

西北工业大学

# 航天飞行动力学课程设计

题目： 基于 Python 语言的  
一级半运载火箭弹道计算

作 者： 李宗霖

学 号： 2021300045

专 业： 飞行器动力工程

学 院： 教育实验学院

指导教师： 闫晓东

2024 年 3 月

## 摘 要

这是在西北工业大学本科毕业设计、硕博研究生毕业论文格式的要求下的一份 LaTeX 文档类模板。使用者无需额外修改格式控制细节，直接在所发布的样例基础上，修改章节标题，撰写内容，即可完成毕业设计论文任务。

**关键词：**学位论文；Python； $\text{\LaTeX}$

# 目 录

摘 要.....	I
第 1 章 绪论.....	1
1.1 这是中标题 .....	1
1.1.1 这是小标题.....	1
1.2 公式 .....	1
1.3 特殊符号.....	1
1.4 参考文献的引用 .....	1
1.5 标点符号的选择 .....	1
1.6 萌新如何编译 .....	2
1.7 如何生成盲评版本 .....	2
1.8 如何生成学位论文评阅人和答辩委员会名单.....	2
第 2 章 插入图表以及如何引用 .....	4
2.1 表格 .....	4
2.2 插图 .....	4
2.3 插入源代码 .....	4
2.4 引用以及其他编写建议 .....	5
参考文献.....	5
附录 A 一份说明顺便测试英文标题 Title .....	6
附录 B 另一份说明 .....	7
附录 C title .....	8
致 谢.....	15

## 第 1 章 绪论

### 1.1 这是中标题

emmmm

#### 1.1.1 这是小标题

emmmmm

##### (1) 这是小小标题

搞这么多层大丈夫?

### 1.2 公式

简单行内公式  $a + b = 233$ , 超高公式会被压缩  $\frac{1}{2} = 0.5$  或者使用 `\displaystyle` 防止被压缩:  $\frac{1}{2} = 0.5$ 。

简单的不标号单行公式

$$a_0 + a_1 + a_2 = \sqrt{233}$$

需要标号和起名的公式如式 (1-1) 所示。测试下 `autoref` 式 (1-1)

$$a_0 + a_1 + a_2 = \sqrt{233} \quad (1-1)$$

### 1.3 特殊符号

用 <http://detexify.kirelabs.org/classify.html>画出来。

### 1.4 参考文献的引用

$\text{\LaTeX}$  中要求参考文献使用 `\cite` 进行参考引用, 若论文要求中说明需在文字的右上角注明引用, 请使用命令 `\cite` 进行参考引用。举个不恰当的例子, 比如本论文模板的原版 “`LaTeX-Template-For-NPU-Thesis`”<sup>[?]</sup> 要求务必声明引用, 同时预配置了插件 “`math-symbols`”<sup>[?]</sup>。对组件的引用是每一名科学工作者的基本素养 (一本正经)。对于需要引用但是并不需要明确指明引用位置的文献, 请使用 `\nocite` 命令。

在此同时感谢真正的 dalao 高德纳开发了全世界版本号最接近  $\pi$  的软件  $\text{\LaTeX}$ <sup>[?]1]</sup>。

测试引用文献<sup>[?][?][?]</sup>。其中倒数第二篇为中文文献, 最后一篇为会议文献。

### 1.5 标点符号的选择

根据《中华人民共和国国家标准 GB/T 15834-1995》及《出版工作中的语言文字规范》中提及, “科学技术中文图书, 如果涉及公式、算式较多, 句号可以统一用英文句号 ‘.’, 省略号用英文三个点的省略号 ‘...’”。如果您是中文的科技论文写作者, 建议您使用全角英文句号 “.” 间隔句子。如果是人文学科则可以不进行处理。您也可以在一开始先使用中文句号 ‘。’, 最后批量替换即可。

## 1.6 萌新如何编译

1. 安装正确版本的 TexLive 2021
2. 使用自带的 TeXworks 打开 `yanputhesis-sample.tex`
3. 左上角下拉框选择工具
4. 依次使用 XeLaTeX-BibTeX-XeLaTeX-XeLaTeX 编译

## 1.7 如何生成盲评版本

1. 在这份样例当中，已经将标题页可能用到的作者姓名、导师姓名添加了空白盲评标记 `\blindreview{text}`。如果需要生成盲评版本，则需要将文档类型设置为 `blindreview=true`，这样便可得到标题页不含作者与导师姓名的版本。
2. 在致谢中，除了导师名字之外，其他老师、同学的名字也应当隐去。同样可以将姓名添加空白盲评标记 `\blindreview{text}` 来得到留空版本的结果。
3. 一般正文中不建议出现留空，因此推荐另外两种盲评标记，涂黑或者打星。使用 `\blackbox{text}` 命令将姓名添加涂黑盲评标记，文本会替换为与文字相同长度的黑色方块，制造涂黑效果。或者使用 `\markname{text}` 命令将姓名添加打星盲评标记，姓名将替换成 3 个星号 “\*\*\*”。
4. 下面给出示例（通过开启盲评选项查看效果）：
  - (a) 不添加任何盲评标记：“感谢某某某教授的悉心指导。”
  - (b) 使用了空白盲评标记：“感谢某某某教授的悉心指导。”
  - (c) 使用了涂黑盲评标记：“感谢某某某教授的悉心指导。”
  - (d) 使用了打星盲评标记：“感谢某某某教授的悉心指导。”

## 1.8 如何生成学位论文评阅人和答辩委员会名单

1. 在这份样例当中，已经预设置了盲评学位论文评阅人和答辩委员会名单，实现代码可参考代码片段 C.1 所示，明审版本可参考 代码片段 1.2 所示。
2. 在学位论文评阅人名单中分为两种情况，即盲评与明审。请根据自身情况填写评委信息。如果是盲评，使用命令 `\fullBlindReview{num}` 来生成盲评表格，其中参数 `num` 表示盲评专家人数，一般是 3 或 5 人。如果是明审，使用命令 `\expert{name}{title}{university}` 登记评委信息，其中参数 `name`、`title`、`university` 分别为专家的姓名、职称、学校。
3. 答辩委员会需登记四个信息：答辩时间、答辩主席、答辩评委以及答辩秘书。其中，答辩时间为 `\committee` 命令后的第一个参数，其余分别使用 `\defenseChair`、`\committeeMember`、`\defenseSecretary` 命令登记专家个人信息，用法与 `\expert` 命令一致。

代码片段 1.1 盲评样例 `makeBlindReviewerCommitteePage.tex`

```
1 \makeCommitteePage{
2   \reviewers{\fullBlindReview{5}}
```

```

3 \committee{2023 年 x 月 y 日}{
4 \defenseChair{赵钱孙}{教授}{西北工业大学}
5 \committeeMember{周吴郑}{教授}{西北工业大学}
6 \committeeMember{冯陈褚}{教授}{西北工业大学}
7 \committeeMember{蒋沈韩}{教授}{西北工业大学}
8 \committeeMember{朱秦尤}{教授}{西北工业大学}
9 \committeeMember{何吕施}{教授}{西北工业大学}
10 \committeeMember{孔曹严}{教授}{西北工业大学}
11 \defenseSecretary{金魏陶}{教授}{西北工业大学}
12 }
13 }

```

代码片段 1.2 明审样例 makeOpenReviewerCommitteePage.tex

```

1 \makeCommitteePage{
2 \reviewers{
3 \expert{周吴郑}{教授}{西北工业大学}
4 \expert{冯陈褚}{教授}{西北工业大学}
5 \expert{蒋沈韩}{教授}{西北工业大学}
6 \expert{朱秦尤}{教授}{西北工业大学}
7 \expert{何吕施}{教授}{西北工业大学}
8 }
9 \committee{2023 年 x 月 y 日}{
10 \defenseChair{赵钱孙}{教授}{西北工业大学}
11 \committeeMember{周吴郑}{教授}{西北工业大学}
12 \committeeMember{冯陈褚}{教授}{西北工业大学}
13 \committeeMember{蒋沈韩}{教授}{西北工业大学}
14 \committeeMember{朱秦尤}{教授}{西北工业大学}
15 \committeeMember{何吕施}{教授}{西北工业大学}
16 \committeeMember{孔曹严}{教授}{西北工业大学}
17 \defenseSecretary{金魏陶}{教授}{西北工业大学}
18 }
19 }

```

## 第 2 章 插入图表以及如何引用

### 2.1 表格

使用 <http://www.tablesgenerator.com/> 生成，可粘贴 Excel。效果如表2-1所示。注意表中的字号（五号）和表格宽度（通栏）。外部请用 `table` 环境，内部使用 `tabularx` 即可。

表 2-1 表格标题

$A$	$B$	$A + B$	$A \times B$
1	6	7	6
2	7	9	14
3	8	11	24
4	9	13	36
5	10	15	50

表 2-2 指定宽度与对齐方式

2 cm	3 cm	4 cm	Other
1	6	7	1
2	7	9	2
3	8	11	3

### 2.2 插图

请直接使用 `figure` 环境，内部使用 `includegraphics` 即可。如果需要多张子图排版，请在 `figure` 环境内部使用 `minipage` 预先设置总的浮动体宽度，然后再使用 `subfigure` 环境进行排版。

测试下文章内的图片引用。如图 2-1 和图 2-2 所示，这是两幅插图。在这其中图 2-2(a) 是第一幅子图，图 2-2(b) 是第二幅子图。

### 2.3 插入源代码

这里给出一个 Hello World 的样例，如代码片段 2.1 所示。

代码片段 2.1 Hello World.cpp

```

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     // output "Hello World!"
7     cout << "Hello World!" << endl;
8     return 0;
9 }

```

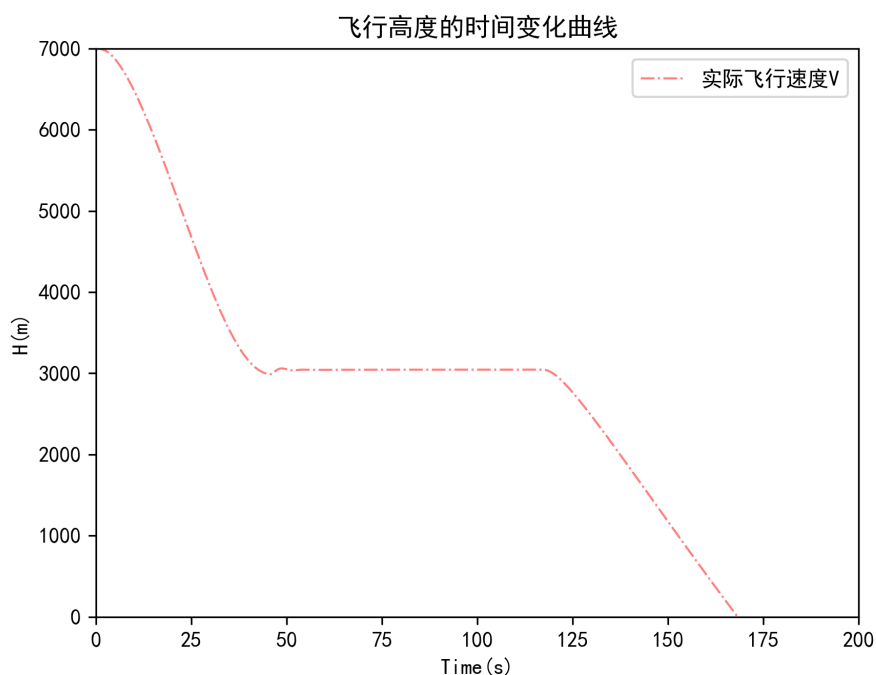


图 2-1 这里是个普通的标题

## 2.4 引用以及其他编写建议

$\text{\LaTeX}$  提供了 `ref` 和 `autoref` 两种引用方式，其中前者只显示序号，后者可以显示提示语，如“代码片段 2.1”表示引用代码，而“图 2-2(b)”表示引用图片的子图. 为了方便引用以及作者阅读，本人强烈建议使用 `autoref` 来统一处理引用问题，同时在每一个 `autoref` 添加提示语，如 `fig` 和 `tab` 分别表示插图和表格。

由于  $\text{\XeLaTeX}$  在处理中文时，会自动在中文之间添加空格，所以请放心地在编写文档时换行，防止某一行过长导致阅读时的不便。另外中英文之间的空格（包括命令）并未做严格限制。本文推荐除在不影响最终成文的结果这一前提下，为保持文档的美观与易读，请自行选择合适的编写方式。

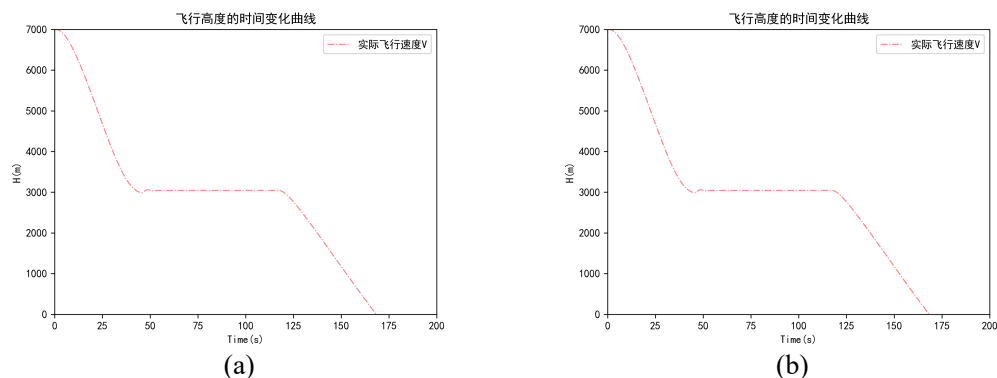


图 2-2 这里是另一个普通的标题



## 附录 A 一份说明顺便测试英文标题 Title

强烈不推荐英文标题！仅供测试，擅自使用后果自负。

### A.1 测试附录子标题

这是一份附录，请放置一些独立的证明、源代码、或其他辅助资料。

$$C = 2\pi r \quad (\text{a-1})$$

$$S = \pi r^2 \quad (\text{a-2})$$

## 附录 B 另一份说明

这是另一份附录，请放置一些独立的证明、源代码、或其他辅助资料。

$$S_{\text{sphere}} = 4\pi r^2 \tag{b-1}$$

$$V_{\text{sphere}} = \frac{4}{3}\pi r^3 \tag{b-2}$$

## 附录 C title

代码片段 C.1 代码 main.py

```

1 """
2     一级半运载火箭弹道计算
3 """
4
5
6 ##### 环境准备 #####
7 import numpy as np
8 from matplotlib import pyplot as plt
9 from scipy.interpolate import interp2d, interp1d, CloughTocher2DInterpolator
10
11 # 展示高清图
12 from matplotlib_inline import backend_inline
13 backend_inline.set_matplotlib_formats('svg')
14
15 plt.rcParams['font.sans-serif'] = ['SimHei']
16 plt.rcParams['axes.unicode_minus'] = False
17
18 # 忽略提示
19 import warnings
20 warnings.filterwarnings("ignore", category=DeprecationWarning)
21
22
23 ##### 定义数值 #####
24 # 导弹参数
25 S_ref = 19.6
26
27 ## 质量
28 # 构型: 芯一级+2个大助推器+2个小助推器
29 m_0 = 645000 # 总质量
30 m0 = 6600 # 有效载荷质量
31
32 # 芯一级
33 m1_0 = 173000 # 芯一级起飞质量
34 m1_1 = 19500 # 芯一级结构质量
35 qm_1 = 334 # 芯一级秒流量
36
37 # 大助推器
38 m2_0 = 158000
39 m2_1 = 14500
40 qm_2 = 820
41
42 # 小助推器(单个助推器):
43 m3_0 = 72700
44 m3_1 = 7100
45 qm_3 = 410
46
47 # 整流罩
48 m4 = 4000
49
50 # 经纬度, 单位度
51 latitude_0 = 19.6 # 纬度
52 longitude_0 = 111 # 经度

```

```

53
54 # 攻角参数
55 alpha_max = 0.65 #单位弧度
56 alpha_costant = 0.1 # 转弯段参数
57
58 # 时间
59 t1 = 15 # 垂直起飞段结束，进入转弯段
60 t2 = 120 # 120s整流罩分离。
61 t3 = 160 # 160秒两小助推关机分离，大助推外侧关机（推力减半，秒流量减半）
62 t4 = 190 # 190秒两大助推内侧关机分离
63 t5 = 459.58 # 芯一级关机
64
65 # 目标轨道
66 Target_orbital_altitude = 500e3
67 Target_orbital_inclination = 60 #单位 度
68 #轨道高度偏差1km，轨道倾角偏差0.5°
69
70 # 放大系数
71 K_phi = 1
72 K_alpha = 1
73
74 # 仿真时间步
75 timestep = 0.01
76 ##### 类的定义 #####
77 # 导弹状态定义
78 class statu():
79     __slot__=['Time','X','H','V','theta','mass','alpha','deltaz']
80
81     # 位置
82     # 速度
83     # 欧拉角
84     # 角加速度
85     # 舵偏角
86
87     # 初始化
88     def __init__(self, Time, X=0, H=0, V=0, theta=0, mass=0):
89         self.Time = Time
90         self.X = X
91         self.H = H
92         self.V = V
93         self.theta = theta
94         self.mass = mass
95         self.alpha = 0
96         self.deltaz = 0
97         self.dq = 0
98
99     # 显式Euler法，给定飞行高度
100     def Euler(self, before, dmass):
101         self.Time = before.Time + timestep
102
103         self.X = before.X + before.V * np.cos(before.theta) * timestep
104         self.H = before.H + before.V * np.sin(before.theta) * timestep
105
106         self.deltaz = K_phi * (self.H - High_goal(self.X)) + K_phi_dot * (
            before.V * np.sin(before.theta) - High_goal_dot(self.X))
107
108         if self.deltaz > 30:
109             self.deltaz = 30

```

```

110     elif self.deltaz < -30:
111         self.deltaz = -30
112
113     self.alpha = 0.24 * self.deltaz
114
115     Y = (0.25 * self.alpha + 0.05 * self.deltaz) * 0.5 * air(self.H) *
        before.V * before.V * S_ref
116
117     X = (0.005 * self.alpha * self.alpha + 0.2) * 0.5 * air(self.H) *
        before.V * before.V * S_ref
118
119     self.mass = before.mass - dmass * timestep
120     if dmass == 0:
121         P = 0
122     else:
123         P = 2000
124
125     self.V = before.V + (P * np.cos(before.alpha * 3.14159625 / 180) - X -
        self.mass * 9.8 * np.sin(before.theta)) / self.mass * timestep
126     self.theta = before.theta + (P * np.sin(self.alpha * 3.14159625 / 180) +
        Y - self.mass * 9.8 * np.cos(before.theta)) / self.mass / self.V *
        timestep
127
128 # 比例导引法，给定目标位置
129 def Euler2(self, before, Xm, Ym):
130     self.Time = before.Time + timestep
131
132     self.X = before.X + before.V * np.cos(before.theta) * timestep
133     self.H = before.H + before.V * np.sin(before.theta) * timestep
134     self.mass = before.mass
135
136     self.r = np.sqrt((self.X - Xm) * (self.X - Xm) + (self.H - Ym) * (self.
        H - Ym))
137
138     self.dq = - before.V * np.sin(before.theta - np.arctan((self.H -
        Ym) / (self.X - Xm))) / self.r
139
140     self.theta = before.theta + K_q * self.dq * timestep
141
142     P = 0
143
144
145     self.alpha = (self.mass * before.V * K_q * self.dq + self.mass * 9.8
        * np.cos(self.theta)) / (P + (0.25 + 0.05 / 0.24) * 0.5 * air(self
        .H) * before.V * before.V * S_ref) / 3.14159 * 180
146
147     self.deltaz = self.alpha / 0.24
148
149     if self.deltaz > 30:
150         self.deltaz = 30
151     if self.deltaz < -30:
152         self.deltaz = -30
153
154     self.alpha = 0.24 * self.deltaz
155
156     X = (0.005 * self.alpha * self.alpha + 0.2) * 0.5 * air(self.H) *
        before.V * before.V * S_ref
157     self.V = before.V + (P * np.cos(before.alpha * 3.14159625 / 180) - X -

```

```

        self.mass*9.8*np.sin(before.theta)) /self.mass*timestep
158
159 def Euler3(self, before, Xm, Ym):
160     self.Time = before.Time + timestep
161
162     self.X = before.X + before.V * np.cos(before.theta) * timestep
163     self.H = before.H + before.V * np.sin(before.theta) * timestep
164     self.mass = before.mass
165
166     self.r = np.sqrt((self.X - Xm)*(self.X - Xm) + (self.H - Ym)*(self.
        H - Ym))
167
168     self.dtheta = - K_q * before.V * np.sin(before.theta - np.arctan((
        self.H - Ym)/(self.X - Xm)))/ self.r
169
170     self.theta = before.theta + self.dtheta * timestep
171
172     P = 0
173
174     self.alpha = (self.mass* before.V * self.dtheta + self.mass * 9.8 *
        np.cos(self.theta))/(P + (0.25 + 0.05/0.24) * 0.5 * air(self.H
        ) * before.V * before.V * S_ref) /3.14159*180
175
176     self.deltaz = self.alpha / 0.24
177
178     if self.deltaz > 30:
179         self.deltaz = 30
180     if self.deltaz < -30:
181         self.deltaz = -30
182
183     self.alpha = 0.24 * self.deltaz
184
185     X = (0.005 * self.alpha * self.alpha + 0.2) * 0.5 * air(self.H) *
        before.V * before.V * S_ref
186     self.V = before.V + ((P*np.cos(self.alpha*3.14159/180) - X)/self.
        mass-9.8*np.sin(self.theta)) *timestep
187
188
189 ##### 部分函数 #####
190 # 大气参数
191 def air (High):
192     rho0 =1.2495
193     T0 = 288.15
194     Temp = T0 - 0.0065*High
195     rho = rho0 * np.exp(4.25588*np.log(Temp / T0))
196     return rho
197
198
199 # 飞行方案
200 def High_goal(X):
201     if X <= 9100:
202         return 2000 * np.cos(0.000314 * 1.1 * X) + 5000
203     elif X <= 24000:
204         return 3050
205     else:
206         return 0
207
208 # 飞行方案的时间导数

```

```

209 def High_goal_dot(X):
210     if X <= 9100:
211         return -2000 * 0.000314 * np.sin(0.000314 * 1.1 * X)
212     elif X <= 24000:
213         return 0
214     else:
215         return 0
216
217 # 读入数据
218 def read_file(file_path):
219     # 读取文档中的数据
220     with open(file_path, 'r', encoding='UTF-8') as f:
221         lines = f.readlines()
222
223     # 提取关键数据
224     data = []
225     for line in lines:
226         items = line.strip().split()
227         if items:
228             data.append(items)
229     return data
230
231 def alpha_cornering( time ):
232     b = np.exp(alpha_constant*(t1-time))
233     return 4* alpha_max * b *(b-1)
234
235
236 ##### 插值函数 #####
237 # cx插值数据预处理
238 data = read_file('data/cx.txt')
239 cx_high_numbers = np.array(data[1], dtype=float)
240 cx_mach_numbers = np.array(data[3], dtype=float)
241 cx_values = [row for row in np.array(data[5:], dtype=float)]
242 # kind参数决定了插值的类型, 'linear'代表线性插值
243 cx_interp_func = interp2d(cx_high_numbers, cx_mach_numbers, cx_values, kind
244                             ='linear')
245 # print(cx_interp_func(0.1,0.1),cn_alpha_interp_func(0.3))
246
247 # cn插值数据预处理
248 data = read_file('data/cn_alpha.txt')
249 cn_alpha_mach = np.array(data[1], dtype=float)
250 cn_alpha_values = [row for row in np.array(data[3:], dtype=float)]
251 # 允许外插
252 cn_alpha_interp_func = interp1d(cn_alpha_mach, cn_alpha_values, kind='
253                                 linear', fill_value='extrapolate')
254
255 ## 芯一级推力P_1 插值预处理
256 data = read_file('data/P_1.txt')
257 P_1_high = np.array(data[1], dtype=float)
258 P_1_values = [row for row in np.array(data[3:], dtype=float)]
259
260 P_1_interp_func = interp1d(P_1_high, P_1_values, kind='linear', fill_value=
261                             'extrapolate')
262
263 ## 两大助推器总推力P_2 插值预处理
264 data = read_file('data/P_2.txt')
265 P_2_high = np.array(data[1], dtype=float)
266 P_2_values = [row for row in np.array(data[3:], dtype=float)]

```

```

264
265 P_2_interp_func = interp1d(P_2_high, P_2_values, kind='linear', fill_value=
    'extrapolate')
266 print(P_2_interp_func(20))
267
268 ## 两小助推器总推力P_3 插值预处理
269 data = read_file('data/P_1.txt')
270 P_3_high = np.array(data[1], dtype=float)
271 P_3_values = [row for row in np.array(data[3:], dtype=float)]
272
273 P_3_interp_func = interp1d(P_3_high, P_3_values, kind='linear', fill_value=
    'extrapolate')
274 print(P_3_interp_func(20))
275
276 ##### 计算初始化 #####
277 # 发射方位角
278 A_0 =
279
280 # 飞行初始状态
281 statu_n = [statu(0, 0, 7000, 250, 0, 320)]
282 statu_n[0].alpha = 0
283 statu_n[0].deltaz = 0
284
285 X_goal = np.arange(0, 24000, 10)
286 H_goal = [High_goal(i) for i in X_goal]
287 plt.plot(X_goal, H_goal, 'b--', alpha=0.5, linewidth=1, label='飞行方案高度',
    )
288
289 # 方位角
290 ##### 计算开始 #####
291 # 第一阶段
292 while statu_n[-1].Time < 15:
293     statu_n.append(statu(statu_n[-1].Time + timestep))
294     statu_n[-1].Euler(statu_n[-2], 0)
295
296
297 ##### 绘图可视化 #####
298 # 绘图
299 X_data = [n.X for n in statu_n]
300 H_data = [n.H for n in statu_n]
301 plt.plot(X_data, H_data, 'r-.', alpha=0.5, linewidth=1, label='实际飞行高度',
    )
302 plt.title("弹道铅垂平面轨迹")
303 plt.legend() #显示上面的label
304 plt.xlabel('X(m)') #x_label
305 plt.ylabel('H(m)') #y_label
306 plt.ylim(0, 8000)
307 plt.xlim(0, 30000) #仅设置y轴坐标范围
308 plt.savefig('img/飞行轨迹.png', dpi=300)
309 plt.clf()
310
311 T_data = [n.Time for n in statu_n]
312 deltaz_data = [n.deltaz for n in statu_n]
313 plt.plot(T_data, deltaz_data, 'r-.', alpha=0.5, linewidth=1, label='舵偏角$\delta z$')
314 plt.title("飞行方案舵偏角")
315 plt.legend() #显示上面的label
316 plt.xlabel('Time(s)') #x_label

```



```

317 plt.ylabel('$\delta z$')#y_label
318 plt.ylim(-50,50)
319 plt.xlim(0,200)
320 plt.savefig('img/飞行舵偏角.png', dpi=300)
321 plt.clf()
322
323 plt.plot(T_data,H_data, 'r-.', alpha=0.5, linewidth=1, label='实际飞行速度V
    ')
324 plt.title("飞行高度的时间变化曲线")
325 plt.legend() #显示上面的label
326 plt.xlabel('Time(s)') #x_label
327 plt.ylabel('H(m)')#y_label
328 plt.ylim(0,7000)
329 plt.xlim(0,200)
330 plt.savefig('img/飞行高度.png', dpi=300)
331 plt.clf()
332
333 V_data = [n.V for n in statu_n]
334 plt.plot(T_data,V_data, 'r-.', alpha=0.5, linewidth=1, label='实际飞行速度V
    ')
335 plt.title("飞行速度的时间变化曲线")
336 plt.legend() #显示上面的label
337 plt.xlabel('Time(s)') #x_label
338 plt.ylabel('速度V')#y_label
339 plt.ylim(100,250)
340 plt.xlim(0,200)
341 plt.savefig('img/飞行速度.png', dpi=300)
342 plt.clf()
343
344 theta_data = [n.theta*180/3.14159 for n in statu_n]
345 plt.plot(T_data,theta_data, 'r-.', alpha=0.5, linewidth=1, label=r'舵偏角$\
    theta$')
346 plt.title("飞行方案弹道倾角")
347 plt.legend() #显示上面的label
348 plt.xlabel('Time(s)') #x_label
349 plt.ylabel(r'$\theta$')#y_label
350 plt.ylim(-50,50)
351 plt.xlim(0,200)
352 plt.savefig('img/飞行弹道倾角.png', dpi=300)
353 plt.clf()

```

## 致 谢

感谢我的老师和我的朋友们……