考试复习-6-8

第六章 插值与逼近

6.1 Lagrange 插值

6.2 Newton插值多项式

6.3 Hermite插值

6.4 分段插值

6.5 【了解】样条插值

6.6 最小二乘法

6.7 正交多项式

第七章 数值积分

7.1 数值积分

7.2 插值型求积分公式

7.3 复化梯形与复化 Simpson

7.4 Gauss 型求积分公式

第八章 常微分方程数值解法

注意第 6—8 章概念较抽象，最好不要直接硬记，而是结合课后习题在具体情境中运用这些概念，通过做题把抽象内容落到可操作步骤，再反过来理解原理并形成清晰框架。

第六章 插值与逼近

在只知道函数若干节点值的情况下，构造一个在这些节点上完全一致的函数，用它代替未知的原函数。满足节点值的函数叫**插值函数**，构造它的过程叫**插值**。

> **若给定 n+1 个互异节点 $x_0, ..., x_n$ 与数值 y_k ，则满足插值条件的 次数 ≤ n 的多项式存在且唯一。**
>
> **这意味着无论用什么插值方法，不管你用 Lagrange 形式还是 Newton 形式来构造插值多项式，它们本质上是同一个多项式，因此它们的余项（误差项）也必须完全一致（意味着做完题可用待定系数法验证）**

6.1 Lagrange 插值

Lagrange 基函数

$$l_k(x)=\frac{x-x_0}{x_k-x_0}\cdot\frac{x-x_1}{x_k-x_1}\cdots\frac{x-x_{k-1}}{x_k-x_{k-1}}\cdot\frac{x-x_{k+1}}{x_k-x_{k+1}}\cdots\frac{x-x_n}{x_k-x_n}=\prod_{\substack{j=0\\j\neq k}}^{n}\frac{x-x_j}{x_k-x_j}$$

- 当 $x=x_k$ 时，每一项都变成 1，所以 $l_k(x_k)=1$ 。
- 当 $x=x_m$ $(m\neq k)$ 时，恰好有一项的分子变成 0，整串乘积变成 0，所以 $l_k(x_m)=0$ 。

因此 $l_k(x)$ 就是“**在第 k 个节点为 1，其他节点为 0 的多项式开关**”，用来拼出拉格朗日插值多项式。

拉格朗日插值多项式

$$L_n(x)=l_0(x)f_0+l_1(x)f_1+\cdots+l_n(x)f_n=\sum_{k=0}^{n}f_k\,l_k(x)$$

$$L_n(x_j)=\sum_{k=0}^{n}f_k\,l_k(x_j)=f_j$$

Lagrange 插值多项式的误差公式–插值余项

对任意 $x\in[a,b]$ ，真实函数 f(x) 与插值多项式 $L_n(x)$ 的误差为：

$$R_n(x)=f(x)-L_n(x)=\frac{f^{(n+1)}(\xi_x)}{(n+1)!}\,\omega_{n+1}(x)$$

1. 分子是导数次数
2. ξ_x 是一个 随 x 而变，落在节点区间 $[x_0,x_n]$ 里的某个数
3. $\omega_{n+1}(x)=(x-x_0)(x-x_1)\cdots(x-x_n)$

这个就是 Lagrange 插值误差项（余项）。

若 $|\,f^{(n+1)}(x)\,|$ 在区间 [a,b] 上有上界 M_{n+1} ，

则 Lagrange 插值余项满足估计：$|R_n(x)|\leq\dfrac{M_{n+1}}{(n+1)!}\,|\omega_{n+1}(x)|$

所有多项式插值（无论拉格朗日、牛顿、范德蒙、分段构造……）的误差公式本质上都是：

$$R_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\,\omega_{n+1}(x),$$

其中 $\omega_{n+1}(x) = (x-x_0)(x-x_1)\cdots(x-x_k) = \prod_{k=0}^{n}(x-x_k)$

只依赖节点，不依赖插值形式。

补充：

$\omega_{k+1}(x)$ **是一个过 k+1 个节点的 (k+1) 次多项式。**

$\omega'_{k+1}(x_j)$ 的意思就是 **把** $\omega_{k+1}(x)$ **对 x 求导后，代入** $x = x_j$ **的值**：

$$\omega'_{k+1}(x_j) = \frac{d}{dx}(x-x_0)(x-x_1)\cdots(x-x_k)\Big|_{x=x_j}$$

$C^2[a,b] = \{\text{在区间 }[a,b]\text{ 上连续且有连续一阶、二阶导数的函数}\}$也就是：

- S(x) 在整个区间上连续；
- S'(x) 在整个区间上连续；
- S''(x) 在整个区间上连续。

连续即可求导！

6.2 Newton插值多项式

一阶差商：两点的“离散导数”

给定两个互异节点 $x_i, x_j \ (i \neq j)$ ： $f[x_i, x_j] = \dfrac{f(x_j) - f(x_i)}{x_j - x_i}$

二阶差商：三个点的离散二阶导数

对三个互异点 x_i, x_j, x_k ：

$$f[x_i, x_j, x_k] = \frac{f[x_j, x_k] - f[x_i, x_j]}{x_k - x_i}$$

k 阶差商：逐级递推

对于节点 $x_0, x_1, \ldots, x_k$， k 阶差商定义为：

$$f[x_0, x_1, \ldots, x_k] = \frac{f[x_1, x_2, \ldots, x_k] - f[x_0, x_1, \ldots, x_{k-1}]}{x_k - x_0}$$

最高阶差商 = 最高次项系数，做k阶差商的结果就等于最高k阶幂的系数。

x_i	$f(x_i)$	一阶差商	二阶差商	三阶差商	$\cdots$	n 阶差商
x_0	$f(x_0)$					
x_1	$f(x_1)$	$f[x_0, x_1]$				
x_2	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$			
x_3	$f(x_3)$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	
x_n	$f(x_n)$	$f[x_{n-1}, x_n]$	$f[x_{n-2}, x_{n-1}, x_n]$	$f[x_{n-3}, x_{n-2}, x_{n-1}, x_n]$		$f[x_0, x_1, \ldots, x_n]$

重要性质

1. 若 $f(x)$ 具有k阶连续导数，则 $f(x)$ 的k阶差商等价于函数某点的 k 阶导数

   $$f[x_0, x_1, \ldots, x_k] = \frac{f^{(k)}(\xi)}{k!}, \quad \xi \in (x_0, x_1, \ldots, x_k) \text{ 之间的某个中间位置}$$

2. **若 f(x) 是 n 次多项式，则：**
   - 当 $k \le n$ ： $f[x_0, \ldots, x_k]$ 是一个 $(n-k)$ 次多项式
   - 当 $k > n$ ： $f[x_0, \ldots, x_k] \equiv 0$

   多项式的“阶数”随着差商减一层一层地下降，超过极限后就变成 0，就像连续求导一样。**n 次多项式的 n+1 阶导数恒为 0。每做一次差商，就是对多项式“做一次离散求导”。**

3. 差商对节点排列不敏感， $f[x_{i_0}, x_{i_1}, \ldots, x_{i_k}] = f[x_0, x_1, \ldots, x_k]$ ，顺序随便换。

Newton插值多项式及其余项

Newton 插值把插值多项式写成一种逐层递推的形式：

$$N_n(x) = f(x_0) + (x - x_0) f[x_0, x_1] + \cdots + \omega_n(x) f[x_0, \ldots, x_n]$$

这种写法最大的特点是：每增加一个新节点 x_{k+1}，只需要在原多项式后面再加一项，因此具有天然的**承袭性（递推性）**： $N_{k+1}(x) = N_k(x) + \omega_{k+1}(x) f[x_0, \ldots, x_{k+1}].$

将这个递推展开式与函数 f(x) 的真实值比较，可把 f(x) 本身也写成 Newton 形式：

$$f(x) = \sum_{k=0}^{n} \omega_k(x) f[x_0, \ldots, x_k] + \omega_{n+1}(x) f[x_0, \ldots, x_n, x].$$

前 n 项正是 Newton 插值多项式 $N_n(x)$，最后一项就是插值误差，因此自然得到余项：

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

由于插值多项式是唯一的，Lagrange 插值的余项与 Newton 的余项完全一致，也可以表示为

$$R_n(x) = \frac{f^{(n+1)}(\xi_x)}{(n+1)!}\omega_{n+1}(x)$$

重节点差商

正常差商要求节点互异，但在 Newton 插值、Hermite 插值等场景里，可能出现同一个节点重复使用的情况，这时差商定义中出现 0 作分母，于是我们必须用**极限**重新定义。**重复节点差商就是导数，阶数越高，差商越像高阶导数除以阶乘。**

一阶“重节点差商”： $f[x_0, x_0] = \lim\limits_{x \to x_0} f[x_0, x] = \lim\limits_{x \to x_0} \dfrac{f(x) - f(x_0)}{x - x_0} = f'(x_0)$

二阶“重节点差商”： $f[x_0, x_0, x_0] = \lim\limits_{x \to x_0} \dfrac{f[x_0, x_0] - f[x_0, x]}{x_0 - x} = \dfrac{f''(x_0)}{2!}$

混合重复节点差商： $f[x_0, x_0, x_1] = \dfrac{f[x_0, x_1] - f[x_0, x_0]}{x_1 - x_0}$

这正是泰勒公式一阶余项的形式，也可写成： $f[x_0, x_0, x_1] = \dfrac{f''(\xi)}{2!}, \qquad \xi \in (x_0, x_1)$

总结：重节点差商与导数关系

如果 f(x) 足够光滑： $\boxed{f[\underbrace{x_0, \dots, x_0}_{k+1}] = \dfrac{f^{(k)}(x_0)}{k!}}$

放到一般形式： $\boxed{f[x_0, x_0, \dots, x_0, x_1] = \dfrac{f^{(k)}(\xi)}{k!}}$ （ξ 在区间内）

6.3 Hermite插值

Hermite 插值多项式 = 不仅逼近函数值 f(x)，还同时逼近导数值 f’(x)、f’’(x)… 的插值多项式。

（带有导数插值条件的插值多项式称为Hermite插值多项式）

三次Hermite插值多项式求解步骤

确定三次插值多项式 $H_3(x)$，使满足条件：

$$H_3(x_0)=f(x_0)=y_0,\qquad H_3(x_1)=f(x_1)=y_1,$$
$$H_3'(x_0)=f'(x_0)=y_0',\qquad H_3'(x_1)=f'(x_1)=y_1'.$$

注： 满足以上插值条件的 Hermite 插值多项式是唯一存在的。

基函数法（结合具体题目理解）

步骤1：设 $H_3(x)$ 表达式（构造四个专门的基函数）

这里面是2个函数，2个导数，但根据题目也可以设置3个函数，一个导数。

> 三次 Hermite 插值的核心规律只有一句：**一个三次多项式只有 4 个自由度，所以无论你用“几个函数+几个导数”来构造，只要一共提供 4 条独立条件，就能唯一确定它。**
>
> 因此可以是“2 个函数值 + 2 个导数值”，也可以是“3 个函数值 + 1 个导数值”，形式不同，本质相同，只要条件数凑满 4 个，方程就能解、多项式就能定。
>
> Hermite 的技巧是：构造四个专门的“基函数”，每个基函数只满足自己的插值条件，其余条件全为 0。

$$H_3(x)=\varphi_0(x)y_0+\varphi_1(x)y_1+\psi_0(x)y_0'+\psi_1(x)y_1' \qquad (1)$$
$$H_3'(x)=\varphi_0'(x)y_0+\varphi_1'(x)y_1+\psi_0'(x)y_0'+\psi_1'(x)y_1' \qquad (2)$$

步骤2：分别将 x_0 和 x_1 代入（1）和（2）：

$$\varphi_0(x_0)=1,\quad \varphi_1(x_0)=0,\quad \psi_0(x_0)=0,\quad \psi_1(x_0)=0$$
$$\varphi_0(x_1)=0,\quad \varphi_1(x_1)=1,\quad \psi_0(x_1)=0,\quad \psi_1(x_1)=0$$
$$\varphi_0'(x_0)=0,\quad \varphi_1'(x_0)=0,\quad \psi_0'(x_0)=1,\quad \psi_1'(x_0)=0$$
$$\varphi_0'(x_0)=0,\quad \varphi_1'(x_0)=0,\quad \psi_0'(x_0)=0,\quad \psi_1'(x_0)=1$$

步骤三：**求基函数（一般就是这两种设的方式，具体题目再分析）**

$$\varphi_0(x)=(x-x_1)^2(ax+b)\quad (\text{代入条件 } \varphi_0(x_0)=1,\ \varphi_0'(x_0)=0 \text{ 解 a,b})$$
$$\varphi_1(x)=(x-x_0)^2(ax+b)\quad (\text{代入条件 } \varphi_1(x_1)=1,\ \varphi_1'(x_1)=0 \text{ 解 a,b})$$
$$\psi_0(x)=C(x-x_1)^2(x-x_0)\quad (\text{代入条件 } \psi_0'(x_0)=1,\ \psi_0'(x_1)=0 \text{ 解出 C})$$
$$\psi_1(x)=C(x-x_0)^2(x-x_1)\quad (\text{代入条件 } \psi_1'(x_1)=1,\ \psi_1'(x_0)=0 \text{ 解出 C})$$

插值多项式余项

$R_3(x)=f(x)-H_3(x)=\dfrac{f^{(4)}(\xi_x)}{4!}(x-x_0)^2(x-x_1)^2.$ （还是和上面误差一样的公式，后面的是 $\omega_{n+1}(x)$ ）

6.4 分段插值

Runge 现象简单说就是：**在等距节点上使用高次多项式插值，越想提高精度，结果越容易在区间两端发生剧烈振荡，反而变差**。可以采用分段插值方法。

注意误差公式

二次

若 $f \in C^2[a,b]$ ，则对任意 $x \in [x_{i-1}, x_i]$ ，

$$f(x) - S_1(x) = \frac{f''(\xi_i)}{2}(x - x_{i-1})(x - x_i)$$

进一步估计： $ab \le \left(\frac{a+b}{2}\right)^2 = \frac{(a+b)^2}{4}$ ， $h_i = x_i - x_{i-1},\qquad h = \max_{1\le i\le n} h_i$

x处于 x_i和x_{i-1} 之间的时候最大，a+b=2倍的1/2区间！

$$|f(x) - S_1(x)| \le \frac{M_2}{8}h^2,\quad M_2 = \max|f''(x)|.$$

三次

设一个距离a，另外两个距离用a表示，求函数最大值即可

6.5 【了解】样条插值

样条插值的出发点是：**用一条高次多项式穿过很多节点既不稳定也容易震荡，因此不如把区间切成许多小段，在每一小段上用低次（通常是三次）多项式来逼近。**

为了让整条曲线看起来自然、平滑又没有折痕，各段之间不仅要求函数值连续（不折断），还要求一阶导数连续（方向一致、不出现尖角），甚至二阶导数也连续（曲率平滑，不出现突然的弯折）。在这些约束下，每一段的三次多项式被唯一确定，所有小段拼接成一条整体上 C^2 连续的插值曲线。

最终结果像一根柔软的竹条被固定在若干钉子的位置，通过自然弯曲形成的平滑曲线：既精准经过所有节点，又不会像高次多项式插值那样在区间两端剧烈震荡，同时改变某个节点时只会影响附近的少数几段，具有良好的稳定性与局部性。

简言之：

样条插值 = 分段多项式 + 低次（通常三次） + 强制光滑拼接（函数值、一阶导、二阶导连续） = 一条自然、平稳、稳定的整体曲线。

6.6 最小二乘法

我们想找一个函数 $\varphi(x)$ 来逼近一堆离散数据点 (x_i, y_i) ，但这些点通常不在同一条光滑曲线上，所以不存在“精确插值”。因此我们引入“误差向量”：

$$\mathbf{e} = \begin{pmatrix} \varphi(x_0) - y_0 \\ \varphi(x_1) - y_1 \\ \vdots \\ \varphi(x_m) - y_m \end{pmatrix}$$

也就是说：每一维表示 $\varphi(x)$ 在某个点与真实数据的差。真实目标不是让误差为零，而是让误差“整体最小”。于是我们度量误差向量的大小： $\|\mathbf{e}\|_2$ 希望它最小。换句话说：**寻找一个函数，使它在所有数据点上的误差平方和最小。**这就是“最小二乘”。

> 拟合空间 = 你允许拟合函数所在的“函数集合 / 子空间”。
>
> 设一组基函数： $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$
>
> 它们的线性组合： $V = \mathrm{span}\{\varphi_0, \varphi_1, \dots, \varphi_n\}$
>
> 这个空间 V 就叫 拟合空间。你的拟合函数 $\varphi(x)$ 必须属于 V。
>
> 最小二乘法就是把数据向量投影到拟合空间 V 上。
>
> **例 ： 二次最小二乘拟合**
>
> 基函数： $\varphi_0(x) = 1, \quad \varphi_1(x) = x, \quad \varphi_2(x) = x^2$
>
> 拟合空间： $V = \mathrm{span}\{1, x, x^2\}$
>
> 所有二次多项式都在这里。

求解步骤

(1) 选拟合函数 (2) 写基函数 (3) 写离散内积 (4) 列正规方程 (5) 解系数 (6) 写出拟合函数 (7) 算误差

已知数据：给定离散点 $(x_0, y_0), (x_1, y_1), \dots, (x_m, y_m)$ ，

（若有权重，就给 $p(x_i) > 0$ ，否则默认 $p(x_i) = 1$ ）。

1. 选拟合函数空间，比如一次多项式： $\varphi(x) = p_1(x) = a_0 + a_1 x$
2. 写出基函数： $\varphi_0(x) = 1, \qquad \varphi_1(x) = x$

   （如果是二次，就再加一个$\varphi_2(x) = x^2$，步骤同理。）

   把每个基函数在所有采样点上的取值看成一个向量：

$$\boldsymbol{\varphi}_0 = \begin{pmatrix} \varphi_0(x_0) \\ \varphi_0(x_1) \\ \vdots \\ \varphi_0(x_m) \end{pmatrix}, \qquad \boldsymbol{\varphi}_1 = \begin{pmatrix} \varphi_1(x_0) \\ \varphi_1(x_1) \\ \vdots \\ \varphi_1(x_m) \end{pmatrix},$$

数据向量：

$$\mathbf{f} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_m \end{pmatrix}$$

3. 向量内积（内积时一定要带权函数，若无权重，就取 $p(x_k) \equiv 1$ ）

$(\varphi_i, \varphi_j) = \sum_{k=0}^{m} p(x_k)\, \varphi_i(x_k)\, \varphi_j(x_k)$ –对应的应该有4个式子

$(\mathbf{f}, \varphi_i) = \sum_{k=0}^{m} p(x_k)\, y_k\, \varphi_i(x_k)$ –对应的有2个式子

4. 写出正规方程（正交条件）

最小二乘要求误差向量（**误差 = 真实数据 — 拟合函数**）

设拟合函数

$$\hat{f} = \sum_{i=0}^{n} a_i \varphi_i,$$

误差向量定义为

$$e = f - \hat{f} = f - \sum_{i=0}^{n} a_i\, \varphi_i.$$

最小二乘要求误差与拟合空间正交，即对所有基函数皆有：

$$(e, \varphi_i) = 0, \quad i = 0, 1, \ldots, n.$$

将内积展开后，可得到 $(n+1) \times (n+1)$ 的**通用正规方程组**：

$$\underbrace{\begin{pmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \cdots & (\varphi_0, \varphi_n) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \cdots & (\varphi_1, \varphi_n) \\ \vdots & \vdots & \ddots & \vdots \\ (\varphi_n, \varphi_0) & (\varphi_n, \varphi_1) & \cdots & (\varphi_n, \varphi_n) \end{pmatrix}}_{\text{Gram矩阵 } G} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} (f, \varphi_0) \\ (f, \varphi_1) \\ \vdots \\ (f, \varphi_n) \end{pmatrix}.$$

5. 求解该线性系统即可得到拟合系数 $a_0, a_1, \ldots, a_n$ ：直接把上面的方程组按普通线性代数求解

6. 写出最小二乘拟合曲线： $\varphi^*(x) = p_n^*(x) = \sum_{i=0}^{n} a_i^*\, \varphi_i(x),$

7. 计算均方误差：

误差向量：

$$\mathbf{e}=\begin{pmatrix}\varphi^*(x_0)-y_0\\ \varphi^*(x_1)-y_1\\ \vdots\\ \varphi^*(x_m)-y_m\end{pmatrix}$$

带权 2–范数：

$$\|\mathbf{e}\|_2=\left(\sum_{i=0}^{m}p(x_i)\left[\varphi^*(x_i)-y_i\right]^2\right)^{1/2}$$

6.7 正交多项式

1. Schmidt 正交化（Gram–Schmidt 正交化）

给定函数序列 $f_0(x),\ f_1(x),\ f_2(x),\ \ldots$

构造正交函数序列 $g_0(x),\ g_1(x),\ g_2(x),\ \ldots$

公式如下：

$$g_0(x)=f_0(x)$$

$$g_1(x)=f_1(x)-\frac{(f_1,g_0)}{(g_0,g_0)}\,g_0(x)$$

$$g_2(x)=f_2(x)-\frac{(f_2,g_0)}{(g_0,g_0)}\,g_0(x)-\frac{(f_2,g_1)}{(g_1,g_1)}\,g_1(x)$$

$$g_3(x)=f_3(x)-\frac{(f_3,g_0)}{(g_0,g_0)}\,g_0(x)-\frac{(f_3,g_1)}{(g_1,g_1)}\,g_1(x)-\frac{(f_3,g_2)}{(g_2,g_2)}\,g_2(x)$$

$$\vdots$$

其中内积一般取：

$$(f,g)=\int_a^b f(x)g(x)\,dx\quad\text{或带权内积}\quad(f,g)=\int_a^b f(x)g(x)w(x)\,dx.$$

2. 由 $1,x,x^2,\ldots,x^n$ 经过 Schmidt 正交化可得到正交多项式

对基底

$$f_0(x)=1,\quad f_1(x)=x,\quad f_2(x)=x^2,\ldots$$

做 Schmidt（Gram–Schmidt）正交化，

可得到一组在 [a,b] 上的 **正交多项式序列**

$p_0(x), p_1(x), \ldots, p_n(x)$

正交的意义 = 让 $p_n(x)$ 的零点恰好成为最优节点

正交性带来两个重要性质：

1. **正交多项式的根全部落在区间** (0,1)

   ——自然成为求积节点，安全可靠。

2. **利用正交性，可以证明此节点选择使公式达最高阶** 2n−1

   对 n=2 即达到 3 阶精度，不浪费自由度。

换句话说： $\text{正交多项式} + \text{取其零点} = \text{自动给出最优的 Gauss 节点}$

第七章 数值积分

7.1 数值积分

基本概念

把积分想成“连续加和”，数值积分就是把这件事离散化：只在区间里选几个点，算出函数在这些点的值，再按一定权重加起来，去逼近真实的积分。不同的选点方式和权重组合，就得到不同的积分公式；它们的好坏取决于能否对一些简单函数（尤其是多项式）算得完全正确，也就是“代数精度”。代数精度越高，公式对一般函数的逼近也越准。

积分本来是“在整个区间上把无穷多个值累加”，数值积分做的事是把它改成“只看有限几个点”。于是把 $\int_a^b f(x)\,dx$ 用一个加权求和来代替：$\int_a^b f(x)\,dx \approx \sum_{k=0}^{n} A_k\, f(x_k)$. 这里 x_k 是你挑的采样点， A_k 是对应的权重。一个求积公式好不好，就看它对多项式能不能算准：如果对所有 $f(x) = x^j,\ j = 0, \ldots, m$

都完全相等 $\int_a^b x^j\,dx = \sum_{k=0}^{n} A_k x_k^j$, 那就说这个公式有 **m 次代数精度**。代数精度越高，公式对一般函数的积分也越接近真实值。

求积分公式的代数精度

1. 求积分式的一般形式

$$\int_a^b f(x)\,dx \;\approx\; \sum_{k=0}^{n} A_k\, f(x_k)$$

（其中 x_k 为节点，A_k 为求积系数）

2. 代数精度（m 次）

若求积公式

$$\int_a^b f(x)\,dx \approx \sum_{k=0}^{n} A_k f(x_k)$$

对 $f(x)=x^j,\quad j=0,1,\dots,m$ 都精确成立，但对 $f(x)=x^{m+1}$ 不再精确，

即：$\displaystyle\int_a^b x^j\,dx \;=\; \sum_{k=0}^{n} A_k x_k^j, \qquad j=0,1,\dots,m$

$$\int_a^b x^{m+1} dx \neq \sum_{k=0}^{n} A_k x_k^{m+1}$$

则称该求积公式具有 **m 次代数精度**。

使用待定系数法确定求积分公式的待定参数。

7.2 插值型求积分公式

插值型求积分公式的核心思想是：在区间 [a,b] 上选取若干节点，用通过这些节点的插值多项式 $P_n(x)$ 近似函数 f(x)，再把积分 $\int_a^b f(x)\,dx$ 替换为 $\int_a^b P_n(x)\,dx = \sum_{k=0}^{n} A_k f(x_k)$，从而得到**线性加权求和形式**的数值积分公式。

Newton–Cotes 公式是在区间 [a,b] 上取 **等距节点** $x_k = a + kh$ ，用插值多项式来逼近 f(x)，再对插值多项式作精确积分，从而得到 $\int_a^b f(x)\,dx \approx \sum_{k=0}^{n} A_k f(x_k)$，其中 A_k 为插值积分得到的权系数。

常用的 **Newton–Cotes 公式（需要都背出来）**包括：

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

选 n+1 个等距节点 $x_k = a + kh, \qquad h = \dfrac{b-a}{n}.$

对 f(x) 做 n 次插值 $P_n(x)$ ，积分近似为：

$$\int_a^b f(x)\,dx \approx \sum_{k=0}^{n} A_k f(x_k),$$

其中系数 A_k 由

$$A_k = \int_0^n l_k(t)\,dt$$ （ $l_k(t)$ 是 Lagrange 基函数）

通过等距节点积分得到。

误差项必定是负数。

梯形公式（n=1，代数精度 1）：

$$T(h) = \frac{h}{2}[f(a) + f(b)]$$

误差项：

$$|I - T(h)| \le \frac{(b-a)^3}{12} \max|f''(x)|$$

Simpson 公式（n=2，代数精度 3）：

$$S(h) = \frac{h}{3}[f(a) + 4f(\frac{a+b}{2}) + f(b)]$$

误差项：

$$|I - S(h)| \le \frac{(b-a)^5}{2880} \max|f^{(4)}(x)|$$

3/8 公式（n=3，代数精度 3）

$$\int_a^b f(x)\,dx \approx \frac{3h}{8}[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]$$

误差项：

$$E = -\frac{(b-a)^5}{6480} f^{(4)}(\xi)$$

Boole 公式（n=4，代数精度 5）—— 狭义的Cotes公式

$$\int_a^b f(x)\,dx \approx \frac{2h}{45}[7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)]$$

误差项：

$$E = -\frac{(b-a)^7}{1935360} f^{(6)}(\xi)$$

记忆方法：

“前面的系数” $\times$ “括号里所有整数之和” 必须等于 “总区间长度 nh ”

$$\begin{aligned} \text{梯形：} &\quad \frac{h}{2}(1+1) = 1h \\ \text{Simpson：} &\quad \frac{h}{3}(1+4+1) = 2h \\ \text{Simpson } 3/8\text{：} &\quad \frac{3h}{8}(1+3+3+1) = 3h \\ \text{Cotes } 4\text{ 点：} &\quad \frac{2h}{45}(7+32+12+32+7) = 4h \end{aligned}$$

对于标准的“下凸”函数（导数大于0），这些求积公式计算出的面积往往比真实面积大（过高估计）。所以这些误差都是负数。

误差项如何记忆

规律一：导数阶数（奇偶跃迁）

这是最重要的性质，决定了公式准不准。

- **n 是奇数**（1, 3）：导数阶数是 **n+1**。
  - 梯形 $(n=1) \to f''$
  - Simp 3/8 $(n=3) \to f^{(4)}$
- **n 是偶数**（2, 4）：导数阶数是 **n+2**（**超收敛**，多送一阶精度！）。
  - Simpson $(n=2) \to f^{(4)}$
  - Cotes $(n=4) \to f^{(6)}$

规律二：幂次跟随导数

误差项中 $(b-a)$ 的幂次，永远比导数阶数 **大 1**。

- $f''(2\text{阶}) \to (b-a)^3$
- $f^{(4)}(4\text{阶}) \to (b-a)^5$
- $f^{(6)}(6\text{阶}) \to (b-a)^7$

阶数 (n)	名称	导数阶数(决定精度)	误差项系数(记住分母)	误差项公式 R[f]

1 (奇)	**梯形**	f'' （2阶）	12	$-\dfrac{(b-a)^3}{\mathbf{12}}f''(\xi)$
2 (偶)	**Simpson**	$f^{(4)}$ （4阶）	2880	$-\dfrac{(b-a)^5}{\mathbf{2880}}f^{(4)}(\xi)$
3 (奇)	Simpson 3/8	$f^{(4)}$ （4阶）	6480	$-\dfrac{(b-a)^5}{\mathbf{6480}}f^{(4)}(\xi)$
4 (偶)	Cotes	$f^{(6)}$ （6阶）	1935360	$-\dfrac{(b-a)^7}{\mathbf{1935360}}f^{(6)}(\xi)$

7.3 复化梯形与复化 Simpson

复化在数值积分里指的是：**把一个大区间分成很多小子区间，在每个小区间上应用一次低阶求积公式，然后把所有小区间的结果加起来。**

把 [a,b] 等分成 n 段，步长 $h = \dfrac{b-a}{n}, \qquad x_k = a + kh.$

在每一段 $[x_k, x_{k+1}]$ 上都套一次你喜欢的求积公式（比如梯形、Simpson）。

1. 复化梯形

公式： $$\int_a^b f(x)\,dx \approx \frac{h}{2}\Big[f(a) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)\Big]$$

误差估计： $$|I - T_n| \le \frac{(b-a)^3}{12\,n^2} M_2, \qquad \Big(M_2 = \max_{a\le x\le b}|f''(x)|\Big)$$

（其中 n 为区间等分数）

2. 复化 Simpson【重点】

公式： $$\int_a^b f(x)\,dx \approx \frac{h}{6}\Big[f(a) + 4\sum_{k=1}^{n} f\big(x_{k-\frac{1}{2}}\big) + 2\sum_{k=1}^{n-1} f(x_k) + f(b)\Big]$$

误差估计： $$|I - S_n| \le \frac{(b-a)^5}{2880\,n^4} M_4, \qquad \Big(M_4 = \max_{a\le x\le b}|f^{(4)}(x)|\Big)$$

复化Simpson有中点出现，一般取两段。

7.4 Gauss 型求积分公式

1. Gauss 型求积公式的定义

$$\int_a^b \rho(x)\, f(x)\, dx \;\approx\; \sum_{i=1}^{n} A_i\, f(x_i) \quad \text{（从1开始）}$$

如果某求积公式具有 **2n−1 次代数精度**，则其对应的节点 $x_1, x_2, \ldots, x_n$ 称为 **Gauss 点**，此时的求积公式称为 **Gauss 型求积公式**。

（注）为讨论方便，本节取 n 个节点，并记节点为 $x_1, x_2, \ldots, x_n$ ，同时，所讨论的积分均为带权函数 $\rho(x)$ 的积分。

2. 构造 Gauss 型求积公式的步骤——重要

(1) 对给定区间 [a,b] 和权函数 $\rho(x)$ ，**由 Schmidt 正交化过程构造正交多项式** $P_0(x), P_1(x), \ldots, P_n(x).$

(2) 求出 $P_n(x)$ 的 n 个零点 $x_1, x_2, \ldots, x_n$，即为 **Gauss 点**。

(3) 计算求积系数 $A_i = \int_a^b \rho(x)\, l_i(x)\, dx, i = 1, 2, \ldots, n,$ 其中 $l_i(x)$ 为节点 x_i 对应的 Lagrange 基函数。

变形的 Gauss 型求积公式–记住权函数

1. Gauss−Legendre 求积公式

区间： $[-1, 1]$

权函数： $\rho(x) = 1$

特殊的：

从 p_0 开始构造正交多项式（Legendre 多项式）

权函数 $\rho(x) = 1$ ，区间 [−1,1]，使用 三项递推公式

$$p_0(x) = 1, \qquad p_1(x) = x$$

$$(n+1)p_{n+1}(x) = (2n+1)xp_n(x) - np_{n-1}(x)$$

Gauss−Legendre 求积权重 通用公式：

$$w_i = \frac{2}{\left(1 - x_i^2\right)[P_n'(x_i)]^2}$$

2. Gauss−Laguerre 求积公式

区间： $[0, +\infty)$

权函数： $\rho(x) = e^{-x}$

3. Gauss−Hermite 求积公式

区间：$(-\infty,+\infty)$

权函数：$\rho(x)=e^{-x^2}$

只要是 **n 点 Gauss 型求积**（不管是哪种权函数），它的**代数精度总是：** $2n-1$

看题目要求精确到几次多项式，用 $2n-1\geq m$ 判断所需的节点数。

或者根据给出的节点数 n 自动知道精度是 $2n-1$ 。

例 1：题目说“对 $1,x,x^2$ 都精确” → 最高次 $m=2$

$$2n-1\geq 2 \quad \Rightarrow \quad n\geq 2.$$

所以用 n=2 点 Gauss 就够了。

例 2：题目要求“三点 Gauss 求积公式”

那就是 n=3 →代数精度 automatically 是：

$$2\times 3-1=5\ (\text{五次精度})$$

看到 $\sqrt{\frac{1}{3}}$ 就是 2 点公式，看到 $\sqrt{\frac{3}{5}}$ 就是 3 点公式。这是记忆 Gauss–Legendre 积分节点最直观的方法。四阶比较复杂，不需要记忆！

标准积分区间为 [−1, 1]，公式形式为：

$$\int_{-1}^{1} f(x)dx \approx \sum_{k=1}^{n} A_k f(x_k)$$

1. 两点 Gauss 公式 (n=2)

关键词： $\pm\sqrt{\frac{1}{3}}$

这是最基础的公式，代数精度为 $2n-1=3$

- **节点 (** x_k **)**: $\pm\frac{1}{\sqrt{3}}$
- **权重 (** A_k **)**: 均为 1

完整公式： $\int_{-1}^{1} f(x)dx \approx 1\cdot f\left(-\frac{1}{\sqrt{3}}\right)+1\cdot f\left(\frac{1}{\sqrt{3}}\right)$

2. 三点 Gauss 公式 (n=3)

关键词： $\pm\sqrt{\frac{3}{5}}$ 和 0

代数精度为 $2n-1=5$ 。由于是对称区间，中间节点一定是 0。

- **节点（** x_k **）**: $0, \pm\sqrt{\frac{3}{5}}$
- **权重 (** A_k **)**: 中间点为 $\frac{8}{9}$ ，两端点为 $\frac{5}{9}$

完整公式： $\int_{-1}^{1} f(x)dx \approx \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right)+\frac{8}{9}f(0)+\frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right)$

正交多项式适用情况

情况	节点是否全给?	任务目标	是否必须正交?
Gauss 求积公式	❌ 节点未知	目标=达到最高代数精度 = 2n−1	✔ 必须正交求节点（最优）

一般求积公式构造	➖ 节点部分已给 but 未说明最优	只需使代数精度尽可能高但有限制	❌ 直接列代数方程即可
题目未要求最优性，只求系数	➖ 甚至节点未知	精度达到几阶是自然结果	❌ 无需正交

Gauss 这个词本身就意味着：在给定节点数时达到最高代数精度。也就是说题目虽然没写，但逻辑上已经默认：这是 Gauss 公式的定义，不需要题目额外声明。**也就意味着这时候当节点未知的时候，需要进行正交多项式！**

Romberg公式

k (分段)	T0 (梯形值)	T1 (一次外推)	T2 (二次外推)	T3 (三次外推)
0 (1段)	$T_0^{(0)}$			
1 (2段)	$T_0^{(1)} \to$	$T_1^{(0)}$		
2 (4段)	$T_0^{(2)} \to$	$T_1^{(1)} \to$	$T_2^{(0)}$	
3 (8段)	$T_0^{(3)} \to$	$T_1^{(2)} \to$	$T_2^{(1)} \to$	$T_3^{(0)}$ –结果

Romberg (龙贝格) 求积法 是一种用于计算定积分数值解的高精度算法。

简单来说，它的核心思想是：**"由粗糙变精细"**。它并不直接使用一个超级复杂的公式去计算积分，而是先用最简单的公式算出几个"粗糙"的结果，然后通过数学上的**外推技术**（Richardson 外推），把这些粗糙的结果组合起来，消除误差，从而得到一个非常精确的最终结果。

以下是通俗易懂的原理拆解：

1. 核心构成：两个"法宝"

Romberg 算法由两部分组成：

1. **复化梯形公式 (Trapezoidal Rule)**：这是基础。通过不断把积分区间一分为二（加密网格），计算出一列近似值。
2. **Richardson (理查森) 外推法**：这是加速器。利用梯形公式的误差规律，通过线性组合消除误差项，从而极大地提高精度。

2. 具体操作步骤

想象我们要计算积分 $I = \int_a^b f(x)dx$ 。

第一步：造“梯形序列”（第一列数据）

初始值公式（第一列 T_0 ）

在套用上面的公式之前，必须先算出第一列 $T_0^{(k)}$ 。这列数据就是**复化梯形公式**的值。

对于积分 $I = \int_a^b f(x)dx$ ：

$k = 0$ （1段）：

- $T_0^{(0)} = \frac{b-a}{2}[f(a) + f(b)]$

$k > 0$ （递推计算 2^k 段）：

为了减少计算量，通常用递推法算 T_0 ：

- $T_0^{(k)} = \frac{1}{2}T_0^{(k-1)} + h_k \sum_{i=1}^{2^{k-1}} f(x_{new})$
  - 其中 $h_k = \frac{b-a}{2^k}$ 是当前的步长。
  - $\sum f(x_{new})$ 表示只把**新增加的分点**的函数值加起来。

我们先用梯形公式计算积分，但是每次都把步长 h 减半（即区间分段数翻倍）：

- $T_0^{(0)}$ ：把区间分成 1 段（用梯形公式算一次）。
- $T_0^{(1)}$ ：把区间分成 2 段（算一次）。
- $T_0^{(2)}$ ：把区间分成 4 段（算一次）。
- ...以此类推。

这一列数据虽然精度在提高，但收敛速度比较慢。

第二步：用公式“加速”（生成后续列）

我们利用 Richardson 外推公式，把第一列的数据两两结合，生成第二列、第三列……

通用的递推公式是：

$$T_m^{(k)} = \frac{4^m T_{m-1}^{(k+1)} - T_{m-1}^{(k)}}{4^m - 1}$$

- 其中 m 代表外推的阶数（第几列），k 代表分段的层级。

符号含义：

- T：代表积分的近似值。
- m：代表**外推的阶数**（即在这个表中是第几列）。

    - m=1 时，分母是 $4^1-1=3$ （对应 Simpson 公式）。
    - m=2 时，分母是 $4^2-1=15$ （对应 Cotes 公式）。
    - m=3 时，分母是 $4^3-1=63$ 。
- k：代表**二分加密的次数**（即在这个表中是第几行）。

$$R=B+\frac{B-A}{4^m-1}$$

- **第 1 次外推 (m=1)**： $R=\dfrac{4B-A}{3}$
- **第 2 次外推 (m=2)**： $R=\dfrac{16B-A}{15}$
- **第 3 次外推 (m=3)**： $R=\dfrac{64B-A}{63}$

这个规律非常工整，分母总是 4^m-1 ，分子总是“精细值的 4^m 倍”减去“粗糙值”。

补充

积分中值定理：

若 f 在 $[a,b]$ 上连续，则存在 $\xi\in(a,b)$ ，使得：

$$\int_a^b f(x)\,dx=f(\xi)\,(b-a)$$

含义：积分就是“区间长度 × 某个点的函数值”。这个点一定存在，但一般找不到具体值。

拓展：

若：

- $g(x)$ 在 $[a,b]$ 上 **连续**，
- $w(x)$ 在 $[a,b]$ 上 **不变号**（即全 ≥ 0 或全 ≤ 0），

则存在某个 $\xi\in(a,b)$ ，使得：

$$\int_a^b g(x)\,w(x)\,dx=g(\xi)\int_a^b w(x)\,dx$$

拉格朗日中值定理：

对可导函数 f，在区间 $[a,b]$ 上**存在**一点 ξ_x ，使得：

$$f(b)-f(a)=f'(\xi)(b-a)$$

对可导函数 f，在区间 $[x,b]$ 上**存在**一点 ξ_x ，使得：

$$f(b)-f(x)=f'(\xi_x)(b-x)$$

可导函数 f，在区间 $[a,x]$ 上**存在**一点 ξ_x ，使得：

$f(x) - f(a) = f'(\xi_x)(x-a)$

$f(b) - f(a)$ 只与端点 a, b 有关，不依赖 x 。

写成 ξ_x 会暗示 ξ_x 随 x 变化，这是错误的。

区间	正确的中值定理写法	是否可写成 ξ_x
$[a,b]$	$f(b)-f(a)=f(\xi)(b-a)$	❌ 不能写 ξ_x(与 x 无关)
$[x,b]$	$f(b)-f(x)=f(\xi_x)(b-x)$	✔ 可以写 ξ_x
$[a,x]$	$f(x)-f(a)=f(\xi_x)(x-a)$	✔ 可以写 ξ_x

泰勒公式（含拉格朗日余项）

原始Taylor公式： $f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!}(x-a)^k$

带余项的Taylor公式： $f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!}(x-a)^k + R_{n+1}$

设 f 在点 $x=a$ 的某邻域内 n+1 次可导，则对任意 x 有：

$$f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!}(x-a)^k + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-a)^{n+1}$$

其中 $\xi \in (a,x)$ 或 $\xi \in (x,a)$ 。

例如三阶： $f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + \frac{f^{(3)}(\xi_x)}{6}(x-a)^3$

泰勒公式的阶由 余项的阶数 决定

第八章 常微分方程数值解法

一阶常微分方程初值问题的一般形式为：

$$\begin{cases} \frac{dy}{dx} = f(x,y), \quad a \le x \le b, \\ y(a) = \alpha \end{cases}$$

(其中 y 是 x 的已知函数， α 为给定的初值)

Lipschitz（利普希茨条件）

若函数 f(x,y) 在区域 $\{a \le x \le b,\ m < y < M\}$ 上连续，且关于 y 满足 Lipschitz（利普希茨）条件：

$$|f(x,y) - f(x,\bar{y})| \le L|y - \bar{y}|$$

对所有 $y, \bar{y}$ 成立，其中 $L > 0$ 为 Lipschitz 常数，则初值问题（在上述区域内）存在唯一解。

【注】此时 Lipschitz 常数 L 不必小于 1!

【例 1】

$$\begin{cases} \dfrac{dy}{dx} = 1 + x\sin(xy), & 0 \le x \le 2, \\ y(0) = 1 \end{cases}$$

解： 设 $f(x,y) = 1 + x\sin(xy)$

对任意 $y, \bar{y}$，对变量 y 应用微分中值定理，存在 η，使得

$$\frac{f(x,y) - f(x,\bar{y})}{y - \bar{y}} = \frac{\partial}{\partial y} f(x,y) = x^2\cos(x\eta)$$

于是 $|f(x,y) - f(x,\bar{y})| = |x^2\cos(x\eta)|\,|y - \bar{y}| \le 4|y - \bar{y}|$

因为此时 f(x,y) 关于 y 满足 Lipschitz 条件，且常数 L=4。

因此，从理论上讲，初值问题存在惟一解。

差分公式的格式

1. Euler公式： $y_{n+1} = y_n + hf(x_n, y_n)$
2. 梯形公式： $y_{n+1} = y_n + \dfrac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right]$
3. Euler中点公式： $y_{n+1} = y_{n-1} + 2hf(x_n, y_n)$
4. 改进Euler方法的两种格式

   格式1：

$$\begin{cases} \bar{y}_{n+1} = y_n + hf(x_n, y_n) \\ y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1})\right] \\ y_0 = \alpha, \qquad n = 0, 1, 2, \ldots, N-1 \end{cases}$$

   格式2

$$
\begin{cases}
y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2) \\
k_1 = f(x_n, y_n) \\
k_2 = f(x_n + h,\ y_n + hk_1) \\
y_0 = \alpha, \qquad n = 0, 1, 2, \dots, N-1
\end{cases}
$$

改进的Euler方法（Improved Euler Method），又被称为**Heun方法**（Heun's Method）或**预测–校正法**（Predictor–Corrector Method），是对经典Euler方法的一种优化。简单来说，经典Euler方法只看“脚下”的斜率来决定下一步往哪走，而改进的Euler方法会**先试探性地走一步，看看终点的斜率，然后取两者的平均值**来决定真正的步伐。

1. 核心公式（预测–校正机制）

改进的Euler方法将计算分为两步：

预测：先用标准的Euler公式算出一个“粗糙”的预估值 $\tilde{y}_{n+1}$ 。

1. $\tilde{y}_{n+1} = y_n + h \cdot f(x_n, y_n)$

校正：利用刚才预测到的点算出终点斜率，将起点斜率和终点斜率取平均，再算一次真正的 y_{n+1} 。

2. $y_{n+1} = y_n + \frac{h}{2}\left[\underbrace{f(x_n, y_n)}_{\text{起点斜率}} + \underbrace{f(x_{n+1}, \tilde{y}_{n+1})}_{\text{预估的终点斜率}}\right]$ （这个是已经将 $k1, k2$ 代入了）

2. 几何意义：梯形公式

- **经典Euler方法**相当于做矩形积分（左矩形公式），它假设这一步长内斜率不变，误差较大。
- **改进的Euler方法**相当于**做梯形积分**。它认为这一段路上的斜率是在变化的，用“起点斜率”和“终点斜率”的平均值来代表这一段的平均变化率，从而让走出的直线更贴近真实的曲线。

局部截断误差 $y(x_{n+1}) - y_{n+1}$

1. 局部截断误差的阶数

若差分格式的局部截断误差为 $O(h^{p+1})$ ，则称该公式为 p **阶公式。**

2. Taylor公式

一元 Taylor 公式：

$$
y(x_{n+1}) = y(x_n + h) = y(x_n) + y'(x_n)h + \frac{y''(x_n)}{2!}h^2 + \frac{y^{(3)}(x_n)}{3!}h^3 + O(h^4)
$$

二元 Taylor 公式：

$$f(x+\Delta x, y+\Delta y)=$$
$$f+f_x\Delta x+f_y\Delta y+\frac{1}{2}(f_{xx}\Delta x^2+2f_{xy}\Delta x\Delta y+f_{yy}\Delta y^2)+O(\Delta x^3+\Delta y^3)$$

3. $y'(x_n)$, $y''(x_n)$, $y'''(x_n)$ **公式**

因为 $y'(x)=f(x,y(x))$ ，所以若 $y(x_n)=y_n$ ，则有：

$$y'(x_n)=f(x_n,y(x_n))=f(x_n,y_n)=f_n$$

$$y''(x_n)=\frac{\partial f(x_n,y_n)}{\partial x}+\frac{\partial f(x_n,y_n)}{\partial y}y'(x_n)=\frac{\partial f_n}{\partial x}+\frac{\partial f_n}{\partial y}f_n$$

$$y'''(x_n)=\frac{\partial^2 f_n}{\partial x^2}+2\frac{\partial^2 f_n}{\partial x\partial y}f_n+\frac{\partial^2 f_n}{\partial y^2}f_n^2+\frac{\partial f_n}{\partial x}\frac{\partial f_n}{\partial y}+\left(\frac{\partial f_n}{\partial y}\right)^2 f_n$$

4. 关于改进 Euler 方法的局部截断误差分析

改进 Euler 方法格式：

$$\begin{cases} y_{n+1}=y_n+\frac{h}{2}(k_1+k_2) \\ k_1=f(x_n,y_n) \\ k_2=f(x_n+h,\ y_n+hk_1) \\ y_0=\alpha, \qquad n=0,1,2,\ldots,N-1 \end{cases}$$

以下考试重点内容

常见 R–K 格式（直接可用）

① 二阶 R–K（含 Euler 改进法）

常见两种：**Heun 公式（梯形思想）**

$$\begin{cases} k_1 = f(x_n, y_n) \\ k_2 = f(x_n + h,\ y_n + hk_1) \\ y_{n+1} = y_n + \dfrac{h}{2}(k_1 + k_2) \end{cases}$$

中点 R–K2

$$\begin{cases} k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \frac{h}{2},\ y_n + \frac{h}{2}k_1) \\ y_{n+1} = y_n + hk_2 \end{cases}$$

均为**2 阶**，局部截断误差 $O(h^3)$ 。

② 三阶 R–K（考试常考）

$$\begin{cases} k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \frac{h}{2},\ y_n + \frac{h}{2}k_1) \\ k_3 = f(x_n + h,\ y_n - hk_1 + 2hk_2) \\ y_{n+1} = y_n + \dfrac{h}{6}(k_1 + 4k_2 + k_3) \end{cases}$$

阶数：**3 阶**，误差 $O(h^4)$ 。

③ 四阶经典 R–K4（必须背）

$$\begin{cases} k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \frac{h}{2},\ y_n + \frac{h}{2}k_1) \\ k_3 = f(x_n + \frac{h}{2},\ y_n + \frac{h}{2}k_2) \\ k_4 = f(x_n + h,\ y_n + hk_3) \\ y_{n+1} = y_n + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{cases}$$

最常用、四阶误差 $O(h^5)$ 。工程/仿真普遍采用。

判断单步方法的 **收敛性**

步骤：

1. **写出** Φ ：根据题目给出的格式，识别出 $y_{n+1} = y_n + h[\dots]$ 中括号里的部分就是 Φ 。
2. **验证相容性**：令 h=0，看 Φ 是否等于 $f(x, y)$ 。（通常显式单步法都满足）。
3. **验证 Lipschitz**：
    - 对 Φ 求 y 的偏导数 $\frac{\partial \Phi}{\partial y}$ 。
    - 如果 f(x,y) 本身满足 Lipschitz 条件（即 $\left|\frac{\partial f}{\partial y}\right| \leq L$ ），且 Φ 是 f 的线性组合，那么通常 $\left|\frac{\partial \Phi}{\partial y}\right|$ 也是有界的。
    - **结论**：因为偏导数有界，所以满足 Lipschitz 条件 $\rightarrow$ **方法收敛**。

会求单步方法的 **绝对稳定区域、稳定区间。**

Step 1: 代入模型方程

将 $y' = \lambda y$ 代入数值公式。即把所有的 $f(x_k, y_k)$ 替换为 λy_k 。

Step 2: 提取 $E(z)$

整理方程，使得左边只剩 y_{n+1} ，右边提取出公因式 y_n 。将出现的 λh 替换为 z 。

形式必然是： $y_{n+1} = E(z) y_n$ 。

Step 3: 解不等式

求解 $|E(z)| < 1$ 。

- 如果是求**区域**（复平面），通常是画图（比如内部是一个圆）。
- 如果是求**区间**（实数），直接解代数不等式 $-1 < E(z) < 1$ 。

题目： 求 **显式 Euler 法** $y_{n+1} = y_n + hf(x_n, y_n)$ 的绝对稳定区间。

解答步骤：

代入模型方程：

将 $f(x_n, y_n) = \lambda y_n$ 代入公式：

1. $y_{n+1} = y_n + h(\lambda y_n)$

整理得到 $E(z)$ ：

2. $y_{n+1} = (1 + h\lambda)y_n$

令 $z = h\lambda$ ，得：

3. $y_{n+1} = (1 + z)y_n$

所以，稳定函数为 $E(z) = 1 + z$ 。

求解稳定区间：

我们需要 $|E(z)| < 1$ ，即：

4. $|1 + z| < 1$

对于实数区间，即：

5. $-1 < 1 + z < 1$

同时减去 1：

6. $-2 < z < 0$

结论：

- 显式 Euler 法的**绝对稳定区域**是复平面上以 $(-1, 0)$ 为圆心，半径为 1 的圆的**内部**。
- 显式 Euler 法的**绝对稳定区间**是 $(-2, 0)$ 。

在常微分方程（ODE）的标准写法中，我们总是写成：

$y' = f(x, y)$

这里的等号告诉我们要怎么理解：

- 左边的 y' 代表 **导数**（即斜率）。
- 右边的 $f(x, y)$ 代表 **计算导数的规则**。