
基于 ElegantNote 的笔记模板

笔记副标题

某某某

XXXX 年 XX 月

目 录

1	第一节	1
1.1	第一小节	1
1.2	第二小节	1
2	LaTeX 常用环境	1
2.1	列表的使用	1
2.2	文献引用	2
2.3	数学公式	2
2.4	定理环境	3
2.5	算法环境	4
3	微分方程的数值方法	5
3.1	有限差分方法	5
3.2	数值格式	6
3.3	矩阵形式	6
4	插图环境	7
4.1	图的使用	7
4.2	插图示例	7
5	表格环境	8
5.1	表的使用	8
5.2	表格示例	9
6	代码高亮	10
	参考文献	11

1 第一节

1.1 第一小节

这是小四号的正文字体, 行间距 1.35 倍.

通过空一行实现段落换行, 仅仅是回车并不会产生新的段落.

自定义了一个命令 `\red{文字}` 可以加红文字, 可以在论文修改阶段方便标记.

这是一个引用的示例 [1] 和 [2–4].

这是中文与 English 混排. 标点统一使用英文的标点 (英文没有的除外). 这是一段文字这是一段文字这是一段文字这是一段文字. 这是一段文字这是一段文字这是一段文字这是一段文字这是一段文字这是一段文字这是一段文字.

1.2 第二小节

这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字这是一大段文字.

2 LaTeX 常用环境

2.1 列表的使用

这是一个计数的列表.

1. 第一项
 - (a). 第一项中的第一项
 - (b). 第一项中的第二项
2. 第二项
3. 第三项

这是一个不计数的列表.

- 第一项
 - 第一项中的第一项
 - 第一项中的第二项
- 第二项
- 第三项

2.2 文献引用

参考文献的样式符合国家标准《信息与文献参考文献著录规则》GB/T7714-2015, 论文中引用和参考的文献必须列出. 参考文献序号按所引文献在论文中出现的先后次序排列. 引用文献应在论文中的引用处加注文献序号, 并加注方括弧.

文献引用示例 [2] 和 [3, 5].

2.3 数学公式

数学公式的使用请参考公式手册 `symbols-a4`, 或者《一份 (不太) 简短的 \LaTeX 2 ϵ 介绍》(`lshort-zh-cn`).

在文中行内公式可以这么写: $a^2 + b^2 = c^2$, 这是勾股定理, 它还可以表示为 $c = \sqrt{a^2 + b^2}$, 还可以让公式单独一段并且加上编号

$$\sin^2 \theta + \cos^2 \theta = 1. \quad (1)$$

还可以通过添加标签在正文中引用公式, 如等式 (1) 或者 1.

读者可能阅读过其它手册或者资料, 知道 \LaTeX 提供了 `eqnarray` 环境. 它按照等号左边—等号—等号右边呈三列对齐, 但等号周围的空隙过大, 加上公式编号等一些 bug, 目前已不推荐使用. (摘自 `lshort-zh-cn`)

多行公式常用 `align` 环境, 公式通过 `&` 对齐. 分隔符通常放在等号左边:

$$a = b + c \quad (2)$$

$$= d + e. \quad (3)$$

`align` 环境会给每行公式都编号. 我们仍然可以用 `\notag` 或 `\nonumber` 去掉某行的编号. 在以下的例子, 为了对齐等号, 我们将分隔符放在右侧, 并且此时需要在等号后添加一对括号 `{}` 以产生正常的间距:

$$a = b + c \quad (4)$$

$$\begin{aligned}
 &= d + e + f + g + h + i + j \\
 &\quad + m + n + o
 \end{aligned} \tag{5}$$

$$= p + q + r + s. \tag{6}$$

如果我们不需要按等号对齐, 只需罗列数个公式, `gather` 将是一个很好用的环境:

$$a = b + c \tag{7}$$

$$d = e + f + g \tag{8}$$

$$h + i = j + k$$

$$l + m = n \tag{9}$$

`align` 和 `gather` 有对应的不带编号的版本 `align*` 和 `gather*`. 对于 `align`、`gather`、`align*` 与 `gather*` 等环境, 若添加命令 `\allowdisplaybreaks` 后 (已添加), 公式可以跨页显示.

多个公式组在一起公用一个编号, 编号位于公式的居中位置, `amsmath` 宏包提供了诸如 `aligned`、`gathered` 等环境, 与 `equation` 环境套用.

这个公式使用 `aligned` 环境 (推荐使用)

$$\begin{cases} -\frac{d^2u}{dx^2} + \frac{du}{dx} = \pi^2 \sin(\pi x) + \pi \cos(\pi x), & x \in [0, 1], \\ u(0) = 0, & u(1) = 0. \end{cases} \tag{10}$$

其中方程的解析解为 $u = \sin(\pi x)$.

这个公式使用 `array` 环境

$$\begin{cases} -\frac{d^2u}{dx^2} + \frac{du}{dx} = \pi^2 \sin(\pi x) + \pi \cos(\pi x), & x \in [0, 1], \\ u(0) = 0, & u(1) = 0. \end{cases} \tag{11}$$

`aligned` 与 `equation` 环境套用, 公式间距自动调节, 如果有分式, 分式也是行间显示. 如果用 `array` 与 `equation` 环境套用, 需要手动调整公式行间距和行间显示.

2.4 定理环境

定义 2.1 这是一个定义.

命题 2.1 这是一个命题.

引理 2.2 (Lemma) 这是一个引理.

定理 2.3 (Theorem) 这是一个定理.

证明. 这是证明环境.

推论 这是一个推论.

命题 2.4 (Proposition) 这是一个命题.

引理 2.5 (参考文献 [2]) 假设单步法具有 p 阶精度, 且增量函数 $\varphi(x_n, u_n, h)$ 关于 u 满足 Lipschitz 条件

$$|\varphi(x, u, h) - \varphi(x, \bar{u}, h)| \leq L_\varphi |u - \bar{u}|. \quad (12)$$

定理 2.6 假设单步法具有 p 阶精度, 且增量函数 $\varphi(x_n, u_n, h)$ 关于 u 满足 Lipschitz 条件

$$|\varphi(x, u, h) - \varphi(x, \bar{u}, h)| \leq L_\varphi |u - \bar{u}|. \quad (13)$$

证明 由定理 2.5 和 (10) 式可以推出以上结论.

推论 假设单步法具有 p 阶精度, 且增量函数 $\varphi(x_n, u_n, h)$ 关于 u 满足 Lipschitz 条件

$$|\varphi(x, u, h) - \varphi(x, \bar{u}, h)| \leq L_\varphi |u - \bar{u}|. \quad (14)$$

评论 这是一个 remark.

例 2.1 这是一个例子.

2.5 算法环境

如下是算法 1.

算法 1 Euclid's algorithm

1: procedure EUCLID(a, b)	▷ The g.c.d. of a and b
2: $r \leftarrow a \bmod b$	
3: while $r \neq 0$ do	▷ We have the answer if r is 0
4: $a \leftarrow b$	
5: $b \leftarrow r$	
6: $r \leftarrow a \bmod b$	
7: end while	
8: return b	▷ The gcd is b
9: end procedure	

如下是算法 2, 算法宽度可以通过 minipage 宏包调节.

算法 2 算法的名字

输入: input parameters A, B, C

输出: output result

```

1: some description 算法介绍
2: for condition do
3:   ...
4:   if condition then
5:     ...
6:   else
7:     ...
8:   end if
9: end for
10: while condition do
11:   ...
12: end while
13: return result

```

3 微分方程的数值方法

本章我们考虑具有以下微分方程:

$$\begin{cases} Lu = -\frac{d^2u}{dx^2} + \frac{du}{dx} + qu = f, & a < x < b, \\ u(a) = \alpha, \quad u(b) = \beta. \end{cases} \quad (15)$$

其中 q, f 为 $[a, b]$ 上的连续函数, $q \geq 0$; α, β 为给定常数. 这是最简单的椭圆方程第一边值问题.

问题 (15) 存在唯一解 (引用示例参考文献 [2]).

3.1 有限差分方法

在偏微分方程的数值解法中, 有限差分法数学概念直观, 推导自然, 是发展较早且比较成熟的数值方法. 由于计算机只能存储有限个数据和做有限次运算, 所以任何一种用计算机解题的方法, 都必须把连续问题 (微分方程的边值问题、初值问题等) 离散化, 最终化成有限形式的线性代数方程组.

3.2 数值格式

将区间 $[a, b]$ 分成 N 等分, 分点为

$$x_i = a + ih, \quad i = 0, 1, \dots, N,$$

其中 $h = (b - a)/N$. 于是我们得到区间 $I = [a, b]$ 的一个网格剖分. x_i 称为网格的节点, h 称为步长.

数值格式:

$$L_h u_i = -\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \frac{u_{i+1} - u_{i-1}}{h} + q_i u_i = f_i, \quad 1 \leq i \leq N-1.$$

其中 $q_i = q(x_i)$, $f_i = f(x_i)$.

以上差分方程对于 $i = 1, 2, \dots, N-1$ 都成立, 加上边值条件 $u_0 = \alpha, u_N = \beta$, 就得到关于 u_i 的差分格式:

$$\begin{cases} L_h u_i = -\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + \frac{u_{i+1} - u_{i-1}}{2h} + q_i u_i = f_i, & i = 1, 2, \dots, N-1, \\ u_0 = \alpha, \quad u_N = \beta. \end{cases} \quad (16)$$

它的解 u_i 是 $u(x)$ 在 $x = x_i$ 处的差分解.

3.3 矩阵形式

先定义向量 \mathbf{u} :

$$\mathbf{u} = (u_1, u_2, \dots, u_{N-1})^T.$$

差分格式可以写为矩阵形式:

$$\mathbf{A}\mathbf{u} = \mathbf{f}.$$

其中矩阵 \mathbf{A} 、向量 \mathbf{f} 的定义如下, 注意向量 \mathbf{f} 的首尾元素已包含了 $x = a$ 和 $x = b$ 处的边界条件.

$$\mathbf{A} = \begin{bmatrix} \frac{2}{h^2} + q_1 & \frac{1}{2h} - \frac{1}{h^2} & & & \\ -\frac{1}{2h} - \frac{1}{h^2} & \frac{2}{h^2} + q_2 & \frac{1}{2h} - \frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & & -\frac{1}{2h} - \frac{1}{h^2} & \frac{2}{h^2} + q_{N-2} & \frac{1}{2h} - \frac{1}{h^2} \\ & & & & -\frac{1}{2h} - \frac{1}{h^2} & \frac{2}{h^2} + q_{N-1} \end{bmatrix}. \quad (17)$$

上一个矩阵用了 `bmatrix` 环境, 也可以使用 `array` 环境.

$$A = \begin{bmatrix} \frac{2}{h^2} + q_1 & \frac{1}{2h} - \frac{1}{h^2} & & & \\ -\frac{1}{2h} - \frac{1}{h^2} & \frac{2}{h^2} + q_2 & \frac{1}{2h} - \frac{1}{h^2} & & \\ & \ddots & \ddots & \ddots & \\ & & -\frac{1}{2h} - \frac{1}{h^2} & \frac{2}{h^2} + q_{N-2} & \frac{1}{2h} - \frac{1}{h^2} \\ & & & -\frac{1}{2h} - \frac{1}{h^2} & \frac{2}{h^2} + q_{N-1} \end{bmatrix}. \quad (18)$$

4 插图环境

4.1 图的使用

XeLaTeX 环境下可以使用 EPS、PDF、PNG、JPEG、BMP 格式的图片, 当然也可以用绘图包直接在 \LaTeX 中绘制图形, 推荐使用宏包 `tikz`. 值得注意的是 `figure` 环境一个浮动体环境, LaTeX 不总是浮动体放在你想要的地方, 但是 LaTeX 总是保证浮动体的相对顺序, 所以对图片 `\label` 和 `\ref` 的交叉引用就显得尤为重要。

4.2 插图示例

插入一个图形并居中放置, 如图 1.

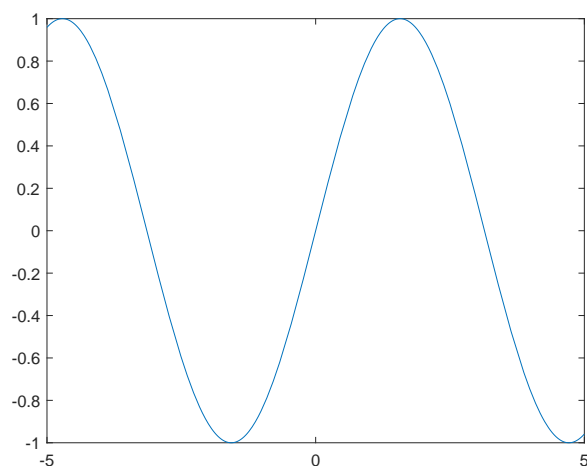


图 1: 函数 $y = \sin(x)$ 的图像

两个图左右并排放置, 共用一个标题, 如图 2.

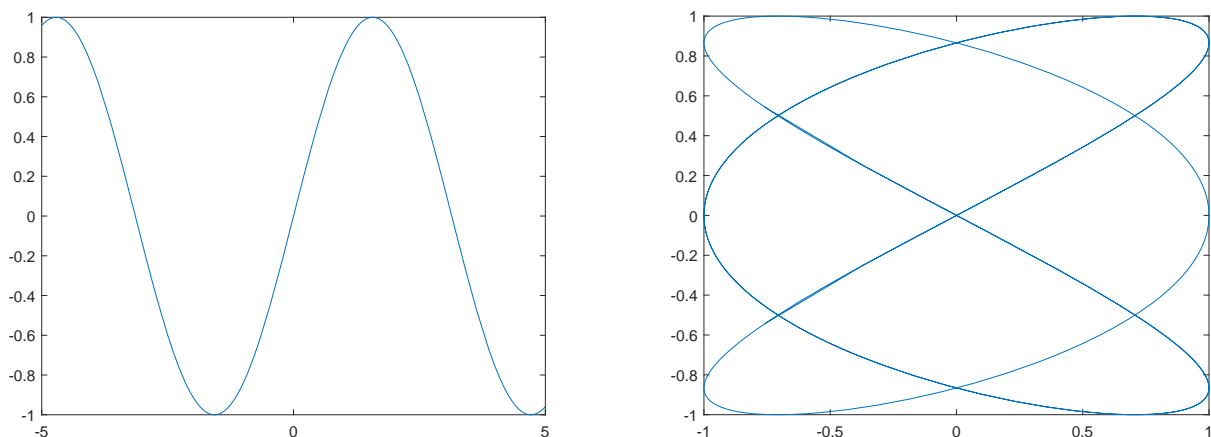


图 2: 左: 图一的描述; 右: 图二的描述.

使用 `minipage` 排版并排插图, 每个图都有单独的标题. 通过 `autoref` 引用图片: 图 3 与 图 4.

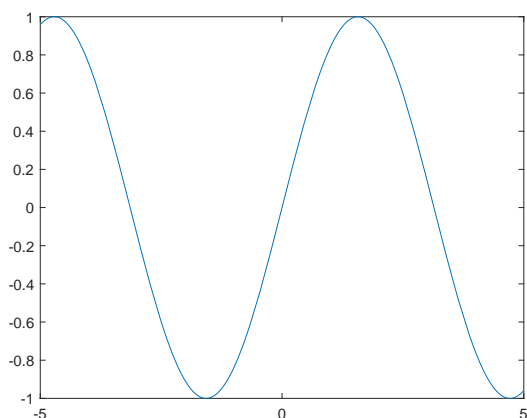


图 3: 图一的描述.

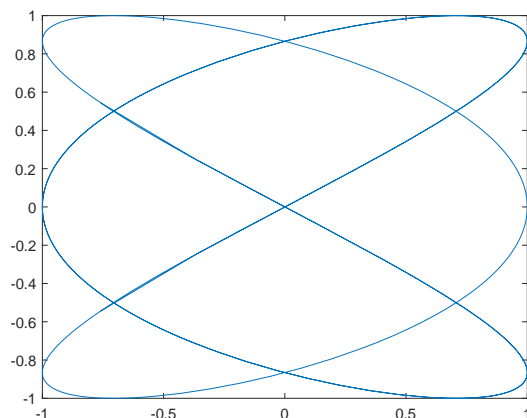


图 4: 图二的描述.

5 表格环境

5.1 表的使用

LaTeX 的 `Table` 环境是一个浮动体环境, 浮动体排版与 `Figure` 环境类似. 作为论文, 推荐使用三线表进行排版. 所谓三线表, 即在标题前有横线, 标题后有横线, 表格最后还有横线, 其他地方无线. 当然这不是死规定, 也可以根据需要在合适的地方加线.

本文基于 `tabularx` 宏包定义了新的左中右 (LCR) 格式, LCR 三个格式会根据表格宽度的设定自行控制宽度, 且其宽度相等, 方便设置和页面相同宽度的表格. 本文还定义

了命令 `P{}`, 它可以设定某一列宽度 (如 `P{1cm}` 控制某一列的宽度为 1cm), 实际上 `P{}` 命令是在 `p{}` 命令的基础上增加了居中功能.

5.2 表格示例

如下表格: 表 1. 通过 `autoref` 引用表格: 表 1.

表 1: 某校学生升高体重样本.

序号	年龄	身高	体重
001	14	156	42
002	16	158	45
003	14	162	48
004	15	163	50
平均	15	159.75	46.25

基于 `tabularx` 环境设置一些格式, 如表 2.

表 2: 数值误差示例.

N	A	B	C	D	E	F
2	9.20E-05	9.90E-05	1.00E-06	8.00E-06	1.50E-05	6.70E-05
4	9.80E-05	8.00E-05	7.00E-06	1.40E-05	1.60E-05	7.30E-05
6	4.00E-06	8.10E-05	8.80E-05	2.00E-05	2.20E-05	5.40E-05
8	8.50E-05	8.70E-05	1.90E-05	2.10E-05	3.00E-06	6.00E-05
10	8.60E-05	9.30E-05	2.50E-05	2.00E-06	9.00E-06	6.10E-05
12	1.70E-05	2.40E-05	7.60E-05	8.30E-05	9.00E-05	4.20E-05
14	2.30E-05	5.00E-06	8.20E-05	8.90E-05	9.10E-05	4.80E-05
16	7.90E-05	6.00E-06	1.30E-05	9.50E-05	9.70E-05	2.90E-05
18	1.00E-05	1.20E-05	9.40E-05	9.60E-05	7.80E-05	3.50E-05
20	1.10E-05	1.80E-05	1.10E-04	7.70E-05	8.40E-05	3.60E-05

6 代码高亮

MATLAB 代码高亮

MATLAB code

```

1 % Euler method for the ODE model
2 % u'(x)=x^2+x-u, x in [0,1]
3 % Initial condition: u(0)=0 ;
4 % Exact solution: u(x)=-exp(-x)+x^2-x+1.
5 clear all; clf
6 h=0.1;
7 x=0:h:1;
8 N=length(x)-1;
9 u(1)=0; % initial value
10 fun=@(t,u) t.^2+t-u; % RHS
11 for n=1:N
12     u(n+1)=u(n)+h.*fun(x(n),u(n));
13 end
14 ue=-exp(-x)+x.^2-x+1; % exact solution
15 plot(x,ue,'b-',x,u,'r+', 'LineWidth',1)
16 legend('Exact','Numerical','location','North')
17 xlabel('x'), ylabel('u')

```

Python 代码高亮

Python code

```

1 #PythonDraw.py
2 import turtle as t
3 t.setup(650, 350, 200, 200)
4 t.penup()
5 t.fd(-250)
6 t.pendown()
7 t.pensize(25)
8 t.pencolor("purple color")
9 t.seth(-40)
10 for i in range(4):
11     t.circle(40, 80)
12     t.circle(-40, 80)
13 t.circle(40, 80/2)
14 t.fd(40)
15 t.circle(16, 180)
16 t.fd(40 * 2/3)
17 t.done()

```

参考文献

- [1] Tadmor E. A review of numerical methods for nonlinear partial differential equations[J]. Bull. Amer. Math. Soc., 2012, 49(4):507-554.
- [2] 李荣华, 刘播. 微分方程数值解法[M]. 第四版. 北京: 高等教育出版社, 2009.
- [3] Adams R A, Fournier J J F. Sobolev spaces[M]. 2nd ed. Amsterdam: Elsevier, 2003.
- [4] Trefethen L N, Weideman J A C. The exponentially convergent trapezoidal rule[J]. SIAM Rev., 2014, 56(3):385-458.
- [5] Shen J. Efficient spectral-Galerkin method I. Direct solvers of second- and fourth-order equations using Legendre polynomials[J]. SIAM J. Sci. Comput., 1994, 15(6):1489-1505.