

混合计标系统软件

译 丛

北京无线电一厂情报室编印

1977年12月

前 言

混合计算系统是由通用模拟计算机和通用数字计算机通过中间的接口装置连接起来而构成的串、并行处理机系统。它是现代进行工程系统仿真的主要计算工具。混合计算系统也是混合计算机，它是由混合软件来进管理 and 编制仿真程序。

为了配合国内开展混合计算系统软件工作，我们翻译了这本译本。加此次翻译、校对工作单位有：七部五〇二所，十七所、海军 518 厂、北京无线电一厂。由于我们外语和软件方面知识有限，时间比较仓促，所以错误之处一定不少，请读者批评指示。

北京无线电一厂 请报发

目 录

1、混合计算软件的过去、现在、和将来	1
2、由数字机设计模拟计算程序	13
3、近代混合计算机的程序生成系统	53
4、利用ALTRAN的混合编译	75
5、关于分时自动排题混合计算机模拟语言编译程序及操作系统	103
6、HYBRID——一个用户研制的混合解释语言	129
7、HELP——一个混合计算机高级语言编译和操作系统	151
8、混合计算机交互性仿真语言(ISL)	161
9、交互性的仿真语言——8 (ISL-8)	177
10、混合——系统——软件：伏以和性能	27
11、混合在计算机	255

混合计算机软件的过去、现在和将来

Harriet B. Rigas

华盛顿国立大学

引言

为了理解和评价混合计算机系统各种软件我们首先必须弄清系统使用环境。它对于我们回顾模拟计算机演变到混合计算机——特别是软件随着硬件的发展过程也是有好处的。

早期的混合计算机

影响模拟和数字计算机结合起来，在模拟计算机技术中最重要的成就要算是伺服设置电位器。这种装置允许使用者通过指定的电位器设置所需要的系数，用按钮启动伺服电机驱动电位器达到规定的值。为了把使用者从大量二进制中解放出来，在研究一套完整的设置方法。按照预先穿孔纸带上的地址和数值自动地进行设置。纸带控制口发出的控制信号与模拟机控制台上来的地址线“或”起来，规定的值通过D/A转换成模拟量用到伺服系统。每次设置由控制信号启动各伺服电机，当时，这样的装置好像是用于点选问题初始值的一个专用数字机。同时，模拟计算机使用者发现许多逻辑判断是至于各个放大口输出的状态。早在六十年代几乎所有稍复杂的模拟计算机都补充足够数量的伺服电位器及大部分逻辑部件。这些逻辑部件一般由与门、或门、非门、与非门、或非门、触发器（其中有些可连成计数器）组成。

模拟计算机连续的和逻辑部分之间的联系是由电子开关构成。即所谓接口部件，逻辑信号是通过对连续信号按一定条件进行测试而产生，当条件成立时产生逻辑“1”，通过继电器或者通过电子开关逻辑信号可以用来开、闭连续信号通路。在计算机运行过

程序中，逻辑符号可动态去控制各部件。在状态，控制元件，控制状态。在这些机中，是控制台控制，所以，必须逻辑符号从其它地方来，以取得控制台上控制。所以，控制元件，控制元件，所以，必要的逻辑符号这不是意外符号。

今天的混合计算机

数字机同模拟机相联结提供两个主要功能：

- (1) 模拟机的设置和监控。
- (2) 执行由于精度要求或复杂性不适合于模拟计算机的算法。

在模拟计算机设置和监控的过程中至少中，数字机要承担设置全部问题中的系数及全部逻辑部件及系统检查。所以，在设置程序提前计算机好，用来确定模拟计算机上，各部件的方法是否必需接口的硬件是为保证两个计算机之间的必要的信息交换。所以，可视为整个系统的一个子系统，接口提供的地址并和两个数据通道使任一部件可被计算机寻址，读出其数值，系数设置和寻址是利用模拟机控制台上的数值寄存器。

复杂的算法需要高速数据传输。今天的混合计算机采用14bit转换率为10万次/秒的A/D，D/A。在混合计算机中，数字计算机的典型应用是作为多变量函数发生器。在这种情况下，数字机必须去不断改变函数值。这个函数是由模拟计算机中一个或几个信号确定，函数值改变是按规定的时间间隔或者按模拟机上由确定条件产生的中断信号来进行。其它方面的应用包括模拟机逐次运行的控制，其中每一次新的运行的初值是由前面模拟计算机结果由数字机按某种算法来给出的。

混合计算机的系统软件

根据上述数字学系统两项基本功能可看出对软件的要求，一般在混合计算机中数字系统是通用数字机厂家提供时较

件为基础。根据计算机规模和外围情况厂家提供一个操作系统。由于处于混合结构中计算机差别很大，小的计算机如 PDP-11，大的计算机如 CDC-6600，虽然系统功能上差别很大，但都是以汇编程序和 Fortran 编译程序为基础。

接口管理软件

通过混合接口，数字机视模拟机为一个很复杂的外围设备。为便于检查和控制使用一系列可调用的子程序，通常以汇编级及编译级来调用，并在系统级上能使用摆脱对模拟机地址。

通过调用子程序可达到的比较重要功能是：

- (1) 设置某个电位器系数装置（电位器或 MDAC）到规定值。
- (2) 读出电位器或 MDAC 设置的系数。
- (3) 读出指定放大器的输出。
- (4) 读出指定的高速 A/D 转换器。
- (5) 设置指定的 D/A 转换器到规定值。
- (6) 设置计算机为规定的执行状态：保持、运行、初始。
- (7) 读出某些或全部逻辑线状态。
- (8) 中断接收器直接到一个规定程序去进行处理。

建立模型基础软件

这部软件除了包含那些便于模拟机操作程序之外还有用于像系数装置设置及从起始程序定义得到初始条件的程序。这样软件例子是 EAI 的混合操作解释程序 (HOI)。

HOI 使用者负责研究模拟机排题量及部件分配，物理系统中变量最大值也必须由程序员来估计，利用这些信息可以写出一个程序去计算所有系数装置值及静态检查值，程序员也能够生模拟机上建立所有部件连接并进行全面静态检查，将读出值同规定值

进行比较并修正其中矛盾。

作为一个解释程序，当模拟机发生了变化时，HOI 就方便地改变程序。这种语言学习和使用都是简单的；命令可以从数字机终端发出也可以从 HOI 的控制程序产生。

从控制台上发出的典型 HOI 命令是

IP006.83.5;

这条命令选择的电位器地址是 P06，装入模拟机寄存器中的值是 0.83，并驱动伺服电机将这个地址的电位器设置为这个值。

本程序是用于系统的解释延迟致使执行很慢，因此在问题求解时间如果数字机速度是受限制因素，在这样的执行期间使用 HOI 是不实际的。

执行时间软件

除了设置和监控外，数字机在问题执行期间有几种工作方式。各种功能原问题有关并随应用而不同。一般是作为函数发生器、积分（通常是模拟的但有时是采用数字的）、寻优、参数识别。

为混合计算机环境中完成这些功能研制了许多软件包，典型的做法是去修改为数字机研制的各种标法来适应混合计算机中两种子系统之间必要的连接。

为混合计算机用 Fortran 编写的某些公用标法是 Fletcher-Powell 寻优，Gear 及 Runge-Kutta 积分，及多变量函数。这些程序包可移动性是受到限制的，由于它们依赖于子系统内的模拟机及由软件支持的子程序。

然而一般说来执行时间软件，像大多数科学计算软件一样是按应用情况设计的并能包括很大范围。虽然这些软件可以用高水平语言使用子程序来写，但很多还是用汇编写的，目的是要同模拟机速度相匹配，以充分利用系统全部能力。

操作系统

虽然数字计算机一般是靠出厂提供的操作系统来运行的，可是混合计算机的操作系统选择受许多因素影响。

长时表，混合计算机是买来用于单个问题或一类相近问题。在化学仿真实验室里就是这种情况，在那里计算机完全服务于某种导弹长期研究工作。在这种情况下，它是不难确定操作系统的指标。

根据问题有可能在单用户和多用户操作系统之间进行选择。如果数字计算机子系统在大部分执行期间工作并使用大部分资源，在单用户系统下运行是合理的，这样的系统连接和维修都简单。反之，如果数字机在大部分执行时间里处于等待，那么多用户系统是实际的，特别当执行要求相当大内存或比较复杂的单元像浮点、流水线或其它昂贵特性时更应如此。在这种情况下，出于经济性要求，数字机必须尽可能高效率使用，允许在混合计算机执行期间能从事其它工作。

在这种方式下几个模拟控制台独立地工作或某种相互结合能够受一个数字机来支持。在这种情况下，操作系统必须能像支持多用户情况一样允许用户的优先权规定。系统软件必须可中断而且能很快地进行处理，对这样用户的混合计算机很多商品系统已经成功的使用了。

在多用户环境中，数字机必须给混合计算机以很高的优先权，这样混合系统才能发挥出它的全部能力。

第一个排队软件

早在1962年模拟计算机程序员就在寻找一种途径以求减轻准备一个标度了的模拟机排队的工作量。Apach 最先尝试按问题的一种高水平描述通过计算机程序去产生模拟机的排队。这？

程序员在 FORTRAN 标准为 EAI 31 只模拟计算机标题。一个简单线性系统以仿真程序探索是解决技术问题的最佳估计问题，但对于一般应用程序员还必须负责提出标题。对于这些标题的估计是模拟计算机程序准备最为麻烦的事。APACK 没有得到广泛的应用。

连续系统的数字机模型

与 APACK 研究几乎同时，开始研究数字连续仿真语言如 MIMA 及 DSLTD。这些语言企图给混合计算机产生另一个仿真工具。通过这些语言程序员可以描述包含解微分方程的问题，然后求得一个完整的仿真运行过程。主要变量随时间变化过程用表格及列表形式来提供。

虽然对很多应用来说数字仿真是可以替代混合计算机的，但不是在所有应用中都能替代混合计算机。

Chubb 及 Daucier 提出过分析报告其中证明了混合计算机比数字仿真更为有效和更经济。

然而，混合计算机程序员感到，数字仿真程序可以提供一种估计变量标题的方式，它是标题模拟计算机所必须的。程序员通常针对一种情况进行一次数字仿真标题，它会反映出标题大小，同时也可以作为对模拟计算机动态检查。

用数字仿真语言来标题排题

毫无疑问模拟计算机程序准备又作进一步的革新应该研究某种自动化的方法。按照这种方法利用数字仿真语言来得到必要的信息。Rigas 及 Combs 描述了怎样修改和补充 CSMP (改进的 DSLTD) 使其不仅能得到完成标题的所需要信息，而且能产生出全部模拟计算机部件连接和输出设置值。这个程序称为 Patch，它可以给出 EAI 310 机的全部连续部分，介面及逻辑部件连接仪

息。

几乎同时在欧洲进行了两个类似工作，Rohr 描述了它的 ACTran 系统，它是一个用 CSMP 作为整个模拟/混合计算机编译的更为有雄心的企图。EIZAS 有一个更进一步计划，HL-1 它具有全局性统一混合编译程序结构，EIZAS 的系统围绕 CSSL 修改文件来设计的。这种语言类似于 CSMP，它由 Sci 确定为连续系统仿真标准语言。EIZAS 的大量工作是去研究这些指标以确定对混合计算机编译更为适宜的语言，而有这三种工作都论证了完整的混合编译能力，它不仅能产生模拟机程序而且能编译数字计算机系统代码及译这个适当的连接。

APSE 也是一个简化的程序，它已在几个设备上运行了，它建立在一种专门设计的语言上，这种语言能产生模拟计算机解题信息。APSE 编译成 HLL 已经成功地用于帮助编制混合计算机中的模拟计算机。

开关排题板

在真正里硬件发展其它方案是自动化编排的模拟计算机。它使全自动化混合计算机更接近于现实，这种可能性曾由 Store 及 Jonsson 在 1968 年提出过，但是所建议的结构需要过多的开关不能代替全部模拟排题板。

Hannauer 在 1969 年提出一个实现必要的开关有效方式。EAI 680 排题板就采用了这种方式并进行了试验。Howe 等人描述了一个小型模拟机，已能在几个用户之间分时地动态重新排题，排题板由一个简要描述互相连接软件包来驱动。

混合计算机未来

当前最有实用价值的是去发展一种混合计算机的完整硬件/软件系统。一个专门的报告描述了未来第四代混合计算机的要求，

新所设 AHCL (Advanced Hybrid Computer Laboratory) 更迅速必须有各种程序许多细节。完全自动化混合计算机系统的适应性已经得到了验证。下一步工作尚需继续去研究使用环境及研究相应的软件。是否利用原级语言来接收？是否分时？这些问题解决了，软件模块才能设计出来以满足其它要求。

总之，未来的用户将可以不针对其计算机去写仿真程序。这种说明必将给予技术人员编写程序的灵活性。

参 考 文 献

1. HCl Reference Handbook, [AI] Publication B0827.0021-2 May 1975
2. "Introduction to a Real-Time Monitor (RTM)" IBM Manua' GR20-08270, July 1970
3. T. T. Grucan "Large Scale Simulation Systems, The Evolution of a State of the Art System," proceedings: 1972 Summer Computer Simulation Conference, San Diego, California, pp. 352-359
4. C. Green, H. P. Hoop, and A. Febroux, "APACHE A Breakthrough in Analog Computing?" IRE Trans. Electron Comput., vol EC-11, October 1962, pp 699-706
5. "Modified Integration Digital Analog Simulation," Syst Eng Group, Aeronautical Syst Div., A.F. Syst, Command, Wright-Patterson AFB, Ohio, Tech Rep.

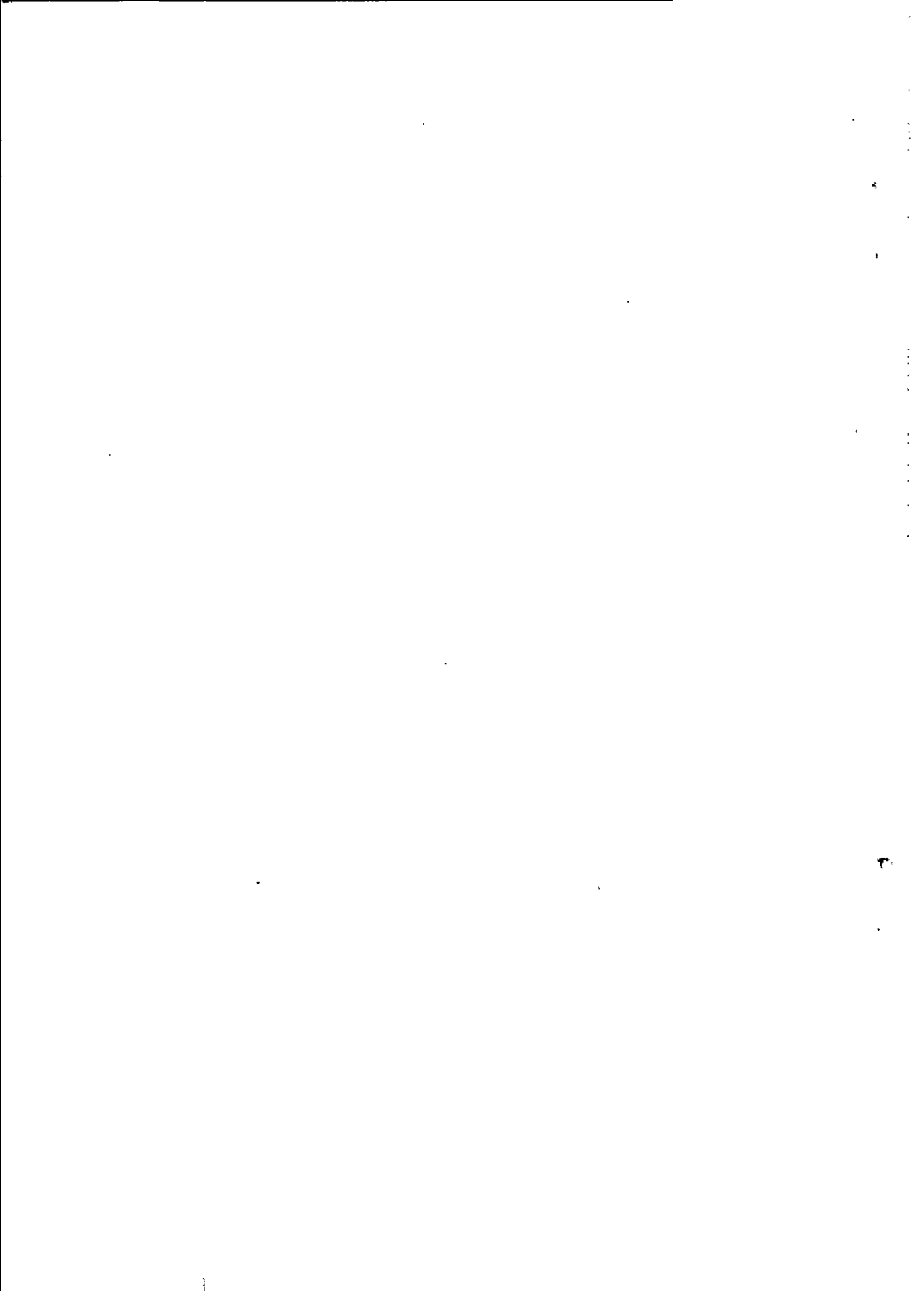
- SEG-TDE-64-1, January 1964.
6. W. H. Syn and D. G. Wyman, "DSL/40 Digital Simulation Language User's Guide" IBM Tech. Rep. TR02-355 San Jose, California, July 1965.
 7. B. A. Chubb, "Economic Evaluation of the CSM? Digital Simulation Language," *Simulation*, March 1970 pp. 101-103.
 8. L. Wolin, "The Wolin-Saucier-Peak (WSP) Scientific Mix: A Quantitative Method for Comparing Hybrid and Digital Computer Performance", Proceedings, Special Symposium on Advanced Hybrid Computing, HQ US Army Materiel Command, Alexandria, Virginia, July 1975. pp. 181-187
 9. H. B. Rigas and D. J. Coombs, "PATCH: Analog Computer Patching from a Digital Simulation Language," *IEEE Transactions on Computers*, Vol. C-20 October 1971. 99-1140-1146
 10. ———, "Modifications to a Digital Simulation Language to Facilitate Automatic Patching" *Simulation*, Vol. 19, October 1972. pp. 133-139
 11. R. D. Brennan and M. Y. Silberberg, "The System/360 Continuous System Modeling Program," *Simulation*, December 1968. pp. 301-308
 12. P. E. Root, "ACTRAN: An Analog/Hybrid Compiler," Proceedings, 6th AICA Congress, Munich, Germany, August 1970

12. ———, "Hybrid Compilers Using Astran," proceedings 1972 Summer Computer Simulation Conference, San Diego, California June 1972, pp. 291-317
14. M. S. Elus, "HL-1, or Toward a Unique Language for All Continuous System Simulation," proceedings, AICA ITR Conference on Hybrid Simulation, Prague, August 1973, pp 11-70
15. ———, "Simulation Language and Hybrid Compilers, Principles and Techniques," proceedings special symposium on Advanced Hybrid Computing, HQ U.S. Army Materiel Command, Alexandria, Virginia, July 1975, pp. 77-79
16. "The SCI system simulation language (SCL)," Simulation, 1966, pp. 281-282
17. Operation Manual, "The APSE Compiler", EAI Pub. 827.0.72-0, May 1975.
18. J. B. Manson, "A Continuous System Simulation Language for an Advanced Hybrid Computing System (A HCSSL)," AICA Vol XVII, October 1971, pp. 283-293.
19. D. A. Starr and J. T. Jonsson, "The Design of an Automatic patching System," Simulation, June 1968, pp 281-288

20. G. Hannerer, "Automatic Patching for Analog and Hybrid Computers," *Simulation*, May 1969, pp. 219-232.
21. R. M. Howe, R. A. Moran, and T. D. Berge, "Time-sharing of Hybrid Computers Using Electronic Patching," *Simulation*, September 1970, pp. 105-112.
22. G. Hannerer and A. Asthana, "Recent Advances in Automatic Patching Software," Proceedings, Special Symposium on Advanced Hybrid Computing, HQ US Army Materiel Command, Alexandria, Virginia, July 1975, pp. 31-36.
23. A. G. Edwards et al., "Technology Report: Advanced Hybrid Computer Systems," HQ US Army Materiel Command, Alexandria, Virginia, June 1973.

译自 COMPUTER Vol. 9 No. 7 July 1976

黄振镛



由数字机设计模拟计算程序

J. K. Denmead, J. S. Gatchouse,
W. G. Littlejohn, M. W. Sage,
G. W. T. White.

摘要

作者参加了英国计算机协会 (The British Computer Society) 组织成立的一个工作小组, 以研制一套帮助通用模拟机排题和校验的数字机程序和子程序。这个工作的主要特点是: 最后得到的程序尽可能是通用的和方便的, 数据输入尽可能简单明了, 尽量使用低水平的程序语言, 得到的结果可以在各种数字机上使用。

源程序是用ALGOL写的; 此外通过人工翻译得到了FORTRAN译本。本工作的重要结果之一是这个翻译容易得到。

本文包括对数据输入的语法与词法说明以及数据在程序中处理的格式。主程序包括方程的简化和整理、选比例尺、校验、分配和数据输出等程序。

许多程序是非常有意义的, 特别是有关方程简化和求解代数方程回路程序已找到在其他方面的应用。

主程序也可以用作连续系统的数字机仿真, 这只要调用其有关动态试验的程序即可; 然而在运行速度上比起专门为此目的而编排的程序要稍微慢一点。

作为这个课题的结果, 工作小组成员对实现低水平仿真语言的问题得到相当深入的理解, 而且希望从有这方面工作的人员能够组成某种形式的协作。

一. 引言

1964年秋, 为了帮助模拟计算机使用者而扩充数字程序库这个明确目的在British Computer Society支持下成立了一个工

作小组。那时，如像 APACHE [1]、COP [2]、PACE SCALER [3] 和 MIDAS [4] 等程序已被採用，但这些程序或者专门对某些特定的机口如 IBM 7094 PACE 231 适用，或者是专门为研制者的特定目的所产生的程序。由于这些缺点，工作小组决定研究一种方法，它将在一个变水平上“接受输入的数学方程数据并产生模拟计算机程序设计数据，这种方式，实质上与所使用数字机如要设计程序的模拟机无关。

和这个数学方法有关的程序在不同的机构内发展，工作小组成员参加了这些机构。因而在早期阶段十分详细地说明自动编制模拟计算机程序所需的方法是很重要的，给这些方法集体地起个名字叫《Automatic Programming and Scaling of Equations》(APSE) 其结构图如图一所示。