第一章 编程及仿真

1.1 数据类型

常用的有下面几类。

(1) 数值：整型 (int)、双精度 (double)、单精度 (single)

(2) 逻辑：logical

(3) 字符串：char

(4) 符号对象：sym

(5) 结构体：structure

(6) 元胞数组：cell

1.1.1 数值型计算

Matlab 默认数值型：算术运算

运算	加	减	乘	幂乘	左除	右除
符号	+	−	*	^	/	\

其他常用的运算如下。

exp(x)	以 e 为底的指数	log(x)	以 e 为底的对数
log2(x)	以 2 为底的对数	log10(x)	以 10 为底的对数
sin(x)	正弦函数	cos(x)	余弦函数
asin(x)	反正弦函数	acos(x)	反余弦函数
tan(x)	正切函数	cot(x)	余切函数
atan(x)	反正切函数	acot(x)	反余切函数
sqrt(x)	平方根函数	abs(x)	绝对值或复数的幅值
real(x)	求 x 的实部	imag(x)	求 x 的虚部
angle(x)	x 的相位	abs(x)	复数 x 的幅度
ceil(x)	朝 $+\infty$ 方向取整	floor(x)	朝 $-\infty$ 方向取整
fix(x)	朝原点方向取整	round(x)	四舍五入到最接近的整数
lcm(m,n)	最小公倍数	gcd(m,n)	最大公约数
rem(m,n)	m 关于 n 求余	sign(x)	符号函数

1.1.2 变量及命名

系统定义的变量：

(1) ans：未指定输出参数时，运算的结果存储在变量 ans 中。
(2) pi：圆周率 π 的近似值。
(3) eps：无穷小的近似值：2.2204e−16。
(4) Inf：无穷大。
(5) NaN：Not-a-Number 的简写，表示计算结果不定，如 0/0。
(6) i, j：虚数单位。

变量名的命名规则：

(1) 变量包括系统变量以及自定义的变量。
(2) 自定义变量，即变量赋值语句用 “=”。
(3) 必须以字母开头，如 bookname, rank。
(4) 区分大小写，如 bookname, Bookname 是不同的变量。
(5) 可由字母、数字、下划线组成。
(6) 变量名尽量反映其含义，如汽车数量可用 numcar 或 num_car。
(7) 变量名不能与内置函数重复。
(8) 不能用中文。

1.1.3 语句的分隔符

1. 语句分隔符：分号或逗号。
2. 语句的末尾不使用分号时，系统会显示执行结果。

例题 1.1 a = 3;　　b = 4　　c = a^2.

1.2 常用的数据结构

主要有下面几类。

(1) 一维数组（向量）

(2) 二维数据（矩阵）

(3) 三维数据（张量）

1.2.1 一维数组的创建

1. 直接输入：a = [1 2 3 4 5 6]
2. 冒号生成：a = 初值: 步长: 终值
3. 线性等分生成：a = linspace(初值，终值，数据个数)

数和数组运算的比较：

运算	加	减	乘	幂乘	左除	右除
数	+	−	*	^	/	\
数组	+	−	.*	.^	./	.\

1.2.2 二维数组的创建

(1) 数据初始化：zeros, ones, eye

(2) 稀疏数组：spalloc, sparse, spones, speye, full

逻辑运算包含如下的几个。

运算	与	或	非
运算符	&	|	~

逻辑为真时，结果为 1，否则为 0。
关系运算包含如下的几个。

小于	小于等于	大于	大于等于	等于	不等于
<	<=	>	>=	==	~=

关系成立时，结果为 1，否则为 0。

1.3 字符和结构体数据

如下。

例题 1.2 a = rand(3); %a 为 double 型

syms x, y, y = 1 + x^2 %x, y 为 sym 类型

b = 'Li San'; %b 为 char 型

F.name = 'Li San', F.birth = 1999 %F 为 struct 型

c = struct('name', 'Li San', 'birth', 1999) %c 为 struct 型

为了让一个数组中的元素的类型可以互不相同，引入 cell 数组。

1.4 cell 数组

基本用法为：cell(m, n)。

例题 1.3 输入 A = cell(2, 3); A{1,1} = 'abc'; A{1,2} = rand(3); A{1,3} = cell(1,2);

例题 1.4 假设有 2 条公交线路，第 1 条公交线路站点编号依次为 101, 102, 103；第 2 条公交线路站点编号依次为 201, 202, 203, 204, 205。分别用 double 型数组和 cell 型数组存储这些信息。

double 型数组表述如下：

```
D = [101 102 103 0 0;          % 第 1 条公交线路信息
     201 202 203 204 205]      % 第 2 条公交线路信息
```

cell 型数组表述如下。

```
A = cell(2, 1);
A{1, 1} = [101 102 103];          % 第 1 条公交线路信息
A{2, 1} = [201 202 203 204 205];  % 第 2 条公交线路信息
```

1.5 常用的 MATLAB 函数

(1) 常用的一些函数:
查找：find，strfind
排序：sort
统计：max，min，sum，mean，median
(2) 符号计算工具箱：syms，subs
(3) 线性代数：
求秩：rank(A)
求行列式：det(A)
求逆矩阵：inv(A)
行初等变换：rref(A)
求齐次方程组的解：null(A)
求解方程组：fzero，fsolve
求特征值、特征向量：eig
矩阵的分解：SVD、QR
(4) 微积分
求极限：limit 函数
求不定积分：int(f, x)
求定积分：int(f, x, a, b)
求导数：diff(f, x), diff(f, x, n)
求解常微分模型：dsolve, ode23, ode45
(5) 数据类
求解插值问题：interp1，interp2
求解拟合问题：lsqcurvefit
主成分：pca
聚类 MATLAB 函数：cluster
分类 MATLAB 函数：svmtrain, svmclassify
支持向量机：
(6) 求解最优化模型
线性规划：linprog，intlinprog
非线性规划：fmincon
遗传算法：ga (求解不含等式约束的几乎所有优化模型)
(7) 求解回归分析
线性回归：regress
非线性最小二乘回归：nonlinfit
(8) 图形处理函数
读取与显示：imread，imshow

1.6 编程中的函数

函数的构成：函数名、参数、函数体、返回值。

函数的主要作用：

(1) 重用：

定义一次，调用多次。

(2) 抽象：

只要知道以下几点，就可以调用：函数的功能、函数名、参数、返回值。

(3) 封装：

将抽象的功能封装起来。

1.7 算法设计

算法要有输入和输出。一个算法必须有零个或以上输入量，以及一个或以上输出量，即算法计算的结果。算法的特征主要有以下几个。

明确性：算法的描述必须无歧义，以保证算法的实际执行结果是精确地符合要求或期望，通常要求实际运行结果是确定的。

有限性：依据图灵的定义，一个算法是能够被任何图灵完备系统模拟的一串运算，而图灵机器只有有限个状态、有限个输入符号和有限个转移函数（指令）。而一些定义更规定算法必须在有限个步骤内完成任务。

有效性：又称可行性，能够实现。算法中描述的操作都是可以通过已经实现的基本运算执行有限次来实现。

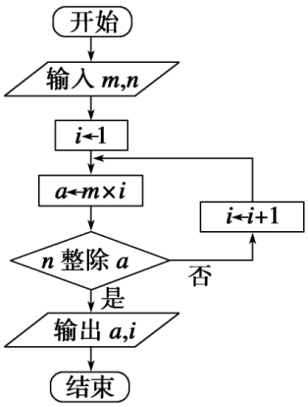
1.8 算法的描述

1. 自然语言

表 **3**　问题三优化问题算法

基于 II 层，IV 层最优厚度网格搜索法
步骤 1. 设定 II 层，IV 层厚度的厚度范围 $L_2 \in [0.6, 25]$,$L_4 \in [0.6, 6.4]$
步骤 2. 代入问题一中递推公式（32）计算皮肤外表层的温度变化数据 $T(x_{skin}, t)$
步骤 3. 判断与约束条件 $(x_s, t[30]]) \leq 47$℃、 $T\,(x_s, t[25]]) \leq 44$℃ 是否吻合
步骤 4. 将满足约束条件的 II 层，IV 层厚度提取出来，代入公式（51）得到最符合的厚度

2. 流程图



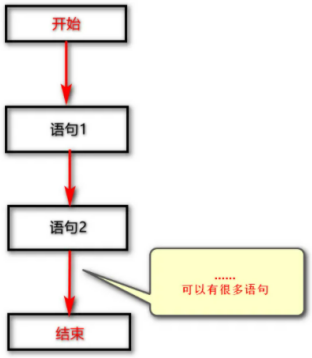
3. 伪代码

如用欧几里得算法求最大公约数：Euclid(m,n)，伪代码如下。

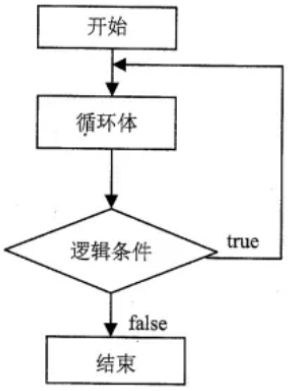
```
while n≠ 0
    do
        r ← m  mod n
        m ← n
        n ← r
    end
return m
```

1.9 语句的结构

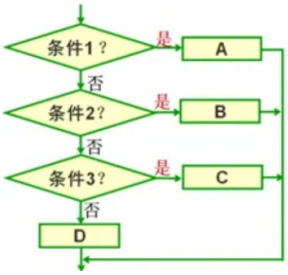
1. 顺序结构：按照语句的顺序自上而下，依次执行。



2. 循环结构：在程序中需要反复执行某个功能而设置的一种程序结构。常用的有 for、while 结构。



3. 分支结构：依据一定的条件选择执行路径，常用的有 if 结构。



1.10 Matlab 函数

包括如下的三类函数。

1. 自编函数 (function)
2. 内敛函数 (inline)
3. 匿名函数 (@)

1.10.1 函数语法

function [输出参数列表] = 函数名 (输入参数列表)

end

注意如下几个常见的问题。

(1) 文件名和函数名一致。

(2) 函数文件的位置和 Matlab 的路径一致。

(3) 函数名最好有一定的实际含义，如 Finonaccia，便于记忆。

(4) 函数名避免与系统内部函数名相同，如 sum、prod 等。

(5) 可通过 “help 函数名” 来检查该函数是否为系统函数。

例题 1.5 编写函数计算斐波那契数列中的元素，函数名命为 Fibonaccia.m，其中，

$$\begin{cases} f_1 = 1,\ f_2 = 1 \\ f_n = f_{n-1} + f_{n-2},\ n \geq 3 \end{cases}$$

```
function f = Fibonaccia(n)
f(1) = 1;
f(2) = 1;
for  i = 3:n
     f(i) = f(i−1) + f(i−2);    % 计算第 3 项至第 n 项
end
end
```

例题 1.6 编写一个计算底面半径为 r，高为 h 的圆柱体体积和表面积的函数，函数名命为 Cylinder_Volume_Surface.m。

```
function [V, S] = Cylinder_Volume_Surface(r, h)
V = pi*r^2*h;
S = 2*pi*r*h;
end
```

调用如下命令，即可求得 $r = 1, h = 2$ 时的体积和表面积：

```
>> [V, S] = Cylinder_Volume_Surface(1, 2)
```

1.11 inline 函数

inline 函数的表达式如下。

1. inline(' 表达式')：根据引号里的表达式建立内联函数，函数自变量符号根据表达式自动确定。
2. inline(' 表达式', ' 参数 1', ' 参数 2', ...)：定义函数时指定自变量符号。

如 f = inline('x∗y∗z', 'x', 'y', 'z')：指定参数为：x, y, z。

例题 1.7 创建 inine 函数计算

$$y = \sin x + \cos x + e^x \sin x$$

在数组 $-2\pi : 0.01 : 2\pi$ 中各点的函数值。

代码如下。

```
f = inline('sin(x)+cos(x)+exp(x).∗sin(x)')
x = −2∗pi:0.01:2∗pi;
y = f(x)
```

1.11.1 匿名函数

用法：@(参数列表) (函数表达式)

例如：>> f = @(x)(x^2)

例题 1.8 用匿名函数求 x^2 在 1：5 处的函数值。

输入命令：>> f = @(x)(x.^2)

再输入命令：

>> val = f(1:5)

运行结果为：val = 1 4 9 16 25

下面给出常用的几类算法：穷举法、直接搜索法、贪心法、Mento Carlo 法。

1.12 穷举法

对于解空间有限的情况下，如果计算量不算太大的情况下，可考虑使用穷举法求出问题的解。该方法通常解决的是离散问题，得到的是全局最优解。

算法思路如下。

```
1. 初始化操作：初始化当前较优的指标值 val，以及较优的解 sol
2. 主循环
for i=1：n                    % 假设有 n 个解
    new_val = fun(i)          % 计算出第 i 个解
    if new_val < val          % 比现有的属性值都更优
        val = new_val         % 将当前较优的值 val 更新为 new_val
        sol = new_sol         % 将当前较优的解 sol 更新为 new_sol
    end
end
```

例题 1.9 找出 $x^2 - 2y = 1(x, y \geq 1, y \leq 1000)$ 所有正整数解，并将所有正整数解以矩阵的形式保存。

思路 1 如下。

1. 已知 y 的取值范围：遍历 y。
2. 求出 x 的范围 x<45：遍历 x。
3. 判断关系 $x^2 - 2y = 1$ 是否成立。
4. 若为整数，存储该解：M=[M; x y]。

代码如下。

```
M =[  ];
for  x=1:45              % 遍历 x 的范围
    for y=1:1000          % 遍历 y 的范围
        if  x^2 – 2*y == 1    % 判断等式是否成立
            M = [M; x y];      % 保存找到的一个解
        end
    end
end
M
```

思路 2 如下。

1. 已知 y 的取值范围：遍历 y。
2. 将 y 代入方程，根据 x = sqrt(1+2*y) 计算出 x。
3. 判断 x 是否为整数：fix(x)==x。
4. 若为整数，存储该解：M=[M; x y]。

代码如下。

```
M =[  ];
y =1;
  while  y<=1000            % 遍历 y 的范围
     x = sqrt(1+2*y);
         if  fix(x)==x              % 判断 x 是否为整数
           M = [M; x y];          % 保存找到的一个解
         end
     y=y+1;
end
M
```

1.13 直接搜索法

简称为直接法，是指仅用计算函数值息来最优问题的一种无约束最优化方法。
例题：用直接搜索法求解最优化模型：

$$\min f(x_1, x_2)$$
$$\text{s.t.} \begin{cases} a_1 \le x_1 \le b_1 \\ a_2 \le x_2 \le b_2 \end{cases}$$

直接搜索算法如下。

```
1. 初始化步长参数 step1，step2
2. 主循环
for x1 = a1 : step1 : b1
    for x2 = a2 : step2 : b2
        new_val = fun(x1, x2)          % 计算目标函数值
          if new_val < val
              val = new_val             % 更新较优值
              sol = [x1, x2]            % 更新变量值
          end
    end
end
```

1.14 贪心算法

没有特定的算法框架，该算法与具体问题密切相关。

例题：（背包问题）已知 n 个物品质量和价值，在不能超过一定重量的条件下，要求所装物品的价值最大。

计算出每个物品单位质量价值，根据这个指标从大到小开始选择，直到不能再选择为止。计算的思路如下。

```
1. 初始化操作
2. 主循环
while 未到达结束条件
    利用贪心策略，求出当前这一步的决策
    更新状态变量的值
end
```

1.15 Monte Carlo 算法

早在 17 世纪，人们就已经知道用事件发生的 “频率” 作为事件的 “概率” 的近似值。从方法特征的角度来说，可以追溯到 18 世纪后半叶的蒲丰随机投针试验。即著名的蒲丰问题。

1777 年，古稀之年的蒲丰在家中请来好些客人玩投针游戏（针长是线距之半），他事先没有给客人讲与 π 有关的事。客人们虽然不知道主人的用意，但是都参加了游戏。他们共投针 2212 次，其中 704 次相交。蒲丰说，2212/704=3.142，这就是 π 值。这着实让人们惊喜不已。

20 世纪四十年代，由于电子计算机的出现，利用电子计算机可以实现大量的随机抽样的试验，使得用随机试验方法解决实际问题才有了可能。其中作为当时的代表性工作便是在第二次世界大战期间，为解决原子弹研制工作中，裂变物质的中子随机扩散问题，美国数学家冯 · 诺伊曼和乌拉姆等提出蒙特卡罗模拟方法。由于当时工作是保密的，就给这种方法起了一个代号叫蒙特卡罗，即摩纳哥的一个赌城的名字。用赌城的名字作为随机模拟的名称，既反映了该方法的部分内涵，又易记忆，因而很快就得到人们的普遍接受。

蒙特卡罗方法又称计算机随机模拟方法。它是以概率统计理论为基础的一种方法。其基本思想是：当所求问题的解是某个事件的概率，或者是某个随机变量的数学期望，或者是与概率、数学期望有关的量时，通过某种试验的方法，得出该事件发生的频率，或者该随机变量若干个具体观察值的算术平均值，通过它得到问题的解。

例题 1.10 设计算法对数 π 进行近似估计。

思路：四分之一的圆面积为：$S_1 = r^2\pi/4$。正方形的面积为：$S_2 = r^2$。记 k 和 n 分别为落在四分之一的圆面积和正方形的面积为的点数。由

$$\frac{S_1}{S_2} = \frac{r^2\pi/4}{r^2} = \frac{\pi}{4} = \frac{k}{n}$$

可得：

$$\pi = \frac{4k}{n}$$

求解的程序如下。

```
n = 1e6;
x = rand(n, 1);
y = rand(n, 1);
InCircle = (x.^2 + y.^2 <= 1);          % 判断点是否在圆内
k = sum(InCircle);          % 统计落在圆内的点数
pi = 4 * k/n;                % 计算 π 的近似值
```

例题 1.11 求解非线性整数规划

$$\max z = x_1^2 + x_2^2 + 3x_3^2 + 4x_4^2 + 2x_5^2 - 8x_1 - 2x_2 - 3x_3 - x_4 - 2x_5,$$

$$\text{s.t.} \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 \le 400, \\ x_1 + 2x_2 + 2x_3 + x_4 + 6x_5 \le 800, \\ 2x_1 + x_2 + 6x_3 \le 200, \\ x_3 + x_4 + 5x_5 \le 200, \\ 0 \le x_i \le 99, \quad i = 1, 2, \ldots, 5 \end{cases}$$

如果用显枚举法试探，共需计算 $(100)^5 = 10^{10}$ 个点，其计算量非常之大。然而应用蒙特卡洛去随机计算 10^6 个点，便可找到满意解。

估计可信度：不失一般性，假定一个整数规划的最优点不是孤立的奇点。假设取一个点的目标函数落在高值区的概率分别为 0.01，0.00001，则当计算 10^6 个点后，至少有一个点能落在高值区的概率分别为：

$$1 - 0.99^{1000000} \approx 0.99\cdots99(100\text{多位})$$

$$1 - 0.99999^{1000000} \approx 0.999954602$$

解：用蒙特卡洛法求得：

$$x_1 = 46, x_2 = 98, x_3 = 1, x_4 = 99$$

目标函数值为：

$$z = 50261$$

计算的程序如下。

```
clc
clear
p0=0; n=1e6;
tic;
for i = 1:n
    x = randi([0, 99], 1, 5);
   [f, g] = FunMonte(x);
    if all (g <= 0)
        if p0 < f
          x0 = x;  p0 = f;
        end
    end
end
x0, p0, toc



function [f, g] = FunMonte(x);
f = x(1)^2 + x(2)^2+3 * x(3)^2+4 * x(4)^2+2 * x(5)- 8 * x(1)-2 * x(2) -3 * x(3) - x(4)
    - 2 * x(5);
g = [sum(x)-400, x(1)+2 * x(2) + 2 * x(3) + x(4) + 6 * x(5) - 800, 2 * x(1) + x(2) + 6 * x(3) - 200,
    x(3) + x(4) + 5 * x(5) - 200];
end
```

说明：用 Monto Carlo 方法每次运算得到的结果可能不一样。

用蒙特卡罗法求解最优化模型的思路如下。

$$\min f(x_1, x_2)$$
$$\text{s.t.} \begin{cases} a_1 \le x_1 \le b_1 \\ a_2 \le x_2 \le b_2 \end{cases}$$

蒙特卡罗算法的求解思路如下。

```
1. 初始化函数值 val 及变量值 sol。
2. 主循环
for i=1 : N
    x1 = RND(a1, b1);
    x2 = RND(a2, b2);
    new_val = fun(x1, x2);          % 计算目标函数值
    if new_val < val
        val = new_val               % 更新较优值
        sol = [x1 x2]               % 更新较优解
    end
end
```

注：调用函数 RND 的表达式 RND(a,b) 表示返回区间 [a,b] 上随机数。

Monte Carlo 方法的优点如下。

1. 能够比较逼真地描述具有随机性质的事物的特点及物理实验过程。用蒙特卡罗方法解决实际问题，可以直接从实际问题本身出发，而不从方程或数学表达式出发。它有直观、形象的特点。
2. 受几何条件限制小。如在计算空间中的任一区域上的积分问题，无论区域的形状多么特殊，只要能给出描述该区域的几何特征的条件，就可以生成均匀的点进行仿真。
3. 收敛速度与问题的维数无关。维数的变化，只引起抽样时间及估计量计算时间的变化，不影响误差。
4. 程序结构简单，分块性强，易于实现。

Monte Carlo 方法的缺点如下。

1. 收敛速度慢，一般不容易得到精确度较高的近似结果。对于维数少（三维以下）的问题，不如其他方法好。
2. Monte Carlo 方法的误差是在一定置信水平下估计的，所以它的误差具有概率性，而不是一般意义下的误差。

蒙特卡罗方法有着广泛的应，如粒子输运问题、统计物理、典型数学问题、真空技术、激光技术以及医学、生物、探矿等。特别适用于在计算机上对大型项目、新产品项目和其他含有大量不确定因素的复杂决策系统进行风险模拟分析。