数值计算方法在翼型实验中的应用

——翼型坐标插值、拟合出轮廓曲线

摘　要：本案例给出了数值计算方法在翼型试验中的应用，依托实际背景以及物理意义，依照翼型坐标数据，通过插值方法，拟合翼型轮廓曲线。在求解过程中，为对比各种差值方法的有效性和准确性，采用了 Lagrange 插值、Newton 插值、分段线性插值、分段三次 Hermite 插值以及三次样条插值进行数值实验。

该案例有助于加深学生对《数值分析》课程中插值算法的理解，并提升其数值计算的能力，也适用于学生针对获得的数据，进行数据分析和拟合。

关键词：翼型实验，数值计算方法，插值方法。

1 背景介绍

1.1 飞机机翼翼型的重要性

飞机要实现飞行，首先依靠机翼的升力。那么升力是怎样产生的呢？这就是人们熟知的伯努利原理：水与空气等流体，流速大的地方，压强小；流体流速小的地方，压强大。把机翼纵向剖开，会形成一个翼截面或翼剖面，在航空上称**翼型**。当空气流过机翼时，气流会沿上下表面分开，并在后缘处汇合。上表面弯曲，气流流过时走的路程较长，下表面较平坦，气流的行程较短。上下气流最后要在一处汇合，因而上表面的气流必须速度较快，才能与下表面气流同时到达后缘。根据伯努利原理，上表面高速气流对机翼的压力较小，下表面低速气流对机翼压力较大，从而产生了一个压力差，也就是向上的升力。

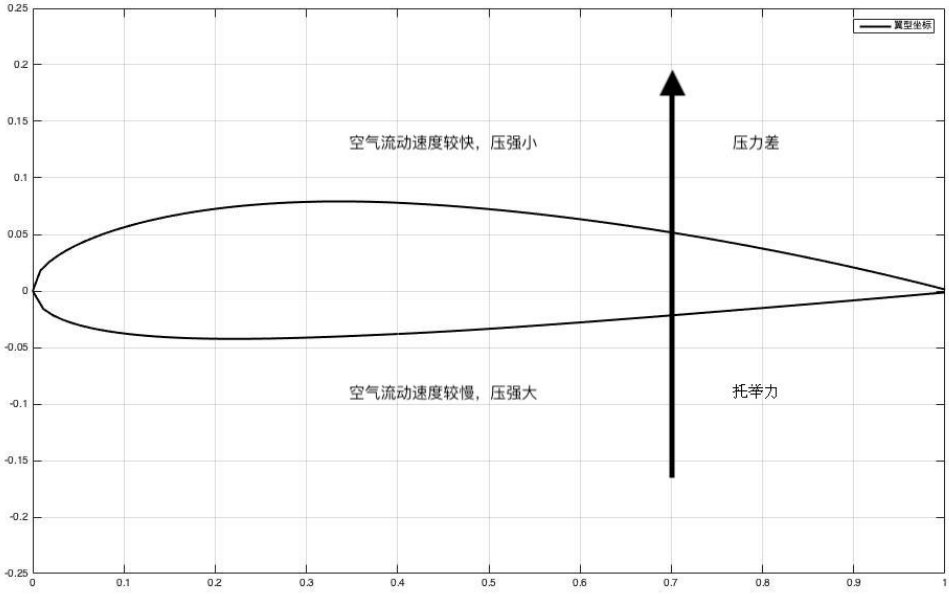

图 1 翼型原理

机翼的翼型选择是机翼设计的关键，优秀的翼型能为飞机在不同飞行状态下提供良好的升力，由此，在机翼设计的不同历史时期，出现了不同类型的翼型。

1.2 翼型数值实验的意义

数值计算方法经常用于飞机外形的设计，尤其是翼型。飞机的制造耗资巨大，在设计过程中必须确定该型飞机的最主要飞行状态，设计师根据其飞行状

态再为其选择最合适的翼型，并不是单独追求某个性能的最优，而是反复权衡比较之后的整体性能最优，是一项关联复杂、逻辑严密的系统工程，因此，设计之初的模型实验必不可少,如何保证模型的数据可以应用到真实飞机上呢？本案例通过采集的翼型数据，采用插值方法拟合出翼型轮廓曲线。

换言之，翼型实验的意义在于，通过实验手段测量可用的翼型数据及其系数，飞机总设计师根据实验结果，结合飞机整体需求选择最合适的翼型。由此可见，通过插值方法，拟合出翼型轮廓曲线在工程中具有很重要的意义。

2 案例内容

在此案例中，我们通过现有的翼型模型，获得翼型坐标数据；根据翼型坐标数据，采用 Lagrange 插值、Newton 插值、分段三次 Hermite 插值、分段线性插值以及三次样条插值进行数值实验，获得较为准确的翼型轮廓曲线。

2.1 翼型坐标数据

本案例研究 NACA2412 翼型的 1 米模型，翼型的数据见附录表格，翼型数据坐标散点图如图 2 所示，已知翼型轮廓线上的部分数据，为了得到经过这些已知数据点的轮廓曲线，接下来通过插值方法，拟合出翼型轮廓曲线。

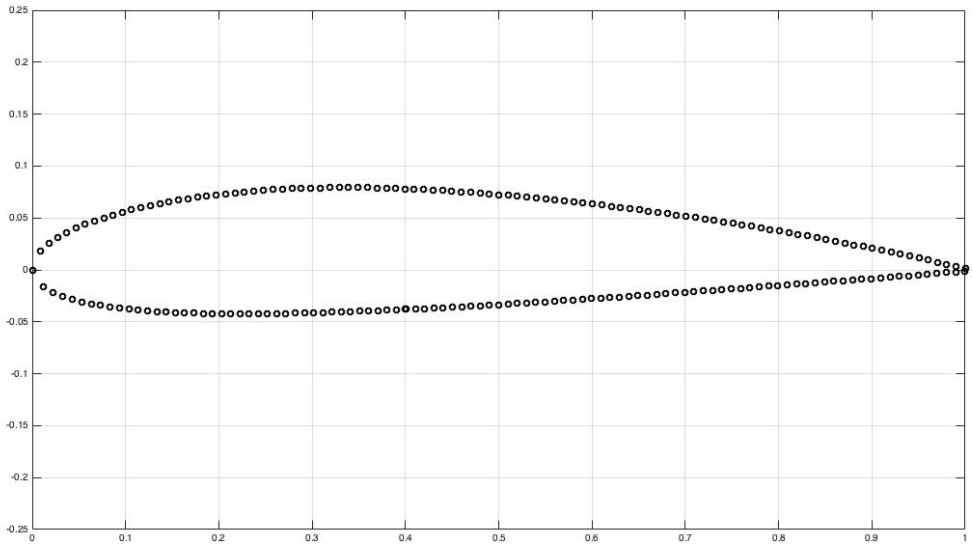

图 2 NACA2412 翼型数据

2.2 插值方法概述

已知 n+1 个数据节点 $x_0 < x_1 < \cdots < x_{\mathrm{n}}$，函数值$f(x_i) = y_i$, $i = 0,1,\cdots,n$，则可构造以下插值方法：

(1) Lagrange 插值多项式

已知 n 次插值基函数

$$l_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{(x - x_j)}{(x_i - x_j)} \ , i = 0,1,\cdots,n,$$

则 Lagrange 插值多项式为：

$$L_n(x) = \sum_{i=0}^{n} y_i l_i(x).$$

(2) Newton 插值多项式

由差商的定义，

$$f[\,x_0, x_1, \ldots, x_{\mathrm{k}}\,] = \frac{f[\,x_0, , \ldots, x_{\mathrm{k}-2}, x_{\mathrm{k}}\,] - f[\,x_0, x_1, \ldots, x_{\mathrm{k}-1}\,]}{x_{\mathrm{k}} - x_{\mathrm{k}-1}},$$

可得差商表：

x_{k}	$f(x_{\mathrm{k}})$	一阶差商	二阶差商	三阶差商
x_0	$f(x_0)$			
x_1	$f(x_1)$	$f[x_0, x_1]$		
x_2	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$	
x_3	$f(x_3)$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$
⋮	⋮	⋮	⋮	⋮

由此可得 Newton 插值多项式为：

$N_n(x) = f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \ldots +$
$f[x_0, x_1, \cdots, x_n](x - x_0) \ldots (x - x_{n-1}).$

(3) Hermite 插值多项式

插值多项式$H(x)$满足：$H(x_i) = f(x_i), H'(x_i) = f'(x_i), i = 0,1,2, \ldots, n.$ 若取 2 个节点，则可构造 3 次 Hermite 插值多项式，$H_3(x) = f(x_0)\alpha_0(x) +$

$f(x_1)\alpha_1(x)+f'(x_0)\beta_0(x)+f'(x_1)\beta_1(x)$, 其中，

$$\alpha_0(x)=\left(1+2\frac{x-x_0}{x_1-x_0}\right)\left(\frac{x-x_1}{x_0-x_1}\right),$$

$$\alpha_1(x)=\left(1+2\frac{x-x_1}{x_0-x_1}\right)\left(\frac{x-x_0}{x_1-x_0}\right)^2,$$

$$\beta_0(x)=(x-x_0)\left(\frac{x-x_1}{x_0-x_1}\right)^2,$$

$$\beta_1(x)=(x-x_1)\left(\frac{x-x_0}{x_1-x_0}\right)^2.$$

(4) **分段线性插值**

通过相邻两个插值节点做线性插值，则有

$$\mathrm{y}=f(x_i)+\frac{(f(x_{i+1})-f(x_i))(x-x_i)}{(x_{i+1}-x_i)},\qquad x\in[x_i,\ x_{i+1}],i=0,1,\cdots,n-1.$$

(5) **三次样条插值**

三次样条插值函数$S(x)$满足插值条件 $S(x_i)=f(x_i),i=0,1,2,\ldots,n$，并且在每个子区间$[x_i,x_{i+1}]$上，$S(x)$都是次数不大于 3 次的多项式，在整个区间$[x_0,x_n]$上有二阶的连续导数。

我们可采用上述插值算法对翼型坐标进行插值，然后拟合出轮廓曲线。

2.3 插值方法数值实验

一般情况下，当插值节点数目较多时，Lagrange 插值、Newton 插值会出现龙格现象，即在区间边缘出现的振荡现象，如图 3 和图 4 所示，由于采用了高次插值，图中出现了非常明显的龙格现象，并且无法收敛。

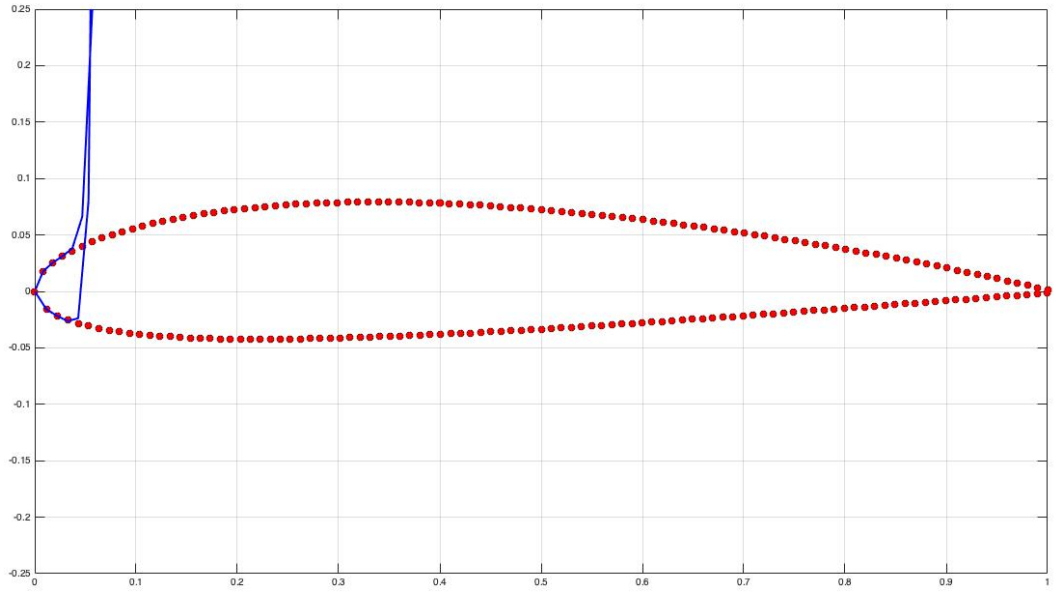

图 3 Lagrange 插值方法

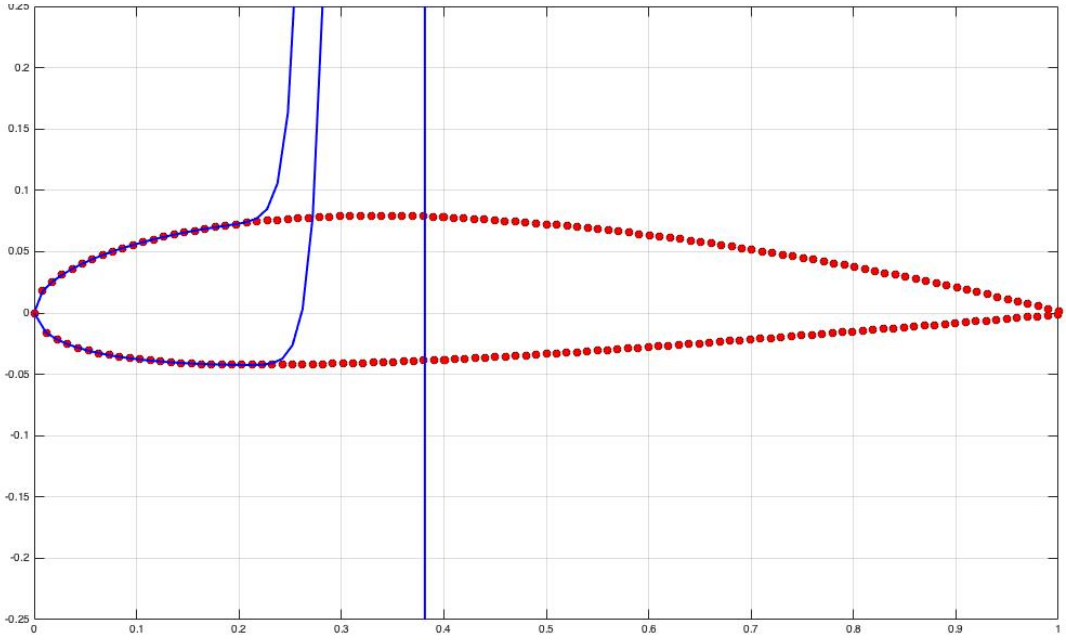

图 4 Newton 插值方法

而采用分段线性插值时，函数光滑性较差。对于当前问题，分段线性插值可以简单地看作将各点连线，数值结果如图 5 所示，图中可见翼型的光滑性较差。

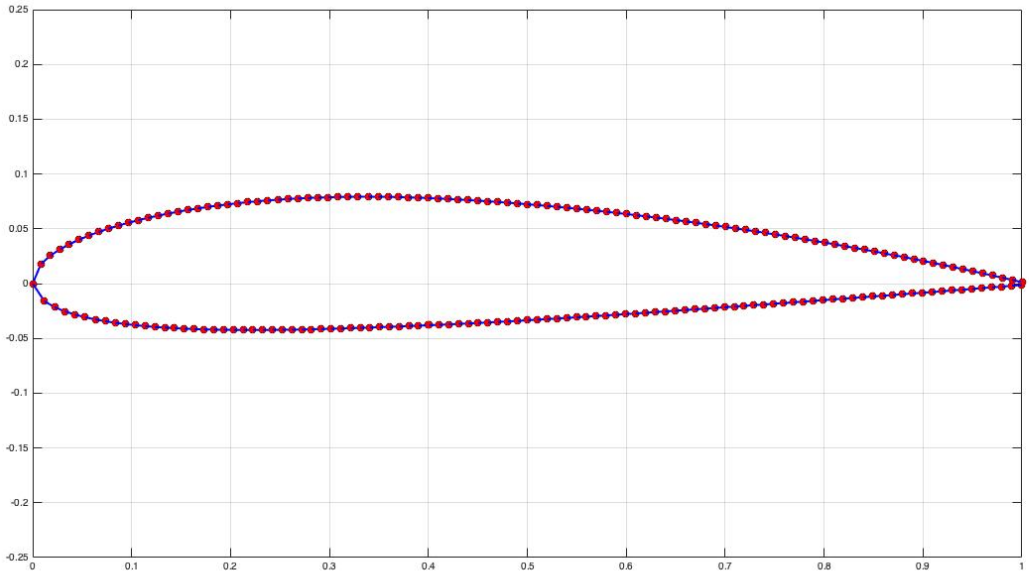

图 5 分段线性插值方法

因此，为了消除龙格现象，并提高翼型的光滑性，我们采用分段三次 Hermite 插值，数值结果如图 6 所示，由图可见翼型的光滑性得到了提升。

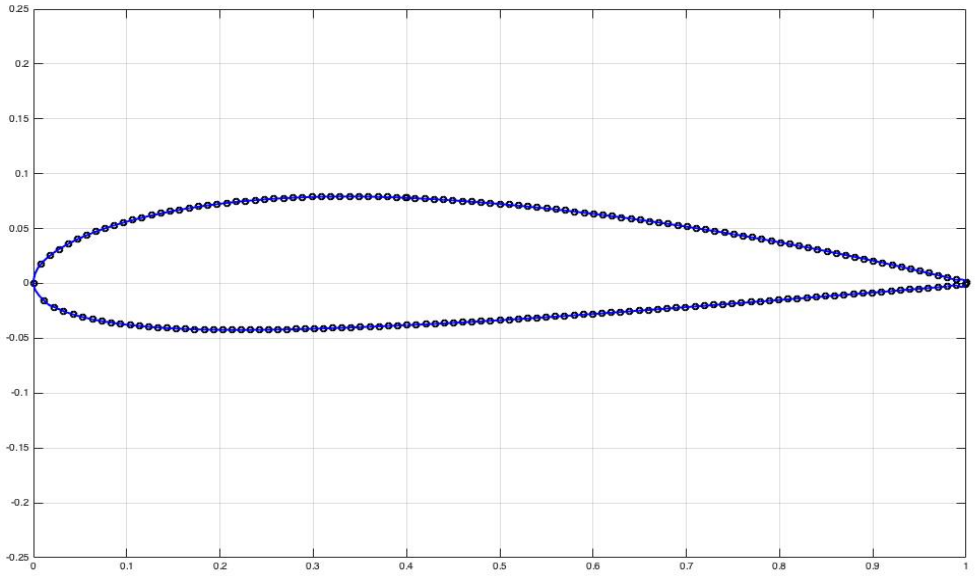

图 6 分段三次 Hermite 插值方法

为了达到二阶连续光滑度，我们可以采用三次样条插值，数值结果如图 7 所示，由图可见翼型的光滑性得到了进一步提升。

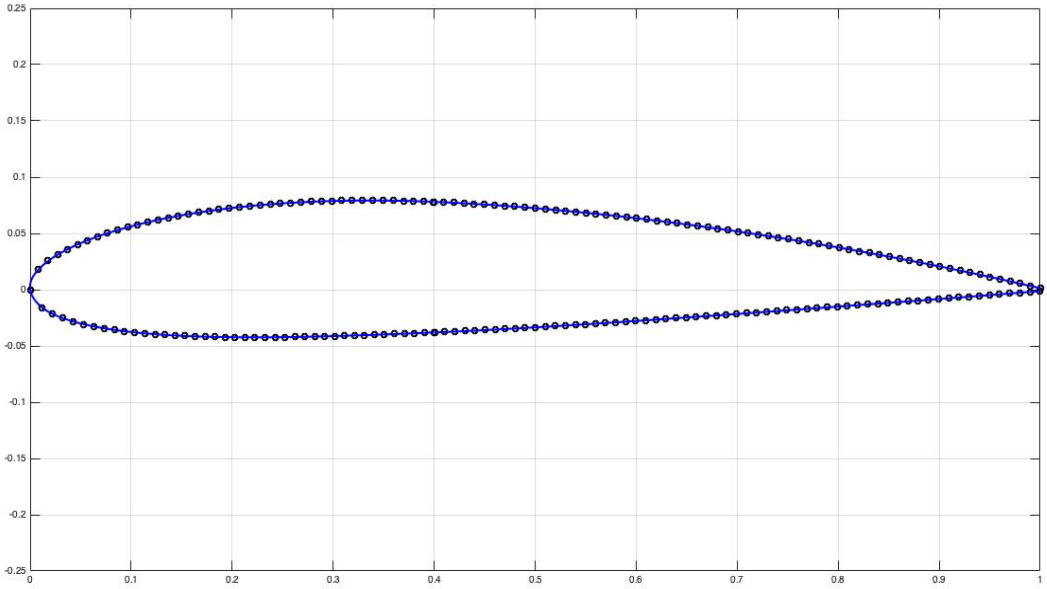

图 7 三次样条插值方法

目前的数值结果，可以用于后续翼型升力系数积分求解，进一步测量翼型的飞行数据。

2.4 结论

在求解过程中采用了 Lagrange 插值、Newton 插值、分段三次 Hermite 插值、分段线性插值以及三次样条插值方法，来对比各种差值方法的有效性和准确性，一般认为插值多项式的次数越高，逼近原函数的精度越好，但实际上并非如此，因为对于任意的插值节点，当$n \to \infty$时，插值多项式不一定收敛于原函数，由此可能产生龙格现象，使得插值结果随着插值次数变高越来越偏离结果。为了解决这种情况，引入了分段插值方法。分段线性插值方法是对相邻插值节点进行连线，相比于此，分段三次 Hermite 插值可以达到一阶导数连续，三次样条插值可以达到二阶导数连续，得到的翼型的光滑性明显改善。

3 附录

在此案例中，我们通过现有的翼型模型，获得翼型坐标数据，供数值实验使用。

基于 NACA 的翼型设计模型，它是依据理想流动理论和风洞吹风数据，通过对一系列性能好的翼型拟合所得到，用户可以自定义最大弯度（m）、最大弯度位置（p）和最大厚度（t）3 个参数，生成相应的翼型，具体参数如图 1 所示。如模型 NACA2412，其含义为：第一个数字 2 表示最大弯度（m）为弦长的 2%，第二个数字 4 表示最大弯度位置（p）为弦长的 40%，后两个数字 12 表示最大厚度（t）为弦长的 12%。

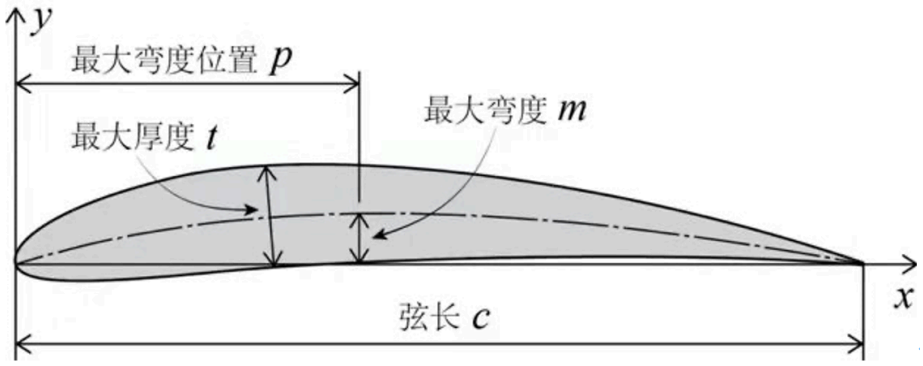

图 8　NACA 翼型图

厚度和弯度的分布规律如下：

弯度分布用中心线的y_c坐标表示：

$$y_c = \frac{m}{p^2}(2px - x^2), 0 \le x \le p,$$

$$y_c = \frac{m}{(1-p)^2}[(1-2p) + 2px - x^2], p \le x \le c.$$

厚度分布用半厚度y_t表示：

$$y_t = \frac{t}{0.2}(0.2969x^{0.5} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4).$$

进一步可给出翼型上下表面的曲线表达式：

$$x_{upper} = x - y_t sin\theta,$$

$$y_{upper} = y_c + y_t cos\theta,$$

$$x_{lower} = x + y_t sin\theta,$$

$$y_{lower} = y_c - y_t cos\theta,$$

$$\theta = \arctan\left(\frac{dy_c}{dx}\right).$$

在此尝试的是 NACA2412 翼型的 1 米模型，根据模型得到翼型数据源。（数据说明：xs1-ys1, xs2-ys2, xx1-yx1, xx2-yx2 为四组对应的翼型坐标，其中 xs1-ys1, xs2-ys2 表示翼型的上半部分，由x_{upper}，y_{upper}得到，xx1-yx1, xx2-yx2 表示翼型的下半部分，由x_{lower}，y_{lower}得到。）

xs1	ys1	xs2	ys2	xx1	yx1	xx2	yx2
0.000000	0.000000	0.399356	0.078028	0.000000	0.000000	0.400644	-0.038028
0.008347	0.017944	0.409424	0.077635	0.011653	-0.015969	0.410576	-0.037646
0.017768	0.025442	0.419492	0.077202	0.022232	-0.021542	0.420508	-0.037247
0.027384	0.031168	0.429559	0.076730	0.032616	-0.025393	0.430441	-0.036830
0.037107	0.035947	0.439625	0.076218	0.042893	-0.028347	0.440375	-0.036396
0.046901	0.040099	0.449690	0.075669	0.053099	-0.030724	0.450310	-0.035947
0.056750	0.043788	0.459755	0.075083	0.063250	-0.032688	0.460245	-0.035483
0.066640	0.047112	0.469818	0.074461	0.073360	-0.034337	0.470182	-0.035006
0.076565	0.050135	0.479880	0.073804	0.083435	-0.035735	0.480120	-0.034515
0.086519	0.052902	0.489941	0.073112	0.093481	-0.036927	0.490059	-0.034012
0.096498	0.055447	0.500000	0.072387	0.103502	-0.037947	0.500000	-0.033498
0.106498	0.057793	0.510058	0.071628	0.113502	-0.038818	0.509942	-0.032973
0.116517	0.059960	0.520115	0.070838	0.123483	-0.039560	0.519885	-0.032438
0.126552	0.061965	0.530170	0.070015	0.133448	-0.040190	0.529830	-0.031893
0.136602	0.063819	0.540223	0.069161	0.143398	-0.040719	0.539777	-0.031339
0.146666	0.065535	0.550275	0.068277	0.153334	-0.041160	0.549725	-0.030777
0.156741	0.067121	0.560325	0.067362	0.163259	-0.041521	0.559675	-0.030206
0.166826	0.068584	0.570373	0.066418	0.173174	-0.041809	0.569627	-0.029629
0.176921	0.069932	0.580420	0.065445	0.183079	-0.042032	0.579580	-0.029045
0.187024	0.071170	0.590464	0.064443	0.192976	-0.042195	0.589536	-0.028454
0.197135	0.072304	0.600507	0.063413	0.202865	-0.042304	0.599493	-0.027857
0.207252	0.073338	0.610547	0.062355	0.212748	-0.042363	0.609453	-0.027255
0.217375	0.074277	0.620586	0.061269	0.222625	-0.042377	0.619414	-0.026647
0.227504	0.075125	0.630622	0.060157	0.232496	-0.042350	0.629378	-0.026035
0.237637	0.075884	0.640656	0.059018	0.242363	-0.042284	0.639344	-0.025418
0.247774	0.076558	0.650688	0.057852	0.252226	-0.042183	0.649312	-0.024796
0.257914	0.077151	0.660718	0.056660	0.262086	-0.042051	0.659282	-0.024171
0.268057	0.077663	0.670745	0.055442	0.271943	-0.041888	0.669255	-0.023542
0.278203	0.078099	0.680770	0.054199	0.281797	-0.041699	0.679230	-0.022910
0.288351	0.078460	0.690793	0.052930	0.291649	-0.041485	0.689207	-0.022275
0.298500	0.078749	0.700813	0.051636	0.301500	-0.041249	0.699187	-0.021636
0.308650	0.078967	0.710831	0.050317	0.311350	-0.040992	0.709169	-0.020995
0.318802	0.079116	0.720847	0.048973	0.321198	-0.040716	0.719153	-0.020351
0.328953	0.079199	0.730859	0.047604	0.331047	-0.040424	0.729141	-0.019704
0.339105	0.079216	0.740870	0.046211	0.340895	-0.040116	0.739130	-0.019055
0.349256	0.079170	0.750877	0.044793	0.350744	-0.039795	0.749123	-0.018404

0.359407	0.079061	0.760882	0.043350	0.360593	-0.039461	0.759118	-0.017750
0.369557	0.078892	0.770884	0.041884	0.370443	-0.039117	0.769116	-0.017095
0.379706	0.078663	0.780883	0.040392	0.380294	-0.038763	0.779117	-0.016437
0.389854	0.078376	0.790880	0.038877	0.390146	-0.038401	0.789120	-0.015777
0.400000	0.078031	0.800874	0.037337	0.400000	-0.038031	0.799126	-0.015115
		0.810864	0.035773			0.809136	-0.014450
		0.820852	0.034184			0.819148	-0.013784
		0.830837	0.032571			0.829163	-0.013115
		0.840819	0.030933			0.839181	-0.012444
		0.850798	0.029271			0.849202	-0.011771
		0.860773	0.027584			0.859227	-0.011095
		0.870746	0.025873			0.869254	-0.010417
		0.880715	0.024136			0.879285	-0.009736
		0.890681	0.022374			0.889319	-0.009052
		0.900643	0.020587			0.899357	-0.008365
		0.910602	0.018775			0.909398	-0.007675
		0.920558	0.016936			0.919442	-0.006981
		0.930510	0.015072			0.929490	-0.006283
		0.940459	0.013182			0.939541	-0.005582
		0.950403	0.011265			0.949597	-0.004877
		0.960345	0.009322			0.959655	-0.004167
		0.970282	0.007352			0.969718	-0.003452
		0.980216	0.005354			0.979784	-0.002732
		0.990145	0.003329			0.989855	-0.002007
		1.000071	0.001276			0.999929	-0.001276

4 参考文献

1. 李庆阳，王能超，易大义编，数值分析，清华大学出版社，2008 年

2. 大飞机报《机翼那些事儿》
http://www.comac.cc/mjkp/xzs/201612/21/t20161221_4665263.shtml?from=timeline

3. 数值分析_维基百科 https://zh.wikipedia.org/zh/数值分析

5 Matlab 程序

%% NACA翼型

```
%NACA翼型
clear;clc;
%定义弦长、厚度、弯度、弯度最大值所在弦长的相对位置
c=1;t=0.12*c;
p=c*0.4;f=c*0.02;

x=0:1e-2:c;
%y厚度表达式
y=(t/0.20)*(0.29690.*sqrt(x)-0.12600.*x-0.35160.*x.*x+0.284330.*x.*x.*x-0.10150.*x.*x.*x.*x);
% plot(x,y,'k','LineWidth',2);
axis([0 c -0.25 0.25]);
hold on;
% plot(x,-y,'k','LineWidth',2);
% legend('????')
box on;grid on

%分段绘出中弧线
xz1=0:1e-2:p;
yz1=(f/(p.*p)).*(2.*p.*xz1-xz1.*xz1);
xz2=p:1e-2:c;
yz2=(f/((1-p).*(1-p))).*((1-2.*p)+2.*p.*xz2-xz2.*xz2);
% plot(xz1,yz1,'r','LineWidth',1);
% plot(xz2,yz2,'r','LineWidth',1);


%画出弦线
% plot(x,x.*0,'b','LineWidth',1)


%定义theta角
theta1=atan((f/(p.*p)).*(2.*p-2.*xz1));
theta2=atan(f/((1-p).*(1-p))).*((1-2.*p)+2.*p-2.*xz2);

%厚度的分段表达式
yt1=(t/0.20)*(0.29690.*sqrt(xz1)-0.12600.*xz1-0.35160.*xz1.*xz1+0.284330.*xz1.*xz1.*xz1-0.10150.*xz1.*xz1.*xz1.*xz1);
yt2=(t/0.20)*(0.29690.*sqrt(xz2)-0.12600.*xz2-0.35160.*xz2.*xz2+0.284330.*xz2.*xz2.*xz2-0.10150.*xz2.*xz2.*xz2.*xz2);
```

```
38
39      %最终成果
40
41      %x上和y上（分段绘出）
42 -    xs1=xz1-yt1.*sin(theta1);
43 -    ys1=yz1+yt1.*cos(theta1);
44      % plot(xs1,ys1,'k','LineWidth',2);
45 -    plot(xs1,ys1,'ko','LineWidth',2);
46 -    xs2=xz2-yt2.*sin(theta2);
47 -    ys2=yz2+yt2.*cos(theta2);
48      % plot(xs2,ys2,'k','LineWidth',2);
49 -    plot(xs2,ys2,'ko','LineWidth',2);
50
51      %x下和y下（分段绘出）
52 -    xx1=xz1+yt1.*sin(theta1);
53 -    yx1=yz1-yt1.*cos(theta1);
54      % plot(xx1,yx1,'k','LineWidth',2);
55 -    plot(xx1,yx1,'ko','LineWidth',2);
56 -    xx2=xz2+yt2.*sin(theta2);
57 -    yx2=yz2-yt2.*cos(theta2);
58      % plot(xx2,yx2,'k','LineWidth',2);
59 -    plot(xx2,yx2,'ko','LineWidth',2);
60      % title('NACA2412');
61      % legend('中弧线','弦线','翼型坐标')
62      % legend('NACA2412')
```

%% Lagrange 插值方法

```
1       %输入m个节点的值(xi,yi)(i = 1,2, ... , n+1)输入行向量X 和Y
2       %可以得到拉格朗日插值多项式 L，基函数 l,系数向量 C，系数矩阵 L1
3       function [C,L,L1,l] = lagran1(X,Y)
4 -     m = length(X); L = ones(m,m); %得到节点个数，得到多项式系数m-1
5 -     for k = 1 : m
6 -         V = 1;
7 -         for i = 1 : m
8 -             if k ~= i %取 i不等于 k
9 -                 V = conv(V,poly(X(i))) / (X(k) - X(i));  %得到第k个节点对应的基函数
10 -            end
11 -        end
12 -        L1(k, :) = V; l(k, :) = poly2sym(V);  %对应基函数记录到向量和矩阵中
13 -    end
14 -    C = Y * L1;
15 -    L = Y * l;  %对应基函数与函数值相乘
16 -    hold on;
17 -    plot(X,Y,'r*','Linewidth',2);
18 -    hold on;
19 -    plot(X,subs(L,X),'b','Linewidth',2);
20 -    hold on;
21      % legend('数据源','拉格朗日插值多项式');
22      % >> lagran1(xs1,ys1);
23      % >> lagran1(xs2,ys2);
24      % >> lagran1(xx1,yx1);
25      % >> lagran1(xx2,yx2);
```

%% Newton插值方法

```
%%求牛顿插值多项式
%输入的量:X是n+1个节点(xi,yi)(i = 1,2, ... , n+1)横坐标向量，Y是纵坐标向量，
%x是以向量形式输入的m个插值点，M在[a,b]上满足|f~(n+1)(x)|≤M
%注: f~(n+1)(x)表示f(x)的n+1阶导数
%输出的量: 向量y是向量x处的插值，误差限R，n次牛顿插值多项式L及其系数向量C，

function[y,R,A,C,L] = Newton_fun(X,Y,x,M)
n = length(X); %得到节点个数，得到多项式系数n-1
m = length(x); %得到要求的插值点的个数，要求m个插值点
for t = 1 : m  %对应每一个插值点求值
    z = x(t);
    A = zeros(n,n); %差商的矩阵A
    A(:,1) = Y'; %差商矩阵的第一列为对应Y的值
    s = 0.0; p = 1.0; q1 = 1.0; c1 = 1.0;
        for j = 2 : n
            for i = j : n
                A(i,j) = (A(i,j-1) - A(i-1,j-1))/(X(i)-X(i-j+1)); %对所有的差商求解
            end
            q1 = abs(q1*(z-X(j-1))); %为求误差，绝对值的相乘
            c1 = c1 * j; %??n?
        end
          C = A(n, n); q1 = abs(q1*(z-X(n)));  %为求误差，绝对值的相乘
          for k = (n-1):-1:1
              C = conv(C, poly(X(k)));%卷积，将对角线上差商与对应基函数相乘
              d = length(C);
              C(d) = C(d) + A(k,k);%在最后一维，也就是常数项加上新的差商
          end
          y(t) = polyval(C,z); %将第t个插值点对应的值计算得出
          R(t) = M * q1 / c1;  %得到对应误差
end
L = poly2sym(C);

hold on;
plot(X,Y,'r*','Linewidth',2);
hold on;
plot(X,subs(L,X),'b','Linewidth',2);
hold on;
% legend('数据源','牛顿插值多项式');
% >> Newton_fun(xs1,ys1,xs1,1);
% >> Newton_fun(xx1,yx1,xx1,1);
% >> Newton_fun(xs2,ys2,xs2,1);
% >> Newton_fun(xx2,yx2,xx2,1);
```

%% 分段线性插值方法

```
function y= T(x0,y0)                %将x0、y0以向量形式输入
 n=length(x0);                          %通过向量的长度表示节点的数目
for s=1:n-1
    X(1)=x0(s);
    X(2)=x0(s+1);
    Y(1)=y0(s);
    Y(2)=y0(s+1);
m = 2; L = ones(m,m);  %得到节点个数，得到多项式系数m-1
for k = 1 : m
    V = 1;
    for i = 1 : m
        if k ~= i %取 i不等于 k
            V = conv(V,poly(X(i))) / (X(k) - X(i));    %得到第k个节点对应的基函数
        end
    end
    L1(k, :) = V; l(k, :) = poly2sym(V);  %对应基函数记录到向量和矩阵中
end
C = Y * L1;
L = Y * l;    %对应基函数与函数值相乘
hold on;
plot(X,Y,'r*','Linewidth',2);
hold on;
plot(X,subs(L,X),'b','Linewidth',2);
hold on;
end;
hold on
% legend('数据源','分段线性插值多项式');
% >> T(xs1,ys1);
% >> T(xs2,ys2);
% >> T(xx1,yx1);
% >> T(xx2,yx2);
```

%% 分段三次Hermite插值方法

```
function  [m_matrix]=hermite3(x,y,y0,yn,x_value)
%% 输入值分配，x_input,y_input均为数组,y_0,y_n为x_0,x_n分别对应的一阶导数值
x_input = x;
y_input = y;
y_0 = y0;
y_n = yn;
%%
[~,number] = size(x_input); %获取输入x_input的大小1×number
delta_h = zeros(1,number-1); %给delta_h分配数组大小1×(number-1)，并全部初始化为0
delta_f = zeros(1,number-1); %给delta_f分配数组大小1×(number-1)，并全部初始化为0
lambda_ = zeros(1,number-2); %给lambda_分配数组大小1×(number-2)，并全部初始化为0
miu = zeros(1,number-2); %给miu_分配数组大小1×(number-2)，并全部初始化为0
e = zeros(1,number-2); %给e分配数组大小1×(number-2)，并全部初始化为0
% 计算delta_h、delta_f的值
for i = 1:(number-1)
    delta_h(i) = x_input(i+1) - x_input(i);
    delta_f(i) = (y_input(i+1) - y_input(i))/ delta_h(i);
end

%%计算lambda,miu,e
for i=1:number-2
    lambda_(1,i) = delta_h(1,i+1) / (delta_h(1,i+1) + delta_h(1,i));
     miu(1,i) = 1 - lambda_(1,i);
     e(1,i) = 3*(lambda_(1,i)*delta_f(1,i) + miu(1,i)*delta_f(1,i+1));
end

A = zeros(number-2,number-2); %初始化系数矩阵A，(n-2)*(n-2)
B = zeros(number-2,1); %初始化系数矩阵B，(n-2)*1
%当i=1时
A(1,1) = 2;
A(1,2) = miu(1,1);
B(1,1) = e(1,1) - lambda_(1,1) * y_0;

%当i=2:n-2时
for i = 2:number-3
    B(i,1) = e(1,i);
    A(i,i-1) = lambda_(1,i);
    A(i,i) = 2;
    A(i,i+1) = miu(1,i);
end
%当i=n-1时
A(number-2,number-3) = lambda_(1,number-2);
A(number-2,number-2) = 2;
B(number-2,1) = e(1,number-2) - miu(1,number-2)*y_n;

%% 计算A的逆，A*B
m_matrix = A\B;
m = zeros(1,number);
m(1,1) = y_0;
m(1,number) = y_n;
for i = 2:number-1
    m(1,i) = m_matrix(i-1,1);
end

```

```
for i =1:number-1
    % 获取相邻两点间的插值函数
    x_ = linspace(x_input(1,i),x_input(1,i+1));
    s1 = y_input(1,i).*((x_-x_input(1,i+1)).^2).*(delta_h(1,i) + 2.*(x_ - x_input(1,i)))./(delta_h(1,i).^3);
    s2 = y_input(1,i+1).*((x_-x_input(1,i)).^2).*(delta_h(1,i) + 2.*(x_input(1,i+1) - x_))./(delta_h(1,i).^3);
    s3 = m(1,i).*((x_ - x_input(1,i+1)).^2).*(x_ - x_input(1,i))./(delta_h(1,i).^2);
    s4 = m(1,i+1).*((x_ - x_input(1,i)).^2).*(x_ - x_input(1,i+1))./(delta_h(1,i).^2);
    s = s1 + s2 + s3 +s4;
    % 判断输入的x属于哪个插值区间，满足则计算对应的f(x)的值
    if x_value>x_input(1,i)&x_value<x_input(1,i+1)
        s1_ = y_input(1,i)*((x_value-x_input(1,i+1)).^2)*(delta_h(1,i) + 2.*(x_value - x_input(1,i)))./(delta_h(1,i).^3);
        s2_ = y_input(1,i+1)*((x_value-x_input(1,i)).^2)*(delta_h(1,i) + 2.*(x_input(1,i+1) - x_value))./(delta_h(1,i).^3);
        s3_ = m(1,i)*((x_value - x_input(1,i+1)).^2)*(x_value - x_input(1,i))./(delta_h(1,i).^2);
        s4_ = m(1,i+1)*((x_value - x_input(1,i)).^2)*(x_value - x_input(1,i+1))./(delta_h(1,i).^2);
        s_ = s1_ + s2_ + s3_ +s4_;
        % 绘制计算点的结果
        plot(x_value,s_,'ro');
    end
    % 绘制每一段插值函数图像
    plot(x_,s,'b','Linewidth',2);
    hold on;
end
%%
% hermite3(xs1,ys1,1,0,xs1)
% hermite3(xs2,ys2,0,-1,xs2)
% hermite3(xx1,yx1,-1,0,xx1)
% hermite3(xx2,yx2,0,1,xx2)

% hermite3(xs1(2:41),ys1(2:41),1,0,xs1)
% hermite3(xs2,ys2,0,-1,xs2)
% hermite3(xx1(2:41),yx1(2:41),-1,0,xx1)
% hermite3(xx2,yx2,0,1,xx2)
% lx(1)=yx1(2);lx(2)=yx1(1);lx(3)=ys1(2),ly(1)=xx1(2);ly(2)=xx1(1);ly(3)=xs1(2);
% xx=-0.0160:0.005:0.0179
% ww = pchip(lx,ly,xx);
% plot(ww,xx,'b','Linewidth',2);
```

%% 三次样条插值方法

```
x_1=0.0083:0.001:0.4
x_2=0.0117:0.001:0.4
x2=0.4:0.001:1
ps1=spline(xs1(2:41),ys1(2:41),x_1);
plot(x_1,ps1,'b','Linewidth',2);
ps2=spline(xs2,ys2,x2);
plot(x2,ps2,'b','Linewidth',2);
px1=spline(xx1(2:41),yx1(2:41),x_2);
plot(x_2,px1,'b','Linewidth',2);
px2=spline(xx2,yx2,x2);
plot(x2,px2,'b','Linewidth',2);
lx(1)=yx1(2);lx(2)=yx1(1);lx(3)=ys1(2),ly(1)=xx1(2);ly(2)=xx1(1);ly(3)=xs1(2);
xx=-0.0160:0.00339:0.0179
ww = spline(lx,ly,xx);
plot(ww,xx,'b','Linewidth',2);
```