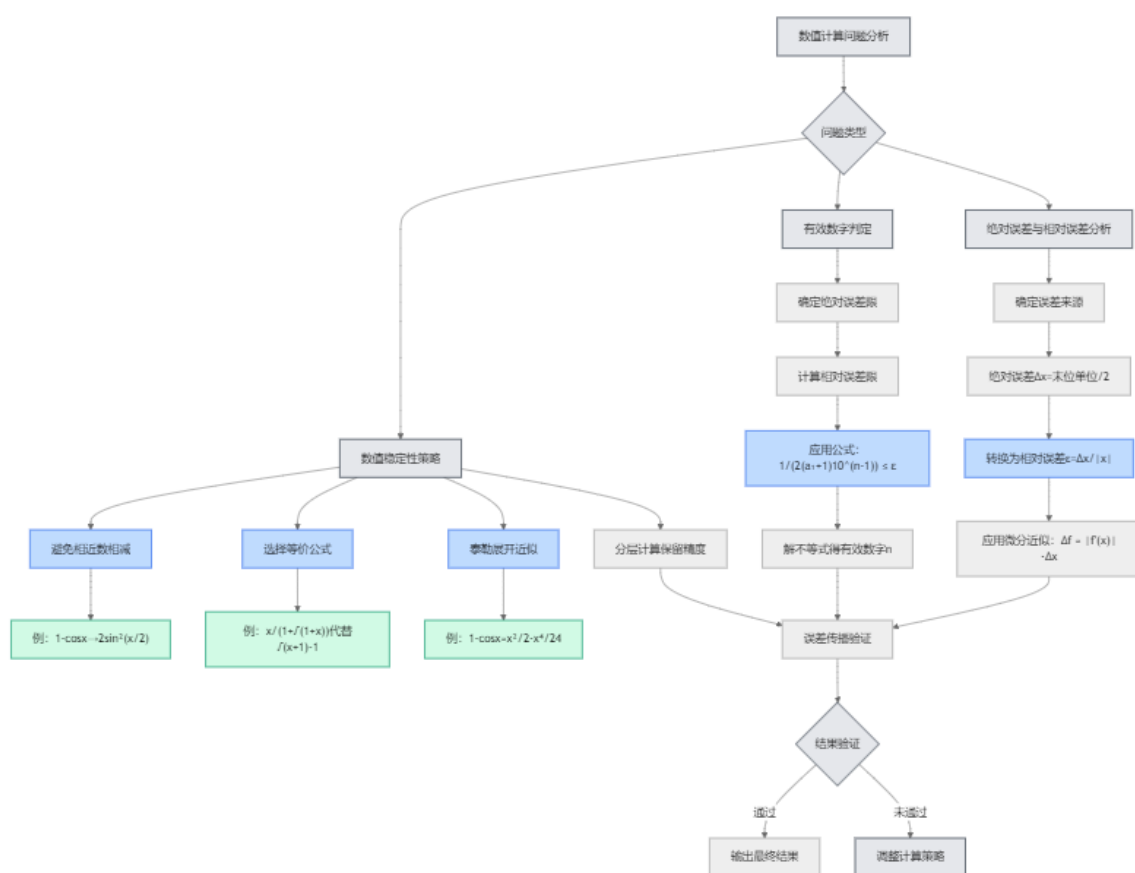


Chapter1-误差与有效数字

姓名：张恒祯 学号：2351071

日期：2025 年 3 月 2 日

1 思路图



2 必做题 (奇数)

1 近似值误差分析

解题诀窍：取最后一位乘 $\frac{1}{2}$ 。(1) $x_1^* = 0.024$

$$\delta_{x_1} = 0.0005, \quad \varepsilon_{x_1} = \frac{0.0005}{0.024} \approx 2.08\%, \quad 2 \text{ 位有效数字} \quad (1)$$

(2) $x_2^* = 0.4135$

$$\delta_{x_2} = 0.00005, \quad \varepsilon_{x_2} = \frac{0.00005}{0.4135} \approx 0.012\%, \quad 4 \text{ 位有效数字} \quad (2)$$

(3) $x_3^* = 57.50$

$$\delta_{x_3} = 0.005, \quad \varepsilon_{x_3} = \frac{0.005}{57.50} \approx 0.0087\%, \quad 4 \text{ 位有效数字} \quad (3)$$

(4) $x_4^* = 60000$

$$\delta_{x_4} = 0.5, \quad \varepsilon_{x_4} = \frac{0.5}{60000} \approx 0.00\%, \quad 5 \text{ 位有效数字} \quad (4)$$

(5) $x_5^* = 8 \times 10^5$

$$\delta_{x_5} = 5 \times 10^4, \quad \varepsilon_{x_5} = \frac{5 \times 10^4}{8 \times 10^5} = 6.25\%, \quad 1 \text{ 位有效数字} \quad (5)$$

3 $\sqrt{11}$ 的有效数字分析

设需要 n 位有效数字，相对误差限满足：

$$\frac{1}{2 \times 10^{n-1}} \leq 0.001 \quad (6)$$

取 $\sqrt{11} \approx 3.3166$ ，首位数字 $a_1 = 3$ ，修正公式：

$$\frac{1}{2(a_1 + 1) \times 10^{n-1}} \leq 0.001 \Rightarrow n \geq 3.70 \quad (7)$$

向上取整得： $\boxed{4}$ 位有效数字

5 边长误差限计算

已知：正方形边长 $l \approx 100 \text{ cm}$ ，要求面积误差 $\Delta A \leq 1 \text{ cm}^2$ 。

步骤：面积公式 $A = l^2$ ，误差传播关系：

$$\Delta A \approx 2l \cdot \Delta l \quad (8)$$

解边长误差限：

$$\Delta l \leq \frac{\Delta A}{2l} = \frac{1}{2 \times 100} = 0.005 \text{ cm} \quad (9)$$

单位换算：

$$0.005 \text{ cm} = 0.05 \text{ mm} \quad (10)$$

结论：边长测量允许最大误差为 $\boxed{\pm 0.005 \text{ cm}}$

问题 7 方程求根

方程: $x^2 - 40x + 1 = 0$, 已知 $\sqrt{399} \approx 19.975$

解法: 判别式计算:

$$\Delta = 40^2 - 4 \times 1 = 1596 = 4 \times 399 \Rightarrow \sqrt{\Delta} = 2\sqrt{399} \approx 39.95$$

根 1 (避免相近数相减):

$$x_1 = \frac{40 + 39.95}{2} = \frac{79.95}{2} = 39.975 \quad (\text{直接计算})$$

根 2 (利用韦达定理):

$$x_2 = \frac{1}{x_1} = \frac{1}{39.975} = 0.02501876 \approx 0.025019$$

结果验证:

$$x_1 \times x_2 = 39.975 \times 0.025019 \approx 1.0000 \quad (\text{满足方程})$$

结论: 根为 $\boxed{39.98}$ 和 $\boxed{0.02502}$ (四位有效数字)。

问题 9 优化计算方法

(1) $1 - \cos 1^\circ$: 采用泰勒展开避免相近数相减:

$$1 - \cos x \approx \frac{x^2}{2} - \frac{x^4}{24}, \quad x = \frac{\pi}{180} \approx 0.017453$$

计算结果:

$$1 - \cos 1^\circ \approx 1.523 \times 10^{-4} \quad (\text{四位有效数字})$$

(2) $\ln(30 - \sqrt{30^2 - \sqrt{30^3 - 1}})$: 分层计算保留中间量高精度:

$$\sqrt{30^3 - 1} \approx 164.3160 \quad (\text{六位开方})$$

$$\sqrt{30^2 - 164.3160} = \sqrt{735.684} \approx 27.1190$$

$$\ln(30 - 27.1190) = \ln(2.8810) \approx 1.0575$$

(3) $\frac{1 - \cos x}{\sin x}$ ($|x|$ 极小): 三角恒等变换:

$$\frac{1 - \cos x}{\sin x} = \tan \frac{x}{2} \approx \frac{x}{2} \quad (|x| \rightarrow 0)$$

(4) $\frac{1}{N+1}$ (N 充分大): 泰勒展开保留主项:

$$\frac{1}{N+1} = \frac{1}{N} - \frac{1}{N^2} + o\left(\frac{1}{N^2}\right)$$

结论: 通过数学变换和分层计算可提高精度。

3 选做题 (偶数)

2 π^{10} 的有效数字分析

设需要 n 位有效数字, 相对误差限满足:

$$\frac{1}{2 \times 10^{n-1}} \leq 0.001 \quad (11)$$

取 $\pi^{10} \approx 93648.05$, 首位数字 $a_1 = 9$, 修正公式:

$$\frac{1}{2(a_1 + 1) \times 10^{n-1}} \leq 0.001 \Rightarrow n \geq 4.47 \quad (12)$$

向上取整得: $\boxed{5}$ 位有效数字

4 纬度计算误差分析

已知: 纬度 $\phi = 45^\circ 0' 2''$ (测量到秒), 求 $\sin \phi$ 的误差。

步骤: 角度测量误差 $\Delta\phi = \pm 0.5''$, 转换为弧度:

$$\Delta\phi = 0.5'' \times \frac{1^\circ}{3600''} \times \frac{\pi}{180} = 2.424 \times 10^{-6} \text{ rad} \quad (13)$$

利用微分近似计算 $\sin \phi$ 误差:

$$\Delta(\sin \phi) \approx |\cos \phi| \cdot \Delta\phi \quad (14)$$

代入 $\phi = 45^\circ$, $\cos 45^\circ = \frac{\sqrt{2}}{2}$:

$$\Delta(\sin \phi) \approx 0.7071 \times 2.424 \times 10^{-6} = 1.715 \times 10^{-6} \quad (15)$$

结论: $\sin \phi$ 的误差为 $\pm 1.7 \times 10^{-6}$

6 多项式误差传播分析

已知: $y = P(x) = x^2 + x - 1150$, 精确值 $x = \frac{100}{3}$, 近似值 $x^* = 33$ 。

步骤:

$$y_{\text{精确}} = P\left(\frac{100}{3}\right) = \left(\frac{100}{3}\right)^2 + \frac{100}{3} - 1150 = -\frac{50}{9} \approx -5.5556 \quad (16)$$

$$y^* = P(33) = 33^2 + 33 - 1150 = -28 \quad (17)$$

相对误差计算:

$$\varepsilon_x = \frac{|33 - \frac{100}{3}|}{\frac{100}{3}} = \frac{\frac{1}{3}}{\frac{100}{3}} = 1\% \quad (18)$$

$$\varepsilon_y = \frac{|-28 - (-\frac{50}{9})|}{|\frac{50}{9}|} = \frac{\frac{202}{9}}{\frac{50}{9}} = 404\% \quad (19)$$

结论: 相对误差为 $\varepsilon_x = \boxed{1\%}$, $\varepsilon_y = \boxed{404\%}$

问题 8 误差传播分析

表达式:

$$(1) y_1 = \frac{x^2 \pm \varepsilon}{100}, \quad (2) y_2 = \frac{x^2 \pm \varepsilon}{0.001}$$

误差分析: 对 $y = \frac{f(x)}{a}$, 绝对误差满足 $\Delta y = \frac{\Delta f}{|a|}$

(1) y_1 计算:

$$\Delta y_1 = \frac{|2x\varepsilon| + \varepsilon}{100} \leq \frac{(2|x| + 1)\varepsilon}{100}$$

特性: 分母 100 使绝对误差缩小 100 倍, 相对误差保持不变。

(2) y_2 计算:

$$\Delta y_2 = \frac{|2x\varepsilon| + \varepsilon}{0.001} = 1000(2|x| + 1)\varepsilon$$

特性: 分母 0.001 使绝对误差放大 1000 倍, 相对误差仍不变。

结论: 表达式 (1) 误差缩小, 表达式 (2) 误差放大。

10. 数值稳定性公式选择

(1) 已知 $|x| \leq 1$

$$(a) y = \frac{1}{1+2x} - \frac{1-x}{1+x}$$

$$(b) y = \frac{2x^2}{(1+2x)(1+x)}$$

选择: (b) 原因: 当 $|x| \ll 1$ 时, (a) 式中两分数项的值近似相等 ($\approx 1-x$), 直接相减会导致有效数字丢失。例如 $x = 0.001$ 时:

$$\frac{1}{1.002} - \frac{0.999}{1.001} \approx 0.9980 - 0.9980 = 0 \quad (\text{实际应有 } y \approx 2 \times 10^{-6})$$

而 (b) 式通过通分消除减法操作, 计算更稳定。

(2) 已知 $|x| \gg 1$

$$(a) y = \frac{2}{x \left(\sqrt{x + \frac{1}{x}} + \sqrt{x - \frac{1}{x}} \right)}$$

$$(b) y = \sqrt{x + \frac{1}{x}} - \sqrt{x - \frac{1}{x}}$$

选择: (a) 原因: 当 $|x| \gg 1$ 时, (b) 式中两个平方根项近似相等, 相减导致有效数字丢失。例如 $x = 1000$ 时:

$$\sqrt{1000.001} - \sqrt{999.999} \approx 31.6228 - 31.6222 = 0.0006 \quad (\text{实际应有 } y \approx 3.16 \times 10^{-8})$$

而 (a) 式通过分母相加保留修正项, 避免减法误差。

(3) 已知 $|x| \leq 1$

$$(a) y = \frac{2 \sin^2 x}{x}$$

$$(b) y = \frac{1 - \cos 2x}{x}$$

选择: (a) 原因: 当 $|x| \ll 1$ 时, (b) 式中 $1 - \cos 2x \approx 2x^2$ 为相近数相减, 例如 $x = 0.001$ 时:

$$\frac{1 - 0.999998}{0.001} = \frac{2 \times 10^{-6}}{0.001} = 0.002 \quad (\text{有效数字丢失})$$

而 (a) 式通过 $\sin^2 x = x^2 - \frac{x^4}{3} + \dots$ 直接计算更稳定。

(4) 已知 $p \gg q > 0$

$$(a) y = \frac{q^2}{p + \sqrt{p^2 + q^2}}$$

$$(b) y = \sqrt{p^2 + q^2} - p$$

选择: (a) 原因: 当 $p \gg q$ 时, (b) 式中 $\sqrt{p^2 + q^2} \approx p + \frac{q^2}{2p}$, 相减后:

$$y \approx \frac{q^2}{2p} \quad (\text{但计算时因大数相减丢失精度})$$

而 (a) 式分母 $p + \sqrt{p^2 + q^2} \approx 2p$, 直接得到:

$$y = \frac{q^2}{2p} \quad (\text{无减法操作, 精度更高})$$

4 有效数字计算函数代码

```
1 function [n, e] = getdigits(xtrue, x)
2 % GETDIGITS Calculate significant digits and error bound
3 % Inputs:
4 %   xtrue - True value
5 %   x      - Approximate value
6 % Outputs:
7 %   n      - Number of significant digits
8 %   e      - Absolute error bound
9
10 err = abs(xtrue - x); % Calculate absolute error
11 [x_norm, m] = scientific_notation(x); % Convert to scientific notation
12 [err_norm, q] = scientific_notation(err); % Convert error
13
14 % Determine significant digits based on leading error digit
15 if err_norm < 5
16     n = m - q;
17 else
18     n = m - q - 1;
19 end
20
21 % Calculate error bound (0.5 * 10^(m-n))
22 e = 0.5 * 10^(m - n);
23
24 % Nested function: Scientific notation conversion
25 function [x_scaled, exponent] = scientific_notation(x)
26 x = abs(x);
27 exponent = 0; % Initialize exponent
28
29 if x == 0
30     x_scaled = 0;
31     exponent = 0;
32     return;
33 end
34
35 % Normalize to 1    x_scaled < 10
36 while x >= 10
37     x = x / 10;
38     exponent = exponent + 1;
39 end
40 while x < 1
41     x = x * 10;
42     exponent = exponent - 1;
43 end
44
45 x_scaled = x;
46 exponent = exponent; % Final exponent
47 end
```

Listing 1: MATLAB 有效数字计算函数 (getdigits.m)

5 附作业手写记录照片

Handwritten calculations and diagrams for various physics and mathematics problems.

1-3 $U = PV = 1.4015 \text{ Pa} \cdot \text{s}$
 $T = U \frac{dU}{dy} = 1.4015 \cdot \frac{4}{2 \times 10^{-3}} = 2.803 \times 10^3 \text{ Pa}$
 $M = \frac{1}{2} \rho V \omega \frac{d^2 u}{dy^2}$

1-7 $U = \frac{1}{2} \omega d$ $U = PV$ $T = U \frac{dU}{dy} = PV \frac{dU}{dy}$
 $A = \pi d L$ $F = TA = \frac{1}{2} \rho V \omega \frac{d^2 u}{dy^2}$
 $M = F \cdot \frac{1}{2} d = \frac{1}{4} \rho V \omega \frac{d^3 u}{dy^3}$ $W = \frac{2\pi \times 1}{60}$
 $U = 1.86 \text{ Pa} \cdot \text{s}$

1-11 $T = U \cdot \frac{dU}{dy} = U \cdot V \left(\frac{1}{h} + \frac{1}{8h} \right)$
 $f = T \cdot a$
 $\text{① } \frac{1}{2} h = 10 \text{ mm}, f_1 = 84 \text{ N}$
 $\text{② } \frac{d}{dt} \frac{dT}{dh} = 0 \Rightarrow h = \frac{1}{2} h, h = 20 \text{ mm}$

1-13 $T_1 = 20^\circ \text{C}, P_1 = 10^5 \text{ Pa}, V_1 = 10 \text{ m}^3, P_2 = 10^5 \text{ Pa}$
 $\text{等温: } V_2 = \frac{P_1 V_1}{P_2} = 2 \text{ m}^3$ $\text{等压: } P_1 V_1 = P_2 V_2, \gamma = \frac{C_p}{C_v} = 1.4$
 $\text{等 } V_2 = 3.16 \text{ m}^3$

3. $\sqrt{11} = 3.317$ $a_1 = 3$
 $\left| \epsilon_r \right| \leq \frac{1}{2a_1} \times 10^{-4} \leq 0.1\% = 10^{-3}$
 $10^{-4} \leq 6 \cdot 10^{-4}$
 $n \geq 4$ 至少4修有效数字

5. $\epsilon(X^2) \leq 1 \text{ cm}^2, X^* = 100$
 $\epsilon(X^{*2}) = |2X^*| \epsilon(X^*) = 200 \epsilon(X^*) \leq 1$
 $\epsilon(X^*) \leq \frac{1}{200} \text{ cm} = 0.005 \text{ cm}$

7. $X^2 - 40X + 1 = 0$
 $X_1 = 20 + \sqrt{399} \approx 39.975$
 $X_2 = 20 - \sqrt{399} = \frac{1}{20 + \sqrt{399}} \approx 0.0125$
 $\sqrt{399} \approx 19.975$

9. (1) $1 - 105X \approx \frac{X^2}{2} - \frac{X^4}{24}$
 $1 - 105 \cdot 10^{-2} \approx \frac{1}{2} \left(\frac{1}{10} \right)^2 - \frac{1}{24} \left(\frac{1}{10} \right)^4$
 ≈ 0.001523

(3) $\frac{1 - 105X}{\sin X} = \frac{2 \sin^2(\frac{X}{2})}{2 \sin \frac{X}{2} \cos \frac{X}{2}} = \tan \frac{X}{2}$

(4) $\int_N^{\infty} \frac{dx}{1+x^2} = \arctan(N+1) - \arctan N$
 $= \arctan \left[\frac{\tan[\arctan(N+1) - \arctan N]}{1 + \tan[\arctan(N+1) - \arctan N]} \right]$
 $= \arctan \frac{N+1-N}{1+(N+1)N} = \arctan \frac{1}{1+N+N^2}$

(2) $\ln(30 - \sqrt{30^2 - 1}) = -\ln(30 + \sqrt{30^2 - 1})$
 $= -\ln(30 + 29.9835) \approx -4.09467$

6 Python 有效数字计算函数

```
1 from typing import Tuple
2 import math
3
4 def getdigits(xtrue: float, x: float) -> Tuple[int, float]:
5     """
6     计算近似值的有效数字位数及误差限
7
8     Args:
9     xtrue (float): 真实值
10    x (float): 近似值
11
12    Returns:
13    Tuple[int, float]: (有效数字位数, 绝对误差限)
14    """
15
16    def scientific_notation(num: float) -> Tuple[float, int]:
17        """
18        将数值转换为标准科学计数法形式 (1 ≤ mantissa < 10)
19
20        Args:
21        num (float): 输入数值
22
23        Returns:
24        Tuple[float, int]: (尾数, 指数)
25        """
26        if num == 0:
27            return 0.0, 0
28
29        num = abs(num)
30        exponent = 0
31
32        # 处理大数
33        while num >= 10:
34            num /= 10
35            exponent += 1
36
37        # 处理小数
38        while num < 1:
39            num *= 10
40            exponent -= 1
41
42        return num, exponent
43
44    # 计算绝对误差
45    err = abs(xtrue - x)
46
47    # 转换为科学计数法
```

```
48 x_scaled, m = scientific_notation(x)
49 err_scaled, q = scientific_notation(err) if err != 0 else (0.0, 0)
50
51 # 获取误差首位数字
52 first_err_digit = int(math.floor(err_scaled)) if err_scaled >= 1 else 0
53
54 # 判断有效位数
55 if first_err_digit < 5:
56     n = m - q
57 else:
58     n = m - q - 1
59
60 # 计算绝对误差限
61 e = 0.5 * (10 ** (m - n))
62
63 return n, e
64
65
66 # 测试用例
67 if __name__ == "__main__":
68     # 示例1: 整数情况
69     print(getdigits(123.45, 123.40)) # (4, 0.05)
70
71     # 示例2: 小数情况
72     print(getdigits(0.0245, 0.024)) # (2, 0.0005)
73
74     # 示例3: 零值处理
75     print(getdigits(0.0, 0.0)) # (0, 0.0)
76
77     # 示例4: 大数情况
78     print(getdigits(123456, 123400)) # (4, 50.0)
```

Listing 2: Python 有效数字计算函数 (getdigits.py)

7 C++ 有效数字计算函数

```
1 #include <iostream>
2 #include <cmath>
3 #include <utility>
4
5 using namespace std;
6
7 pair<int, double> getdigits(double xtrue, double x) {
8     /**
9     * 计算近似值的有效数字位数及误差限
10    * @param xtrue 真实值
11    * @param x 近似值
12    * @return pair<有效数字位数, 绝对误差限>
```

```
13 */
14
15 auto scientific_notation = [](double num) -> pair<double, int> {
16 /**
17 * 将数值转换为标准科学计数法形式 (1 mantissa < 10)
18 * @param num 输入数值
19 * @return pair<尾数, 指数>
20 */
21 if (num == 0) return {0.0, 0};
22
23 num = abs(num);
24 int exponent = 0;
25
26 // 处理大数
27 while (num >= 10) {
28 num /= 10;
29 exponent++;
30 }
31
32 // 处理小数
33 while (num < 1) {
34 num *= 10;
35 exponent--;
36 }
37
38 return {num, exponent};
39 };
40
41 // 计算绝对误差
42 double err = abs(xtrue - x);
43
44 // 转换为科学计数法
45 auto [x_scaled, m] = scientific_notation(x);
46 auto [err_scaled, q] = (err != 0) ?
47 scientific_notation(err) : make_pair(0.0, 0);
48
49 // 获取误差首位数字
50 int first_err_digit = (err_scaled >= 1) ?
51 floor(err_scaled) : 0;
52
53 // 判断有效位数
54 int n;
55 if (first_err_digit < 5) {
56 n = m - q;
57 } else {
58 n = m - q - 1;
59 }
60
61 // 计算绝对误差限
62 double e = 0.5 * pow(10, m - n);
```

```
63
64 return {n, e};
65 }
66
67 int main() {
68 // 示例测试
69 auto test1 = getdigits(123.45, 123.40);
70 cout << "(" << test1.first << ", " << test1.second << ")\n"; // (4, 0.05)
71
72 auto test2 = getdigits(0.0245, 0.024);
73 cout << "(" << test2.first << ", " << test2.second << ")\n"; // (2, 0.0005)
74
75 auto test3 = getdigits(0.0, 0.0);
76 cout << "(" << test3.first << ", " << test3.second << ")\n"; // (0, 0.0)
77
78 auto test4 = getdigits(123456, 123400);
79 cout << "(" << test4.first << ", " << test4.second << ")\n"; // (4, 50.0)
80
81 return 0;
82 }
```

Listing 3: C++ 有效数字计算函数 (getdigits.cpp)