

双色球 Fortran

全攻略

——红球组号方法及程序

严谨的算法 完备的数据 一学就会一看就懂

任安禄 / 著

SHUANGSEQIU FORTRAN
QUANGONGLUE

提供
程序源代码
免费下载
服务



中国经济出版社
CHINA ECONOMIC PUBLISHING HOUSE

双色球 Fortran 全攻略

——红球组号方法及程序

任安禄 / 著

SHUANGSEQIU FORTRAN
QUANGONGLUE



中国经济出版社
CHINA ECONOMIC PUBLISHING HOUSE

· 北 京 ·

图书在版编目 (CIP) 数据

双色球 Fortran 全攻略——红球组号方法及程序 / 任安禄著.

北京: 中国经济出版社, 2014. 2

ISBN 978 - 7 - 5136 - 2740 - 5

I. ①双… II. ①任… III. ①社会福利—彩票—基本知识—中国 IV. ①F832.5

中国版本图书馆 CIP 数据核字 (2013) 第 207951 号

责任编辑 张玲玲

责任审读 贺 静

责任印制 马小宾

封面设计 兆天书装

出版发行 中国经济出版社

印 刷 者 北京科信印刷有限公司

经 销 者 各地新华书店

开 本 710mm × 1000mm 1/16

印 张 13.5

字 数 192 千字

版 次 2014 年 2 月第 1 版

印 次 2014 年 2 月第 1 次

书 号 ISBN 978 - 7 - 5136 - 2740 - 5

定 价 35.00 元

中国经济出版社 网址 [www. economyph. com](http://www.economyph.com) 社址 北京市西城区百万庄北街 3 号 邮编 100037

本版图书如存在印装质量问题, 请与本社发行中心联系调换 (联系电话: 010 - 68319116)

版权所有 盗版必究 (举报电话: 010 - 68359418 010 - 68319282)

国家版权局反盗版举报中心 (举报电话: 12390) 服务热线: 010 - 68344225 88386794

| 前 言 |

目前,国内有许多双色球选号、投注方法介绍的书籍,也有许多彩票软件供各位彩民玩家选号组号,但综观这些著作和软件,都缺少组号方法介绍,而专门性的教材、课本也很少关注具体的组号方法,有的只有极其简单的介绍。我根据自己的知识结构和程序能力,编写了本书,既可供彩民玩家参考,也可供工程、物流、国防工业中需要用到组合程序编写的科技工作者参考,对于其他乐透型彩票编程也具有直接的参考价值。本书中介绍的所有方法、程序,如组号程序、缩水程序,以及和值、散度、偏度、AC 值等分析程序及旋转矩阵程序全部由本人编写,并经过计算验证。读者可从中国经济出版社网站“中经理财咨询台”下载。

33 个红球取 6 个红球组成一个投注号码是一个组合问题,共有 1107568 组不同的组合,即 $C_{33}^6 = \frac{33!}{6! 27!} = \frac{33 * 32 * 31 * 30 * 29 * 28}{6!}$,但如何把这 100 多万的投注号码书写出来,而且不重复不遗漏,却不是一件简单的事情。本书就是独立完成了这个任务,而且经过各种数码机会均等性的验证,各个号码在组号程序中出现的频率与理论频率一致。

本书采用的组号方法是逐次分解法,就是利用组合公式 $C_{m+n}^k = \sum_{j=0}^k C_m^j C_n^{k-j}$ 。但在每次分解时, m 、 n 各是多少,这是有讲究的,并不是每种分解均可得到完美的结果,我是经过反复试验才得到经验的。虽然我不能从理论上证明哪一种组号的分解方法是最好的,但却找到了一些分解方法,使得我得到了均衡完美的分解组号方法,并完成了所有组合程序。我感到, m 、 n 要比较平衡地分配,才可以收到较好的效果。例如,取 $m=16$, $n=17$ 来分解组号,它们可被看为 16 个偶数和 17 个奇数的分解组号,也可以看作 16 个小数和 17 个大数的组号。更进一步,可取 1 偶 5 奇、2 偶 4 奇、3 偶 3 奇、4 偶 2

奇、5 偶 1 奇来组号,或 1 小 5 大、2 小 4 大、3 小 3 大、4 小 2 大、5 小 1 大来组号。这些组号安排正好与彩票玩家对开奖号码的分析用语一致。

当然,本书也有 16 个全偶或全小数组号、17 个全奇数或全大数的组号。本书中的组号方法,只作为模板数据,读者可以任取 16 个数码或 17 个数码来组号,只要改写输入数据即可。

对 16 个或 17 个数码直接进行组号还是太大,容易出错,要进一步对 16 个或 17 个数码进行分解,可再分解为 8+8 个或 8+9 个来组号,依此类推。

在本书编写过程中,我编写了奇偶组合程序,包括全偶 TIC166 程序、1 偶 5 奇程序 TIC161175、2 偶 4 奇程序 TIC162174、3 偶 3 奇程序 TIC163173、4 偶 2 奇程序 TIC164172、5 偶 1 奇程序 TIC165171 程序及全奇数程序 TIC176。如果把输入数据 lev、lod 改为 Isl、lbg 数据,则可得到全小数 TIC166 程序、1 小 5 大程序 TIC161175、2 小 4 大程序 TIC162174、3 小 3 大程序 TIC163173、4 小 2 大程序 TIC164172、5 小 1 大程序 TIC165171 程序及全大数程序 TIC176。在程序编写时把小数程序放前面,是由开奖号码数据存放时总把小数放在前面这个习惯而来的,这样在大小混合程序时可不必重排序号。对于奇偶程序,也把偶数放前面,经过重排程序完成大小排号。

考虑到篇幅不宜太大,又因为奇偶程序与大小数程序编制方法及程序架构完全一样,只是输入输出数据不一样,因此先详细介绍奇偶程序 TIC166、TIC176,包括主程序与子程序,然后简单介绍五偶 TIC165、四偶 TIC164、三偶 TIC163,以及五奇 TIC175、四奇 TIC174、三奇 TIC173 程序。这些程序只有主程序,子程序可用前面已介绍的子程序适当改变而得,具体讲,就是改变填号位置。在中国经济出版社网站也可查到相应的子程序源码。

对大小数组合,我在书中也给出了主程序:全小数程序 TIC166、全大数程序 TIC176、五小数组号程序 TIC165、四小数组号程序 TIC164、三小数组号程序 TIC163;五大数程序 TIC175、四大数程序 TIC174、三大数程序 TIC173,等等。

小大组合与偶奇组合方法完全一样。对混合组合 TIC161175、TIC162174、TIC163173、TIC164172、TIC165171,既有偶奇组合,又有小大数组合,以 3 偶 3 奇 TIC163173 程序为例来构建组合模块。在每个程序完成后尚有许多辅助程序,如重新排序程序、分析和值 AC 值程序,还需进一步分割

AC 值与和值。本书以 3 偶 3 奇程序 TIC163173 为例进行辅助程序的组成并给出运算结果。

全部程序运算得到的输出结果可在中国经济出版社网站“中经理财咨询台”下载 (www.economyph.com), 部分关键数据已在本书后面的表格中给出。下载文件中所给出的每六点和值在大小、奇偶及各 AC 值下的输出数据和 012 路分析数据可立即用于有关网站和投注站。

在本书和所附程序中安排了很多打印语句和打印文件, 可方便检查程序的执行情况和出错情况及位置。本书中所描述的逐步分解编程方法, 可推广到其他需要进行组合算法的地方。我的体会是, 用基本平衡的分解方法有利于得到机会均等的结果。

我的主要学术方向是流体力学, 并偏重于计算流体力学, 长期的职业训练使得我擅长于编程求解问题。本书虽然稍微偏离了主方向, 但也检验了我编写程序的能力。能够正式出版当然很高兴, 本书是我的收山之作, 在这里要感谢中国经济出版社和张玲玲主任对本书出版所付出的精力和爱心。本书在投稿一个月内即获得肯定出版的答复, 使我免于东奔西投地投寄书稿了。

我还要感谢我的爱人杨曼仙女士, 在日常家务劳动的操持中, 在日常家庭生活的互助中, 显示出她的价值和品德。

由于本书中所附程序是逐次完成的, 历时近两年, 书写用语和文件命名难免前后稍有差别, 说明语句有重复或疏漏, 敬请读者谅解指正。


 作者 ZHEJIANG UNIVERSITY

2013 年 8 月于
杭州浙江大学求是村家中

| 目 录 |

第一章 16 数码组号方法	1
1.1 全偶组号程序 TIC166	1
1.2 五偶组号程序 TIC165	28
1.3 四偶组号程序 TIC164	35
1.4 三偶组号程序 TIC163	41
1.5 6 个数码以下基本组号模板	47
第二章 17 数码组号方法	50
2.1 全奇组号程序 TIC176	50
2.2 五奇组号程序 TIC175	97
2.3 四奇组号程序 TIC174	104
2.4 三奇组号程序 TIC173	111
2.5 16 数码、17 数码组号均衡性验证	116
第三章 混合编排程序	121
3.1 三奇三偶混合程序 TIC163173	121
3.2 重新排序程序	122
第四章 辅助程序——和值、散度、偏度、AC 值计算及和值、AC 值分布计算	127
4.1 和值、AC 值的计算程序	127
4.2 用 AC 值分割 380800 总投注注数	129
4.3 和值、AC 值分布计算	131
4.4 中奖号码分析程序	137

4.5 投注号码的分布计算	153
第五章 旋转矩阵组号方法及程序	167
第六章 部分组号数据	175
6.1 AC = 0 组号数据	175
6.2 和值 97 ~ 102、103 ~ 108 范围内组号数据	183
6.3 程序中的符号含义	184
6.4 软件中的符号含义	184
6.5 计算数据	190
6.6 程序清单	192
6.7 程序使用介绍	196
参考文献	201

特别说明:本程序均在 Compag Visual Fortran 环境下,用 Dell 双核台式机完成。该电脑为 32 位字长,本软件也只适用于 32 位字长电脑。

| 表格索引 |

表 2-1	奇偶编号自查表	116
表 2-2	混合编号自查表	117
表 2-3	大小编号自查表	118
表 4-1	2013 年开奖号码数据分析结果	144
表 4-2	奇偶组号对各 AC 值所占组数分布	153
表 4-3	TIC163173(3 偶 3 奇)和值与 AC 值分布表	154
表 4-4	TIC162174(2 偶 4 奇)和值与 AC 值分布表	155
表 4-5	TIC161175(1 偶 5 奇)和值与 AC 值分布表	156
表 4-6	TIC164172(4 偶 2 奇)和值与 AC 值分布表	156
表 4-7	TIC165171(5 偶 1 奇)和值与 AC 值分布表	157
表 4-8	TIC166(6 偶)和值与 AC 值分布表	158
表 4-9	TIC176(6 奇)和值与 AC 值分布表	159
表 4-10	大小组合对各 AC 值所占组数分布	160
表 4-11	TIC163173(3 小 3 大)和值与 AC 值分布表	161
表 4-12	TIC162174(2 小 4 大)和值与 AC 值分布表	161
表 4-13	TIC161175(1 小 5 大)和值与 AC 值分布表	162
表 4-14	TIC164172(4 小 2 大)和值与 AC 值分布表	163
表 4-15	TIC165171(5 小 1 大)和值与 AC 值分布表	164
表 4-16	TIC166(6 小)和值与 AC 值分布表	164
表 4-17	TIC176(6 大)和值与 AC 值分布表	165
表 6-1	AC=0 等差级数组号	175
表 6-2	中间 12 点和值总注数、中间值 102 注数表	183
表 6-3	组号及分析程序名称表	192

第一章 16 数码组号方法

1.1 全偶组号程序 TIC166

按均衡分解的原则，把 16 个数码分为 8 个加 8 个来实施组号。

$$C(16, 6) = C(8, 6) + C(8, 5) * C(8, 1) + C(8, 4) * C(8, 2) + C(8, 3) * C(8, 3) + C(8, 2) * C(8, 4) + C(8, 1) * C(8, 5) + C(8, 6)$$

上述分解法用编号程序中的数组名表达如下，以便逐段编写：

$$\begin{aligned} IC166(8008, 123456)[1, 2, \dots, 16] &= IC86(28, 123456) + IC85(56, \\ 12345) * IC81_1(8, 6) &+ IC84(70, 1234) * IC82_1(28, 56) + IC83(56, \\ 123) * IC83_1(56, 456) &+ IC82(28, 12) * IC84_1(70, 3456) + IC81 \\ (8, 1) * IC85_1(56, 23456) &+ IC86_1(28, 123456)[9, \dots, 16] = 28 + 448 + \\ 1960 + 3136 + 1960 + 448 + 28 &= 8008 \end{aligned}$$

在上面多维整型变量 $IC ** (i, j)$ 中， i 表示变量组号组数， j 表示每一注数组号的投注号码。该变量是 6 维变量，对应各红球号码，组号时将数码放入对应的位置。如：1、2、3、4、5、6 表示组号时分别放入第 1、第 2……第 6 号红球的位置。

程序中的 $[1, 2, \dots, 16]$ 、 $[1, 2, \dots, 8]$ 、 $[9, 10, \dots, 16]$ 表示编程时的取数位置。

组号 IC166 可用于全偶数组号，也可用于小数 $[1, 2, \dots, 16]$ 组

号, 或者任意 16 个数的组号。

IC86 与 IC86_1 可编制子程序, 进一步分解为: IC86 (28, 123456) [1, 2, ..., 8] = IC66 (1, 123456) [1, 2, ..., 6] + IC65 (6, 12345) * IC21 (2, 6) + IC64 (15, 1234) * IC22 (3, 56) [7, 8]。

调用时, 只要改变取数位置及需填充的数码位置即可。同样, IC85、IC84、IC83 均可编制子程序, 以便调用。

用这样的方法详细演绎编号方法, 使得编程十分容易。逐条书写就是了, 全部程序约 1100 条语句。

全偶数组号 TIC166 主程序

```

Program TIC166
include 'COM_ CP16'
dimension IC86 (28, 6), IC85 (56, 6), IC84 (70, 6), IC83 (56, 6),
$ IC82 (28, 6), IC81 (8, 6), IC85_1 (56, 6), IC84_1 (70, 6),
$ IC83_1 (56, 6), IC82_1 (28, 6), IC81_1 (8, 6), IC86_1 (28, 6)
c 以下为说明语句, 一般置顶, 随时参考
C (16, 6) = C (8, 6) + C (8, 5) * C (8, 1) + C (8, 4) * C (8, 2) +
C (8, 3) * C (8, 3) + C (8, 2) * C (8, 4) + C (8, 1) * C (8, 5) + C
(8, 6) =
IC166 = IC86 (28, 123456) [1, 2, ..., 8] + IC85 (56, 12345) * IC81_1
(8, 6) + IC84 (70, 1234) * IC82_1 (28, 56) + IC83 (56, 123) * IC83_1
(56, 456) + IC82 (28, 12) * IC84_1 (70, 3456) + IC81 (8, 1) *
IC85_1 (56, 23456) [9, 10, ..., 16] + IC86_1 (28, 123456) [9, ..., 16]
= 28 + 56 * 8 + 70 * 28 + 56 * 56 + 28 * 70 + 8 * 56 + 28 = 28 + 448 + 1960 + 3136 +
1960 + 448 + 28 = 8008'
c 投注注数组成用局部分解法再分解
Open (5, file = 'lev.dat', status = 'old')
read (5, *) (lev (i), i = 1, 16)
close (5)
open (6, file = 'PrTIC166.dat')
do i = 1, 8
Inp (i) = lev (i)
end do
call SIC86
    
```

```

do i = 1, 28
  do k = 1, 6
    IC86 (i, k) = SIC (i, k)
  end do
end do
print *, 'first call IC86 '
write (6, *) 'first call IC86 '

do i = 9, 16
  Inp (i-8) = Iev (i)
end do
call SIC86
do i = 1, 28
  do k = 1, 6
    IC86_1 (i, k) = SIC (i, k)
  end do
end do
print *, 'Second call IC86 '
write (6, *) 'Second call IC86 '

do i = 1, 8
  Inp (i) = Iev (i)
end do

call SIC85
do i = 1, 56
  do k = 1, 5
    IC85 (i, k) = SIC (i, k)
  end do
end do

print *, 'IC85 first call'
write (6, *) 'IC85 first call'

open (10, file = 'IC85. dat')
write (10, '(514) ') ( (IC85 (i, j), j=1, 5), i=1, 56)
close (10)
do i = 9, 16
  Inp (i-8) = Iev (i)
end do

```

```

call SIC85
do i = 1, 56
    do k = 1, 5
        IC85_1 (i, k+1) = SIC (i, k)
    end do
end do

print *, 'IC85 second call'
write (6, *) 'IC85 second call'
open (10, file = 'IC85_1.dat')
write (10, '(5I4) ') ( (IC85_1 (i, j), j=2, 6), i=1, 56)
close (10)

do i = 1, 8
    Inp (i) = Iev (i)
end do
call SIC84
do i = 1, 70
    do k = 1, 4
        IC84 (i, k) = SIC (i, k)
    end do
end do

print *, 'first call IC84'
write (6, *) 'first call IC84'
open (10, file = 'IC84.dat')
write (10, '(4I4) ') ( (IC84 (i, j), j=1, 4), i=1, 70)
close (10)
c    IC82 (28, 12) * IC84_1 (70, 3456)
do i = 9, 16
    Inp (i-8) = Iev (i)
end do
call SIC84
do i = 1, 70
    do k = 1, 4
        IC84_1 (i, k+2) = SIC (i, k)
    end do
end do

print *, 'after second call IC84_1'
write (6, *) 'after second call IC84_1'

```

```

open (11, file = 'IC84_1.dat')
write (11, '(414)') ( (IC84_1 (i, j), j=3, 6), i=1, 70)
close (11)
c IC83 (56, 123) * IC83_1 (56, 456)
do i = 1, 8
  Inp (i) = Iev (i)
end do

call SIC83
print *, 'first call IC83'
write (6, *) 'first call IC83'
do i = 1, 56
  do k = 1, 3
    IC83 (i, k) = SIC (i, k)
  end do
end do

open (11, file = 'IC83.dat')
write (11, '(314)') ( (IC83 (i, j), j=1, 3), i=1, 56)
close (11)
do i = 9, 16
  Inp (i-8) = Iev (i)
end do

call SIC83
print *, 'Second call IC83'
write (6, *) 'Second call IC83'

do i = 1, 56
  do k = 1, 3
    IC83_1 (i, k+3) = SIC (i, k)
  end do
end do
open (11, file = 'IC83_1.dat')
write (11, '(314)') ( (IC83_1 (i, j), j=4, 6), i=1, 56)
close (11)
print *, 'after IC83_1'
write (6, *) 'after IC83_1'
c Make IC81
do i = 1, 8
  IC81 (i, 1) = Iev (i)

```

```

      end do
c    Make IC81_1 (8, 5)
      do i = 9, 16
        IC81_1 (i-8, 6) = Iev (i)
      end do
c    Make IC82 (28, 12) [1, ..., 8]
      il = 0
      do i = 2, 8
        il = il + 1
        IC82 (il, 1) = Iev (1)
        IC82 (il, 2) = Iev (i)
      end do
c 7 set
      do i = 3, 8
        il = il + 1
        IC82 (il, 1) = Iev (2)
        IC82 (il, 2) = Iev (i)
      end do
c 6 set 7+6=13
      do i = 4, 8
        il = il + 1
        IC82 (il, 1) = Iev (3)
        IC82 (il, 2) = Iev (i)
      end do
c 5 set 13+5=18
      do i = 5, 8
        il = il + 1
        IC82 (il, 1) = Iev (4)
        IC82 (il, 2) = Iev (i)
      end do
c 4 set 18+4=22
      do i = 6, 8
        il = il + 1
        IC82 (il, 1) = Iev (5)
        IC82 (il, 2) = Iev (i)
      end do
c 3 set 22+3=25
      do i = 7, 8
        il = il + 1
        IC82 (il, 1) = Iev (6)
        IC82 (il, 2) = Iev (i)

```

```

        end do
c 2 set 25 + 2 = 27
        il = il + 1
        IC82 (il, 1) = lev (7)
        IC82 (il, 2) = lev (8)
c 1 set 27 + 1 = 28
        print *, 'IC82 il = ', il
        write (6, *) 'IC82 il = ', il
        open (11, file = 'IC82. dat')
        write (11, '(2I4) ') ( (IC82 (i, j), j = 1, 2), i = 1, 28)
        close (11)
c      Make IC82_ 1 (28, 56) [9, 10, ..., 16]
        il = 0
        do i = 10, 16
            il = il + 1
            IC82_ 1 (il, 5) = lev (9)
            IC82_ 1 (il, 6) = lev (i)
        end do
c 7 set
        do i = 11, 16
            il = il + 1
            IC82_ 1 (il, 5) = lev (10)
            IC82_ 1 (il, 6) = lev (i)
        end do
c 6 set 7 + 6 = 13
        do i = 12, 16
            il = il + 1
            IC82_ 1 (il, 5) = lev (11)
            IC82_ 1 (il, 6) = lev (i)
        end do
c 5 set 13 + 5 = 18
        do i = 13, 16
            il = il + 1
            IC82_ 1 (il, 5) = lev (12)
            IC82_ 1 (il, 6) = lev (i)
        end do
c 4 set 18 + 4 = 22
        do i = 14, 16
            il = il + 1
            IC82_ 1 (il, 5) = lev (13)
            IC82_ 1 (il, 6) = lev (i)

```

```

        end do
c 3 set 22 + 3 = 25
        do i = 15, 16
            il = il + 1
            IC82_1 (il, 5) = lev (14)
            IC82_1 (il, 6) = lev (i)
        end do
c 2 set 25 + 2 = 27
        il = il + 1
        IC82_1 (il, 5) = lev (15)
        IC82_1 (il, 6) = lev (16)
c 1 set 27 + 1 = 28
        print *, 'IC82_1 il =', il
        write (6, *) 'IC82_1 il =', il
        open (11, file = 'IC82_1.dat')
        write (11, '(2I4) ') ( (IC82_1 (i, j), j=5, 6), i=1, 28)
        close (11)
        il = 0
        do i = 1, 28
            il = il + 1
            do k = 1, 6
                IC166 (il, k) = IC86 (i, k)
            end do
        end do
        print *, 'after IC86 il =', il
        write (6, *) 'after IC86 il =', il
c 28 set
        do i = 1, 56
            do j = 1, 8
                il = il + 1
                do k = 1, 5
                    IC166 (il, k) = IC85 (i, k)
                end do
                IC166 (il, 6) = IC81_1 (j, 6)
            end do
        end do
c 56 * 8 = 448 set 28 + 448 = 476
        print *, 'after IC166 IC85 IC81_1 il =', il
        write (6, *) 'after IC166 IC85 IC81_1 il =', il
        do i = 1, 70
            do j = 1, 28

```