高等数值算法与应用（十一）

Advanced Numerical Algorithms & Applications

计算机科学与技术系 喻文健

Lecture 10 – ODE-IVP Techniques

Two implicit methods

- Backward Euler $y_{k+1} = y_k + h_k f(t_{k+1}, y_{k+1})$
- Disadvantage: solve algebraic equation to get y_{k+1}
- Advantage: unconditionally stable
- Trapezoid (二阶准确度)

对稳定的问题，无论取怎样的步长**h**，<u>求解过程都稳定</u>

$$y_{k+1} = y_k + h_k \left(f(t_k, y_k) + f(t_{k+1}, y_{k+1}) \right) /2$$

Stiffness, stiff problem

Stability: rapid component
Accuracy: slow component

- Two restrictions for step size
- Some numerical methods are inefficient because of severe stability limitation

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

- Factors: ODE; numerical method; initial conditions

Lecture 10 – ODE-IVP Techniques

Runge-Kutta method

- Single-step, need not compute higher derivatives
- Order-2 scheme $y_{k+1} = y_k + \frac{h_k}{2}(k_1 + k_2)$
- Order-4 scheme $y_{k+1} = y_k + \frac{h_k}{6}(k_1 + 2k_2 + 2k_3 + k_4)$
- Self-starting; automatic Runge-Kutta solver

Multistep method

- 线性多步 $y_{k+1} = \sum_{i=1}^{m} \alpha_i y_{k+1-i} + h \sum_{i=0}^{m} \beta_i f(t_{k+1-i}, y_{k+1-i})$
- 减少隐格式计算：PECE scheme (predict, eval, correct, eval)
- Order-4 Adams PECE, backward differentiation formulas

局部截断误差阶与稳定区间

不容易更改步长

- 设等号右边项准确，$y_k = y(t_k)$，计算$l_{k+1} = O(h^{p+1})$
- 推导误差增长因子$e_{k+1} = (?)e_k + O(h^{p+1})$, $|(?)| \le 1$

也可对模型问题推导

对模型问题(λ为实数/复数)推导，得到稳定阈值，即**hλ≥?**

Matlab issues

- Matlab commands for ODE
  - nonstiff: **ode45** (Runge-Kutta(4,5)), **ode23** (Runge-Kutta(2,3)), **ode113** (variable order Adams PECE)
  - moderately stiff: **ode23t** (trapezoidal)
  - stiff: **ode23tb** (implicit Runge-Kutta), **ode15s** (BDF/Gear's method), **ode23s** (Rosenbrock order 2)
  - 都是Variable-order/variable-step solver
  - **deval**: Evaluate solution of differential equation problem 求解ODE后，想求任意时间点对应的函数值
  - For more, ask "Matlab help"

Ordinary Differential Equations (BVP)

Wenjian Yu

内容概要

- 常微分方程边值问题的数值解法
  - 问题的定义与分类 (两点边值问题)
  - 解的存在性、唯一性
  - **BVP**问题的稳定性
  - 求解**BVP**问题的数值方法
    - 打靶**(shooting)**法
    - 有限差分法
    - **Collocation**法（配点法）
    - **Galerkin**法
  - 特征值问题
  - **Matlab**的有关功能

Boundary Value Problems

IVP:
time integration

规定

Side conditions prescribing solution or derivative values at specified points required to make solution of ODE unique

In initial value problem, all side conditions specified at single point, say t_0

边界条件

In *boundary value problem* (BVP), side conditions specified at more than one point

kth order ODE, or equivalent first-order system, requires k side conditions

需要**k**个边界条件

For ODE, side conditions typically specified at two points, endpoints of interval $[a, b]$, so we have two-point boundary value problem

我们只讨论两点
边值问题

General first-order two-point BVP has form 两点边值问题

$$y' = f(t, y), \qquad a < t < b,$$

with boundary conditions

$$g(y(a), y(b)) = o,$$

where $f\colon \mathbb{R}^{n+1} \to \mathbb{R}^n$ and $g\colon \mathbb{R}^{2n} \to \mathbb{R}^n$

n个一阶方程,
n个边界条件

Boundary conditions are *separated* if any given component of g involves solution values only at a or at b, but not both

可分离边界条件

Boundary conditions are *linear* if of form 线性边界条件

$$B_a\, y(a) + B_b\, y(b) = c,$$

where $B_a, B_b \in \mathbb{R}^{n\times n}$ and $c \in \mathbb{R}^n$ 常数矩阵、向量

BVP is *linear* if both ODE and boundary conditions are linear 线性**BVP**问题

Example: Separated Linear BC

可分离、线性边界条件

Two-point BVP for second-order scalar ODE

$$u'' = f(t, u, u'), \qquad a < t < b,$$

with boundary conditions

$$u(a) = \alpha, \qquad u(b) = \beta,$$

is equivalent to first-order system of ODEs

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = \begin{bmatrix} y_2 \\ f(t, y_1, y_2) \end{bmatrix}, \qquad a < t < b,$$

以后常讨论二阶标量**ODE**的例子，**why?**

with separated linear boundary conditions

$$\underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}}_{B_a} \underbrace{\begin{bmatrix} y_1(a) \\ y_2(a) \end{bmatrix}}_{a\text{点的函数值}} + \underbrace{\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}}_{B_b} \begin{bmatrix} y_1(b) \\ y_2(b) \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

可分离、线性边界条件：对$1\le i \le n$, B_a的第i行全为零或B_b的第i行全为零

Existence and Uniqueness

Unlike IVP, with BVP we cannot begin at initial point and continue solution step by step to nearby points

Instead, solution determined everywhere simultaneously, so existence and/or uniqueness may not hold

For example,

$$u'' = -u, \qquad 0 < t < b,$$

with boundary conditions

$$u(0) = 0, \qquad u(b) = \beta,$$

with b integer multiple of π, has infinitely many solutions if $\beta = 0$, but no solution if $\beta \neq 0$

IVP解的存在唯一性一般都成立(直观上好理解)

BVP的情况则不同

解析解：

$$\boldsymbol{u} = c_1 \sin(t) + c_2 \cos(t)$$

If $b = k\pi$

Existence and Uniqueness, continued

In general, solvability of BVP

$$y' = f(t, y), \qquad a < t < b,$$

with boundary conditions

$$g(y(a), y(b)) = o,$$

depends on solvability of algebraic equation

$$g(x, y(b; x)) = o,$$

where $y(t; x)$ denotes solution to ODE with initial condition $y(a) = x$ for $x \in \mathbb{R}^n$

$y(t;\ x)$表示相应初值问题的解

关于x的方程

Solvability of latter system is difficult to establish if g is nonlinear

非线性方程的可解性很复杂，难于讨论

For *linear* BVP, existence and uniqueness are more tractable 易驾驭、处理的

线性BVP问题的解的存在、唯一性

Consider linear BVP

$$y' = A(t)\, y + b(t), \qquad a < t < b,$$

where $A(t)$ and $b(t)$ are continuous, with boundary conditions

$$B_a\, y(a) + B_b\, y(b) = c$$

Fundamental solution matrix

Let $Y(t)$ denote matrix whose ith column, $Y_i(t)$, called ith *mode*, is solution to $y' = A(t)y$ with initial condition $y(a) = e_i$ 单位基向量

模态

Then BVP has unique solution if, and only if, matrix

$$Q \equiv B_a Y(a) + B_b Y(b)$$

is nonsingular

思考前一例子

$u'' = -u$

$Q = B_a + B_b Y(b)$

Existence & Uniqueness for Linear BVP

mode: $\begin{cases} \boldsymbol{y}' = \boldsymbol{A}(t)\boldsymbol{y} \\ \boldsymbol{y}(a) = \boldsymbol{e}_i \end{cases}$ 初值问题解 $\boldsymbol{Y}_i$，则 $\begin{cases} \boldsymbol{Y}_i' = \boldsymbol{A}(t)\boldsymbol{Y}_i \\ \boldsymbol{Y}_i(a) = \boldsymbol{e}_i \end{cases}$

考察函数 $\boldsymbol{y} = \left[\boldsymbol{Y}_1 \boldsymbol{Y}_2 \cdots \boldsymbol{Y}_n\right] \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix} = \boldsymbol{Y}\boldsymbol{x}$ ，$\boldsymbol{x}_i$ 为实数, 是齐次问题的通解

$\boldsymbol{y}' = \boldsymbol{A}(t)\,\boldsymbol{y}$ $\quad a < t < b$

非齐次解为： $\boldsymbol{y} = \boldsymbol{Y}\boldsymbol{x} + \boldsymbol{p}(t)$ **(带入ODE验证)**

$\boldsymbol{p}(t)$是满足 $\boldsymbol{p}'(t) - \boldsymbol{A}(t)\boldsymbol{p}(t) = \boldsymbol{b}(t)$ 的非齐次问题的一个特解

$$\boldsymbol{p}(t) = \boldsymbol{Y}(t)\int_a^t \boldsymbol{Y}^{-1}(s)\boldsymbol{b}(s)ds$$

(验证？)

利用边界条件 $B_a\,\boldsymbol{y}(a) + B_b\,\boldsymbol{y}(b) = c$ 求$\boldsymbol{x}$

$$\boldsymbol{B}_a\boldsymbol{Y}(a)\boldsymbol{x} + \boldsymbol{B}_a\boldsymbol{p}(a) + \boldsymbol{B}_b\boldsymbol{Y}(b)\boldsymbol{x} + \boldsymbol{B}_b\boldsymbol{p}(b) = \boldsymbol{c}$$

$$\boldsymbol{x} = \left[\boldsymbol{B}_a\boldsymbol{Y}(a) + \boldsymbol{B}_b\boldsymbol{Y}(b)\right]^{-1}\left[\boldsymbol{c} - \boldsymbol{B}_a\boldsymbol{p}(a) - \boldsymbol{B}_b\boldsymbol{p}(b)\right]$$

$\boldsymbol{Q}$ matrix

线性ODE-BVP问题的解存在/唯一等价于矩阵Q非奇异

Existence and Uniqueness, continued

$$\boldsymbol{x} = \left[\boldsymbol{B}_a\boldsymbol{Y}(a) + \boldsymbol{B}_b\boldsymbol{Y}(b)\right]^{-1} \cdot \left[\boldsymbol{c} - \boldsymbol{B}_a\boldsymbol{p}(a) - \boldsymbol{B}_b\boldsymbol{p}(b)\right] = \boldsymbol{Q}^{-1}(\boldsymbol{c} - \boldsymbol{B}_b\boldsymbol{p}(b)) \quad \boldsymbol{p}(a) = \boldsymbol{0}$$

$$\boldsymbol{y} = \boldsymbol{Y}(t)\boldsymbol{x} + \boldsymbol{p}(t) = \boldsymbol{Y}(t)\boldsymbol{Q}^{-1}(\boldsymbol{c} - \boldsymbol{B}_b\boldsymbol{p}(b)) + \boldsymbol{p}(t)$$

$$\boldsymbol{p}(t) = \boldsymbol{Y}(t)\int_a^t \boldsymbol{Y}^{-1}(s)\boldsymbol{b}(s)ds$$

注意：实际的 $\boldsymbol{Y}(t)$若无解析表达，此公式不实用

稳定性分析：考虑非齐次项和边界条件的扰动

$$\left\|\hat{\boldsymbol{y}}(t) - \boldsymbol{y}(t)\right\|_\infty \le \kappa\left(\|\Delta\boldsymbol{c}\| + \int_a^b \|\Delta\boldsymbol{b}(s)\| ds\right)$$

Assuming $\boldsymbol{Q}$ is nonsingular, define

$$\boldsymbol{\Phi}(t) = \boldsymbol{Y}(t)\,\boldsymbol{Q}^{-1}$$

and *Green's function* 格林函数

$$\boldsymbol{G}(t,s) = \begin{cases} \boldsymbol{\Phi}(t)\boldsymbol{B}_a\boldsymbol{\Phi}(a)\boldsymbol{\Phi}^{-1}(s), & a \le s \le t \\ -\boldsymbol{\Phi}(t)\boldsymbol{B}_b\boldsymbol{\Phi}(b)\boldsymbol{\Phi}^{-1}(s), & t < s \le b \end{cases}$$

课后思考

Then solution to BVP given by

$$\boldsymbol{y}(t) = \boldsymbol{\Phi}(t)\,\boldsymbol{c} + \int_a^b \boldsymbol{G}(t,s)\,\boldsymbol{b}(s)\,ds,$$

This result also gives absolute condition number for BVP,

$$\kappa = \max\{\|\boldsymbol{\Phi}\|_\infty, \|\boldsymbol{G}\|_\infty\}$$

条件数

$$\left\|\hat{\boldsymbol{y}}(t)-\boldsymbol{y}(t)\right\|_\infty \le \kappa\left(\|\Delta\boldsymbol{c}\| + \int_a^b \|\Delta\boldsymbol{b}(s)\|\,ds\right)$$

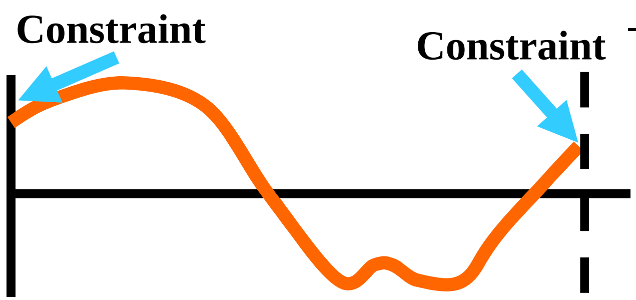
$$\kappa = \max\{\|\boldsymbol{\Phi}\|_\infty, \|\boldsymbol{G}\|_\infty\}$$

Conditioning or stability of BVP depends on interplay between growth of solution modes and boundary conditions

For IVP, instability is associated with modes that grow exponentially as time increases

IVP问题定义域: $t_0 \to \infty$

For BVP, solution is determined everywhere simultaneously, so there is no notion of "direction" of integration in interval $[a, b]$



Growth of modes increasing with time is limited by boundary conditions at b, and "growth" of decaying modes is limited by boundary conditions at a

类似于一般问题的病态性分析，**BVP**问题不会出现误差无限放大的情况 **(**这段不同于**IVP**问题**)**

思考：**pp. 367,** 例**10.5**，推导 $\Phi(t)$ 及条件数

Numerical Methods for BVPs

For IVP, initial data supply all information necessary to begin numerical solution method at initial point and step forward from there

For BVP, we have insufficient information to begin step-by-step numerical method, so numerical methods for solving BVPs are more complicated than those for solving IVPs

We consider four types of numerical methods for two-point BVPs:

- Shooting 试射法
- Finite difference 有限差分
- Collocation 配点，点配置
- Galerkin 伽辽金法

解线性方程组的投影方法——**Galerkin**原理

Shooting Method

最简单的两点边值问题

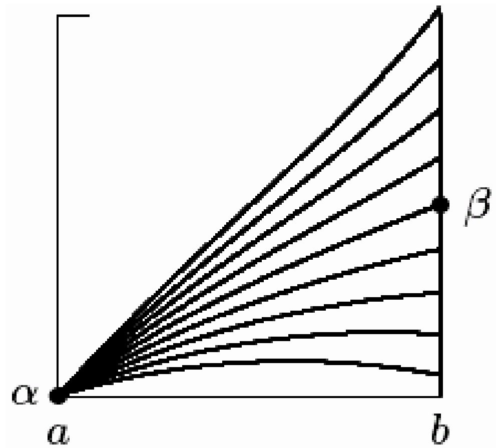
In statement of two-point BVP, we are given value of $u(a)$

$$u'' = f(t, u, u'), \qquad a < t < b, \quad u(a) = \alpha, \qquad u(b) = \beta,$$

If we also knew value of $u'(a)$, then we would have IVP that we could solve by methods previously discussed

Lacking that information, we try sequence of increasingly accurate guesses until we find value for $u'(a)$ such that when we solve resulting IVP, approximate solution value at $t = b$ matches desired boundary value, $u(b) = \beta$

进行多次time integration



Shooting Method, continued

For given γ, value at b of solution $u(b)$ to IVP

$$u'' = f(t, u, u'),$$

with initial conditions

$$u(a) = \alpha, \qquad u'(a) = \gamma,$$

can be considered as function of γ, say $g(\gamma)$

Then BVP becomes problem of solving equation $g(\gamma) = \beta$

One-dimensional zero finder can be used to solve this scalar equation

课本 § **5.5**
Matlab命令fzero

一般的求根方法，如：
Interval bisection
Fixed-point iteration
Newton's method
……

Example: Shooting Method

Consider two-point BVP for second-order ODE

$$u'' = 6t, \qquad 0 < t < 1,$$

with BC

$$u(0) = 0, \qquad u(1) = 1$$

For each guess for $u'(0)$, we integrate ODE using classical fourth-order Runge-Kutta method to determine how close we come to hitting desired solution value at $t = 1$

We transform second-order ODE into system of two first-order ODEs

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = \begin{bmatrix} y_2 \\ 6t \end{bmatrix}$$

Example Continued

We try initial slope of $y_2(0) = 1$ → 即$\boldsymbol{u}'$

Using step size $h = 0.5$, we first step from $t_0 = 0$ to $t_1 = 0.5$

Classical fourth-order Runge-Kutta method gives approximate solution value at t_1

$$y^{(1)} = y^{(0)} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$= \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \frac{0.5}{6}\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 1.0 \\ 1.5 \end{bmatrix} \right.$$

$$\left. +2\begin{bmatrix} 1.375 \\ 1.500 \end{bmatrix} + \begin{bmatrix} 1.75 \\ 3.00 \end{bmatrix} \right) = \begin{bmatrix} 0.625 \\ 1.750 \end{bmatrix}$$

Example Continued

Next we step from $t_1 = 0.5$ to $t_2 = 1$, getting

$$y^{(2)} = \begin{bmatrix} 0.625 \\ 1.750 \end{bmatrix} + \frac{0.5}{6}\left(\begin{bmatrix} 1.75 \\ 3.00 \end{bmatrix} + 2\begin{bmatrix} 2.5 \\ 4.5 \end{bmatrix}\right.$$

$$\left. +2\begin{bmatrix} 2.875 \\ 4.500 \end{bmatrix} + \begin{bmatrix} 4 \\ 6 \end{bmatrix}\right) = \begin{bmatrix} 2 \\ 4 \end{bmatrix},$$

so we have hit $y_1(1) = 2$ instead of desired value $y_1(1) = 1$

We try again, this time with initial slope ~~$y_2(0) = -1$,~~ obtaining

$$y^{(1)} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} + \frac{0.5}{6}\left(\begin{bmatrix} -1 \\ 0 \end{bmatrix} + 2\begin{bmatrix} -1.0 \\ 1.5 \end{bmatrix}\right.$$

$$\left. +2\begin{bmatrix} -0.625 \\ 1.500 \end{bmatrix} + \begin{bmatrix} -0.25 \\ 3.00 \end{bmatrix}\right) = \begin{bmatrix} -0.375 \\ -0.250 \end{bmatrix},$$

Example Continued

$$y^{(2)} = \begin{bmatrix} -0.375 \\ -0.250 \end{bmatrix} + \frac{0.5}{6}\left(\begin{bmatrix} -0.25 \\ 3.00 \end{bmatrix} + 2\begin{bmatrix} 0.5 \\ 4.5 \end{bmatrix}\right.$$

$$\left. +2\begin{bmatrix} 0.875 \\ 4.500 \end{bmatrix} + \begin{bmatrix} 2 \\ 6 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 2 \end{bmatrix},$$

so we have hit $y_1(1) = 0$ instead of desired value $y_1(1) = 1$

We now have initial slope bracketed between -1 and 1

We omit further iterations necessary to identify correct initial slope, which turns out to be $y_2(0) = 0$:

$$y^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{0.5}{6}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 0.0 \\ 1.5 \end{bmatrix}\right.$$

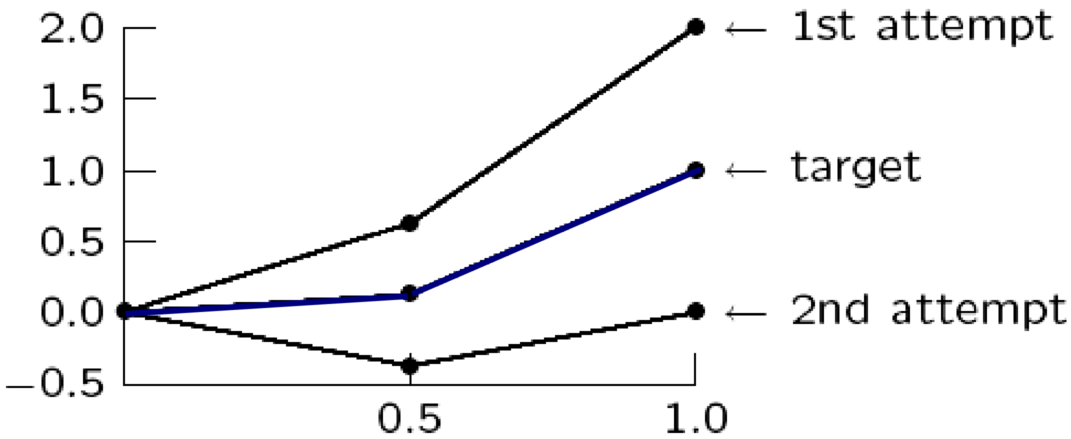
$$\left. +2\begin{bmatrix} 0.375 \\ 1.500 \end{bmatrix} + \begin{bmatrix} 0.75 \\ 3.00 \end{bmatrix}\right) = \begin{bmatrix} 0.125 \\ 0.750 \end{bmatrix},$$

Example Continued

$$y^{(2)} = \begin{bmatrix} 0.125 \\ 0.750 \end{bmatrix} + \frac{0.5}{6}\left(\begin{bmatrix} 0.75 \\ 3.00 \end{bmatrix} + 2 \begin{bmatrix} 1.5 \\ 4.5 \end{bmatrix} \right.$$

$$\left. + 2 \begin{bmatrix} 1.875 \\ 4.500 \end{bmatrix} + \begin{bmatrix} 3 \\ 6 \end{bmatrix} \right) = \begin{bmatrix} 1 \\ 3 \end{bmatrix},$$

so we have indeed hit target solution value, $y_1(1) = 1$



程序实现

- 定义常微分方程:

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = \begin{bmatrix} y_2 \\ 6t \end{bmatrix}$$

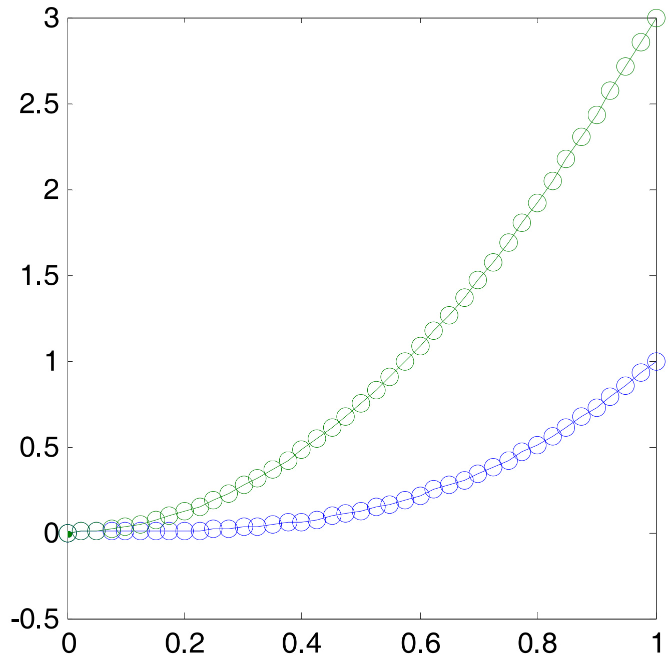
```
function dydt=lec11ode(t, y)
    dydt=[y(2)
          6*t ];
```

- IVP问题的求解器: **ode45, 等等**

- 定义非线性方程 $g(x)\text{-}y_1(b)=0$

额外参数

```
function res=lec11equ(x, a, b, y1a, y1b)
    [T, Y]=ode45(@lec11ode, [a, b], [y1a; x]);
    res= Y(size(T, 1), 1)-y1b;     %g(x)-y1(b)
```



- 求解非线性方程用**fzero:**

指定有根区间,
也可只输一个数

```
a=0; b=1; y1a=0; y1b=1;
x=fzero(@lec11equ, [-1, 1], [], a, b, y1a, y1b)
ode45(@lec11ode, [ts, te], [y1s; x]);  %验证结果
```

Multiple Shooting

Simple shooting method inherits stability (or instability) of associated IVP, which may be unstable even when BVP is stable

IVP及求解的稳定性影响非线性方程迭代求解过程的收敛

Such ill-conditioning may make it difficult to achieve convergence of iterative method for solving nonlinear equation

区间减小，不稳定性也减小

Potential remedy is *multiple shooting*, in which interval $[a, b]$ is divided into subintervals, and shooting is carried out on each

一般的**n**维一阶向量**ODE**，需要最多设**n**个初值实现**shooting;**

Requiring continuity at internal mesh points provides BC for individual subproblems

若分**m**个子区间，变成**mn**维一阶向量**ODE**，最多设**mn**个初值实现**shooting**，同时增加**n(m-1)**个中间结点连续性条件

高维非线性方程的求解

Multiple shooting results in larger *system* of nonlinear equations to solve

Finite Difference Method

Finite difference method converts BVP into system of algebraic equations by replacing all derivatives by finite difference approximations

For example, to solve two-point BVP

$$u'' = f(t, u, u'), \qquad a < t < b,$$

with BC

$$u(a) = \alpha, \qquad u(b) = \beta,$$

we introduce mesh points $t_i = a + ih$, $i = 0, 1, \ldots, n+1$, where $h = (b-a)/(n+1)$

We already have $y_0 = u(a) = \alpha$ and $y_{n+1} = u(b) = \beta$, and we seek approximate solution value $y_i \approx u(t_i)$ at each mesh point t_i, $i = 1, \ldots, n$

We replace derivatives by finite difference quotients, such as

注意：这里y_i是离散变量，代表函数u在离散点上的值

$$u'(t_i) \approx \frac{y_{i+1} - y_{i-1}}{2h}$$

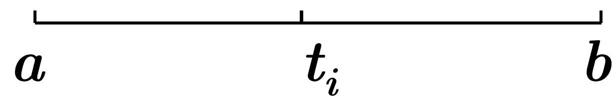
and

$$u''(t_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2},$$

截断误差$O(h^2)$

yielding system of equations

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = f\left(t_i, y_i, \frac{y_{i+1} - y_{i-1}}{2h}\right),$$



to be solved for unknowns y_i, $i = 1, \ldots, n$

System of equations may be linear or nonlinear, depending on whether f is linear or nonlinear

线性：三对角线性方程组
非线性：**Jacobi**矩阵为三对角矩阵

这是一维问题，对高维问题**FDM**总是生成条带稀疏矩阵

Example: Finite Difference Method

Consider two-point BVP

$$u'' = 6t, \qquad 0 < t < 1,$$

with BC

$$u(0) = 0, \qquad u(1) = 1$$

To keep computation to minimum, we compute approximate solution at one mesh point in interval $[0, 1]$, $t = 0.5$

Including boundary points, we have three mesh points, $t_0 = 0$, $t_1 = 0.5$, and $t_2 = 1$

From BC, we know that $y_0 = u(t_0) = 0$ and $y_2 = u(t_2) = 1$, and we seek approximate solution $y_1 \approx u(t_1)$

Example Continued

Approximating second derivative by standard finite difference quotient at t_1 gives equation

$$\frac{y_2 - 2y_1 + y_0}{h^2} = f\left(t_1, y_1, \frac{y_2 - y_0}{2h}\right)$$

Substituting boundary data, mesh size, and right hand side for this example,

$$\frac{1 - 2y_1 + 0}{(0.5)^2} = 6t_1,$$

or

$$4 - 8y_1 = 6(0.5) = 3,$$

so that

$$y(0.5) \approx y_1 = 1/8 = 0.125$$

which agrees with approximate solution at $t = 0.5$ that we previously computed by shooting method

Example Continued

只讨论了最简单的两点边值问题，对一般问题还有更复杂的差分格式和方程构造

In a practical problem, much smaller step size and many more mesh points would be required to achieve acceptable accuracy

We would therefore obtain *system* of equations to solve for approximate solution values at mesh points, rather than single equation as in this example

通常求解一个大的方程组

关于收敛性：当网格点数目增加时，**FDM**的解逼近准确解

两个条件：差分格式的截断误差 **→ 0**
小扰动带来的误差有界

一致**(**相容**)**性和稳定性

(对**BVP**问题
一般都成立**)**

Collocation Method

Collocation method approximates solution to BVP by finite linear combination of basis functions

For two-point BVP

$$u'' = f(t, u, u'), \qquad a < t < b,$$

with BC

$$u(a) = \alpha, \qquad u(b) = \beta,$$

we seek approximate solution of form

$$u(t) \approx v(t, \boldsymbol{x}) = \sum_{i=1}^{n} x_i \phi_i(t),$$

where ϕ_i are basis functions defined on $[a, b]$ and $\boldsymbol{x}$ is n-vector of parameters to be determined

Collocation Method

Popular choices of basis functions include polynomials, B-splines, and trigonometric functions

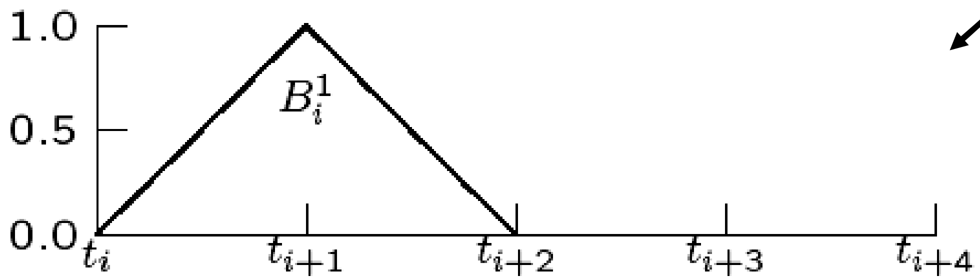
多项式函数、或B-样条函数、或三角函数

谱方法

Basis functions with global support, such as polynomials or trigonometric functions, yield *spectral* or *pseudospectral method*

有限元法

Basis functions with highly localized support, such as B-splines, yield *finite element method*



Collocation Method, continued

To determine vector of parameters x, define set of n *collocation points*, $a = t_1 < \cdots < t_n = b$, at which approximate solution $v(t, x)$ is forced to satisfy ODE and boundary conditions

这就是点配置法的根本

即微分方程的剩余为**0**

Common choices of collocation points include equally-spaced mesh or Chebyshev points

选取的基函数都可以解析微分

Suitably smooth basis functions can be differentiated analytically, so that approximate solution and its derivatives can be substituted into ODE and BC to obtain system of algebraic equations for unknown parameters x

得到关于x 的代数方程组

Example: Collocation Method

Consider again two-point BVP

$$u'' = 6t, \qquad 0 < t < 1,$$

with BC

$$u(0) = 0, \qquad u(1) = 1$$

To keep computation to minimum, we use one interior collocation point, $t = 0.5$

为了说明的简单，只取一个配置点

Including boundary points, we have three collocation points, $t_0 = 0$, $t_1 = 0.5$, and $t_2 = 1$, so we will be able to determine three parameters

As basis functions we use first three monomials, so approximate solution has form

$$v(t, \boldsymbol{x}) = x_1 + x_2 t + x_3 t^2$$

Example Continued

Derivatives of approximate solution function with respect to t are given by

$$v'(t, \boldsymbol{x}) = x_2 + 2x_3 t, \qquad v''(t, \boldsymbol{x}) = 2x_3$$

Requiring ODE to be satisfied at interior collocation point $t_2 = 0.5$ gives equation

$$v''(t_2, \boldsymbol{x}) = f(t_2, v(t_2, \boldsymbol{x}), v'(t_2, \boldsymbol{x})),$$

or

$$2x_3 = 6t_2 = 6(0.5) = 3$$

Left boundary condition at $t_1 = 0$ gives equation

$$x_1 + x_2 t_1 + x_3 t_1^2 = x_1 = 0,$$

and right boundary condition at $t_3 = 1$ gives equation

$$x_1 + x_2 t_3 + x_3 t_3^2 = x_1 + x_2 + x_3 = 1$$

Example Continued

Solving this system of three equations in three unknowns gives

$$x_1 = 0, \qquad x_2 = -0.5, \qquad x_3 = 1.5,$$

so approximate solution function is quadratic polynomial

$$u(t) \approx v(t, \boldsymbol{x}) = -0.5t + 1.5t^2$$

At interior collocation point, $t_2 = 0.5$, we have approximate solution value

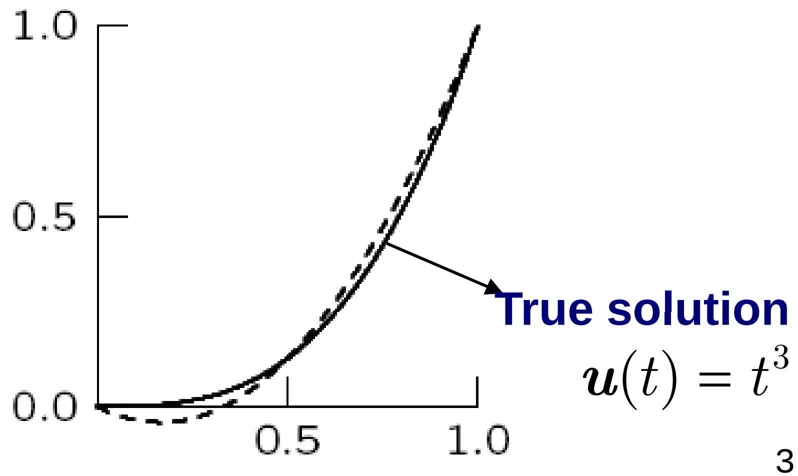
$$u(0.5) \approx v(0.5, \boldsymbol{x}) = 0.125,$$

which agrees with solution value at $t = 0.5$ obtained previously by other two methods

在配置点上满足ODE并不意味着在这点上解准确

事实上，近似解仅在这三个时间点上和准确解相等。

碰巧**Collocation**的解和前面用**shooting, FDM**得到的完全一样
不同之处：得到解析的解函数



Rather than forcing residual to be zero at finite number of points, as in collocation, we could instead minimize residual over entire interval of integration

在整个区间最小化剩余，而不是在几个点上设其为零

For example, for scalar *Poisson* equation in one dimension,

$$u'' = f(t), \qquad a < t < b,$$

with homogeneous boundary conditions

$$u(a) = 0, \qquad u(b) = 0,$$

subsitute approximate solution

$$u(t) \approx v(t, \boldsymbol{x}) = \sum_{i=1}^{n} x_i \phi_i(t)$$

into ODE and define residual

连续函数逼近

$$r(t, \boldsymbol{x}) = v''(t, \boldsymbol{x}) - f(t) = \sum_{i=1}^{n} x_i \phi_i''(t) - f(t)$$

线性最小二乘

函数的二范数、内积…

Using *least squares* method, we can minimize

$$F(\boldsymbol{x}) = \frac{1}{2}\int_a^b r(t,\boldsymbol{x})^2\,dt$$

关于$\boldsymbol{x}$的多变量函数

by setting each component of its gradient to zero, which yields symmetric system of linear algebraic equations $\boldsymbol{Ax} = \boldsymbol{b}$, where

$$a_{ij} = \int_a^b \phi_j''(t)\phi_i''(t)\,dt \quad \text{and} \quad b_i = \int_a^b f(t)\phi_i''(t)\,dt,$$

whose solution gives vector of parameters $\boldsymbol{x}$

More generally, *weighted residual* method forces residual to be orthogonal to each of set of *weight functions* or *test functions* w_i, i.e.,

加权余量法

正交于测试函数空间(或投影空间)

$$\int_a^b r(t,\boldsymbol{x})w_i(t)\,dt = 0, \quad i = 1,\ldots,n,$$

which yields linear system $\boldsymbol{Ax} = \boldsymbol{b}$, where now

$$a_{ij} = \int_a^b \phi_j''(t)w_i(t)\,dt \quad \text{and} \quad b_i = \int_a^b f(t)w_i(t)\,dt,$$

whose solution gives vector of parameters $\boldsymbol{x}$

"最小二乘"属于加权余量法?

Galerkin Method, continued

Matrix resulting from weighted residual method is generally not symmetric, and its entries involve second derivatives of basis functions

Both drawbacks overcome by *Galerkin* method, in which weight functions are chosen to be same as basis functions, i.e., $w_i = \phi_i$, $i = 1, \dots, n$

回忆线性方程组求解问题

Orthogonality condition then becomes

$$\int_a^b r(t, \boldsymbol{x})\phi_i(t)\, dt = 0, \quad i = 1, \dots, n,$$

or

$$\int_a^b v''(t, \boldsymbol{x})\phi_i(t)\, dt = \int_a^b f(t)\phi_i(t)\, dt, \quad i = 1, \dots, n$$

$$\boldsymbol{v(t, x)} = \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{\phi}_i$$

$$\boldsymbol{v'(t, x)} = \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{\phi}_i'(\boldsymbol{t})$$

Galerkin Method, continued

分部积分

Degree of differentiability can be reduced using integration by parts, which gives

$$\begin{aligned}\int_a^b v''(t,\boldsymbol{x})\phi_i(t)\,dt &= v'(t)\phi_i(t)|_a^b - \int_a^b v'(t)\phi_i'(t)\,dt \\ &= v'(b)\phi_i(b) - v'(a)\phi_i(a) \\ &\quad - \int_a^b v'(t)\phi_i'(t)\,dt\end{aligned}$$

齐次边界条件

Assuming basis functions ϕ_i satisfy homogeneous boundary conditions, so $\phi_i(0) = \phi_i(1) = 0$, orthogonality condition then becomes

非齐次项移到=右边

$$-\int_a^b v'(t)\phi_i'(t)\,dt = \int_a^b f(t)\phi_i(t)\,dt, \quad i = 1,\ldots,n,$$

which yields system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, with

Stiffness matrix

Load vector

$$a_{ij} = -\int_a^b \phi_j'(t)\phi_i'(t)\,dt \quad \text{and} \quad b_i = \int_a^b f(t)\phi_i(t)\,dt,$$

whose solution gives vector of parameters $\boldsymbol{x}$

对称且，
基函数**1**阶导数可积就行

$\boldsymbol{A}$ is symmetric and involves only first derivatives of basis functions

Example: Galerkin Method

Consider again two-point BVP

$$u'' = 6t, \qquad 0 < t < 1,$$

with BC

$$u(0) = 0, \qquad u(1) = 1$$

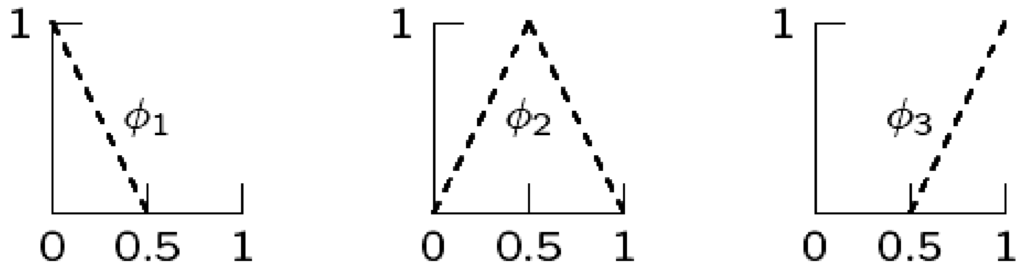
B样条?

详见课本§7.4.3

We will approximate solution by piecewise linear polynomial, for which B-splines of degree 1 ("hat" functions) form suitable set of basis functions

To keep computation to minimum, we again use same three mesh points, but now they become knots in piecewise linear polynomial approximation

绳结



Example Continued

Thus, we seek approximate solution of form

$$u(t) \approx v(t, \boldsymbol{x}) = x_1\phi_1(t) + x_2\phi_2(t) + x_3\phi_3(t)$$

From BC, we must have $x_1 = 0$ and $x_3 = 1$

因为 $\phi_2(t)$满足齐次边界条件，**p.40**的推导成立，即只剩下一阶导数积分

To determine remaining parameter x_2, we impose Galerkin orthogonality condition on interior basis function ϕ_2 and obtain equation

$$-\sum_{j=1}^{3}\left(\int_0^1 \phi_j'(t)\phi_2'(t)\,dt\right)x_j = \int_0^1 6t\phi_2(t)\,dt,$$

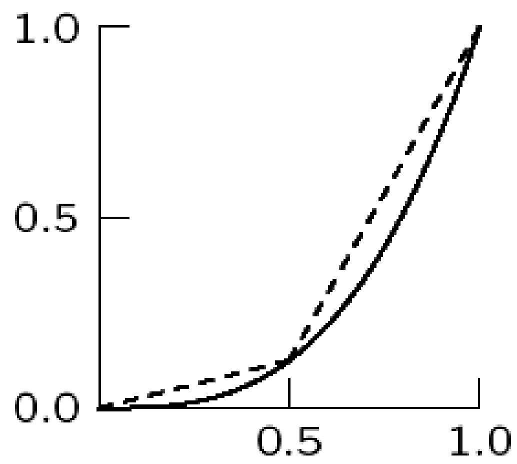
or, upon evaluating these simple integrals analytically,

$$2x_1 - 4x_2 + 2x_3 = 3/2$$

Example Continued

Substituting known values for x_1 and x_3 then gives $x_2 = 1/8$ for remaining unknown parameter, so piecewise linear approximate solution is

$$u(t) \approx v(t, \boldsymbol{x}) = 0.125\phi_2(t) + \phi_3(t)$$



解为分段线性函数

We note that $v(0.5, \boldsymbol{x}) = 0.125$, which again is exact for this particular problem

Example Continued

More realistic problem would have many more interior mesh points and basis functions, and correspondingly many parameters to be determined

Resulting system of equations would be much larger but still sparse, and therefore relatively easy to solve, provided local basis functions, such as "hat" functions, are used

Resulting approximate solution function is less smooth than true solution, but becomes more accurate as more mesh points are used

小结

- 加权余量法: $u(t) \approx v(t,x) = \sum_{i=1}^{n} x_i \phi_i(t),$ 基函数、试函数

  $\int_a^b r(t,x) w_i(t)\, dt = 0, \quad i = 1, \ldots, n,$ $\boldsymbol{r(t,x) = v'' - f(t,v,v'')}$

  - 点配置法适用范围广(试函数为δ函数)
  - Galerkin法的试函数与基函数相同，对某些问题得到形式较为简单的方程(基函数可导性要求低、矩阵对称)
- 谱方法和有限元法(按基函数的不同分类)

  Global support vs. local support

  Dense matrix vs. sparse matrix **h, p refinement**

  One way vs. two ways to improve accuracy
- 只对最简单的两点边值问题做了介绍，对一般问题常采用**local support**基函数，方程的构造也较复杂 **(bvp4c)**

Eigenvalue Problems

Standard eigenvalue problem for second-order ODE has form

$$u'' = \lambda f(t, u, u'), \qquad a < t < b,$$

f 为已知函数

with BC

$$u(a) = \alpha, \qquad u(b) = \beta,$$

where we seek not only solution u but also parameter λ as well

Scalar λ (possibly complex) is *eigenvalue* and solution u corresponding *eigenfunction* for this two-point BVP

Discretization of eigenvalue problem for ODE results in algebraic eigenvalue problem whose solution approximates that of original problem

Example: Eigenvalue Problem

线性两点边值问题
的特征值问题

Consider linear two-point BVP

$$u'' = \lambda g(t) u, \qquad a < t < b,$$

with BC

$$u(a) = 0, \qquad u(b) = 0$$

有限差分离散

Introduce discrete mesh points t_i in interval $[a, b]$, with mesh spacing h and use standard finite difference approximation for second derivative to obtain algebraic system

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = \lambda g_i y_i, \quad i = 1, \dots, n,$$

where $y_i = u(t_i)$ and $g_i = g(t_i)$, and from BC $y_0 = u(a) = 0$ and $y_{n+1} = u(b) = 0$

Example Continued

Assuming $g_i \neq 0$, divide equation i by g_i for $i = 1, \ldots, n$, to obtain linear system

$$\boldsymbol{A}\boldsymbol{y} = \lambda \boldsymbol{y},$$

where $n \times n$ matrix $\boldsymbol{A}$ has tridiagonal form

$$\boldsymbol{A} = \frac{1}{h^2} \begin{bmatrix} -2/g_1 & 1/g_1 & 0 & \cdots & 0 \\ 1/g_2 & -2/g_2 & 1/g_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1/g_{n-1} & -2/g_{n-1} & 1/g_{n-1} \\ 0 & \cdots & 0 & 1/g_n & -2/g_n \end{bmatrix}$$

This standard algebraic eigenvalue problem can be solved by methods discussed previously

矩阵特征值的求解问题

Matlab有关命令

- bvp4c, bvp5c
  - 使用局部支撑的点配置法(collocation method)
  - Syntax: sol = bvp4c(odefun, bcfun, solinit)
  - odefun: $\boldsymbol{y}'=\boldsymbol{f}(t, \boldsymbol{y})$
  - bcfun: $\boldsymbol{g}(\boldsymbol{y}(a), \boldsymbol{y}(b))=\boldsymbol{0}$
  - Solinit= bvpinit(x, yinit)设定网格点和非线性方程的迭代初始解
  - x代表自变量初始网格位置, yinit为长度n的向量， yinit_i为$\boldsymbol{y}_i$在所有网格点上的初始解
  - 程序会自动调整网格; 采用分段三次函数近似
  - 程序可能出错（得到的非线性方程**Jacobi**矩阵奇异）

Matlab有关命令

- 例子 $u'' = 6t, \quad 0 < t < 1, \quad u(0) = 0, \quad u(1) = 1$
  - 定义边界条件的函数：
  - function res=lec11bc(ya, yb)
  -     res=[ya(1)
  -             yb(1)-1];
  - 求解命令：
  - solinit = bvpinit(linspace(0,1,5),[0.5 0.5]);
  - sol=bvp4c(@lec11ode,@lec11bc,solinit);
  - plot(sol.x, sol.y, 'o-')
  - x=linspace(0,1, 101); y=deval(sol, x);
  - figure(2); plot(x, y, ‘o-’)   %更精细绘图