《化工数值计算与 MATLAB》复习指南

——计算机化工应用 || **ChemEngMATLAB**

作者：王世强

组织：华东理工大学化工学院

时间：December 28, 2019

版本：0.4 || 本文模板来自ElegantLATEX 项目组

ElegantLATEX Program

注：封面图选自《日常》第零集

目　录

绪论部分 **2**

0.1 误差 2

0.2 浮点数与浮点数运算 2

0.3 MATLAB 的通用命令 3

1 MATLAB 程序设计语言与初等数学运算 **4**

1.1 变量与数据 4

1.2 数据输出 5

1.3 图形输出 6

1.4 函数 8

1.5 关系与逻辑运算 8

1.6 MATLAB 程序控制 9

2 矩阵操作与线性方程组求解 **10**

2.1 矩阵生成与性质 10

2.2 矩阵操作与分析 11

2.3 线性方程组的求解 12

3 非线性方程 (组) 求解 **14**

3.1 求解函数 14

3.2 常用数值求解方法 16

4 插值与拟合 **17**

4.1 插值函数的基本概念 17

4.2 MATLAB 中的插值函数 19

4.3 拟合函数的基本概念 20

4.4 MATLAB 中的拟合函数 20

5 数值微分与积分 **24**

5.1 数值微分 24

5.2 数值积分 25

6 常微分方程数值解 **27**

6.1 常微分方程 27

6.2 MATLAB 解常微分方程 27

写在前面[1]

期末临近，本文将作为华理化工学院专业必修课程“计算机化工应用”的复习指南。

本文制作过程中主要参考了隋志军老师的教材《化工数值计算与 MATLAB》、作业、课件，并参考了付金硕学长的相关资料。在文档制作过程中还得到了 LaTeX 技术交流 1 群与 LaTeX 科技排版工作室中网（da）友（lao）们的帮助，在此感谢他们！

```
% 此环境下主要介绍函数的写法
其中，⟨空缺⟩通常表示占位符。
while ⟨condition⟩
    if ⟨something-happens⟩
        % do something useful
    end
end
```

```
% 此环境下主要给出例题的解答，各位可以直接复制到MATLAB中运行
function exam5_3_1
t=3:3:30;
G1=[10.2 13.9 12.16 13.49 13.74 12.01 11.55 11.02 12 12.12];
G2=[6.91 8.94 8.69 10.09 10.63 9.58 9.53 9.29 10.13 10.24];
A=1.83;
X=(G1-G2)./G2;
tt=linspace(3,30);
% 插值
pp=spline(t,X);
ppv=-fnval(pp,tt)/A;
% 微分
dp=fnder(pp);dpv=fnval(dp,tt);
plot(t,X,'ro',tt,ppv,tt,dpv,'.-')
axisX=refline(0,0)
axisX.Color='c'
legend('原始数据','拟合曲线','导数曲线','导数参考直线')
```

时间仓促，错误[2]与疏漏[3]在所难免，请各位辩证地使用！

最后，预祝各位期末顺利，门门 4.0！

[1]点击目录后的页码可快速跳转到指定内容，也可以在书签页快速浏览。
[2]各位如果发现写错的地方，请联系我 QQ568365675!
[3]后续更新于https://github.com/WsinGithub/ChemEngMATLAB

绪论部分

内容提要

- 误差
- 浮点数及其运算
- MATLAB 的常用命令

0.1 误差

0.1.1 误差来源

- **模型误差**，问题简化过程产生
- **截断误差**，有限次运算限制产生
- **舍入误差**，机器字长限制产生

有关误差定义不再赘述，举 Work1 中一例：

例题 0.1 已知某化工管道的真实长度为 1000m，某次测量结果为 1001m，其测量的绝对误差，相对误差为多少？

解 绝对误差为 1m，相对误差为 0.1%。

0.2 浮点数与浮点数运算

0.2.1 浮点数

由于计算机资源的有限，在计算机上只能表示有限的实数，这些数被称为浮点数。浮点数有如下性质：**有限个、有界、非连续**。

0.2.2 IEEE 标准双精度浮点运算体系

定义 0.1. 特殊浮点数

最大实数（上溢：超过用 inf 或 Inf 表示）：

$$real_{max} = 1.79e + 308$$

最小正实数（下溢：若为小于其的正数，则记为 0）：

$$real_{min} = 2.225e - 308$$

从 1 到下一个较大浮点数的距离（机器精度）：

$$eps = 2^{-52} = 2.220e - 16$$

♣

0.2.3 浮点数运算

浮点数运算，加法和乘法运算交换律仍然适用，但是其结合律和分配律已不再适用。

定义 0.2. NaN

Not a Number，非数。♣

计算下列情况时会出现，

- 0 / 0
- ∞/∞
- (+Inf) + (-Inf)
- 0 * Inf

0.3 MATLAB 的通用命令

定义 0.3. 常用命令

clc：清除命令窗口内容

clear：清除内存变量

save：保存内存变量到指定文件 ♣

```
clc,clear %通常用作初始化
```

第 1 章 MATLAB 程序设计语言与初等数学运算

内容提要

- ❏ 变量与数据类型
- ❏ 数据 / 图形输出
- ❏ 逻辑运算
- ❏ 函数
- ❏ 控制语句

注意 本章内容较繁杂，仅列举考试主体，细节请各位自行翻阅课本及课件！

1.1 变量与数据

1.1.1 变量与数据

- 命名规则
- 常用变量

1.1.2 数据类型

- 数值（向量、矩阵，用 [] 标识，注意区分 , / ␣ 与 ; ）
- 字符（用'⟨*string*⟩' 标识）
- 单元数组，cell array（注意'{}' 的使用方法）
- 结构体,structure（用'.' 标识）
- 函数句柄

1.1.3 生成向量

```
a=1:10 %默认以1为间距
a=1:2:10 %以2为间距
% 线性等分
y=linspace(<start>,<end>(,<n>))
% 对数等分
y=logspace(<start>,<end>(,<n>))
```

1.2 数据输出

1.2.1 disp 函数

```
%显示⟨X⟩到屏幕，并自动换行
disp (⟨X⟩)
% 输出矩阵
a=[1 2;3 4];
disp(a)
% 输出字符
disp('A') %将输出字母A
disp(['1+1=',num2str(2)]) %使用 [] 连接字符
```

1.2.2 fprintf 函数

定义 1.1. 常用转义字符

%n 换行 %t 制表符 %f 固定位数小数 %s 字符或字符串 %e 指数形式 ♣

举例如下：

```
% Work2_4 计算压降
L=3000;d=45;V=1600;
deltP=0.03*L*(V/1000)^1.84/d^1.24;
disp('L=3000m d=45mm V=1600m/min')
disp('压降计算值为：')
fprintf('\tdeltP=%.2e\n',deltP)
```

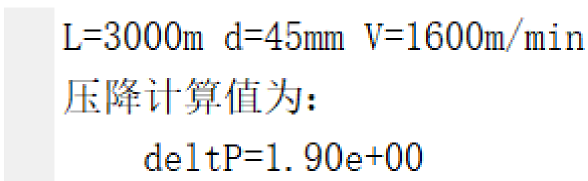

图 1.1: 运行结果

1.3 图形输出

1.3.1 基本概念

```
% 可将多条曲线绘制于同一图形窗口
% '⟨S⟩'对线型、颜色、数据点形貌等设置
% 线型 -实线 : 虚线 -.点划线 --双画线
% 颜色 b蓝色 g绿色 r红色 k黑色
% 点貌 .黑心实点 o空心圆圈 p五角星符
plot(⟨X1⟩,⟨Y1⟩,'⟨S1⟩',⟨X2⟩,⟨Y2⟩,'⟨S2⟩',⋯)
```

示例如下：

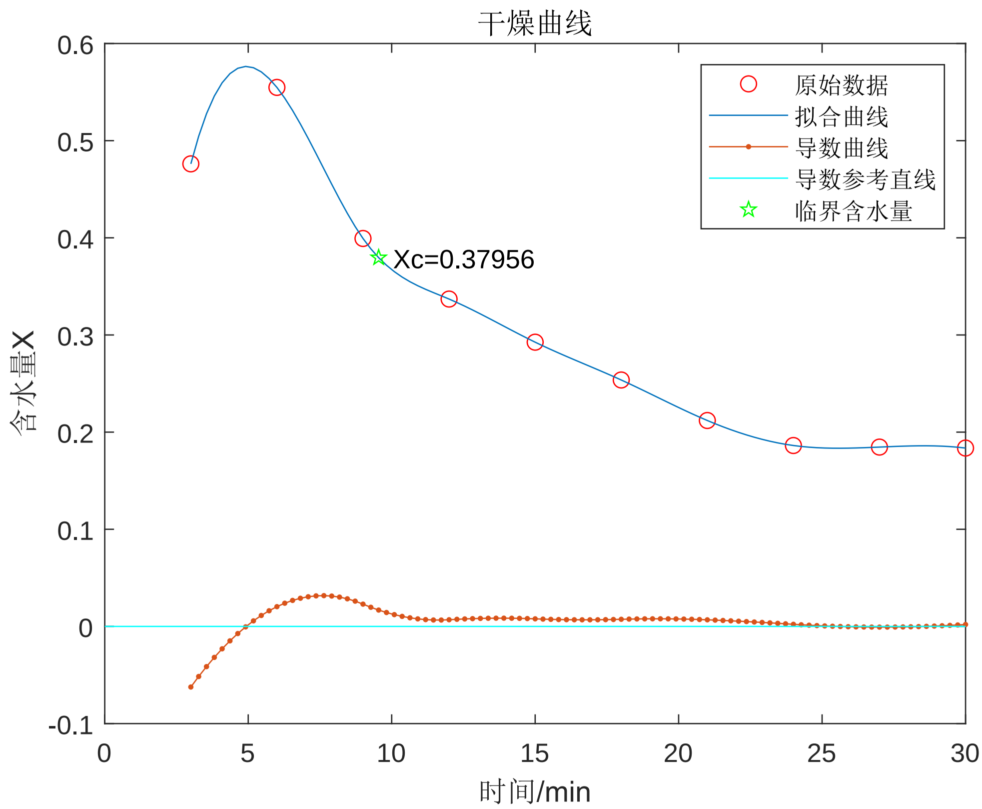

图 1.2: 运行示例

```
% 数据来自上机材料5 3.1
function exam5_3_1
t=3:3:30;
G1=[10.2 13.9 12.16 13.49 13.74 12.01 11.55 11.02 12 12.12];
G2=[6.91 8.94 8.69 10.09 10.63 9.58 9.53 9.29 10.13 10.24];
A=1.83;
X=(G1-G2)./G2;
tt=linspace(3,30);
% 插值
pp=spline(t,X);
ppv=fnval(pp,tt);
plot(t,X,'ro') %绘制原始数据，红色圆圈
hold on %保持图形窗口，继续绘制曲线
plot(tt,ppv) %绘制拟合曲线，观察插值结果
% 微分
figure %打开新图形窗
dp=fnder(pp);
dpv=-fnval(dp,tt)./A;% 干燥速率
plot(t,X,'ro',tt,ppv,tt,dpv,'.-') %同时绘制两条曲线
axisX=refline(0,0) %绘制斜率0，截距0的参考直线
axisX.Color='c'
% 计算临界含水量并标注
Um=max(dpv);
loc=find(dpv>Um/2,1,'last'); %临界含水量在向量中索引
tc=tt(loc);% 临界时间
hold on
plot(tc,ppv(loc),'gp') %临界含水量
% 图名
title('干燥曲线')
% 添加图例
legend('原始数据','拟合曲线','导数曲线','导数参考直线','临界含水量')
% 设置坐标轴
xlabel('时间/min')
ylabel('含水量X')
% 文字标注
text(tc+0.5,ppv(loc),strcat('Xc=',num2str(ppv(loc))))
```

1.4 函数

1.4.1 基本用法与子函数

```
% 注意：[]与()的使用
function [⟨y1⟩,⟨y2⟩,..]=FunName(⟨x1⟩,⟨x2⟩,..)
%主函数下，没有输入，只有输出的子函数，如果不位于末尾，需要写end
%仅能在主函数内部调用，无法在外部调用
function y=SubFunName
```

1.4.2 匿名函数

函数 $f(x, y) = x^2 + y^2 = -1$ 可表示如下：

```
f=@(x,y) x^2+y^2-1;
f(1,2) %输出为4
```

1.5 关系与逻辑运算

1.5.1 主要运算符与函数

关系操作符：==、 =、>

关系运算函数：find() 非零元素下标

```
a=[1 2 3];
find(a>=2) %输出为[2 3]
```

逻辑运算：& (与) | (或) ~ (非) xor (异或)

```
a=[0 1 1];b=[0 0 1];
find(a&b) %输出为 3
```

1.5.2 优先级

<table>
<tr><th>优先级</th><th colspan="6">运算符</th></tr>
<tr><td>1</td><td>()</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>.'</td><td>'</td><td>.^</td><td>^</td><td></td><td></td></tr>
<tr><td>3</td><td>代数正</td><td>代数负</td><td>~</td><td></td><td></td><td></td></tr>
<tr><td>4</td><td>.*</td><td>.\</td><td>./</td><td>*</td><td>\</td><td>/</td></tr>
<tr><td>5</td><td>+</td><td>-</td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>:</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>7</td><td><</td><td>></td><td>==</td><td>>=</td><td><=</td><td>~=</td></tr>
<tr><td>8</td><td>&</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>9</td><td>|</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>10</td><td>&&</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>11</td><td>||</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

图 1.3: 运算优先级

1.6 MATLAB 程序控制

1.6.1 if 选择语句

```
if ⟨condition1⟩
    ⟨statements1⟩
elseif ⟨condition2⟩
    ⟨statements2⟩
else
    ⟨statements3⟩%譬如error(〈报错内容〉)
end
```

1.6.2 for 循环语句

```
for 〈循环变量〉=〈初值〉:〈步长〉:〈终值〉
    〈循环语句〉
end
```

1.6.3 whlie 循环语句

```
while 〈条件表达式〉
    〈循环语句〉
end
```

注意 了解 continue、break、return 函数的作用。break 跳出该层循环，continue 进入该层循环的下一次迭代，return 退出程序或函数返回。

第 2 章 矩阵操作与线性方程组求解

内容提要

- cat、repmat 函数
- find、max、sum 函数
- 索引与下标

2.1 矩阵生成与性质

2.1.1 矩阵的拼接与复制

```
% 拼接矩阵
cat(⟨DIM⟩,A,B)
% 举例如下
A=[1 2];
B=[3 4];
% DIM=1，按行拼接
cat(1,A,B) %将得到[1 2 3 4]
% DIM=2，按列拼接
cat(2,A,B) %将得到[1 2;3 4]
```

```
% 将矩阵A复制M行N列
repmat(A,⟨M⟩,⟨N⟩)
% 举例如下
A=[1 0];
repmat(A,2,3) % 将得到[1 0 1 0 1 0;1 0 1 0 1 0]
```

2.1.2 常见工具矩阵

```
[] %空阵
zeros(n); %n阶全零阵
zeros(m,n); %m行n列全零阵
%ones用法同zeros，全1阵
```

2.1.3 矩阵基本性质函数

```
% size函数与length函数
X=zeros(2,3);
D=size(X) %得D=[2,3], 行数与列数
L=length(X) %得L=3, 相当于max(size(X))
```

2.2 矩阵操作与分析

2.2.1 索引与下标

```
% 索引
A(n) %按照(维数>)列>行的顺序标注矩阵元素
% 举例
A=[1,2,3;4,5,6];
A(1:6) %先列后行, 结果为 1 4 2 5 3 6
A([end,end-1]) %结果为 6 3
```

```
% 下标
A(m,n) %第m行第n列元素
% 举例
A=[1,2,3;4,5,6];
A(2,1) %结果为4
```

```
% 排序
sort(A) %第m行第n列元素
% 举例
A=[1,2,3;4,5,6];
A(2,1) %结果为4
```

2.2.2 逻辑与关系运算与查找

```
% 查找元素值
A(⟨condition⟩)
% 举例
A=[1,2,3;4,5,6];
A(A-2==0|A-3==0|A-4==0) %按列查找, 结果为[4;2;3]
```

```
% 查找元素索引
find(⟨condition⟩)
```

```
% 举例
A=[1,2,3;4,5,6];
find(A-2==0|A-3==0|A-4==0) %按列查找, 结果为[2;3;5]
```

```
% 查找元素最大值
max(A)
% 举例
A=[1,2,3;4,5,6];
max(A) %每列最大值, 结果为[4 5 6]
[Rmax I]=max(A) %Rmax=[4 5 6],I=[2;2;2],返回在指定维度中的位置
```

2.2.3 矩阵分析

```
% 求和
sum(A)
% 举例
A=[1,2,3;4,5,6];
sum(A) %按列求和, 结果为[5 7 9]
sum(A,2) %按行求和, 结果为[6;15]
```

2.3 线性方程组的求解

2.3.1 高斯消元法

定义 2.1. 高斯消元法

基本思路为：化为上三角阵，回代求解。 ♣

对于方程组

$$AX = \boldsymbol{b}$$

解可以表示为

$$X = A\_\boldsymbol{b}$$

2.3.2 作答模板

```
% 得到所需矩阵A、b，注意对齐方式！
A=[〈系数矩阵〉]; b=[〈增广矩阵〉];
% 使用左除命令求解
x=A\b;
```

例题 2.1 假设一混合物由硝基苯 $C_6H_5NO_2$、苯胺 C_6H_7N、氨基丙酮 C_3H_7NO 和乙醇 C_2H_6O 组成。对该混合物进行元素分析，结果各元素[1]的质量百分数为：$w_C = 57.78\%, w_H = 7.92\%, w_N = 11.23\%, w_O = 23.07\%$。试编写一个 MATLAB 函数:

1) 确定上面四种化合物在混合物中所占的质量百分数，采用 fprintf 函数将结果显示在屏幕上；

2) 检验求得的质量分数之和是否为 1，如果是则在屏幕上显示信息：Calculation succeed。如果否，则显示警告信息：The calculation is wrong。

解

```
function example2_1
N=[6 5 1 2;6 7 1 0;3 7 1 1;2 6 0 1]; %四种分子中的C、H、N、O数,顺序下同
M=[12 1 14 16];% 原子量
M=repmat(M,4,1);
A=M.*N; % 原子量之和
% 分子中，四种元素的质量分数
A(1,:)=A(1,:)/sum(A(1,:));
A(2,:)=A(2,:)/sum(A(2,:));
A(3,:)=A(3,:)/sum(A(3,:));
A(4,:)=A(4,:)/sum(A(4,:));
b=[0.5778 0.0792 0.1123 0.2307]';% 总质量分数
x=A'\b;% 关键命令!
% 两位精度显示，并自动换行
fprintf('The percentage of C6H5NO2 is %.2f%%\n',100*x(1))
fprintf('The percentage of C6H7N is %.2f%%\n',100*x(2))
fprintf('The percentage of C3H7NO is %.2f%%\n',100*x(3))
fprintf('The percentage of C2H6O is %.2f%%\n',100*x(4))
if sum(x)==1
    disp('Calculation succed')
else
    warning('The calculation is wrong')
end
```

```
The percentage of C6H5NO2 is 38.62%
The percentage of C6H7N is 15.16%
The percentage of C3H7NO is 23.74%
The percentage of C2H6O is 22.49%
Calculation succed
```

图 2.1: 运行结果

[1]原子量：C 为 12，H 为 1，O 为 16，N 为 14。

第 3 章 非线性方程 (组) 求解

内容提要

- ❑ 求解函数 fzero、fslove、roots
- ❑ 数值求解方法

3.1 求解函数

3.1.1 fzero 函数

fzero 函数是 MATLAB 里用于求解单个非线性方程的函数。它的基本使用格式如下：

```
% 完整形式
[x,fval,exitflag,output]=fzero(fun,x0,options, p1, p2, ...)
```

```
% 常用形式
x=fzero(⟨fun⟩,⟨x0⟩)
% 其中，⟨fun⟩为匿名函数或函数句柄
% ⟨x0⟩为迭代初值，
% 也可为区间[x1,x2]，函数值在端点处异号，在端点内求根
% x为x0附近的解，如有多根，则与x0选取有关
```

对于 '⟨fun⟩'的使用举例如下:

```
% 匿名函数
fun1=@(x) x-1;
fzero(fun1,0) %结果为1
```

```
% 函数句柄
% 编写.m文件（函数文件也可）
fzero(@fun2,0) %结果为1
function y=fun2(x)
y=x-1;
end %脚本中的所有函数都必须以 'end' 结束，函数(文件)中则不需要
```

二者实现的结果类似。

3.1.2 fsolve 函数

fsolve 函数是 MATLAB 里用于求解非线性方程组的函数。它的基本使用格式如下：

```
% 完整形式
[x,fval,exit]=fsolve(fun,x0,option)
```

对于求解方程组[1]：

$$y=\begin{cases}x^2+y^2-1=0\\0.75x^3-y+0.9=0\end{cases}$$

可新建函数文件求解：

```
% 新建文件ExperFsolve.m
function ExperFsolve
x0=[0 0];
[x,fval,exit]=fsolve(@fun,x0)
function y=fun(x)
y=zeros(2,1);
y(1)=x(1)^2+x(2)^2-1;
y(2)=0.75*x(1)^3-x(2)+0.9;
```

也可使用匿名函数求解, 二者运行结果相同：

```
x0=[0 0];
f=@(x) [x(1)^2+x(2)^2-1;0.75*x(1)^3-x(2)+0.9];
[x,fval,exit]=fsolve(f,x0)
```

注意 fzero 函数和 fsolve 函数的运行结果为初值附近的解，而非全部解。

3.1.3 roots 函数

fsolve 函数是 MATLAB 里用于求解多项式的函数。它的基本使用格式如下：

```
r=roots(⟨p⟩)
% 其中⟨p⟩为n次多项式行向量,从高(n次)到低(0次)排列
```

譬如，求解方程 $x^2-1=0$,

```
roots([1 0 -1]) %结果为[-1;1]
```

注意 roots 函数可以获得多项式的所有根。

[1]摘自上机材料 3

3.2 常用数值求解方法

非线性方程求数值解方法有：

逐步扫描法，二分法，牛顿法，割线法，逆二次插值...

定义 3.1. 二分法

[基本思想]

区间逐次二等分至精度要求。

[求解精度]

可靠，只要次数足够多一定能到指定精度。

[效率]

收敛速度较慢。♣

命题 3.1. 收敛性对比

[牛顿法]

二次收敛；收敛速度不稳定，某些条件下收敛慢或根本不收敛；导数计算不便。

[截弦法]

超线性收敛；类似牛顿法；无需计算导数。

[逆二次插值]

三点构造抛物线交坐标轴迭代；整个过程收敛速度不稳定；接近终点时迭代快。♠

第 4 章 插值与拟合

内容提要

- ❑ 插值与插值函数
- ❑ interp1，spline，pchip
- ❑ 拟合与拟合函数
- ❑ regress、ployfit、nlinfit

4.1 插值函数的基本概念

4.1.1 插值函数

定义 4.1. 插值函数

a）插值函数 $\phi(x)$ 近似于原函数 $f(x)$

$$y = f(x) \approx \phi(x)$$

b) 在插值点 x_i 处，插值函数值 $\phi(x_i)$ 等于原函数值 $f(x_i)$

$$\phi(x_i) = f(x_i) = y_i$$

♣

常见形式有：

拉格朗日多项式插值、分段三次埃米特插值、分段三次样条插值...

4.1.2 拉格朗日插值法

定义 4.2. 拉格朗日插值法

拉格朗日 n 次插值多项式：

$$P_n(x) = \sum_{i=0}^{n} y_i L_i(x)$$

需要 (n+1) 个样本点, 为了实现

$$P_n(x_i) = y_i$$

我们让基函数 $L_i(x)$ 满足

$$L_i(x) = \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{if } x \neq x_i \end{cases}$$

我们令

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

即为基函数定义。

♣

Lagrange 插值多项式的优点在于不要求数据点是等间隔的，其缺点是数据点数不宜过大，通常不超过 7 个，否则计算工作量大且误差大，计算不稳定。

定义 4.3. 龙格现象

当次数 n 增大时，部分区间上误差严重，这种现象称“龙格现象”。

[解决方法]：分段插值。 ♣

注意 拉格朗日插值多项式次数高，不等于插值效果好！

4.1.3 三次样条插值

需要知道：区间 $[a, b]$ 内节点 x_i 处函数值 y_i，以及端点附近导数值等。

定义 4.4. 样条插值函数 $S(x)$

满足条件：

a)$S(x)$ 在分段区间 $[x_i, x_{i+1}]$ 上为三次多项式

b)$S(x)$、$S'(x)$、$S''(x)$ 在 $[a, b]$ 上连续

c) 节点处 $S(x) = y_i$ ♣

注意 [自然边界条件[1]]： $S''(x_0) = S''(x_n) = 0$

4.1.4 分段三次埃米特插值

需要知道：x_i 处的函数值 y_i 和导数值 y_i'。

定义 4.5. Hermite 插值函数 $H(x)$

a) 每个插值区间上 H(x) 为三次多项式

b) 插值函数值等于节点值

$$H(x_i) = y_i$$

c) 插值函数一阶导数等于节点处导数值

$$H'(x_i) = y_i'$$

♣

注意 二者不同之处在于一阶导数的确定方法不同。其中，埃米特插值可保持函数形状。

[1]我的理解是：可在边界附近有较好的单调性。

4.2 MATLAB 中的插值函数

4.2.1 interp1 函数

```
% 调用格式
yi=interp1(x,y,xi,'⟨method⟩')
% x，y为样本数据，xi为插值点
% ⟨method⟩省略时默认线性插值
% ⟨method⟩可选：
% 最近插值nearest、埃米特插值pchip、样条插值spline
```

4.2.2 pichp 函数与 spline 函数

```
% 调用格式
% 插值函数值
yi=pchip(x,y,xi)
% 插值函数
pp=pchip(x,y)
% spline函数使用方式相同，从略
```

以下三种调用形式结果相同：

```
y1=interp1(x,y,xi,'pchip')
```

```
yi=pchip(x,y,xi)
```

```
pp=pchip(x,y);
yi=fnval(pp,xi)
```

注意 可以参考4.4.4中的关于函数 fnval 的介绍，应用见5.1.3。

4.3 拟合函数的基本概念

拟合函数**不要求**函数完全通过数据点，常用方法为**最小二乘法**。

定义 4.6. 最小二乘法

[原理]

近似函数在各实验点的计算结果与实验结果的偏差平方和最小。

[线性最小二乘]

可通过解方程组得到拟合函数。

[非线性最小二乘]

a) 线性化；

b) 直接迭代。

♣

4.4 MATLAB 中的拟合函数

4.4.1 ployfit 函数

polyfit 是 MATLAB 的多项式拟合函数。

```
% 调用格式
p=polyfit(x,y,n)
% x,y为样本，n为多项式次数
```

常与 polyval 联合使用，计算指定点的值:

```
% 调用格式
yy=polyval(p,xx)
```

注意 此处 p 定义与 roots 函数3.1.3中相同，请对照理解。

4.4.2 regress 函数

regress 是 MATLAB 的多元线性拟合 函数。

```
% 调用格式
b=regress(y,x)
% n个自变量xi，进行m次实验，得到m次结果yi
% y为m行1列向量
% x为m行n列矩阵
```

注意 可以拟合线性化后的非线性函数。

例题 4.1 已知 x1=1:6，x2=1.5:0.2:2.5，y=[5.9512 10.6730 15.5676 20.6234 25.8599 31.2622]。试编写函数，采用以上数据拟合 y=a*x1+b*x2+c*x1^2+d*x1*x2 中的系数[2]。

解 将 x1, x2, x1^2 和 x1*x2 视为自变量，则以上的拟合为多元线性拟合，可以采用 regress 函数求解。

```
function LinearFitting
x1=1:6;
x2=1.5:0.2:2.5;
y=[5.9512 10.6730 15.5676 20.6234 25.8599 31.2622];
X=[x1;x2;x1.^2;x1.*x2];
% 关键函数
% 注意按照列的形式输入
b=regress(y',X') %结果为[0;1.0775;-0.5687;3.2694]，即[a;b;c;d]
```

可进一步作图观察拟合结果：

```
bb=repmat(b,1,6);
yi=sum(bb.*X);
plot(x1,y,'rp',x1,yi,'bo')
legend('原始数据','拟合数据')
```

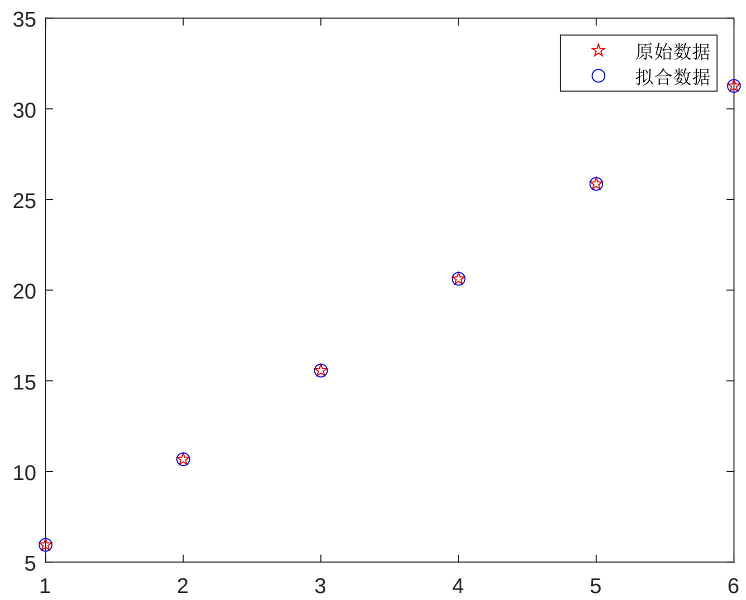

图 4.1: regress 拟合效果图

[2]摘自隋志军老师课件 W09

4.4.3 nlifit 函数

```
beta=nlinfit(x,y,fun,beta0)
% x,y用法与regress函数类似
% fun可为匿名函数或函数句柄
% beta0是回归系数初值,beta为估计出的回归系数
```

注意 非线性方程建议采用直接非线性最小二乘法拟合，fun 调用可参考3.1。

例题 4.2 用 nlifit 函数求解例题4.1：

解

```
function NonLinearFitting
x1=1:6;
x2=1.5:0.2:2.5;
y=[5.9512 10.6730 15.5676 20.6234 25.8599 31.2622];
beta0=[0 0 0 0];%对结果影响较大
% 关键函数
beta=nlinfit([x1',x2'],y',@fun,beta0) %与初值选取有关
% 验证结果
yi=fun(beta,[x1',x2']); %拟合函数在样本处点的函数值
plot(x2,y,'ro',x2',yi,'bp') %作图对比
errMax=max(abs(y'-yi)) %最大偏差为0.0034,与regress相同

%拟合函数部分
function y=fun(Beta,x)
a=Beta(1);b=Beta(2);c=Beta(3);d=Beta(4); %提高代码可读性
x1=x(:,1);x2=x(:,2); %与上一行本质上可以略去,不过下一行会比较复杂
y=a*x1+b*x2+c*x1.^2+d*x1.*x2;
```

4.4.4 csaps 函数

```
% Cubic smoothing spline 平滑三次样条拟合函数
% 拟合函数
pp=csaps(x,y,p,[],w)
% p: 平滑参数，范围[0,1],默认为1，即spline
% w: 误差权重，默认为1
% 拟合函数值
values=csaps(x,y,p,xx,w)
```

```
pp=csaps(x,y,p,[],w);
values=fnval(pp,xx)
% 函数fnval
fnval(f,x) %计算函数f在x处的函数值
```

注意 可以参考4.2.2中的相关介绍。

第 5 章 数值微分与积分

内容提要

- ❑ 数值微分的三种思路
- ❑ 插值型积分公式
- ❑ diff 与 fnder 函数
- ❑ quad 与 quadl 函数

5.1 数值微分

5.1.1 建立数值微分公式

定义 5.1. 三种思路

[差分]

从微分定义出发，通过近似处理，得到数值微分的近似公式。

[插值]

从插值近似公式出发，对插值公式的近似求导可得到数值微分的近似公式。

[拟合]

先用最小二乘拟合方法根据已知数据或得近似函数 (如样条函数)，再对此近似函数求微分可得到数值微分的近似公式。 ♣

5.1.2 向前差分及 diff 函数

在 MATLAB 中，可用 diff 求向量相邻元素的差值:

```
% 向前差分函数diff
% 调用形式
Y=diff(X,n,dim)
% X: 输入矩阵
% n: 差分阶数，默认1
% dim: 行1列2，默认按行差分
% Y=diff(X,2)等价于Y=diff(diff(X))
```

```
% 常用形式
Y=diff(X)
% 若X为m阶向量，则上式等价于
Y=[X(2)-X(1) X(3)-X(2)..X(m)-X(m-1)]
```

则可用一阶向前差分近似计算 $\frac{dx}{dy}$。

```
diff(y)./diff(x)
```

5.1.3 三次插值与 fnder 函数

注意 关于 spline/phchip 的函数介绍见4.2.2，不再赘述。

解决该类问题需要用到 fnder 函数，介绍如下:

```
% Differentiate function
% 调用格式
fprime=fnder(f,dorder) %dorder求导阶数，默认为1
```

```
% 常用形式
pp=spline(x,y); %pp为三次样条插值函数
fprime=fnder(pp); %fprime为pp的一阶导函数
value=fnval(fprime,x) %计算在x处的导数值
```

5.1.4 拟合与微分

注意 此节5.1.4隋志军老师于考试要点 W12 中未作要求，部分内容可参考拟合章节4.4。

```
% 多项式拟合求微分
p=ployfit(x,y,n); %多项式向量
dp=polyder(p); %多项式导函数
dpv=ployval(dp,x); %计算函数值，也可用fnval(dp,x)
```

```
% 样条拟合求微分
% 可用cspas/spaps/spap2
% 举cspas为例:
pp=csaps(x,y); %pp为样条拟合函数
fprime=fnder(pp); %fprime为pp的一阶导函数
value=fnval(fprime,x) %计算在x处的导数值
```

5.2 数值积分

5.2.1 基本思路与方法

基本思路来自于插值法，通过构造一个插值多项式 $Pn(x)$ 作为 $f(x)$ 的近似表达式，用 $Pn(x)$ 的积分值作为 $f(x)$ 的近似积分值。

定义 5.2. Newton-Cotes 求积公式

a) 等距节点，分为 n 份;

b) 使用 Lagrange 插值多项式近似。♣

特别地，改变其中等分数 n=1,2，可得：

1. **梯形求积公式;**
2. **Simpson 求积公式。**

定义 5.3. 复化求积

a) 将积分区间 $[a, b]$ 分为 n 个相等子区间;

b) 对每个子区间使用梯形求积或 Simpson 求积公式。♣

定义 5.4. 自适应求积

a) 考虑区间上及其二等分以后的两个 Simpson 积分和，分别记为 S1、S2;

b) S1、S2 满足精度要求则不再二等分，取该区间积分值为 S2。

[注]：可以不等步长。♣

注意 牛顿-科特斯和高斯-勒让德求积公式，从略。

5.2.2 quad 和 quadl 函数

自适应 Simpson 法数值积分：**quad**。

```
% 调用形式
q=quad(fun,a,b,tol,trace,p1,p2,…)
% fun 被积函数
% a,b 积分上下限
% tol 求解精度
```

```
% 常用形式
q=quad(fun,a,b)
```

自适应 Lobatto 数值积分：**quadl**，结果更精确。

```
% quadl函数与quad函数的使用完全一致
q=quadl(fun,a,b)
q=quadl(fun,a,b,tol,trace,p1,p2,…)
```

注意 被积函数支持向量化运算, 函数表达式中的必须使用点运算符号!

注意 fun 用法请参考3.1。

第 6 章 常微分方程数值解

内容提要

- 定义与分类
- 数值解方法
- MATLAB 中微分方程的表示方法
- ode45 函数

6.1 常微分方程

6.1.1 定义

含有未知函数导数的方程。

6.1.2 分类

分为**初值问题**与**边值问题**。

本节主要关注利用**步进法**解决**初值问题**。

6.1.3 初值问题的数值解方法

- 欧拉法：单步，矩形公式；
- 龙格－库塔法：单步，线性组合；
- 阿达姆斯法：多步。

6.2 MATLAB 解常微分方程

6.2.1 微分方程的表示

对于微分方程：

$$y' = 2x + y$$

```
% 可表示为
function dy=odefun(x,y)
dy=2*x+y

% 或
odefun=@(x,y) 2*x+y
```

对于微分方程组：

$$\begin{cases} y_1' = y_1 - y_2 + x \\ y_2' = y_1' + y_2 \end{cases}$$

```
% 可表示为
function dy=odefun(x,y)
dy1=y(1)-y(2)+x;
dy2=dy1+y(2);
dy=[dy1;dy2]; %方程组的dy用列向量输出!
```

对于高阶微分方程组：

$$\begin{cases} y_1' = xy_2' + y_1 \\ y_2'' = y_1' + sin(x)y_2 \end{cases}$$

```
% 变量代换
% 令y(1)=y1,y(2)=y2,y(3)=y2'
% 可表示为
function dy=odefun(x,y)
dy1=0;dy2=0;ddy2=0;
dy1=x*ddy2+y1;
dy2=y(3);
ddy2=dy1+sin(x)*y(2);
dy=[dy1;dy2;ddy2];
```

6.2.2 ode45 函数

```
% 常用形式
[T,Y]=ode45(fun,TSPAN,Y0)% TSPAN 求解区间; Y0 初值
% 带参数
[T,Y]=ode45(fun,TSPAN,Y0,options)
% 常用求解参数
%输出T与Y的关系图，类似plot(T,Y)
options=odeset( ‘outputfcn’ , ‘odeplot’ )
%输出求解变量Y之间的二维相平面图，类似plot(Y(2),Y(2))
options=odeset('outputfcn','odephas2')
```

例题 6.1 某串联反应各物质的浓度与反应时间 t 的关系如下[1]：

$$\begin{cases} \dfrac{dC_A}{dt} = -k_1 C_A \\ \dfrac{dC_B}{dt} = k_1 C_A - k_2 C_B \\ \dfrac{dC_c}{dt} = k_2 C_B - k_2 C_C \\ \dfrac{dC_D}{dt} = k_3 C_C - k_4 C_D \\ \dfrac{dC_E}{dt} = k_4 C_D \end{cases}$$

已知 k1=0.04，k2=0.05，k3=0.10，k4=0.08，反应开始时只有 A 存在，其浓度为 1。试编写一个 MATLAB 函数, 求前 60s 中每隔 5s 时各物质的浓度，并作图。

解

```
function solveC
T=0:5:60;
C0=[1 0 0 0 0];
% 设置图形输出
options=odeset('outputfcn','odeplot');
% 核心函数
[T C]=ode45(@odefun,T,C0,options);
% 常微分方程组
function dC=odefun(T,C)
k1=0.04;k2=0.05;k3=0.10;k4=0.08;
dCA=-k1*C(1);
dCB=k1*C(1)-k2*C(2);
dCC=k2*C(2)-k2*C(3);
dCD=k3*C(3)-k4*C(4);
dCE=k4*C(4);
dC=[dCA;dCB;dCC;dCD;dCE]; %按列
```

[1]改自隋志军老师“课程练习题”