

Contents

2	Numerical Approximation in Functions	2
2.1	方程求根	2
2.1.1	The mathematical model	2
2.1.2	Numerical methods for $f(x) = 0$	3
2.1.3	Numerical methods for equation system	5
2.2	函数逼近问题	6
2.2.1	Interpolation with polynomials	7
2.2.2	Interpolation with piecewise polynomials	9
2.2.3	Least square approximation	11
2.2.4	Best approximation	11
3	Numerical Approximation in Calculus	12
3.1	数值求积问题	12
3.1.1	积分问题	12
3.1.2	数值积分公式	13
3.1.3	插值型公式(Newton-Cotes系列)	14
3.1.4	提高积分精度的方法	15
3.1.5	最佳求积	15
3.1.6	突破维数诅咒: Monte Carlo求积方法	16
3.2	Ordinary Differential Equations for Initial Value Problems(IVPs)	16
3.2.1	Initial Value Problem	16
3.2.2	Numerical Methods for ODEs	17
3.2.3	Stability Improvements	20

Chapter 2

Numerical Approximation in Functions

2.1 Roots

2.1.1 The mathematical model

Definition 2.1.1 (求根问题). 已知 \mathbf{f} 是从 \mathbb{R}^n 到 \mathbb{R}^n 的一个非线性映射。对于任意给定的 $\mathbf{y} \in \mathbb{R}^n$, 问: \mathbf{x} 取何值时, \mathbf{f} 的取值恰好为 \mathbf{y} , 即使得

$$\mathbf{f}(\mathbf{x}) = \mathbf{y}$$

成立? 更一般地, 将两边同时减去 \mathbf{y} , 使其变为标准形式(为简单起见这里仍用 f 表示新的映射):

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \quad (2.1)$$

例2.1.1. 一元非线性方程

$$f(x) = x^2 - 4 \sin x = 0$$

的一个近似解为 $x = 1.93375$ 。二元非线性方程组

$$f(x) = \begin{bmatrix} x^2 - y + 0.25 \\ -x + y^2 + 0.25 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

的解为 $x = [0.5, 0.5]^T$.

对于一般的 $f(x) = 0$, 我们无法给出该非线性方程组解的通用表达式。线性方程组只有 $0/1/\infty$ 个解等情况, 非线性方程组解的个数可能是任意个。一个经典的做法是以一维情形为突破线索:

例2.1.2 (Some classical 1-dimensional problems). $e^x + 1 = 0$, unsolvable

$e^x - x = 0$, unique solution

$x^2 - 4 \sin x = 0$, multi solutions

$\sin x = 0$, infinite solutions

回顾数学分析中的方法，几个存在性定理是研究方程解存在性的几类典型方法：

Lemma 2.1.1 (介值定理). 若 f 是闭区间上 $[a, b]$ 的连续函数， c 介于 $f(a)$ 和 $f(b)$ ，之间，则必存在一个 $x^1 \in [a, b]$ ，满足 $f(x^1) = c$ ，取 c 为 0 即可证明 f 在 $[a, b]$ 上一定有根。

其次，反函数定理 $\mathbf{x} = f^{-1}(\mathbf{0})$ 。更进一步地，还有压缩映射理论(也是迭代法收敛性的基本定理)也可以获得解的存在性证明。

Definition 2.1.2 (不动点). $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ 是集合 $S \in \mathbb{R}^n$ 上的压缩映射，即存在 $\gamma \in [0, 1]$ ，对任意集合 S 内的两个点 x, z 满足 $\|f(x) - f(z)\| \leq \gamma \|x - z\|$ 。满足 $g(x) = x$ 的 x 称为 g 的不动点。

定理 2.1.1 (压缩映射(不动点)定理). 若 g 在闭集 S 上是压缩映射，且 $g(S) \in S$ ，则 g 在 S 内存在唯一的不动点。此时如果非线性方程有形如 $f = x - g$ ，则我们可称， $f = 0$ 在闭集 S 上是有唯一解的。

而关于根的进一步研究，需要借助 topological degree of function f 与重根理论。

2.1.2 Numerical methods for $f(x) = 0$

让我们通过讨论标量非线性方程的求根方法来回顾

二分法

假设我们有一个比较小的区间 $[a, b]$ ，而函数 f 在此区间上有符号变化，则连续函数 f 在这个区间内一定有零点。设初始的函数为 f ，满足 $f(a)$ 与 $f(b)$ 的符号不相同，最终区间长度的误差上线为 tol 。我们可以得到如下二分法的算法。

例 2.1.3 (Dichotomy). Using dichotomy method to solve function $f(x) = x^3 - 4\cos x = 0$.

```

1 function x = bisect(f,a,b,tol)
2   xlow = a; plow = f(xlow);
3   xhigh = b; phigh = f(xhigh);
4   while xhigh - xlow > 2*tol,
5     xmid = (xlow + xhigh)/2; pmid = f(xmid);
6     if pmid*plow < 0,
7       xhigh = xmid; phigh = pmid;
8     elseif pmid*phigh < 0,
9       xlow = xmid; plow = pmid;
10    else
11      xlow = xmid; xhigh = xmid;
12    end
13  end
14  x = [xlow, xhigh];
15 end

```

x_1	$f(x_1)$	x_2	$f(x_2)$
1	-1.1612	1.5	3.0921
1	-1.1612	1.25	0.6918
1.125	-0.3009	1.25	0.6918
1.125	-0.3009	1.1875	0.1786
1.15625	-0.0653	1.1875	0.1786
1.15625	-0.0653	1.171875	0.05562
1.1640625	-0.00509	1.171875	0.05562
1.1640625	-0.00509	1.16796875	0.025201
1.1640625	-0.00509	1.166015625	0.010037

不动点迭代

不动点迭代的思想为，将原问题变形为： $x = g(x)$ 的形式，再进行求解，我们可以构造格式 $x_{k+1} = g(x_k)$ 来对非线性方程进行求解。值得一提的是，这样的构造方式并不唯一，相对应的，不同的构造方式也有不同的稳定性以及收敛速度。

例 2.1.4. 非线性方程 $f(x) = x^2 - x - 2 = 0$ 的根为 $x^* = 2$ 和 $x^* = -1$ ，等价的不动点问题可以包括：

1. $g(x) = x^2 - 2$;

2. $g(x) = \sqrt{x+2}$;

3. $g(x) = 1 + 2/x$;

4. $g(x) = (x^2 + 2)/(2x - 1)$;

根据不同的迭代方程，都可求出相应的零点。

牛顿迭代

牛顿法的思想可以理解为：由于零点是其切线与 x 的交点，非零点则不是，于是在点 x_k 附近，使用 $f(x_k)$ 处的切线来近似，使用这个切线的零点作为新的近似值，以此来靠近真正的零点。其算法流程可表示为：

```

 $x_0$  = 初始值
for k = 0, 1, 2, ...
 $x_{k+1} = x_k - f(x_k)/f'(x_k)$ 
end

```

例2.1.5 (Newton Iteration). Using the method of Newton iteration to solve nonlinear equation $f(x) = x^2 - 4\cos x = 0$.

It is obvious that $f'(x) = 2x + 4\sin x$, so that the iteration scheme is

$$x_{n+1} = x_n - \frac{x_n^2 - 4\cos x}{2x + 4\sin x}$$

The calculation is performed in matlab

```

1 f = @(x) (x.^2-4*cos(x))./(2*x + 4*sin(x));
2 x0 = 3; temp = x0; err = 1; i = 1;
3 while err > 1e-5 && i <= 10
4     x(i) = temp - f(temp);
5     err = abs(f(temp));
6     temp = x(i); i = i+1;
7 end

```

k	x_k	$f(x_k)$
1	1.0257	0.18672
2	1.2125	-0.01089
3	1.2016	-3.3033×10^{-5}
4	1.2015	-3.0633×10^{-10}

The initial guess is $x_0 = 3.0$ and the iterative numerical solution are listed in the following table. It can be found that the convergence is fast as soon as x_n approaching the roots.

割线法

牛顿法有一个计算上的缺陷，即在每次迭代时都要计算函数及其导数的值，导数在计算中往往不方便或者计算量很大，因此在步长较小的情况下，我们可以用有限差分来近似代替导数，即用相邻两次迭代的函数值来代替其导数。这种方法叫割线法。

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

其算法流程可以表示为：

```

 $x_0$  = 初始值
for k = 0, 1, 2, ...
 $x_{k+1} = x_k - f(x_k)(x_k - x_{k-1})/[f(x_k) - f(x_{k-1})]$ 
end

```

例2.1.6 (Secant method). Using the Secant method to solve nonlinear equation $f(x) = x^2 - 4\cos x = 0$.

如何计算多项式的所有零点？这是一个被人们研究了很久的问题。对于某些特殊情形的 n 阶多项式 $p(x)$ ，有时要求求出它的所有 n 个零点，具体可以使用以下的思想来进行求值：用之前的方

k	x_k	$f(x_k)$
1	2	-5.6646
2	1.223538	-0.13576
3	1.204472	-0.018008
4	1.201556	-0.0001094
5	1.201538	-8.9895×10^{-8}

Table 2.1: Em:3.2.5

法，如牛顿法求出一个根 x_1 ，考虑低一阶的收缩多项式 $p(x)/(x-x_1)$ ，重复此过程，直到求出全部的根。形成给定多项式的友阵，利用之前讲过的计算特征值的方法计算特征值，即计算出多项式的根。一些专门计算多项式零点的方法，如Laguerre法、Traub法等。

2.1.3 Numerical methods for equation system

Considering numerical solution to a general nonlinear equation system

$$\mathbf{F}(\mathbf{x}) = \mathbf{0},$$

which is more competitive than the previous case. It lies in several aspects facts.

- 由于这类问题所涉及的范围更加广泛，所以对解的存在性和个数的分析也更加复杂一些。
- 利用传统的数值方法一般无法既绝对安全又收敛保证地产生有解区间。
- 随着问题维数的增加，计算量也将显著增加。

解决非线性方程组的方法也有很多，如不动点迭代、牛顿法、割线修正法、稳健性牛顿法等，这里主要介绍前三种方法。

Fixed point iteration

给定函数 $g: \mathbb{R}^n \rightarrow \mathbb{R}^n$ ，则不动点问题为：寻找 $\mathbf{x} \in \mathbb{R}^n$ ，使

$$\mathbf{x} = g(\mathbf{x}).$$

在一维情形，不动点迭代的收敛与否与收敛速度由 g 导数的绝对值来决定。而在高维情形下，我们有类似的谱半径条件：

$$\rho(\mathbf{G}(\mathbf{x}^*)) < 1,$$

若满足上述条件，当初始向量充分接近解时，不动点迭代收敛，谱半径越小，收敛速度越快。

Newton's method

Definition 2.1.3 (Jacobian's Matrix).

$$\{\mathbf{G}(\mathbf{x})_{ij} = \frac{\partial g_i(\mathbf{x})}{\partial x_j}\}.$$

Furthermore, $\mathbf{G}(\mathbf{x})$ means to evaluate G at position \mathbf{x} .

对可微函数 $\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^n$, 利用泰勒展开可得

$$\mathbf{f}(\mathbf{x} + \mathbf{s}) \approx \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\mathbf{s},$$

其中 $\mathbf{J}(\mathbf{x})\mathbf{s}$ 是雅可比矩阵, $\{\mathbf{J}(\mathbf{x})\}_{ij} = \partial f_i(\mathbf{x}) / \partial x_j$, 即可通过这个迭代过程来求出非线性方程组的零点, 其算法为:

$\mathbf{x}_0 =$ 初始值

for $k=0,1,2,\dots$

解 $\mathbf{J}(\mathbf{x}_k)\mathbf{s}_k = -\mathbf{x}_k$ 求 \mathbf{s}_k

$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$

end

每一步牛顿法都会解一个线性方程组, 因此其运算量是很大的。

定理 2.1.2 (local Convergence). \mathbf{F}'' exists at x^* .

定理 2.1.3 (Global Convergence). \mathbf{F}'' exists and be either non-positive or non-negative.

定理 2.1.4 (Semi-local Convergence). Han and Wang's theorem in 1998 paper.

Secant Method

与割线法类似, 它是将牛顿法中的雅可比矩阵中的偏导数用每个坐标方向上的有限差分近似代替, 从而可以节省不少计算成本, 其具体算法可表示为:

$\mathbf{x}_0 =$ 初始值

$\mathbf{B}_0 =$ 初始雅可比近似

for $k=0,1,2,\dots$

解 $\mathbf{B}_k\mathbf{s}_k = -\mathbf{x}_k$ 求 \mathbf{s}_k

$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$

$\mathbf{y}_k = \mathbf{f}(\mathbf{x}_{k+1}) - \mathbf{f}(\mathbf{x}_k)$

$\mathbf{B}_{k+1} = \mathbf{B}_k + [(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)\mathbf{S}_k^T] / (\mathbf{s}_k^T\mathbf{s}_k)$

end

这个方法即可加快迭代进程, 提升计算的效率。

例 2.1.7. Solving a nonlinear equation system,

$$F(x)\dots = \dots$$

via above numerical methods.

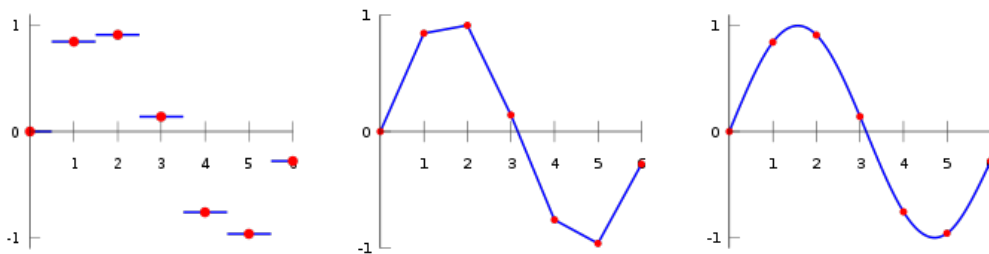
2.2 Interpolation and Approximation

插值问题的来源于各类实际应用, 如要求画出一条通过所有离散点的光滑曲线、利用列表函数求中间值、求列表函数的导数或积分、使用简单函数代替复杂函数, 从而可以快速方便的求出需要求得的值等等。当然, 这些问题又是来自于各类科学与工程计算问题的实际需求中抽象出来的。让用数学的语言进行定义插值问题:

Definition 2.2.1 (插值问题). 已知数据 $\{t_i, y_i\}_{i=0}^n$, 寻找函数 $f: \mathbb{R} \rightarrow \mathbb{R}$, 满足:

$$f(t_i) = y_i, \quad \forall i = 0, 1, \dots, n$$

其中 f 称为插值函数, 简称为插值。称 $\{t_i\}_{i=0}^n$ 为插值节点。



如下几个例子描述了如何找到一些“简单”函数来“穿过”给定的数据点：插值函数的选择主要考虑因素是多方面的。首先，函数的简单程度是考虑的重要因素，因为越简单的函数越有利于计算机进行求值等各种运算，如多项式是计算机最擅长的数学工具。其次，也要考虑与拟合数据在性态方面的接近程度（如：光滑性，单调性，周期性等），比如分段多项式、三角函数、指数函数和有理函数等也是数值计算中常用的数学工具。

对于给定的数据点集 $(t_i, y_i), i = 1, 2, \dots, n$, 选择基函数 $\phi_1(t), \phi_2(t), \dots, \phi_n(t)$, 在这组基函数所张成的函数空间中选择一个插值函数。将插值函数 f 写成这些基函数的线性组合：

$$f(t) = \sum_{j=1}^n x_j \phi_j(t),$$

其中 x_j 是待定的参数，则数据点 (t_i, y_i) 上的插值函数 f 应满足：

$$f(t_i) = \sum_{j=1}^n x_j \phi_j(t_i) = y_i,$$

将其写成 $\mathbf{Ax} = \mathbf{y}$ 的矩阵形式， \mathbf{A} 的元素为 $a_{ij} = \phi_j(t_i)$ 。

利用上述形式，结合线性代数的知识我们有如下结论：

2.2.1 Interpolation with polynomials

最简单直接的基底是单项式基底，即对于 n 个数据点，进行选取 $k=n-1$ ，则有单项式基底：

$$\phi_j(t) = t^{j-1}, \quad j = 1, 2, \dots, n,$$

任何 $n-1$ 次多项式，皆可用这个基底的线性组合来表示。此时，插值多项式的参数可由下述方程组求解：

$$\mathbf{Ax} = \begin{bmatrix} 1 & t_1 & \dots & t_1^{n-1} \\ 1 & t_2 & \dots & t_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_n & \dots & t_n^{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \mathbf{y}$$

这里， \mathbf{A} 是一个范德蒙矩阵，当 $x_0 < x_1 < \dots < x_n$ 时非奇异。单项式基底具有存在及唯一性，因此解此方程组需要的工作量是 $\mathcal{O}(n^3)$ ，计算成本高！

Proposition 2.2.1 (存在唯一性). 若基函数个数 n 与数据个数 m 相等，则得到的是一个方阵线性方程组。若矩阵 \mathbf{A} 非奇异，则一定有且仅有唯一解。而若矩阵 \mathbf{A} 奇异，则可以有无数参数的解，代表着数据点不能被精确拟合。

Proposition 2.2.2 (病态性). 基函数可以有許多选择的方式, 相对应的会有許多矩阵 \mathbf{A} 的表达式。 \mathbf{A} 若是单位阵, 下三角矩阵, 三对角矩阵等等特殊的矩阵, 会大大提升求解参数的效率, 降低求解的难度, 在之后的具体例子里有所体现。

拉格朗日插值公式

对给定的数据点 $(t_i, y_i), i = 1, 2, \dots, n$, $n-1$ 次拉格朗日基函数可以表示为:

$$l_j(t) = \frac{\prod_{k=1, k \neq j}^n (t - t_k)}{\prod_{k=1, k \neq j}^n (t_j - t_k)}, j = 1, 2, \dots, n,$$

显然, 我们有结论:

$$l_j(t) = \begin{cases} 1, & i = j, \\ 0, & i \neq j \end{cases}, i, j = 1, 2, \dots, n,$$

说明对于拉格朗日插值来说, \mathbf{A} 是单位矩阵 \mathbf{I} , 因此参数 x_1, x_2, \dots, x_n 可以直接由 y_i 得到。拉格朗日插值求参数是很容易的, 但是同单项式基底表达式相比, 它基函数形式更加复杂, 并且积分与微分的操作会困难许多。

牛顿插值公式

对给定的数据点 $(t_i, y_i), i = 1, 2, \dots, n$, $n-1$ 次牛顿基函数为:

$$\pi_j(t) = \prod_{k=1}^{j-1} (t - t_k), \quad \forall j = 1, 2, \dots, n,$$

注意到 k 始终比 j 小1, 也就是说, 多项式可以表示为:

$$Q_n(x) = x_1 + x_2(t - t_1) + x_3(t - t_1)(t - t_2) + \dots + x_n(t - t_1)(t - t_2) \cdots (t - t_{n-1}). \quad (2.2)$$

这里, \mathbf{A} 是下三角矩阵, 解 $\mathbf{Ax} = \mathbf{y}$ 复杂度 $O(n^2)$ 同前两种方法相比, 牛顿插值既可以节省计算量, 又可以节约计算给定点的值的成本。牛顿法的另一大优势在于, 它可以逐步进行计算。如果我们需增加插值点的个数的时候, 可以直接计算增加的点即可, 这是其他两种方法做不到的。让我们考虑加入点 (t_{n+1}, y_{n+1}) , 利用牛顿法可得新的插值多项式:

$$Q_{n+1}(t) = Q_n(t) + x_{n+1}\pi_{n+1}(t)$$

其中 π 的定义与之前的基函数相同, 且

$$x_{n+1} = \frac{y_{n+1} - Q_n(t_{n+1})}{\pi_{n+1}(t_{n+1})}.$$

而之前两种插值方式只能将所有点再重新计算, 再继续求解参数。

这里我们给出一例以体现三种不同插值公式的区别:

例2.2.1 (Base function with three formulas of interpolation). Plot the base function of Lagrange, monominal and Newton interpolation and point out the difference between them. Try to compute the corresponding secondary Interpolation polynomials with datas $(-1, 1), (0, 0), (1, 1)$.

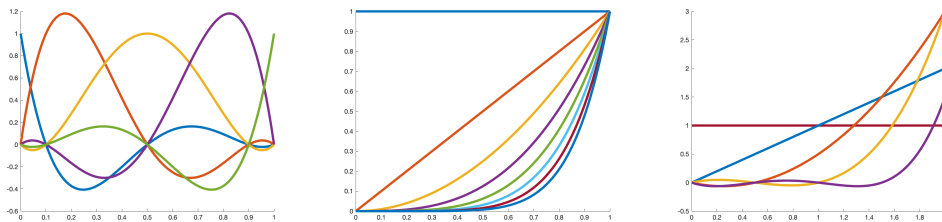


Figure 2.1: (Left) Lagrange base, (Middle) monominal base, (Right) Newton base

差商

差商的定义: $f[x_0, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}$

差商的性质:

- $f[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!}$, 其中 $\min\{x_i\} < \xi < \max\{x_i\}$;
- 交换不变
- $\frac{\partial}{\partial x} f[x_0, \dots, x_n, x] = f[x_0, x_1, \dots, x_n, x, x]$

则可以得到插值多项式:

$$f(x) = P_n(x) + \omega_{n+1}(x)f[x_0, x_1, \dots, x_n, x]$$

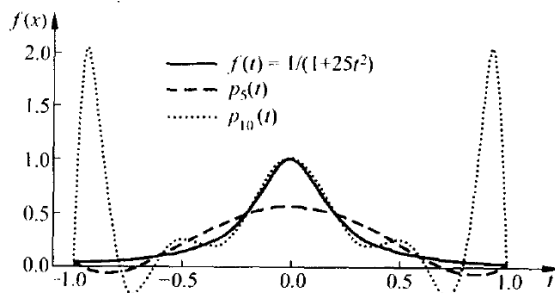
并且此时的误差为:

$$E_{n+1} = f(x) - P_n(x) = \omega_{n+1}(x)f[x_0, x_1, \dots, x_n, x].$$

其中可以计算出二阶的误差为: $E_2 = -\frac{h^2}{8}f''(\xi)$

2.2.2 Interpolation with piecewise polynomials

基函数和数据点的合适选择可以减轻高次多项式插值的困难, 但有时候用一个多项式来拟合大量数据点, 会出现令人不悦的振荡现象。



分片思想

分段多项式主要思想是：对给定的数据点集 $(t_i, y_i), i = 1, \dots, n, t_1 < t_2 < \dots < t_n$, 作分段多项式插值时，在每个子区间 $[t_i, t_{i+1}]$ 上用不同的多项式。典型的分段插值方法包括三次样条插值和三次Hermit插值，如下图所示：

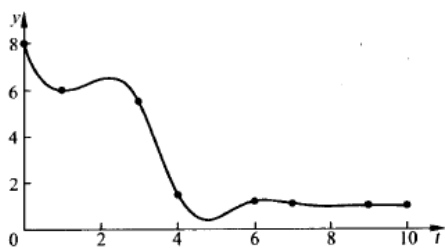


图 7.10 单调数据的三次样条插值

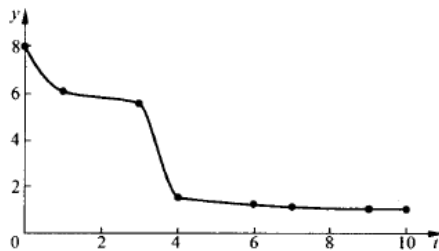


图 7.9 单调数据的三次埃尔米特插值

B-样条

采用B样条作为插值基函数有很多优点。

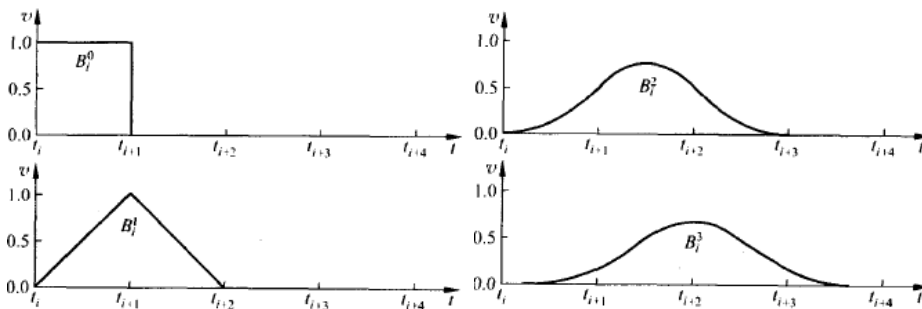
作为递推初始值, 定义 0 次 B-样条

$$B_i^0(t) = \begin{cases} 1, & t_i \leq t < t_{i+1}, \\ 0, & \text{其他.} \end{cases}$$

对 $k > 0$, 定义 k 次 B-样条为

$$B_i^k(t) = v_i^k(t) B_{i-1}^{k-1}(t) + (1 - v_{i+1}^k(t)) B_{i+1}^{k-1}(t).$$

由于 B_i^0 是分段常数, v_i^k 是线性的, 从定义可知, B_i^1 是分段线性的. 类似地, B_i^2 是分段二次



关于回归、插值、逼近、拟合的区别：插值：插值曲线要经过型值点；逼近：为复杂函数寻找近似函数，其误差在某种度量意义下最小；拟合：在插值问题中考虑给定数据点的误差，其误差在某种度量意义下最小；回归：最小二乘解，在 $\|\cdot\|_2$ 度量意义下最小。

例2.2.2 (Multi-variable approximation: triangulation mesh). The 2 dimensional area can be represented by a series of triangulations. The errors between them can approach to zero while the radius of the triangulation is close to zero.

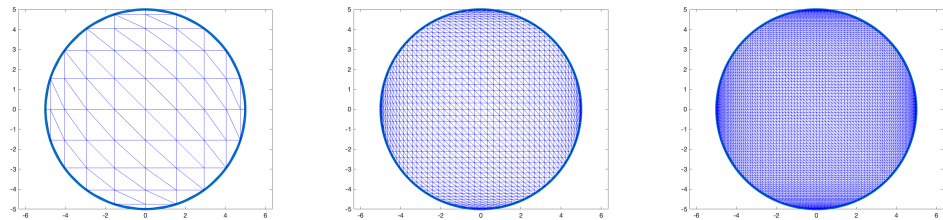
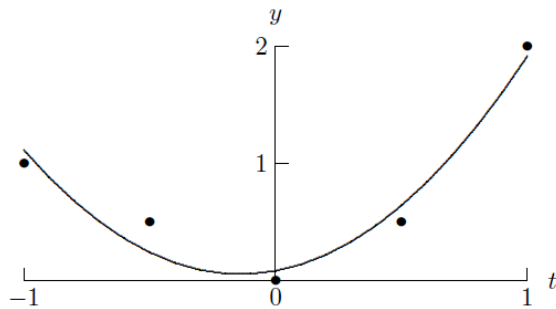


Figure 2.2: Different radius of triangulation meshes

2.2.3 Least square approximation

例2.2.3 (数据拟合). 找一条二次曲线逼近如下观测点： 类似问题：多元线性回归、非多项式数据

t	-1.0	-0.5	0.0	0.5	1.0
y	1.0	0.5	0.0	0.5	2.0



拟合等

2.2.4 Best approximation

Chapter 3

Numerical Approximation in Calculus

3.1 数值求积问题

3.1.1 积分问题

积分的应用十分广泛，在各个领域都有一些应用，主要在以下几个方面：积分变换，如拉普拉斯变换，傅里叶变换、偏微分方程的有限元法和边界元法、积分方程和变分法、概率统计，其中总的许多基本概念，如概率分布、期望等由积分定义。此外，在古典和量子物理中的系统能量定义。

Definition 3.1.1 (积分). 对区间 $[a, b]$ 上的函数 $f: \mathbb{R} \rightarrow \mathbb{R}$, 现代积分定义

$$\mathbf{I}(f) = \int_a^b f(x) dx$$

是以形如下式的黎曼和为基础的

$$R_n = \sum_{i=1}^n (x_{i+1} - x_i) f(\xi_i),$$

其中 $a = x_1 < x_2 < \cdots < x_n < x_{n+1} = b$, $\xi \in [x_i, x_{i+1}]$.

记 $h_n = \max_{i=1, \dots, n-1} \{x_{i+1} - x_i\}$, 若

$$\lim_{h_n \rightarrow 0} R_n = R \neq \infty,$$

则称 f 在区间上黎曼可积。

积分（模型/方法）的正则性（存在/唯一/条件数）是一个很重要的理论问题。

存在性: f 有界、连续 \rightarrow Riemann 可积。唯一性: 极限存在性与剖分无关、与 ξ_i 取法无关。

问题的刚性: 数据扰动/误差传播分析。我们通过如下两种途径来分析误差的扰动:

杆的分散载荷问题, 图 8.1 中, 若 $\rho(x)$ 为载荷密度关于横坐标 x 的函数, 则杆受到的总载荷 W 为曲线 $\rho(x)$ 所围区域的面积, 由积分

$$W = \int_0^L \rho(x) dx$$

给出. 而且, 等价的集中载荷的作用点为曲线所围区域的重心, 其横坐标 \bar{x} 由积分

$$\bar{x} = \frac{1}{W} \int_0^L x \rho(x) dx$$

给出.

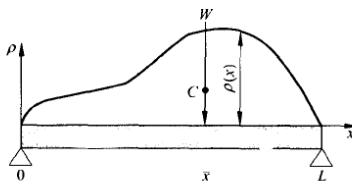


图 8.1 木杆上的分散载荷与集中载荷等价

很久以前就有人提出了近似计算不规则区域面积的方法, 其中包括阿基米德 (Archimedes), 他的方法是将区域用小正方形铺贴, 然后计算区域内小方块的个数.

被积函数的扰动

$$|I(\hat{f}) - I(f)| \leq \int_a^b |\hat{f}(x) - f(x)| dx \leq (b-a) \|\hat{f} - f\|_\infty$$

积分区间的扰动

$$\left| \int_a^{\hat{b}} f(x) dx - \int_a^b f(x) dx \right| \leq \int_b^{\hat{b}} f(x) dx \leq (\hat{b} - b) \|f\|_\infty$$

3.1.2 数值积分公式

定积分的数值求解称为数值求积。定积分：

$$I(f) = \int_a^b f(x) dx,$$

数值求积的基本想法是,

Definition 3.1.2 (数值积分). 寻找一个 $I(f)$ 的近似:

$$Q_n(f) = \sum_{i=1}^n \omega_i f(x_i),$$

其中 $a < x_1 < \dots < x_n < b$, 称为节点, ω 称为权。数值求积的目标则是选择节点和权, 利用合理的计算成本来求得达到满足要求的精度的近似 $Q_n(f)$ 。

3.1.3 插值型公式(Newton-Cotes系列)

求积公式的节点选择 $[a, b]$ 上等距的节点, 即:

$$x_i = a + i(b-a)/(n+1), i = 1, 2, \dots, n.$$

求积公式可以由多项式插值得到. 事实上, 利用被积函数 f 在点 $x_i (i=1, \dots, n)$ 上的值, 可以确定被积函数在这 n 个点上的 $n-1$ 次插值多项式(见第 7 章), 然后用插值多项式的积分作为原始函数积分的近似. 事实上, 具体计算积分时, 并不需要直接得到插值多项式. 相反, 可以用插值多项式确定求积公式的节点和权, 由节点和权可以计算积分区间上任何函数积分的近似值. 特别地, 若使用拉格朗日插值, 则对给定点集 $x_i (i=1, \dots, n)$, 可用其相应的拉格朗日基函数的积分确定权,

$$w_i = \int_a^b L_i(x) dx, \quad i = 1, \dots, n.$$

可以看出, 对任何被积函数这些权都是相同的. 因此, 这种求积公式称为插值型的.

梯形(Trapezoid)公式

取两个节点, 取权值为两点函数值的平均值, 则可以得到:

$$Q_n(f) = (b-a) \frac{f(a) + f(b)}{2}$$

误差:

$$E_2 = \left| \int_a^b f(x) dx - Q_n(f) \right| = \int_a^b \frac{(b-1)^2}{8} f''(\xi) dx = \frac{(b-a)^3}{8} \|f''\|_\infty$$

复化梯形公式($n+1$ 个节点):

$$T_n = h \sum_{i=0}^{n-1} \frac{f(x_i) + f(x_{i+1})}{2} = \frac{b-a}{n} \left[\frac{1}{2} f(x_0) + f(x_1) + \dots + f(x_{n-1}) + \frac{1}{2} f(x_n) \right]$$

此时的误差为: $E_{rr}^T = -\frac{(b-a)\|f''\|_\infty}{12} h^2$

待定系数法

得到插值型求积公式的另一种方法是待定系数法, 它是通过使积分公式对前 n 个多项式基函数精确成立, 产生一个 n 元方程组, 从而得到权. 例如, 若使用单项式基底, 得到的是力矩方程组

$$\begin{aligned} w_1 \cdot 1 + w_2 \cdot 2 + \dots + w_n \cdot 1 &= \int_a^b 1 dx = b-a, \\ w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n &= \int_a^b x dx = (b^2 - a^2)/2, \\ &\vdots \\ w_1 \cdot x_1^{n-1} + w_2 \cdot x_2^{n-1} + \dots + w_n \cdot x_n^{n-1} &= \int_a^b x^{n-1} dx = (b^n - a^n)/n, \end{aligned}$$

例: 在区间 $[a, b]$ 上构造三节点积分公式:

$$Q_3(f) = w_1 f(x_1) + w_2 f(x_2) + w_3 f(x_3),$$

取三个节点分别为积分区间的两个端点及中点, 即 $x_1 = a, x_2 = (a+b)/2, x_3 = b$. 写成矩阵形式, 力矩方程为范德蒙方程组

$$\begin{bmatrix} 1 & 1 & 1 \\ a & (a+b)/2 & b \\ a^2 & ((a+b)/2)^2 & b^2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} b-a \\ (b^2-a^2)/2 \\ (b^3-a^3)/3 \end{bmatrix}.$$

用高斯消去法求解, 得到权

$$w_1 = (b-a)/6, \quad w_2 = 2(b-a)/3, \quad w_3 = (b-a)/6.$$

辛普生(Simpson)公式

$$S(f) = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]$$

误差:

$$E_3 = \left| \int_a^b f(x) dx - S(f) \right| = -\frac{(b-a)^5}{2880} f^{(4)}(\eta), a < \eta < b$$

复化Simpson公式($n+1$ 个节点):

$$S_n = \frac{h}{6} \sum_{i=0}^{n-1} \left[f(x_i) + 4f(x_{i+\frac{1}{2}}) + f(x_{i+1}) \right].$$

此时的误差为: $E_{rr}^S = -\frac{(b-a)\|f^{(4)}\|_{\infty}}{2880} h^4$.

例3.1.1 (numerical quadrature: Newton-Cotes). Using different formulas in Newton-Cotes to compute $I(f) = \int_0^1 e^{-x^2} dx$. (The exact solution is 0.746824)

$$M(f) = (1-0)e^{-0.25} = 0.778801$$

$$T(f) = \frac{1}{2}(e^0 + e^{-1}) = 0.683940$$

$$T(f) = \frac{1}{6}(e^0 + 4e^{-0.25} + e^{-1}) = 0.747180$$

3.1.4 提高积分精度的方法

复化同理于分段插值

自适应方法- 区间分半递归

Richardson外推- Romberg积分法 (基于Newton-Cotes系列的外推)

其他典型积分相关问题. 基于表格化数据求积分- 先插值, 再积分. 奇异积分- 取极限、变量代换. 二/三重积分- 张量积、单纯型高维积分- Monte Carlo方法. 积分方程- 通常是 ill-conditioned.

3.1.5 最佳求积

固定节点的数值积分公式精度有限, 并没有充分利用已有的“信息”。

Clenshaw-Curtis求积

一些研究发现：用Chebyshev多项式零点作为积分节点的时候可以证明：积分权重总是正数！其次，有效的自包含求法：不用事先求得权重。可以利用FFT和递推算法，代数精度不是最优阶！

高斯求积

基本思想:选择最佳的节点和权值，使其代数精度最大化。

以构造 $[-1,1]$ 上的两点公式 $G_2(f)$ 为例：

$$I(f) = \int_{-1}^1 f(x)dx \approx \omega_1 f(x_1) + \omega_2 f(x_2) := G_2(f),$$

令其对前四阶单项式精确成立：

$$\begin{cases} \omega_1 + \omega_2 = \int_{-1}^1 1dx = 2, \\ \omega_1 x_1 + \omega_2 x_2 = \int_{-1}^1 xdx = 0, \\ \omega_1 x_1^2 + \omega_2 x_2^2 = \int_{-1}^1 x^2 dx = \frac{2}{3}, \\ \omega_1 x_1^3 + \omega_2 x_2^3 = \int_{-1}^1 x^3 dx = \frac{2}{3}, \end{cases}$$

可解得

$$x_1 = -1/\sqrt{3}, x_2 = 1/\sqrt{3}, \omega_1 = 1, \omega_2 = 1$$

即得到了三阶两点高斯求积公式为：

$$G_2(f) = f(-1/\sqrt{3}) + f(1/\sqrt{3}).$$

使用高斯求积公式也需要注意如下几点：如何通过Legendre多项式等典型的正交多项式零点计算；只在标准区间上给出，实际使用需放射变换；Gauss-Kronrod递推型高斯求积。

3.1.6 突破维数诅咒： Monte Carlo求积方法**3.2 Ordinary Differential Equations for Initial Value Problems(IVPs)****3.2.1 Initial Value Problem**

Considering first-order Ordinary Differential Equations(ODEs)

$$\mathbf{y}' = \mathbf{y}'(t) := \begin{bmatrix} y_1'(t) \\ y_2'(t) \\ \vdots \\ y_n'(t) \end{bmatrix} = \begin{bmatrix} f_1(t, \mathbf{y}) \\ f_2(t, \mathbf{y}) \\ \dots \\ f_n(t, \mathbf{y}) \end{bmatrix} := \mathbf{f}(t, \mathbf{y}), \quad (3.1)$$

where $\mathbf{y} := (y_1, y_2, \dots, y_n)^T$ and $\mathbf{f} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$. When $n > 1$, it is a system of coupled ODEs.

例3.2.1. An given n -th order ODE

$$y^{(n)} = f(t, y, y', y'', \dots, y^{(n-1)})$$

can be transformed into a first-order system

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

The above ODEs (3.1) will admit a unique solution as soon as an initial condition

$$\mathbf{y}(t_0) = \mathbf{y}_0 \quad (3.2)$$

is given. In the literature, people referred (3.1)+(3.2) as the Initial Value Problems (IVPs) for ODEs. By integrating the ODE, its solution is given by the integral

$$\mathbf{y}(t) = \mathbf{y}_0 + \int_{t_0}^t \mathbf{f}(s) ds \quad (3.3)$$

定理 3.2.1 (Existence and Uniqueness of the solution to IVPs). If $\mathbf{f} : \mathbf{R}^{n+1} \rightarrow \mathbf{R}^n$ is Lipschitz continuous on D , which means

$$\|\mathbf{f}(t, \hat{\mathbf{y}}) - \mathbf{f}(t, \mathbf{y})\| \leq L \|\hat{\mathbf{y}} - \mathbf{y}\|,$$

then the solution to IVPs (3.1)+(3.2) has unique solution for any given \mathbf{y}_0 defined on D .

例 3.2.2 (Newton's Second Law of Motion). As well known as the formula $F = ma$, force equals mass times acceleration. It is straightforward that $a = s''(t)$, where t means the time and $s(t)$ is the displacement at time t . Considering the trajectory of falling object under the force of Earth's gravity, which means $F = -mg$. When the initial position $s(0)$ and initial velocity $v(0) = s'(0)$ are given, we have its solution

$$s(t) = -\frac{1}{2}gt^2 + s'(0)t + s(0).$$

Considering the linear and homogeneous system of ODEs in the form of

$$\mathbf{y}' = A\mathbf{y},$$

where A is an $n \times n$ constant coefficient matrix, which is assumed to be diagonalizable. The initial condition $\mathbf{y}(t_0) = \mathbf{y}_0$ is given, then it is easy to confirm that

$$\mathbf{y}(t) = \sum_{i=1}^n \alpha_i \mathbf{v}_i e^{\lambda_i t},$$

where $\{\lambda_i\}_{i=1}^n$ and $\{\mathbf{v}_i\}_{i=1}^n$ are eigenvalues and corresponding eigenvectors of A , and $\mathbf{y}_0 = \sum_{i=1}^n \alpha_i \mathbf{v}_i$.

Proposition 3.2.1 (Stability of Solutions).

3.2.2 Numerical Methods for ODEs

Finite Difference Approximations

It is essential to explain the approximation of the first order and the second order derivatives.

1. 向前 (向后) 一阶差商

$$f'(x) \approx D_h^+ := \frac{f(x+h) - f(x)}{h}$$

2. 中心一阶差商

$$f'(x) \approx D_{2h} := \frac{f(x+h) - f(x-h)}{2h}$$

3. 单边差商

$$f'(x) \approx D_{2h}^+ := \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h}$$

请用Taylor公式验证上述格式的精度分别为 $o(h)$ 以及 $o(h^2)$.

中心二阶差商

$$f''(x) = (f(x+h) - 2f(x) + f(x-h))/h^2$$

假设 $D_h^2 f(x) = Af(x+h) + Bf(x) + Cf(x-h)$,则由Taylor展开:

$$\begin{cases} A + B + C = 0; \\ h(A - C) = 0; \\ \frac{h^2}{2}(A + C) = 1. \end{cases}$$

此时的误差为 $-\frac{h^2}{12}f^{(4)}(x)$ (How to prove?)

Euler's Method

考虑一维一阶常微分初值问题:

$$\frac{dy}{dx} = f(x, y), x \in [a, b]$$

$$y(a) = y_0,$$

基本思想: 从初值出发, 每一步都沿切线方向积分一步:

$$y(x_k) = y(x_{k-1}) + (x_k - x_{k-1})y'(x_{k-1}) = y(x_{k-1}) + h_k f(x_{k-1}, y)$$

其中, 第 k 步的误差, 即截断误差可以定义为:

$$R_k = \int_{x_{k-1}}^{x_k} f(s, y(s))ds - h_k f(x_{k-1}, y(x_{k-1})).$$

这里, 记 $h_k := (x_k - x_{k-1})$ 是步长.

例3.2.3 (Euler). Considering the following initial problem:

$$y' = y, y(0) = 1.$$

整体误差和局部误差 整体误差不是简单的局部误差的和, 注意: 发散情形: 局部误差会被放大, 其和大于整体误差; 收敛情形: 整体误差小于局部误差的和.

隐式欧拉

$$y(x_k) = y(x_{k-1}) + (x_k - x_{k-1})y'(x_k)$$

无条件稳定(对 h 无限制), 精度只有一阶:

$$\left| \frac{1}{1 - h\lambda} \right| = 1 + h\lambda + (h\lambda)^2 + \dots < 1$$

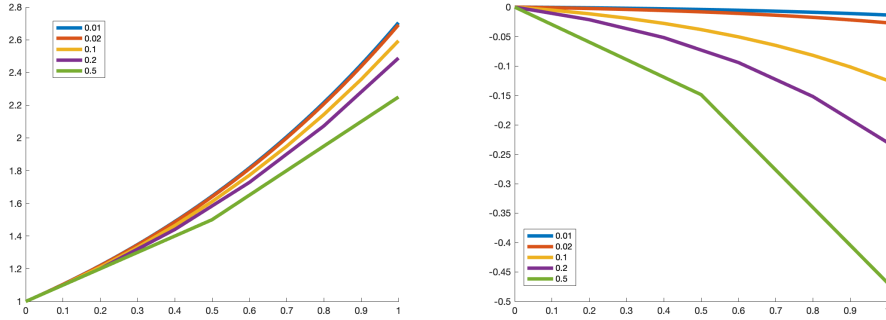


Figure 3.1: (Left) The Euler solution, (Right) Errors

梯形/中点法

$$y(x_k) = y(x_{k-1}) + (x_k - x_{k-1})(y'(x_{k-1}) + y'(x_k))/2$$

(部分情形)无条件稳定，二阶收敛(与 $e^{h\lambda}$ 的展开式比较)：

$$\left| \frac{1 + h\lambda/2}{1 - h\lambda/2} \right| = 1 + h\lambda + \frac{(h\lambda)^2}{2} + \frac{(h\lambda)^3}{4} + \dots < 1$$

Definition 3.2.1 (相容性). 如果单步法的增量函数 φ 满足：

$$\varphi(x, y, 0) = f(x, y)$$

则称单步法与微分方程是相容的。

Definition 3.2.2 (稳定性). 如果 φ 对于任意 $(x, y) \in \Omega$ 以及小于某个步长 h 时关于 y 满足Lipschitz条件，则单步方法是稳定的。

Definition 3.2.3 (收敛性). 由增量函数 $\varphi(x, y, h)$ 所确定的单步法是收敛的，若

$$\lim_{h \rightarrow 0} y_m = y(x), x = x_m$$

若 φ 关于 x 、 h 满足Lipschitz条件，则收敛性与相容性等价

Runge-Kutta方法

记 $a_{i,j}, c_i (i = 2, 3, \dots, s; j < i)$ 和 $b_i (i = 1, 2, \dots, s)$ 是一些待定的实数权值。则 s 级显Runge-Kutta方法：

$$y_{m+1} = y_m + h(b_1 k_1 + \dots + b_s k_s),$$

其中，

$$\begin{aligned} k_1 &= f(x_m, y_m), \\ k_2 &= f(x_m + c_2 h, y_m + h a_{2,1} k_1), \\ k_3 &= f(x_m + c_3 h, y_m + h(a_{3,1} k_1 + a_{3,2} k_2)), \\ &\dots \dots \dots \\ k_s &= f(x_m + c_s h, y_m + h(a_{s,1} k_1 + \dots + a_{s,s-1} k_{s-1})) \end{aligned}$$

可以按Taylor公式确定系数。由于定解条件少于参数个数，因此这里公式中的参数的值不唯一，常用的有Heun公式、Gill公式等。此外，我们也既避免计算y的高阶导数，也保证差分的高精度。

The most famous numerical methods for ordinary differential equation is the forth order Runge-Kutta formula, which is

$$y_{k+1} = y_k + \frac{h_k}{6}(k_1 + 2k_2 + 2k_3 + k_4), \quad (3.4)$$

with the definition of momentum

$$\begin{aligned} k_1 &= f(t_k, y_k), \\ k_2 &= f(t_k + \frac{1}{2}h_k, y_k + \frac{1}{2}h_k k_1), \\ k_3 &= f(t_k + \frac{1}{2}h_k, y_k + \frac{1}{2}h_k k_2), \\ k_4 &= f(t_k + h_k, y_k + h_k k_3), \end{aligned}$$

例3.2.4 (explicit Runge-Kutta method(example 9.12)). Solve the following ODE numerical via Runge-Kutta method:

$$y' = -2ty^2, y(0) = 1$$

Starting from $t_0 = 0$ to $t_1 = 0.25$ with time-step length $h = 0.25$, then we have $k_1 = f(t_0, y_0) = 0, k_2 = f(t_0 + h, y_0 + hk) = -0.5$

It is not difficult to know that the analytic solution for the current problem is $y(t) = \frac{1}{1+t^2}$, so that one can evaluate at the specific time, for example $y(0.25) = 0.9412, y(0.5) = 0.8$, for the purpose of calculating numerical errors. We plot the numerical solution as well as it's error in Fig.3.2.4

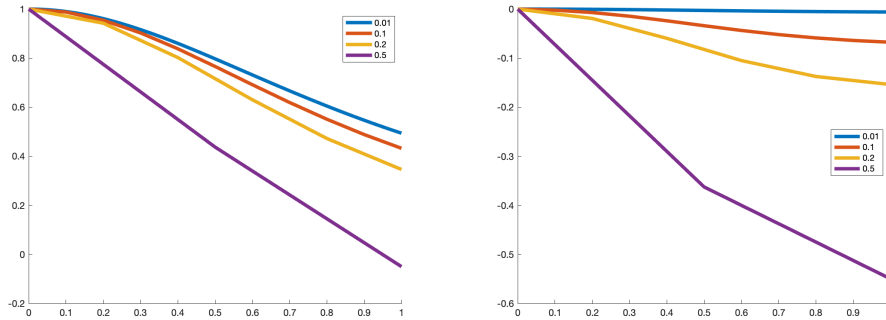


Figure 3.2: (Left)The Runge-Kutta solution with Heun formula, (Right) Errors

3.2.3 Stability Improvements

implicit scheme

Let us present a numerical example as the exercise

例3.2.5 (implicit Runge-Kutta method). Using Hammer method to solve the following ODE:

$$y' = -2xy^2, y(0) = 1$$

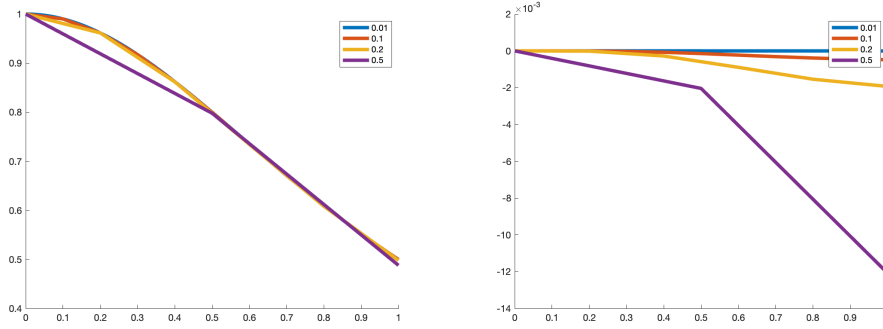


Figure 3.3: (Left)The Runge-Kutta solution with Hammer formula, (Right) Errors

如果 k_i 的计算中用到了所有的 k 值, 即:

$$y_{m+1} = y_m + h(b_1 k_1 + \cdots + b_s k_s),$$

其中 k_i 满足:

$$k_i = f(x_m + c_i h, y_m + h(a_{i,1} k_1 + \cdots + a_{i,s} k_s))$$

常用的隐式方法有: 隐式中点公式, Hammer 和 Hollingsworth 公式, Kuntzmann 和 Butcher 公式等。

线性多步法

It is worth to mention that 形如的方法都属于单步法:

$$y_{m+1} = y_m + h\varphi(x_m, y_m, h),$$

其中 φ 被称为单步法的增量函数。单步法的截断误差可表示为

$$R_m = y_{m+1} - y_m - h\varphi$$

As the improvements, the multi-step schemes yield larger stable region as well as better precision. Considering problem $y' = f(x, y)$, where $y_m = y(x_m)$, $f_m = f(x_m, y_m)$. We have

$$y_{m+1} = \sum_{i=1}^k \alpha_i y_{m-i+1} + h\Phi(x_{m+1}, x_m, \dots, x_{m-k+1}, y'_{m+1}, \dots, y'_{m-k+1}; h),$$

where $k \in \mathbb{N}^+$, $\{\alpha_i\}$ is given real number and h is the time-step length. Then the truncation error is

$$R(x_m, y_m, h) = y_{m+1} - \sum_{i=1}^k \alpha_i y_{m-i+1} - h\Phi$$

p阶：使误差等于 $O(h^{p+1})$ 成立的最大整数！

多步法的典型构造方法是多样的。Adams外插法是一种显式方法。相应的有Adams内插法是一种隐式方法。待定系数法求p阶m步线性多步法

例3.2.6 (Linear multi-step method). Using linear multi-step method to solve the following ODE:

$$y' = -2xy^2, y(0) = 1$$

Stiffness

ODE的Jacobi矩阵特征值相互差别极大，则称它是刚性的！

软件包与参考材料

Exercise

例 9.13 多步法的推导 为说明多步法的其他推导方法,我们导出形如

$$y_{k+1} = \alpha_1 y_k + h(\beta_1 y'_k + \beta_2 y'_{k-1})$$

的显式两步法,其中 α_1 、 β_1 和 β_2 是待定参数. 为使符号简化,只考虑一维情形,结果可推广到方程组的向量情形. 用待定系数法,令公式对前三个单项式精确成立. 若 $y(t) = 1$, 则 $y'(t) = 0$, 得到方程

$$1 = \alpha_1 \cdot 1 + h(\beta_1 \cdot 0 + \beta_2 \cdot 0).$$

若 $y(t) = t$, 则 $y'(t) = 1$, 得到方程

$$t_{k+1} = \alpha_1 t_k + h(\beta_1 \cdot 1 + \beta_2 \cdot 1).$$

若 $y(t) = t^2$, 则 $y'(t) = 2t$, 得到方程

$$t_{k+1}^2 = \alpha_1 t_k^2 + h(\beta_1 \cdot 2t_k + \beta_2 \cdot 2t_{k-1}).$$

由于这三个方程对任何 t_k 都成立,不妨取 $t_{k-1} = 0, h = 1$ (因而 $t_k = 1, t_{k+1} = 2$), 求解相应的 3×3 线性方程组,得 $\alpha_1 = 1, \beta_1 = \frac{3}{2}, \beta_2 = -\frac{1}{2}$. 这样,两步显式法为

$$y_{k+1} = y_k + \frac{h}{2}(3y'_k - y'_{k-1}),$$

由约束条件,它是二阶的. 类似地,还可以推导形如

$$y_{k+1} = \alpha_1 y_k + h(\beta_0 y'_{k+1} + \beta_1 y'_k)$$

的隐式两步法. 同样,用待定系数法,令公式对前三个单项式精确成立,得到三个方程

$$\begin{cases} 1 = \alpha_1 \cdot 1 + h(\beta_0 \cdot 0 + \beta_1 \cdot 0), \\ t_{k+1} = \alpha_1 \cdot t_k + h(\beta_0 \cdot 1 + \beta_1 \cdot 1), \\ t_{k+1}^2 = \alpha_1 \cdot t_k^2 + h(\beta_0 \cdot 2t_{k+1} + \beta_1 \cdot 2t_k). \end{cases}$$

为简化,取 $t_k = 0, h = 1$ (从而 $t_{k+1} = 1$), 求解相应的 3×3 线性方程组,得 $\alpha_1 = 1, \beta_1 = \frac{1}{2}, \beta_0 = \frac{1}{2}$. 这样,隐式两步法为

$$y_{k+1} = y_k + \frac{h}{2}(y'_{k+1} + y'_k).$$

可以看出,这就是梯形方法,由约束条件知道,它是二阶的. 用同样的方法,还可以得到更高阶的方法,只需令公式对若干个单项式精确成立,单项式的个数要与公式中待定参数的个数相同,然后求解相应的线性方程组,得到参数的值即可. ■

例 9.14 预估-校正方法 为说明预估-校正方法,我们用例 9.13 中得到的两步法来解例 9.11 和例 9.12 中的一维非线性常微分方程

$$y' = -2ty^2,$$

初值为 $y(0) = 1$. 由于二阶显式格式需要两个初值,所以除了 $t_0 = 0$ 时的初值 $y_0 = 1$ 外,还要使用例 9.12 中用单步 Heun 法得到的 $t_1 = 0.25$ 时的值 $y_1 = 0.9375$. 从 $t_1 = 0.25$ 到 $t_2 = 0.5$,取步长 $h = 0.25$,使用二阶显式方法,得到预估值

$$\hat{y}_2 = y_1 + \frac{h}{2}(3y'_1 - y'_0) = 0.9375 + 0.125(-1.3184 + 0) = 0.7727.$$

计算 f 在预估值 \hat{y}_2 处的值,得到相应的导数值 $\hat{y}'_2 = -0.5971$. 在相应的隐式方法中(在这里为梯形公式)将这些预估值代入,得到解的校正值

$$y_2 = y_1 + \frac{h}{2}(y'_2 + y'_1) = 0.9375 + 0.125(-0.5971 - 0.4395) = 0.8079.$$

用这个新的 y_2 再计算 f 的值,得到改进值 $y'_2 = -0.6528$,以便在下一步计算时使用. 到这里,完成了 PECE 程序的一步. 如果需要的话,还可以反复校正,直到收敛. 这个问题的精确解为 $y(t) = 1/(1+t^2)$,因而在积分点上的真值为 $y(0.25) = 0.9412, y(0.5) = 0.8$. ■

最常用的成对多步方法之一是显式四阶 Adams-Bashforth 预估格式

$$y_{k+1} = y_k + \frac{h}{24}(55y'_k - 59y'_{k-1} + 37y'_{k-2} - 9y'_{k-3})$$

和隐式四阶 Adams-Moulton 校正格式

$$y_{k+1} = y_k + \frac{h}{24}(9y'_{k+1} + 19y'_k - 5y'_{k-1} + y'_{k-2}).$$

例 9.10 刚性 为说明刚性常微分方程的数值求解,考虑初值条件为 $y(0)=1$ 的初值问题

$$y' = -100y + 100t + 101.$$

这个常微分方程的通解为 $y(t) = 1 + t + ce^{-100t}$, 满足初始条件的特解为 $y(t) = 1 + t$ (即 $c=0$). 由于解是线性的, 所以这个问题的欧拉方法理论上是精确的. 为了说明截断误差或舍入误差的影响, 给初始值一个轻微的扰动. 取步长 $h=0.1$, 对给定的初值, 结果为

t	0.0	0.1	0.2	0.3	0.4
精确解	1.00	1.10	1.20	1.30	1.40
欧拉解	0.99	1.19	0.39	8.59	-64.2
欧拉解	1.01	1.01	2.01	-5.99	67.0