内容说明

- 以下为CUMCM **2019A、2020A、2022A、2023A、2024A**解题思路。
- 代码运行环境为matlab r2022b、Python 3.11。
- 当论文附录所列程序与文件给出程序不一致时，以后者为准。

CUMCM2019A 高压油管的压力控制

模型 常微分方程 解法 差分法

需要注意的是，我们队此题论文格式并不完美，可供解题思路参考

问题一

燃油压强和密度的关系

题目注1给出了燃油压力变化量和密度变化量的关系，改用数学语言描述：

$$\mathrm{d}P=\frac{E}{\rho}\mathrm{d}\rho$$

E只和P有关，因此可用分离变量法解微分方程：

$$\int\frac{1}{E}\mathrm{d}P=\int\frac{1}{\rho}\mathrm{d}\rho$$

我们没有E关于P的解析表达式，但是附件3给出了一些数值关系，因此左侧的积分我们考虑用数值积分。

于是，我们就得到了P和ρ的数值关系。为了便于后续使用，我们需要拟合出一个解析关系。类比理想气体物态方程$PV=\nu RT$，我们直接进行一次多项式拟合，拟合优度为$R^2=0.9925$，非常完美。

流出

把题图2改写成流出流量随时间的数学关系式，便于后续使用：

$$Q_{\mathrm{out}}(t)=\begin{cases}100\tau_1, & \tau_1<0.2\mathrm{ms},\\ 20, & 0.2\mathrm{ms}\le\tau_1\le2.2\mathrm{ms},\\ -100\tau_1+240, & 2.2\mathrm{ms}<\tau_1\le2.4\mathrm{ms},\\ 0, & \tau_1>2.4\mathrm{ms}.\end{cases}$$

流出质量

$$\mathrm{d}m_{\mathrm{out}}(t)=\rho_1(t)Q_{\mathrm{out}}(t)\mathrm{d}t$$

流入

由题注2公式可得流入流量随时间的关系：

$$Q_{\mathrm{in}}(t)=\begin{cases}CA\sqrt{\frac{2[P_2(t)-P_1(t)]}{\rho_{160}}}, & \tau_2\le T_{\mathrm{in}}-10\mathrm{ms},\\ 0, & \tau_2>T_{\mathrm{in}}-10\mathrm{ms}.\end{cases}$$

流入质量

$$\mathrm{d}m_{\mathrm{in}}(t)=\rho_{160}Q_{\mathrm{in}}(t)\mathrm{d}t$$

净流量

流入质量与流出质量之差即为净流入质量

$$\begin{aligned}\mathrm{d}m(t) &= \mathrm{d}m_{\mathrm{in}}(t) - \mathrm{d}m_{\mathrm{out}}(t)\\ &= \rho_{160}Q_{\mathrm{in}}(t)\mathrm{d}t - \rho_1(t)Q_{\mathrm{out}}(t)\mathrm{d}t\end{aligned}$$

密度和质量线性相关，也就是说，

$$\mathrm{d}\rho_1(t) = \frac{\mathrm{d}m(t)}{V_0}$$

开始差分：

$$\rho_1(t+\Delta t) - \rho_1(t) = \frac{m(t+\Delta t) - m(t)}{V_0}$$

由于我们有P和ρ的一次解析关系，很容易就可以根据ρ来求出管压。

目标函数

我们将管压的最大偏移量作为目标函数进行优化，使其最小：

$$Z = \max\left|P_1(t) - 100\mathrm{MPa}\right|$$

$$\min Z(T_{\mathrm{in}})$$

模型求解

由于决策变量只有一个T_{in}，直接用多重搜索算法求解。

问题二

凸轮曲线拟合

附件1给出了凸轮曲线的数值形式，肯定是要拟合出一个解析式的。用极坐标拟合：

```
f=fit(data(:,1),data(:,2),"fourier1");
```

发现精确到一次三角函数，拟合优度就能达到0.9999999997了。

柱塞高度确定

使用矩阵描述凸轮的旋转，旋转矩阵为

$$\boldsymbol{T} = \begin{pmatrix}\cos\gamma & -\sin\gamma\\ \sin\gamma & \cos\gamma\end{pmatrix}$$

其中，γ为绕原点逆时针旋转角度。

旋转前坐标为$(x(\theta), y(\theta))$的点，在旋转后坐标变成了

$$\begin{pmatrix}x_1(\theta)\\ y_1(\theta)\end{pmatrix} = \begin{pmatrix}\cos\gamma & -\sin\gamma\\ \sin\gamma & \cos\gamma\end{pmatrix}\begin{pmatrix}x(\theta)\\ y(\theta)\end{pmatrix}$$

求坐标最大值与初始值的差，即为柱塞高度

$$h(t) = \max\left[-x_1(\theta)\right] - r(\pi)$$

流出流量的修改

如论文图9、图10，流出流量和针阀最小开口面积有关，这一面积应为

$$A = \min\{S_1, S_2\}$$

重新代入题注2公式，得流出流量。

问题三

模型建立与求解

我们考虑两个喷油嘴开启时间是均匀错开的，如论文图13所示。

增加一个减压阀后，流出流量需要重新修改，这部分描述在论文中已是否详细了，此处从略。

总结

从头到尾都是优化模型，但本质仍是常微分方程。我们队一开始考虑正常求解常微分方程，但实在无法解出。参考了网络资源后采用差分格式求解，才一路顺风顺水。从这时开始，我们才真正认识到数值解法的强大。

不过，如果条件允许（题目不难，时间充裕），仍可以考虑解析解。

CUMCM2020A 炉温曲线

模型 偏微分方程 解法 有限差分法

问题一

模型建立

由于题目中给出了电路板的厚度（0.15 mm），但没有给出长度和宽度数据，所以，出题者希望我们构建一维热传导模型，而非三维。

热传导方程的建立过程和牛顿冷却定律可以参考姚端正《数学物理方法》（第四版）P103~105、P108，论文中也已经给出了详细推导过程。在这里需要说明的是，我们的论文给出的模型是偏微分方程形式的：

$$u_t = Du_{xx}$$

$$\begin{cases} u(x,t)\big|_{t=0} = T_0, \\ \left.\frac{\partial u(x,t)}{\partial x}\right|_{x=0} = \sigma\left[u(x,t)\big|_{x=0} - U(t)\right], \\ \left.\frac{\partial u(x,t)}{\partial x}\right|_{x=d} = \sigma\left[u(x,t)\big|_{x=d} - U(t)\right]. \end{cases}$$

在查阅资料时，你看到的牛顿冷却定律形式可能是

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = -hA\left(T - T_0\right)$$

这对应的是常微分方程的边界条件（由全导符号$\frac{\mathrm{d}Q}{\mathrm{d}t}$知$Q$只是时间$t$的函数），因此，模型应调整为常微分方程模型，使用常微分方程数值解法求解。这样做意味着温度在焊接区域内于同一时刻均匀分布，即表面的热量将在瞬间传入中心。因此，使用常微分方程描述这一过程将带来一定误差，但考虑到焊接区域的厚度仅有0.15 mm，因此可以近似忽略这一误差。

不同区域的炉内温度（炉内空气温度）存在差异，各小温区之间的间隙温度如何分布在题内并未给出，我们对此进行了线性插值，当然，可以使用其他函数连接。

有限差分

我们使用有限差分法进行数值求解。这里如果采用显示格式，可能出现无法拟合的情况，即温度达到10^{218}℃，这是时间与空间差分步长取值出现了问题。这两者应满足关系

$$\frac{\Delta t}{(\Delta x)^2}<\frac{1}{2}$$

才能稳定。（安妮·戈林鲍姆《数值方法：设计、分析和算法实现》P305式14.23）

如果采用隐式格式，则无需考虑这一问题。隐式方法的递推过程在论文图3已经讲述得很完善了，将线性方程组改写成矩阵格式

$$\boldsymbol{Au}=\boldsymbol{B}$$

之后理论上可以直接使用matlab把向量$\vec{u}$解出来：

```
u=B/A
```

但由于$\boldsymbol{A}$是一个三对角矩阵，有更快的方法求解，matlab函数如下。

```
function [ x ] = Chase_method( A, b )
    %Chase method 追赶法求三对角矩阵的解
    %   A为三对角矩阵的系数，b为等式右端的常数项，返回值x即为最终的解
    %   注：A尽量为方阵，b一定要为列向量
    %% 求追赶法所需L及U
    T = A;
    for i = 2 : size(T,1)
        T(i,i-1) = T(i,i-1)/T(i-1,i-1);
        T(i,i) = T(i,i) - T(i-1,i) * T(i,i-1);
    end
    L = zeros(size(T));
    L(logical(eye(size(T)))) = 1;   %对角线赋值1
    for i = 2:size(T,1)
        for j = i-1:size(T,1)
            L(i,j) = T(i,j);
            break;
        end
    end
    U = zeros(size(T));
    U(logical(eye(size(T)))) = T(logical(eye(size(T))));
    for i = 1:size(T,1)
        for j = i+1:size(T,1)
            U(i,j) = T(i,j);
            break;
        end
    end
    %% 利用matlab解矩阵方程的遍历直接求解
    y = L\b;
    x = U\y;
end
```

参数确定

如果每个温区有自己的参数D和σ，那么恭喜你，你有26个参数需要确定。我们在多次调整后，将小温区1~4、小温区10~11分别合并为一组(D,σ)，这样我们的参数有18个。理论上来说你应当使用优化算法进行求解，构造一个函数用来评价曲线与标准曲线的误差，就像我们论文里写的那样，但是这样的结果是，求解耗费时间，最终的曲线也不一定能贴合得很好，代码改好几轮还是改不好。我们实际上采用的方法是*大师纯手工调参*，即从炉前往炉后，一步步调整参数，比较实际曲线和标准曲线，用眼睛目测参数应该往大调还是往小调。这样做就要求调参大师必须非常了解D和σ的作用，这两个参数是怎样影响曲线的。

σ是表面传热系数，σ的值越大，焊接区域的表面温度就越接近于外界温度（即炉内温度），也就是说，焊接区域表面的温度对炉内温度改变的响应越迅速。例如，刚进入炉内时，环境温度迅速上升，如果σ的值较小，那么焊接区域表面的温度上升得就比较慢，可能出炉了还没升到100℃。

D是复合常数：$D=\frac{k}{c\rho}$，其中k是热导率，c是电路板比热容，ρ是电路板密度。看似复杂，但你只需要知道，D越大，焊接区域**内部**的温度传递越快。这里给“内部”加粗，意味着D和外界环境无关，也就是说，炉内的热量传递到电路板中央，首先需要看σ，把热量传递到电路板表面，然后看D，把电路板表面的热量向内部传递。

总而言之，如果曲线低了，就调大σ和D，如果高了，就调小。

问题二

模型建立与求解

显然，要建立优化模型，找到满足条件的最大传送速度。传送速度即是决策变量又是约束条件，因此，一个二分法就可以满足要求。二分法要求“目标函数”是随决策变量改变的单调函数，因此，我们给的目标函数实质上是

$$f=\begin{cases}1, & \text{满足约束条件},\\ 0, & \text{不满足约束条件}.\end{cases}$$

速度太慢也会导致制程界限表内3、4行不满足，因此目标函数整体上并不满足单调的要求，我们需要选取其中单调的一部分进行二分查找，因此我们需要先坐一轮大步长搜索，找到二分查找范围，如论文表7所示。

问题三

模型建立与求解

阴影部分的面积要最小，那么就是一个单目标优化模型。要确定传送带的过炉速度和各温区设定的温度，那么这些统统是决策变量。考虑到决策变量多，约束条件和目标函数复杂，按照我们队的调性，直接套遗传算法（GA），谁也拦不住。

问题四

模型建立与求解

又多了一个目标，是多目标优化模型了。考虑到这已经是问题四了，怎么简单怎么来，直接构造一个曲线对称评价指标，和面积进行一个加权求和。理论上来说，要给两个目标函数分别归一化，权重也要不断尝试，但是，这是问题四！！权重直接随便给，归一化的事在论文里随便说说就行了。

总结

问题一是整个模型的难点，并且问题会越做越多。怎么构造模型？待定参数选哪些？参数怎么确定？因此，按照我们队的调性，在发题当天就敲定问题一的做法，并且完成70%左右工作量。虽然开局熬夜可能不利于后期的奋斗，但我认为，这是比较稳妥的方法，万一哪年A题出来五个问题呢。

CUMCM2022A 波浪能最大输出功率设计

模型 常微分方程组 解法 四阶RK

问题一

模型建立与求解

分别对浮子和振子做受力分析，得描述浮子和振子垂荡运动的微分方程组：

$$\begin{cases} f\cos\omega t-\rho g x_1\pi R_0^2+k(x_2-x_1)+\beta(\dot{x}_2-\dot{x}_1)-\lambda\dot{x}_1=(m_1+m_f)\ddot{x}_1, \\ -k(x_2-x_1)-\beta(\dot{x}_2-\dot{x}_1)=m_2\ddot{x}_2. \end{cases}$$

设初始状态下浮子、振子位移和速度均为0：

$$\begin{cases} x_1(0)=0, \\ x_2(0)=0, \\ \dot{x}_1(0)=0, \\ \dot{x}_2(0)=0. \end{cases}$$

由于四阶龙哥库塔（RK）算法只能用于求解一阶微分方程，因此对二阶微分方程组降阶：

$$\begin{cases} \dot{x}_1=x_3, \\ \dot{x}_2=x_4, \\ \dot{x}_3=\frac{f\cos\omega t-\rho g x_1\pi R_0^2+k(x_2-x_1)+\beta(x_4-x_3)-\lambda x_3}{m_1+m_f}, \\ \dot{x}_4=\frac{-k(x_2-x_1)-\beta(x_4-x_3)}{m_2}, \\ x_1(0)=0, x_2(0)=0, x_3(0)=0, x_4(0)=0. \end{cases}$$

使用四阶RK算法求解：

```
[t,m]=ode45('Q1_func1',t0:0.2:tf,[0,0,0,0]);
```

```
% 以下提供一种可行的四阶RK matlab自定义函数及其用法案例
% 原文链接https://blog.csdn.net/Ruins_LEE/article/details/127105046
clc;clear;close all;

%% 主函数
% 时间变量
t0 = 0;
tf = 1000;
% 初值
y10 = 1;
y20 = 1;
y30 = 3;
y = [y10,y20,y30];
% RK 算法步长
h = 0.25;
% 对该微分方程组用ode45和自编的龙格库塔函数进行比s较，调用如下：
```

```m
% ode45()函数
tic;
[T,F] = ode45(@fun,[t0 tf],y);
time_record_ode = toc;
toc;

% 自编RK函数
tic;
[T1,F1]=runge_kutta1(@fun,y,h,t0,tf);
time_record_rk = toc;
toc;

%% 画图
figure('color',[1 1 1],'position',[600,100,500*1.5,416*1.5]);
subplot(121);
plot(T,F,'LineWidth',1.5);
title(['ode45','($t_f$=',num2str(tf),')'],'Interpreter','Latex');
set(gca,'FontSize',15,'FontName','Times New Roman','LineWidth',1.5);

subplot(122);
plot(T1,F1,'LineWidth',1.5);
title(['RK4','($t_f$=',num2str(tf),'),
($h$=',num2str(h),')'],'Interpreter','Latex');
set(gca,'FontSize',15,'FontName','Times New Roman','LineWidth',1.5);

% 保存数据
str = ['tf_',num2str(tf),'_h_',num2str(h),'.mat'];
str_fig = ['tf_',num2str(tf),'_h_',num2str(h),'.jpg'];
save(str);
saveas(gcf,str_fig)
%% 子函数部分
% 微分方程
function dy = fun(t,y)
dy = zeros(3,1);%初始化列向量
dy(1) = y(2) * y(3);
dy(2) = -y(1) + y(3);
dy(3) = -0.51 * y(1) * y(2);
end

%--- RK 算法 ---%
% ufunc - 微分方程组的函数名
% y0 - 初始值
% h - 步长
% a - 初始时间
% b - 末端时间
function [x,y]=runge_kutta1(ufunc,y0,h,a,b)
    % 步数
    n = floor((b-a)/h);
    % 时间起点
    x = zeros(n,1);
    x(1) = a;
    % 初始值
    y(:,1)=y0;
    % 算法开始
    for iter=1:n
        % 时间变量
```

```
        x(iter+1) = x(iter)+h;
        % 开始迭代
        k1 = ufunc(x(iter),       y(:,iter));
        k2 = ufunc(x(iter) + h/2, y(:,iter) + h*k1 / 2);
        k3 = ufunc(x(iter) + h/2, y(:,iter) + h*k2 / 2);
        k4 = ufunc(x(iter) + h,   y(:,iter) + h*k3 );
        % 得到结果
        y(:,iter+1) = y(:,iter) + h * (k1 + 2*k2 + 2*k3 + k4) / 6;
    end
end
```

```python
# 以下以下提供一种可行的四阶RK python自定义函数及其用法案例，基于上述matlab代码重构而成
import numpy as np


dim = 3 # 带求解方程所含自变量（x）个数
def func(t, y):
    """
    待求解的一阶微分方程（组），高阶微分方程（组）需降阶为一阶微分方程组
    :param t: 时间
    :param y: 函数y（np.array对象）
    :return: 函数y的一阶导函数dy（np.array对象）
    """
    dy = np.zeros(dim)
    dy[0] = y[1] * y[2]
    dy[1] = -y[0] + y[2]
    dy[2] = -0.51 * y[0] * y[1]
    return dy


def rk(ufunc, y0, h, a, b):
    """
    龙格库塔算法函数
    :param ufunc: 待求解函数
    :param y0:y初始值（np.array对象）
    :param h:步长
    :param a:时间下限
    :param b:时间上限
    :return:两个变量，分别为x,y（np.array对象）
    """
    n = int((b - a) // h) # 迭代次数
    x = np.array([a + h * i for i in range(n+1)])
    y = np.zeros((dim, n+1)) # [0 for _ in range(dim)] for _ in range(n+1)]
    y[:,0] = y0
    for i in range(n):
        # 开始迭代
        k1 = ufunc(x[i],           y[:,i])
        k2 = ufunc(x[i] + h / 2,   y[:,i] + h * k1 / 2)
        k3 = ufunc(x[i] + h / 2,   y[:,i] + h * k2 / 2)
        k4 = ufunc(x[i] + h,       y[:,i] + h * k3)
        y[:,i+1] = y[:,i] + h * (k1 + 2*k2 + 2*k3 + k4) / 6
    return x,y


# 主函数开始
```

```
y0 = np.array([1, 1, 3])
h = 0.25    # 步长
t0 = 0      # 初始时间
tf = 10   # 终止时间
x, y = rk(func, y0, h, t0, tf)
print(x, y)
```

问题二

模型建立

输出功率由公式

$$P = Fv$$

计算，这里F是阻尼器的张力$F_{阻}$，v是浮子和振子的相对速度$|\dot{x_2} - \dot{x_1}|$。

求t_0开始一段时间T内的平均功率，即积分后除以积分区间范围：

$$\bar{P} = \frac{1}{T}\sum_{i=t_0/\Delta t}^{(t_0+T)/\Delta t} \alpha[\dot{x_2}(i) - \dot{x_1}(i)]^{2+k}\Delta t$$

Δt是时间步长，为了精确点，我们取了0.01s。

模型求解

这个模型求解就比较有意思了，我们一开始打算直接套遗传算法的，但是先看看搜索区域内的目标函数大概长什么样子（论文图5，大步长搜索）。

显然，这是一个单峰函数，那么就有更适合他的解法了。

对于第一小问，只有一个决策变量，目标函数和决策变量是单峰关系，因此可以使用黄金分割法查找。黄金分割法和三分查找类似，但和二分查找不同，后者适用于单调函数，而前者适用于单峰函数。

对于第二小问，有两个决策变量，我们使用了“从好点出发法”，没错，就叫这个名字。。。算法流程如论文图7，本质上也是不断缩减搜索空间。

这些方法都属于优选法，在人教版高中数学选修4-7《优选法与试验设计初步》里都有提及，感兴趣的可以看看。

这个方法快得很，我印象中不到10分钟就跑出来了，遗传要跑几个小时（因为GA里套了RK）。

问题三

问题一plus，在问题一的基础上再对力矩进行一个分析。这里我们算出来的结果和优秀论文相比有点误差，但是，这是问题三！！who cares?

公式的推导在论文里已经很详细了，流程和问题一大同小异，这里就不再赘述了。

问题四

问题二plus，在问题二的基础上把纵摇的功率直接加上去。进行大步长搜索，发现还是单峰函数（论文图13），接着套优选法。

总结

学校培训的时候强调少用智能优化算法。一些题目的决策变量要成百上千了，那毫无疑问是要用智能算法的。但是，这回最多只有两个决策变量，求解方法就很多了。秉承着少用智能算法的态度，我们采用了“从好点出发法”，实际上，梯度下降（GD）等方法也可以尝试（我们尝试过，由于目标函数不是解析函数，会有一些误差，即峰值处不光滑）。

```
% 以下提供一个GD matlab自定义函数
%% 梯度下降
function [result,times,way,endgrad]=GD(func,x0,xmax,xmin,alpha,endcon,way_record)
% 梯度下降函数
% 输入：func-目标函数，x0-初始点（行向量），xmax-定义域上限（行向量），xmin-定义域下限（行向量），
%       alpha-学习率，endcon-允许的最大梯度（结束条件），way_record-路径记录开关（bool）
% 输出：result-结果，times-下降次数，way-路径，endgrad-结束时梯度范数
    h=0.1;  % 数值梯度步长
    times=0;
    [~,dim]=size(x0);
    way=[];
    while true
        times=times+1;
        %% 计算梯度
        grad=zeros(1,dim);
        fx0=func(x0);
        for idx=1:dim
            x1=x0;
            x1(idx)=x0(idx)+h;
            grad(idx)=(func(x1)-fx0)/h;
        end
        %% 记录路径
        if way_record
            way(end+1,:)=[x0,fx0];
            plot3(x0(1),x0(2),fx0,"r*")
        end
        %% 判断是否结束
        endgrad=norm(grad);
        if endgrad<endcon
            result=x0;
            return
        end
        %% 修改位置
        x0=x0-grad*alpha;
        if sum(x0>xmax)+sum(x0<xmin)>0
            result=x0;
            times=-1;
            return
        end
    end
end
```

CUMCM2023A 定日镜场优化设计

模型 几何优化模型 解法 坐标变换

我队参与了2023年全国大学生数学建模竞赛，完成A题，获省二等奖。

2024年夏，我队对模型进行了完善，以下展示的是完善后的过程。

问题一

定日镜法向确定

我们队采用了比较“朴素”的方法：正余弦定理。我们对每种情况进行分类讨论，利用“太阳中心点发出的光线经定日镜中心反射后指向集热器中心”这一条件，确定镜场内各个定日镜的法向。显然，这里采用向量计算更为合理，也无需分类讨论了。

总之，一通计算后，能够得到定日镜的镜面法向量的高度角α_{m}和方位角γ_{m}。

定日镜布阵

精彩的来了。我们队的配置是三个物理专业学生，因此不难发现，定日镜镜面的高度角和方位角恰是《理论力学》中的进动角和章动角。

参考周衍柏《理论力学》（第五版）P121，刚体的旋转可以使用三个角度完全描述，分别是进动角、章动角、自转角，三者合称欧拉角。这三个角度分别描述刚体的进动、章动和自转，旋转顺序不可颠倒。进动、章动、自转分别可以使用矩阵描述：

$$
\text{进动}\boldsymbol{T_1}=\begin{pmatrix}\cos\varphi & \sin\varphi & 0\\ -\sin\varphi & \cos\varphi & 0\\ 0 & 0 & 1\end{pmatrix}
$$

$$
\text{章动}\boldsymbol{T_2}=\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\theta & \sin\theta\\ 0 & -\sin\theta & \cos\theta\end{pmatrix}
$$

$$
\text{自转}\boldsymbol{T_3}=\begin{pmatrix}\cos\Phi & \sin\Phi & 0\\ -\sin\Phi & \cos\Phi & 0\\ 0 & 0 & 1\end{pmatrix}
$$

假定刚体上某点坐标为$\begin{pmatrix}a\\ b\\ c\end{pmatrix}$，经过旋转后，坐标变为

$$
\begin{aligned}
\begin{pmatrix}a'\\ b'\\ c'\end{pmatrix}&=\boldsymbol{T_3T_2T_1}\begin{pmatrix}a\\ b\\ c\end{pmatrix}\\
&=\begin{pmatrix}\cos\Phi & \sin\Phi & 0\\ -\sin\Phi & \cos\Phi & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ 0 & \cos\theta & \sin\theta\\ 0 & -\sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}\cos\varphi & \sin\varphi & 0\\ -\sin\varphi & \cos\varphi & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}a\\ b\\ c\end{pmatrix}
\end{aligned}
$$

在本题中，我们的定日镜需要经过进动、章动和平移。假定操作前镜面的初始位置是中心位于原点，镜面朝向正北：

$$
\begin{cases}x=a,\\ y=0,\\ z=b.\end{cases}
$$

其中，$a \in [-w/2, w/2], b \in [-h/2, h/2]$，是描述镜面平面的参数。

那么，经过旋转与平移变换后的镜面方程就变成了

$$\begin{pmatrix} x_{\mathrm{m}} \\ y_{\mathrm{m}} \\ z_{\mathrm{m}} \end{pmatrix} = \boldsymbol{T_2 T_1} \begin{pmatrix} a \\ 0 \\ b \end{pmatrix} + \begin{pmatrix} x \\ y \\ h_1 \end{pmatrix}$$

式中，(x, y)是镜面中心坐标，h_1是安装高度。

不要害怕不要哭，这个式子不是让你算的，是让matlab或者numpy算的。

阴影遮挡

阴影遮挡分两方面，一是光线被吸收塔遮挡，二是光线被其他定日镜遮挡。根据评阅要点，吸收塔阴影导致的镜场太阳辐照度的损失小于千分之一，可忽略。

和我们论文中写的一样，假定定日镜A参与光路反射，定日镜B可阻挡光线。我们对定日镜A上的入射光和反射光做粗颗粒化处理，即对镜面进行网格离散，只有离散节点上会反射光照。当网格划分足够密的时候，我们有理由认为采样光线被阻挡的频率等于所有光线被阻挡的概率：$\eta_{\mathrm{sb}} = \frac{\text{未被遮挡的采样光线数}}{\text{总采样光线数}}$。

针对每个镜A上的采样点，和太阳连线得入射光方程，反射后得反射光方程。两者之一和定日镜B矩形相交，则这一条光路被阻挡。

联立光路和定日镜B矩形方程，得到线性方程组

$$\begin{cases} (\cos\alpha_{\mathrm{mB}} \sin\gamma_{\mathrm{mB}} + \frac{s_y}{s_x}\cos\gamma_{\mathrm{mB}})a + \sin\alpha_{\mathrm{mB}} b + M_y - (M_x + x_{\mathrm{B}})\frac{s_y}{s_x} - y_{\mathrm{B}} = 0, \\ (\sin\alpha_{\mathrm{mB}} \sin\gamma_{\mathrm{mB}} + \frac{s_z}{s_x}\cos\gamma_{\mathrm{mB}})a - \cos\alpha_{\mathrm{mB}} b + M_z - (M_x + x_{\mathrm{B}})\frac{s_z}{s_x} - h_{\mathrm{B1}} = 0. \end{cases}$$

使用《线性代数》里的Cramer法则解出a和b，如果满足$a \in [-w/2, w/2], b \in [-h/2, h/2]$，那么光线就被定日镜B阻挡了，因为交点落入了镜B的范围内。

截断效率

为了简化模型，减少计算量，我们只考虑了光锥里的五条光线，其中有一条是轴线。论文里给出的过程已经非常详细了，此处不再赘述。

模型求解

遍历求解。但是，根据论文图13，很明显，第四象限蓝色点的定日镜计算是有误的（颜色理应连续变化），个人感觉是定日镜法向计算过程中存在错误。

问题二、问题三

输出热功率要最大，那么输出热功率就是目标函数。按照优化模型进行构建。具体细节此处不宜多言。

总结

需要说明的是，在国赛时我们忽略了定日镜间光线遮挡，仅考虑了吸收塔对光线的遮挡，所以只有省二。遮挡的考虑和截断效率的计算是问题一的重点和难点，我们国赛论文的完成度实质上只有40%，此外，问题二和问题三的计算结果也不是很乐观，能拿到省二已是侥幸。

23年的A题的最大阻碍是定日镜镜面如何使用数学语言描述，我们使用坐标变换进行布阵，如果不会这个方法，那么整题就全崩了。

问题二、三我们选用GA是理所应当的。一是决策变量十分多，过多考虑如何布阵只会让程序变得更复杂，bug更对；二是我们队在队内培训时智能优化算法只完全掌握了GA，同时多变量的GA是我们自己研究出来的。我们自研的GA缺陷是没能考虑变量间的相互影响，在进化过程中各变量独立了，这回浪费相当大的计算时间。但无论如何，都比线性搜索、多重搜索来得划算，没有哪个队伍会认为让编程手每隔一个小时从被窝里爬起来改那搜索范围和步长是合理的。

CUMCM2024A “板凳龙”闹元宵

模型 几何优化模型 解法 坐标变换

我队参与了2024年全国大学生数学建模竞赛，完成A题，获省一等奖。

论文未做修改，文件展示原始论文。

问题一

龙头位置

查阅少得可怜的文献，得等距螺线方程为

$$r = b\frac{\theta}{2\pi}$$

同时，0到θ的弧长为

$$s(\theta) = \frac{b}{4\pi}\left[\theta\sqrt{1+\theta^2} + \ln\left(\theta + \sqrt{1+\theta^2}\right)\right]$$

不过我希望得到的是θ关于s的关系（也就是上式的反函数），用来根据龙头线速度算θ来确定龙头位置。将上式直接敲到matlab解反函数，没有办法得到解析解。

那就数值解！不用注意，想想都知道$s(\theta)$是严格单调递增的（θ越大，累积的弧长s就越长），那就直接上二分法。这样就得到了$\theta(s)$。

龙头速度恒为v，那么t时刻龙头走过的路程就是vt。设16圈和x轴交点位置对应的弧长为s_0，那么t时刻龙头处于

$$\theta = \theta(s_0 - vt)$$

的位置。有了θ，龙头的位置就唯一确定了。

龙身位置

确定前一把手位置，如何根据龙身长度递推出下一把手的位置，是龙身位置递推的基本问题。换言之，等距螺线上给定一点，如何找到割线长度为某一值的另一点。

我们队的做法是，不解决这个问题。我们直接采取一个近似，把前一把手所在等距螺线视为曲率相同的圆形，求两个圆的交点还不容易吗。

不论使用哪种方法，都会得到两个解，其中，对盘入过程而言，离原点更远的那个解才是下一把手的位置。

龙身速度

由物理规律，$i+1$把手的绝对速度应为

$$v_{i+1} = v_i\frac{\cos\beta_i}{\cos\beta_{i+1}}$$

式中，β是把手前向和螺线切向的夹角。

模型求解

从龙头位置向后递推得到龙身位置，在此基础上，由龙头速度向后递推得到龙身速度。这就是求解的过程。

问题二

模型建立与求解

如何判断是否发生碰撞，是问题二的核心。

我们的判断方法是，是否存在板凳A的顶点与板凳B的矩形重合的情况。也就是说，板凳A的顶点到板凳B横向轴线的距离小于板凳宽度，或板凳A顶点到板凳B纵向轴线的距离小于板凳长度，就发生了碰撞。

如何找到板凳A顶点坐标？毫无疑问，坐标变换。与去年相比，这次的坐标变换是二维的，更简单了。

构造旋转矩阵

$$\boldsymbol{T}=\begin{pmatrix}\cos\gamma & -\sin\gamma\\ \sin\gamma & \cos\gamma\end{pmatrix}$$

描述逆时针旋转γ的过程。先旋转，后平移，得到板凳顶点的坐标为

$$\begin{pmatrix}x_i^{'}\\ y_i^{'}\end{pmatrix}=\begin{pmatrix}\cos\gamma & -\sin\gamma\\ \sin\gamma & \cos\gamma\end{pmatrix}\begin{pmatrix}x_i\\ y_i\end{pmatrix}+\begin{pmatrix}x_A\\ y_A\end{pmatrix}$$

式中，$(x_i,y_i)=(\pm w/2,\pm l/2)$，是板凳A四顶点相对于中心的坐标，$(x_A,y_A)$是板凳A中心相对于大地的坐标。

由点到直线的距离公式，完成碰撞发生的判断。

问题三

模型求解

螺距缩小到一定程度时，约束条件不再满足。因此，在问题二的基础上直接套一个二分，找到最小的螺距即可。

问题四

模型建立与求解

这题有歧义。歧义来源于一眼就能看出掉头空间固定时，S弯的入点和出点就固定，掉头曲线也就唯一缺点。因此，我们队和其他很对队伍一样，一开始认为是需要缩小掉头空间，找到在S弯内不发生碰撞的最小掉头空间半径，此时对应的曲线即为最短曲线。

在经过一段时间的思考后，我们没有按照上面的过程来做。因为，考虑到题目后半部分需要给出完整数据excel表，如果真如前文所述建立优化模型，那么每个队伍得到的excel就会完全不同，评阅过程也就无法根据这一数据直接打分。所以，得出的结果理应是每个队伍近乎一致的，也就是说，掉头空间不能变，这题不是优化问题。这题真正的难点在于S弯和盘出过程龙身位置、速度的递推。

我们队通过分类讨论各种情况，得到了S弯内和盘出过程龙身位置、速度的递推模型，同时计算得到了较为合理的结果。

问题五

模型建立与求解

这才是优化问题，是问题四对应的优化问题。行进速度超过某一最大值后，约束条件不再成立，因此，还是二分法。

总结

问题一如何递推龙身的位置和速度，问题二如何判断板凳龙是否发生了碰撞，是我们解决的主要问题。递推龙身位置时，我们采用了曲率圆进行近似，这会丢失一定精度，且越靠近中心，误差会越大。这导致我们队伍计算的结果和其他队伍略有差异。不过，我们采用了最简单、最朴素的方法比较完美地解决了这些问题，这也是我们从去年到今年态度的转变。去年我们认为应当采用最高大上、最完美的方法求解模型，甚至在训练时C题一个简单的拟合，我们都用上了BP神经网络。时间磨平了我的棱角，对于一个喜欢偷懒的我来说，现在我只追求怎样用最简单、最适合的办法来解题了。