

“华为杯”第十三届全国研究生
数学建模竞赛

学 校 XX 大学

参赛队号 4321

	1. 小米
队员姓名	2. 向左
	3. 哈哈



题 目 全国研究生数学建模竞赛论文标题

plusminus

本模板是为全国研究生数学建模竞赛编写的 \LaTeX 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. 本手册是相应的参考, 其中提供了一些环境和命令可以让模板的使用更为方便. 同时需要注意, 使用者需要有一定的 \LaTeX 的使用经验, 至少要会使用 `ctex` 宏包的一些功能, 比如调节字距或修改字体大小等等.

另外, 欢迎大家购买我们是视频教程, 点击[这里](#).

欢迎收藏和光顾我们的店铺: <http://latexstudio.taobao.com>

关注我们的微信公众号:



关键字: 折叠桌 曲线拟合 非线性优化模型 受力分析

1. 问题重述

1.1 引言

创意平板折叠桌注重于表达木制品的优雅和设计师所想要强调的自动化与功能性。为了增大有效使用面积。设计师以长方形木板的宽为直径截取了一个圆形作为桌面，又将木板剩余的面积切割成了若干个长短不一的木条，每根木条的长度为平板宽到圆上一点的距离，分别用两根钢筋贯穿两侧的木条，使用者只需提起木板的两侧，便可以在重力的作用下达达到自动升起的效果，相互对称的木条宛如下垂的桌布，精密的制作工艺配以质朴的木材，让这件工艺品看起来就像是工业革命时期的机器。

1.2 问题的提出

围绕创意平板折叠桌的动态变化过程、设计加工参数，本文依次提出如下问题：

(1) 给定长方形平板尺寸 ($120\text{cm} \times 50\text{cm} \times 3\text{cm}$)，每根木条宽度 (2.5cm)，连接桌腿木条的钢筋的位置，折叠后桌子的高度 (53cm)。要求建立模型描述此折叠桌的动态变化过程，并在此基础上给出此折叠桌的设计加工参数和桌脚边缘线的数学描述。

(2) 折叠桌的设计应做到产品稳固性好、加工方便、用材最少。对于任意给定的折叠桌高度和圆形桌面直径的设计要求，讨论长方形平板材料和折叠桌的最优设计加工参数，例如，平板尺寸、钢筋位置、开槽长度等。对于桌高 70cm ，桌面直径 80cm 的情形，确定最优设计加工参数。

(3) 给出软件设计的数学模型，可以根据客户任意设定的折叠桌高度、桌面边缘线的形状大小和桌脚边缘线的大致形状，给出所需平板材料的形状尺寸和切实可行的最优设计加工参数，使得生产的折叠桌尽可能接近客户所期望的形状，并根据所建立的模型给出几个设计的创意平板折叠桌。要求给出相应的设计加工参数，画出至少 8 张动态变化过程的示意图。

2. 模型的假设

- 忽略实际加工误差对设计的影响；
- 木条与圆桌面之间的交接处缝隙较小，可忽略；
- 钢筋强度足够大，不弯曲；
- 假设地面平整。

3. 符号说明

符号	意义
D	木条宽度 (cm)
L	木板长度 (cm)
W	木板宽度 (cm)
N	第 n 根木条
T	木条根数
H	桌子高度 (cm)
R	桌子半径 (cm)
R	桌子直径 (cm)

4. 问题分析

4.1 问题一分析

题目要求建立模型描述折叠桌的动态变化图, 由于在折叠时用力大小的不同, 我们不能描述在某一时刻折叠桌的具体形态, 但我们可以用每根木条的角度变化来描述折叠桌的动态变化。首先, 我们知道折叠桌前后左右对称, 我们可以运用几何知识求出四分之一木条的角度变化。最后, 根据初始时刻和最终形态两种状态求出桌腿木条开槽的长度。

4.2 问题二分析

题目要求从折叠桌的稳固性好、加工方便、用材最少三个角度, 确定设计加工参数。我们可以从应力、支撑面积考虑稳固性, 从开槽长度考虑加工方便, 从木板长度考虑用材最少。而它们之间又是相互制约, 我们需要确定最优设计加工参数, 可以建立非线性规划模型, 用 lingo 软件来求解最优设计加工参数 (平板尺寸、钢筋位置、开槽长度等), 这里以合力的方向 (斜向上) 与最长木条 (桌腿) 的夹角方向最小为目标函数, 以木条所承受应力小于木条的许用应力、支撑面积大于桌面面积、木条的开槽长度小于木条本身长为约束条件。

4.3 问题三分析

题目要求制作软件的意思就是客户给定折叠桌高度、桌面边缘线的形状大小和桌脚边缘线的大致形状, 将这些信息输入程序就得到客户想要的桌子。我们在求解最优设计

加工参数时，自行给定桌面边缘线形状（椭圆、相交圆等），桌脚边缘线形状，折叠桌高度，应用第二问的非线性规划模型，用 MATLAB 软件绘制折叠桌截面图，得到自己设计的创意平板折叠桌。

问题三流程图：

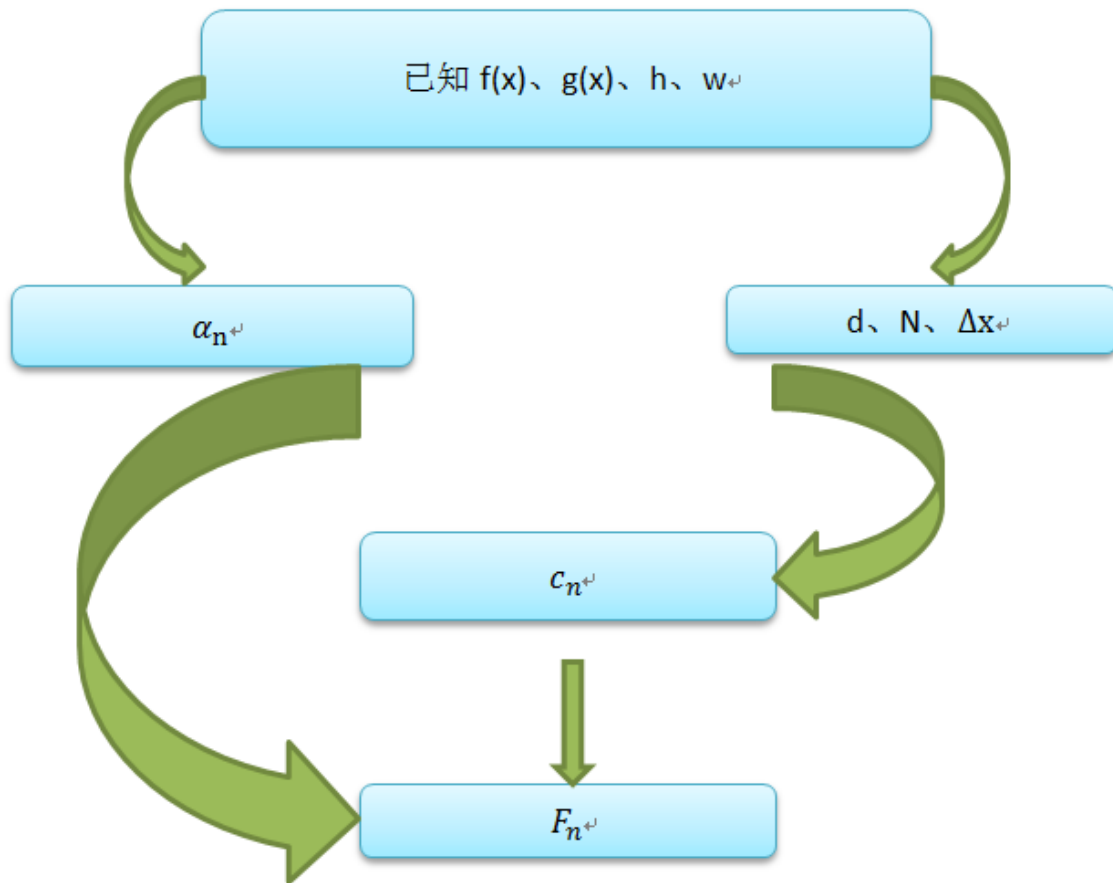


图 1 问题三流程图

5. 参考文献

[1]

[2]

附录 A 我的 MATLAB 源程序

```

kk=2; [mdd, ndd]=size(dd);
while ~isempty(V)
    [tmpd, j]=min(W(i, V)); tmpj=V(j);
    for k=2:ndd
        [tmp1, jj]=min(dd(1, k)+W(dd(2, k), V));
        tmp2=V(jj); tt(k-1, :)= [tmp1, tmp2, jj];
    end
    tmp=[tmpd, tmpj, j; tt]; [tmp3, tmp4]=min(tmp(:, 1));
    if tmp3==tmpd, ss(1:2, kk)=[i; tmp(tmp4, 2)];
    else, tmp5=find(ss(:, tmp4)~=0); tmp6=length(tmp5);
    if dd(2, tmp4)==ss(tmp6, tmp4)
        ss(1:tmp6+1, kk)=[ss(tmp5, tmp4); tmp(tmp4, 2)];
    else, ss(1:3, kk)=[i; dd(2, tmp4); tmp(tmp4, 2)];
    end; end
    dd=[dd, [tmp3; tmp(tmp4, 2)]]; V(tmp(tmp4, 3))=[];
    [mdd, ndd]=size(dd); kk=kk+1;
end; S=ss; D=dd(1, :);

```