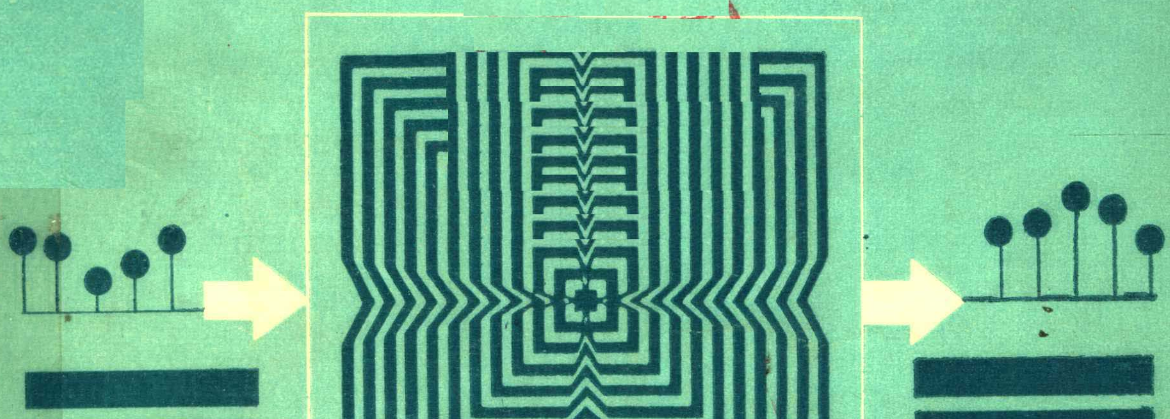
数字信号处理实验

软件部分

江大辉 编



华中工学院出版社

数字信号处理实验

华中工学院出版社

内 容 简 介

本书共编写了十一个实验，包括四个方面的内容：数字系统的单位冲激响应、系统函数和差分方程，快速傅里叶变换（FFT）及其应用，数字滤波器的设计，平稳随机过程的相关函数和功率谱估值的计算。每个实验一般包括实验目的、实验原理、实验程序、实验步骤（或例子）、实验前准备、实验报告要求和思考题等部分。程序用BASIC语言或FORTRAN语言写成，在IBM-PC机上通过。

本书可作为无线电技术、通信、雷达、自控、计算机、仪表、信息工程和其他有关专业的实验教材，也可供有关工程技术人员参考。

数字信号处理实验（软件部分）

江太辉　编

责任编辑　朱洪

●

华中工学院出版社出版发行

（武昌喻家山）

新华书店湖北发行所经销

华中工学院出版社沔阳印刷厂印刷

●

开本：787×1092　1/16　印张：7.75　插页：1　字数：160 000

1988年2月第1版，1988年2月第1次印刷

印数：1—2 000

ISBN 7-5609-0148-4/TN·4

定价：1.40元

前　言

数字信号处理是一门新兴学科，近十几年来，这个学科在我国发展迅速，应用十分广泛。由于信号的数字处理与模拟处理相比较，具有更多的优点，因此随着数字电子计算机技术的进步，人们越来越重视这个学科的发展。目前，国内已有许多院校开设了数字信号处理课程。为了加强学生的感性认识，使理论与实际结合，我们配合这门课程的教学内容，自1983年起，开设了数字信号处理实验课，并编写了实验讲义，本书就是在原来实验讲义的基础上，经过充实和提高写成的。

本书一共编写了十一个实验，包括四个方面的内容。第一，数字系统的单位冲激响应、系统函数和差分方程，即实验一；第二，快速傅里叶变换（FFT）及其应用，包括实验二、三、四、五、六，实验二和三分别介绍时间抽选和频率抽选的FFT算法，实验四、五、六分别介绍FFT在信号谱分析、计算线性卷积和相关函数中的应用；第三，数字滤波器的设计，包括实验七、八和九，实验七介绍IIR数字滤波器的设计，包括数字巴特沃斯和切比雪夫滤波器的设计，实验八介绍设计数字巴特沃斯低通、高通和带通滤波器的一些程序，实验九介绍用窗函数设计FIR数字滤波器；第四，平稳随机过程的相关函数和功率谱估值的计算，包括实验十、十一，实验十介绍用FFT计算平稳随机信号的相关函数及其功率谱估值的方法，实验十一介绍功率谱估计的周期图方法。任课教师可根据实际情况，在这些实验中选择若干个必做的实验，但建议以FFT和数字滤波器为重点进行选择。

每个实验一般包括实验目的、实验原理、实验程序、实验步骤（或实验例子）、实验前准备、实验报告要求和思考题等部分。实验原理只是课堂内容的概括，未作详细论述，必要时可参阅书中指出的文献。对实验报告，任课教师可根据实际情况提出自己的具体要求。

本书的程序用BASIC语言或FORTRAN语言写成。全部程序都在IBM-PC微型计算机上通过。对其它类型的微计算机，根据实际情况，对用FORTRAN语言编写的程序稍作修改就可使用。例如，IBM-PC微型机的输入输出设备号为“*”，对其它微型机，应把它修改为相应的设备号。

本书可作为无线电技术、通信和信息工程及其它有关专业本科生的实验教材，也可供研究生参考。每个实验的时间可定为2.5～3小时。较短的实验程序可由学生在实验时从键盘输入机器，然后调试运行。较长的程序可先复制在软盘上，实验时由学生调入机器。

姚天任副教授对本书进行了全面审阅，提出了许多宝贵意见，对此表示衷心感谢。华中工学院出版社的许多同志为本书的出版付出了辛勤的劳动，在此一并致谢。

由于水平所限，时间仓促，书中难免有错误或不妥之处，敬请读者批评指正。

作　者

1987.4

目　录

实验一　系统的单位冲激响应

（一）实验目的

1．学习使用计算机求解差分方程的方法，加强学生关于系统单位冲激响应的概念；

2．学习编写求解差分方程的BASIC程序的方法；

3．学习求解系统对输入的响应的方法。

（二）实验原理和实验程序

一个线性非移变系统可用图1.1所示的框图表示。图中$x(n)$表示输入信号，$T[\ ]$表示变换函数。于是有

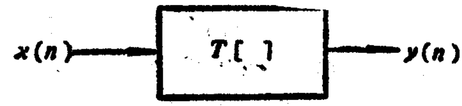


图1.1　系统的框图表示

$$y(n)=T[x(n)] \tag{1.1}$$

当输入信号$x(n)$为单位取样序列$\delta(n)$时，输出$y(n)=h(n)$，并称为系统的单位取样响应，或称为单位冲激响应，可表示为

$$h(n)=T[\delta(n)] \tag{1.2}$$

我们知道，一个因果的线性非移变系统可用一个常系数线性差分方程来描述，其一般形式为

$$\sum_{k=0}^{N}a_k y(n-k)=\sum_{r=0}^{M}b_r x(n-r) \tag{1.3}$$

若$M=N$，则式(1.3)可表示为

$$\sum_{k=0}^{N}a_k y(n-k)=\sum_{k=0}^{N}b_k x(n-k) \tag{1.4}$$

将式(1.4)展开后得

$$\begin{aligned}a_0 y(n)=&b_0x(n)+b_1x(n-1)+b_2x(n-2)+\cdots+b_Nx(n-N)-\\&a_1y(n-1)-a_2y(n-2)-\cdots-a_Ny(n-N)\end{aligned} \tag{1.5}$$

如果给定如下条件：

差分方程的阶次N，

输入序列通项$x(n)=f(h)$，

初始条件，$y(j)$，$j=0,1,2,\cdots,N-1$，

系数序列，a_i，b_i，$i=0,1,2,\cdots,N-1$，

则可求解输出序列的有限项$y(0)$，$y(1)$，$y(2)$，…，$y(S)$。其中，S为有限值。采用递推法，可以对式(1.5)编写程序并用计算机求解输出的有限项。程序1.1就是一个用BASIC语言编写的求解差分方程的一般程序。

下面给出一个计算系统单位冲激响应的实例。设一个因果的线性非移变系统的系统函数如下式所示，

$$H(z)=\frac{0.0065695(1+z^{-1})^4}{(1-1.24258z^{-1}+0.60889z^{-2})(1-0.9733z^{-1}+0.26034z^{-2})} \quad (1.6)$$

由以上系统函数求得表示该系统的差分方程为

$$\begin{aligned} y(n)=&0.0065695x(n)+0.026278x(n-1)+0.039417x(n-2)\\ &+0.026278x(n-3)+0.0065695x(n-4)+2.21588y(n-1)\\ &-2.07863y(n-2)+0.91612y(n-3)-0.15852y(n-4) \end{aligned} \quad (1.7)$$

这是一个四阶差分方程。为了求出系统的单位冲激响应的前33项，我们给出如下条件：

$N=4$，$S=32$

$$f(n)=\delta(n)=\begin{cases}1 & n=0\\ 0 & \text{其它}\end{cases}$$

$y(0)=0.0065695$，$y(1)=0.040835$，$y(2)=0.116246$，$y(3)=0.205002$

$a_0=1$，$a_1=-2.21588$，$a_2=2.07863$，$a_3=-0.91612$，$a_4=0.15852$，$b_0=0.0065695$，

$b_1=0.026278$，$b_2=0.039417$，$b_3=0.026278$，$b_4=0.0065695$。

根据式(1.7)和这些条件，并参照程序1.1，我们可以编写出求系统单位冲激响应的BASIC程序，如程序1.2所示，程序1.2是我们要用的实验程序。在这个程序中，输入的数据有系统的阶数N和系统单位取样响应的项数S，用键盘输入语句（INPUT语句）输入这两个变量。程序运行后，首先要求我们输入数据N和S。程序使用READ读数语句和DATA置数语句对系数变量$a_i(i=0,1,\cdots,N-1)$进行赋值。DATA语句放在本程序的最末一行，数据的排列顺序是先系数a_i，然后b_i，最后四个数据为初值$y(i)$。输入函数$\delta(n)$利用函数$\cos(x)$和正负函数SGN(x)的特性来实现。

（三）实验前准备和要求

1．实验前应读懂程序1.1和程序1.2。

2．求下列差分方程的单位阶跃响应的前11项，首先用递推法笔算其结果，然后参照程序1.1和程序1.2自己用BASIC语言编写程序，并上机验证。

$$y(n)=x(n)+5y(n-1)-6y(n-2) \quad (1.8)$$

其中$x(n)=u(n)$，初始条件为$y(0)=0$，$y(1)=1$。对本例，$N=2$，$S=10$。

（四）实验步骤及结果

1．按规定的步骤启动计算机的外部设备和主机。

2．首先敲入程序1.2，并检查程序是否有错误，然后执行该程序，如果程序无错误，则在屏幕上出现$N,S=4,32$和?号，这时可键入4和32，再按回车键。程序运行后，打印机将打印出系统冲激响应$h(n)$的前33项及其图形，如图1.2所示。

3．键入自己编写的求解式(1.8)的差分方程的单位阶跃响应的程序，运行该程序并将结果与笔算的结果相比较。

注意，在调试程序时，为了节约打印纸，可将程序中的LPRINT语句去掉，LPRINT语句是命令打印机打印的语句，程序调试完毕后再把LPRINT语句加上去。

（五）实验报告要求

1．说明用计算机求解差分方程的原理；

2．自己编写的求系统单位阶跃响应的程序；

3．运行程序1.2和自己编写的程序的运行结果；

4．调试程序及实验的心得体会。

（六）思考题

1．什么叫IIR和FIR数字滤波器？如果用差分方程来描述它们，在结构上有何区 别？

2．当系统的状态不为零时，能否用卷积和，即用$y(n)=x(n)*h(n)$来求解系统 的 响应？为什么？

3．如何利用常系数线性齐次差分方程的特征根判断系统的稳定性？

程序1.1 求解差分方程的BASIC程序

```
10  PRINT"N, S="
20  INPUT N, S
30  PRINT
40  DIM  A(N+1), B(N+1), Y(S+1)
50  FOR  I=0  TO N
60  READ  A(I), B(I)
70  DATA  1, b0, a1, b1, . . . , aN, bN
80  NEXT  I
90  DEF  FNX(H)=f(h)
100 FOR  J=0 TO  N-1
110 READ  Y(J)
120 DATA  Y(0), Y(1), Y(2), . . . , Y(N-1)
130 PRINT "Y("; J; ")="; Y(J)
140 NEXT  J
150 FOR  J=N  TO S
160 Y(J)=B(0)*FNX(J)
170 FOR  T=1  TO N
180 Y(J)=Y(J)+B(T)*FNX(J-T)-A(T)*Y(J-T)
190 NEXT    T
200 PRINT"Y("; J; ")="; Y(J)
210 NEXT  J
220 END
```

程序1.2 求系统单位冲激响应的BASIC程序

```
10  PRINT"N, S=4,32"
20  LPRINT"N, S=4, 32"
30  INPUT  N, S
40  DIM  A(N+1), B(N+1), Y (S+1)
50  FOR  I=0  TO N
60  READ  A(I), B(I)
70  NEXT  I
80  FOR  J=0  TO N-1
90  READ  Y(J)
100 NEXT  J
110 FOR  J=N  TO S
120 Y(J)=B(0)*COS(SGN(J)*3.14159/2)
130 FOR  I=1  TO  N
140 Y(J)=Y(J)+B(T)*COS(SGN(J-T)*3.14159/2)-A(T)*Y(J-T)
150 NEXT    T
160 NEXT    J
170 FOR  J=0  TO  S
```

```
180  PRINT"Y("; J; ")="; Y(J);
190  LPRINT"Y("; J; ")="; Y(J);
200  IF  Y(J)<0  THEN  260
210  FOR  K=30  TO  30+100*Y(J)
220  PRINT  TAB(K); "*";
230  LPRINT  TAB(K); "*";
240  NEXT  K
250  GOTO  300
260  FOR  B=30+100*Y(J)  TO  30
270  PRINT  TAB(B); "*";
280  LPRINT  TAB(B); "*";
290  NEXT  B
300  PRINT
310  LPRINT
320  NEXT  J
330  END
340  DATA  1, 0.0065695, -2.21588, 0.026278, 2.07863, 0.039417, -0.91612, 0.026278, 0.15852, 0.0065695,
     0.0065695, 0.040835, 0.116246, 0.205002
   N,S=4,32
   Y( 0 )=  .0065695               *
   Y( 1 )=  .040835                * * * * *
   Y( 2 )=  .116246                * * * * * * * * * * * *
   Y( 3 )=  .205002                * * * * * * * * * * * * * * * * * * * * *
   Y( 4 )=  .2555654               * * * * * * * * * * * * * * * * * * * * * * * * * *
   Y( 5 )=  .2402012               * * * * * * * * * * * * * * * * * * * * * * * * *
   Y( 6 )=  .1704104               * * * * * * * * * * * * * * * * * *
   Y( 7 )=  7.995138E-02           * * * * * * * *
   Y( 8 )=  2.483472E-03           *
   Y( 9 )=-4.264642E-02            * * * * *
   Y( 10 )=-5.342982E-02           * * * * * *
   Y( 11 )=-4.014654E-02           * * * * *
   Y( 12 )=-1.736187E-02           * * *
   Y( 13 )=  2.790302E-03          *
   Y( 14 )=  1.396265E-02          * *
   Y( 15 )=  1.559816E-02          * *
   Y( 16 )=  1.084905E-02          * *
   Y( 17 )=  3.966669E-03          *
   Y( 18 )=-1.68494E-03            *
   Y( 19 )=-4.512316E-03           *
   Y( 20 )=-4.582097E-03           *
   Y( 21 )=-2.946211E-03           *
   Y( 22 )=-8.705588E-04           *
   Y( 23 )=  7.127041E-04          *
   Y( 24 )=  1.416241E-03          *
   Y( 25 )=  1.326401E-03          *
   Y( 26 )=  7.863627E-04          *
   Y( 27 )=  1.699893E-40          *
```

Y(28)=−2.671076E−04
Y(29)=−4.349486E−04
Y(30)=−3.773663E−04
Y(31)=−2.036174E−04
Y(32)=−2.277661E−05

图1.2 系统的单位冲激响应

实验二　时间抽选的FFT算法

（一）实验目的

1．进一步掌握时间抽选FFT算法及其流图的特点；

2．学习根据FFT流图编写FFT程序的方法；

3．通过实验，掌握实数序列的FFT系数的实部和虚部的特点。

（二）实验原理

本实验采用库利-图基最初提出的时间抽选（decimation-in-time）的FFT流图（$N=8$），如图2.1所示。现在我们来分析流图的一些特点，以便读者看懂FFT的BASIC程序。

如图2.1所示的时间抽选FFT流图具有以下的一些特点：

1．正序输入，混序（码位倒置）输出。

2．每个节点只能有两个输入信号，其中一个乘以1，故通过一次复乘法运算和一次复加法运算就能得到一个节点的信号。

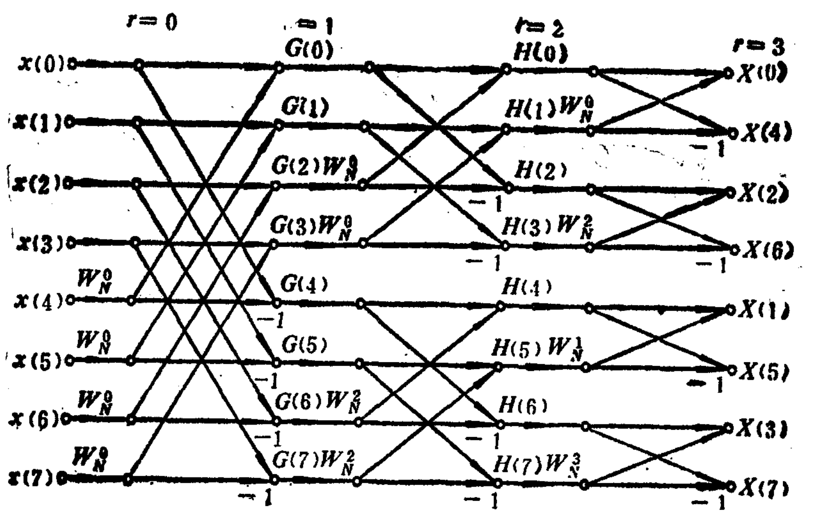


图2.1　时间抽选的FFT流图（$N=8$）

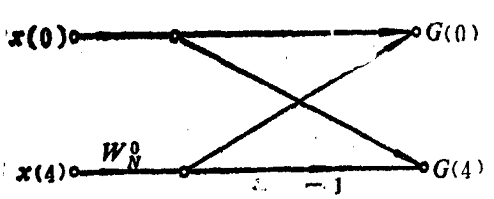


图　2.2　基本蝶形运算单元

3．基本运算为蝶形运算，即四点组成一组对偶节点对。如图2.1所示，$G(0)$和$G(4)$是从$x(0)$和$x(4)$计算得到的，而且$x(0)$和$x(4)$与其它中间节点无关，所以$x(0)$和$x(4)$构成对偶节点对，$G(0)$和$G(4)$也是对偶节点对，由这些对偶节点构成了一个基本蝶形单元，如图2.2所示。

4．蝶形运算具有同址（原位）计算的特点，也就是由$x(0)$和$x(4)$计算出$G(0)$和$G(4)$之后，无需再保存，而将结果立即送回原来存贮$x(0)$和$x(4)$的单元。这样，从输入到输出

只要一列存贮单元就够了，于是可大大减少存贮单元。

5．设取样点数$N=2^r$，从输入信号数据序列$x(n)$到输出序列$X(k)$，要经过r次迭代运算，例如$N=8=2^3$，则$r=3$，即要经过3次迭代运算。显然，$r=\log_2 N$。

6．对偶节点的距离是有规律的。以图2.1为例，一般对偶节点对的距离等于$N/2^r$，$r=1,2,\cdots,\log_2 N$。例如，$G(n)$中的对偶节点间的距离为$8/2^1=4$，而$H(n)$中的对偶节点间的距离为$8/2^2=2$。

7．对偶节点的计算（即蝶形计算）。计算一对对偶节点只需一次复乘法和两次复加法，从图2.1中找出任意一对对偶节点，如$H(4)$和$H(6)$，见图2.3。从图中可得

$$\begin{cases} H(4)=G(4)+W_N^2 G(6) \\ H(6)=G(4)-W_N^2 G(6) \end{cases}$$

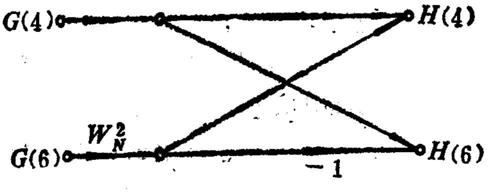


图2.3 蝶形运算

8．变址。从图2.1看出，以正序（自然顺序）输入的数据经迭代计算后输出的结果是"混序"（码位倒置）的，因而对输出要进行整序（变址），最后得到正序的结果，以便打印机打印。

（三）实验程序

根据图2.1所示的FFT流图，用BASIC语言编写的FFT程序如程序2.1所示，其框图见图2.4。如果在程序2.1中增加一些打印语句和实际数据输入语句便变成实验程序2.2。在程序2.2中，变换序列$x(n)$的数据是实验一中的系统冲激响应的数据序列，因为实验一中的系统冲激响应序列为实数序列，所以在使用读数语句READ和置数语句DATA时，输入序列的虚部为0，见程序2.2的720～810行。本程序用键盘输入语句INPUT来向变量R提供数据，见程序2.2的第50行。R为FFT流图的迭代级数，$R=\log_2 N$，N为变换序列$x(n)$的点数，若$N=32$，则$R=\log_2 32=5$。这样，在程序开始运行后，要求从键盘输入R的值。

（四）实验前准备及要求

1．实验前根据给出的FFT流图（图2.1），阅读程序2.1，在图2.4所示框图的括号里填入相应操作的起始语句标号；

2．读懂实验程序2.2；

3．实验者手算一个$N=4$或$N=8$的DFT例题，以备上机实验。下面给出几个序列，实验者可以自选一两题求其DFT，然后上机验证；

（1）$x(n)=\delta(n)$ $\quad 0\leqslant n\leqslant N-1$

（2）$x(n)=\delta(n-n_0)$ $\quad 0<n_0<N$

（3）$x(n)=a^n$，$a=0.8$ $\quad 0\leqslant n\leqslant N-1$

（4）$x(n)=u(n)$ $\quad 0\leqslant n\leqslant N-1$

（5）$x(n)=\sin(\omega n)$ $\quad 0\leqslant n\leqslant N-1$

4．将程序2.2改为求IFFT的程序，然后将以上例题所得到的FFT数据做为输入，求其IFFT，验证得到的结果是否是原来的时间序列。

（五）实验步骤

1．键入程序2.2，并检查和改错，然后运行该程序。程序运行后，屏幕上将出现"？"号，这时可键入R的值，对本实验，键入5，然后按下回车键，约一分钟后，打印机开始打印结果。如果程序运行正常，则打印机打印的结果如表2.1和图2.5所示。表2.1所示的是

FFT系数$X(k)$的实部$R(k)$和虚部$I(k)$，图2.5所示的是$|X(k)|$的数值及其示意图形,

2．验证实验者手算的DFT例题。

3．将FFT程序修改为IFFT程序，然后进行验证。

（六）实验报告要求

1．在图2.4所示框图的括号里填入相应操作的起始语句标号。

2．说明时间抽选的FFT流图的特点。

3．指明将程序2.2修改为IFFT程序的语句。

4．运行程序2.2和验证自己手算DFT例题的结果，运行IFFT程序的结果。

5．回答思考题中提出的问题。

（七）思考题

1．实数序列的FFT的实部和虚部有何性质？

2．在实验中，对一个实数序列求FFT，你如何判断结果正确与否？

程序2.1　时间抽选FFT算法的BASIC程序

```
10  PRINT "FAST FOURIER TRASFORM"
20  PRINT "R="
30  INPUT  R
40  N=2↑R
 ⋮  (输入N个数据，实部为TR(k),
 ⋮  虚部为TI(k)，k=0，1，…，N-1)
510 FOR  L=1  TO  R
520 N1=2↑(R-L)
530 P=0
540 G=P
550 GOSUB  1000
560 A=6.28318531*D/N
570 A1=COS(A)
580 A2=SIN(A)
590 FOR  K=P*N1  TO  (P+1)*N1-1
600 TR1=TR(K+N1)*A1+TI(K+N1)*A2
610 TI1=TI(K+N1)*A1-TR(K+N1)*A2
620 TR(K+N1)=TR(K)-TR1
630 TI(K+N1)=TI(K)-TI1
640 TR(K)=TR(K)+TR1
650 TI(K)=TI(K)+TI1
660 NEXT  K
670 P=P+2
680 IF  (P+1)*N1<N GOTO 540
690 NEXT L
700 FOR K=0  TO N-1
710 G=K
720 GOSUB  1000
730 IF  K>=D  GOTO  800
740 TR1=TR(K)
750 TI1=TI(K)
760 TR(K)=TR(D)
```

```
770   TI(K)=TI(D)
780   TR(D)=TR1
790   TI(D)=TI1
800   NEXT K
810   FOR  K=0  TO  N-1
820   PRINT  K, TR(K), TI(K), SQR(TR(K)*TR(K)+TI(K)*TI(K))
830   NEXT  K
840   END
1000  D=0
1010  FOR  Q=1  TO  R
1020  C=INT(G/2)
1030  D=2*D+G-2*C
1040  G=C
1050  NEXT  Q
1060  RETURT
```

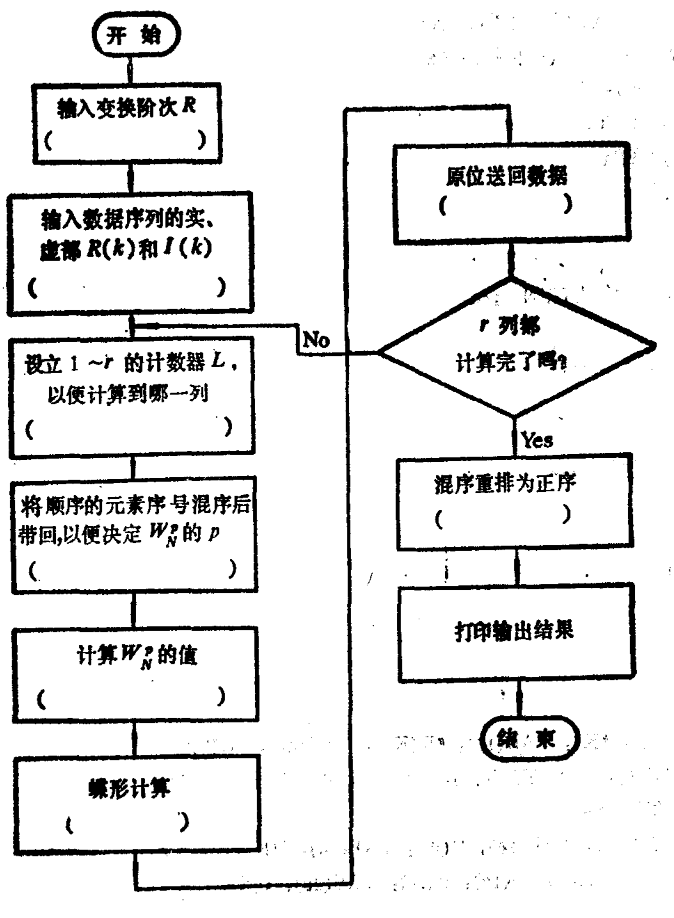


图2.4　程序2.1的框图表示

程序2.2　FFT实验程序

```
10    PRINT "FFT"
20    LPRINT "FFT"
30    PRINT "R="
```

```
40  LPRINT "R="
50  INPUT R
60  N=2↑R
70  DIM R(N), I(N), X(N)
80  FOR K=0 TO N-1
90  READ R(K), I(K)
100 NEXT K
110 FOR L=1 TO R
120 N1=2↑(R-L)
130 P=0
140 G=P
150 GOSUB 650
160 A=6.2831853*D/N
170 A1=COS(A)
180 A2=SIN(A)
190 FOR K=P*N1 TO (P+1)*N1-1
200 TR1=R(K+N1)*A1+I(K+N1)*A2
210 TI1=I(K+N1)*A1-R(K+N1)*A2
220 R(K+N1)=R(K)-TR1
230 I(K+N1)=I(K)-TI1
240 R(K)=R(K)+TR1
250 I(K)=I(K)+TI1
260 NEXT K
270 P=P+2
280 IF (P+1)*N1<N THEN 140
290 NEXT L
300 FOR K=0 TO N-1
310 G=K
320 GOSUB 650
330 IF K>=D THEN 400
340 TR1=R(K)
350 TI1=I(K)
360 R(K)=R(D)
370 I(K)=I(D)
380 R(D)=TR1
390 I(D)=TI1
400 NEXT K
410 PRINT TAB(5); "K"; TAB(15); "R(K)"; TAB(30); "I(K)"
420 LPRINT TAB(5); "K"; TAB(15); "R(K)"; TAB(30); "I(K)"
430 FOR K=0 TO N-1
440 PRINT TAB(4); K; TAB(10); R(K); TAB(26); I(K)
450 LPRINT TAB(4); K; TAB(10); R(K); TAB(26); I(K)
460 NEXT K
470 FOR I=0 TO 4
480 PRINT
490 LPRINT
500 NEXT I
510 FOR K=0 TO N-1
```

```
520  X(K)=SQR(R(K)*R(K)+I(K)*I(K))
530  PRINT "X("; K; ")="; X(K);
540  LPRINT "X("; K; ")="; X(K);
550  PRINT  TAB(25);
560  LPRINT  TAB(25);
570  FOR  B=1  TO  30*X(K)
580  PRINT "-";
590  LPRINT "-";
600  NEXT  B
610  PRINT "*"
620  LPRINT "*"
630  NEXT  K
640  END
650  D=0
660  FOR  Q=1  TO  R
670  C=INT (G/2)
680  D=2*D+G-2*C
690  G=C
700  NEXT  Q
710  RETURN
720  DATA  6.5695E-03, 0,0.040835, 0, 0.116246, 0, 0.205002,0
730  DATA  0.255565, 0, 0.240201, 0, 0. 170411, 0,0.0799519, 0
740  DATA  2.48406E-03, 0, -0.0426459, 0, -0.0534294, 0
750  DATA  -0.0401463, 0, -0.0173618, 0, 2.79031E-03, 0
760  DATA  0.0139627, 0, 0.0155983, 0, 0.0108493, 0, 3.96698E-03,
770  DATA  -1.68461E-03, 0, -4.51202E-03, 0, -4.58184E-03,0
780  DATA  -2.94599E-03, 0, -8.70364E-04, 0, 7.12896E-04,0
790  DATA  1.41644E-03, 0, 1.32662E-03, 0, 7.86598E-04,0
800  DATA  1.70233E-04, 0, -2.68864E-04, 0, -4.34709E-04,0
810  DATA  -3.77133E-04, 0, -2.0339E-04,0
```

```
X( 0 )=.9993855              ..............................................................*
X( 1 )=.9994059              ..............................................................*
X( 2 )=.9951589              ..............................................................*
X( 3 )=.8895635              .....................................................*
X( 4 )=.4890096              ...................................*
X( 5 )=.198225               ...........*
X( 6 )=8.257939E-02          .....*
X( 7 )=3.642482E-02          ...*
X( 8 )=1.656611E-02          *
X( 9 )=7.54827E-03           *
X( 10 )=3.34908E-03          *
X( 11 )=1.404117E-03         *
X( 12 )=5.407643E-04         *
X( 13 )=1.930703E-04         *
X( 14 )=7.839711E-05         *
X( 15 )=5.328709E-05         *
X( 16 )=5.164743E-05         *
X( 17 )=5.334657E-05         *
```

```
X( 18 )=7.834681E-05          *
X( 19 )=1.931055E-04          *
X( 20 )=5.407643E-04          *
X( 21 )=1.404123E-03          *
X( 22 )=3.349085E-03          *
X( 23 )=7.548324E-03          *
X( 24 )=1.656611E-02          *
X( 25 )=3.642485E-02          ...*
X( 26 )=8.257941E-02          ......*
X( 27 )=.198225               ................*
X( 28 )=.4890006              ............................................*
X( 29 )=.8895634              ..................................................................................*
X( 30 )=.995159               ...........................................................................................*
X( 31 )=.9994059              ...........................................................................................*
```

图2.5　$X(k)$的模$|X(k)|$及其示意图