

# 动态模量预测

## 一、实验背景

在路面自然环境下，沥青热氧老化引起的流变性能衰变，是影响沥青混合料力学性能的重要因素之一，直接影响沥青在路面的使用效率，因此提出评价与预测沥青老化流变性能的方法具有重要的意义。目前已有的评价沥青老化性能指标的方法多集中在既定温度或频率（时间）下的测定结果，如针入度、软化点和延度等，其无法真实反映沥青在非等温环境下流变力学性能演变过程，也无法精确反映沥青老化程度和老化速率。沥青是一种粘弹性材料，其性能在广泛的温度和频率范围内随温度或频率会连续变化，因此基于流变学的沥青复数剪切模量与相位角主曲线作为评价沥青老化性能的指标具有坚实的理论基础和明显的优势。在热氧老化过程中，影响沥青流变性能衰变的两个主要因素为老化温度和老化时间，然而沥青复数剪切模量与相位角主曲线仅体现了沥青在测试温度和加载频率范围内的变化规律，无法有效的表征老化温度和老化时间对复数剪切模量与相位角的影响，更无法预测沥青老化流变性能。

鉴于此提出一种沥青老化流变主曲线的预测方法，可在较宽的老化温度和老化时间范围内预测沥青复数剪切模量和相位角，由此可进一步预测沥青混合料动态模量  $E$ ，最终实现沥青路面的抗车辙性能、疲劳性能和抗低温开裂性能的精确预测。



图 1：动态模量测试仪。

通过旋转压实实验得到如下数据：

表 1：实验数据

再生骨料掺量（实验环境）	温度/°C	模量/MPa					
		加载频率/Hz					
		25	10	5	1	0.5	0.1
0%	-10	23984	22332	20924	17422	15920	12563
	5	17966	15948	14357	10778	9408	6567
	20	8838	7173	5973	3696	3016	1897
	35	3817	2815	2212	1259	1056	728.4
	50	1345	993.4	821.3	546.6	505.4	420
17%	-10	22947	21419	20151	16941	15507	12355
	5	16132	14064	12498	9097	7779	5223
	20	9061	7297	6073	3730	2985	1745
	35	3760	2782	2185	1278	1071	754.3
	50	1423	908	716.3	430.6	386.4	301.2
37%	-10	18857	17725	16721	14216	13092	10565
	5	14023	12420	11159	8394	7287	5004
	20	8396	6892	5816	3697	2997	1799
	35	3180	2355	1854	1074	902.7	636.1
	50	1166	809.4	629.4	373.2	327.3	249.6
56.7%	-10	16916	15839	14951	12731	11737	9531
	5	12336	10923	9833	7425	6472	4545
	20	7330	6040	5125	3323	2735	1701
	35	2753	2082	1680	1046	905.5	678.9
	50	1161	835.5	666.8	403.2	357.2	272.4

## 二、数学模型

以第一组数据为例进行计算，绘制第一组数据的离散图以便观察趋势。如下图所示

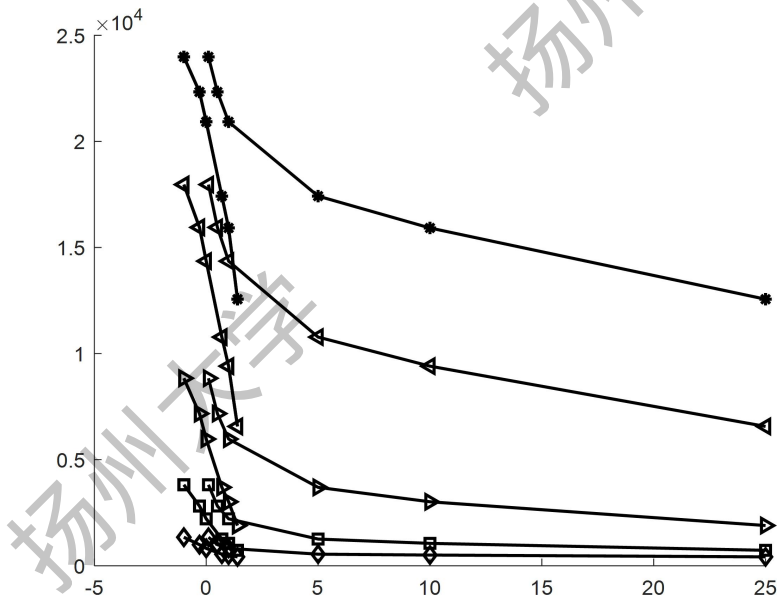


图 2：离散点图。

为了平衡数据的稀疏，对横纵坐标进行对数化处理，得到的数据进行拟合：

$$\begin{cases} f' = \lg f \\ E' = \lg E \end{cases}$$

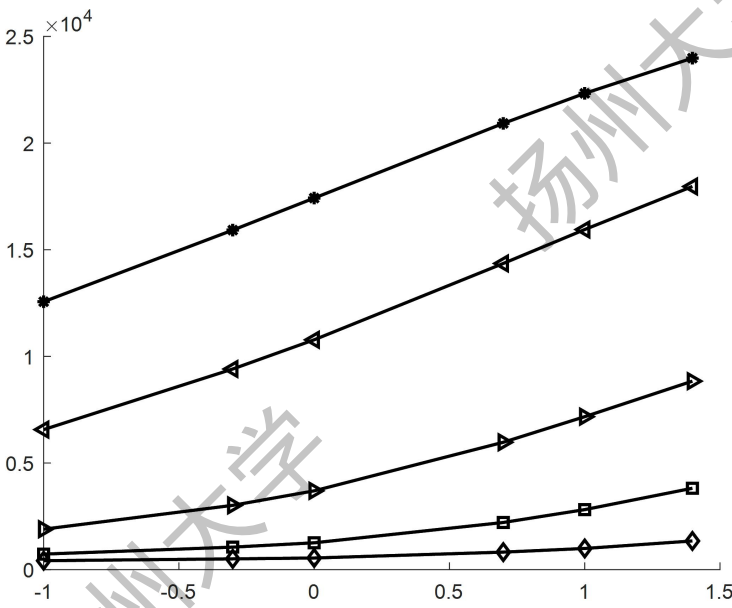


图 3：对数处理后的图像。

动态模量主曲线使用的 sigmoid 函数形式如下所示。

$$\lg(|E^*|)=\delta+\frac{\alpha-\delta}{1+e^{\beta+\gamma\cdot\lg f_r}}$$

式中： $|E^*|$ 为沥青混合料的动态模量（MPa）； $\delta$ 为该组沥青混合料动态模量的最小值的对数； $\alpha$ 为该组沥青混合料动态模量的最大值的对数； $\beta$ 、 $\gamma$ 为描述sigmoid函数的形状参数； $f_r$ 为参考温度下的缩减频率，单位为 Hz。公式如下：

$$g(x)=\lg(|E^*|)=\delta+\frac{\alpha-\delta}{(1+e^{\beta+\gamma\cdot\lg f_r})}$$

利用最小二乘法求解系数，构建函数

$$\varphi(\beta,\gamma)=\sum_{i=1}^n(f_i-f)^2$$

其中  $n$  为数据个数， $f_i$  为函数拟合值， $f$  为实际数据点的数据。

要使其  $\varphi$  达到最小值，利用矛盾方程组，采用多元函数求偏导的方法求解，如下所示：

$$\begin{cases} \frac{\partial \varphi}{\partial \beta}=0 \\ \frac{\partial \varphi}{\partial \gamma}=0 \end{cases}$$

根据该原理，利用 Matlab 的 cftool 工具箱求解可得所求系数解，系数汇总如下所示：

表 2：再生骨料掺量条件下系数

	$\delta$	$\alpha$	$\beta$	$\gamma$	$R^2$
0	2.6232	4.3799	-4.463	-0.5565	99.40
17	2.4789	4.3607	-3.805	-0.5949	99.57
37	2.3972	4.2755	-3.778	-0.5921	99.64
56.7	2.4352	4.2283	-3.163	-0.5770	99.49

绘制拟合函数，结果如图所示

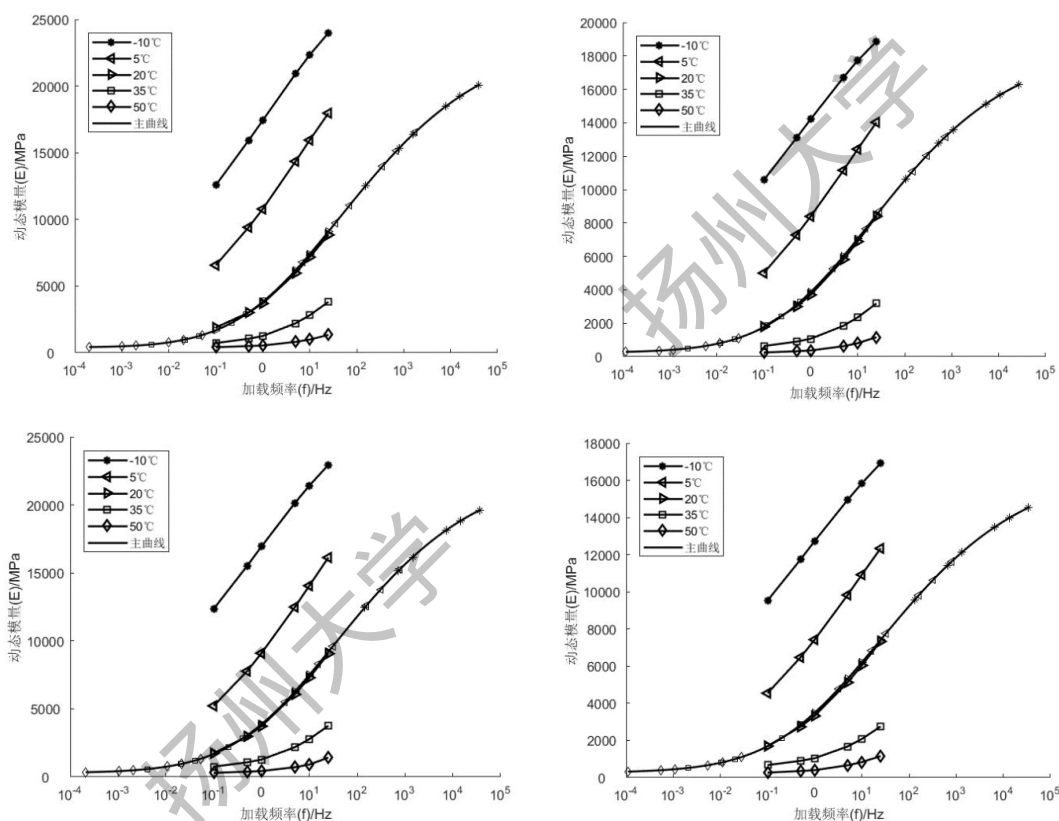


图 4：四种掺量的沥青混合料主曲线。

通过拟合函数的方法，采用离散数据构建一条实数范围内的连续函数，根据该函数可在动态模量  $E$  取任意数值时，反推对应频率  $f$  的大小，根据频率大小可进而推测沥青混合料的材料性质。该方法即为数值分析中的牛顿迭代法。

$$x_{k+1} = x_k - \frac{g(x)}{g'(x)}$$

得知函数之后，对函数  $g(x)$  进行求导可得  $g'(x) < 1$ ，在离散的实验数据中，动态模量分别为：23984，22332，20924，17422，15920，12563，对应的频率分别为：25，10，5，1，0.5，0.1，那么当动态模量为 10000 时，对应的频率  $f$  为多少呢？显然在得到拟合函数后，求解对应的  $f$ ，但是方程较为复杂不易求解，因此可以使用牛顿迭代法任取需要的  $E$ ，求解相应的解频率  $f$ ，根据频率  $f$  大小可判断材料的高温或低温性能。

### 三、算例

例如，在第一组试验条件下，当  $E=10000$  时，选定初值  $f$  为 10，利用牛顿迭代法可以解得  $x^* \approx 100$ 。

### 四、代码

```
%参考温度 20°C
%move 是你要的位移因子
%横坐标 lgf_p 是频率取对数
%TEMP 是模量取对数
T=[-10 5 20 35 50]';
f=[25 20 10 1 0.5 0.1];
%example (可以直接从 excel 复制粘贴)
%温度/°C 0.1Hz 0.5Hz 1Hz 5Hz 10Hz 25Hz
%-10°C 22456 20114 18892 17710 15199 13433
%5°C 18436 15989 14011 11571 9257 6299
%20°C 8308 6138 5552 3569 2102 1056
%35°C 2391 1689 1237 613 507 346
%50°C 1387 877 712 473 402 354
E1=[23984 22332 20924 17422 15920 12563
17966 15948 14357 10778 9408 6567
8838 7173 5973 3696 3016 1897
3817 2815 2212 1259 1056 728.4
1345 993.4 821.3 546.6 505.4 420];
lgf=log(f)./log(10);
figure(1)
hold on
plot(lgf,E1(1,:),'k*-', 'LineWidth',1.5)
plot(lgf,E1(2,:),'k<-', 'LineWidth',1.5)
plot(lgf,E1(3,:),'k>-', 'LineWidth',1.5)
plot(lgf,E1(4,:),'ksquare-', 'LineWidth',1.5)
plot(lgf,E1(5,:),'kdiamond-', 'LineWidth',1.5)

h1 = plot(lgf,E1(1,:),'k*-', 'LineWidth',1.5);
h2 = plot(lgf,E1(2,:),'k<-', 'LineWidth',1.5);
h3 = plot(lgf,E1(3,:),'k>-', 'LineWidth',1.5);
h4 = plot(lgf,E1(4,:),'ksquare-', 'LineWidth',1.5);
h5 = plot(lgf,E1(5,:),'kdiamond-', 'LineWidth',1.5);
```

```

%plot(lgf,E1(1,:),'Color','#575C73','*-', 'LineWidth',1.5)
%plot(lgf,E1(2,:),'<-', 'Color','#CED6E7','LineWidth',1.5)
%plot(lgf,E1(3,:),'>-', 'Color','#778BBE','LineWidth',1.5)
%plot(lgf,E1(4,:),'square-', 'Color','#91B2D7','LineWidth',1.5)
%plot(lgf,E1(5,:),'diamond-', 'Color','#807C93','LineWidth',1.5)

%找距离最近的点
E1=[sort(E1(1,:))
sort(E1(2,:))
sort(E1(3,:))
sort(E1(4,:))
sort(E1(5,:))];
sort_lgf=sort(lgf);

%第二条曲线位移多少
c1=abs(E1(3,:)-min(E1(2,:)));
c1_min=min(c1);
sit_c1=find(c1==c1_min);
if sit_c1 < 6
    newp=sort_lgf(sit_c1)+(min(E1(2,:))-E1(3,sit_c1))*(sort_lgf(sit_c1+1)-sort_lgf(
sit_c1))/(E1(3,sit_c1+1)-E1(3,sit_c1));
else
    newp=sort_lgf(5)+(min(E1(2,:))-E1(3,4))*(sort_lgf(4+1)-sort_lgf(4))/(E1(3,4+1)
-E1(3,4));
end
move2 = newp - sort_lgf(1)

%第一条曲线位移多少
c1=abs(E1(2,:)-min(E1(1,:)));
c1_min=min(c1);
sit_c1=find(c1==c1_min);
if sit_c1 < 6
    newp=sort_lgf(sit_c1)+(min(E1(1,:))-E1(2,sit_c1))*(sort_lgf(sit_c1+1)-sort_lgf(
sit_c1))/(E1(2,sit_c1+1)-E1(2,sit_c1));
else
    newp=sort_lgf(5)+(min(E1(1,:))-E1(2,4))*(sort_lgf(4+1)-sort_lgf(4))/(E1(2,4+1)
-E1(2,4));
end
move1 = newp - sort_lgf(1) + move2

%第四条曲线位移多少
c1=abs(E1(3,:)-max(E1(4,:)));
c1_min=min(c1);
sit_c1=find(c1==c1_min);

```

```

if sit_c1 > 1
    newp=sort_lgf(sit_c1)+(max(E1(4,:))-E1(3,sit_c1))*(sort_lgf(sit_c1+1)-sort_lgf(
sit_c1))/(E1(3,sit_c1+1)-E1(3,sit_c1));
else
    newp=sort_lgf(1)+(max(E1(4,:))-E1(3,1))*(sort_lgf(2)-sort_lgf(1))/(E1(3,2)-E1(
3,1));
end
move4 = newp - sort_lgf(6)
%第五条曲线位移多少
c1=abs(E1(4,:)-max(E1(5,:)));
c1_min=min(c1);
sit_c1=find(c1==c1_min);
if sit_c1 > 1
    newp=sort_lgf(sit_c1)+(max(E1(5,:))-E1(4,sit_c1))*(sort_lgf(sit_c1+1)-sort_lgf(
sit_c1))/(E1(4,sit_c1+1)-E1(4,sit_c1));
else
    newp=sort_lgf(1)+(max(E1(5,:))-E1(4,1))*(sort_lgf(2)-sort_lgf(1))/(E1(4,2)-E1(
4,1));
end
move5 = newp - sort_lgf(6) + move4

%绘制主曲线
move=[move1 move2 0 move4 move5];
lgf_p=[sort_lgf+move5      sort_lgf+move4      sort_lgf      sort_lgf+move2
sort_lgf+move1];
lgE1=log(E1)./log(10);
TEMP1=lgE1(1,:);
TEMP2=lgE1(2,:);
TEMP3=lgE1(3,:);
TEMP4=lgE1(4,:);
TEMP5=lgE1(5,:);
TEMP=[TEMP5 TEMP4 TEMP3 TEMP2 TEMP1];
delta=min(TEMP)
alpha=max(TEMP)
disp(strcat('最小对数值为',num2str(delta)))
disp(strcat('最大对数值为',num2str(alpha)))

%请使用 cftool 拟合相关参数!!!点击软件上方选项 curve fitting
%函数一开始默认 ploynomial, 选择自定义函数选项 custom equation
%自定义函数, 复制粘贴到输入框中, delta+(alpha-delta)/(1+exp(b+r*x))
%x 选择 logf_p,y 选择 TEMP

b=input('请输入 b 的拟合数值');
r=input('请输入 r 的拟合数值');

```



```

l=input('请输入 l 的拟合数值');
x_lgf=[lgf_p(1):0.01:lgf_p(end)];
y_lgE=delta+(alpha-delta)./(1+l.*exp(b+r.*x_lgf)).^l;
y_E=10.^y_lgE;
plot(x_lgf,y_E,'k-','LineWidth',1.5)
h6 = plot(x_lgf,y_E,'k-','LineWidth',1.5);
E11=10.^(delta+(alpha-delta)./(1+l.*exp(b+r.*(sort_lgf+move1)))).^l;
E12=10.^(delta+(alpha-delta)./(1+l.*exp(b+r.*(sort_lgf+move2)))).^l;
E13=10.^(delta+(alpha-delta)./(1+l.*exp(b+r.*sort_lgf)).^l);
E14=10.^(delta+(alpha-delta)./(1+l.*exp(b+r.*(sort_lgf+move4)))).^l;
E15=10.^(delta+(alpha-delta)./(1+l.*exp(b+r.*(sort_lgf+move5)))).^l;

plot(sort_lgf+move5,E15,'kdiamond','MarkerSize',5)
plot(sort_lgf+move4,E14,'ksquare','MarkerSize',5)
plot(sort_lgf,E13,'k>','MarkerSize',5)
plot(sort_lgf+move2,E12,'k<','MarkerSize',5)
plot(sort_lgf+move1,E11,'k*','MarkerSize',5)
xlabel('加载频率(f)/Hz')
ylabel('动态模量(E)/MPa')
set(gca, 'xticklabel', get(gca, 'xtick'), 'yticklabel', get(gca, 'ytick'));
set(gca,'XTickLabel',{'10^{-4}','10^{-3}','10^{-2}','10^{-1}','0','10^{1}','10^{2}',
'10^{3}','10^{4}','10^{5}'})

```