
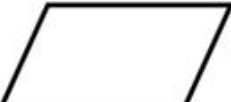

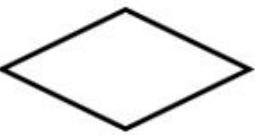


在必修3我们学习了算法的程序框图，在本章中，我们将继续学习利用流程图来刻画数学问题以及其他问题的解决过程。

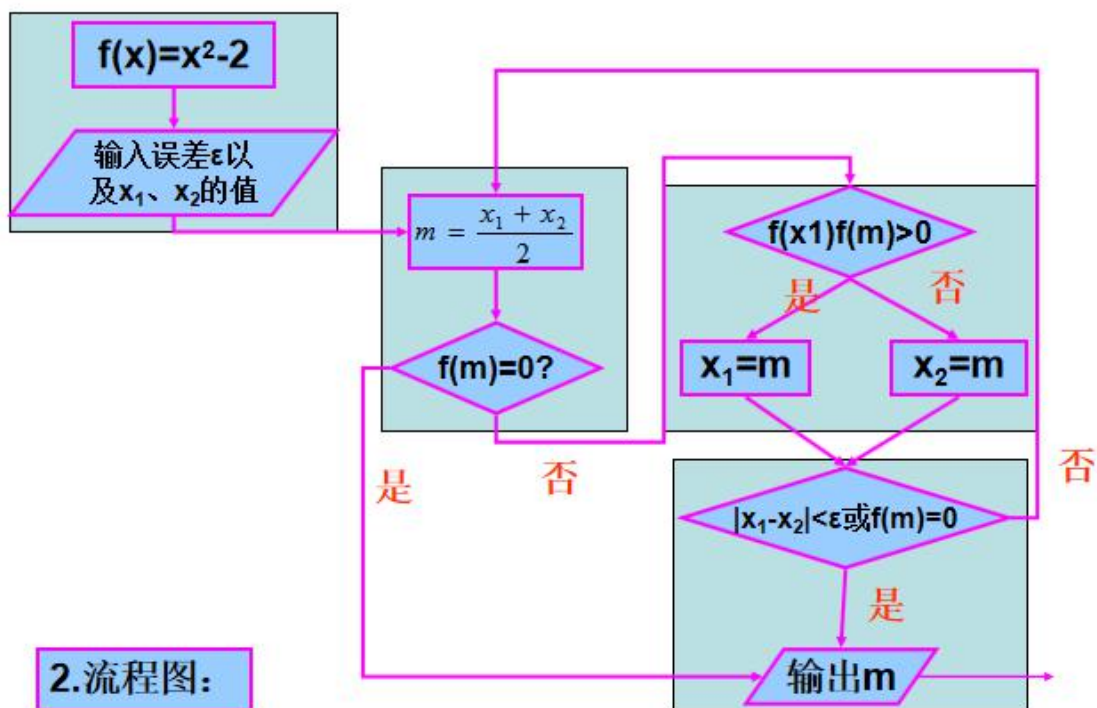
流程图事实上是将自然语言转化为算法的一种**过渡形式**，一般需要将每个算法的步骤分解为若干输入、输出、条件结构、循环结构等基本单元，再根据各个单元之间的**逻辑关系**，用流程线将它们连接起来，下面我们来用实例说明这个问题。

| 图形符号 | 名称 | 功能 |
|---|--------------|--|
|  | 终端框 (起止框) | 表示一个算法的起始和结束 |
|  | 输入、 输出框 | 表示一个算法输入和输出的信息 |
|  | 处理框 (执行框) | 赋值、计算 |
|  | 判断框 | 判断某一条件是否成立,成立时在出口处标明“是”或“Y”,不成立时标明“否”或“N”。 |

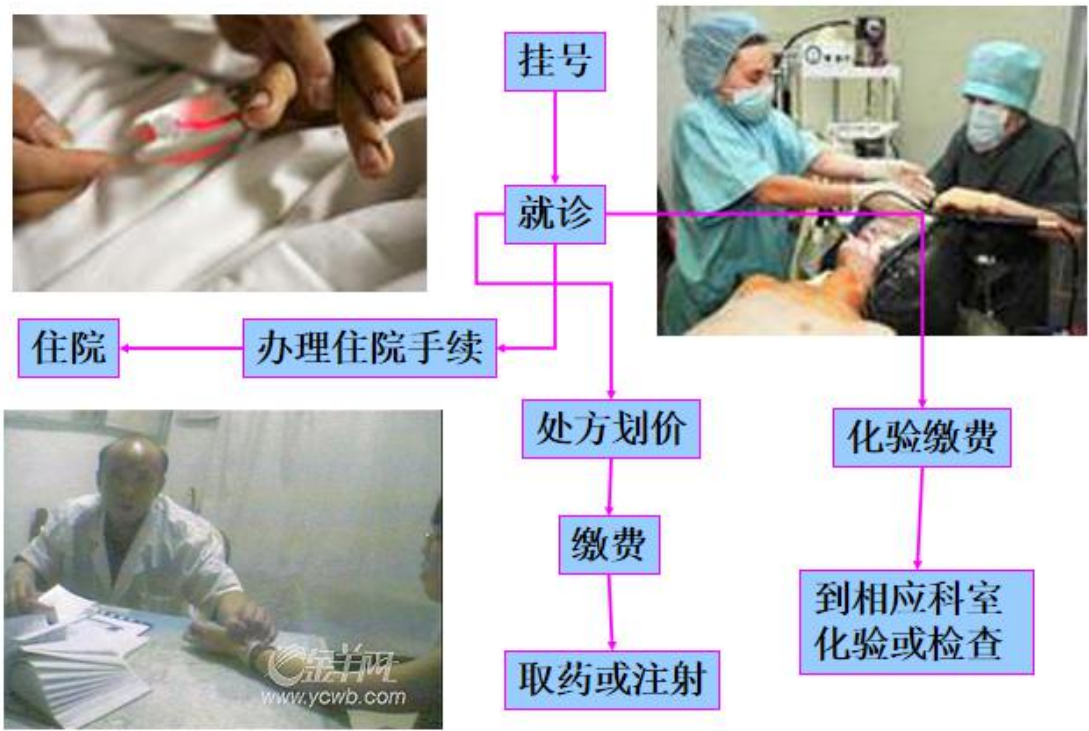
例1.画出利用二分法求方程 $x^2-2=0$ 的近似根的程序图框

1.自然语言:

- (1) 设置误差:令 $f(x)=x^2-2$, 误差 ϵ . $f(1)<0, f(2)>0$, 设 $x_1=1, x_2=2$
- (2) 平分区间:取区间中点 m , 判断 $f(m)$ 是否为0, $f(m)=0, x=m$, 否则, 判断 $f(m)f(x_1)$ 的符号
- (3) 判断新区间:若 $f(x_1)f(m)>0$, 则令 $x_1=m$, 否则令 $x_2=m$
- (4) 判断 $|x_1-x_2|<\epsilon$ 是否成立, 成立, 则 m 为方程的近似根, 否则, 返回(2)



其他的一些关于流程图的例子：



流程图的特点：可以**直观、明确**地表示某个算法或工序的**动态**的从开始到结束的全部过程

通常的流程图都是按照**时间顺序**来执行，从上到下、从左到右的顺序阅读的，所以流程图一般要求从左到右或从上到下的顺序来书写，程序图框有一定的规范和标准，而日常生活中的流程图相对要自由一些。

绘制流程图的一般过程：

- (1) 用**自然语言描述**流程步骤；
- (2) **分析**每一步骤是否可以**直接表达**，或需要**借助于逻辑结构**来表达（“**细化**”流程步骤）
- (3) 分析各步骤之间的关系；
- (4) 画出流程图表示整个流程。