

## 高压燃油控制问题

## 摘 要

高压油管是燃油发动机供油系统中连接喷油泵与喷油器的重要零部件,其内部压力的变化会影响发动机的工作效率。对于给定参数规格的高压油管,本文通过建立数学模型,研究了不同进油、喷油方式下油管内压力变化情况。

对于问题一,首先根据题目给定的压力变化量与密度变化量的对应关系,通过数值积分的方式,得到了一定精度下,不同燃油密度对应的燃油压力。建立进出油过程中油管内部压力变化的微分方程,即高压油管内部压力变化的微分模型在此基础上研究了油管系统随时间的波动情况和稳态性质。将单向阀开启时间作为决策变量,以稳态阶段油管燃油压力与期望压力偏差的平方和为目标函数,建立基于油管压力变化微分模型的优化模型,并将微分模型改写成差分形式,通过编程遍历求解得到最优单向阀开启时长为 0.288ms。通过对模型稳态的研究,将期望压力设定为 150MPa,结合上述微分模型,通过数值模拟求解得出,调整时间为 2s 的策略为 0 到 2s 内单向阀每次开启时长为 0.90ms,随后开启时长变为 0.765ms;调整时间为 5s 的策略为 0 到 5s 内单向阀每次开启时长为 0.760ms,随后开启时长变为 0.765ms;调整时间为 10s,由于达到原方法的阈值极限,我们使用了一次函数策略动态调整进油速率,最终确定调整方案为 0 到 4.1s 内单向阀每次开启时长为  $0.041t+0.4$ ms,随后开启时长变为 0.765ms。

对于问题二,针对重新定义的高压油泵供油方式和喷油嘴的出油方式,首先研究柱塞腔内压力变化与凸轮转动的关系,建立柱塞腔内压力变化的微分模型。接着分析针阀运动,确定喷油嘴有效面积与针阀运动的关系。最后,建立油管系统压力变化的微分模型,并将其转化为差分形式,以凸轮转动角速度作为决策变量建立优化模型。通过编程遍历求解得出凸轮转动角速度的最优值为 0.0273rad/ms。

对于问题三,将两个喷油嘴工作时间的间隔和凸轮转动角速度共同作为决策变量,重新定义出油量,建立了两个喷油嘴的高压油管系统二元最优化模型。通过数值模拟的方法得到凸轮角速度最优值为 0.069rad/s,喷油嘴工作时间间隔为 4ms。进一步考虑安装减压阀后的情况,对于含泄压阀的油管系统,将泄压阈值作为新的决策变量,改进上一小问的模型,通过数值模拟求解得到一个最优解为压力阈值取 100MPa,凸轮角速度为 0.045 rad/ms。

最后,给出了模型的优缺点和改进方法。

**关键词:** 常微分方程 差分法 数值模拟 数值积分 最优化模型

# 1 问题重述

燃油发动机中的高压油管连接了喷油嘴、高压油泵两部分，其工作过程分为向外喷油和向内注油两部分。燃油进入和喷出的间歇性工作过程会导致高压油管内压力的变化，使得所喷出的燃油量出现偏差，从而影响发动机的工作效率。为此，我们需要设计高压油管、喷油嘴和高压油泵的行为，以控制高压油泵内的压力。具体需要解决如下问题：

## 问题一：

问题 1 分为两个要求：给定高压油管型号，内腔长度、内径，供油入口直径给定喷油嘴工作频率、喷油方式，单向阀打开后的关闭时间，喷油压力：

- 要求通过设置单向阀每次开启的时长，使得高压油管内的压力尽量稳定在 100MPa。
- 要求调节单向阀的开启时间，使得分别经过 2s、5s、10s 的时间后，高压油管内压力由 100MPa 升至 150MPa，并尽量稳定在 150MPa。

## 问题二：

在实际工作过程当中，出油口为一针阀控制的喷油嘴，进油口为一凸轮控制的高压油泵。凸轮匀速转动带动柱塞运动，使得高压油泵中的压强产生周期性的变化。当高压油泵中的油压大于高压油管中的油压时，单向阀打开，油从高压油泵进入高压油管。给定柱塞枪各参数，柱塞运动到上、下止点时柱塞内的残余容积和低压燃油压力。在问题一给出的喷油器工作次数，高压油管尺寸，初始压力条件下，确定凸轮的角速度，使得高压油管内的压力尽量稳定在 100MPa。

## 问题三：

问题 3 包含两个小问：

- 在问题 2 的基础上，再加一个喷油嘴，两喷油嘴喷油规律相同，如何调节喷油和供油策略。
- 在高压油管上安装一个减压阀，高压油管中的油可以通过减压阀流回，以减小高压油管内的油压，要求给出高压油泵和减压阀的控制方案。

# 2 问题分析

## 2.1 问题一的分析

问题一的第一小问要求控制高压油管内的压力恒定在 100MPa 下，题目一定义了一个尺寸恒定的容器，此时，压力通过燃油质量与进出流量相联系。同时，题目一给定了喷油嘴的行为，进油压力的大小，系统唯一可改变的量为单向阀开启的时长。于是我们考虑以单向阀开启时长为决策变量，建立微分方程，优化该问题。

问题一的第二小问要求将高压油管内压力由 100MPa 在给定时间内调节到 150MPa 并稳定在该压力。一方面，我们可以进一步调节单向阀的控制策略，使得管内压力可以稳定在 150MPa，另一方面，我们需要使得压力可以在给定时间内尽可能精确地上升到 150MPa。

## 2.2 问题二的分析

第二问重新定义了出油方式和进油方式，其中，进油由高压油泵子系统实现，凸轮角速度为唯一决策变量，我们需要用合理的方程描述该系统的行为，并通过控制凸轮角速度实现该子系统功能。

## 2.3 问题三的分析

第一小问在第二问的基础上将出油子系统定义为两个出油方式相同的喷油阀，决策变量变为两喷油嘴喷油间隔与凸轮角速度。我们需要以此为基础建立双变量优化模型。

第二小问在前一小文的基础上定义了一个泄压阀，由于泄压阀的行为定义较为模糊，我们需要对其行为进行定义，并将其作为决策变量。

# 3 模型假设

1. 高压油管是刚体，不考虑其形变;
2. 高压油管的体积是理想圆柱体体积;
3. 不考虑高压油管内壁对油的粘滞力;
4. 不考虑液体压力波动和压力传播的时间，腔内液压处处相等; 整个系统温度恒定;
5. 不考虑供油入口和喷油口的阀门开关的延迟时间。

# 4 符号说明

符号	含义
$P(t)$	$t$ 时刻高压油管的压力
$m(t)$	$t$ 时刻高压油管内的压力
$\rho(t)$	$t$ 时刻高压油管储油总质量
$I(t)$	$t$ 时刻进油速率
$E(t)$	$t$ 时刻出油速率

(部分符号将在下一部分说明)

符号	含义
$T$	单向阀开启时间
$T_{100}$	稳态 100MPa 时, 单向阀开启时间
$T_{150}$	稳态为 150MPa 时, 单向阀开启时间
$t_e$	期望的压力
$P_e$	期望的稳态压力
$P_{low}$	喷油嘴及减压阀外界的压力
$t_n$	系统达到稳态所需的压力
$[t_{n1}, t_{n2}]$	系统稳态后的某段周期整数倍的时间
$T_{lcm}$	喷油周期与供油周期的最小公倍数的某一倍数
$\rho_c, P_c, m_c, V_c$	柱塞腔内燃油密度, 压力, 质量, 体积
$\omega$	凸轮转动角速度
$T_C$	凸轮转动的周期
$f_1(t), f_2(t)$	进出油的状态变量
$P_R$	泄压阀阈值压力

## 5 模型建立与求解

### 5.1 问题一的建模与求解

#### 5.1.1 模型的建立

**基本模型：高压油管压力变化微分方程** 我们将高压油管内的压力作为研究对象，研究其在一段连续时间内的动态变化。

以单向阀第一次开启时间为 0 时刻，设单向阀每次打开时间为  $T$ ，首次喷油时刻为  $t_0$ ， $t_0 \in [0, 97.6]$

另设  $t$  时刻高压油管的压力为  $P(t)$ ，储油总质量为  $m(t)$ ，燃油密度为  $\rho(t)$ ；设高压油管的容积恒为  $v$ ， $\rho_1$  为 160MPa 下的燃油密度。

描述高压油泵进油时间段：

$$\Omega_{1,1}, \Omega_{1,2}, \dots, \Omega_{1,n_1}$$

$$\Omega_{1,n} = [t_0 + 100 \cdot (i - 1), 2.4 + t_0 + 100 \cdot (i - 1)], i = 1, 2, \dots, n_1$$

描述喷油嘴出油时间段：

$$\Omega_{2,1}, \Omega_{2,2}, \dots, \Omega_{2,n_2}$$

$$\Omega_{2,i} = [t_0 + 100 \cdot (i - 1), 2.4 + 100 \cdot (i - 1)], i = 1, 2, \dots, n_2$$

于是可以得到  $t$  时刻进油速率  $I(t)$  和出游速率  $O(t)$  的分段函数：

$$I(t) = \begin{cases} Q_1(t) & t \in \Omega_{1,i} \\ 0 & t \in U \setminus \Omega_{1,i} \end{cases} \quad (1)$$

$$O(t) = \begin{cases} Q_2(t) & t \in \Omega_{2,i} \\ 0 & t \in U \setminus \Omega_{2,i} \end{cases} \quad (2)$$

其中

$$Q_1(t) = CA\sqrt{\frac{2\Delta P(t)}{\rho_1}} \quad (3)$$

$$Q_2(t) = \begin{cases} 100t' & t' \in [0, 0.2) \\ 20 & t' \in [0.2, 2.2) \\ 240 - 100t' & t' \in [2.2, 2.4) \end{cases} \quad (4)$$

$$t' = t - t^0 + 100 \cdot (i - 1) \quad (5)$$

考虑  $t$  时刻到  $t + t_0$  时刻高压油管中燃油的质量，有

$$m(t + dt) - m(t) = (\rho_I \cdot I(t) - \rho(t) \cdot E(t))dt$$

于是

$$\frac{dm}{dt} = \rho_I \cdot I - \rho(t) \cdot E \quad (6)$$

另考虑系统的初始时刻状态

$$m(0) = \rho(0) \cdot V \quad (7)$$

其中  $\rho(0)$  为  $0.850\text{mg/mm}^3$

**系统的波动与稳态** 考虑从 0 时刻到无限远时刻之间系统的变化，由式 (6) 可知，当某一时刻燃油的压力较高，则  $I$  会减小，而整体出油速率不变，于是  $\frac{dm}{dt}$  减小。当某一时刻燃油压力较低，则  $I$  会增加，同理可知  $\frac{dm}{dt}$  增加。

因此，经过一段足够长的时间，高压油管内的压力应趋于一个常数，并维持动态平衡，记系统单位时间内压力的均值为

$$h_p(t) = \frac{\int_{t-T/2}^{t+T/2} P(t)dt}{T}$$

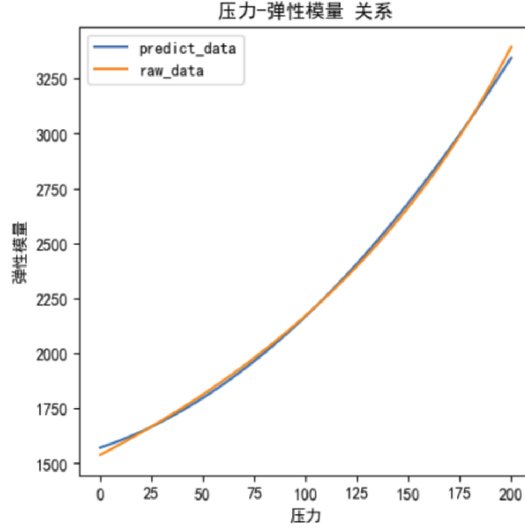
那么，在某一足够大的时刻后

$$h_p(t) \rightarrow \text{constant}$$

并且该时刻后，系统达到稳态，即满足方程：

$$\begin{cases} \frac{dm}{dt} = \rho_I - \rho(t) \cdot E \\ m(nT_{lcm}) = m((n-1)T_{lcm}) \quad n > n_0 \end{cases} \quad (8)$$

其中  $T_{lcm}$  为喷油周期和进油周期的公倍数。



燃油压力  $p$  与密度  $\rho$  的关系 设燃油压力变化量为  $dp$ , 燃油密度变化量为  $d\rho$ , 则由  $dp$  与  $d\rho$  的关系可知

$$dp = \frac{E}{\rho} d\rho$$

于是

$$\int \frac{1}{E} dp = \int \frac{1}{\rho} d\rho$$

而  $E$  是关于  $p$  的函数, 做出  $E-p$  关系图: 使用二次函数拟合并积分, 得到  $p-\rho$  关系式为:

$$p = \frac{\rho - 0.81}{1.86 - 1.70} \times 10^3 \quad (9)$$

建立优化模型 设决策变量单向阀开启时间:

$$T$$

设目标函数为足够长时间后高压油管压力的标准差

$$\min Z = \int_{t_1}^{t_2} (P(t) - P_e)^2 dt \quad (10)$$

约束条件有:

1. 模型满足的基本方程组
2. 模型满足的燃油密度和燃油压力的关系

即

$$\left\{ \begin{array}{l} \frac{dm}{dt} = \rho_I \cdot I - \rho(t) \cdot E \\ p = \frac{\rho - 0.81}{1.86 - 1.70} \times 10^3 m(0) = \rho(0) \cdot V \\ \rho(t) = \frac{m(t)}{V} \\ I(t) = \begin{cases} Q_1(t) & t \in \Omega_{1,i} \\ 0 & t \in U \setminus \Omega_{1,i} \end{cases} \\ O(t) = \begin{cases} Q_2(t) & t \in \Omega_{2,i} \\ 0 & t \in U \setminus \Omega_{2,i} \end{cases} \\ Q_1(t) = CA \sqrt{\frac{2\Delta P(t)}{\rho_1}} \\ Q_2(t) = \begin{cases} 100t' & t' \in [0, 0.2) \\ 20 & t' \in [0.2, 2.2) \\ 240 - 100t' & t' \in [2.2, 2.4) \end{cases} \\ t' = t - t^0 + 100 \cdot (i - 1) \\ \Omega_{1,n} = [t_0 + 100 \cdot (i - 1), 2.4 + t_0 + 100 \cdot (i - 1)], i = 1, 2, \dots, n_1 \\ \Omega_{2,i} = [t_0 + 100 \cdot (i - 1), 2.4 + 100 \cdot (i - 1)], i = 1, 2, \dots, n_2 \end{array} \right. \quad (11)$$

### 5.1.2 模型的求解

我们使用多重搜索算法来寻找使得目标函数最小的  $T$  值：算法的步骤如下：

**Step1:** 本题需要得出一个最优解  $T$ ，使得高压油管内阈值稳定在 100MPa 左右，算法采用多重搜索法，首先假定范围为 (0,10) ms，以 0.1ms 为步长搜索  $T$ ，并计算出对应的  $Z$  值，采用  $Z$  值最小的  $T$ ，根据计算结果可以将范围缩小至 (0, 0.5) ms。

**Step2:** 进一步缩小范围至 (0, 0.5) ms，缩小步长至 0.01ms，重复上述操作。

得到的  $Z$  值如下图所示：

通过多重搜索，我们得到：使得压力均值  $Z$  最小的开阀时间  $T$  为  $T = 0.288ms$ 。

当  $T = 0.288ms$  时，压力的波动如下图所示：

### 5.1.3 第二小问模型的建立

在第一小问中，我们论述了不同初始状态，不同开阀时长  $T$  的系统，最终都会达到稳态，压力值在一个常数附近波动。该结论对任一开阀时长与初始压强都是成立的。

我们根据第一小问的模型，绘制了压强趋于稳定值的过程：

计算稳态时压力的均值，不难证明最后的压力在 100MPa 上下波动。

分析压力的变化过程，我们可以将压力变化过程分为稳定阶段和未稳定阶段。由第一小问的分析，我们应当在压力变化过程中对  $T$  进行动态调整，使其同时满足一定时长的未稳定阶段



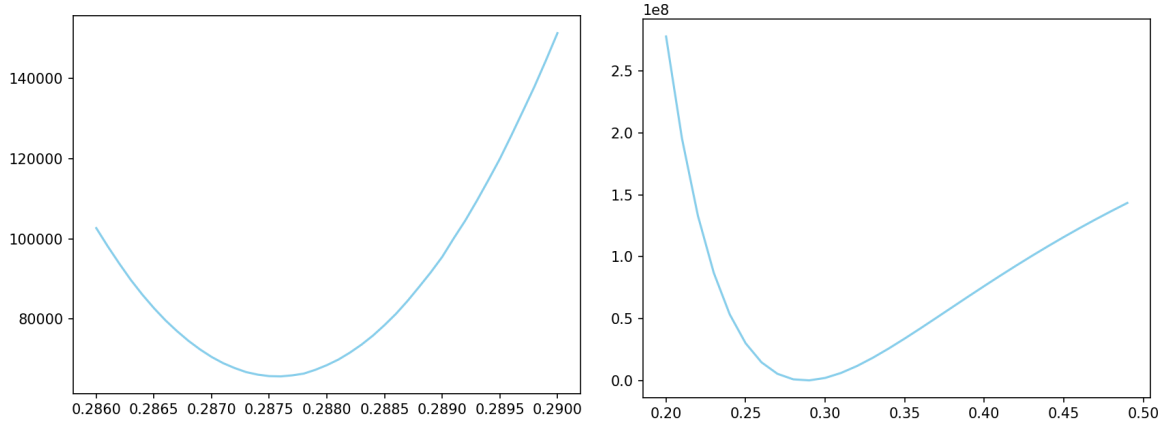


图 1: Z-T 关系图

和 150MPa 的稳定压力。

在第一小问的基础上，将高压油管内压力的期望值设定为  $P_e = 150MPa$ ，初始状态不变，期望时间为  $t_e$ ，喷油嘴开启时刻取  $t_0 = 0$ ；使用第一小问的模型，得到稳定阶段下的开阀时间  $T_m$ 。用  $Z$  来描述一段时间内模型压力稳定在 150MPa 下的能力。

然后，我们分别就不同的未稳定阶段时间，对未稳定阶段的开阀时间  $T$  进行调整，使得在这段时间里，压力恰好调整至目标压力。

## 5.2 第二小问模型的求解

首先按照上述操作，得到  $T_{150} = 0.752ms$

设  $t_e = 2000ms$ ，调整未稳定阶段单向阀开启时间  $T_m$ ，得到不同  $T_m$  下 2s 时的压力值，绘制如下图像：

可以得到，未稳定状态为 2000ms 下，单向阀的调整策略为：

$$T_m = \begin{cases} 0.90ms & t < 2000ms \\ 0.765ms & t > 2000ms \end{cases} \quad (12)$$

设  $t_e = 5000ms$ ，调整未稳定阶段单向阀开启时间  $T_m$ ，得到不同  $T_m$  下 2s 时的压力值，绘制如下图像：

可以得到，未稳定状态为 5000ms 情况下，单向阀的调整策略为：

$$T_m = \begin{cases} 0.760ms & t < 2000ms \\ 0.765ms & t > 2000ms \end{cases} \quad (13)$$

设  $t_e = 10000ms$ ，调整未稳定阶段单向阀开启时间  $T_m$ ，得到不同  $T_m$  下 2s 时的压力值，绘制如下图像：

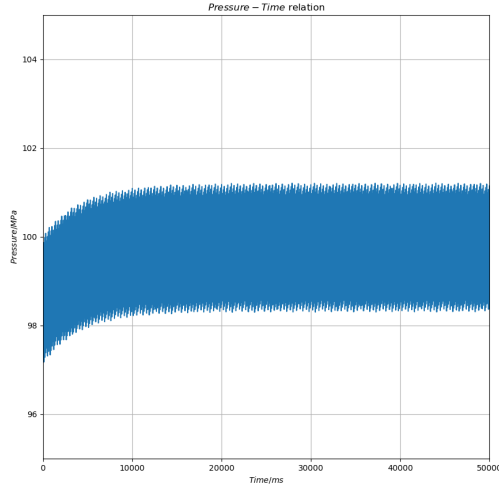


图 2: 第一小问压力-时间关系图

可以得到，未稳定状态为  $10000ms$  情况下，单向阀的调整策略为：

$$T_m = \begin{cases} 0.193ms & t < 2000ms \\ 0.752ms & t > 2000ms \end{cases} \quad (14)$$

### 5.2.1 稳态的检验

## 5.3 问题二的建模与求解

问题二在问题一的基础上，重新定义了高压油泵供油方式，重新给出了喷油阀在一个喷油周期内的工作方式，要求我们重新设置高压油泵的行为，即在新的约束条件下重新确定  $I(t)$  和  $O(t)$  函数，使得压力稳定在  $100MPa$ 。

### 5.3.1 进油子系统模型的建立

供油的状态首先关联于高压油泵子系统和高压油管系统的压力差，我们以凸轮的旋转为研究对象，进而确定高压油泵的压力表达式。

凸轮轮廓线的拟合直接服务于其运动规律的反解，即凸轮角度与柱塞行程对应关系的反解，由附件 1 所给的凸轮边缘曲线坐标，得到极坐标形式下转角与极径的对应关系，使用插值法进行处理。

完成插值以后，使用各个插值区间的三次多项式函数求得凸轮上  $[0, 2\pi]$  各个角度  $\theta$  对应的极径  $r$ ，以水平和竖直方向建立直角坐标系，所对应的直角坐标由以下公式计算：

$$\begin{cases} x = r \cos \theta \\ y = r \sin \theta \end{cases} \quad (15)$$

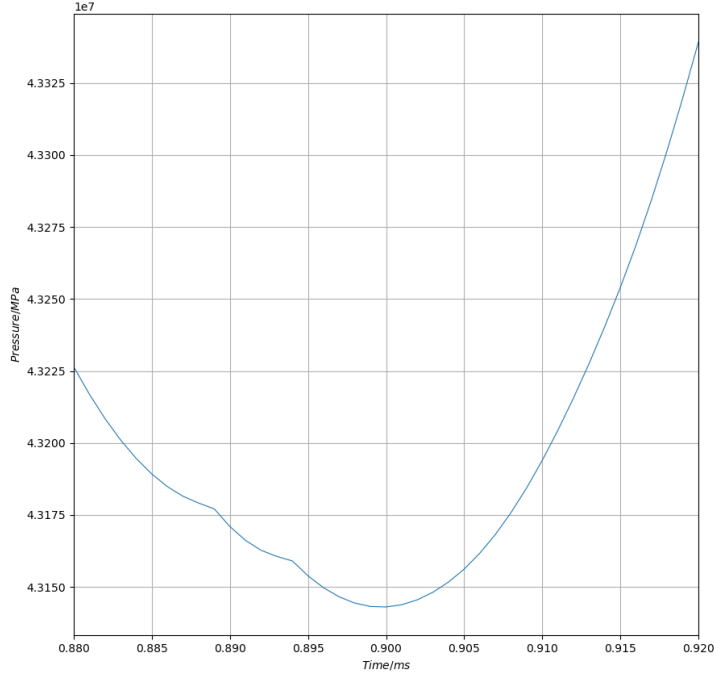


图 3: 2s 未稳态开阀时间-2s 时压力关系图

对于驱动柱塞的凸轮机构，如图 7 所示，O 为盘型凸轮的基圆圆心位置，M 为某时刻凸轮的最高点，柱塞底部与 x 轴平行，故 M 点也是凸轮与柱塞底部的接触点。

在初始位置下转角为  $\phi_0 = 0$ ；凸轮球形底端接触柱塞底面时，即 1 柱塞运动到下止点时低压燃油充满柱塞腔，凸轮边缘曲线横坐标  $X$  为

$$X_0 = [x_{1,0}, x_{2,0}, \dots, x_{m,0}, x_{n,0}]$$

凸轮边缘曲线纵坐标  $Y$  为

$$Y_0 = [y_{1,0}, y_{2,0}, \dots, y_{m,0}, y_{n,0}]$$

对凸轮旋转  $k$  角度后，得到凸轮边缘个点坐标的函数表达式

$$\begin{cases} x_{m,k} = x_{m,0} \cdots \cos \phi_k - y_{m,0} \cdots \sin \phi_k \\ y_{m,k} = x_{m,0} \cdots \sin \phi_k - y_{m,0} \cdots \cos \phi_k \end{cases} \quad (16)$$

则凸轮旋转  $k$  角度时，凸轮轮廓的横坐标为：

$$X_m = [x_{1,m}, x_{2,m}, \dots, x_{n,m}]$$

凸轮轮廓的纵坐标为：

$$Y_m = [y_{1,m}, y_{2,m}, \dots, y_{n,m}]$$

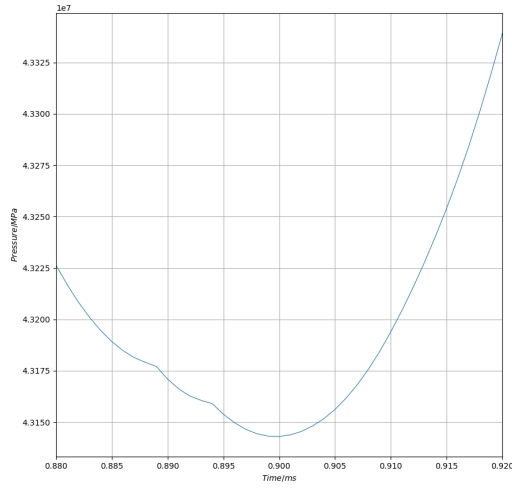


图 4: 5s 未稳态开阀时间-5s 时压力关系图

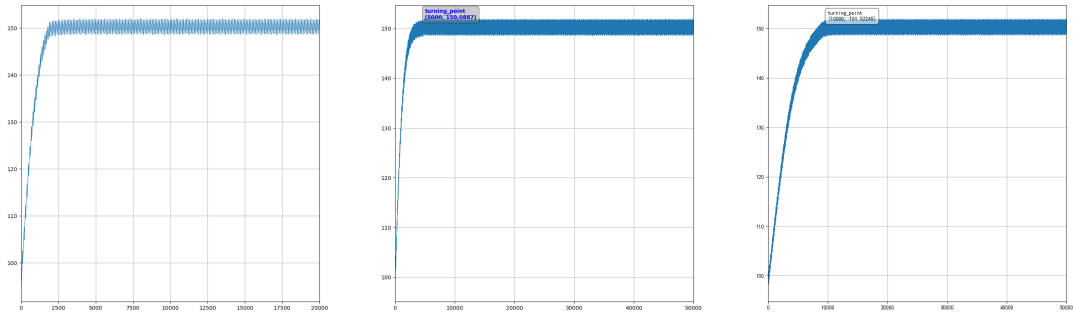


图 5: 2s,5s,10s 压力-时间关系图

设凸轮的基圆半径为  $R_0$ ，则当凸轮旋转角度为  $\phi_k$  时，凸轮的升程为

$$s_k = \max(Y_k) - R_0 \quad (17)$$

由于柱塞腔底面直径为  $5mm$  设柱塞腔底面面积为  $S$ ，由于凸轮转角为  $\phi_0$  时柱塞的升程为  $0$ ，则凸轮转角为  $\phi_k$  时，高压油泵内的燃油体积  $V$  为

$$V = (h_0 - s_k)S \quad (18)$$

其中， $h_2$  为柱塞运动到下止点时，柱塞底端与高压油泵顶端的距离，当柱塞腔运动到下止点时，低压燃油的压力为  $0.5MPa$ ，由燃油压力与密度的关系式，可以得到低压燃油的密度  $\rho_0$ 。

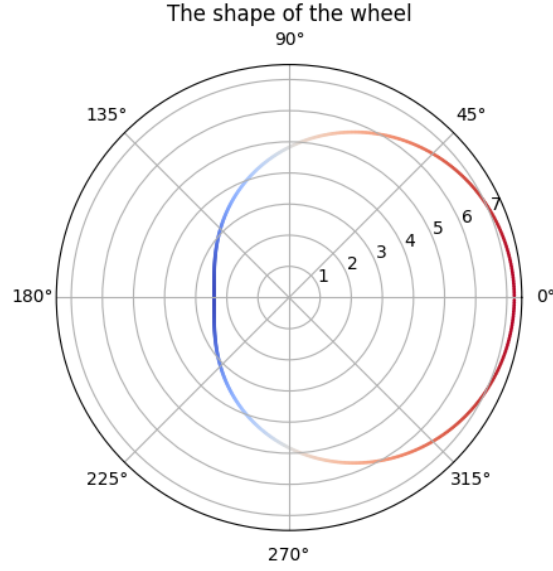


图 6:  $Z-x_{i,j}$  关系图

当凸轮旋转角为  $\phi_k$  时，腔内燃油体积不变，设燃油密度为  $\rho_1$ ，则  $\rho_1$  的表达式为

$$\rho_1 = \frac{\rho_0 h_2}{h_2 - s_2} \quad (19)$$

设凸轮的角速度为  $\omega$ ，则在一段极小时间 ( $dt$ ) 内凸轮转过的角度为

$$\phi_{t+dt} - \phi_t = \omega dt \quad (20)$$

当柱塞腔内压力随时间变化时，凸轮转角为  $\phi_k$  时柱塞腔内压力为  $P_1(t)$ ，高压油管内压力为  $P_2$ ，当  $P_1 > P_2$  时，单向阀开启，高压油泵在像高压油管内进油，进油流量满足的方程为

$$I(0) = CA \sqrt{\frac{2[P_1(t) - p_2(t)]}{\rho_1}} \quad (21)$$

### 5.3.2 燃油喷出

设针阀在某时刻的升程为  $h(t)$ ，由于题目给出了升程与时间的函数关系，又因为密封座与针阀空隙面积关联于高度，设空隙半径为  $R(t)$ 。设喷孔距离圆锥顶部的距离为  $L_0$ ，圆锥半角为  $9^\circ$ ，如图为喷油嘴横截面示意图

由几何关系，密封座的截面半径为

$$R_1(t) = [h(t) + L_0] \tan 9^\circ$$

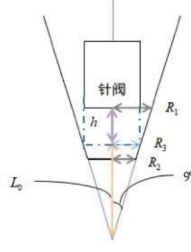


图 7: 针阀剖面图

针阀与密封座之间的空隙面面积为

$$S_1(t) = \pi[h_t + L_0]^2 \tan^2 9^\circ$$

喷孔半径为  $R_2$ , 喷孔面积为  $S_2 = \pi R_2^2$

则喷出燃油的横截面面积为  $S(t)$

$$S(t) = \min\{S_1(t), S_2\}$$

由于喷油嘴外部压力无法计算, 设外部压力为 1 个大气压  $P_e$ , 则高压油管出油量  $O(t)$  为

$$O(t) = CS(t) \sqrt{\frac{2[P_2(t) - P_e]}{\rho_2}}$$

### 5.3.3 模型的建立

以高压油管内压力作为研究对象, 研究其进油与出油情况。在基本初值条件下, 建立约束条件模型表达式。设改变的时间为  $\delta t$ , 则在  $t$  时刻下,  $\delta t$  时间段内, 高压油管内燃油的变化量为

$$m(t + \delta t) - m(t) = \rho_1 I(t) \delta t - \rho_2 O(t) \delta t$$

将该式变形, 取  $\delta t \rightarrow t$  得到高压油管内燃油变化量的微分方程为

$$\frac{d\rho_2(t)}{dt} = \frac{1}{V} [\rho_1 I(t) - \rho_2 O(t)] \quad (22)$$

联立得到高压油管内压力  $P(t)$  的模型表达式

$$\begin{cases} P(t) = f[\rho(t)] \\ I(t) = CA \sqrt{\frac{2[P_1(t) - P_2(t)]}{\rho_1}} \\ \frac{d\rho_2(t)}{dt} = \frac{1}{V} [\rho_1 I(t) - \rho_2 O(t)] \end{cases} \quad (23)$$

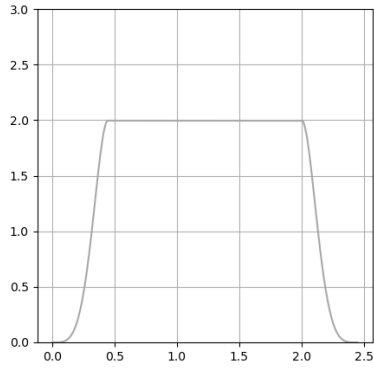
建立目标函数为

$$\min Z = \int (P_1(t) - P_2(t))^2 dt \quad (24)$$

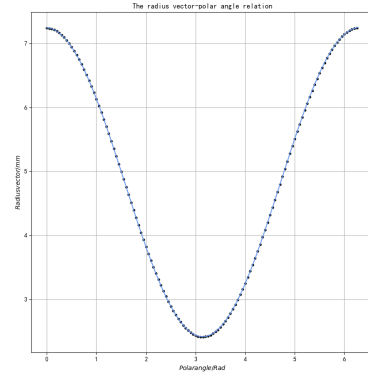
由该约束条件确定单目标规划模型，得出决策变量最优解。

## 5.4 模型的求解

对附件 1 的凸轮边缘曲线数据和附件 2 的针阀运动曲线数据进行处理，得到关系曲线分别如下图所示。其中，左下图凸轮运动曲线为当柱塞运动到下止点为初始状态时的处理结果。右下图为针阀运动曲线处理结果。



(a) 一周期内出油量-时间关系图



(b) 凸轮转角-凸轮升程关系图

图 8: 出油-进油子系统行为拟合

通过有限差分法，近似计算微分方程 (7) 的数值解，代入已知高压油管参数初始值，由高压油管压力求解模型求得管内压力。当管内压力大于 100MPa 时，油泵向管内进油；当针阀升程不为 0 时，喷油嘴处燃油喷出。以 0.01 为步长搜索凸轮运动角速度，迭代求解各值，使管内压力波动尽可能小。

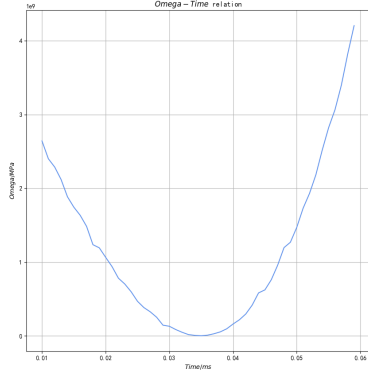
求得凸轮角速度为 0.0273rad/ms 时，高压油管压力稳定在 100MPa 左右，并在其附近小幅度上下波动。20s 的求解周期内，压力随时间变化关系如图所示：

为了验证模型的可行性，将凸轮角速度  $\omega = 0.0273\text{rad/ms}$  代入差分方程，计算 20s 的求解周期内，高压油管内压力在时间上的变化量，作出压力关于时间的波动关系如下图所示，可以看到在一段时间的波动状态后，压力逐渐趋于 100MPa，并在 100MPa 附近维持动态平衡。

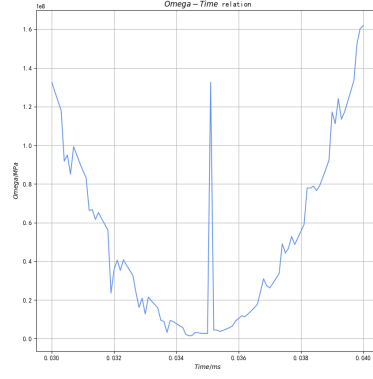
## 5.5 问题三的建模与求解

### 5.5.1 第一小问模型的建立

在问题二的基础上，再增加一个喷油嘴，每个喷油嘴喷油规律相同，但喷油开始时间不同。不失一般性，单个喷油周期内，喷油嘴 C 喷油开始时间落后喷油嘴 B 的喷油时间，设落后的



(a) 凸轮角速度-压力稳定程度关系 (初步搜索)



(b) 凸轮角速度-压力稳定程度关系 (精确搜索)

图 9: 凸轮角速度-压力稳定程度关系图

时间为  $t'_0$ ; 这种情况下, 设出油量  $E(T)$ :

$$E(t) = \begin{cases} Q_2(t) & t \in \Omega_{2,i} \\ Q_3(t) & t \in \Omega_{3,i} \\ 0 & t \notin \Omega_{2,i} \cup \Omega_{3,i} \end{cases} \quad i = 1, 2, \dots, n_2 \quad (25)$$

其中

$$\Omega_{2,i} = [t_0 + 100 \cdot (i - 1), 2.45 + t_0 + 100(i - 1)]; i = 1, 2, \dots, n_2$$

$$\Omega_{3,i} = [t_0 + t'_0 + 100 \cdot \dots (i - 1), 2.45 + t_0 + t'_0 + 100 \cdot \dots (i - 1)]; i = 1, 2, \dots, n_2$$

于是问题三第一小问的模型为: 决策变量

$$t'_0, \omega$$

目标函数

$$\min Z = \int_{t_1}^{t_2} (P(t) - P_e)^2 dt$$

约束条件

$$\text{方程组 (38) (39)}$$

### 5.5.2 第一小问模型的求解

基本部分的差分模型与第二问基本一致。

由于该模型为一个无约束条件的二元规划问题, 设一个开阀周期内, 凸轮的转速数组为  $\omega[1, 2, \dots, n_1]$ , 同时如上所述, 喷油嘴喷油的时间差数组为  $t'_0[1, 2, \dots, n_2]$ , 设二元向量组  $X$



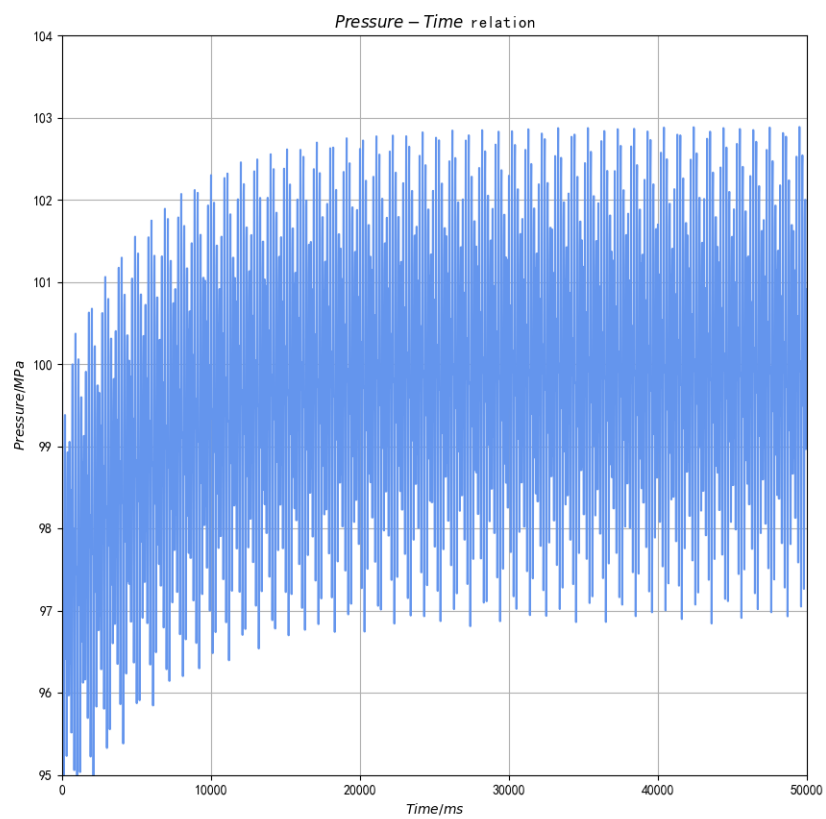


图 10: 最佳角速度下燃油压力-时间关系图

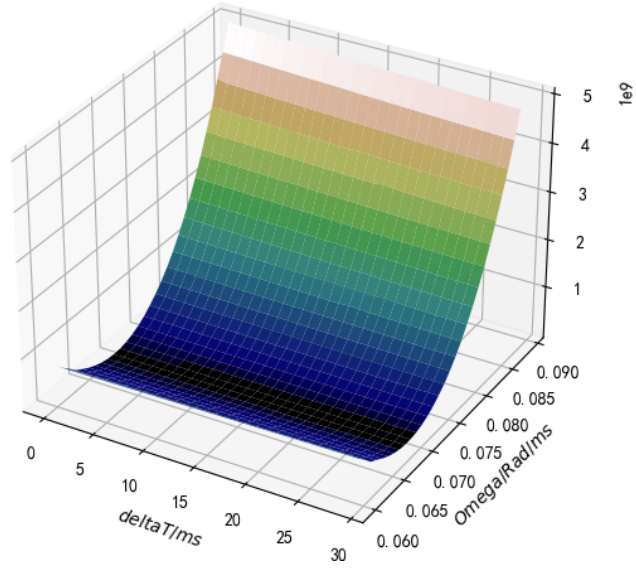


图 11:  $Z-x_{i,j}$  关系图

为模型的决策向量组

$$X = \{x_{1,1}, x_{1,2}, \dots, x_{1,n_2}, x_{2,1}, \dots, x_{n_1,1}, \dots, x_{n_1,n_2}\}$$

其中

$$x_{i,j} = (t'_0[i], \omega[j])$$

结合问题一所述的模型的性质与稳态与决策变量向量有唯一对应关系，可知该问题有唯一的全局最优解，则可以通过双目标多重搜索算法来找到使得压力波动最小的位点。

多重搜索算法的步骤和与第一问一致，在此不过多赘述，用数值模拟的方法近似得到条件矢量对应目标函数，通过差分方程的求解，我们得出目标函数  $Z$  与决策向量  $x_{i,j}$  的关系如下图所示：

通过多重搜索，我们得到  $\min Z$  的最小值为：，此时凸轮转速为：，

### 5.5.3 第二小问模型的建立

在第一小问的基础上安装了一个减压阀，通过查找资料等形式，我们发现泄压阀式减压阀为高压油管内最常见的泄压阀解决方案。因此，本模型选用了减压式泄压阀来作为连接高压油管和外部的阀门。

减压式减压阀工作方式与传统泄压阀一致，即其存在一个泄压阈值压力  $PR$ ；当  $t$  时刻高压油泵压力  $P(t) > B$  时，减压阀打开，否则关闭。

因此，考虑到减压阀的工作模式，我们通过引入 0-1 变量  $f(t)$ ，描述泄压状态：

$$f_2(t) = \begin{cases} 1 \\ 0 \end{cases}$$

于是通过减压阀泄压的出油量为

$$Q_4(t) = f_2 \cdots A \sqrt{\frac{2 \cdots (P(t) - P_{low})}{\rho(t)}}$$

于是重新定义减压泄压阀的泄压阀出油量为

$$E'(t) = Q_4(t) + E(t)$$

所以这种情况下，该模型为决策变量为

$$t'_0, \omega, P_R$$

目标函数为

$$\min Z = \int_{t_1}^{t_2} (P(t) - P_e)^2 dt$$

约束条件为

$$\text{方程组 (38) (39) (42)}$$

的单目标多决策变量微分方程组。

#### 5.5.4 第二小问模型的求解

根据第一小题建立的模型，我们先以 0.01 为步长，逐步移动决策向量，由于燃油压力不会产生骤变，我们通过搜索二维曲面上的凹点，初步确定决策向量的可能位置，然后再缩小步长，小步移动决策向量，采取高精度搜索的方式进一步确定决策向量。

最终，我们得到：使得油泵内压力稳定的决策向量为： $\Delta T = 50ms, \omega = 0.045rad/ms$

## 6 模型的评价与改进

### 6.1 模型的优点

1. 模型在一定程度上精确且巧妙地拟合了问题，并且方法便于复现，有一定的通用性。
2. 模型在该问题规模和精度上运行时间短，空间占用小。

### 6.2 模型的缺点

1. 模型采用数值拟合的方法对现实问题进行近似求解，在求解规模扩大的情况下或者精度要求扩大的情况下容易出现计算复杂度飙升的情况。
2. 在泄压阀式减压阀的引入时，程序无法对阈值的概念进行精确的描述，因此理论压力出现明显的波动。
3. 无法对模型进行检验。

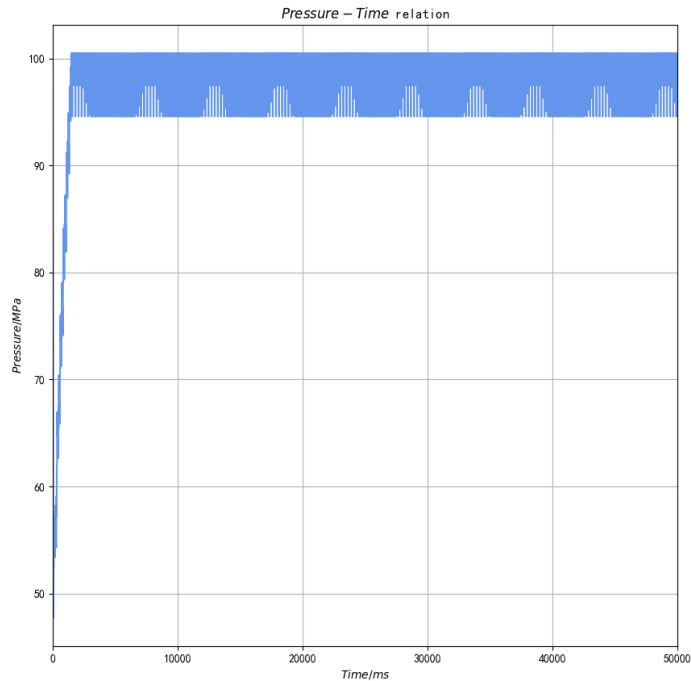


图 12: 加装减压阀后压力-时间关系图

4. 没有考虑管壁的黏着效应对燃油流速的影响。

### 6.3 模型的改进

(1) 由于高压油管的长度较直径更大，显得同时长度量级较大，所以可以考虑油管中压力传播的时间。将油管视为一维的长线，设  $t$  时刻  $x$  位置油管的压力为

$$P(x, t)$$

可以考虑其传播速度为

$$u = \sqrt{\frac{E(p)}{\rho(P)}}$$

将压力视为机械波，油管中压力的波动与传播情况。

(2) 第三个问题第二小问中的电磁阀式减压阀，可以进一步考虑其为可控制流量的以更好的控制压力的变化。

(3) 某些地方可以通过对常微分模型进行相轨分析，以此弥补模型精度的不足。

## 7 参考文献

## 8 附录

### 部分代码

---

```
#pragma GCC optimize(3,"Ofast","inline")
#include<bits/stdc++.h>
#define V 39269.92
#define C 0.85
#define A (3.1415926 * (1.4 * 1.4) / 4)
#define Height_left 1.0185916358172436119748521050582
#define Rho_0 0.81024129489933556476994554628566
#define Height_Max 7.239
#define Height_Min 2.413
#define radius 2.5
/*
ms
mm
MPa
mg/mm3
*/

long long Tot;

long double density1;

long double t0, m0, Omega, density_now, Tot_P_delta, P_now, r, deltaT;

long double min_Omega, min_deltaT;

long double xx0[10000], yy0[10000];

inline long double double_Mod(long double, long double);

inline long double Out_Oil(long double);

inline long double delta_increase_density2(long double);

inline long double Presure(long double);

inline long double In_Oil(long double);

inline void PreOri();

inline long double Height_Now(long double);
```

```

inline long double delta_h(long double);

inline long double Pow(long double);

inline long double Out_Oilx(long double);

int Now, whe;

int Tot_where[100];

int main(){
    PreOri();
    printf("DATA READY\n");
    m0 = 0.0;
    freopen("1_3_pres.txt", "w", stdout);
    density1 = 0.0;
    long double Min_Tot = (long double)INT_MAX * 100;
    Omega = 0.450;
    //Omega = 0.069;
    Tot = 0;
    whe = 1;
    for(int i = 1; i <= 10; ++ i){
        Tot_where[i] = 283 * i;
    }
    // while (Omega <= 0.09){
        deltaT = 50;
        // while (deltaT <= 30){
            density_now = 0.8512;
            Tot_P_delta = 0;
            for(register long double k = 0.0; k <= 50000; k += 0.01){
                P_now = Presure(density_now);
                if (k >= 5000)
                    Tot_P_delta += ((P_now - 100)) * ((P_now - 100));
                density_now += delta_increase_density2(k);
                // printf("%Lf\n", delta_increase_density2(k));
                if (Tot % 10 == 0){
                    printf("%Lf\n", P_now);
                    // getchar();
                }
                Tot ++;
            }
            // if (Tot_P_delta < Min_Tot){
                // Min_Tot = Tot_P_delta;
            }
        }
    }
}

```

```

        //      min_Omega = Omega;
        //      min_deltaT = deltaT;
        //      }
    //      printf("%.3Lf", Tot_P_delta);
    //Tot ++;
    //if (Tot == Tot_where[whe]){
        //freopen("CON", "w", stdout);
        // whe ++;
        // printf("percent %d finish...\n", (whe * 5));
        //freopen("1_3.csv", "w", stdout);
        //}
    //      deltaT += 1;
    //      }
    //      printf("\n");
    //      Omega += 0.001;
    // }
    // freopen("CON", "w", stdout);
    // printf("%Lf\n%Lf", min_Omega, min_deltaT);
}

```

```

inline long double Pow(long double x, int y){
    long double ans = 1.0;
    for(int i = 1; i <= y; ++ i)
        ans *= x;
    return ans;
}

```

```

inline long double double_Mod(long double x, long double y){// calculate x mod y
    int k = x / y;
    x -= (long double)k * y;
    return x;
}

```

```

inline long double delta_increase_density2(long double t){
    long double ans = 1.0 / V;
    ans *= (density1 * In_Oil(t) - density_now * (Out_Oil(t) + Out_Oilx(density_now)));
    ans *= 0.01;
    //if (ans > 0)
    // printf("%Lf\n", ans);
    return ans;
}

```

```

inline long double Out_Oilx(long double dens){
    long double ans = 0;

```

```

    if (Pressure(dens) > 100.5){
        long double delta_p = Pressure(dens) - 0.5;
        if (delta_p > 0)
            ans = C * M_PI * 0.7 * 0.7 * sqrt(delta_p * 2 / dens);
    }
    return ans;
}

inline long double Pressure(long double dens){
    long double P = 1000 * (dens - 0.81) / (1.86 - 1.70 * dens);
    //printf("%Lf\n", dens);
    //getchar();
    //long double P = 3000 * dens - 2450;
    return P;
}

inline void PreOri(){
    Now = 0;
    for(register long double Theta = 0; Theta <= 6.28; Theta += 0.001){
        r = 0.002396583358025873 * Pow(Theta, 6) - 0.04516945210675776 * Pow(Theta, 5) +
            0.25934369942307445 * Pow(Theta, 4) - 0.28719125885798014 * Pow(Theta, 3) -
            0.949901668038579 * Pow(Theta, 2) - 0.09408556096323666 * Pow(Theta, 1) +
            7.247157128703328;
        Now ++;
        xx0[Now] = - r * sin(Theta);
        yy0[Now] = r * cos(Theta);
        //printf("%Lf\n", r);
        //getchar();
    }
}

inline long double Height_Now(long double t){
    register long double Phi = double_Mod((t * Omega), 6.28);
    register long double Y_MAX = 0;
    register long double x_new, y_new;
    for(register int i = 1; i <= Now; ++ i){
        x_new = xx0[i] * cos(Phi) + yy0[i] * sin(Phi);
        y_new = -xx0[i] * sin(Phi) + yy0[i] * cos(Phi);
        if (y_new > Y_MAX) Y_MAX = y_new;
    }
    //printf("%Lf\n", Y_MAX);
    //getchar();
    return Y_MAX;
}

inline long double In_Oil(long double t){
    long double V_ori = ((Height_Max - Height_Min + Height_left) * M_PI * radius *

```



```

        radius);
    long double ans = 0.0;
    if (double_Mod((t * Omega), 6.28) >= 3.1395 && double_Mod((t * Omega), 6.28) <=
        3.1405)
    m0 = V_ori * Rho_0;
    //printf("%Lf\n", m0);
    density1 = m0 / (((Height_Max - Height_Now(t)) + Height_left) * M_PI * radius *
        radius);
    // if (density1 > 0){
    //     printf("%Lf\n", density1);
    //     getchar();
    // }
    long double delta_P = 2 * (Presure(density1) - Presure(density_now));
    if (delta_P > 0)
    ans = C * A * (long double)sqrt(delta_P / density1);
    else
    ans = 0.0;
    m0 -= (ans * density1 * 0.01);
    // if (ans > 0)
    //     printf("%Lf %Lf\n", t, ans);
    return ans;
}
//
inline long double delta_h(long double lalala){
    if(lalala == 0) return 0;
    else if(lalala == 0.01) return 1.2337E-06;
    else if(lalala == 0.02) return 0.000019739;
    else if(lalala == 0.03) return 0.000099928;
    else if(lalala == 0.04) return 0.00031581;
    else if(lalala == 0.05) return 0.00077096;
    else if(lalala == 0.06) return 0.0015984;
    else if(lalala == 0.07) return 0.0029607;
    else if(lalala == 0.08) return 0.005049;
    else if(lalala == 0.09) return 0.0080834;
    else if(lalala == 0.10) return 0.012312;
    else if(lalala == 0.11) return 0.018008;
    else if(lalala == 0.12) return 0.025473;
    else if(lalala == 0.13) return 0.035029;
    else if(lalala == 0.14) return 0.047021;
    else if(lalala == 0.15) return 0.061809;
    else if(lalala == 0.16) return 0.079768;
    else if(lalala == 0.17) return 0.10128;
    else if(lalala == 0.18) return 0.12674;
    else if(lalala == 0.19) return 0.15651;
    else if(lalala == 0.20) return 0.19098;
    else if(lalala == 0.21) return 0.23049;

```

```

else if(lalala == 0.22) return 0.27535;
else if(lalala == 0.23) return 0.32583;
else if(lalala == 0.24) return 0.38214;
else if(lalala == 0.25) return 0.44443;
else if(lalala == 0.26) return 0.51275;
else if(lalala == 0.27) return 0.58705;
else if(lalala == 0.28) return 0.66718;
else if(lalala == 0.29) return 0.75283;
else if(lalala == 0.30) return 0.84357;
else if(lalala == 0.31) return 0.93878;
else if(lalala == 0.32) return 1.0377;
else if(lalala == 0.33) return 1.1393;
else if(lalala == 0.34) return 1.2426;
else if(lalala == 0.35) return 1.3461;
else if(lalala == 0.36) return 1.4484;
else if(lalala == 0.37) return 1.5477;
else if(lalala == 0.38) return 1.6423;
else if(lalala == 0.39) return 1.73;
else if(lalala == 0.40) return 1.809;
else if(lalala == 0.41) return 1.8771;
else if(lalala == 0.42) return 1.9321;
else if(lalala == 0.43) return 1.972;
else if(lalala == 0.44) return 1.995;
else if (0.45 <= lalala && lalala <= 2.00) return 2.0;
else if(lalala == 2.01) return 1.9942;
else if(lalala == 2.02) return 1.9704;
else if(lalala == 2.03) return 1.9296;
else if(lalala == 2.04) return 1.8739;
else if(lalala == 2.05) return 1.8052;
else if(lalala == 2.06) return 1.7258;
else if(lalala == 2.07) return 1.6376;
else if(lalala == 2.08) return 1.5427;
else if(lalala == 2.09) return 1.4432;
else if(lalala == 2.1) return 1.3409;
else if(lalala == 2.11) return 1.2373;
else if(lalala == 2.12) return 1.1341;
else if(lalala == 2.13) return 1.0326;
else if(lalala == 2.14) return 0.93382;
else if(lalala == 2.15) return 0.83882;
else if(lalala == 2.16) return 0.74833;
else if(lalala == 2.17) return 0.66296;
else if(lalala == 2.18) return 0.58312;
else if(lalala == 2.19) return 0.50912;
else if(lalala == 2.2) return 0.44111;
else if(lalala == 2.21) return 0.37912;
else if(lalala == 2.22) return 0.32311;

```

```

else if(lalala == 2.23) return 0.27292;
else if(lalala == 2.24) return 0.22835;
else if(lalala == 2.25) return 0.18911;
else if(lalala == 2.26) return 0.15488;
else if(lalala == 2.27) return 0.12534;
else if(lalala == 2.28) return 0.10009;
else if(lalala == 2.29) return 0.078769;
else if(lalala == 2.3) return 0.060981;
else if(lalala == 2.31) return 0.046344;
else if(lalala == 2.32) return 0.034486;
else if(lalala == 2.33) return 0.025044;
else if(lalala == 2.34) return 0.017677;
else if(lalala == 2.35) return 0.012063;
else if(lalala == 2.36) return 0.0079019;
else if(lalala == 2.37) return 0.0049215;
else if(lalala == 2.38) return 0.0028753;
else if(lalala == 2.39) return 0.0015448;
else if(lalala == 2.4) return 0.00073997;
else if(lalala == 2.41) return 0.0003;
else if(lalala == 2.42) return 0.000093301;
else if(lalala == 2.43) return 0.000017801;
else if(lalala == 2.44) return 1.0005E-06;
else if(lalala == 2.45) return 0;

}

inline long double Out_Oil(long double t){
    register long double t_Mod = double_Mod(t, 100.0);
    long double ans = 0;
    if (t_Mod > 2.45)
        ans = 0.0;
    else{
        long double S1 = 0.49 * M_PI;
        long double d = delta_h(t_Mod) * sin(M_PI / 180.0 * 9.0);
        long double R = 1.25 + d * cos(M_PI / 180.0 * 9.0);
        long double S2 = M_PI * d * (r + R);
        long double S = 0.0;
        (S1 > S2) ? (S = S2) : (S = S1);
        long double delta_p = Presure(density_now) - 0.1;
        ans = C * S * sqrt(delta_p * 2 / density_now);
        //printf("%Lf\n", density1);
        //getchar();
    }
    t_Mod = double_Mod(t - deltaT, 100.0);
    long double ansx = 0;
    if (t_Mod > 2.45)

```

```

ansx = 0.0;
else{
    long double S1 = 0.49 * M_PI;
    long double d = delta_h(t_Mod) * sin(M_PI / 180.0 * 9.0);
    long double R = 1.25 + d * cos(M_PI / 180.0 * 9.0);
    long double S2 = M_PI * d * (r + R);
    long double S = 0.0;
    (S1 > S2) ? (S = S2) : (S = S1);
    long double delta_p = Presure(density_now) - 0.1;
    ansx = C * S * sqrt(delta_p * 2 / density_now);
    //printf("%Lf\n", density1);
    //getchar();
}
return ans + ansx;
}

```

---

```

import numpy as np
import matplotlib.pyplot as plt
from math import *

plt.rcParams['font.family']='[Adobe Fangsong Std]'
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

def loadtxtmethod(filename):
    data = np.loadtxt(filename, dtype=np.float32, delimiter=',')
    return data

if __name__ == "__main__":
    data = loadtxtmethod('1_3_pres.txt')
    x = np.arange(0.0, 50000, 0.1)
    plt.figure(figsize=(10, 10))
    plt.plot(x, data, color = 'cornflowerblue', linestyle='solid')
    plt.title("$\Omega$-Time$ relation")
    plt.xlabel('$Time/ms$')
    plt.ylabel("$\Omega$/MPa$")
    # plt.ylim((95, 104))
    plt.xlim((0, 50000))
    plt.grid()
    plt.savefig("1_3pres.png")

```

---

```

import numpy as np

```

```

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from scipy.optimize import curve_fit

plt.rcParams['font.family']='[Adobe Fangsong Std]'
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.rcParams['axes.unicode_minus'] = False

fig = plt.figure()
ax = Axes3D(fig,auto_add_to_figure=False)
fig.add_axes(ax)

# x = [1200,1600,2000,2400,2800,3200,3600,4000]
# y = [1200,1600,2000,2400,2800,3200,3600]
x = [aa for aa in range(0, 30)]
y = [bb / 1000 for bb in range(60, 90, 1)]
xx,yy= np.meshgrid(x,y)
data = np.loadtxt('1_3_plus.csv',delimiter=',',usecols=(range(0, 30)))
# Z=data.split()
# z=[Z[i:i+3] for i in range(0,len(Z),3)]
zz = np.array(data)
# a = []
# for i in zz:
#     a += list(i)
# a = np.array([np.array(a)])
print(zz)
plt.xlabel('$\Delta T/ms$')
plt.ylabel('$\Omega/\text{Rad}/ms$')
plt.clabel('$evaluate$')

ax.plot_surface(xx, yy, zz, rstride=1, cstride=1,cmap='gist_earth')
# plt.show()
plt.grid()
plt.savefig("1_3_plus.png")

```

---