

---

# 承诺书

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）。

日期: 2015 年 08 月 22 日

(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（由赛区组委会填写）：

---

**2016 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（由赛区组委会填写）：

全国评阅随机编号（由全国组委会填写）：

（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，即电子版论文的第一页为标题、摘要和关键词页。）

# 创意平板折叠桌

## Abstract

目前住宅空间的紧张导致越来越多的折叠家具的出现。某公司设计制作了一款折叠桌以满足市场需要。以此折叠桌为背景提出了三个问题，本文运用几何知识、非线性约束优化模型等方法成功解决了这三个问题，得到了折叠桌动态过程的描述方程以及在给定条件下怎样选择最优设计加工参数，并针对任意形状的桌面边缘线等给出了我们的设计。

针对问题一，根据木板尺寸、木条宽度，首先确定木条根数为 19 根，接着，根据桌子是前后左右对称的结构，我们只以桌子的四分之一为研究对象，运用空间几何的相关知识关系，推导并建立了几何模型。接着用 MATLAB 软件编程，绘制出折叠桌动态变化过程图。然后求出折叠桌各木条相对桌面的角度、各木条长度、各木条的开槽长度等数据，相关结果见表 1。然后建立相应的三维坐标系，求出桌角各端点坐标，绘出桌角边缘线曲线图，并用 MATLAB 工具箱作拟合，求出桌角边缘线的函数关系式，并对拟合效果做分析（见表 3）。

针对问题二，在折叠桌高度、桌面直径已知情况下，综合考虑桌子稳固性、加工方便、用材最少三个方面因素，我们运用材料力学等相关知识，对折叠桌作受力分析，确定稳固性、加工方便、用材最少三个方面因素间的相互制约关系，建立非线性优化模型。用 lingo 软件编程，求出对于高 70 cm，桌面直径 80 cm 的折叠桌，平板尺寸 172.24cm×80cm×3cm、钢筋位置在桌腿上距离铰链 46.13cm 处、各木条的开槽长度（见表 3）、最长木条（桌脚）与水平面夹角 71.934°。

针对问题三，对任意给出的桌面边缘线 ( $f(x)$ )，不妨假定曲线是对称的（否则，桌子的稳定性难以保证），将对称轴上  $n$  等份，依照等份点沿着木板较长方向平行的方向下料，则这些点即是铰接处到木板中垂线（相对于木板长方向）的距离。然后修改问题二建立的优化模型，用 lingo 软件编程，得到最优设计加工参数（平板尺寸、钢筋位置、开槽长度等）。最后，我们根据所建立的模型，设计了一个桌面边缘线为椭圆的折叠桌，并且给出了 8 个动态变化过程图（见图 10）和其具体设计加工参数（见表 5）。

最后，对所建立的模型和求解方法的优缺点给出了客观的评价，并指出了改进的方法。

**关键字：** 折叠桌   曲线拟合   非线性优化模型   受力分析

## 一、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 ( $2.5\text{ cm}$ )，连接桌腿木条的钢筋的位置，折叠后桌子的高度 ( $53\text{ cm}$ )。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  $70\text{ cm}$ ，桌面直径  $80\text{ cm}$  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。

## 二、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。

### 三、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)

### 四、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：

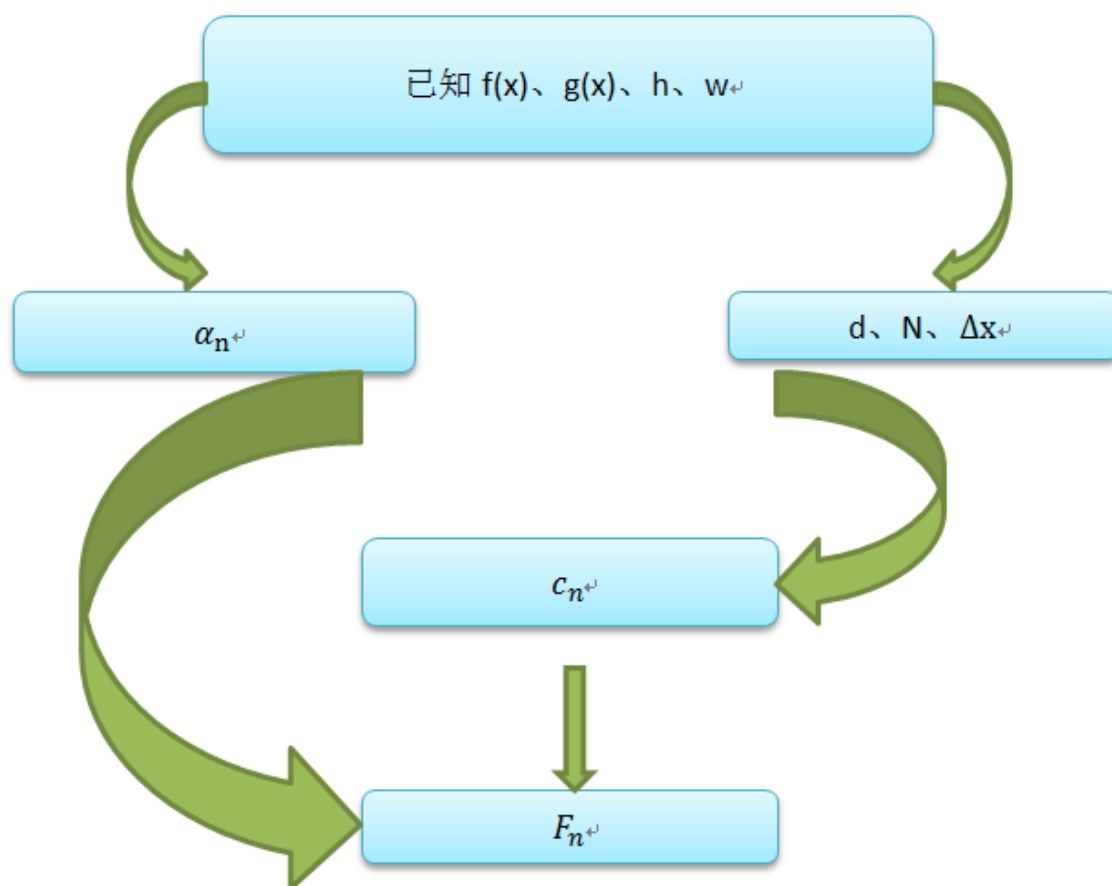


Figure 1 问题三流程图

## AppendixA 我的 MATLAB 源程序

lkjsdf

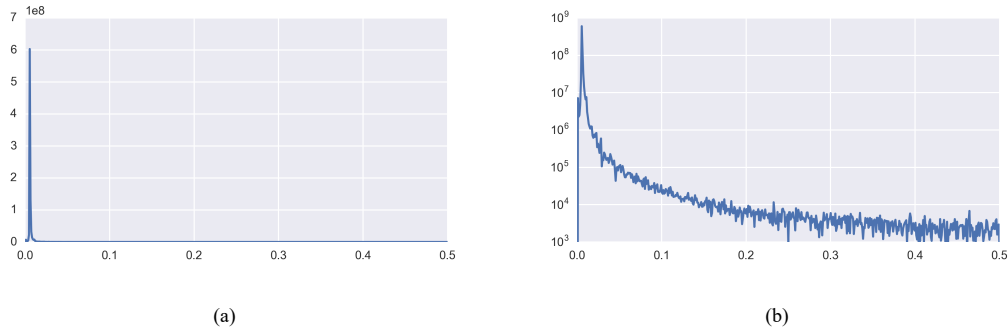


Figure 2 (a) Normal y-scale; (b) Log y-scale;

### Use FFT to carry out spectral analysis and wave filtering

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import statsmodels.tsa.stattools as stt
5 import scipy.signal as signal
6 import seaborn as sns
7
8 # use pandas to process time series data type
9 data = pd.read_excel('E:\SYS Files\Documents\Python files\Comp.Phy\LAB_ThermalWave\data.xlsx')
10 #print data[:10]
11 data.columns = ['Time', 's1', 's2', 's3', 's4', 's5', 's6', 's7', 's8', 's9', 's10', 's11']
12 data.index = data['Time'].values
13 data.drop('Time', axis=1, level=None, inplace=True, errors='raise')
14 data.plot(rot=10,
15         #ylim=(0,100),
16         #xlim=(0,23*12 - 1)
17         )
18 cut = 2100 # choose a period of data that is stationary
19 s1 = data_cut.loc[cut:, 's1']
20 ss1 = s1 - s1.mean()
21 #####acf pacf analysis stationary
22 #ss1_acf = stt.acf(ss1, nlags=len(ss1))
23 #ss1_pacf = stt.pacf(ss1, nlags=300)
24 #ss1_acovf = stt.acovf(ss1)
25 #ss1_std_acovf = np.sqrt(ss1_acovf)
26 #plt.figure(dpi=300)
27 #plt.plot(ss1)
28 #plt.figure(dpi=300)
29 #plt.plot(ss1_acf, label='acf')
30 #plt.plot(ss1_pacf, label='pacf')
31 #plt.legend(loc='best')
32
33 #####
34 #ss1_perio = signal.periodogram(ss1)
35 #plt.semilogy(ss1_perio[0], ss1_perio[1])
36 #plt.ylim([1e2, 1e8])
37 #plt.xlabel('frequency [Hz])'
```

```

38  plt.ylabel('PSD [V**2/Hz]')
39  plt.show()
40
41  ##### FFT Spectral analysis
42  '''
43  When the input a is a time-domain signal and A = fft(a),
44  np.abs(A) is its amplitude spectrum and np.abs(A)**2 is its power spectrum.
45  The phase spectrum is obtained by np.angle(A).
46  '''
47  signal = ssl.values #take out array
48  fourier = np.fft.rfft(signal) # rfft only need to compute half because of symmetry
49  fff = (np.abs(fourier)) ** 2 #square of abs. so this is energy spectrum
50  n = signal.size
51  sample_rate = 1. # sample rate of SENSOR
52  freq = np.fft.rfftfreq(n, d=1./sample_rate) # Z[]array
53  plt.figure(figsize=(20,10),dpi=300)
54  plt.semilogy(freq, fff) # LOG SCALE
55  plt.ylim([1e3, 1e9])
56
57  ##### Wave Filtering
58  order = 10
59  # this is a fancy index tricks. not only quick but also convenient
60  truncated_fourier = np.array([x if i <= order and i >= 4 else 0 for i,x in enumerate(fourier)])
61  signal_filted = np.fft.irfft(truncated_fourier, len(signal))
62  plt.figure(dpi=300)
63  plt.semilogy(freq, fff, lw=1) # LOG SCALE
64  plt.semilogy(freq, (np.abs(truncated_fourier)) ** 2, lw=2)
65  plt.ylim([1e3, 1e10])
66
67  plt.figure(dpi=300)
68  plt.plot(signal, lw=1)
69  plt.plot(signal_filted, lw=2)
70  #plt.plot(np.zeros_like(signal), lw=2)
71  plt.show()

```

---