

关于程序设计定型的 初步意見

(內部資料)

中国科学院計算技术研究所

1960年6月 北京

目 录

第一章 程序规划	(1)
§ 1 計算問題的提出	(1)
§ 2 計算問題的数学加工	(2)
§ 3 程序的分块	(3)
§ 4 存儲的分配	(5)
§ 5 計算正确性的檢查	(8)
§ 6 程序的灵活性	(12)
第二章 程序定型	(13)
§ 1 框图的符号及其使用	(13)
§ 2 邏輯图的符号及其使用	(15)
§ 3 靜态檢查	(20)
§ 4 調整程序	(20)
§ 5 导程序	(22)
§ 6 操作說明書之使用說明	(23)
操作說明書附表	(27)

第一章 程序规划

§ 1 計算問題的提出

为了便利工作进行，对提出的計算問題和提出計算問題的單位有以下要求：

（一）机器一般只对已形成的数学問題进行計算，而对形成数学問題以前的工作不在工作范围之內。

（二）提出問題的單位，应保證所提出問題在数学上和物理上的正确性，无不合理現象。

（三）提出必备的和参考的資料：

必备資料：正式公文和計算任务委托書。

正式公文：对委托任务的扼要說明。

資料的內容和数量。

要求完成時間。

机密程度。

固定联系人，通訊地址，联系手續。

計算任务書和資料：

問題的提出。

完整的数学描述。（要求系統一致，繕写清楚，注解准确。）

必备初始数据。

必要参考文献。

參考資料:

数学公式的原始形式。

一般解此問題所用的方法。

解的範圍，規律和可能提出解的正确性的檢查方法。

可能变化的一些参数和公式等。

(四) 計算要求的合理性: 根据計算問題的需要和机器实现的可能, 提出合理的計算工作量, 輸入, 輸出量和精确度等。

(五) 問題提出的單位和負責人, 应以高度負責精神, 保証計算公式和原始数据的正确, 无書写錯誤及考虑不周等現象。

(六) 为了減少錯誤和返工, 提出資料的負責人必須簽名盖章, 以示慎重。

(七) 計算任务說明書要修改或补充, 必須有正式公函及修改或补充說明書。

§ 2 計算問題的数学加工

用生产上提出的数学問題来直接編制程序, 常常是不适宜的, 甚至不能实现, 因此要求計算工作者对上述原始資料进行必要的数学加工。

(一) 加工的目的与要求:

1. 加工后的公式必須正确反映原来的計算問題。不影

响解的正确性和解的精确度。

2. 加工后公式要适于编制程序的要求:

- 1) 程序的邏輯簡單。
- 2) 节省工作量。(机器时间, 准备时间)
- 3) 节省存储。
- 4) 简化信息, 减少初始数据。

3. 加工后的公式必須:

符号統一, 角标清楚, 系統完备。

4. 注解清楚:

- 1) 变数范围。
- 2) 解的范围估计。
- 3) 封锁情况。(无意义之解不算)
- 4) 参数选取。
- 5) 数学和物理检查方法。
- 6) 特殊处理。

5. 公式与数据的绝对正确, 多次核对, 經手人签名。

6. 形成完整的計算資料, 当作下一步工作的直接依据。并作为計算任务書提出。

§ 3 程序的分块

一般計算問題可以分为若干部分, 对每一部分独立的編制程序, 即所謂程序分块。其实質, 就是将整个程序子程序化, 把复杂的結構化簡, 把大問題化小, 然后由控制程序把它們相互联系起来, 构成一个完整的工作程序。

(一) 程序分塊的優點:

1. 便于編制, 便于檢查。因為分塊程序結構簡單, 邏輯清楚。便于分塊調整, 將錯誤局部化。

2. 有靈活性, 便于修改和補充。由於程序塊的獨立性, 對某塊進行修改時不致影響全部程序的變動。

3. 便于發揮集體力量, 可以若干人同時平行的編制程序, 加速工作的進行。

4. 便于分配單元, 有助於從框圖或邏輯圖直接編寫地址程序。

5. 便于過渡到程序的標準化。

(二) 程序分塊的原則和依據:

1. 根據數學問題的特點來分塊。將相對獨立的數學公式, 邏輯控制, 或若干程序塊需要共用的算子, 分開獨立編制。

2. 要求每一個程序塊能完成一個獨立、明確的計算任務。一般程序塊可以自帶OAFP*等算子及專用的常數指令。

3. 要求邏輯結構簡單。易于編制以避免或減少錯誤。

4. 有固定的入口和出口, 一般為一個入口和一個出口。

5. 長短適宜, 實用問題的程序塊不宜過多, 免得零

* OAFP之意義見第二章 § 2。

碎，每个程序块也不宜太長。一般在32—128条指令之間。

6. 程序块应尽量灵活。每块末尾留些空白單元便于修改和补充。

(三) 程序块的联接法:

1. “自然銜接法” 每块之間通过条件、无条件轉移或順次相联自动轉向另一块。

2. “控制程序串联法”，用專門的控制程序把每块联接起来。块与块之間的轉移都通过控制程序。

前者适用于小块程序之間的联接，后者适用于較大程序块的联接。

§ 4 存儲的分配

存儲的分配是程序編制中的重要环节之一，分配合理与否，常常影响到編制程序的速度，解題需要的時間，甚至影响到程序能否正常的工作。因此在編制程序时必须作到：存儲分配合理，使用經濟，内外存交換正确。

(一) 内存的分配原則:

1. 能够确定的單元先分配。

2. 需要从外部設備輸入信息的單元集中在一起，并为輸出和檢查的方便，尽可能的将程序、数据、常数、工作單元等分別集中在一起。

3. 程序块之間留出一定的空白單元，便于修改和补

充，但空白不宜太多。

4. 为了檢查方便，程序中經常出現的單元最好給予明显的易于記憶的地址碼。

5. 为了代真地址方便，每程序块文字編碼K，尽可能給予便于計算的数据，如：0010，0030，……等。

6. 当内存較緊張时，必須严格考虑單元的节省，以尽量避免在計算过程中使用外存，而增加机器工作時間。但在内存不緊張时，有时特意多用一些工作單元，保留一些中間結果，便于进行檢查。

（二）外存的使用：

1. 有些問題需要計算的時間較長，为了不致因为机器中間发生錯誤需再从头开始計算，因此計算一定時間后，將内存的內容或重要的中間結果記入外存。

2. 有些問題初始数据太多或者程序过長，内存无法全部容納，可以将暫時不用的数据或沒有工作的程序記入外存。

3. 内外存交換需要的時間較長，因此内外存交換不宜过于頻繁。該应尽可能的將用到机会不多的数据和工作次数不多的程序块記入外存，这样可以減少内外存交換的次数。

4. 應該經濟的使用外存，如果只需要記入重要的中間結果，就可以保証当机器发生偶然故障不致再需从头开始計算，那么就不必將内存的全部內容記入外存。

(三) 内外存信息交换正确性措施:

既然在机器上工作难免不使用外存, 那么必须保证内外存交换的完全正确, 否则就无法再继续工作下去, 或者会得出不正确的计算结果。可以采用将记入外存的信息, 立即“假读”出来, 比较余数是否相等的办法来判断记入的正确与否。在读出时采用将余数与记入时的余数比较, 是否相等的办法来判断读出的正确性。有时可求出记入外存的信息的代码和, 读出时利用代码和进行比较。

为了工作起来方便, 减少错误的发生, 最好将内存和外存的地址编号一致, 比如将内存0125单元之内容记入外存某组某组0125单元, 将内存1325单元之内容记入外存0325单元等等。要是内存某一部分单元为许多程序块或数据所公用, 内外存无法一一对应起来。那么在程序纸上同时标出内存地址号码和外存地址号码。工作起来是很方便的。

(四) 内存分配的参考方案:

- 1.0005—0005 临时性工作单元,
- 0010——标准工作单元 (如与格网点对应的单元)
- 灵活性因素 (可变信息与参数等),
- 数据,
- 常数 (固定的),
- 程序块,
- 控制程序,

常数（不固定的），
 其他工作單元，
 1221—1355标准子程序占用單元
 2.0005—0005临时性工作單元
 0010——灵活性因素（可变信息与参数等）
 程序块，
 控制程序，
 标准工作單元，
 常数，
 原始数据，
 —1220其他工作單元，
 1221—1355标准子程序占用單元。

§ 5 計算正确性的檢查

在程序能正确实现計算方案时，但由于机器工作可能产生偶然性或系統性的錯誤，或是因为操作錯誤得出不正确的結果。因此应该有必要的措施，来保证計算的正确性。对它要求簡單可靠，迅速有效。介紹几种方法如下：

（一）数学檢查：

1. 依据数学問題的特点来验证結果的正确性。如：代数方程組求解，可将求得的解代入原方程以判断計算結果的正确性。又如求某数 x 的立方根，設其根为 y ：

即
$$\sqrt[3]{x} = y$$

那么可以利用

$$y^3 = x,$$

来验证计算的正确性。

2. 依据所采用的计算方法的特点来进行检查。例如解线性代数方程组

$$Ax = b.$$

若采用共轭斜量法求解，那么可利用两次相邻迭代的余量的正交性来检查。

令第 i 次迭代的余量为：

$$r_i = b - Ax_i$$

根据正交性有下面关系式

$$(r_i, r_{i+1}) = 0$$

假若采用高斯消去法求解：那么可以利用检查和进行验证。

3. 利用数学恒等式进行检查。如求三角函数可利用它们之间的恒等关系来检查。

如：

$$\sin^2 x + \cos^2 x = 1$$

4. 采用不同的计算方案进行计算，比较计算结果是否一致。此法的工作量较大，往往相当于重新编制程序，所以不是常用的方法。

对数学检查，由于计算过程的舍入或截断误差不同，往往不能用“=”来检查，只能看其差是否在误差允许范围之内。

（二）物理檢查：

根据其物理意义，如解的数值范围，解的变化规律等来检查。

（三）复算检查：

复算至少要多花一倍的机器工作时间，而且很难发现机器系统性的错误，因此若能用上面两种检查，就不宜用复算检查，也可以只对重要的计算过程进行复算。应该注意，不要只作了几个运算就复算，因为机器偶然故障，在瞬间不易改变过来，一般在2—3分钟内进行复算。

（四）对程序与数据求代码和检查：

在运算时间较长时，可每隔一定时间对程序或数据求一次代码和，与事先给好的检查数比较是否全等。使用此法时应注意要对程序的某一固定状态求代码和。

（五）调机检查（自带调机方案）

对于题目很大、参数很多、计算时间很长的問題，应该有适合本问题的检查机器的程序，这个检查程序应该与工作程序完全一致，它对某一组已知正确结果的参数进行计算，或者解高阶代数方程组时，可以让它解一个已知正确结果的低阶代数方程组。在开始工作时或工作一段时间后利用检查程序计算一遍将结果与正确结果比较，判断机器工作的正确性。

(六) 一定时期向外存記入中間結果或程序: 以便一旦發生錯誤, 不致于重新開始計算, 但必須保證記入的正確性, 記入外存不宜過於頻繁, 最好 5—10 分鐘記一次。

(七) 輸入輸出的檢查:

為了得到正確結果, 初始數據和印刷的結果均應保證其正確, 所以 $10 \rightarrow 2$, $2 \rightarrow 10^*$ 的正確性均應檢查。可採用重複兩遍比較其代碼和的辦法。又因打印機可能出故障, 因而對重要的結果應打印兩次。

用遞推過程求解的問題應保證每步均為正確, 一般可用數學檢查或復算方法進行檢查。用迭代方法求解的問題, 不必每步檢查, 因迭代過程本身有矯正錯誤的能力, 可用分析迭代誤差的變化情況來判斷其正確性。

在程序中還應考慮到各種意外情況的處理, 對某些可能出現的, 但物理上無意義的結果進行封鎖。對一些在數學上不合理的現象進行封鎖。

例如計算球體的半徑不應出現負數。求負數的平方根是沒有意義的。

* $10 \rightarrow 2$ 表十進制代碼譯為二進制代碼。

$2 \rightarrow 10$ 表二進制代碼譯為十進制代碼。

§ 6 程序的灵活性

(一) 程序灵活的必要性

便于计算公式的改变。

便于某些参数（步长，误差控制参数 ϵ ）或数据的选择。

便于一些意外情况的处理。

(二) 单元分配的灵活性

1. 尽量留出需要改变方案时可能要用到的工作单元。

例如：用尤拉法解 n 阶常微分方程初值问题改用龙格-库塔法时，需要增加 n 个标准单元。

2. 把改变可能性大的算子集中在一起，便于修改。

(三) 公式的灵活性

处理改变性大的公式时，可以将公式一般化，例如：某公式可能以一次也可能以二次形式出现，则按二次形式编程序。

(四) 利用信息使程序具有灵活性

可以使用信息来选择计算公式和控制逻辑转移。充分利用0001和0002来修改参数和控制逻辑转移。

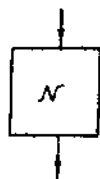
第二章 程序定型

§ 1 框图的符号及其使用

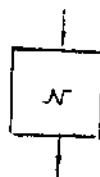
下面提出一种框图的符号供定型使用。



N 为算子编号，方框 N 的含意为：
一个算子；程序块或引用的一个标准程序。



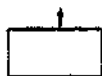
表示算子 N 中某条指令要用 ИЦУК 带第二地址返回的指令，转到算子 M 上去工作。



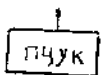
表示算子 N 中某条指令用 ИМУК 转到算子 M 上去工作，（由 ПЦУК 自动返回）



表示逻辑算子，它有二个出口，只实现一个逻辑条件，但可包含为了造比较常数的简单算术运算。



表为存放返回指令的单元（通常称为空白单元）



表自动返回指令



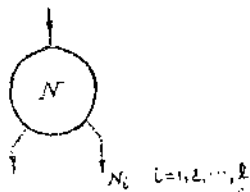
表自动返回指令



表31oct



表28oct



当遇到逻辑转移方向为可变时，可将可变的那个出口用夹点线---表示。

例 $N+1$ CK $N+2$ $l+1$ $N+2$

$N+2 < A_1 \quad A_2^* \quad A_3^*$

其中 $l+1$ 为修改地址的指令常数，而 $N+2$ 为逻辑条件， A_2^* 、 A_3^* 为可变地址，其 A_3^* 转移依赖于 A_2^* 之内容，这样 A_3^* 为可变出口，无固定方向，故用夹点线表示之。



表示算子 N 与所联接算子方向为可变的，有多个联接方向，故用夹点线表示之。

§ 2 逻辑图的符号及其使用

(一) 算术算子 A ，表示计算的数学公式，或数学公式的代表符号，在算术算子前面加一符号 $\lfloor_N$ 表示算子的编号为 N 。

(二) 逻辑算子：

1. 条件转移：

$P_N(a \sim b \lfloor_{N_2}^{N_1})$ a 与 b 比较成立转 N_1 不成立转 N_2 (如果 N_i 是 $N+1$ 算子，可省略不写)

$$P_N(x \overline{N} \lfloor_{N_1}^{N_1} \{a_1 b_1\}, \dots, \lfloor_{N_k}^{N_k} \{a_k b_k\})$$

若 $x \in \{a_i b_i\}$ 则转 N_i $i=1, \dots, k$

若 $x \in$ 所有 $\{a_i b_i\}$ 之补集，则转 $\overline{N}$ ，“ N ”逻辑算子本身的编号。

注：本章框图逻辑图所采用的符号主要是参考斯梅格列夫斯基著程序设计基础(计算所译已出版)和叶尔莫夫著快速电子计算机编制程序的程序(中译本已出版)二书，并结合我国 104 电子计算机指令系统的特点编制的。