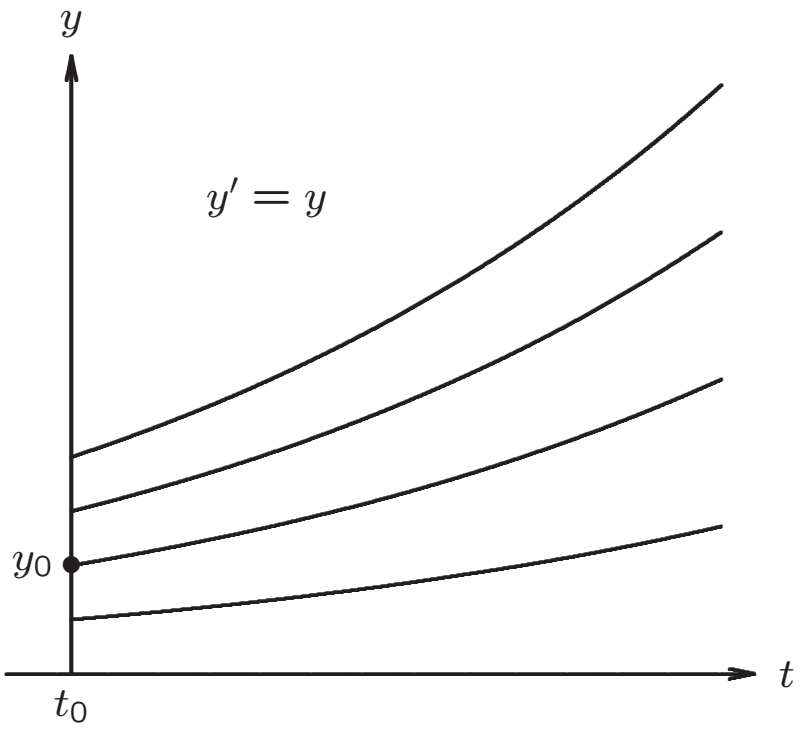
Family of solutions is given by $y = ce^t$, where c is any real constant

Imposing initial condition $y(t_0) = y_0$ singles out unique particular solution

For this example, if $t_0 = 0$, then $c = y_0$, which means that solution is $y(t) = y_0 e^t$

Example: Initial Value Problem

Family of solutions for ODE $y' = y$



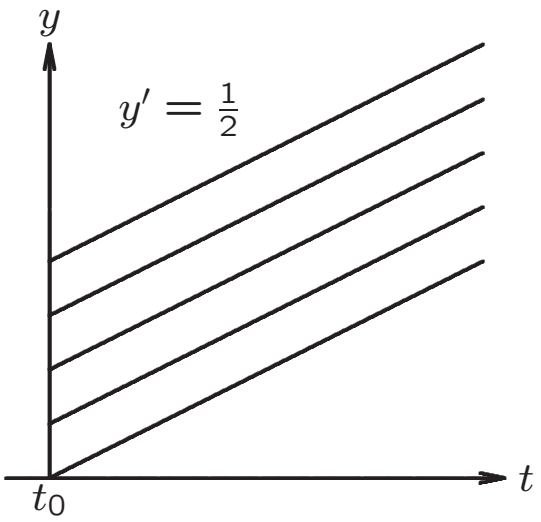
Stability of Solutions

Solution of ODE is

- *Stable* if solutions resulting from perturbations of initial value remain close to original solution
- *Asymptotically stable* if solutions resulting from perturbations converge back to original solution
- *Unstable* if solutions resulting from perturbations diverge away from original solution without bound

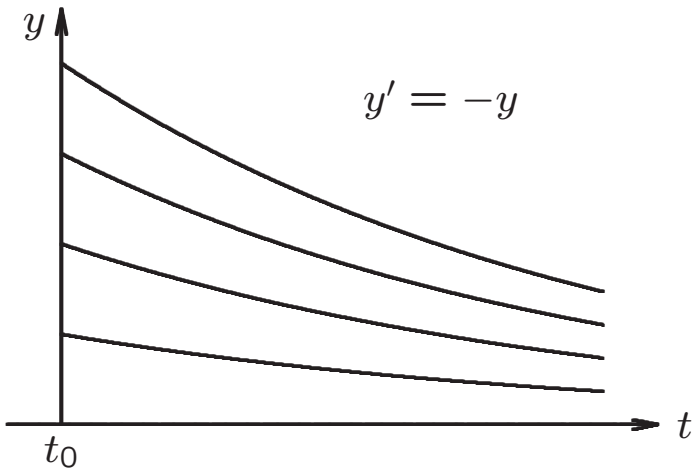
Example: Stable Solutions

Family of solutions for ODE $y' = \frac{1}{2}$



Example: Asymptotically Stable Solutions

Family of solutions for ODE $y' = -y$



Example: Stability of Solutions

Consider scalar ODE

$$y' = \lambda y,$$

where λ is constant.

Solution given by

$$y(t) = y_0\, e^{\lambda t},$$

where $t_0 = 0$ is initial time and $y(0) = y_0$ is initial value

If $\lambda > 0$, then all nonzero solutions grow exponentially, so every solution is unstable

If $\lambda < 0$, then all nonzero solutions decay exponentially, so every solution is not only stable, but asymptotically stable

If λ is complex, then solutions are unstable if $\mathrm{Re}(\lambda) > 0$, asymptotically stable if $\mathrm{Re}(\lambda) < 0$, and stable but not asymptotically stable if $\mathrm{Re}(\lambda) = 0$

Euler's Method

For general ODE $\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$, consider Taylor series

$$\begin{aligned} \boldsymbol{y}(t+h) &= \boldsymbol{y}(t) + h\boldsymbol{y}'(t) + \frac{h^2}{2}\boldsymbol{y}''(t) + \cdots \\ &= \boldsymbol{y}(t) + h\boldsymbol{f}(t, \boldsymbol{y}(t)) + \frac{h^2}{2}\boldsymbol{y}''(t) + \cdots \end{aligned}$$

Euler's method results from dropping terms of second and higher order to obtain approximate solution value

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + h_k \boldsymbol{f}(t_k, \boldsymbol{y}_k)$$

Euler's method advances solution by extrapolating along straight line whose slope is given by $\boldsymbol{f}(t_k, \boldsymbol{y}_k)$

Euler's method is *single-step* method because it depends on information at only one point in time to advance to next point

Example: Euler's Method

Applying Euler's method to ODE $y' = y$ with step size h, we advance solution from time $t_0 = 0$ to time $t_1 = t_0 + h$:

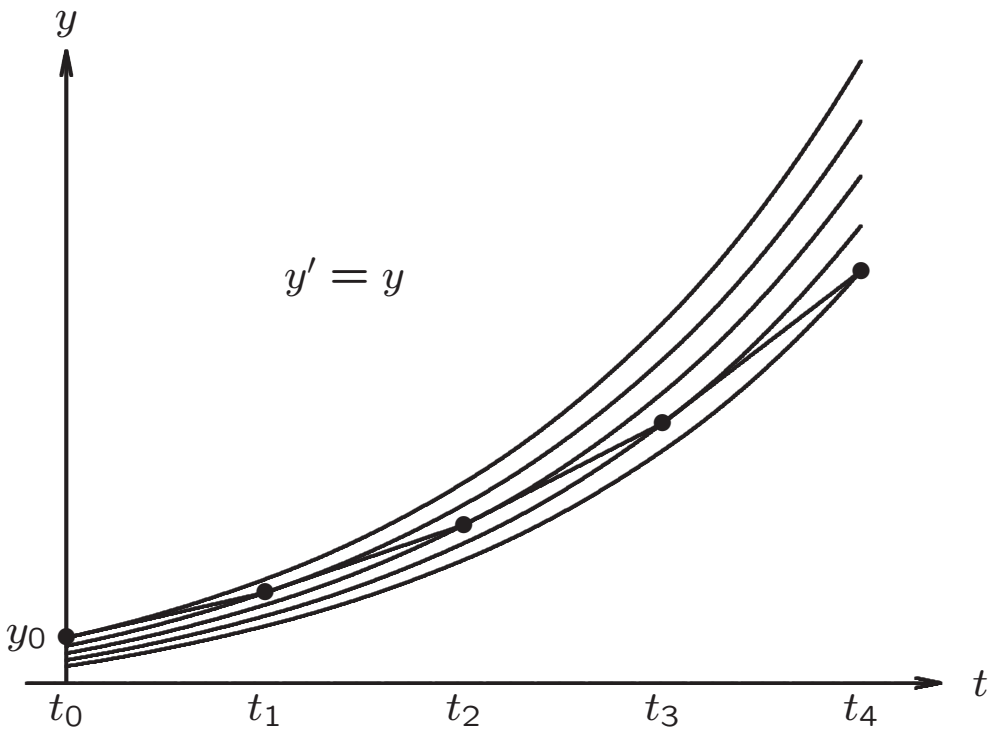
$$y_1 = y_0 + hy_0' = y_0 + hy_0 = (1 + h)y_0$$

Value for solution we obtain at t_1 is not exact, $y_1 \neq y(t_1)$

For example, if $t_0 = 0$, $y_0 = 1$, and $h = 0.5$, then $y_1 = 1.5$, whereas exact solution for this initial value is $y(0.5) = \exp(0.5) \approx 1.649$

Thus, y_1 lies on different solution from one we started on

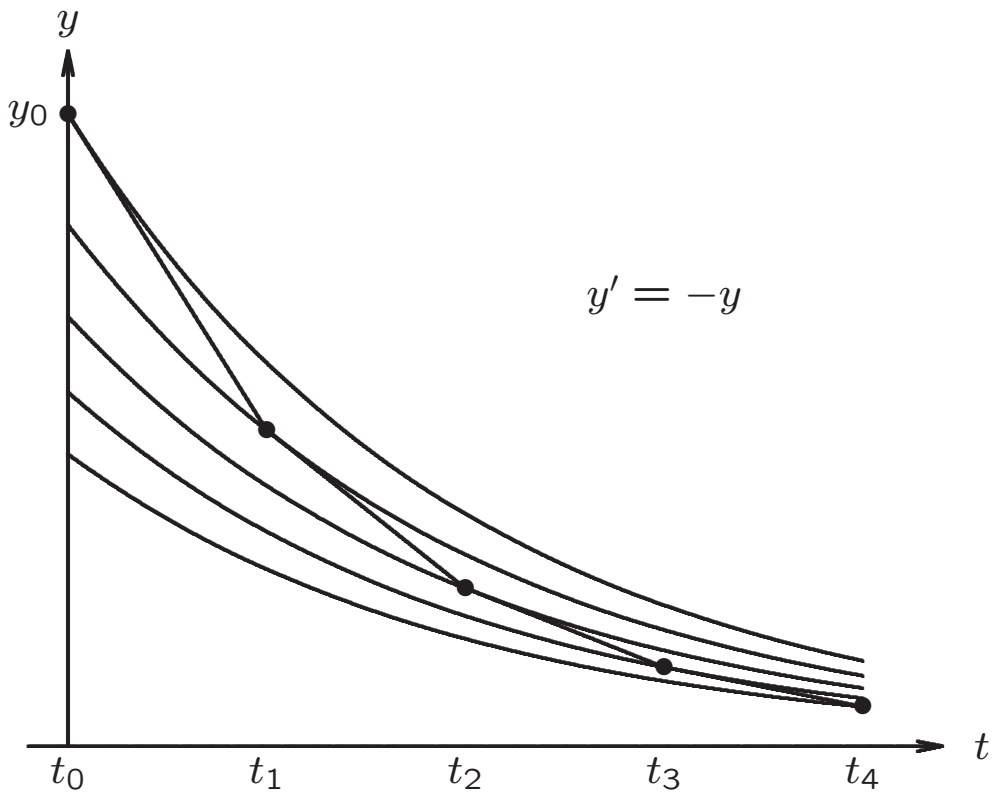
Example Continued



For unstable solutions, errors in numerical solution grow with time

Example Continued

For stable solutions, errors in numerical solution may diminish with time



Errors in Numerical Solution of ODEs

Numerical methods for solving ODEs suffer from two distinct sources of error:

- *Rounding* error, which is due to finite precision of floating-point arithmetic
- *Truncation* (or discretization) error, which is due to method used and would remain even if all arithmetic were exact

In practice, truncation error is dominant factor determining accuracy of numerical solutions of ODEs, and we shall henceforth ignore rounding error

Global Error and Local Error

Truncation error can be broken down into:

- *Global* error, which is difference between computed solution and true solution determined by initial data at t_0:

$$\boldsymbol{e}_k = \boldsymbol{y}_k - \boldsymbol{y}(t_k)$$

- *Local* error, which is error made in one step of numerical method:

$$\boldsymbol{\ell}_k = \boldsymbol{y}_k - \boldsymbol{u}_{k-1}(t_k),$$

where $\boldsymbol{u}_{k-1}$ is solution through $(t_{k-1}, \boldsymbol{y}_{k-1})$

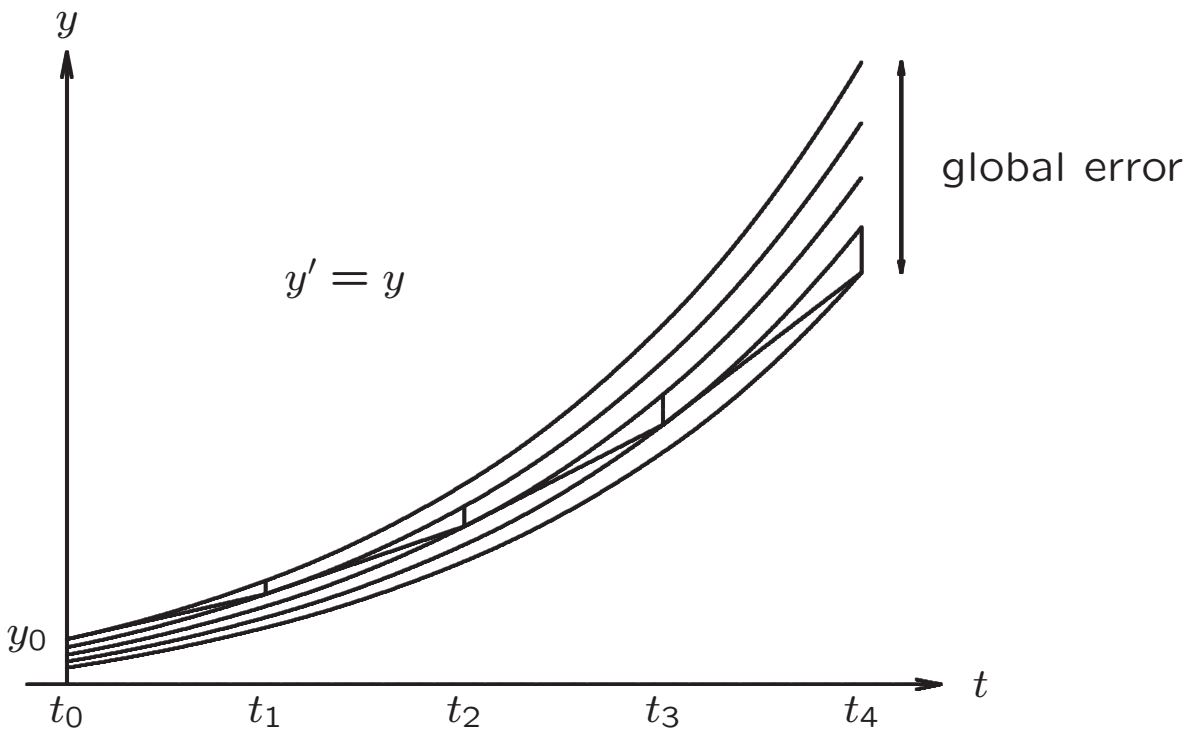
Global Error and Local Error, cont.

Global error is not necessarily sum of local errors

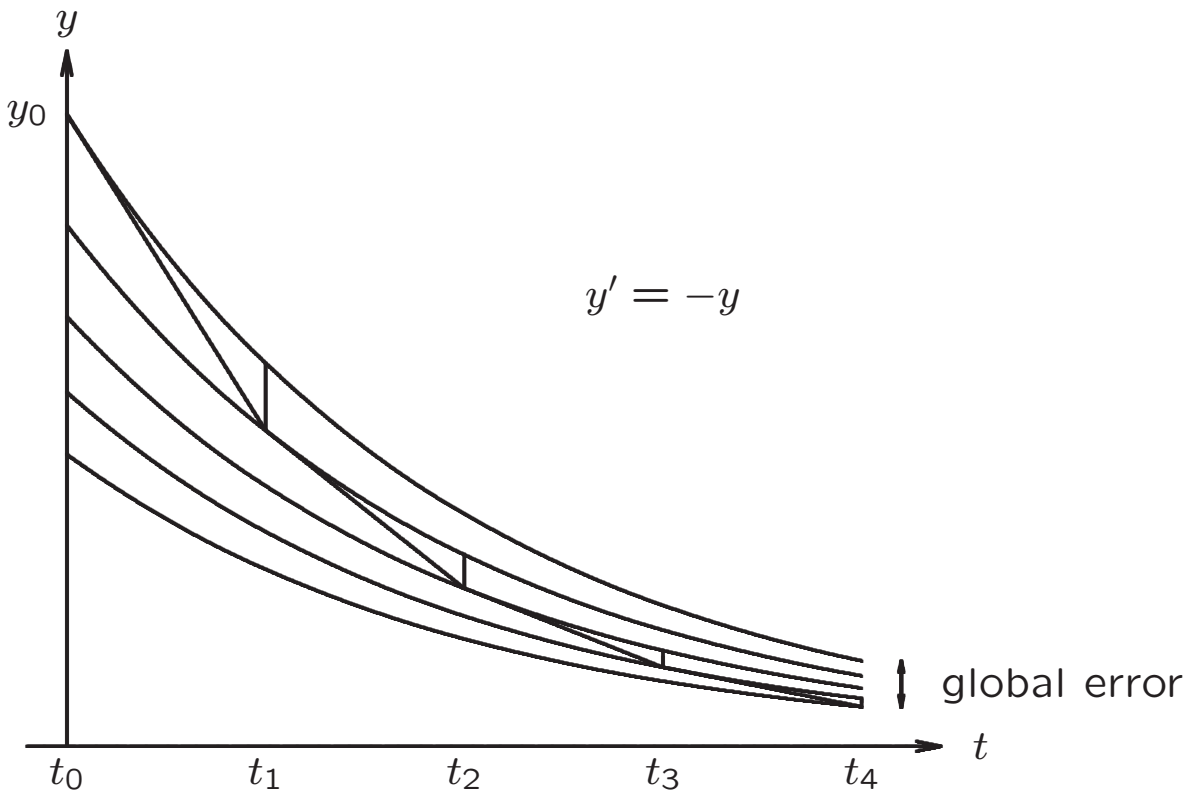
Global error generally greater than sum of local errors if solutions unstable, but may be less than sum if solutions stable

Having small global error is what we want, but we can control only local error directly

Global Error and Local Error, cont.



Global and Local Error, cont.



Determining Stability and Accuracy

Simple approach to determining stability and accuracy of numerical method is to apply it to scalar ODE $y' = \lambda y$, where λ is (possibly complex) constant

Exact solution given by $y(t) = y_0 e^{\lambda t}$, where $y(0) = y_0$ is initial condition

For given numerical method, we can

- Determine stability by characterizing growth of numerical solution
- Determine accuracy by comparing exact and numerical solutions

Example: Euler's Method

Applying Euler's method to $y' = \lambda y$ using fixed step size h, we have

$$y_{k+1} = y_k + h\lambda y_k = (1 + h\lambda)y_k,$$

which means that

$$y_k = (1 + h\lambda)^k y_0$$

If $\mathrm{Re}(\lambda) < 0$, exact solution decays to zero as t increases, as does computed solution if

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

which holds if $h\lambda$ lies inside circle in complex plane of radius 1 centered at -1

If λ is real, then $h\lambda$ must lie in interval $(-2, 0)$, so for $\lambda < 0$, we must have $h \leq -2/\lambda$ for Euler's method to be stable

Euler's Method, continued

For general ODE $\boldsymbol{y}' = \boldsymbol{f}(t, \boldsymbol{y})$, consider Taylor series

$$\begin{aligned} \boldsymbol{y}(t+h) &= \boldsymbol{y}(t) + h\boldsymbol{y}'(t) + \mathcal{O}(h^2) \\ &= \boldsymbol{y}(t) + h\boldsymbol{f}(t, \boldsymbol{y}(t)) + \mathcal{O}(h^2) \end{aligned}$$

If we take $t = t_k$ and $h = h_k$, we obtain

$$\boldsymbol{y}(t_{k+1}) = \boldsymbol{y}(t_k) + h_k\boldsymbol{f}(t_k, \boldsymbol{y}(t_k)) + \mathcal{O}(h_k^2)$$

Subtracting this from Euler's method, we get

$$\begin{aligned} \boldsymbol{e}_{k+1} &= \boldsymbol{y}_{k+1} - \boldsymbol{y}(t_{k+1}) \\ &= [\boldsymbol{y}_k - \boldsymbol{y}(t_k)] + \\ &\quad h_k[\boldsymbol{f}(t_k, \boldsymbol{y}_k) - \boldsymbol{f}(t_k, \boldsymbol{y}(t_k))] - \mathcal{O}(h_k^2) \end{aligned}$$

Euler's Method, continued

If there were no prior errors, then we would have $\boldsymbol{y}_k = \boldsymbol{y}(t_k)$, and differences in brackets on right side would be zero, leaving only $\mathcal{O}(h_k^2)$ term, which is local error

This means that Euler's method is first-order accurate

Step Size Selection

In choosing step size for advancing numerical solution of ODE, want to take large steps to reduce computational cost, but must also take into account both stability and accuracy

To yield meaningful solution, step size must obey any stability restrictions

In addition, local error estimate is needed to ensure that desired accuracy is achieved

With Euler's method, for example, local error is approximately $(h_k^2/2)\boldsymbol{y}''$, so choose step size to satisfy

$$h_k \leq \sqrt{2\ tol/\|\boldsymbol{y}''\|}$$

Implicit Methods

Euler's method is *explicit* in that it uses only information at time t_k to advance solution to time t_{k+1}

This may seem desirable, but Euler's method has rather limited stability region

Larger stability region can be obtained by using information at time t_{k+1}, which makes method *implicit*

Simplest example is *backward Euler method*,

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + h_k \boldsymbol{f}(t_{k+1}, \boldsymbol{y}_{k+1})$$

Method is implicit because we must evaluate $\boldsymbol{f}$ with argument $\boldsymbol{y}_{k+1}$ before we know its value

Backward Euler Method

To determine stability of backward Euler, we apply it to scalar ODE $y' = \lambda y$, obtaining

$$y_{k+1} = y_k + h\lambda y_{k+1},$$

or

$$(1 - h\lambda)y_{k+1} = y_k,$$

so that

$$y_k = \left(\frac{1}{1 - h\lambda}\right)^k y_0$$

Thus, for backward Euler to be stable we must have

$$\left|\frac{1}{1 - h\,\lambda}\right| \leq 1,$$

which holds for *any* $h > 0$ when $\mathrm{Re}(\lambda) < 0$

So stability region for backward Euler method includes entire left half of complex plane, or interval $(-\infty, 0)$ if λ is real

Unconditionally Stable Methods

Great virtue of unconditionally stable method is that desired accuracy is only constraint on choice of step size

Thus, we may be able to take much larger steps than for explicit method of comparable order and attain much higher overall efficiency despite requiring more computation per step

Although backward Euler method is unconditionally stable, its accuracy is only of first order, which severely limits its usefulness