# 以FORTRAN为基础的仿真语言

## ——CGPSS-F和CSIMAN程序设计

王 庆 编著

河南大学出版社

# 序　言

CGPSS-F 和 CSIMAN 语言是以 FORTRAN 为基础的仿真语言。

CGPSS-F（汉化的 General Purpose System Simulatin—FORTRAN）是一种专门用于离散事件系统的通用系统仿真语言。CGPSS-F 仿真语言实际上是以 FORTRAN 语言为基础的仿真子程序库。子程序库由 71 个子例程子程序和函数子程序（功能语句块）组成。使用时，用户只需根据仿真问题的要求，在主程序模型中调用相应的子程序，并填入给定的相应参数即可。

由于 CGPSS-F 全部是用 FORTRAN 语言编成的，其本质上仍是 FORTRAN 语言，所以它与 FORTRAN 77 兼容，因此，只要配有通用的 FORTRAN 77 系统软件的内存容量较大的微型或小型计算机即可运行 CGPSS-F，原则上不受机型限制。

CGPSS-F 仿真主程序通常由 12 个固定程序段组成，其中除用户自定义部分、用户设定值及模型段需要用户自编外，其余各段内容基本不变，子程序库更是预先已准备好了的。因此，只要仿真目的明确，模型流程清楚，CGPSS-F 的仿真系统模型就很容易建立起来。并且，时间流程安排由子程序库中的专用流程管理子程序（ACTIV1 和 ACTIV2）自动完成，不须用户安排，使用起来很方便。同时，由于 CGPSS-F 不需要专用文本，不受机型限制，便于移植，故极易推广应用。

CSIMAN 语言也是以 FORTRAN 为基础的一种仿真语言。CSIMAN 可对离散的、连续的、离散-连续混合的各种系统进行制模、仿真和结果分析。它是一种功能很强的新型混合仿真分析语言。CSIMAN 语言是近年来出现的一种新型的多功能计算机仿真分析语言，对于离散变化系统，它可用面向过程的方法来描述模型；对于连续变化系统，它可用代数差分/微分方程建立模型。因此，它可对各种离散变化系统、连续变化系统和离散-连续混合变化系统进行仿真。

CSIMAN 语言采用全新的软件结构设计而成，比 CGPSS-F 语言具有更多的功能。它的仿真方法具有物理过程逼真、适用范围广泛、仿真源程序简单明了、易读性好，代表了当代仿真语言的最新水平。CSIMAN 语言具有新颖的软件结构特点，使得系统模型建立、实验架确定、仿真运行及输出数据分析，都可以分开来独立进行。这使得各阶段所需计算机内存容量明显减少，从而使复杂的 CSIMAN 仿真能在具有 640KB 内存的 IBM XT、长城 0520 计算机、华光计算机等微型计算机上运行，比 CGPSS-F 语言对计算机内存容量的要求更宽松。

由于 CSIMAN 这种新的软件结构允许它运行在微型计算机、小型计算机以及大中型

计算机上，这种对机器的广泛适应性，使得CSIMAN语言易于推广应用。

因为CSIMAN语言的模型架和实验架是分开的，这种新的软件结构特点，允许在运行时确定问题的大小，只在实验架中指定实体、属性、文件、动态变量等的数量或变化范围，而不用对模型部分的源程序进行重新编制。也就是说，在不同开发阶段和仿真研究过程中，可以实现局部中断，使一种模型与多种实验条件匹配，一种数据可用多种数据分析方式进行分析。反之，使多种模型与一种实验条件匹配，多种数据以同样数据分析方式进行分析以便于比较。此外，CSIMAN软件系统具有很强的数据处理和统计分析能力。CSIMAN拥有10个随机数流（整型随机发生器），加上用户自定义的离散、连续概率分布，可为用户提供15种满足不同概率分布的随机变量发生器。

在输出数据分析方面，CSIMAN借助于数据处理命令语句，提供了丰富的数据处理和图形显示功能。

本书分为十二章：

第一章到第五章　CGPSS-F仿真语言的介绍：

第一章　CGPSS-F系统结构，重点是主程序的构成；

第二章　CGPSS-F主程序中固定调用的功能语句块；

第三章　CGPSS-F中有关实体活动的功能语句块；

第四章　CGPSS-F中有关站的功能语句块；

第五章　CGPSS-F中数据生成、收集统计与打印语句块，并同时介绍了CGPSS-F仿真系统程序设计实例。

第六章到第十二章　CSIMAN仿真语言的介绍：

第六章　CSIMAN语言的概述和CSIMAN系统仿真；

第七、八、九章　CSIMAN离散变化系统的仿真。这三章分别叙述用于仿真一般系统的初级、中级和高级的模型语句块和实验架语句元的基本概念。并在第七章中讨论了利用COUTPT输出处理程序进行输出数据的分析处理方法；

第十章　用于仿真物资搬运系统的模型专用语句块和实验架专用语句元；

第十一章　仿真连续系统的方法；

第十二章　仿真离散-连续混合系统的方法。

本书强调在一般仿真系统中对实际问题的解答技能，书中提供了18个完整的仿真程序设计实例，对仿真源程序中的主要功能语句块和语句元进行比较详细地解释，特别是在CSIMAN模型架中，每一个语句块的后面都加了汉字注释，使程序力求易读，便于学习。

本书可供高等学校有关计算机应用专业、自动控制专业的教师、研究生、本科生作为计算机仿真语言课程的教材，也可供有关专业的科研人员、计算机软件人员、工程技术人员、管理人员使用和参考。

编　者

1994年5月于郑州大学

# 目　录


第一章　CGPSS-F 系统结构 ……（1）
§1.1　CGPSS-F 系统主程序模型 ……（1）
§1.2　CGPSS-F 系统子程序库 ……（11）
§1.3　CGPSS-F 系统的运行 ……（12）
第二章　CGPSS-F 主程序固定调用语句块 ……（14）
§2.1　清除数据区语句块 ……（14）
§2.2　初始化功能语句块 ……（15）
§2.3　流程管理、事件安排和仿真停续判定语句块 ……（17）
第三章　CGPSS-F 有关实体活动的数组和语句块 ……（21）
§3.1　实体活动数组 ……（21）
§3.2　实体排队等待状态数组 ……（22）
§3.3　实体活动语句块 ……（22）
第四章　CGPSS-F 有关站的语句块及站的排队策略 ……（26）
§4.1　单设备、多设备及多设备使用原则 ……（26）
§4.2　地址存储器，非地址存储器与存储器存放策略 ……（36）
§4.3　同族实体收集站和普通实体收集站 ……（41）
§4.4　1 型门和 2 型门 ……（43）
§4.5　用户链和触发站 ……（44）
§4.6　站的排队策略 ……（46）
§4.7　站前等待实体重设优先级 ……（47）
第五章　CGPSS-F 数据的生成、收集统计与打印语句块 ……（49）
§5.1　随机数发生器语句块 ……（49）
§5.2　数据收集统计语句块 ……（52）
§5.3　频率表生成与打印语句块 ……（54）
§5.4　数据打印语句块 ……（55）
§5.5　CGPSS-F 仿真系统程序设计实例 ……（57）
仿真系统程序设计实例(一)：机械零件加工的随机系统 ……（58）
仿真系统程序设计实例(二)：医院管理系统 ……（70）
第六章　CSIMAN 系统仿真 ……（84）
§6.1　系统-理论结构 ……（84）
§6.2　建模面向方式 ……（84）
§6.3　软件结构 ……（85）
§6.4　系统模型(模型架) ……（87）

§6.5 实验框架(实验架) …… (90)
第七章 CSIMAN 初级语句块 …… (93)
§7.1 实体、变量、属性和函数 …… (93)
§7.2 语句块操作数 …… (96)
§7.3 初级功能语句块 …… (98)
§7.4 实验架 …… (104)
§7.5 CSIMAN 的输入,连接及运行 …… (115)
§7.6 CSIMAN 输出结果分析 …… (118)
§7.7 离散仿真系统程序设计实例 …… (128)
仿真系统程序设计实例(三):银行系统 …… (129)
仿真系统程序设计实例(四):在一组机床上加工工件 …… (133)
仿真系统程序设计实例(五):玻璃箱封装作业系统 …… (138)
仿真系统程序设计实例(六):餐厅模型 …… (144)
第八章 CSIMAN 中级语句块 …… (149)
§8.1 模型架 …… (149)
§8.2 实验架 …… (163)
§8.3 离散仿真系统程序设计实例 …… (168)
仿真系统程序设计实例(七):电视机检查与调整系统 …… (168)
仿真系统程序设计实例(八):急诊室轮换值班系统 …… (172)
仿真系统程序设计实例(九):钻床作业系统 …… (177)
第九章 CSIMAN 高级语句块 …… (181)
§9.1 模型架 …… (181)
§9.2 实验架 …… (188)
§9.3 离散仿真系统程序设计实例 …… (190)
仿真系统程序设计实例(十):长途电话亭 …… (190)
仿真系统程序设计实例(十一):结帐台 …… (193)
仿真系统程序设计实例(十二):车间生产线 …… (195)
仿真系统程序设计实例(十三):加工车间模型 …… (199)
第十章 模拟货物搬运系统 …… (203)
§10.1 运送器模型架的功能语句块 …… (203)
§10.2 传送器模型架的功能语句块 …… (209)
§10.3 运送器和传送器实验架的功能语句元 …… (213)
§10.4 离散仿真系统程序设计实例 …… (217)
仿真系统程序设计实例(十四):自动加工中心 …… (217)
仿真系统程序设计实例(十五):印刷电路板组装系统 …… (221)
仿真系统程序设计实例(十六):胶合板厂木材切削系统 …… (227)
第十一章 CSIMAN 连续模型 …… (232)
§11.1 连续模型 …… (232)

第十二章　复合模型…… (241)
§12.1　离散成分引起连续成分的变化 …… (241)
§12.2　连续成分引起离散成分的变化 …… (241)
§12.3　执行复合模型 …… (244)
§12.4　复合系统模型程序设计实例 …… (245)
仿真系统程序设计实例(十八):均热炉 …… (245)
附录　CGPSS-F 仿真系统子程序总库 …… (250)
功能语句块与功能语句元索引…… (340)
参考文献…… (343)

# 第一章　CGPSS-F 系统结构

CGPSS-F 仿真系统，包括一个结构基本相同的、通常由 12 个程序段组成的主程序模型和一个用子例程子程序和函数子程序组成的子程序库。现分别加以讨论。

## §1.1　CGPSS-F 系统主程序模型

CGPSS-F 为用户提供了一个通常由 12 个程序段组成的主程序模型框架，其中除三部分，即用户自定义部分、用户设定值部分和第 9 段——模型段部分需要用户自己填写以外，其他各段基本不变。也就是说，用户在每次编写仿真程序时，只需在主程序模型中，填入一些所需的变量和参数说明等由用户自定义和设定的内容，以及在模型段中填入能反映系统模型的一串子程序调用语句，其他部分基本不用改变。每段都有固定任务的 12 个程序段，用户依照规定的格式填写。

图 1-1 是一个机械零件加工的确定性仿真系统的主程序模型。我们通过这个具体的主程序模型来分析 CGPSS-F 仿真系统主程序模型框架的组成结构，以便于掌握和理解。

机械零件加工的确定性仿真系统描述如下：

一个机械零件加工车间，零件到达的时间间隔为 5 分钟，机床加工时间为 6 分钟，排队原则为“先进先出”。若零件从 8 点开始传送，10 点半停止。试求：

(1)全部零件加工完毕的时刻；

(2)全部零件加工完毕共用了多少时间；

(3)零件平均等待时间。

```
C       确定性的机械零件加工系统
        PROGRAM MAIN
C       1. 变量类型说明、数组定维及 COMMON 语句
        IMPLICIT INTEGER(A-Z)
        REAL BINSTA
        DOUBLE PRECISION DRN,DFACT,DMODUL,DCONST
C
        REAL MWT,EWT
C
        DIMENSION TX(200,30)
        COMMON AL(200,2),LAL,EL(30,2),
     $LEL,LEV,LFAM,NADDR,STATE(6122),OK
        COMMON N,T,IT,RT
```

```
      COMMON /ASM/ASM(200,10)
      COMMON /BIN/BIN(50,8),BINSTA(50,2)
      COMMON/DRN/DRN(30),DFACT(30),DMODUL,
     $  DCONST(30)
      COMMON /FAC/FAC(20,3)
      COMMON /FAM/FAM(200,3)
      COMMON /GA1/GATHF(200,10)
      COMMON /GA2/GATHT(10)
      COMMON /MFA/LSE,MFAC(2,2),MBV(2),SE(20,3)
      COMMON /PLA/PLAMA(2,2)
      COMMON /POL/POLVEC(200),POLC,POL(20,2)
      COMMON /SBV/LSM,SBV(20),SM(1024,2)
      COMMON /SRC/SRC(20,2)
      COMMON NTXC
      COMMON /STO/STO(20,2)
      COMMON /STR/STRAMA(20,2)
      COMMON /UC1/UCHF(200,10,2)
      COMMON /UC2/UCHT(10,2)
      WRITE(*,101)
101   FORMAT(1X,'读入 iprint')
2000  read(*,*) iprint
      write(*,701) iprint
701   format(1x,'iprint=',i2)
C
C
C     2. 用户自定义函数
C
C
C     3. 调用 RESET 块,清除数据区
3000  CONTINUE
      CALL RESET (TX,LTX,iprint)
C
      TWT=0
      MWT=0
C
C
C     4. 调用 INIT1,INIT2 及 INIT3 块,进行初始化
C
4000  CONTINUE
C
C
      CALL INIT1(iprint)
```

```
      CALL INIT2( * 9999,iprint)
      CALL INIT3( * 9999,iprint)
C
C     5. 读入仿真时间 N、实体极限数 ZTX、
C     控制变量 CONTIN,SECURE 及 IPRINT
C
C
5000  CONTINUE
      WRIET( * ,5001)
5001  FORMAT('1')
C
      WRITE( * ,102)
102   FORMAT( * ,'读入仿真时间 N')
      READ( * ,5002)N
5002  FORMAT(I10)
      WRITE( * ,5003)N
5003  FORMAT(1X,'N',8X,I10/)
      IF(N. LE. 0) GOTO 5097
      WRITE( * ,103)
103   FORMAT(1X,'读入实体极限数 ZTX')
      READ( * ,5002)ZTX
      WRITE( * ,5004)ZTX
5004  FORMAT(1X,'ZTX',6X,I10/)
      IF(ZTX · LE · 0)GOTO 5097
      WRITE( * ,104)
104   FORMAT(1X,'读入控制变量 CONTIN')
      READ( * ,5002) CONTIN
      WRITE( * ,5005) CONTIN
5005  FORMAT(1X,'CONTIN',3X,I10/)
      WRITE( * ,105)
105   FORMAT(1X,'读入控制变量 SECURE')
      READ( * ,5002) SECURE
      WRITE( * ,5006) SECURE
5006  FORMAT(1X,'SECURE',3X,I10/)
      WRITE( * ,106)
106   FORMAT(1X,'读入 IPRINT')
      READ( * ,5002) IPRINT
      WRITE( * ,5007) IPRINT
5007  FORMAT(1X,'PRINT',4X,I10/)
      GOTO 5500
5097  WRITE( * ,5098)
5098  FORMAT('O',' * * * BAD INPUT * * * ')
```

```
      GOTO 9999
C
C
5500  CONTINUE
C
C
      IF (IPRINT.NE.0) WRITE(*,5001)
      IF (CONTIN.EQ.0) GOTO 6000
      CALL CONT(TX,LTX,iprint)
      GOTO 1001
C
C
C     6. 调用 EVENT 块,安排首次事件
C
C
6000  CONTINUE
      CALL EVENT(480,1,1,TX,LTX,*1006,IPRINT)
      CALL EVENT(630,4,0,TX,LTX,*1006,IPRINT)
C
C
C     7. 流程管理 1,调用 ACTIV1 块
C
1001  CALL ACTIV1 (TX,LTX,*1006,iprint)
      LT=T
      NT=RT
C
C
C     8. 目标选择器
C
C
1003  CONTINUE
C
C
      if(iprint.eq.1) write(*,702) naddr
702   format(1x,'naddr=',i2)
C
      GOTO(1,2,3,4),NADDR
C
C
C     9. 模型段
C
C
```

```
C       生成实体
1       CALL GENERA(5,ZTX,0,1,TX,LTX,*1006,IPRINT)
C
C       到达和空出单设备
C
2       CALL SEIZE(1,2,TX,LTX,*1005,IPRINT)
        TWT=TWT+(T-TX(LTX,3))
3       CALL WORK(6,1,3,0,TX,LTX, *1005,*1006,IPRINT)
        CALL CLEAR(1,TX,LTX,*1005,*1006,IPRINT)
C
C       消除实体
C
        CALL TERMIN(TX,LTX,*1005,IPRINT)
C
C       打印出系统状态
C
4       WRITE(*,5001)
        CALL REPORT(0,20,0,0,30,200,200,0,TX,LTX,iprint)
C       10. 流程管理 2,调用 ACTIV2 块
C
C
1005    CONTINUE
        CALL ACTIV2(TX,LTX,*1001,*1003,iprint)
C
C
C       11. 调用 ENDBIN 块,进行最后结果分析
C
1006    CONTINUE
        T=LT
        RT=NT
        CALL   ENDBIN(TX,LTX,iprint)
C
C
        EWT=FLOAT(LT/60)+FLOAT(MOD(LT,60))/100
        MWT=FLOAT(TWT)/FLOAT(NTXC)
C
C
C       12. 打印结果
C
C
8000    CONTINUE
        WRITE(*,5001)
```

```
      CALL REPORT(0,20,0,0,30,200,200,0,TX,LTX,iprint)
      IF(SECURE.EQ.1) CALL SAVE (TX,LTX,iprint)
C
C
      WRITE(*,8010)
8010  FORMAT(/////21X,40('*'),2(/21X,'*',38X,'*'))
      WRITE(*,8020) EWT
8020  FORMAT(21X,'*',2X,'全部零件加工完毕的时刻:',F6.2,'小时.','*')
      WRITE(*,8030) MWT
8030  FORMAT(21X,'*',1X,'零件平均等待时间:',7X,F6.2,'分钟.','*')
      WRITE(*,8040)
8040  FORMAT(2(21X,'*',38X,'*'/),21X,40('*'))
      WRITE(*,5001)
C
      WRITE(*,107)
107   FORMAT(1X,'读入 INPUT')
C
      READ(*,8992) INPUT
8992  FORMAT(I1)
      IF(INPUT.EQ.1) GOTO 2000
C
9999  STOP
      END
```

图 1-1 机械零件加工的 CGPSS-F 确定性仿真系统主程序模型

下面就根据这个实际的主程序模型来讲述各程序段的结构和作用。

1. 变量类型说明、数组定维及 COMMON 语句

这一段分两部分，第一部分是变量类型说明部分。整型变量通常都用整型隐含类型说明语句：

IMPLICIT INTEGER(A－Z)

来说明，也就是说，如果在下面不用其他变量类型语句重新定义的话，所有变量均定义为整型。当某些变量为实型时，需重新定义。例如，BINSTA 为实型，则用实型变量说明语句：

REAL BINSTA

重新定义。

CGPSS-F 中所用的(0,1)分布均匀随机数发生器：

FUNCTION RN(RNUM)

生成的(0,1)分布均匀随机数是采用混合同余法计算的，其具体算式中涉及到的四个变量：

DRN,DFACT,DMODUL,DSONST

要求使用双精度数型，故须重新用双精度类型说明语句：

DOUBLE PRECISION DRN,DFACT,DMODUL,DCONST

来加以定义。而随机数发生器编号 RNUM 可取 1～30 的正整数，已在前面由整型隐含说明语句说明了。如果使用逻辑变量，还须用逻辑变量定义符：LOGICAL 重新定义。简而言之，重新定义就是将其从整型变量中划分出来。

第 1 段的第 2 部分是数组定维和公用语句 COMMON。数组定维，通常由 COMMON 语句来实现，例如

COMMON，AL（200，2），LAL，EL（30，2），LEL，LEV，LFAM，NADDR，STATE（6122），OK

定义活动数组 AL(200,2)为具有 200 行，2 列的 200×2 个数组元素的数组；

定义事件数组 EL(30,2)为具有 30 行，2 列的 30×2 个数组元素的数组。

COMMON 语句除了完成数组定维的任务以外，还有一个重要的任务就是在主程序与子程序之间，传递变量数据。为了实现这一任务，必须在主程序模型的开头可执行语句之前和子程序的开头可执行语句之前，通过 COMMON 语句设置变量数据公用区。如果在主程序模型中设置了 COMMON，AL(200,2)，LAL，EL(30,2)，...语句，而在相应的子程序中没有设置这一语句，活动数组 AL(200,2)，活动数组行号 LAL，和事件数组 EL(30,2)的数据就不能在主程序与相应的子程序之间进行传递。

如果某一数组的数据通过 COMMON 语句在主程序与子程序，或子程序与子程序之间进行传递不堪奏效，可以采用变量“虚实结合”的途径来传递。这时，需要采用 DIMENSION语句来为数组定维。例如，实体数组 TX(200,30)就是通过

DIMENSION TX(200,30)

维数定义语句来定维的。

在这一段的第 1 部分中，数组定维与数据传递均可采用两种方式。但应注意，用 DIMENSION定义的数组，其数据须通过子程序参数“虚实结合”的途径传递，而不能通过 COMMON 语句设置的公用区传递，否则将出现数组重复定义的错误。

如果数据传递奏效，当然是数组定维任务与数据传递任务由 COMMON 语句一并完成最为理想了。究竟采取何种方式，在程序调试过程中可以调整。

2. 用户自定义函数

若没有用户自定义语句函数，这第 2 段可以空白，不须填写。

3. 调用 RESET 块，清除数据区

在 CGPSS-F 仿真系统子程序库中，有一个专用的清除所有数据区的子程序，由于在子程序库中的每一个子程序均执行一个特定功能，因此，以后我们将称子程序为“功能语句块”。

在主程序中，使用

CALL RESET(TX,LTX,IPRINT)

调用语句，调用“所有数据区清除语句块——RESET 块”，就可以将 CGPSS-F 系统的变量和数组的所有直属数据区清除。

即令：

SR(20,2),EL(30,2),AL(30,2),TX(200,30),FAC(20,3),SE(20,3),STO(20,2),

SM(1024,2),BIN(50,7),BINSTA(50,2),FAM(200,3),POL(20,2)等均赋零值。

将 LEL,LAL,STATE(6122),BIN(50,8)等均赋 1。

假如用户尚有自定义的数据区,也应在第 3 段自行清除,为仿真作好准备。

关于这些数组的意义和作用,我们将在以后各章中根据讲述内容的需要,逐步加以介绍。

4. 调用 INIT1,INIT2 及 INIT3 块,进行初始化

在第 4 段中,调用三个初始化语句块。

调用第一个初始化语句块——"均匀随机数发生器初始化语句块——INIT1 块",将 0 到 1 的均匀随机数发生器初始化,即为均匀随机数发生器设置初始值,如令:

```
DFACT(1)=10753813
DFACT(2)=228181237
.....
DFACT(30)=10767581
DMODUL=2. * *30
DCONST(1~30)=227623267
DRN(1~30)=1
```

调用第二个初始化语句块——"多设备数据区设置语句块——INIT2 块",为多设备设置数据区,如令:

多设备数组 SE(20,3)的行号 LSE=1,再由多设备状态数组 MFAC(2,2)的值(忙状态为 0,闲状态为 1)来计算多设备数组 SE(20,3)的行号 LSE 值。即

```
      LSE=1
      DO  100  I=1,2
          IF(MFAC(I,2).EQ.0) GO TO 100
          MBV(I)=LSE
          LSE=LSE+MFAC(I,2)
          IF(LSE-1.GT.20) GO TO 200
100   CONTINUE
      RETURN
2000  WRITE(*,3000) I
      .....
```

其中,MBV(I)为行标志。

调用第三个初始化语句块——"地址存储器数据区设置语句块——INIT3 块",为地址存储器设置数据区。即依存储器容量 STO(I,2)和 LSM=LSM+STO(I,2)为地址存储器数组 SM(LSM,1)和 SM(LSM,2)设置行号 LSM 的初始值。并依

```
SM(LSM,1)=STO(I,2)
SM(LSM,2)=-1
```

为地址存储器长度(单元数)SM(LSM,1)和地址存储器状态码 SM(LSM,2)设置初始值。

在第 4 段中,同时还要设置排队原则、存储策略、进入和离开服务站的原则等等。

5. 读入仿真时间 N、实体极限数 ZTX、控制变量 CONTIN、SECURE 及 IPRINT

在第 5 段，用户根据仿真运行的实际需要，将仿真时间上限 N，调用“实体生成语句块——GENERA 块”时产生实体数的极限参数值 ZTX，仿真控制变量：CONTIN、SECORE 及打印标志数 IPRINT 的选择值。在仿真运行一开始，都要一一读入，以便控制仿真运行的最大行程（即仿真运行的最长持续时间），仿真规模和仿真数据选择及跟踪与否。

一般来讲，每次仿真运行的最大行程 $T_{max}$ 由读入的仿真时间 N 和实体极限数 ZTX 来控制。当仿真时间 N 相当大时，仿真运行的最大行程由实体极限 ZTX 来控制，而当实体极限数 ZTX 相当大时，仿真运行的最大行程由仿真时间 N 来控制。但实体极限数 ZTX 又受计算机内存容量的制约。因此，仿真规模的大小，就受到了机器内存容量的限制，仿真规模过大，运行时超过了机器内存的允许容量，仿真将自动中断。这一点，在输入 N、ZTX 值时，应加以注意。

仿真选择控制变量 CONTIN 有两个值：0 或 1。

当仿真是首次运行时，只为 CONTIN 赋 0 值；

当仿真不是首次运行时，可以为 CONTIN 赋 0 或 1 值。

若本次仿真数据不是前次仿真数据的继续时，为 CONTIN 赋 0 值；

若本次仿真数据是前次仿真数据的继续时，为 CONTIN 赋 1 值，在这种情况下，仿真系统主程序将调用“按通道数据存入语句块——CONT 块”，通过指定的通道 REWIND 25，将前次仿真保存在文件中的数据重新读入，本次仿真的数据继续前次仿真的数据，接着存储下来。

仿真选择控制变量 SECORE 也只有两个值：0 或 1。

若本次仿真结束后，此次获得的数据下次运行不再接续使用，则将SECORE赋 0 值；

若本次仿真结束后，保存此次仿真数据，以便下次仿真运行时接续使用，则将 SECORE赋 1 值。此时，仿真系统主程序将调用“指定通道保存全部数据语句块——SAVE 块”，通过指定的通道 REWIND 26，将本次仿真运行的全部数据都写在文件上保存下来，以备下次仿真时接续使用。所以，CONTIN 变量和 SECORE 变量、CONT 块和 SAVE 块是相辅相成的。

IPRINT 是打印标志数，其值有两个：0 或 1。

当 IPRINT 值为 0 时，仿真运行过程中不进行跟踪，只打印出最终结果；

当 IPRINT 值为 1 时，仿真运行过程中进行跟踪打印，同时，也打印出仿真运行的最终结果。

6. 调用 EVENT 块，安排首次事件

在第 6 段，调用“事件安排语句块——EVENT 块”，首先安排第一个事件，使实体进入系统。例如，当 1 号实体源产生的实体从 8 点钟开始到达系统，到达后传送到语句号为 1 的目标语句时，事件产生时间 EVT＝480（分），目标语句号 EVAD＝1，实体源号 NS＝1。则使用调用语句：

```
CALL EVENT(EVT,EVAD,NS,TX,LTX,*1006,IPRINT)
```

即

```
CALL EVENT(480,1,1,TX,LTX,*1006,IPRINT)
```

来实现第一个事件的安排。

当实体到 10 点 30 分钟停止传送时，应在事件表中预先安排一个事件，以表示在 10 点 30 分钟时系统的状态，因此，使用调用语句：

CALL EVENT(630,4,0,TX,LTX,*1006,IPRINT)

来实现 10 点 30 分钟时的事件安排。其中，*1006 表示非正常返回时，返回到最终结果分析语句处。

7. 流程管理 1，调用 ACTIV1 块

在第 7 段，调用“流程管理 1 语句块——ACTIV1 块”，以搜索下一事件或活动。

从本段开始正式进入仿真，先由 ACTIV1 块搜索最早事件，设置实体活动。

在 ACTIV1 块中，通过事件表中当前执行的事件行号 LEV 及事件表中的事件时间 EL(LEV,2)，活动表中当前执行的活动行号 LTX 及活动表中的活动时间 AL(LTX,2)，计算出当前仿真时间(仿真钟)T，同时，计算出当前执行的事件与活动同前次事件与活动的时间差 RT。而且，还要通过事件表中当前执行的事件行号 LEV 及事件表中的目标语句号 EL(LEV,1)和活动表中当前执行的活动行号 LTX 及活动表中的目标语句号 AL(LTX,1)，计算出主程序模型第 8 段中的目标选择器的目标通道号 NADDR 值。

最后通过公用语句 COMMON，将 T，RT，NADDR 值传递给主程序。以使仿真在 ACTIV1的管理之下，继续进行下去。

8. 目标选择器

第 8 段，目标选择器，由计算 GO TO 语句

GO TO(1,2,3,4),NADDR

来实现。

在第 7 段——“流程管理 1——ACTIV1 块”，依照程序规定，通过计算得到的目标通道号 NADDR，使程序跳到模型段部分设定的相应语句处。

9. 模型段

第 9 段——模型段是 CGPSS-F 仿真主程序模型的核心部分。由用户自己根据具体的仿真系统来编写。由图 1-1 可以看出，模型段主要由一组子程序调用语句构成。模型流程由流程管理 1——ACTIV1 和流程管理 2——ACTIV2 来控制。

在模型段，必须根据仿真问题的要求，详细列出相应的功能语句块的调用语句。对于图 1-1 所示的主程序模型，在模型段中，先后依次调用了“实体生成语句块——GENERA 块”、“单设备占用语句块——SEIZE 块”、“单设备服务语句块——WORK 块”、“单设备清除语句块——CLEAR 块”、“实体清除语句块——TERMIN 块”等等。

同时，根据仿真问题的要求，还要将功能语句块中的参数具体化。例如，调用“实体生成语句块——GENERA 块”的格式为：

CALL GENERA(ET,ZTX,PR,ID,TX,LTX,*,IPRINT)

根据仿真问题的要求：

生成实体时间间隔 ET=5；

生成实体数的极限值 ZTX，由前面读语句读入；

优先级 PR=0；

GENERA 块自己调用自己的语句号 ID=1；

实体数组 TX 元素值和实体数组行号 LTX 值随时传送；

*为 *1006，非正常返回到最终结果分析语句标号；

IPRINT 值从前面读入。

这样，具体的 GENERA 语句块的调用语句应为：

CALL GENERA(5,ZTX,0,1,TX,LTX,*1006,IPRINT)

关于上述涉及到的功能语句块的意义和作用，我们将在以后各章中阐述。

10. 流程管理 2，调用 ACTIV2 块

在第 10 段中，调用“流程管理 2 语句块——ACTIV2 块”，搜索条件活动。

每当实体变成不活动时，就要调用 ACTIV2 块，去测试条件活动是否成立。若活动条件成立，实体被激活，通过入口 *1003，返回主程序的目标选择器；若活动条件不成立，则通过入口 *1001，返回主程序，调用“流程管理 1 语句块——ACTIV1 块”，继续搜索设置的实体活动。

在仿真运行过程中，ACTIV2 块随时计算设备状态数组 STATE(6122)的元素值 STATE(K)(K 为设备号)。当 STATE(K)值为“0”时，表示设备(或站、存储器等)忙；当 STATE(K)值为“1”时，表示设备空。当 STATE(K)=1 时，即设备空时，ACTIV2 块再计算与排队选择原则有关的变量 POLC 和 POLVEC(POLC)。进而调用“K 站排队原则语句块——POLICY 块”，以确定实体是按最高优先级服务原则——PFIFO 原则，还是按先进先出，即先到先服务原则——FIFO 原则进行排队。

上述 POLVEC(POLC)是存储队列中实体数组行号 LTX 的数组。

11. 调用 ENDBIN 块，进行最后结果分析

在第 11 段，调用“计算平均等待时间语句块——ENDBIN 块”。通过 ENDBIN 块，计算出存放实体平均等待时间 BINSTA(NBN,1)值。并通过仿真时间 LT，计算出实体在系统中的总时间 EWT。

12. 打印结果

在第 12 段，调用“数据打印语句块——REPORT 块”，打印等待状态、多设备、存储器、实体族、事件表、活动表、实体等数组的所有数据。

如果控制变量 SECURE 赋 1 值，还要调用“保存全部数据区数据语句块——SEVE 块”，将全部数据由通道 REWIND 26 写入文件，以备下次仿真接续使用。

最后，打印出仿真运行结果摘要报告。

## §1.2 CGPSS-F 系统子程序库

CGPSS-F 仿真系统，除主程序模型外，主要是一个用 FORTRAN 语言编写的子例程子程序和函数子程序构成的仿真子程序库。它由 71 个子程序组成，其中 69 个子例程子程序，2 个函数子程序，这 71 个子程序放在名称为 CGPSS-FLZ·OBJ 的子程序库中，主程序利用 CALL 语句或函数语句对每一个需要的子程序进行调用，或子程序之间进行调用。每一个被调用的子程序都完成一个特定的模拟功能。如前面所说，为叙述方便，我们