

# 基于 T<sub>E</sub>X 的数学建模论文排版

## 摘要

问题背景.....

**针对问题一：**这是一个中文段落，用于测试 LaTeX 中的中文排版。希望这段文字能展示出正确的格式和样式。这里我们会继续添加一些随机的内容，以便更好地展示效果。我们可以使用许多不同的句子来填充段落，使其看起来更像真正的文本。最后，再加上一些补充说明，确保段落长度适中。

**针对问题二：**

**针对问题三：**

**针对问题四：**

**关键词：** 关键词 1   图片   表格   公式

## 一、问题重述

### 1.1 问题背景

这是一个中文段落，用于测试 LaTeX 中的中文排版。希望这段文字能展示出正确的格式和样式。这里我们会继续添加一些随机的内容，以便更好地展示效果。我们可以使用许多不同的句子来填充段落，使其看起来更像真正的文本。最后，再加上一些补充说明，确保段落长度适中。

这是一个中文段落，用于测试 LaTeX 中的中文排版。希望这段文字能展示出正确的格式和样式。这里我们会继续添加一些随机的内容，以便更好地展示效果。

### 1.2 问题提出

这是怎么回事

1. 这是调整后的第一点
  1. 这是调整后的第二点
  1. 这是调整后的第三点
- 
1. 这是第一点
  2. 这是第二点
  3. 这是第三点

## 二、问题分析

这是一个中文段落，用于测试 LaTeX 中的中文排版。希望这段文字能展示出正确的格式和样式。这里我们会继续添加一些随机的内容，以便更好地展示效果。我们可以使用许多不同的句子来填充段落，使其看起来更像真正的文本。最后，再加上一些补充说明，确保段落长度适中。

## 三、模型假设

这是一个中文段落，用于测试 LaTeX 中的中文排版。希望这段文字能展示出正确的格式和样式。这里我们会继续添加一些随机的内容，以便更好地展示效果。我们可以使用许多不同的句子来填充段落，使其看起来更像真正的文本。最后，再加上一些补充说明，确保段落长度适中。

## 四、符号说明

符号	说明
$a$	加速度
$a$	加速度
$a$	加速度
$a$	加速度

## 五、模型的建立与求解

### 5.1 问题一模型的建立和求解

#### 5.1.1 问题一模型建立

#### 5.1.2 问题一结果

#### 5.1.3 模型检验

如果想强调部分内容,可以使用加粗的手段来实现。加粗字体可以用 `\textbf{加粗}` 来实现。例如: **这是加粗的字体。This is bold fonts**。

### 5.2 问题二模型的建立和求解

### 5.3 问题三模型的建立和求解

## 六、模型检验与误差分析

这是一个中文段落,用于测试 LaTeX 中的中文排版。希望这段文字能展示出正确的格式和样式。这里我们会继续添加一些随机的内容,以便更好地展示效果。我们可以使用许多不同的句子来填充段落,使其看起来更像真正的文本。最后,再加上一些补充说明,确保段落长度适中。

## 七、模型的评价、改进与推广

### 7.1 模型的优点

中文字体没有斜体设计,但是英文字体有。斜体 *Italics*。

引用样式 [?] 引用样式<sup>[?]</sup> 引用样式<sup>[?]</sup>



## 附录 A 支撑材料列表

- /code: 包括问题的求解源代码，均为 python 文件。
  - Q1.py: 对问题一的求解代码。
  - Q2.py: 对问题二的求解代码。
  - Q3.py: 对问题三的求解代码。
  - Q4.py: 对问题四的求解代码。
- /data: 包括对题目所给的数据文件进行处理后的文件，同时设置为相对路径以用来代码运行时调用。
- /img: 包括解决问题中出现的图片文件，部分显示在论文中。

## 附录 B 附加代码

### python 代码

```
1 def floyd_algorithm(graph, start, end):
2     """
3     Floyd算法求解最短路径
4     :param graph: 邻接矩阵，表示各点之间的距离。如果不连通，则相应权重为无穷大（例如
5         1000）
6     :param start: 起点
7     :param end: 终点
8     :return: 最短路径长度，最短路径
9     """
10    inf = 0x3f3f3f3f
11    n = len(graph) # 图中节点的个数
12    distances = {node: {n: inf for n in range(n)} \
13        for node in range(n)} # 初始化距离矩阵
14    path = {node: {n: [] for n in range(n)} \
15        for node in range(n)} # 初始化路径矩阵
16
17    # 将graph的值复制到distances中，并初始化path矩阵
18    for i in range(n):
19        for j in range(n):
20            distances[i][j] = graph[i][j]
21            if i != j and graph[i][j] != inf:
22                path[i][j].append(i)
```

```

22         path[i][j].append(j)
23
24     # Floyd算法核心代码
25     for k in range(n):
26         for i in range(n):
27             for j in range(n):
28                 new_distance = distances[i][k] + distances[k][j]
29                 if new_distance < distances[i][j]:
30                     distances[i][j] = new_distance
31                     path[i][j] = path[i][k] + path[k][j][1:]
32
33     # 返回最短路径长度和最短路径
34     return distances[start][end], path[start][end]

```

## matlab 代码

```

1  clear; clc, close all;
2
3
4  data = [data; data + change];
5  figure(1)
6  n = size(data, 1);
7  % forma_data = data ./ repmat(sum(data .* data).^0.5, n, 1);
8  forma_data = zscore(data);
9  distance = pdist(forma_data);
10 square_distance = squareform(distance);
11 Tree = linkage(distance);
12 dendrogram(Tree);
13 ans = cluster(Tree, 3);
14
15 figure(2)
16 x = 1:M;
17 plot(x, ans(1:M), 'g-h', 'LineWidth', 2);
18 hold on;
19 plot(x, ans(M + 1:2 * M), 'r-x', 'LineWidth', 2);

```

## 八、临时展示效果

这里是一些插图展示，插个图先

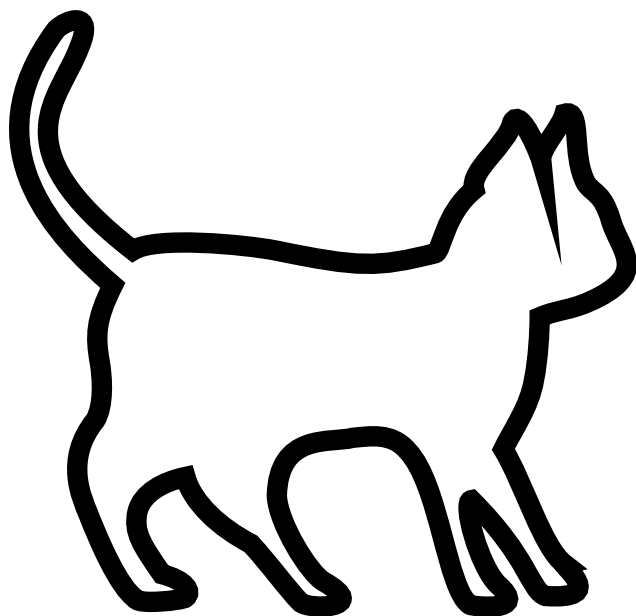
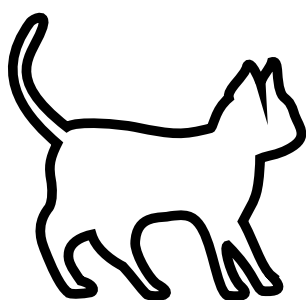
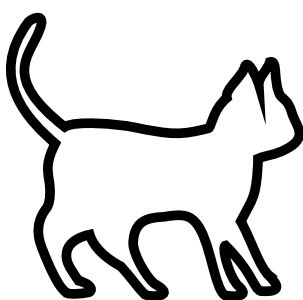


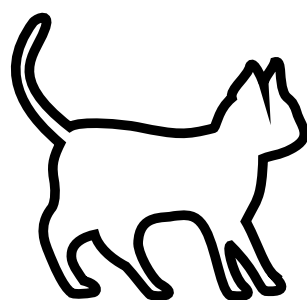
图 1 这是一张图片



(a) 流程图



(b) 流程图



(c) 流程图

图 2 多图并排示例

这相当于整体是一张大图片，大图片引用是??，子图引用别分是??、??、??。

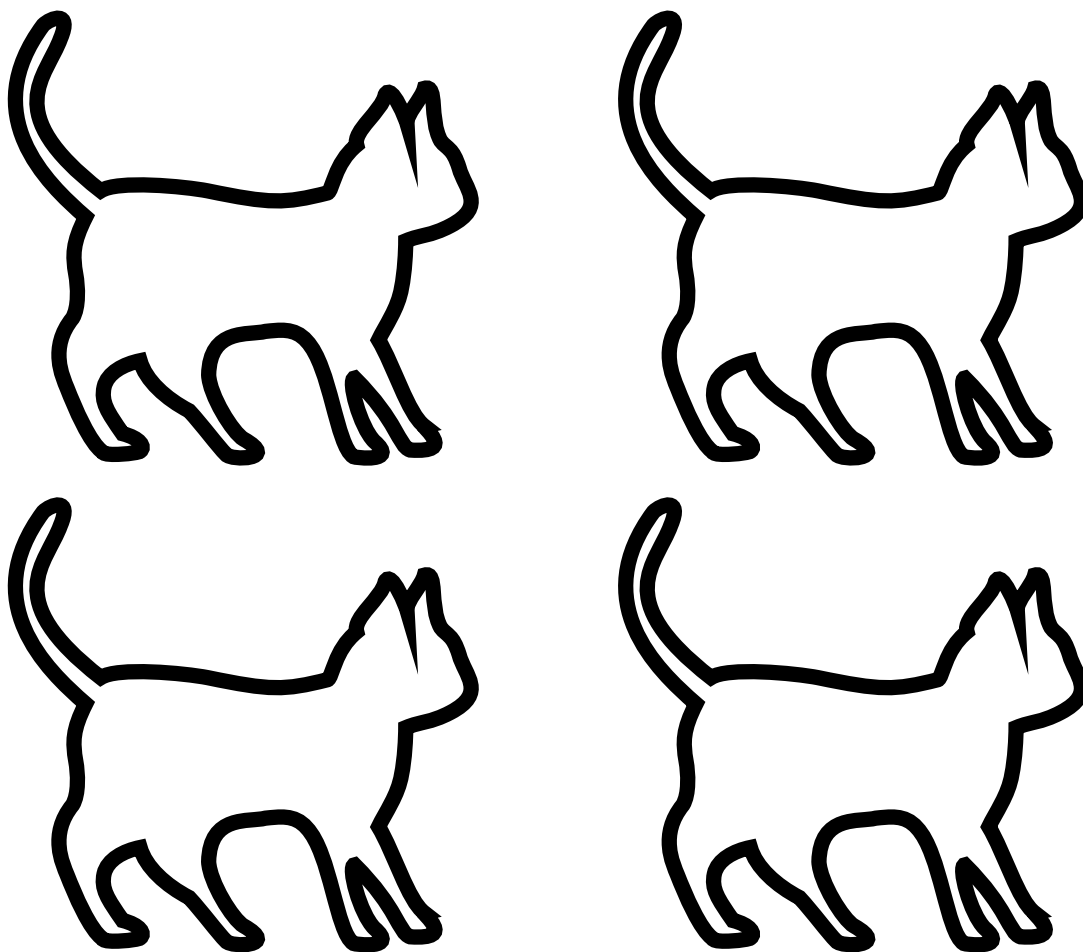


图3 演示4排图片

表1 这是个表

这是三线表	nb	???
这是一个	这是另一个	使用三线表

```

\begin{table}[H]
\caption[标签名]{中文标题}
\begin{tabular}{cc...c}
\toprule[1.5pt]
表头第1个格 & 表头第2个格 & ... & 表头第n个格 & \\
\midrule[1pt]
表中数据(1,1) & 表中数据(1,2) & ... & 表中数据(1,n) & \\
表中数据(2,1) & 表中数据(2,2) & ... & 表中数据(2,n) & \\
..... & & & & \\
表中数据(m,1) & 表中数据(m,2) & ... & 表中数据(m,n) & \\
\end{tabular}
\end{table}

```



```

\bottomrule[1.5pt]
\end{tabular}
\end{table}

```

1. 1
2. f
- 3.

**表 2 网站生成图表，点击进入**

标准角度与实际的差	倾斜角度	标准角度与实际的差	倾斜角度

### 使用矩阵排版

1	0.3420047	0.3339529	0.3381766	0.3362277	0.3547872	0.3355143
0.3374153	0.4495615	0.3652695	0.3334384	0.334335	0.333648433	0.3340383

### 下面是多排表格

**表 3 这是多排表格的标题**

这是表格			非常不错		
good	1.000	0.080	good	1.000	0.080
nice	0.080	1.000	nice	0.080	1.000

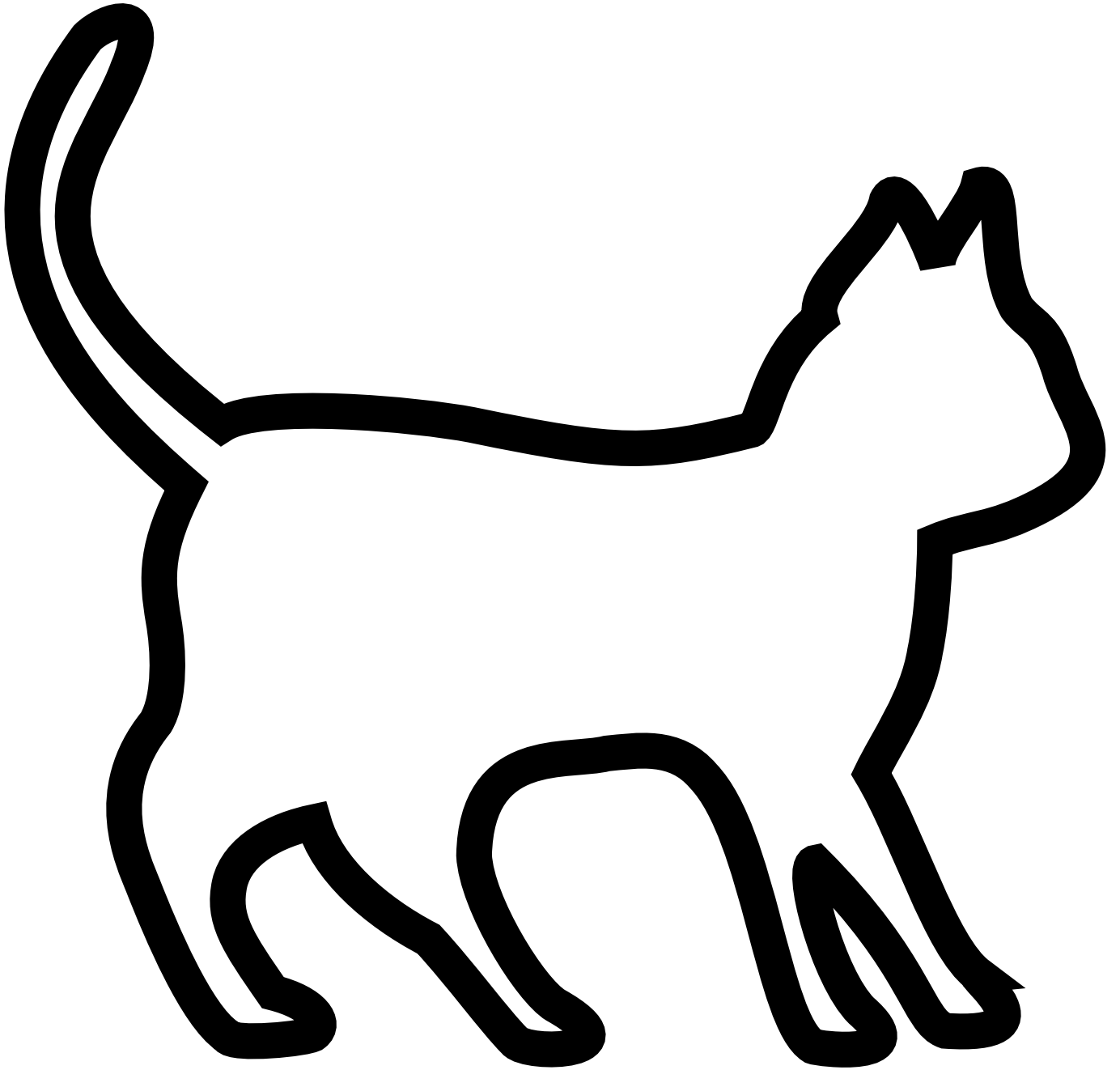


图 4 这是测试，这里可以插入较大图片

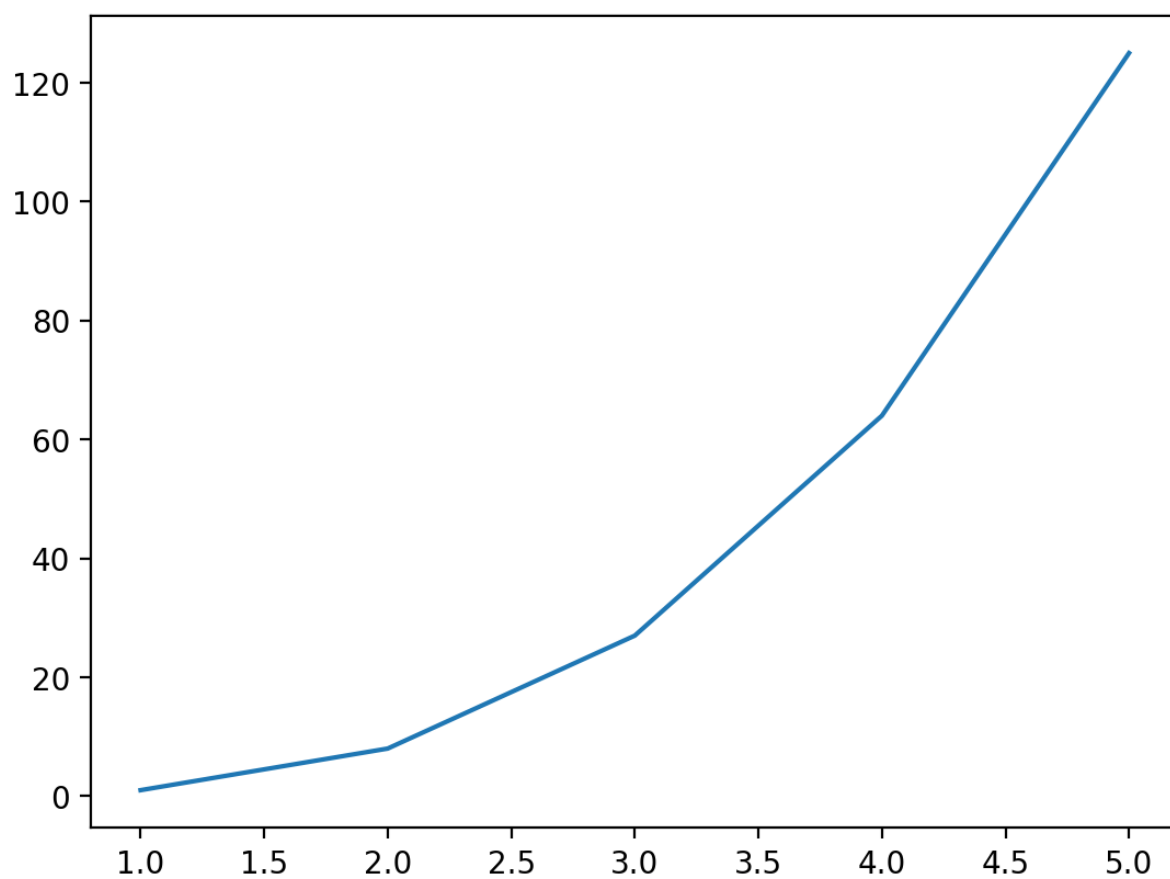


图5 插入图片