

2017 年全国大学生数学建模竞赛参赛总结

计 63 陈晟祺^{*}

计 64 周聿浩[†]

计 62 陈智康[‡]

1 简单回顾

2017 年 9 月，我们三个人参加了全国大学生数学建模竞赛（以下简称“国赛”），获得了全国一等奖。事实上作为没有任何数学建模经验的新手队，我们对最终这么好的成绩都颇感惊讶。下面分享一些关于比赛的拙见，希望对将来同学们参与建模竞赛有些帮助。

2 题目选择

首先，在题目旁边如果有关于比赛须知等文档**一定要先打开认真看一遍**不要忽略它。我们当时因为没有认真看差点选了 D 题……

然后 A 和 B 两题，还是看看感觉哪题顺眼就选哪题吧。我们当时看着感觉 A 题比较好做（其实后来发现似乎不是这样）就选了。

3 比赛进程

至少在这次国赛我们做的过程中，我觉得并不是需要太高深的数学方面的知识，大概大一时候的线性代数、微积分和概率统计基本上就挺够的了。如果涉及到特别的，比如这回 A 题中 Radon 变换里用到的信号处理方面的知识例如傅立叶变换等有了前面的知识基本上花一些时间现场学也是可以学会的。

不过了解一些建模或者优化方面常用的工具比如线性规划之类的也有好处，并不需要具体到这个东西是干什么的，关键是知道它大概能干什么到时候如果发现似乎能够用上的话再去学也是来得及的。

具体在做题方面，我也是第一次做没有太多经验。只是感觉一点就是，可以多画画图，拿到一组数据之后如果可行就先用 Python 或者 Matlab 进行可视化，这样可能会有一些启发。还有就是对于一个值的估计，如果可以计算方差/标准差一定要算出来，从这里面有时候能够发现一些有意思的事情，同时报告它们还能够让人了解你的估计的准确性。

有一点就是如果不是万不得已，一定要睡好觉，让脑子清醒一点，否则出错的概率会非常高。

^{*}harry-chen@outlook.com

[†]miskcoo@gmail.com

[‡]chenzk16@mails.tsinghua.edu.cn

还有就是代码一定要组织好,不要太零散,如果使用 Python 的话最好用 Jupyter Notebook 这样的交互式工具来辅助,当然 Matlab 和 Mathematica 本身就是交互的,这就没有什么好说的了。

另外,关于国赛的时间,这次是在开学前三天,有些院系(比如贵系)在这段时间可能刚好是小学期,所以最好事先考虑一下如果有大作业之类的东西提前找老师问一下题目之类的先把它做完,还有考试也是可以先把它们复习完不然如果刚好在那段时间有考试或者赶大作业可能时间就会比较紧张了。

4 论文排版

在排版上,我们首先推荐使用 \LaTeX ,它在处理数学公式上具有巨大的优势。作为计算机系的学生,我们三个人之前都有使用 \LaTeX 排版的经验,所以没有太高的学习成本。对于新手来说,学习曲线还是比较陡峭的,因此推荐事先进行一些摸索和学习,尤其是图表、浮动体的处理等比较棘手的部分。目前 \TeX Live 在全平台上都很成熟易用,也有现成的模板,避免对格式不必要的纠结。在这里推荐一些在线资源:

- 全国大学生数学建模竞赛 \LaTeX 模板: <https://github.com/latexstudio/CUMCMThesis>
- MCM/ICM Thesis Template: <https://github.com/Liam0205/mcmthesis>
- 一份其实很短的 \LaTeX 入门文档: <https://liam0205.me/2014/09/08/latex-introduction/>
- \LaTeX 笔记第二版 v2.03: <http://maths.nju.edu.cn/~meijq/tex/lnotes2.pdf>
- \LaTeX 科技排版工作室: <http://www.latexstudio.net/>
- China \TeX : <http://www.chinatex.org/>

如果时间仓促或者实在不适应 \LaTeX , Word 依旧是一个可行的选择。但请一定认真学习并正确使用 Word 的各项高级功能,如交叉引用、目录生成、样式等,有助于用撰写出专业、美观的论文,并减少在格式调整等细节上浪费的时间。

5 辅助软件

虽然 MATLAB 基本占据了建模界的半壁江山,但我们三个人都从没有接触过,也懒得学。所以我们主要采用了 Python 和 Mathematica 进行辅助运算,Python 是主力,借助于 Jupyter Notebook 这一可视化神器以及 matplotlib、sklearn 等专业的绘图、统计工具包,我们可以快速、直观地进行各种运算与尝试。而 Mathematica 用在代数表达式的推导和数值方程的求解上: A 题有解析几何部分,符号运算量很大,多亏了这一套代数系统的帮助;我们的解题方法中涉及了数万个方程的联合求解,经过测试 Python 的 scipy 和 numpy 等库虽然使用方便,但是效率比较低,而 Mathematica 可以很快地求解。同时 Mathematica 的函数式编程语法也给表达和计算带来了很大的便利,也没有 MATLAB 的向量化那么反直觉。

在此之外,强烈推荐使用 Git 进行版本控制,包括论文、代码与所有的中间文件。一方面这对多人协作解题、撰写论文带来了极大的便利(尤其是与 \TeX 这种纯文本格式结合时);另一方面,所有的历史版本都会被记录下来,方便随时核查更改、回溯之前的思路,以及及时发现谁写错了然后扣锅。