

赛区评阅编号（由赛区组委会填写）：

2023 年高教社杯全国大学生数学建模竞赛

承 诺 书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称“竞赛章程和参赛规则”，可从 <http://www.mcm.edu.cn> 下载）。

我们完全清楚，在竞赛开始后参赛队员不能以任何方式，包括电话、电子邮件、“贴吧”、QQ 群、微信群等，与队外的任何人（包括指导教师）交流、讨论与赛题有关的问题；无论主动参与讨论还是被动接收讨论信息都是严重违反竞赛纪律的行为。

我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D/E 中选择一项填写）：_____A_____

我们的报名参赛队号（12 位数字全国统一编号）：_____1234_____

参赛学校（完整的学校全称，不含院系名）：_____

参赛队员（打印并签名）：1. _____

2. _____

3. _____

指导教师或指导教师组负责人（打印并签名）：_____

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期：_____2023_____年_____9_____月_____8_____日

（请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

赛区评阅编号：_____ 全国评阅编号：_____
(由赛区填写) (全国组委会填写)

2023 年高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅记录（可供赛区评阅时使用）：

评阅人						
备注						

送全国评阅统一编号：
(赛区组委会填写)

(请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。)

全国大学生数学建模竞赛 Typst 模板

摘要

本文档是为全国大学生数学建模竞赛编写的 Typst 模板，旨在让大家专注于论文的内容写作，而不用花费过多精力在格式的定制和调整上。本文档默认页边距为 2.5cm，正文中文字体为中易宋体 (SimSun)，英文字体为 Times New Romans，字号为 12pt (小四)。

本模板文件由主要以下六部分组成：

- main.typ 主文件
- template.typ 文档格式控制，包括一些基础的设置、函数
- refs.bib 参考文献
- fonts 字体文件夹
- figures 图片文件夹

关键字： Typst 模板 数学建模

一、模板的基本使用

使用本模板之前，请阅读模板的使用说明文档。下面是本模板使用的基本样式：

```
#import "template.typ": *

#show: project.with(
  title: "论文题目",
  problem_chosen: "A", // 选择的题号
  team_number: "1234", // 队伍编号
  college_name: "学校名称",
  member: (
    A: "成员 A",
    B: "成员 B",
    C: "成员 C",
  ),
  advisor: " ", // 指导老师
  date: datetime(year: 2023, month: 9, day: 8), // 日期

  cover_display: true, // 是否显示封面与编号页

  abstract: [
    此处填写摘要内容
  ],

  keywords: ("关键字 1", "关键字 2", "关键字 3"),
)

// 正文内容

// 参考文献
#bib("refs.bib")

// 附录
#heading("附录 A XXXX", numbering: none)
```

代码 1 基本样式

根据要求，电子版论文提交时需去掉封面和编号页。可以将 `cover_display` 设置为 `false` 来实现，即：

```
cover_display: false, // 是否显示封面与编号页
```

这样就能实现了。

请确保你的论文内容符合要求，包括但不限于页数、格式、内容等。

下面给出写作与排版上的一些示例。

二、图片

在数学建模中，我们经常需要插入图片。Typst 支持的图片格式有“png”，“jpeg”，“gif”，“svg”，其他类型的图片无法插入文档。我们可以通过改变参数 `width` 来改变图片的大小，详见 `typst/docs/image`。下面是一些图片插入的示例：

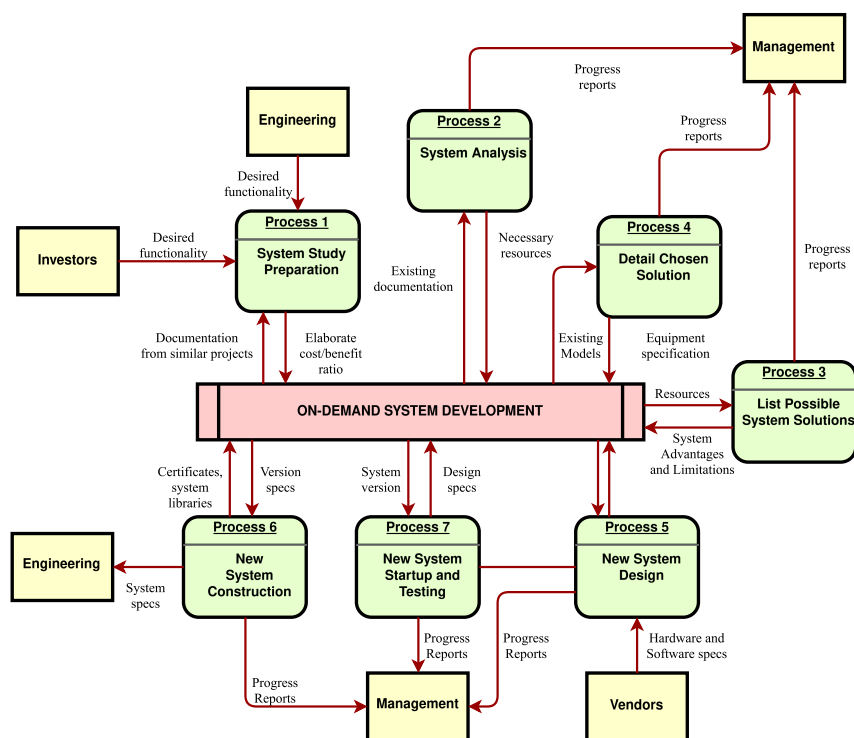


图 1 单图示例

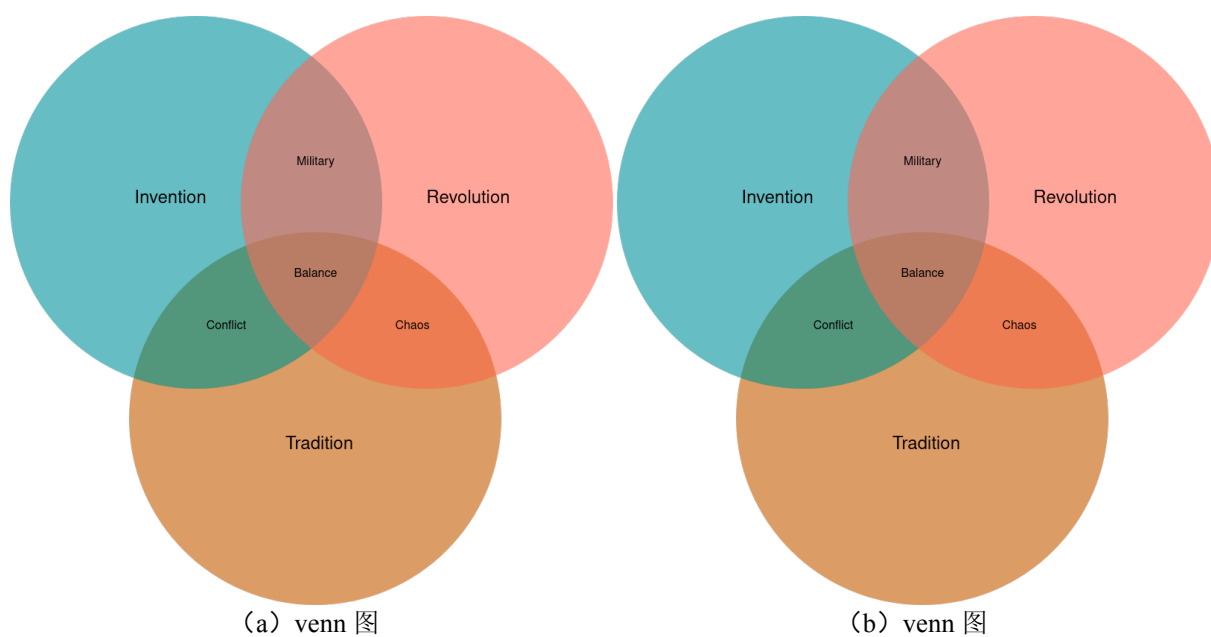


图 2 多图并排示例

三、表格

本模板创建表格引入了 `tablex` 包，可以方便地创建表格。其标准格式及其代码说明如下：

Names	Properties	
	Type	Size
Machine	Steel	5 cm ³
Frog	Animal	6 cm ³
Frog	Animal	6 cm ³
Frog	Animal	6 cm ³
Frog	Animal	6 cm ³
Frog	Animal	6 cm ³

表 1 表格示例

```
#figure(  
  kind: table, // 设置 figure 的类型为 table ▲  
  tablex(  
    columns: 3, // 列数  
    align: center + horizon, // 对齐方式  
    auto-hlines: false, // 是否自动添加横线  
    auto-vlines: false, // 是否自动添加竖线  
    repeat-header: true, // 是否重复表头  
  
    /* --- header --- */  
    hlinex(), // 添加横线  
    rowspanx(2)[Names], colspanx(2)[Properties], ()  
    // rowspanx(2) 表示跨两行, colspanx(2) 表示跨两列  
    hlinex(start: 1, end: 3, stroke: 0.2pt),  
    (), [Type], [Size],  
    hlinex(stroke: 0.2pt),  
    /* ----- */  
  
    [Machine], [Steel], [5 $"cm"^3$],  
    [Frog], [Animal], [6 $"cm"^3$],  
    [Frog], [Animal], [6 $"cm"^3$],  
    [Frog], [Animal], [6 $"cm"^3$],  
    [Frog], [Animal], [6 $"cm"^3$],  
    [Frog], [Animal], [6 $"cm"^3$],  
    hlinex(),  
  ),  
  caption: "表格示例"  
)
```

更多使用方法可以查看 [GitHub](#) 页面。

四、公式

数学建模中，公式的使用是必不可少的。Typst 可以使用 Typst 原生语法插入公式，参考 `typst/docs/math`。下面是一些公式插入的示例：

首先是行内公式，例如 $a^2 + b^2 = c^2$ 。行内公式使用 `$$` 包裹，公式和两端的 `$$` 之间没有空格。

其次是行间公式，例如：

$$\iiint_{\Omega} \left(\frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y} + \frac{\partial R}{\partial z} \right) dv = \oiint_{\Sigma} P dydz + Q dzdx + R dxdy \quad (1)$$

式 (1) 是高斯公式。行间公式使用 `$$` 环境包裹，公式和两端的 `$$` 之间至少有一个空格。

公式内可以使用换行符 `\` 换行。若需要对齐，每行可以包含一个或多个对齐点 `&` 对其进行对齐。例如：

$$\begin{aligned} \sum_i b_i &= \sum_i \sum_{h,j \neq i} \frac{\sigma_{hj}(i)}{\sigma_{hj}} \\ &= \sum_{h \neq j} \frac{1}{\sigma_{hj}} \sum_{i \neq h,j} \sigma_{hj}(i) \end{aligned} \quad (2)$$

`&` 是对齐的位置，`&` 可以有多个，但是每行的个数要相同。

矩阵输入示例：

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \quad (3)$$

分段函数可以使用 `case` 环境：

$$f(x) = \begin{cases} 0 & x \text{ 为无理数,} \\ 1 & x \text{ 为有理数.} \end{cases} \quad (4)$$

假如要公式里面有个别文字，需要把这部分放在 `text` 环境里面，即 `text[文本内容]`。

如果公式中有个别需要加粗的字母，可以使用 `bold()` 进行加粗。如， $\alpha a \alpha a$ 。

以上仅为一些简单的公式示例，更多的公式使用方法可以查看 `typst/docs/math`。

五、定理环境

在本模板中，我们定义了一些常用的定理环境，包括 theorem、lemma、corollary、assumption、conjecture、axiom、principle、problem、example、proof、solution。可以根据论文的实际需求合理使用。

定义 1: 这是一个定义

引理 1: 这是一个引理

推论 1: 这是一个推论

假设 1: 这是一个假设

猜想 1: 这是一个猜想

公理 1: 这是一个公理

定律 1: 这是一个定律

问题 1: 这是一个问题

例 1: 这是一个例子

证明 1: 这是一个证明

解 1: 这是一个解

六、其他功能

6.1 脚注

利用 `#footnote(脚注内容)` 可以生产脚注¹

6.2 无序列表与有序列表

无序列表例：

- 元素 1
- 元素 2

¹脚注例

- ...

有序列表例:

1. 元素 1
2. 元素 2
3. ...

6.3 字体粗体与斜体

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 `*需要加粗的内容*` 或 `#strong[需要加粗的内容]` 来实现。例如: **这是加粗的字体, This is bold fonts**。

中文字体没有斜体设计,但是英文字体有。斜体 *Italics*。

七、参考文献与引用

参考文献对于一篇正式的论文来说是必不可的,在建模中重要的参考文献当然应该列出。Typst 支持使用 BibTeX 来管理参考文献。在 `refs.bib` 文件中添加参考文献的信息,然后在正文中使用 `#cite(<引用的文献的 key>)` 来引用文献。例如: ^[1]。最后通过 `#bib("refs.bib")` 来生成参考文献列表。

参考文献

- [1] ASTLEY R, MORRIS L. At-scale impact of the Net Wok: A culinarily holistic investigation of distributed dumplings[J]. Armenian Journal of Proceedings, 2020, 61: 192-219.

附录 A 线性规划 - Python 源程序

```
import numpy as np
from scipy.optimize import linprog

c = np.array([2, 3, 1])
A_up = np.array([[ -1, -4, -2], [-3, -2, 0]])
b_up = np.array([-8, -6])

r = linprog(c, A_ub=A_up, b_ub=b_up, bounds=((0, None), (0, None), (0, None)))

print(r)
```

附录 B 非线性规划 - Python 源程序

```
from scipy import optimize as opt
import numpy as np
from scipy.optimize import minimize

# 目标函数
def objective(x):
    return x[0] ** 2 + x[1] ** 2 + x[2] ** 2 + 8

# 约束条件
def constraint1(x):
    return x[0] ** 2 - x[1] + x[2] ** 2 # 不等约束

def constraint2(x):
    return -(x[0] + x[1] ** 2 + x[2] ** 2 - 20) # 不等约束

def constraint3(x):
    return -x[0] - x[1] ** 2 + 2

def constraint4(x):
    return x[1] + 2 * x[2] ** 2 - 3 # 不等约束

# 边界约束
b = (0.0, None)
bnds = (b, b, b)

con1 = {'type': 'ineq', 'fun': constraint1}
con2 = {'type': 'ineq', 'fun': constraint2}
con3 = {'type': 'eq', 'fun': constraint3}
con4 = {'type': 'eq', 'fun': constraint4}
cons = ([con1, con2, con3, con4]) # 4 个约束条件
x0 = np.array([0, 0, 0])

# 计算
solution = minimize(objective, x0, method='SLSQP', bounds=bnds, constraints=cons)
x = solution.x

print('目标值: ' + str(objective(x)))
print('答案为')
print('x1 = ' + str(x[0]))
print('x2 = ' + str(x[1]))
```