高等数值算法与应用（九）

Advanced Numerical Algorithms & Applications

计算机科学与技术系 喻文健

内容概要

- 常微分方程初值问题
  - 基本概念
  - **ODE**的稳定性**(**解的稳定性**)**
  - 简单的数值解法与算法稳定性、截断误差分析
  - 实用的数值解法
  - **Matlab**有关命令和应用例子

Ordinary Differential Equations (I)

Wenjian Yu

- 本节主要内容
  - 常微分方程(ODE)基本概念
  - 初值问题(IVP)
  - 初值问题的稳定性(敏感性)
  - 例子与Matlab演示
  - ODE-IVP数值解法
    - 欧拉方法
  - 数值解法的稳定性
  - 数值解法的截断误差与准确度
  - 数值解法的步长选择

常微分方程

- 要求解的是单自变量的函数(一个或多个)
- 自变量常为时间t, 未知函数为系统状态量y
- 微分方程表示未知函数与自变量之间关系, 其中对t的最高求导阶数为常微分方程的阶数

$$\textbf{k阶ODE:}\quad g\left(t, y, y', y'', \cdots, y^{(k)}\right) = \mathbf{0}$$

- 一般可转化为显式常微分方程

$$\textbf{显式ODE:}\quad y^{(k)} = f\left(t, y, y', y'', \cdots, y^{(k\text{-}1)}\right)$$

- 主要考虑显式ODE，特例是一阶、单个ODE问题

$$y' = f\left(t, y\right) \qquad \text{例如：}\quad y' = \lambda y$$

高阶常微分方程 → 一阶常微分方程组

Given k-th order ODE

$$y^{(k)} = f(t, y, y', \ldots, y^{(k-1)}),$$

define k new unknowns

$$\begin{aligned} u_1(t) &= y, \\ u_2(t) &= y', \\ &\vdots \\ u_k(t) &= y^{(k-1)} \end{aligned}$$

Original ODE equivalent to first-order system

$$\begin{bmatrix} u_1' \\ u_2' \\ \vdots \\ u_{k-1}' \\ u_k' \end{bmatrix} = \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_k \\ f(t, u_1, u_2, \ldots, u_k) \end{bmatrix}$$

不失一般性，仅讨论一阶常微分方程组：$y' = f(t, y)$

Example: Newton's Second Law

牛顿运动第二定律

Newton's Second Law of Motion, $F = ma$, is second-order ODE, since acceleration a is second derivative of position coordinate, denoted by y

加速度：
运动位置y的
二阶时间导数

Thus, ODE has form

$$y'' = F/m,$$

where F and m are force and mass, respectively

Defining $u_1 = y$ and $u_2 = y'$ yields equivalent system of two first-order ODEs

$$\mathbf{u}' = \mathbf{f}(t, \mathbf{u})$$

$$\begin{bmatrix} u_1' \\ u_2' \end{bmatrix} = \begin{bmatrix} u_2 \\ F/m \end{bmatrix}$$

$$\mathbf{u}' = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{u} + \begin{bmatrix} 0 \\ \mathbf{F/m} \end{bmatrix}$$

线性、常系数微分方程

常微分方程的初值问题

- 仅根据 $y' = f(t, y)$, 无法唯一确定解函数
- 要确定唯一解，需在一些自变量点上给出函数值
- 一种定解条件是：$y(t_0) = y_0$ 　t_0点上y的各分量函数值都已知
- t_0常表示初始时间点，这种定解问题为初值问题

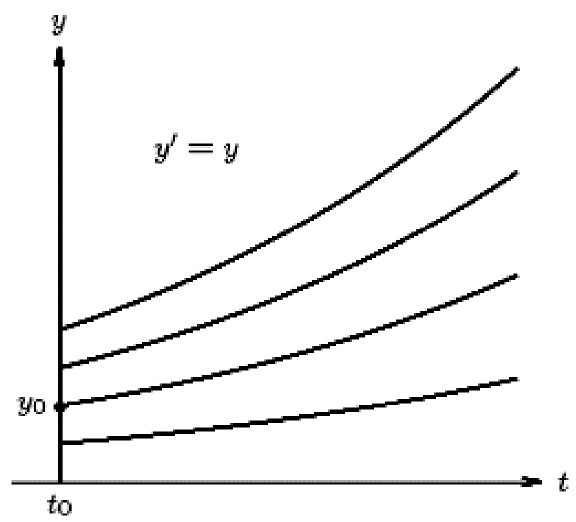
Consider scalar ODE

$$y' = y$$

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

Family of solutions for ODE $y' = y$



ODE初值问题解的存在性

$$\begin{cases} y' = f(t,y) \\ y(t_0) = y_0 \end{cases}, a \le t \le b$$

若f在**D**上关于y是利普希兹连续的，则初值问题存在唯一解

(*f* 可导就满足这个条件)

f 定义域为 $D = [a,b] \times \Omega \subset \mathbb{R}^{n+1}$

Stability of Solutions

由于历史的原因，在不至于混淆的情况下，将**ODE-IVP**问题的敏感性称为**ODE**问题**(**或解**)**的稳定性。

Solution of ODE is

- *Stable* if solutions resulting from perturbations of initial value remain close to original solution

渐进稳定，比**Stable**更强的要求

- *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

初始数据是 $y(t_0)$

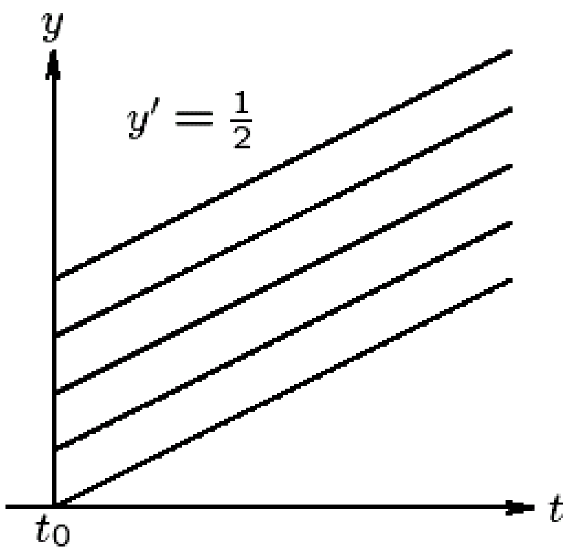
解是函数 $y(t)$

关注**t**→∞时解的误差：

有界、**0**、或无穷大

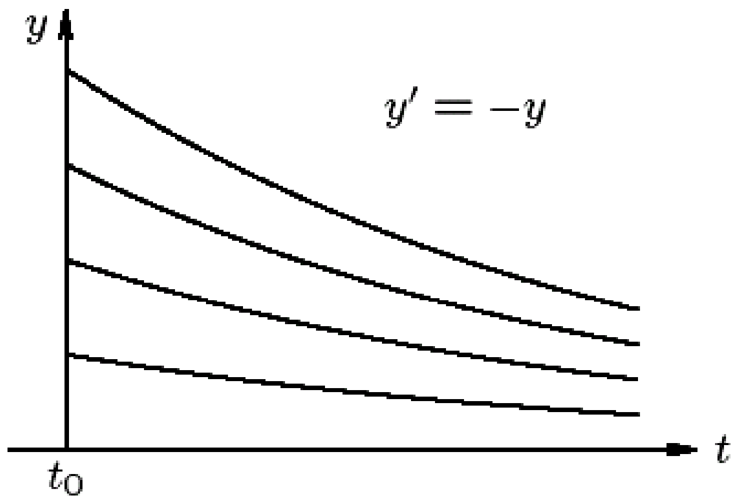
Example: Stable Solutions

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 is given by:

$$y(t) = y_0 e^{\lambda(t-t_0)}$$

t→∞时解的趋势即反映误差Δy的趋势

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 $\mathrm{Re}(\lambda) = 0$

定义**:**若常微分方程中具有 $y' = f(y)$ 的形式，称其为自治**(autonomous)**常微分方程。

本例为最简单的自治**ODE**

注意**,** 非自治的**ODE**可转化为自治的**ODE**

Example: Linear System of ODEs

线性常系数、齐次

也适用于线性常系数、非齐次**ODE**

向量函数

Linear, homogeneous system of ODEs with constant coefficients has form

最简单的自治常微分方程组

$$y' = Ay,$$

where A is $n \times n$ matrix, and initial condition is $y(0) = y_0$

Suppose A is diagonalizable, with eigenvalues λ_i and corresponding eigenvectors v_i, $i = 1, \ldots, n$

Express y_0 as linear combination

$$y_0 = \sum_{i=1}^{n} \alpha_i v_i$$

Then

$$y(t) = \sum_{i=1}^{n} \alpha_i v_i e^{\lambda_i t}$$

Example Continued

- Eigenvalues of A with positive real parts yield exponentially growing solution components
- Eigenvalues with negative real parts yield exponentially decaying solution components
- Eigenvalues with zero real parts (i.e., pure imaginary) yield oscillatory solution components

震荡

Solutions stable if $\mathrm{Re}(\lambda_i) \leq 0$ for every eigenvalue, and asymptotically stable if $\mathrm{Re}(\lambda_i) < 0$ for every eigenvalue, but unstable if $\mathrm{Re}(\lambda_i) > 0$ for some eigenvalue

A-stable

只要有一个特征值实部**>0**，则不稳定

若**A**不能对角化**:** 重特征值，要求其实部**<0**才保证解稳定

变系数矩阵***A*(*t*)**，特征值分析只能讨论具体***t***值附近的局部稳定性

Stability of Solutions, cont.

For general nonlinear ODE $y' = f(t, y)$, determining stability of solutions is more complicated

f是y的非线性函数

ODE can be linearized locally about solution $y(t)$ by truncated Taylor series, yielding linear ODE

$$z' = J_f(t, y(t))\, z,$$

where J_f is Jacobian matrix of f with respect to y

?

$\{J_f(t, \mathbf{y})\}_{i,j} = \partial f_i(t, \mathbf{y}) / \partial y_j$

Eigenvalues of J_f determine stability locally but conclusions drawn may not be valid globally

1. 时间t附近的局部
2. 解函数f 附近的局部

来自实际的常微分方程初值问题大多数是稳定的，如果不稳定，须从物理过程数学建模的角度重新考虑

实例与Matlab demo

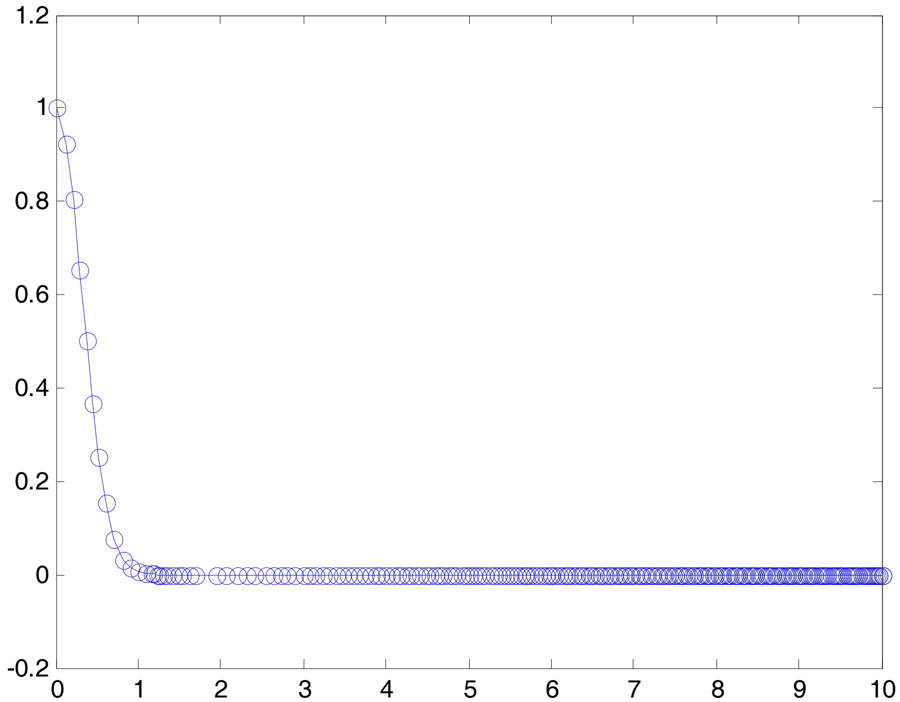
- 单个ODE：分析稳定性, Matlab求解

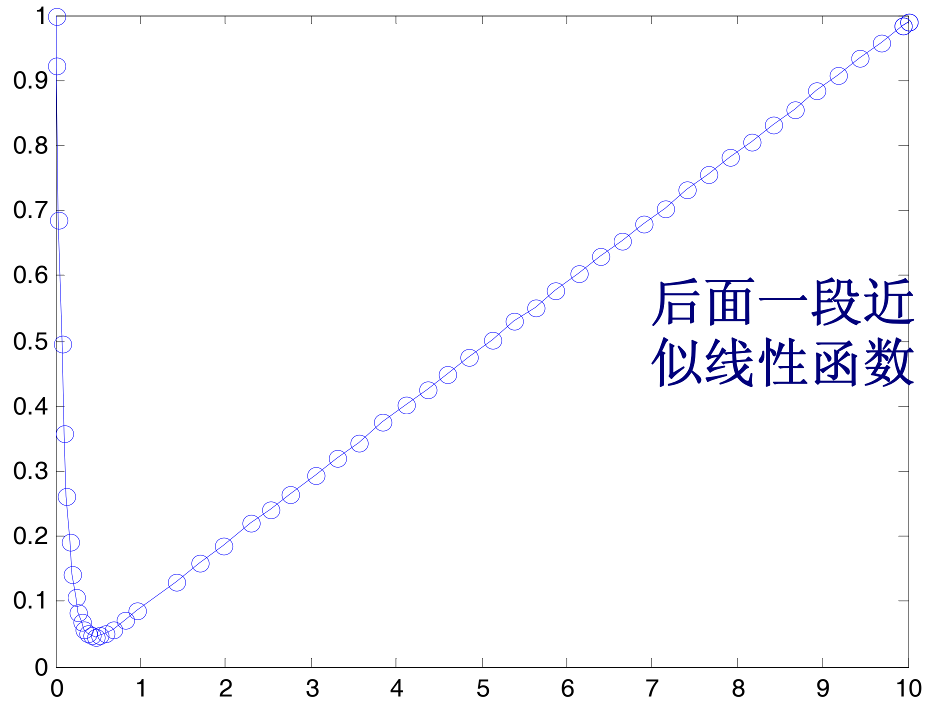
$$\begin{cases} y' = -10yt \\ y(0) = 1 \end{cases} \qquad J_f = -10t$$

$$\begin{cases} y' = -10y + t \\ y(0) = 1 \end{cases} \qquad J_f = -10$$

True solution: $y(t) = e^{-5t^2}$

f=inline('-10*y*t', 't','y'); ode23(f, [0,10], 1);　　Matlab: ode_1





后面一段近似线性函数

实例与Matlab demo

- 常微分方程组: "牛二"定律

$$\begin{cases} y_1' = y_2 \\ y_2' = 6t \\ y_1(0) = 0 \\ y_2(0) = 1 \end{cases}$$

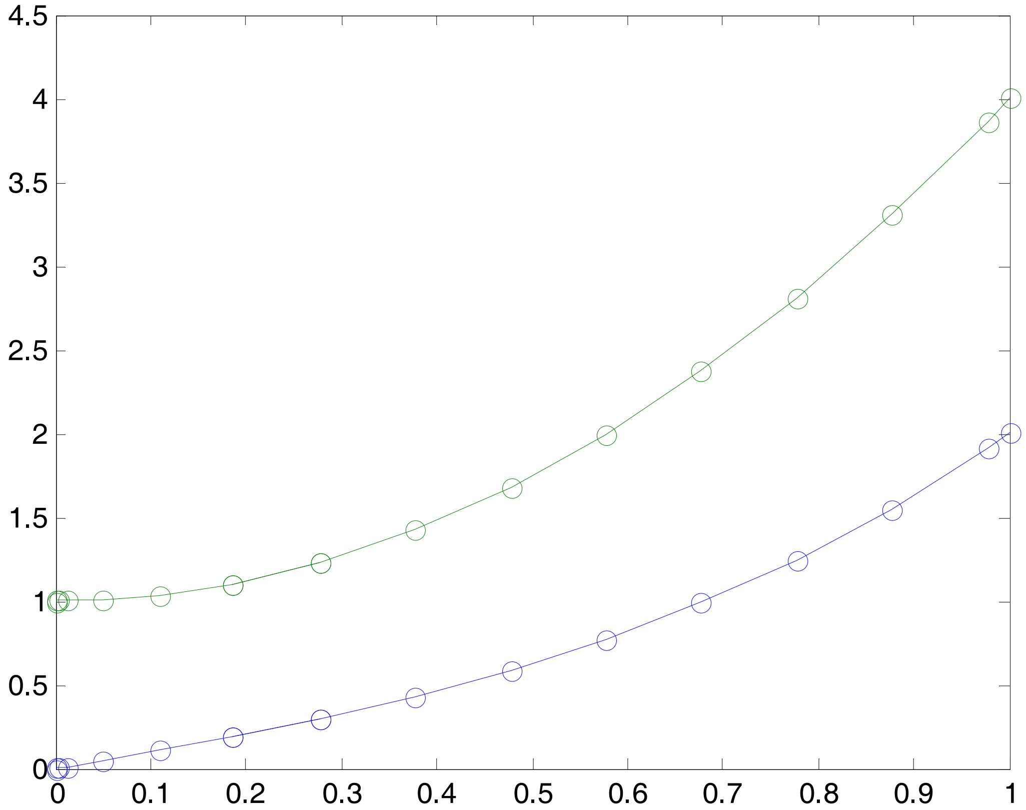
有解析解

稳定吗？

ode23(@myode2, [0, 1], [0; 1]);

```
function ydot = myode2(t,y);
ydot= [y(2); 6*t];  %column vector
```

Matlab: ode_2



- 本节主要内容
  - 常微分方程(ODE)基本概念
  - 初值问题(IVP)
  - 初值问题的稳定性(敏感性)
  - 例子与Matlab演示
  - ODE-IVP数值解法
    - 欧拉方法
  - 数值解法的稳定性
  - 数值解法的截断误差与准确度
  - 数值解法的步长选择

ODE-IVP的数值解法

解析解往往不存在，采用数值解法。数值解法的思想是，求一系列离散点上的函数近似值。离散点和对应的函数近似值通常记为：

$$t_0 < t_1 < \cdots < t_n < t_{n+1} < \cdots$$

$$y_0,\quad y_1,\quad \cdots\quad y_n,\quad y_{n+1},\cdots$$

$y' = f(t, y)$

步长$h_n = t_{n+1} - t_n$。根据微积分知识，

$$y(t+h) = \int_t^{t+h} f(s, y(s))ds + y(t)$$

由于积分中包含了未知函数$y(s)$，这里不能直接用数值积分的办法求出这个积分，但将$y(s)$加以近似再采用数值积分的一些技术常常可推导出一些数值方法。

数值解法通常构成形如$y_{n+1} = G(y_{n+1}, y_n, y_{n-1}, \dots, y_{n-k})$的方程，它是若干邻近节点函数值满足的关系。解法的分类如下：

$$k: \begin{cases} = 0, \text{单步法} \\ > 0, \text{多步法} \end{cases} \qquad G: \begin{cases} \text{不含}y_{n+1}, \text{显格式} \\ \text{含}y_{n+1}, \quad \text{隐格式} \end{cases}$$

Euler's Method

$$y(t+h) = \int_t^{t+h} f(s, y(s))ds + y(t)$$

考虑从时间点t_k到t_{k+1}

$$y_{k+1} = \int_{t_k}^{t_k+h_k} f(s, y(s))ds + y(t_k)$$

$$\approx h_k f(t_k, y(t_k)) + y(t_k)$$

最简单的数值积分

由于无法得到$y(t_k)$准确值，用y_k代替

$$y_{k+1} = y_k + h_k f(t_k, y_k)$$

单步、显格式

Euler's method advances solution by extrapolating along straight line whose slope is given by $f(t_k, y_k)$

沿近似切线方向外推

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 $y(0.5) = \exp(0.5) \approx 1.649$

准确解是：

$$y(t) = e^t$$

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

Example Continued

To continue numerical solution process, we take another step from t_1 to $t_2 = t_1 + h = 1.0$, obtaining $y_2 = y_1 + hy_1 = 1.5 + (0.5)(1.5) = 2.25$

准确解是：

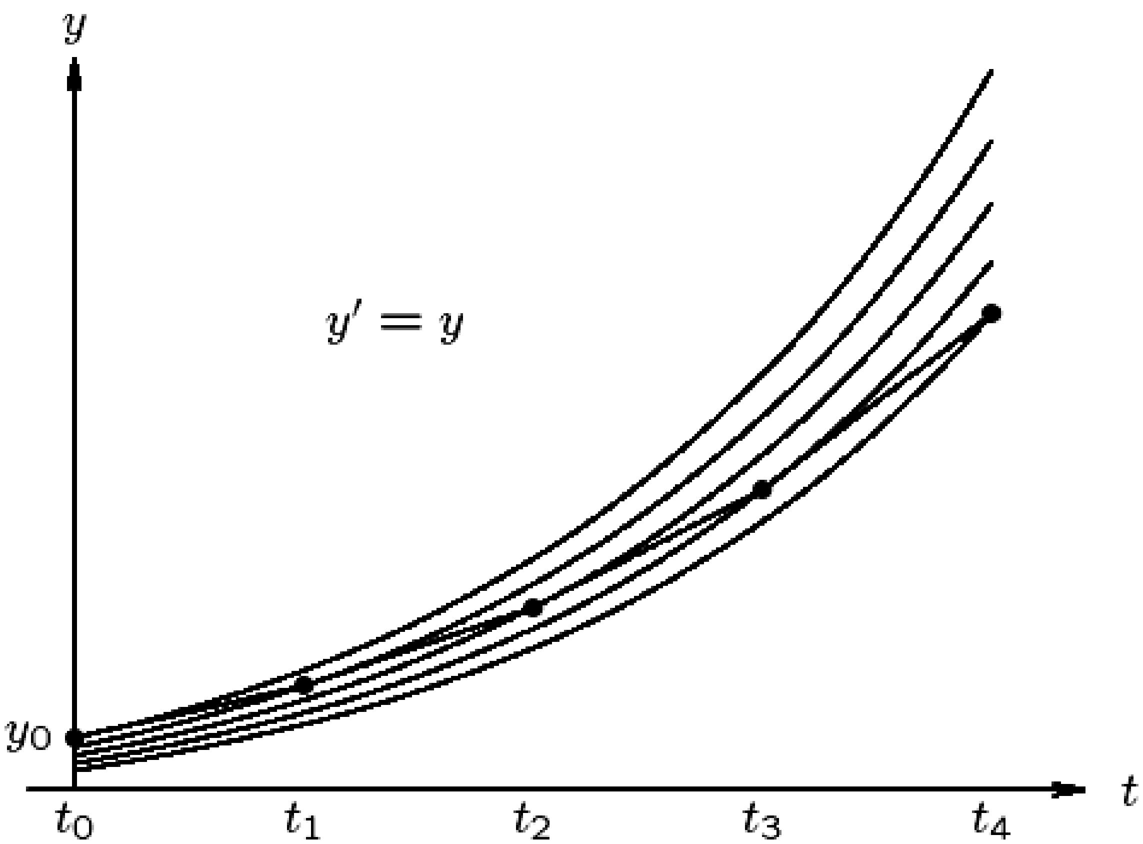
$$y(t) = e^t$$

Now y_2 differs not only from true solution of original problem at $t = 1$, $y(1) = \exp(1) \approx 2.718$, but it also differs from solution through previous point (t_1, y_1), which has approximate value 2.473 at $t = 1$

由它和微分方程构成的解：$y(t) = 1.5e^{t-0.5}$

Thus, we have moved to still another solution for this ODE

Example Continued



若ODE本身是不稳定的：

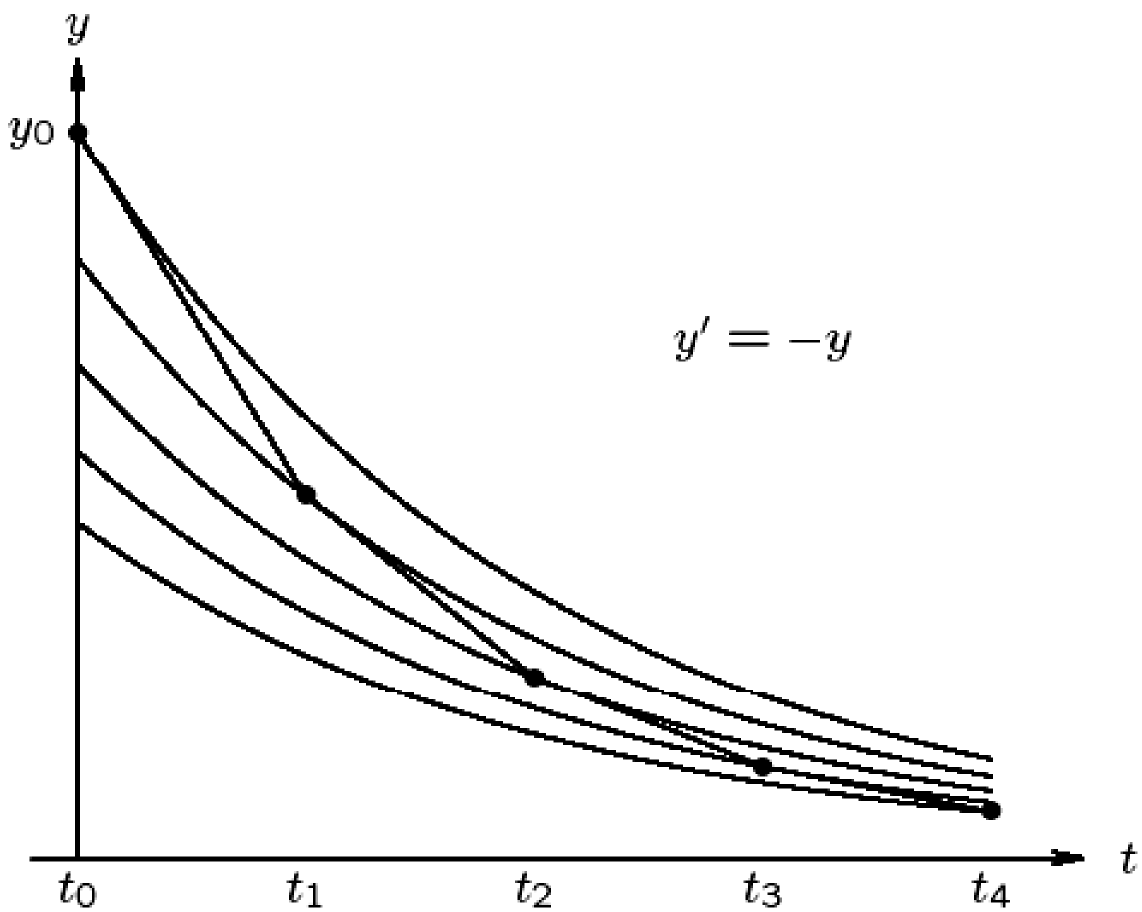
1.数值解法的误差造成“初值”扰动，**t→∞**时理论解误差无穷大。

2.即便在**t≠∞**处，结果误差为理论解误差**+**数值误差，也会非常大。

For unstable solutions, errors in numerical solution grow with time

Example Continued

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



若**ODE**本身是稳定的：

1.数值解法的误差造成“初值”扰动，**t→∞**时理论解误差趋于**0**。
2.而在**t≠∞**处，结果误差为理论解误差**+**数值误差，结果是否准确主要看数值误差。

若对稳定的**ODE**，结果误差随**t**增大而增大，则算法不稳定。

数值求解方法的稳定性

数值方法稳定，
结果才有意义!

为避免常微分方程本身稳定性的干扰，只考察稳定的常微分方程初值问题，来说明数值方法的稳定性。

为了方便，通常先考虑简单的模型问题：

$$y' = \lambda y, \quad \mathrm{Re}(\lambda) < 0$$

对每一个**ODE-IVP**数值解法，需要：

- 研究其稳定性 **(**通过数值解或整体误差的变化趋势**)**
- 研究其准确度 **(**通过推导局部截断误差，得准确度阶数**)**

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

因此，我们主要讨论精确计算模式下的截断误差，通过它评价算法的稳定性和准确度

Example: Euler's Method

数值方法的稳定性

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

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

误差也按此规律变化

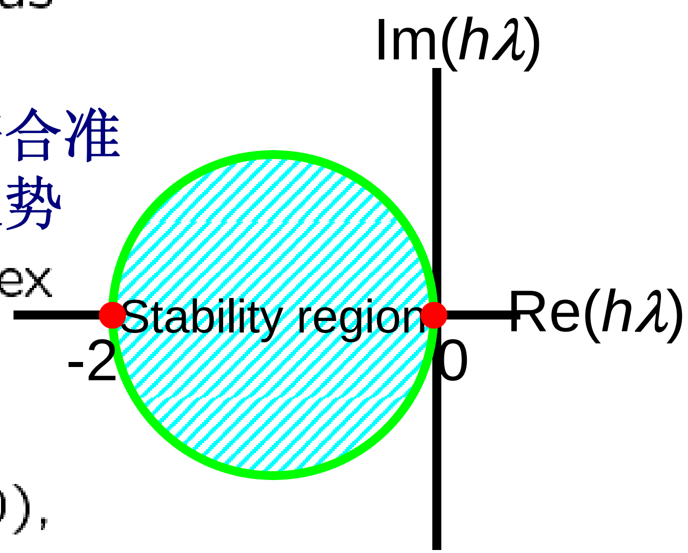
Euler法保持稳定需对步长h做限制

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

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

否则不符合准确解的趋势

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

若**hλ≈-1**，这个**h**是最好的步长吗？

Euler's Method, continued

求解一般问题 $y' = f(t, y)$ 时的稳定性

$$e_k \xrightarrow{?} e_{k+1}$$

$$e_{k+1} = y_{k+1} - y(t_{k+1})$$
$$= y_k + h_k f(t_k, y_k) - y(t_k) - h_k f(t_k, y(t_k)) - O(h_k^2)$$
$$= e_k + h_k \left[f(t_k, y_k) - f(t_k, y(t_k)) \right] - O(h_k^2)$$

From Mean Value Theorem, we have

$$f(t_k, y_k) - f(t_k, y(t_k)) = J_f(t_k, \xi)(y_k - y(t_k))$$

for some (unknown) value ξ, where J_f is Jacobian matrix of f with respect to y

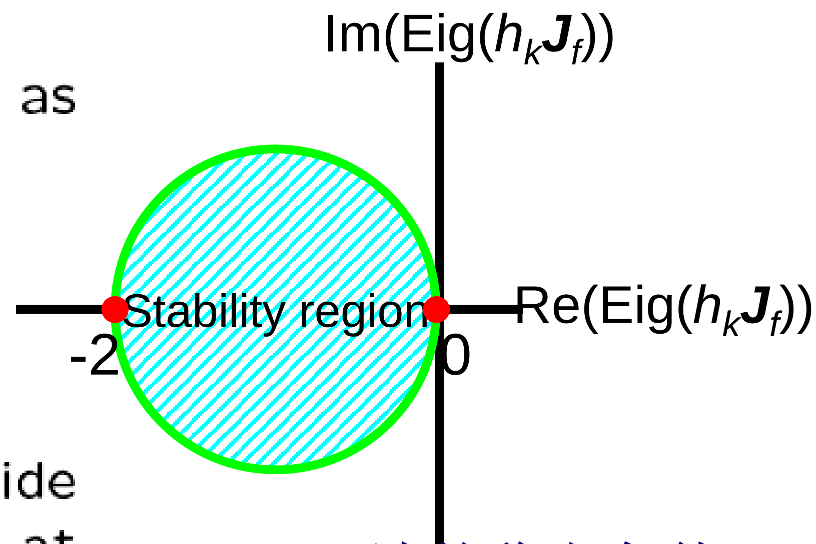
So we can express global error at step $k+1$ as

$$e_{k+1} = (I + h_k J_f) e_k + O(h_k^2)$$

Errors do not grow if spectral radius

$$\rho(I + h_k J_f) \le 1,$$

which holds if all eigenvalues of $h_k J_f$ lie inside circle in complex plane of radius 1 centered at -1



Euler法的稳定条件

Global Error and Local Error

用两个概念来描述截断误差

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

$$e_k = y_k - y(t_k)$$

整体误差

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

$$\ell_k = y_k - u_{k-1}(t_k),$$

where u_{k-1} is solution through (t_{k-1}, y_{k-1})

局部截断误差：当前这一步的误差

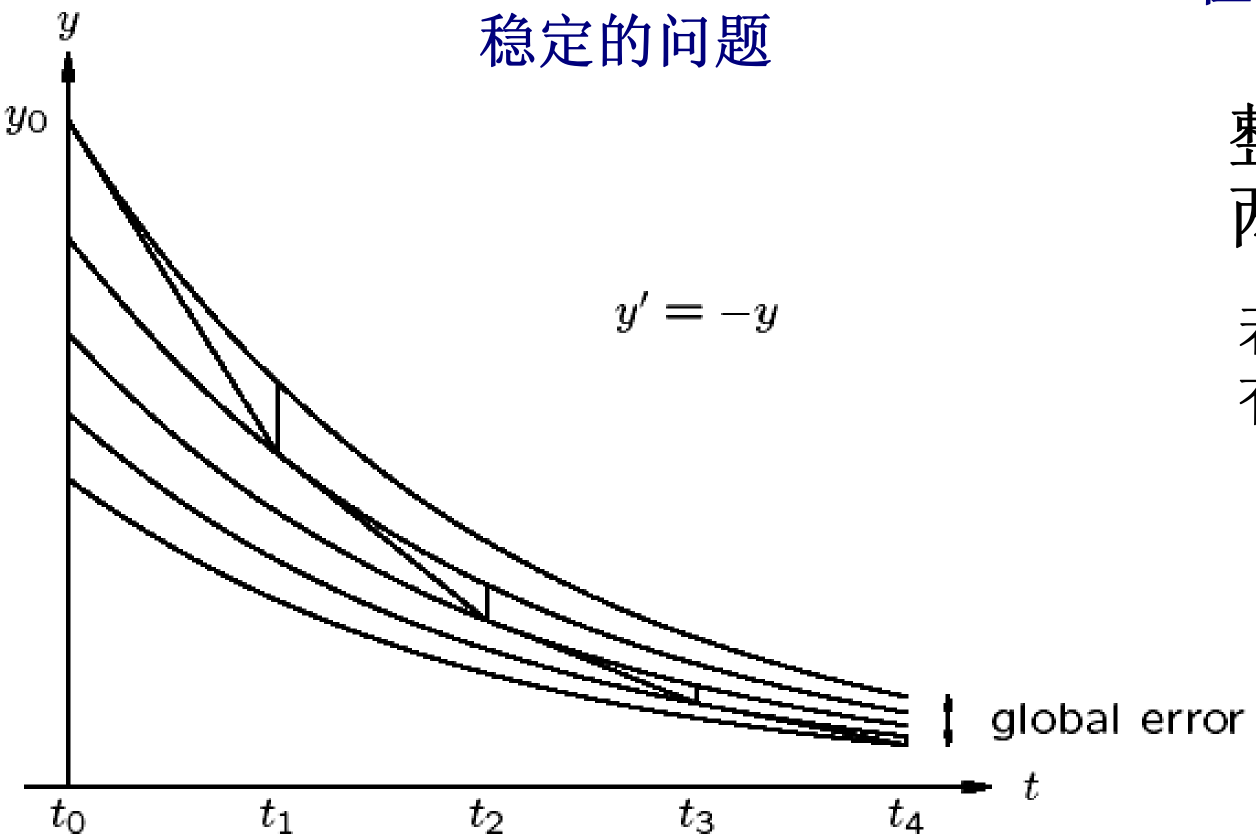
在前一步得准确解前提下，考虑当前解的误差

Global error is not necessarily sum of local errors

Global and Local Error, cont.

对不稳定的问题，整体误差往往大于所有局部误差之和。

稳定的问题



整体、局部截断误差两者关系:

若 $\ell_k = \mathcal{O}(h_k^{p+1})$，一般也有 $e_k = \mathcal{O}(h^p)$，

h为平均步长

Accuracy of numerical method is of *order* p if

(一般仅能控制局部误差)

$$\ell_k = \mathcal{O}(h_k^{p+1}) \qquad \le C \cdot h_k^{p+1}，当h_k足够小$$

Local error per unit step, $\ell_k / h_k = \mathcal{O}(h_k^p)$

称此方法有**p**阶准确度

Example: Euler's Method

数值方法的准确度

Applying Euler's method to $y' = \lambda y$ using fixed step size h, we have $y_k = y_{k-1} + h\lambda y_{k-1}$

局部误差：$\ell_k = y_k - u_{k-1}(t_k) = (1 + h\lambda)y_{k-1} - y_{k-1}e^{\lambda h}$

$$e^{h\lambda} = 1 + h\lambda + \frac{(h\lambda)^2}{2} + \frac{(h\lambda)^3}{6} + \cdots$$

所以 $\ell_k \sim \mathcal{O}(h^2)$，为1阶准确度

p阶准确度:

$$\ell_k / h_k = \mathcal{O}(h_k^p)$$

For general ODE $y' = f(t, y)$, consider Taylor series

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

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

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

Euler's Method, continued

Subtracting this from Euler's method, we get

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

(整体误差=传递误差+局部截断误差)

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

加入定义局部误差的假设

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

对任何问题**Euler**法均为**1**阶准确度

数值求解ODE-IVP时步长的选取

对给定问题和选定的求解方法，写出 e_{k+1} 和 e_k 之间的关系式

growth factor $I + h_k J_f$

得出误差“增长因子”，根据误差不扩大的稳定性要求得到步长的限制条件

$$h \le -2/\lambda$$

$$\left| h \cdot \mathrm{Eig}(J_f) + 1 \right| \le 1$$

为保证计算的准确性，也需根据局部截断误差选择合适步长

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

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

*tol*为控制局部误差的阈值

We do not know value of y'', but we can estimate it by difference quotient

$$y'' \approx \frac{y'_k - y'_{k-1}}{t_k - t_{k-1}}$$

为了计算效率，寻求稳定算法、高阶算法

$$y_0 e^{(a+bi)t} = y_0 e^{at}(\cos bt + i\sin bt), \quad a = \mathrm{Re}(\lambda)$$