计算机工程与应用

Computer Engineering and Applications

3
1986

计算机工程与应用编辑部

解析运算语言对科技计算的应用*

中国科学院物理研究所　张淑誉

中国科学院理论物理研究所　郝柏林

摘要　解析运算语言在国外已有近三十年的发展史，而且已经筛选出像MACSYMA，SMP，REDUCE等等相当成熟的语言，应用范围日益广泛。中国科学院计算中心和物理研究所IBM计算机上的REDUCE2语言也已经运行了一两年，积累了一些经验，目前正在向REDUCE3.0过渡。因此，向我国数学软件工作者介绍和推广这类语言的时机已经成熟。

解析运算语言为科技数学软件提供了新的工具，要求有新的算法，开辟着新的应用前景。本文先介绍解析运算语言的发展简史和基本功能。然后讨论这类语言的几种应用。最后还简要分析了解析运算语言的实现问题。

我们期望，不久的将来在科技数学软件工作者的语汇中，“数值方法”将被更一般的“计算机方法”（包括数值、图象、解析和智能等各种方法）所替代。

一、引　言

随着计算机处理能力的提高，各种非数值方法将在科学计算中起更多作用。解析运算，即由公式符号到公式符号的运算，只是最为简单，因而也比较成熟的一类非数值计算。自从1953年有人编写解析微分程序以来，三十年中发展出多种专门用于解析运算的语言，召开过多次关于解析运算的国际会议，发展了许多适用于解析运算的算法，开辟了愈来愈多的应用方面。这是一个广阔的领域，应当引起科技软件工作者的注意。

本文作者近五年来一直为引入和实现解析运算语言作调查和准备。近两年先后在中国科学院物理研究所和中国科学院计算中心的IBM计算机上实现了标准LISP语言和基于LISP的REDUCE语言。LISP主要是为了人工智能方面的应用，REDUCE则是一种功能强大的解析运算语言。我们使用这些语言的经验还不多。下面讲一点我们的认识，希望引起讨论。

二、解析运算语言的发展简史

我们不提那些为了某种专门目的（如微分或者求群表示）而用机器指令或某种高级语言所写的程序，而只关注具有相当通用性的解析运算语言。

FORMAC可以作为早期解析运算语言的一个代表。它不是一个独立的语言，而是用FORTRAN语言（后来也用PL/1）写出的解析运算子程序包（其中约有80个子程序）。用宿主语言写出的一系列调用，组成这个语言的处理过程，它可以进行多项式运算。初等函数表达式的形式微分，但不能作形式积分和矩阵运算。这个语言从1954—55年开始发展，后来它的设计者转去作其它事情；虽然七十年代后期还有人作过改进，现在已基本过时了。

六十年代初物理工作者几乎同时设计了两种解析运算语言，但是走了两条不同的道路。一个是M·Veltman提出的SCHOONSCHIP语言。它是用CDC6600（开始是IBM7090）

*第二次全国科技应用软件学术会议论文

主要特点是：

设计思想新颖，硬件软化，稳定可靠，用户使用、调试维护方便，利于扩充（最多可达32个I/O八位并行通道）。

四、交通灯驱动电路

图4示出交通灯驱动电路．采用光耦合器，使计算机与可控硅功率输出电路严格隔离，保证了整个微机控制系统的安全运行。

此光电控制电路，与由灯色分配、单晶振荡、脉冲隔离变压器器等组成的繁杂的交通信号控制电路相比较，性能/价格比高，宜于推广。

首先，我们设计了“灯色控制六步程序”，详见下表。

六步程序 \ 二进制代码 位	$\overline{Q_7}\overline{Q_6}\overline{Q_5}\overline{Q_4}\overline{Q_3}\overline{Q_2}\overline{Q_1}\overline{Q_0}$	十六进制代码	符　　号
东西绿　南北红	00000011	03	EWG，SNR
东西绿闪　南北红	00000110	06	EWGF，SNR
东西黄　南北红	00001010	0A	EWA，SNR
南北绿　东西红	00110000	30	SNG，EWR
南北绿闪　东西红	01100000	60	SNGF，EWR
南北黄　东西红	10100000	A0	SNA，EWR

其中：$\overline{Q_0}$代表东西绿，$\overline{Q_1}$”南北红，$\overline{Q_2}$”东西绿闪，$\overline{Q_3}$”东西黄，$\overline{Q_4}$”南北绿，$\overline{Q_5}$”东西红，$\overline{Q_6}$”南北绿闪，$\overline{Q_7}$”南北黄。

微计算机发出运行指令，第一步灯色控制是“东西绿，南北红”，相应的十六进制代码是（03）。一个高电平信号通过二极管2AK03J加到GD光耦合器中，使得GD的输出端处于饱和状态，而与非驱动器Q_{01}、Q_{02}输出高电平，触发SCR_1、SCR2的控制极。可控硅导通，输出功率，点亮南北方向的交通灯。

为了提示司机，保证行车安全，给出一个固定的2秒钟绿闪时间。

由三个与非门Y_{10}、Y_{11}、Y_{12}、电阻器Ro和电容器C_0组成的一个环形振荡器，其振荡频率为每秒5 HZ，专门为“绿闪”信号提供服务。

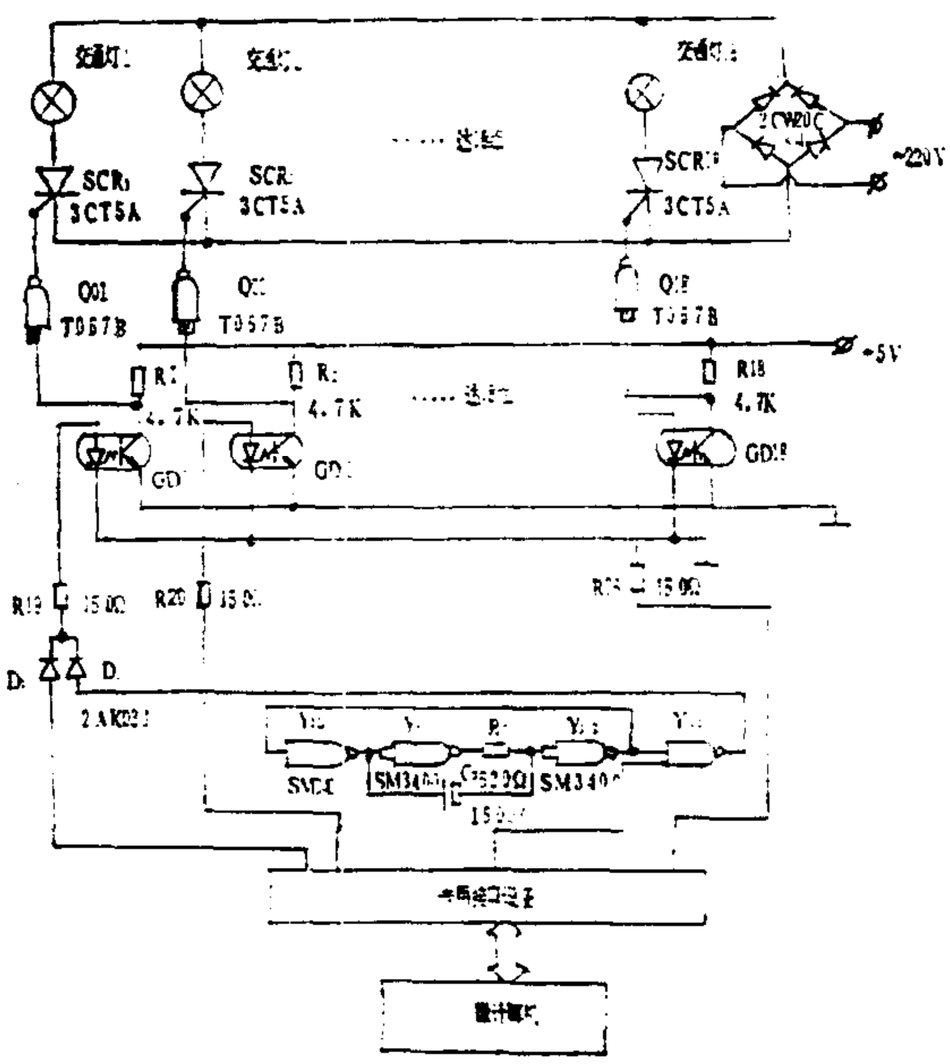

图 4　交通灯驱动电路

第二步灯色控制是：“东西绿闪，南北红”，相应的十六进制代码是（06）。此时$\overline{Q_2}$由“0”变成“1”，与绿闪输出电路中的与非门Y_{13}SM3400相与，Y_{13}输出振荡信号，再经过一个或门馈至光电控制电路，交通灯按其给定的频率闪烁。“南北红”的控制过程同前。

第三步、四步……乃至第六步的控制过程，与前边所讲的道理是一样的。

可控硅主控电路的电源为直流220V，输出功率≤1000W，光耦合电路的电源为＋5V。

的机器指令写成的，采用了不少压缩存贮、提高效率的技巧，因而执行速度较高，但难于纳入高级语言的行列，也不容易移植到其它型号的计算机上去。SCHOONSCHIP至今仍在CDC系列的许多机器上运行，受到使用者推崇。

另一个语言是A。C。Hearn设计的REDUCE，它是基于LISP语言的（后来又专门规定了它的宿主语言“标准LISP”。基于LISP这样的通用表处理语言，使得REDUCE比较容易移植到其它机器上（目前在DEC、VAX、IBM、CDC、UNIVAC、Burroughs、Siemens等机器上均有REDUCE在运行），然而它的处理速度也就比直接用机器指令编写的语言为纸。因此，有人实现了功能与REDUCE接近，但用IBM机器指令写出的AMP语言，大为提高了运行效率。

基于LISP的解析推导语言还有麻省理工学院设计的MACSYMA。事实上宿主语言也进行过扩充，成为MACLISP。这个语言的发展自1969年以来已耗费了100人年，目前已经商品化。它包含约30万行指令，可在DEC，VAX，Tops 20，HoneywellMultics，symbolic LM-2等机器的UNIX操作系统下运行。目前一般认为MACSYMA是功能最为完备的解析运算语言。

最近出现了用C语言写的SMP语言，它也是在VAX等机器的UNIX系统下运行。SMP诞生在加州理工学院，目前受到了一些物理工作者的好评。

国内最早从事解析推导语言的是中国科学院数学研究所的陆启铿和陈东岳。陈东岳设计并先后在DJS6和FELIX512机上实现了FCY（“符号程序语言”），并在1983年9月通过了鉴定。数学研究所还引进了用FORTRAN写的解析推导语言SAC 2。本文作者则在IBM系统上实现了1979年版的REDUCE，目前正在向1983年版的REDUCE30过渡。天津大学张春霆等同志，则实现了微型机上的π—MATH。

我们完全没有提到的语言还有很多，例如IBM公司的SCRATCHPAD，贝尔实验的ALTPAN，以及苏联人搞的ANALYTIK等等。还有许多更为专门的语言，如天体力学计算用的MAO，AMC，广义相对论用的ALAM，CLAM，SHEEP，CAMAL，量子场论计算用的ASHMEDAI等。六十年代初以来发展了约60种解析运算语言，其中多数只有语言设计者自己在使用。真正通过竞争筛选下来的，只有MACSYMA，SMP，REDUCE等十来种较为通用的语言。当然，今后还会发展出更完备的这类系统。下面的介绍中，往往用REDUCE来举例，因为这是我们较为熟悉的语言。

三、解析运算语言的特点和功能

与数值运算相比，解析运算有两个显著不同之处，第一。运算过程中要求的存贮容量基本上不能事先估计，且容量通常随问题规模指数增长。这是因为许多表达式在展开后会得出许多项（项数通常按二项式系数组合出来）。但最终相消后又可能给出很短的结果。第二，运算时间往往也是指数增长，而且难于予先估算。这两条就决定了现有解析运算语言的一些共同特点。例如：

（一）采取树枝状的表结构来存放数据。计算过程中要多次自动回收已经不再使用的存贮空间。

（二）多数解析运算语言以会话方式为主，以便借助人的干预来提高处理效率。例如，人总是可以根据前面的计算经验判断会出现什么样的公因子，并且可以随时加以检验来修正判断。如果猜测有一个因子X+Y，可令X=—Y看是否为零。这比由程序去从头析因子要快得多。许多语言允许人们作各种“提示”。例如，在析因子时指定一个参考素数。

（三）如果说数值运算语言的基本手段是循环，则解析语言的基本手段是递归，而且要保证极为可观的递归深度。FORTRAN语言

根本不允许递归地定义函数。ALGOL文本虽然原则上允许递归，但多数编译程序没有实现。在解析运算语言中，用下面的过程定义契比雪夫多项式就是完全合法的：

```
PROCEDURE TCH(N, X);
IF N=0  THEN 1 ELSE  IF N= 1
THEN  X ELSE
2*X*TCH (N—1,X)-TCH(N—2,X);
```

这时要求

```
TCH(4,Z) ;
```

就会印出

$$8*Z^4-8*Z^2+1$$

（四）同样由于不能事先估计结果表达式的长短，输出时的格式描述意义不大。通常只提供几种“自然”的输出方式。例如明显写出下标和幂次的“二维”输出：

$$X^2_1+X_2{}^3-SIN(Y)^2$$

然而这种输出不能直接用作输入。因此，还要提供与输入语法一致的输出方式，例如上式可印成：

```
X(1)**2+X(2)**3-SIN(Y)**2;
```

解析运算输出的许多冗长结果，往往只用作进一步数值计算的公式。因此，不少语言提供了FORTRAN语句形式的输出，使得计算结果不经手工处理就能变为FORTRAN子程序。

（五）解析计算中往往有多种选择，例如，是否通分母，是否提出公因子，多项式系数是保持精确的有理数形式，还是化为浮点数，以及变量和表达式的代换规则，函数的微分规则等等。因此，解析语言中设有大量开关和说明。数值计算语言中的说明，通常在编译阶段就完成使命，而解析运算语言中的开关和说明，正是为了控制运算进程。如果一切开关和说明都同时有效，处理速度一定会大为降低，因此，这些开关和说明的作用应视需要而随时启停。这在会话方式中是容易的，在批处理中也要允许这样作。

下面介绍解析运算语言的基本功能，它们本身就是一些最直接的应用：

（一）多项式运算：符号多项式的相加、相乘、归并同类项，取出各项的系数表达式，以及因式分解等等。整系数多项式的因式分解，特别在多变量情况下，要求有专门算法。在REDUCE语言中，因式分解程序包是最长的一个。甚至比形式积分还长。

（二）形式微分：初等函数的微分仍是初等函数，其它函数的微分规则也可以具体定义。复合函数的高阶导数，很快就会变得极为复杂，而这正是计算机处理的拿手好题。例如，表达式S是

```
S: =SIN(X*EXP(X*COS(Y) )+Z),
```

我们要计算六阶偏导数

$$\frac{\partial^6 S}{\partial^3 X\,\partial^2 Y\,\partial Z}$$

在REDUCE语言中只要利用微分算子DF写成：

```
DF(S,X,3,Y,2.Z);
```

计算结果长达35项。手工计算虽然原则上仍是可能的，但很难不出错误地完成。

虽然没有显式函数关系，也可以只说明F依赖于X，而在最终结果中保留微分符号。例如：

```
DEPEND F,X;
A: =DF(SIN(X*F),X,2) ;
```

其结果写成普通数学形式是：

$$A=\frac{\partial^2 F}{\partial X^2}XCos(xF)-\left(\frac{\partial F^1}{\partial x}\right)^2 x^2Sin(xF)+2\frac{\partial F}{\partial x}Cos(xF)$$

$$-2\frac{\partial F}{\partial x}xFSin(xF)-Sin(xF)F^2$$

这对于一般公式的推导当然很有用。

（三）**形式矩阵运算**：首先允许说明符号矩阵，完成矩阵的相加和相乘。其次，对于方阵可以求逆，求行列式和求阵迹。如果矩阵 A 的阶数不高，且所含的符号不多，求行列式只须写

DET(A)；

然而，只要矩阵稍大稍复杂，最好由人直接选择恰当算法并干预运算过程，才能收到实效。我们曾经计算一个16×16的行列式，其中共有11个不同的代数符号。直接使用DET算符，立即遇到存贮问题。后来改用了Raynal建议的算法，才得出结果。数值方法中的高斯一约旦消去等等，是完全不适用于解析运算的。

（四）**形式积分**：不定积分的求积是困难得多的问题，因为原函数可能无法用已知的函数写出来。然而既使是完全可以积出来的三角函数或有理函数的积分，往往也要花费巨量手工劳动。因此许多较为完备的解析运算语言都配备了积分程序包。在REDUCE中为了对X 求积XSin(ax)，只须写

INT(X*SIN(A*X),X)；

而对于误差函数

$$\mathrm{erf}(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-y^2}dy$$

曾经用MACSYMA语言计算出如下的积分：

$$\int \mathrm{erf}(ax)\mathrm{erf}(bx)dx=-\frac{\sqrt{a^2+b^2}\,\mathrm{erf}(\sqrt{a^2+b^2}\,x)}{ab\sqrt{\pi}}$$

$$+x\mathrm{erf}(ax)\mathrm{erf}(bx)+\frac{e^{-a^2x^2}\mathrm{erf}(bx)}{a\sqrt{\pi}}$$

$$+\frac{e^{-b^2x^2}\mathrm{erf}(ax)}{b\sqrt{\pi}},$$

而这是在现有积分表中查不到的结果！

这里顺便提一下常微分方程（特别是非线性的）求积问题。解析运算语言中目前还没有定义这类算子，需要由用户编写出相应过程。人们已经作过的尝试是相当成功的，Kamke常微分方程手册中的333个方程作出90%，Murphy手册中的715个方程作出95%，而两本大学生习题积中的方程则全部解出来了。

（五）**表达式的代换**：解析计算中经常遇到表达式中的某些成分要代换成其它表达式，然后再加以整理和简化。这些代换有时只进行一次，有时要不断递归地进行到底，有时只对特定的变量名字进行，有时则对一切可能的变量的特定函数关系进行。只进行一次的代换可用算子SUB实现。例如：

SUB(X=1+A*Y,F(X))；

就把F(X)换成Y的函数。多次进行代换则可用LET语句说明，例如说明：

FORALL X LET SIN(X)**2=1-COS(X)**2,

SIN(X+Y)=SIN(X)*COS(Y)+SIN(Y)*COS(X)；

等关系之后，可以容易地进行三角函数恒等式的推导或验证。

（六）非对易量的计算

早期的解析运算语言只处理对易量，例如A*B-B*A将简化为零。现在REDUCE中可以说明非对易量，例如

NONCOM A, B；

于是可以进行量子力学中的算子运算，或者定

义各种代数。高能物理计算中常遇到的GAMMA矩阵，实际上就是Clifford代数的运算，目前在几种解析运算语言中均可进行。

我们不再一一列举解析运算语言的众多功能，而从应用的角度再看几个实例。

四、解析运算语言的应用

在科技计算中推导途经清楚，但须花费极大手工劳动才能完成的计算课题不胜枚举。这类计算的自动化是人工智能的简单应用，远没有“定理证明”、自然语言的理解等等问题复杂，因而也有较多的应用结果。下面指出几个领域。

（一）天体力学中从三体问题开始，所有多体问题都不可能精确地求得解析答案。上个世纪就发展了许多级数解法，但其计算十分冗长。1867年法国天文学家C.Delaunnay发表了“月球运动理论”一书，集中了耗费二十年心血推得的四万个公式。这些公式由于人造卫星的轨道计算而获得了新的意义。1970年有人[13]用解析运算语言重新检验了Delaynnay的结果，在计算机上用了20个小时，只发现了三个错误（实际是一处错误引起的）。这使人想起出现数值计算机后，人们重新检验“巴罗数值表”，只在最后一位有效值上发现一两处小错的故事。这些当时只能用手工完成大量精确计算的科学工作者，至今使后人肃然起敬。Delaunnay的两卷巨著，目前反倒成为检验新的解析运算程序的范本。

（二）广义相对论和引力理论的计算是另一个最早使用解析运算的领域，这里最主要的计算是函数的微分：从度规张最g_{ij}出发，求克里斯多费符号Γ^{i}_{jh}，黎曼张量，Ricci张量和爱因斯坦张量，写出爱因斯坦场方程，再进一步寻求场方程的某些精确解，每一步都是微分和求和。这里典型的效果是：使用笔和纸作三个月的推导，在计算机上只用两三分钟(当然，必须先有现成的解析运算语言，然后再花一定精力去熟悉它并写出程序）。早期的实例如旋转星引力辐射的计算，使用CLAM语言，从度规张量到爱因斯坦张量的计算在CDC6500计算机上只用32秒。近来已有人实现了外微分程序包，用以进行使用现代微分几何语言的广义相对论计算。

（三）粒子物理学中高阶微扰论的计算从三个方面对解析处理提出了要求。第一、高阶微扰论中的每一项通常用一个费曼图代表，怎样保证画出了一切可能的费曼图，而没有重复或遗漏。这是一个用计算机产生费曼图的问题。第二，每一个高阶图的计算，都要遇到许多个GAMMA矩阵（Clifford代数的元素）先求乘积，再求阵迹的问题。REDUCE语言中有一个专门的程序包来处理这类计算。第三，每一个高阶微扰项最终归结为一个高维积分的计算，这里又要求助于解析积分（当然还有数值积分）程序。我们不在这里深入技术细节，只指出一些文献[15]。上述计算通常在四维时空中进行，但近十几年来规范场理论的发展，往往要求把积分推广到一般的n维空间，n不必是整数，实行所谓“维数正规化”。用nREDUCE语言实现维数正规化，也已有人作过尝试。

（四）极高清度的数值计算。有一些计算（例如等离子体稳定性的讨论）对于数值精度十分敏感。使用普通数值语言如FORTRAN时，即使在CDC计算机上双倍精度只有120位（二进制），这往往是不够的。然而，数值计算只是符号运算的特例，凡是用表结构贮存数据的解析运算语言，原则上都可以进行任意精度的数值计算。实际精度不受具体计算机的限制，而只取决于总的系统资源。例如，在REDUCE中有一个BIGFLOAT（大浮点数）程序包，只要写

```
ON BIGFLOAT, NUMVAL,
PRECISION 100;
```

以后包括SQRT，SIN，LOG等等初等函数的数值计算，就一律保持100位十进制有效值。另一类高精度数值计算，是只使用整数和有理数（整数之比）的完全准确的计算。这更是一

从度
计算
程序
义相

算从
高阶
怎

重复
的问
到许
）先
言中
三，
分的
有数
节，
维时
，往
必是

)UC
试。
计算
精度
AN
0位
值计
存数
意精
的限
RED
程序

数的
值。
有理
是一

切解析运算语言本来所具备的。

（五）微分方程的对称性，可积性和稳定性分析。数值计算中通常每次针对一组固定参数求积微分方程，得出一个具体解。为了在参数空间中考察整个解族的性质，要花费很大计算量。然而微分方程理论中早就发展了一些强大的分析方法，但它们的实际应用一直受到解析计算量的限制。我们只略举数例。首先是常微和偏微分方程的局部变换群，对于方程降阶或解的分类有直接用途，实际计算涉及大量形式微分运算。最近有人写出了REDUCE程序，专门处理这一问题。其次，常微分方程往往在参数空间的某些点上成为可积的，于是提出在这些点寻求精确解并在附近作微扰展开的问题。这显然是一类适于用解析处理的问题。第三，非线性微分方程的稳定性分析，例如Hopf分岔后产生的极限环的稳定性条件，就涉极为冗长的计算，但算法本身已经清楚。看来编写相应的解析运算程序的时机已经成熟。

其实在线性常微分方程组的求解方面，解析语言也有用武之地。例如，使用拉氏变换方法，先把高阶有理分式分解为低阶有理分式之和以便套用现成的反变换公式，中间计算量有时就颇为可观，而有理分式的处理在任何解析运算语言中都是常规功能。

（六）最后，举作者之一曾经使用解析运算语言处理的另一类问题。在混沌问题所涉及的一维非线性映象的研究中，要求解如下的函数方程

$$\alpha^{-1}f(\alpha X)=-fofo\cdots fo(X),$$

同时定出“本征值”α。如果把未知函数展开成幂级数。

$$f(X)=1+AX^2+BX^4+CX^6+\cdots,$$

则比较两端系数，可得出定α，A，B，C…的非线性方程组（再用数值方法求解）。这里遇到往多项式中代多项式的手续，在REDUCE语言中只须写：

```
OPERATOR F;
ARRAY W(8). 存放系数方程的数组
LET X..9=0; 只展开到8次幂
FOR ALL X LET F(X)=1+A.X..2
+B.X..4+C.X..6+D.X..8;
COEFF(F(ALPHA.X)+ALPHA.F
(F(F(F(X))), X, W);
```

这里只写了n=4的情形，只令W(0)，W(2)，W(4)，W(6)，W(8)分别等于零就得到了决定α，A,B,C，D的五个非线性方程式。

我们再简略指出一些可以运用解析运算语言的问题，它们目前尚未引起足够的重视：

（一）各种微分方程级数解的运用：自上一个世纪以来发展的许多级数解，由于收敛太慢，往往只有证明意义，而无实际用途，而且人工推导只能限于求得级数的前几项，使用解析语言，求得1000项展开也不算困难。如果这样长的级数足以保证所需精度，则只需一劳永地求得级数展开，解方程就归结为算函数，在经济上也可能是合算的。前面提到人们重新检验上个世纪求得的月球运行公式，也是因此引起的。

即使对于收敛极慢，甚至发散的级数，近几十年来也发展了许多从中提取有用的精确信息的方法。其中之一是Pade′变换法。对于每个具体级数的Pade′变换，可以靠数值计算程序解决，为求得变系数的一般公式，则非求助于解析运算语言不可。

（二）高阶差分格式的推导。十二阶隆格—库塔法公式的推导已经非常繁琐，如果要利用推导中出现的一点任意性（这类公式中总有一点可以选择的余地），来作某种改进，更要增加工作量。如果带着一些可变系数来推导高阶差分公式，提出消去某些特定项的要求以提高逼近精度，再进一步推导余项，估计误差等等，都可以使用解析语言。我们希望这一新的可能性引起计算数学工作者注意。

（三）与群论有关的大量计算，目前已有一批用数值运算语言写出的专用程序。然而，它们都是针对具体的群和具体的应用问题编写的，没有普遍适用性。事实上可以提出许多一

般问题，例如在有限群情况下，只要给定生成元，就产生一切不可约表示，或是给定大群的不可约表示，计算由之诱导出的子群的表示，或反过来由子群不可约表示，推出大群不可约表示，等等，这些都是算法早就成熟，但每一个具体情况下仍须付出大量劳动的问题。应当用解析运算语言，写出相应的程序，逐步使群论中的计算课题，成为常规软件。

在结束本节之前，我们想强调指出，目前的解析运算语言只提供了解决问题的工具，并不能立即给出问题的答案。其实这一点与数值计算语言是一样的，有了FORTRAN语言，人们还得去编写求解线性方程组，计算本征值，求数值积分的各种子程序。解析运算面临着更为严重的算法问题，因为绝大多数数值方法是不能搬用到解析问题中来的。这里还有大量研究工作要作。事实上，古典的形式积分理论，近年来就在解析运算自动化推动下，有了新的发展。

五、解析语言的实现问题

这不是本文主旨，我们只略加叙述。目前解析运算语言有多种实现方式。

效率最高的办法，当然是采用某种压缩存贮的技术保存表达式，然后用汇编语言写出处理程序。SCHOONSCHIP 就是这样实现的，因而至今限于CDC系列的计算机，也很难与其它高级语言耦合。也可以使用现成的高级语言，如FORTRAN或PL/1来编写解析运算程序，这是FORMAC和SAC-2 走过的道路。这种办法可移植性提高了，但运行效率降低。

采用表结构存贮数据，直接用LISP语言写出处理程序，特别适用于某些类型的解析运算（如微分），也便于移值，但终究由于LISP语言本身过于专门，很难普遍推广。于是人们保留 LISP 作为内部语言，而为解析运算部分设计更为接近ALGOL，FORTRAN等高级语言的外部形式。REDUCE和MACSYMA都是这样作的，这类语言易于移植和推广，但仍然面临存贮结构不经济和运行效率低的问题。然而，随着硬件的发展，对于多数应用课题而言这两个问题正在成为过去。在16兆字节虚存支持下的 REDUCE，目前已能处理相当大的实际问题。我们认为MACSYMA和REDUCE是有发展前途的。

另一方面，REDUCE的发展，又反过来给LISP赋予了更方便的外部形式。目前在 REDUCE的符号模式中，可以使用 IF…THEN…ELSE，WHILE…DO这样的语句结构，调用任何LISP函数（这是所谓RLISP 语言）。这一定会使LISP成为更普及的语言。

至于用C语言编写解析运算程序（如SMP），我们没有实际感受，目前尚不能评论。

六、结　语

计算机科学的迅猛进步，展现出许多过去难以想像的发展前景。即使是每天与计算机打交道的科技工作者，也必须时时展望全局与未来，才不会囿于原来熟悉的领域，于不知不觉中逐渐落后于时代的步伐。结合本文所介绍的解析运算语言，我们愿意指出几个应当引起注意的方向：

（一）数值计算与解析运算的结合，将大为开拓科技应用软件的应用范围，提出许多新的数学方法，提高科技计算全过程的处理效率。

（二）计算机硬件处理能力很快将受到基本物理原理的限制（小型化有其限度，讯号传播速度也有限度，等等）。但数值机与模拟机（特别是光学模拟机）的杂交，可能再使人类的信息处理能力大为提高一步。

（三）大规模集成的中央处理机成本将越来越低，用这些片子推成专门的处理机，寄生于现有大型机上（借助后者的海量存贮和输出输入设备），将提供很经济的处理能力。也许解析运算语言也可以用这种方式“硬化”，以大大提高运行效率。

（四）用大量片子堆成二维、三维乃至高

一个档案管理系统的设计与实现及体会点滴

长沙铁道学院　　乐　晓　波

摘要：本文介绍了用微型计算机进行中文文书档案管理的设计方法和数据处理技术，具体介绍ZDGX中文档案管理的设计与实现过程，进而总结了一般信息检索系统的设计与实现的一般方法。ZDGX管理系统完成中文文书档案的存贮，检索，编目，造表，修改，删除等管理工作。在检索和编目时，系统可用档案的任何数据项作为关键字进行检索或编目，并可根据用户要求按这些关键字的“与”、“或”等逻辑条件进行检索或编目。本文着重介绍整个系统的设计过程和思想，最后对系统的经济效益作了初步分析。

目前，用微型计算机进行大量的数据检索在内存容量及检索速度上均存在问题。本系统由于采用内存复盖及信息压缩等技术，同时合理使用硬件资源，合理选择程序编制语言，对磁盘文件严格控制，有效地解决了一些普遍存在的问题。本系统已在铁道部办公厅档案处得到应用达半年之久，并已收到良好的经济效益。

一、引　　言

随着微型计算机的飞速发展，微型计算机越来越广泛地应用到社会主义建设的各个领域。但微型计算机是否能应用于信息量较大的档案管理部门进行“海量”的数据处理还有不少人表示怀疑。的确，微型计算机在档案管理中的应用是存在不少难以解决的问题。对于这些问题，我们作了一些工作，并且在CROMEMCO SYSTEM-Ⅲ微型机上比较成功地研制出了ZDGX中文档案管理系统，已收到良好的经济效益。尽管我们是在一具体的机型上对一具体的档案管理进行设计的，但我们认为，任何档案管理（科技档案管理，人事档案管理，图书档案管理，仓库档案管理等）都大同小异，存在着许多共同的特点和管理方式，在设计思想上不存在根本的差异。因此，本文中我们，从一具体的档案管理的设计及实现出发，总结出一般的软件管理系统的设计及实现方法。

我们的经验是：开发任何一个档案管理系统一般都要经过如下几个步骤：1．与用户合作，了解手工管理过程，提出系统要求，进行系统设计。2.系统可行性研究，包括技术可行性及其经济效益的估计。3.确定方案，制定文本，进行过程设计。4.程序设计，编码，程序调试。5.实验运行，纠错，改进。6.提交用户使用。7.维护。

下面我们具体介绍ZDGX中文档案管理系统的设计思想和点滴体会。

二、系统功能的确定

在一个管理系统研制项目提出后，首先应

（接上页）

维的阵列机，不久将成为发展主流。这里提出的高维并行算法问题，有些超出了人类的天赋本能（大脑皮层只是接近二维的处理系统！），要求作出革命性的突破。在并行算法方面，解析运算提出与数值计算完全不同的问题，也许更易于处理。目前还没有这方面的经验。

对于解析运算这样广阔的发展领域，这篇短文难免挂一漏万。我们希望引起讨论，听到批评。

该确定系统的功能。系统功能的确定依赖于具体用户的现有工作方式以及可能提供的管理和控制信息。因此在具体确定系统功能之前，应进行仔细地，全面地科研调查。但是，确定系统的功能（包括功能的强弱程度）并非一件轻而易举的事。由于一般档案管理人员不甚了解微型计算机，他们往往对计算机能干些什么，怎些干缺乏确切的概念，从而没有定量的标准，以致提出的要求往往是含糊不定。同时，又由于他们往往对于管理方式叙述不甚全面，因而他们所提出的要求也是不完全、不稳定的。另外，系统研制人员往往对档案管理并不熟悉，容易自圆其说地对任务要求加以补充，删减或规定数量界限，结果往往与实际情况有很大距离。再者，由于档案管理人员与信息处理系统研制人员的专业背景及其经历不同，因此在科研调查过程中，他们之间的思想交流过程中的误解也是在所难免的，这会给系统的功能的确定带来隐患。因此，我们认为，反复与用户商讨，仔细了解手工管理过程，甚至下到具体档案管理部门实习考查是很有必要的。若忽视了这些重要环节而轻率地确定系统的功能并进入实际系统的设计阶段，往往造成返工，浪费大量的时间和精力。在这个问题上我们是深有体会的。

我们对国家档案馆的标准管理方式进行了解后，经过几个月的努力，在铁道部办公厅档案处实习考查，进行了深入细致地调研工作，最后根据所了解的全部情况，在对技术的可行性和有效性进行分析之后的基础上，对系统的具体要求作了某些容许且必要的限制，从而确定了ZDGX中文档案管理系统有如下功能：

（1）能将档案文件方便地建立（即存贮）在用户软盘上，可灵活地停止和继续建立。建立过程中，能进行全屏幕编辑，可方便地删除和改错。

（2）以某立档单位的全部文档信息为一系统文件来进行建盘，检索，修改，删除和编目等工作。

（3）对所有文件能用全宗号，文件号，期限，姓名，日期(日期可仅指定年或年、月，也可年、月、日均指定)，主题词等数据项的“与”、“或”逻辑条件来进行检索，编目，分类，修改，删除等工作。其中主题词可分为三级，每条文件目录可拥有三套主题词。

（4）输入、出及各种系统提示均采用中文，并在用户错误操作时提示用户更正错误，并对用户文件和系统文件施加必要的保护措施。

（5）对所有按指定要求命中的文件均能自动在屏幕上显示或打印，并能随时“暂停”或“继续”。其中打印格式分为“文件编目表”，“文件目录表”等若干种不同格式。

（6）具有对某一指定文件记录或某一类型文件记录的各项信息进行修改的功能。

（7）具有对某一指定文件记录或某一类型文件记录进行删除的功能。并为慎重起见，先把查找到的文件记录显示出来，不作真正删除，在用户认可之后再用人机会话方式决定是否真正删除该记录。

（8）具有保密措施，允许用户自行设置密码和修改密码。

（9）对所有档案材料能进行自动统计。

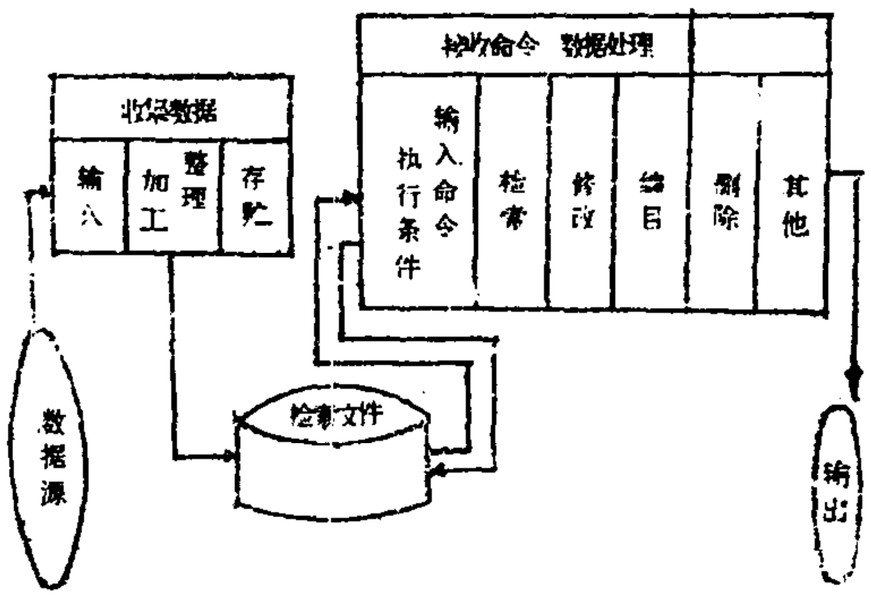

图 1　ZDGX系统的功能示意图

三、ZDGX系统的设计与实现

档案管理的大量工作主要花费在档案检索上，因而档案管理系统质量的优劣，效率的高