EVOLVE v1 FORTRAN Source.txt Page 1 of 43 Saved: 2003/04/28, 10:14:09 Printed For: Frank Price

NOTE: this is listing of the minimal revisions that were done to the original Terak FORTRAN code to work as a glass teletype on the Mac in 6/89.

2

file: MA.IN.TEXT vrsn: 103 rvsd: 27 Aug 81 main routine, block data, reset

C r 5

C PROGRAM : EVOLVE 8

C 9 10

11

C WRITTEN BY: FRANK E. PRICE C **BIOLOGY DEPARTMENT** HAMILTON COLLEGE

12 C 13 | C

CLINTON, NEW YORK 13323 (315) 859-4387, (315) 853-8268

14 15 C C

C

C

C

C

C

COPYRIGHT 1981 BY FRANK E. PRICE

16 17 18

THIS PROGRAM IS DISTRIBUTED ON AN "AS IS" BASIS. CAVEAT EMPTOR, NO WARRANTY IS EXPRESSED OR IMPLIED.

23

25 C

EVOLVE IS A MONTE CARLO SIMULATION OF EVOLUTION OF A C DIALLELIC LOCUS WITH ALLELES 'W' AND 'M'. THE USER MAY VARY POPULATION SIZE, AND RATES OF REPRODUCTION, SURVIVAL, EMIGRATION AND IMMIGRATION OF EACH OF THE THREE GENOTYPES. THUS, EVOLVE WILL SIMULATE THE EFFECTS OF GENETIC DRIFT, GENE FLOW AND NATURAL SELECTION, EITHER ONE AT A TIME OR TOGETHER.

26 27

29 C C

C

C

C

C

C

C

C 36

C 37

C 38

C 39

C 40

C 41

C 45 C

C

C 51

C 52

C 53

C 54

C 55

C 56

C 57 C

58 C

59

42 C

43 C

44

46 C

47 C

48

50

MUCH INFORMATION HAS BEEN LEFT OUT OF THIS LISTING TO KEEP IT SUCCINCT. SEE USERS' AND INSTRUCTORS' MANUALS FOR ADDITIONAL INFORMATION.

34

35

## MISCELLANEOUS NOTES:

- 1) FORTRAN DIALECT: DEC FORTRAN IV V2.1. NOTE THE FOLLOWING:
  - A) USE OF 'D' IN PLACE OF 'C' IN COL. 1 DENOTES CONDITIONAL COMPILE OF DEBUG STATEMENTS.
  - B) USE OF 'ERR=' EXTENSION IN READ STATEMENTS TO AVOID SYSTEM IO CHECK ERRORS AND AWKWARD ABORTS. THESE MUST BE REVISED TO IMPROVE TRANSPORTABILITY.
  - C) USE OF "RAN", "SQRT", "ALOG" FORTRAN LIBRARY FUNCTIONS.
- 2) THIS VERSION IS RUNNING AN A TERAK (LSI-11 CPU) MICROCOMPUTER UNDER THE RT-11 OPERATING SYSTEM. INPUT IS VIA KEYBOARD, COMMUNICATION FROM PROGRAM TO USER IS VIA CRT, PROGRAM OUTPUT IS VIA 132-COLUMN LINE PRINTER.
- 3) CURRENTLY CONFIGURED FOR A 16-BIT CPU. IF EVOLVE IS TO BE IMPLEMENTED ON CPU'S WITH SMALLER WORD SIZES SOME VARIABLES MAY NEED TO BE CHANGED FROM INTEGER TO REAL TO AVOID INTEGER OVERFLOW PROBLEMS.
- 4) EVOLVE USES UNIFORMLY AND NORMALLY DISTRIBUTED PSEUDO-RANDOM NUMBERS (GENERATED BY EXTERNAL FUNCTION "RAN" AND INTERNAL SUBROUTINE "GAUSS"). THE OVERALL RANDOMNESS IN THIS VERSION IS PERHAPS A BIT LOW. IT HAS BEEN KEPT SO FOR TEACHING AND DEBUGGING PURPOSES. THIS ASPECT OF THE PROGRAM NEEDS MORE THOROUGH STUDY AND TESTING. THE RANDOMNESS COULD BE SIGNIFICANTLY CHANGED; SEE "RNOTE" IN SUBROUTINES REPROD AND SRVIVL FOR MORE ON MANNER AND PLACES WHERE RANDOMNESS MAY BE CHANGED.
- 5) NOTES ON ARRAYS
  - A) 2-ELEMENT ARRAYS STORING INFORMATION ON ALLELES (E.G., ALSUR(2), ALFIT(2)): FIRST ELEMENT CORRESPONDS TO THE "W"

EVOLVE v1 FORTRAN Source.txt
Page 2 of 43
Saved: 2003/04/28, 10:14:09
Printed For: Frank Price

```
ALLELE, THE SECOND ELEMENT CORRESPONDS TO THE "M" ALLELE.
60
   C
            B) 3-ELEMENT ARRAYS STORING INFORMATION ON GENOTYPES (E.G.,
61
               REPR8(3), SURVR8(3)): FIRST ELEMENT CORRESPONDS TO THE "WW"
   C
62
               GENOTYPE, THE SECOND TO THE "WM" GENOTYPE, THE THIRD TO
63
   C
               THE "MM" GENOTYPE.
64
   C
            C) RSLT(3,125) STORES THE RESULTS, I.E., THE NUMBER OF ADULTS
   C
65
   C
               OF EACH GENOTYPE IN EACH GENERATION. THE GENOTYPE
67
   C
               IS INDICATED BY THE FIRST INDEX, THE NUMBER OF THE
   C
               GENERATION IS INDICATED BY THE SECOND INDEX.
68
   C
69
      NOTES ON POSSIBLE CHANGES
70
   C
         1) STATEMENT NUMBERS SHOULD BE CLEANED UP -- SET IN NUMERICAL
71
   C
   C
            ORDER, INCREMENT BY 10'S.
72
73
   C
         2) CLEAN UP LEVELS OF INDENTATION
74
   C
         3) IO WILL HAVE TO BE MADE MORE TRANSPORTABLE.
75
   C
         4) TOO MUCH OUTPUT IS SENT TO LINE PRINTER. SHOULD MODIFY TO DO
            GENERATIONS REQUESTED, DISPLAY SUMMARY TO CRT, ASK USER WHETHER
   C
76
   C
            SUMMARIES &/OR GRAPHS ARE NEEDED. USER SHOULD BE ABLE TO MOVE
77
   C
            TO & FROM SUMMARIES AND THE VARIOUS GRAPHS ON THE CRT BEFORE
78
   C
            SENDING HARD COPY TO LINE PRINTER IF NEEDED. IMPLEMENTING
79
   C
            THIS FEATURE WILL PROBABLY REQUIRE ANOTHER SUBROUTINE
80
            ("DISPLA"?) TO CONTROL MENUES & CALL UP VARIOUS GRAPHS.
81
   C
            NEED TO ADD VARIOUS GLOBAL VARIABLES (PAGESIZE, LINESIZE &
82
            #LINESPRINTED, ETC.) TO PAGE SUMRYS OUTPUT, SET GRAPH SIZES TO,
   C
83
            FOR E.G., 80 X 24 AND 66 X 132, ETC.
84
   C
85
86
   C**BEGIN MAIN ROUTINE
87
88
   C
   C
       FUNCTION: CONTROL OVERALL FLOW OF THE PROGRAM
89
   C
90
   C
       CALLS: INPUT - INPUT AND CHECK PARAMETERS FOR RUN; RETURNS ALL
91
   C
                       VARIABLES NEEDED FOR A RUN.
92
              REPROD - DETERMINE NUMBER OF JUVENILES IN CURRENT
   C
93
                       GENERATION; RETURNS "JUV" ARRAY.
   C
94
   C
              SRVIVL - DETERMINE NUMBER OF ADULTS IN CURRENT GENERATION
95
   C
                       FOLLOWING EMIGRATION, MORTALITY & IMMIGRATION;
96
   C
                       CRASH EXCESSIVELY LARGE POPULATION; RETURNS
97
                       NUMBER OF ADULTS IN "RSLTS" ARRAY.
   C
98
              SUMRYS - PRINT SUMMARIES OF GENERATIONS
99
   C
              GRAPH - PRINT VARIOUS GRAPHS OF RESULTS
   C
100
              YESNO - READ & TEST YES/NO RESPONSES; RETURNS EITHER "Y" OR
   C
                       "N" IN VARIABLE "ANS".
102
   C
              RESET - REINITIALIZE VARIABLES BEFORE A NEW RUN.
103
   C======
104
         INCLUDE 'COMMON.f'
105
106
         LOCAL DECLARATIONS
107
         INTEGER ALMFLG, ALWFLG, ATOT, ICOUNT, POPFLG, SCRCNT
108
   109
   C
110
      INPUT DATA, SET ALLELE & POP. FLAGS & PRINT SUMMARY OF INITIAL POP.
   C
111
112
       Change scrolling of output window to single line scrolling
113
   C
         CALL OutWindowScroll(9999)
114
115
   C
         WRITE (OUTSCR, 10)
116
            FORMAT ('1',80('*'),/,20X,'EVOLVE version 3.1, 27 Aug. 1981',/,
117
               80('*'),/)
118
119
     100 CALL INPUT
```

```
120 | D
          WRITE(OUTLP, 200) IGEN, (RSLT(K, IGEN), K=1,3)
121
   D
   D 200 FORMAT(' On return from INPUT:',/,' IGEN = ',I4,' RSLT = ',3(I7))
122
123
   D
          WRITE(OUTLP, 205) MXPOP1, MXPOP2
   D 205 FORMAT(' K = ', I5, 10X, 'POST-K = ', I5)
124
   D
          WRITE(OUTLP, 210) (SURVR8(K), K=1, 3)
125
   D 210 FORMAT(' SURVR8 = ',3(F10.2))
127
          WRITE(OUTLP, 215) (REPR8(K), K=1,3)
   D 215 FORMAT(' REPR8 = ',3(F10.2))
128
          WRITE(OUTLP, 220) (INFLO(K), K=1,3)
129
130
   D 220 FORMAT(' INFLO = ',3(I7))
          WRITE(OUTLP, 225) (OUTFLO(K), K=1,3)
   D
131
   D 225 FORMAT(' OUTFLO = ',3(I7))
132
   D
          WRITE(OUTLP, 230) NOGENS, SUMRY, NEXT
   D 230 FORMAT(' NOGENS = ',I7,'SUMRY = ',I7,'NEXT = ',A4)
134
   D
          WRITE(OUTLP, 235) IGEN
135
   D 235 FORMAT(' IGEN = ', I4,/)
136
    C
137
   C
          INITIALIZE FLAGS FOR ALLELE & POPULATION EXTINCTION, COUNTER FOR
138
              SUMMARIES, HEADER FOR PROGRESS REPORT TO SCREEN.
    C
139
140
      120 \text{ ALWFLG} = 0
          ALMFLG = 0
141
          POPFLG = 0
142
          ICOUNT = 0
143
          SCRCNT = 1
144
          IF (IGEN .GT. 1) GO TO 300
145
146
          SUMGEN = IGEN
147
          PRINT2 = OUTLP
          CALL SUMRYS
148
      300 WRITE (OUTSCR, 301)
149
             FORMAT ('0',X,'Doing generation:')
150
      301
151
152
    C DO NUMBER OF GENERATIONS REQUESTED
153
    C----
154
          DO 390 I=1,NOGENS
155
    C
156
             TEST THAT # GENS DONE < 126; CONTINUE IF TRUE, EXIT IF FALSE.
    C
157
              IF (IGEN .GT. 125) GO TO 400
158
    C
159
             WRITE PROGRESS REPORT TO SCREEN
    C
160
             IGEN = IGEN + 1
161
              SCRCNT = SCRCNT + 1
162
             WRITE (OUTSCR, 303) IGEN
      302
163
                 FORMAT (2X, I3, $)
      303
164
              IF (SCRCNT .LE. 19) GO TO 305
165
              SCRCNT = 0
166
    C
167
           REPRODUCTION AND SURVIVAL - HEART OF THE PROGRAM
    C
168
      305
169
              CALL REPROD
              CALL SRVIVL
170
    C
171
         SUMMARIES (PRINTING THESE TAKES A LOT OF TIME)
172
    C
              IF (SUMRY .EQ. 0) GO TO 309
173
              ICOUNT = ICOUNT + 1
174
175
              IF (SUMRY .NE. ICOUNT) GO TO 309
              PRINT2 = OUTLP
176
              SUMGEN = IGEN
177
              CALL SUMRYS
178
179
              ICOUNT = 0
```

```
309
             CONTINUE
180
   C
181
             TEST FOR EXTINCTIONS & PRINT MESSAGE IF HAS JUST
   C
182
183
    C
                HAPPENED (THESE COULD GO INTO A SUBROUTINE ["XTNCTN"?]
                TO CLARIFY LOGIC & CLEAN UP MAIN ROUTINE)
184
   C
   C
185
   C
             TEST FOR ALLELE EXTINCTION
186
187
             IF ((RSLT(1,IGEN) .GT. 0) .OR. (RSLT(2,IGEN) .GT. 0))
             GO TO 320
188
             ALWFLG = ALWFLG + 1
189
             IF (ALWFLG .GT. 1) GO TO 325
190
             WRITE (OUTSCR, 310) IGEN
191
      310
             FORMAT ('0',/,' W ALLELE BECAME EXTINCT IN GENERATION ',I3)
192
193
             WRITE (OUTLP, 310) IGEN
194
             G0 T0 325
      320
             ALWFLG = 0
195
    C
196
      325
             IF ((RSLT(3,IGEN) .GT. 0) .OR. (RSLT(2,IGEN) .GT. 0))
197
             GO TO 340
198
             ALMFLG = ALMFLG + 1
199
             IF (ALMFLG .GT. 1) GO TO 350
200
             WRITE (OUTSCR, 330) IGEN
201
      330
             FORMAT ('0',/,' M ALLELE BECAME EXTINCT IN GENERATION ',I3)
202
             WRITE (OUTLP, 330) IGEN
203
             GO TO 350
204
      340
             ALMFLG = 0
205
   C
206
   C
207
             TEST FOR POPULATION EXTINCTION
208
      350
             ATOT = RSLT(1, IGEN) + RSLT(2, IGEN) + RSLT(3, IGEN)
209
             IF (ATOT .GT. 0) GO TO 370
210
             POPFLG = POPFLG + 1
211
             IF (POPFLG .GT. 1) GO TO 390
212
             WRITE (OUTSCR, 360) IGEN
213
      360
             FORMAT ('0',/,' POPULATION BECAME EXTINCT IN GENERATION ',I3)
214
             WRITE (OUTLP, 360) IGEN
215
             GO TO 390
216
      370
             POPFLG = 0
217
   C
218
219
   C
     390 CONTINUE
220
          WRITE (OUTSCR, 395)
221
             FORMAT ('0',/)
222
   C
223
    C---
224
      CLEANUP AFTER GENERATIONS HAVE BEEN FINISHED
225
          (THIS WOULD BE THE PLACE TO ADD THE "DISPLA" SUBROUTINE; MUCH OF
226
    C
   C
           THIS CODE WOULD GO INTO THAT SUBROUTINE)
227
   C--
228
   C
229
    C
          DETERMINE WHETHER GRAPHS OR PARAMETER CHANGE DESIRED
230
      400 IF (NEXT .EQ. 'GR') GO TO 500
231
          IF (NEXT .EQ. 'CH') GO TO 100
232
   C
233
          CALL SUMRYS, IF REQUESTED
234
      500 IF ((SUMRY .NE. 0) .AND. (ICOUNT .EQ. 0)) GO TO 600
235
          SUMGEN = IGEN
236
          PRINT2 = OUTLP
237
          CALL SUMRYS
238
239
   C
```

```
240 C
        PRINT 1ST & LAST GENERATIONS TO SCREEN, ASK IF GRAPHS DESIRED,
        CALL GRAPH IF NEEDED
241
    600 \text{ SUMGEN} = 1
242
243
        PRINT2 = OUTSCR
244
        CALL SUMRYS
        SUMGEN = IGEN
245
        CALL SUMRYS
247
        WRITE (OUTSCR, 610)
    610 FORMAT (' Do you wish graphs (Y/N)? ',$)
248
        CALL YESNO (ANS)
249
        IF (ANS .EQ. 'N') GO TO 700
250
        IF (GRF(1) .EQ. ' ') GO TO 700
251
        CALL GRAPH
252
   C
253
254
        ASK FOR CONTINUATION. IF NO, STOP; IF YES, RESET GENERATION
        NUMBER AND RESULTS ARRAY
255
    700 WRITE (OUTLP,710)
256
    710
          FORMAT (1H1)
257
        WRITE (OUTSCR,720)
258
    720 FORMAT (//,3X,'Run finished. Do you wish to make another ',
259
       +'run (Y/N)? ',$)
260
        CALL YESNO (ANS)
261
        IF (ANS .EQ. 'N') GO TO 999
262
          CALL RESET
263
          GO TO 100
264
   C
265
    999 STOP
266
        END
267
268
   269
270
271
   C
272
        BLOCK DATA
273
        SUBROUTINE INITIALIZEDATA
274
   C
275
      FUNCTION: INITIALIZES VARIABLES IN COMMON BLOCKS
   C
276
277
278
        INCLUDE 'COMMON.f'
279
   280
   C *** INITIALIZE COMMON VARIABLES
282
             ALSUR/2*0./, AM/0./, ANS/' '/, ASELCO/2*0./,
283
          FITAL/2*0./, FITGNT/3*0./, FRQMP/0./, GNTSUR/3*0./,
284
          GRF/10*' '/, GSELCO/3*0./, IGEN/1/, IN/5/, INFLO/3*0/,
285
          IY/12345/, IZ/12345/, JUV/3*0./, LGEN/0/, MATES/6*0/,
286
          MXPOP1/0/, MXPOP2/0/, NEXT/' '/, NOADLT/3*0/, NOGENS/0/,
287
          NOOUT/3*0./, NOYONG/6*0./, OBSFR/3*0./, OUTFLO/3*0/
288
        DATA OUTLP/6/, OUTSCR/6/, PRINT2/6/, REPR8/3*0./,
289
          RSLT/375*0/, S/0./, SUMGEN/1/, SUMRY/0/, SURVR8/3*0./,
290
          TITLE/40*' '/, TOADLT/0/, V/0./
291
        FND
292
   293
294
   295
        SUBROUTINE RESET
296
   C
297
      FUNCTION: REINITIALIZES VARIABLES FOLLOWING A RUN.
   C
298
299
   C
```

EVOLVE v1 FORTRAN Source.txt

Page 6 of 43
Saved: 2003/04/28, 10:14:09

Printed For: Frank Price

```
300 C
      NOTES:
         1) APPROPRIATE ARRAYS AND VARIABLES ARE RETURNED TO VALUES IN
301 C
            BLOCK DATA, EXCEPT THAT FIRST GENERATION (INITIAL POPULATION)
302 | C
            IN RESULTS ARRAY RETAINS THE VALUES INPUT BY THE USER.
   C
303
   C
         2) VARIABLES INPUT BY USER RETAIN THEIR VALUES SO USER CAN MAKE
304
            REPEAT RUNS WITHOUT REENTERING ALL DATA, HOPEFULLY, ANY
  ∥ (
305
  | C
            REMAINING VARIABLES NOT RESET ARE NOT IMPORTANT.
306
307
   l C
         3) ALL MAJOR VARIABLES ARE IN COMMON.
   C
308
   C
       CALLED FROM: MAIN
309
310
       CALLS: NOTHING
311
   INCLUDE 'COMMON.f'
312
314 C
315 C
        Type *,'In Reset'
         D0 100 I = 1,3
316
           D0 10 J = 2,125
317
              RSLT(I,J) = 0
318
      10
           CONTINUE
319
           JUV(I) = 0.
320
           NOOUT(I) = 0
321
            NOIN(I) = 0
322
     100 CONTINUE
323
        D0 200 J = 1,6
324
           MATES(J) = 0
325
           NOYONG(J) = 0.
326
     200
           CONTINUE
327
        IZ = 12345
328
         IGEN = 1
329
         RETURN
330
         END
331
332
   333
             file: in.pu
                         vrsn: 145
334
  || C
                                    rvsd: 27 Aug 81
                   subroutines input, yesno, inpsum
335
  336
  || C
337
         SUBROUTINE INPUT
338
339
   C
     FUNCTION: THIS SUBROUTINE READS, CHECKS & PRINTS DATA INPUT.
340 C
341 C
342 | C
      SYNOPSIS:
343 | C
          1) THE FOLLOWING VARIABLES (ALL IN COMMON) ARE READ IN:
            TITLE - 80 CHARACTER (40A2) TITLE FOR OUTPUT
  C
344
345
   С
                   RANDOM NUMBER SEED; 0->32760
   C
            RSLT - NUMBER OF EACH GENOTYPE IN FIRST GENERATION; ARRAY
346
               ELEMENTS (1,1), (2,1), (3,1); EACH 0->4000, TOTAL < 4001
  II (
347
            MXPOP1 - CARRYING CAPACITY, MAXIMUM PERMISSIBLE POPULATION
348
  || C
               SIZE; 10->5000
349 C
350 C
            MXPOP2 - POST-CRASH POPULATION SIZE; 2->4000
            SURVR8 - SURVIVAL RATE OF EACH GENOTYPE; 0.0->100.0
351 || C
  | C
            REPR8 - REPRODUCTIVE RATE OF EACH GENOTYPE; 0.0->10.0
352
  || C
            INFLO - NUMBER OF EACH GENOTYPE IMMIGRATING PER GENERATION;
353
354 | C
               0->2000
  || C
            OUTFLO - % OF EACH GENOTYPE EMIGRATING PER GENERATION; 0->100
355
  || C
            NOGENS - NUMBER OF GENERATIONS TO BE RUN WITH THESE DATA;
356
  || C
357
            SUMRY - NUMBER OF GENERATIONS BETWEEN PRINTED SUMMARIES;
  || C
358
   C
359
               0 = NO SUMMARIES, 1->99
```

EVOLVE v1 FORTRAN Source.txt
Page 7 of 43
Saved: 2003/04/28, 10:14:09
Printed For: Frank Price

```
360 || C
                      - WHAT TO DO AFTER GENERATIONS ARE FINISHED;
                  "CH" = CHANGE PARAMETERS, "GR" = GRAPH RESULTS
361 C
   || C
                      - UP TO 10 2-LETTER CODES FOR TYPES OF GRAPHS DESIRED;
362
363
   C
                  SEE USERS' MANUAL FOR SPECIFICS
           2) USER IS PROMPTED FOR EACH DATUM, WHICH IS CHECKED AGAINST
364
   C
              LIMITS & AN ERROR MESSAGE PRINTED IF NEEDED. USER CAN THEN
   l (
365
   | C
              REENTER DATA.
366
367
           3) AFTER ALL DATA ARE ENTERED, SUBROUTINE INPSUM IS CALLED TO
   l C
              PRINT A SUMMARY TO THE SCREEN FOR THE USER'S INSPECTION.
368
   C
           4) IF DATA ARE OK, A SUMMARY IS PRINTED TO THE LINE PRINTER BY
369
370
   C
              SUBROUTINE INPSUM AND CONTROL RETURNS TO MAIN.
371
   C
           5) IF USER DETECTS AN ERROR, VARIABLE "FIX" IS SET TO 'Y' AND
   II C
              CONTROL RETURNS TO THE TOP OF SUBROUTINE INPUT. VALUES OF
372
   || C
              DATA ARE DISPLAYED & THE USER PROMPED TO CORRECT THEM IF
373
374
              NECESSARY.
   || C
           6) IF VAR. "NEXT" = "CH", SUBROUTINE INPUT HAS BEEN CALLED
375
              BEFORE, SO "IY" AND INITIAL POPULATION SHOULD NOT BE CHANGED;
   C
376
              CONTROL JUMPS TO CARRYING CAPACITY.
377
   C
378
   C
        NOTES:
379
           1) THIS ROUTINE RELIES HEAVILY ON THE "ERR=" DEC FORTRAN
   C
380
              EXTENSION TO AVOID AWKWARD SYSTEM ABORTS. THIS WILL NEED
381
              TO BE CHANGED TO IMPROVE TRANSPORTABILITY.
   C
382
           2) THE RANDOM NUMBER SEED MAY NEED TO BE CHANGED FOR OTHER
   C
383
              SYSTEMS. THE SEED COULD BE SET IN OTHER WAYS (E.G., FROM A
384
385
   C
              SYSTEM CLOCK), BUT I HAVE FOUND IT USEFUL TO HAVE STUDENTS
              BE ABLE TO REPEAT RUNS EXACTLY.
   C
386
   C
387
   C
        CALLED FROM: MAIN
388
   C
        CALLS: YESNO - TO READ AND CHECK YES/NO INPUT; RETURNS EITHER "Y"
389
                         OR "N" IN VARIABLE "ANS".
   C
390
               INPSUM - TO PRINT SUMMARIES OF DATA TO SCREEN
    C
391
                         (IF "PRINT2" = "OUTSCR") OR TO LINE PRINTER
    C
392
                         (IF "PRINT2" = "OUTLP")
    \mathbf{C}
393
394
          INCLUDE 'COMMON.f'
395
396
      LOCAL DECLARATIONS
397
          INTEGER ATOT, GNOT(3), IFLOW, TGENS
398
          REAL RRATE, SRATE
399
          STRING*2 FIX, GRFTST(11)
400
                   ATOT/0/, FIX/'N'/,
401
             GNOT/'WW','WM','MM'/,
402
             GRFTST/' ','AF','GF','PO','WW','WM','MM','AW','GW','AS',
403
         3
                 'GS'/,
404
405
         4
             RRATE/0./, SRATE/0./
406
   C
407
          Type *, 'In Input'
408
409
          IF NEXT IS 'CH', SUBROUTINE INPUT HAS ALREADY BEEN
   C
410
          CALLED, JUMP OVER INITIAL PARAMETERS TO CARRYING CAPACITY
   C
411
       10 IF (FIX .EQ. 'Y') GO TO 100
412
          IF (NEXT .EQ. 'CH') GO TO 500
413
   C
414
415
   l C
       INPUT DATA
416
   C
   C
417
418
   C
   C
419
          TITLE FOR OUTPUT
```

```
420 | C
             (THERE MUST BE A MORE ELEGANT WAY TO TEST FOR BLANK TITLE)
421
      100 DO 102 I=1,40
422
             IF (TITLE(I) .EQ. ' ') GO TO 102
423
424
             GO TO 108
             CONTINUE
      102
425
          GO TO 120
426
427
428
      108 WRITE (OUTSCR, 110) TITLE
      110 FORMAT (/,5X,'Title is:'/,X,40A2)
429
430
          WRITE (OUTSCR, 115)
431
      115 FORMAT (5X, 'Is this o.k. (Y/N)? ',$)
          CALL YESNO (ANS)
432
433
          IF (ANS .EQ. 'Y') GO TO 200
434 | C
      120 WRITE (OUTSCR, 130)
435
      130 FORMAT ('0',5X,'Enter a title (79 characters max) for output:')
436
          READ (IN, 140) TITLE
437
      140 FORMAT(40A2)
438
   D
          WRITE (OUTSCR, 150) TITLE
439
    D 150 FORMAT ('0 1. ', 40A2)
441
   || C
442
443 | C
      RANDK SEED
444
          THE SEED SIZE MAY NEED TO BE CHANGED FOR DIFFERENT COMPUTERS
445
    C
      200 IF (IY .EQ. 12345) GO TO 220
446
             WRITE (OUTSCR, 210) IY
447
      210
                 FORMAT (/,5X,'Seed is: ',I5)
448
             WRITE (OUTSCR, 115)
449
             CALL YESNO (ANS)
450
             IF (ANS .EQ. 'Y') GO TO 300
451
      220 WRITE (OUTSCR, 230, ERR=250)
452
             FORMAT (/,5X,'Enter random number seed (0-32760): ',$)
453
          READ (IN,240,ERR=250) IY
454
      240 FORMAT (I5)
455
   C
456
          TEST: RANDK SEED MUST BE BETWEEN 00000 AND 32760
457
          IF ((IY .GE. 00000) .AND. (IY .LE. 32760)) GO TO 270
458
      250
             WRITE (OUTSCR, 260)
459
                FORMAT (' ****ERROR: Seed is incorrect.',/,13X,
      260
460
                'it must be an integer between 0 and 32760)')
461
             WRITE (OUTSCR, 265)
462
      265
                 FORMAT (13X, 'Please try again.')
463
             GO TO 220
464
      270 CONTINUE
465
466
          WRITE (OUTSCR, 280) IY
467
    D 280 FORMAT ('0 2. RANDOM NUMBER SEED : ',I5)
469
470 C
   || C
      INITIAL POPULATION
471
472
      300 ATOT = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
473
          IF (ATOT .EQ. 0) GO TO 320
474
             WRITE (OUTSCR, 310) RSLT(1,1), RSLT(2,1), RSLT(3,1), ATOT
475
      310
                 FORMAT (/,5X,'The initial population is: WW = ',I4,
476
                 ' WM = ',I4,' MM = ',I4,' Total = ',I4)
477
             WRITE (OUTSCR, 115)
478
479
             CALL YESNO (ANS)
```

Page 8 of 43

Printed For: Frank Price

```
IF (ANS .EQ. 'Y') GO TO 500
480
    C
481
      320 DO 380 I=1,3
482
             WRITE (OUTSCR, 340) GNOT(I)
483
      330
                 FORMAT (/,5X,'Enter initial number of ',A2,' genotype ',
      340
484
                 '(0-4000): ',$)
485
             READ (IN, 350, ERR=360) RSLT(I, 1)
486
      350
             FORMAT (I4)
487
             IF ((RSLT(I,1) .GE. 0) .AND. (RSLT(I,1) .LE. 4000)) GO TO 380
488
                WRITE (OUTSCR, 370) GNOT(I)
      360
489
                    FORMAT (' ****ERROR: Initial number of ',A2,
490
      370
                    ' genotype is incorrect. ',/,13X,'It must be an ',
         $
491
         $
                    'integer between 0 and 4000.')
492
                WRITE (OUTSCR, 265)
493
494
                 G0 T0 330
      380
             CONTINUE
495
    C
496
    C
          TEST: INITIAL POPULATION MUST TOTAL BETWEEN 2 AND 4000 INDIVIDUALS
497
          ATOT = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
498
          IF ((ATOT .GE. 2) .AND. (ATOT .LE. 4000)) GO TO 400
499
             WRITE (OUTSCR, 390)
500
      390
                 FORMAT (' ****ERROR: Initial population is out of ',
501
         $
                 'range.',/,13X,'The total initial population must be ',
502
                 'between 2 and 4000')
503
             WRITE (OUTSCR, 265)
504
             G0 T0 320
505
      400 CONTINUE
506
   || C
507
   D
          WRITE (OUTSCR, 410) (RSLT(I, 1), I=1, 3), ATOT
508
             FORMAT ('0 3. INITIAL POPULATION : WW = ', I4, 2X, 'WM = ', I4,
    D 410
509
    D
             2X, 'MM = ', I4, 2X, 'TOTAL = ', I4)
510
511
    C
512
       CARRYING CAPACITY & POST-CRASH POP SIZE
    C
513
514
      500 IF ((MXPOP1 .EQ. 0) .AND. (MXPOP2 .EQ. 0)) GO TO 509
515
             WRITE (OUTSCR,501) MXPOP1, MXPOP2
516
             FORMAT (/,5X, 'Carrying capacity is: ',I4,' Post-crash level ',
      501
517
             ' is:',I4)
518
             WRITE (OUTSCR, 115)
519
             CALL YESNO (ANS)
520
             IF (ANS .EQ. 'Y') GO TO 600
522
    C
523
      509
             WRITE (OUTSCR,510)
524
             FORMAT (/,5X,'Enter carrying capacity (10-5000): ',$)
525
          READ (IN,520,ERR=525) MXPOP1
526
             FORMAT (I4)
      520
527
   C
528
          TEST: CARRYING CAPACITY MUST BE BETWEEN 10 AND 5000 INDIVIDUALS
529
          IF ((MXPOP1 .GE. 10) .AND. (MXPOP1 .LE. 5000)) GO TO 540
530
      525
             WRITE (OUTSCR,530)
531
                 FORMAT (' ****ERROR: Carrying capacity is incorrect.',/,13X,
      530
532
                 'it must be an integer between 10 and 5000.')
533
             WRITE (OUTSCR, 265)
534
             G0 T0 509
535
536
    C.....
537
      540 WRITE (OUTSCR, 550)
538
539
      550
             FORMAT (/,5X,'Enter post-crash population size (2-4000): ',$)
```

Page 10 of 43

```
READ (IN,560,ERR=575) MXPOP2
540
     560
             FORMAT (I4)
541
_{542}\,\|\,C
         TEST: POST-CRASH SIZE MUST BE BETWEEN 2 AND 4000 INDIVIDUALS
543
   C
     570 IF ((MXPOP2 .GE. 2) .AND. (MXPOP2 .LE. 4000)) GO TO 590
544
     575
             WRITE (OUTSCR, 580)
545
     580
             FORMAT (' ****ERROR: Post-crash pop. size is incorrect.',/,
             13X,'It must be an integer between 2 and 4000.')
547
             WRITE (OUTSCR, 265)
548
             G0 T0 540
549
550
   C.....
551 C
         TEST: CARRYING CAPACITY MUST BE GREATER THAN POST-CRASH SIZE
552
     590 IF (MXPOP1 .GT. MXPOP2) GO TO 594
553
554
             WRITE (OUTSCR, 592)
             FORMAT (' ****ERROR: Post-crash pop. size is greater than',
555
             ' carrying capacity.',/13X,'Please reenter both parameters.')
556
             G0 T0 509
557
     594 CONTINUE
558
   C
559
             WRITE (OUTSCR,596) MXPOP1, MXPOP2
   D
560
             FORMAT (' 4. CARRYING CAPACITY
                                              : ',I4,/,4X,
561
           ' POST-CRASH POP. SIZE : ', I4)
562
   C-
563
564
      SURVIVAL & REPRODUCTIVE RATES OF EACH GENOTYPE
   C
565
   C
566
     600 RRATE = REPR8(1) + REPR8(2) + REPR8(3)
567
          SRATE = SURVR8(1) + SURVR8(2) + SURVR8(3)
568
          IF ((RRATE .EQ. 0.) .AND. (SRATE .EQ. 0.)) GO TO 700
569
             WRITE (OUTSCR,620) SURVR8(1), SURVR8(2), SURVR8(3), REPR8(1),
570
                REPR8(2), REPR8(3)
571
             FORMAT (/,5X,'Survival rates are:
                                                 WW = ', F5.1, 'WM = '
     620
572
                F5.1, 'MM = ',F5.1,/,5X,'Reproductive rates are: WW =
         $
573
                F4.1,' WM = ',F4.1,' MM = ',F4.1,/,5X,'Are survival ',
         $
574
                'rates o.k. (Y/N)?',$)
575
             CALL YESNO (ANS)
576
             IF (ANS .EQ. 'Y') GO TO 760
577
578
579
         SURVIVAL RATES
580
     700 DO 750 I=1,3
581
             WRITE (OUTSCR,715) GNOT(I)
582
      715
             FORMAT (/,5X,'Enter survival rate of genotype ',A2,
583
             ' (000.0-100.0%): ',$)
584
585
             READ (IN,720,ERR=730) SURVR8(I)
      720
             FORMAT (F5.1)
586
             IF ((SURVR8(I) .GE. 0.) .AND. (SURVR8(I) .LE. 100.)) GO TO 750
587
     730
                WRITE (OUTSCR, 735) GNOT(I)
588
     735
                FORMAT (' ****ERROR: Survival rate of ',A2,' genotype ',
589
         $
                'is incorrect.',/,13X,'It must be a decimal number ',
590
                'between 000.0 and 100.0%.')
591
         $
                WRITE (OUTSCR, 265)
592
                GO TO 710
593
     750
             CONTINUE
594
595
   С..
596
     REPRODUCTIVE RATES OF EACH GENOTYPE
597
     760 IF (SRATE .EQ. 0.) GO TO 770
598
599
             WRITE (OUTSCR, 765)
```

```
765
             FORMAT (/,5X,'Are reproductive rates o.k. (Y/N)? ',$)
600
             CALL YESNO (ANS)
601
             IF (ANS .EQ. 'Y') GO TO 800
602
603
      770 DO 790 I=1,3
604
      772
             WRITE (OUTSCR,775) GNOT(I)
605
      775
             FORMAT (/,5X,'Enter reproductive rate of genotype ',A2,
606
             ' (0.0-10.0): ',$)
607
             READ (IN, 780, ERR=785) REPR8(I)
608
      780
             FORMAT (F4.1)
609
610
             IF ((REPR8(I) .GE. 0.) .AND. (REPR8(I) .LE. 10.)) GO TO 790
      785
                WRITE (OUTSCR, 786) GNOT(I)
611
      786
                FORMAT (' ****ERROR: Reproductive rate of genotype ',A2,
612
         $
                 ' is incorrect.',/,13X,'It must be a decimal number ',
613
                'between 00.0 and 10.0.')
614
                WRITE (OUTSCR, 265)
615
                G0 T0 772
616
      790
             CONTINUE
617
    C
618
          WRITE (OUTSCR, 795) (SURVR8(I), I=1,3)
619
    D 795 FORMAT (' 5. GNOT. SURVIVAL RATES : WW = ',F5.1,'% WM = ',
620
             F5.1,'% MM = ',F5.1,'%')
621
          WRITE (OUTSCR, 796) (REPR8(I), I=1,3)
622
   D 796 FORMAT (' GNOT. REPROD. RATES : WW = ',F4.1,2X,'WM = ',
623
        F4.1,2X,'MM = ',F4.1
    D
624
625
   | C-
626 | C
   C
      GENE FLOW
627
   | C
628
      800 \text{ IFLOW} = 0
629
          D0 810 I=1,3
630
             IFLOW = INFLO(I) + OUTFLO(I)
631
      810
             CONTINUE
632
          IF (IFLOW .EQ. 0) GO TO 830
633
          WRITE (OUTSCR,820) INFLO(1), INFLO(2), INFLO(3), OUTFLO(1),
634
             OUTFLO(2), OUTFLO(3)
635
             FORMAT (/,5X,'Gene\ flow: # in : WW = ',I4,' WM = ',I4,
      820
636
                 ' MM = ',I4,/16X,'% out: WW = ',I3,' WM = ',I3,
         $
637
                 ' MM = ', I3)
638
          WRITE (OUTSCR, 115)
639
          CALL YESNO (ANS)
640
          IF (ANS .EQ. 'Y') GO TO 1000
641
642
      830 WRITE (OUTSCR, 840)
643
             FORMAT (/,5X,'Do you wish to have gene flow (Y/N)? ',$)
      840
644
645
             CALL YESNO (ANS)
             IF (ANS .EQ. 'Y') GO TO 900
646
                 DO 850 I=1,3
647
                    INFLO(I) = 0
648
                    OUTFLO(I) = 0
649
      850
                    CONTINUE
650
                GO TO 1000
651
652
653
    C
          IMMIGRATION RATES
654
    C
      900 DO 950 I=1,3
655
      910
             WRITE (OUTSCR, 915) GNOT(I)
656
             FORMAT (/,5X,'Enter number of ',A2,' individuals entering ',
      915
657
             'per generation (0-4000): ',$)
658
659
             READ (IN,930, ERR=940) INFLO(I)
```

Page 12 of 43

```
930
            FORMAT (I4)
660
            IF ((INFLO(I) .GE. 0) .AND. (INFLO(I) .LE. 4000)) GO TO 950
661
     940
            WRITE (OUTSCR, 945) GNOT(I)
662
            FORMAT (' ****ERROR: Number of ',A2,' individuals entering ',
663
     945
             'is incorrect.',/,13X,'It must be an integer between 0 and ',
664
             '4000.')
665
            WRITE (OUTSCR, 265)
666
            GO TO 910
667
     950
            CONTINUE
668
   C.....
669
670
671
         EMIGRATION RATES
         DO 990 I=1,3
672
     960
            WRITE (OUTSCR, 965) GNOT(I)
673
674
     965
            FORMAT (/,5X,'Enter % of ',A2,' individuals leaving per ',
             'generation (0-100): ',$)
675
            READ (IN, 970, ERR=980) OUTFLO(I)
676
     970
            FORMAT (I3)
677
            IF ((OUTFLO(I) .GE. 0) .AND. (OUTFLO(I) .LE. 100)) GO TO 990
678
     980
            WRITE (OUTSCR, 985) GNOT(I)
679
            FORMAT (' ****ERROR: % of ',A2,' individuals leaving is '
     985
            'incorrect.',/,13X,'It must be an integer getween 0 and 100.')
681
            WRITE (OUTSCR, 265)
682
            G0 T0 960
683
     990
            CONTINUE
684
   C.....
685
686 || C
         WRITE (OUTSCR,995) (INFLO(I), I=1,3)
687
  ∥ D
688 D 995
            FORMAT (' 6. FLOW: # IN PER GNOT. : WW =', I5, 3X, 'WM =', I5,
            3X,'MM =',I5)
689
         WRITE (OUTSCR,996) (OUTFLO(I), I=1,3)
690
            FORMAT (11X, '% OUT PER GNOT.: WW = ', I3, 3X, 'WM = ', I3, 3X,
   D 996
691
             'MM = ', I3)
   D
692
   (-
693
694
   l C
   C NUMBER OF GENERATIONS
695
696
    1000 IF (NOGENS .EQ. 0) GO TO 1002
697
            WRITE (OUTSCR, 1001) NOGENS
698
            FORMAT (/,5X,'Model set to run for ',I3,' generations.')
    1001
699
         WRITE (OUTSCR, 115)
700
         CALL YESNO (ANS)
701
         IF (ANS .EQ. 'Y') GO TO 1014
702
703
    1002 WRITE (OUTSCR, 1003)
704
            FORMAT (/,5X, 'Enter number of generations you wish to run ',
705
     1003
             'with these data (1-124): ',$)
706
         READ (IN, 1004, ERR=1006) NOGENS
707
    1004
            FORMAT (I3)
708
         IF ((NOGENS .GE. 1) .AND. (NOGENS .LE. 124)) GO TO 1010
709
   C
710
            WRITE (OUTSCR, 1008)
    1006
711
            FORMAT (' ****ERROR: No. of generations is incorrect.',/,13X,
    1008
712
             'It must be an integer between 1 and 124.')
713
            WRITE (OUTSCR, 265)
714
            GO TO 1002
715
716
    1010 TGENS = IGEN + NOGENS
717
         IF (TGENS .LE. 125) GO TO 1012
718
719
         WRITE (OUTSCR, 1011)
```

Page 13 of 43

```
FORMAT (' +++++WARNING: Total no. of generations requested ',
720
     1011
             'exceeds 125.',/,15X,'Run will stop at generation 125.')
721
          NOGENS = 125 - IGEN
722
723
    C
724
    C
          INTERVAL BETWEEN SUMMARIES
725
    C
726
     1012 IF (SUMRY .NE. 0) GO TO 1014
727
             WRITE (OUTSCR, 1013)
728
     1013
                 FORMAT (/,5X,'Summaries will be printed for first and ',
729
730
         $
                    'last generations only.')
             GO TO 1016
731
    C
732
    1014 WRITE (OUTSCR, 1015) SUMRY
733
734
             FORMAT (/,5X,'Summaries will be printed after every ',I2,
             ' generation(s).')
735
    C
736
     1016 WRITE(OUTSCR, 115)
737
             CALL YESNO (ANS)
738
             IF (ANS .EQ. 'Y') GO TO 1030
739
    С
740
          WRITE (OUTSCR, 1017)
741
     1017
             FORMAT (/,5X,'How many generations do you wish between ',
742
             'summaries?',/,13X,'(Enter 0 for no summaries, or 1-99): ',$)
743
           READ (IN, 1018, ERR=1020) SUMRY
744
             FORMAT (I2)
     1018
745
           IF ((SUMRY .GE. 0) .AND. (SUMRY .LE. 99)) GO TO 1030
746
     1020
             WRITE (OUTSCR, 1022)
747
             FORMAT (' ****ERROR: Summary request is incorrect.',/,13X,
748
     1022
             'It must be an integer between 0 and 99.')
749
             WRITE (OUTSCR, 265)
750
             GO TO 1016
751
752
   C
753
          WHAT TO DO AFTER GENERATIONS ARE FINISHED
   C
754
755
     1030 IF ((NEXT .NE. 'CH') .AND. (NEXT .NE. 'GR')) GO TO 1040
756
             IF (NEXT .EQ. 'GR') GO TO 1032
757
                WRITE (OUTSCR, 1031)
758
     1031
                 FORMAT (/,5X,'The model is set to change parameters.')
759
                 G0 T0 1034
760
761
    1032 WRITE (OUTSCR, 1033)
762
             FORMAT (/,5X,'The model is set to print graphs.')
    1033
763
764
     1034 WRITE (OUTSCR, 115)
765
          CALL YESNO (ANS)
766
          IF (ANS .EQ. 'Y') GO TO 1060
767
768
     1040 WRITE (OUTSCR, 1042)
769
     1042
              FORMAT (/,5X,'What do you wish to do after the generations ',
770
              'are finished?',/,13X,'Enter "CH" to change parameters or ',
771
              '"GR" to graph results: ',$)
772
             READ (IN, 1044, ERR=1046) NEXT
773
     1044
             FORMAT (A2)
774
             IF ((NEXT .EQ. 'CH') .OR. (NEXT .EQ. 'GR')) GO TO 1060
775
     1046
                 WRITE (OUTSCR, 1048)
776
     1048
                 FORMAT (' ****ERROR: Request is incorrect.',/,13X,
777
                 'It must be either "CH" or "GR".')
778
         $
779
                WRITE (OUTSCR, 265)
```

```
GO TO 1040
780
    1060 CONTINUE
781
   C
782
          WRITE (OUTSCR, 1070) NOGENS, SUMRY, NEXT
783
    D
             FORMAT (5X, I4, 'GENERATIONS WILL BE RUN, ',/,5X,
784
    D1070
              'SUMMARIES WILL BE PRINTED EVERY ', I2, ' GENERATIONS', /, 5X,
   D
785
             'THEN: ',A2)
   D
786
787
          PAUSE 'HIT <RET> TO CONTINUE.'
788
   C-
   C
789
790
    C
       GRAPH REQUESTS
791
   C
          IF (NEXT .NE. 'GR') GO TO 1400
792
793
   C
794
          PRINT MENU
     1200 IF (GRF(1) .EQ. ' ') GO TO 1210
795
             WRITE (OUTSCR, 1201)
796
     1201
             FORMAT (/,5X,'Graphs requested: ',$)
797
                 WRITE (OUTSCR, 1202) (GRF(I), I=1,10)
798
     1202
                 FORMAT (10X, 10(A2, 2X))
799
             WRITE (OUTSCR, 115)
             CALL YESNO (ANS)
801
             IF (ANS .EQ. 'Y') GO TO 1400
802
803
     1210 WRITE (OUTSCR, 1212)
804
             FORMAT (/,5X,'You may request up to 10 of the following '
     1212
805
              'graphs:',/,10X,'AF (allele frequency)
                                                              GF (genotype ',
806
             'frequency)')
807
          WRITE(OUTSCR, 1215)
808
             FORMAT(10X, 'PO (population size)
                                                         WW
                                                              (number of ',
809
              'WW''S)',/,10X,'WM
                                   (number of WM''S)
                                                                    (number of ',
                                                              MM
810
              'MM''S)')
811
          WRITE (OUTSCR, 1220)
812
             FORMAT (10X,'AW (allele fitness)
                                                          GW (genotype ',
     1220
813
              'fitness)',/10X,'AS (allele selection coef.) GS (genotype ',
814
              'selection coef.)')
815
    C
816
          ENTER GRAPH REQUESTS
817
          WRITE (OUTSCR, 1230)
818
             FORMAT (/,5X,'Now enter the codes for the graphs you wish '
     1230
819
              'printed',/,13X,'(Hit <RET> twice after your last request): ')
820
          DO 1280 I=1,10
821
             WRITE (OUTSCR, 1240) I
822
     1240
             FORMAT (/,5X,'What is the code for graph number ',I2,'? ',$)
823
             READ (IN, 1250, ERR=1260) GRF(I)
824
             FORMAT (A4)
825
    1250
    C
826
    C
             IF CURRENT REQUEST IS NOT BLANK, CHECK FOR ERRORS, IF IT IS
827
             BLANK, ENSURE THAT REMAINING GRAPH REQUEST SLOTS ARE ALSO
    C
828
             BLANK.
829
             IF (GRF(I) .NE. GRFTST(1)) GO TO 1254
830
                 II = I
831
                 DO 1252 J=II,10
832
                    GRF(I) = GRFTST(1)
833
     1252
                    CONTINUE
834
835
                 G0 T0 1290
836
     1254
             DO 1255 J=2,11
837
                 IF (GRF(I) .EQ. GRFTST(J)) GO TO 1280
838
839
     1255
                 CONTINUE
```

```
840 C
    1260
            WRITE (OUTSCR, 1270) I
841
               FORMAT (' ****ERROR: Graph request # ',I2,' is ',
842
    1270
               'incorrect.',/13X,'It must be one of the following:',/,13X,
843
        $
               '"AF", "GF", "PO", "WW", "WM", "MM",',/,13X, '"AW", "GW", "AS" OR "GS".')
844
845
               WRITE (OUTSCR, 265)
846
               GRF(I) = '
847
               G0 T0 1235
848
   C
849
850
    1280
            CONTINUE
    1290 CONTINUE
851
  || C.
852
   D
         WRITE (OUTSCR, 1300) (GRF(I), I=1,10)
853
854
   D1300 FORMAT (5X, 'THE FOLLOWING GRAPHS WILL BE PRINTED: ',10(2X,A2))
         PAUSE 'TYPE <RET> TO CONTINUE.'
   D
855
   856
857
   C
      PRINT SUMMARY TO SCREEN, ASK IF OK. IF YES, CONTINUE,
858
   C
         IF NO, REDO INPUT
859
   C
860
    1400 PRINT2 = OUTSCR
861
         CALL INPSUM
862
863
864
    1520 WRITE (OUTSCR, 1530)
865
    1530
         FORMAT ('0 Are these data o.k. (Y/N)?',/,5X,
866
           '(If answer is "N", input will restart)',/,5X,
867
           '(If answer is "Y", and nothing happens, be sure printer is',
868
        $ ' on line.): ',$)
869
         CALL YESNO (ANS)
870
         IF (ANS .EQ. 'Y') GO TO 1600
871
            FIX = 'Y'
872
            G0 T0 10
873
            FIX = 'N'
    1600
874
875
   C--
   C
876
       PRINT SUMMARY TO LINE PRINTER
   C
877
         PRINT2 = OUTLP
878
         CALL INPSUM
879
   C-
880
   C
881
    9999 RETURN
882
         END
883
   884
885
   886
   C
         SUBROUTINE YESNO (ANS)
887
888
   C
       FUNCTION: READS AND CHECKS YES/NO INPUT, RETURNS EITHER "Y" OR "N" IN
889
   C
            VARIABLE "ANS".
890
   C
891
   C
       NOTES:
892
          1) THIS ROUTINE RELIES HEAVILY ON THE "ERR=" DEC FORTRAN
   C
893
             EXTENSION TO AVOID AWKWARD SYSTEM ABORTS. THIS WILL NEED
   C
894
             TO BE CHANGED TO IMPROVE TRANSPORTABILITY.
895
   C
896
   C
   C
       CALLED FROM: INPUT, MAIN
897
       CALLS: NOTHING
   C
898
899
```

```
INCLUDE 'COMMON.f'
900
   901
   C
        Type *, 'In YesNo'
902
903
   C
    100 READ (IN,110,ERR=120) ANS
904
          FORMAT (A1)
905
        IF ((ANS .EQ. 'Y') .OR. (ANS .EQ. 'N')) GO TO 140
906
907
    120
           WRITE (OUTSCR, 130)
              FORMAT (5X, *****ERROR: Response is incorrect.',/,13X,
    130
908
              'It must be either a "Y" or an "N".',/,13X,
       $
909
              'Please try again: ',$)
910
        GO TO 100
911
912 | C
913
    140 RETURN
914
        FND
  915
   916
917
        SUBROUTINE INPSUM
918
919
      FUNCTION: PRINTS SUMMARY OF DATA INPUT BY USER.
920
921 || C
  || C
      NOTES:
922
         1) OUTPUT GOES TO SCREEN IF "PRINT2" = "OUTSCR" AND TO LINE
923 C
           PRINTER IF "PRINT2" = "OUTLP"
   C
924
925
   C
926 | C
      CALLED FROM: INPUT
  || C
927
      CALLS: NOTHING
928
   INCLUDE 'COMMON.f'
929
930
   C LOCAL DECLARATIONS
931
        INTEGER ATOT
932
   C============
933
                         Type *,'In Inpsum'
  C
934
935
        IF (IGEN .GT. 1) GO TO 5
936
          WRITE (PRINT2,2)
937
             FORMAT (1H1)
938
      5 IF (PRINT2 .EQ. OUTLP) GO TO 15
939
           WRITE (PRINT2,10)
940
          FORMAT('0 The following is a summary of the data you ',
941
         'have just input:')
942
          G0 T0 18
943
944
     15 IF (IGEN .GT. 1) GO TO 45
945
          WRITE (PRINT2,16)
946
          FORMAT(80(1H-),/,' The following is a summary of ',
     16
947
          'the data you have input: ')
948
949
  C
     18 WRITE (PRINT2,20)TITLE
950
          FORMAT (5X, 'Title: ',/,X,40A2)
951
   C
952
        WRITE (PRINT2,30) IY
953
         FORMAT (5X, 'Random number seed: ',I5)
     30
954
  C
955
        ATOT = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
956
        WRITE (PRINT2,40) ((RSLT(I,1), I=1,3), ATOT)
957
                                       WW = ', I4, 3X, 'WM = ', I4,
          FORMAT (5X, 'Initial population:
     40
958
959
          3X,'MM = ',I4,3X,'TOTAL = ',I4)
```

```
960 | C
       45 WRITE (PRINT2,46) IGEN
961
            FORMAT('0After generation ',I3,' parameters are as follows:')
962
       46
963
         WRITE (PRINT2,50) MXPOP1, MXPOP2
964
       50
            FORMAT (5X, 'Carrying capacity:
                                                ',I4,/,4X,
965
            ' Post-crash pop. size:
                                    ', I4)
966
    C
967
         WRITE (PRINT2,60) (SURVR8(I), I=1,3)
968
            FORMAT (5X, 'Survival rates:
                                                WW = ', F5.1, '% WM = ',
       60
969
970
          F5.1,'% MM = ',F5.1,'%')
971
    C
         WRITE (PRINT2,70) (REPR8(I), I=1,3)
972
973
       70
            FORMAT (5X, 'Reproductive rates:
                                                WW = ',
974
           F4.1,2X,'WM = ',F4.1,2X,'MM = ',F4.1
    C
975
         WRITE (PRINT2,80) INFLO(1), INFLO(2), INFLO(3), OUTFLO(1),
976
            OUTFLO(2), OUTFLO(3)
977
       80
            FORMAT (5X, 'Flow: # in per gen.:
                                               WW = ', I5, 3X, 'WM = ', I5, 3X,
978
                                              WW = ', I4, 3X, 'WM = ',
            'MM =',I5,/,11X,'% out per gen.:
979
            I4,3X,'MM = ',I4)
980
    C
981
         WRITE (PRINT2,90)
982
            FORMAT (/,5X,'Summaries:
                                                  ',$)
983
          IF (SUMRY .GT. 0) GO TO 110
984
            WRITE (PRINT2.100)
985
      100
                FORMAT ('+ First and last generations only')
986
            GO TO 130
987
988
      110 IF (SUMRY .GT. 1) GO TO 120
989
            WRITE (PRINT2,112)
990
      112
               FORMAT ('+ Every generation')
991
            GO TO 130
992
993
      120 WRITE (PRINT2, 122) SUMRY
994
            FORMAT ('+ Every ',I2,' generations')
995
996
      130 WRITE (PRINT2, 140) NOGENS
997
            FORMAT (5X, 'Generations to be run:
                                              'I3,/,5X,
998
            'Then:
999
                                     ',$)
    C
1000
         IF (NEXT .NE. 'CH') GO TO 160
1001
          WRITE (PRINT2, 150)
1002
            FORMAT ('+ Parameters will be changed ',/,80(1H-))
      150
1003
         G0 T0 180
1004
1005
      160 WRITE (PRINT2,170) (GRF(I), I=1,10)
1006
            FORMAT ('+ The following graphs will be printed:',/,28X,
1007
            10(2X,A2),/,80(1H-))
1008
      180 CONTINUE
1009
         RETURN
1010
          END
1011
1012
    1013
1014
1015
    1016
                           vrsn: 29
    C
                                     rvsd: 29 May 81
              file: MA.RE
1017
    \mathbf{C}
1018
                    subroutines reprod, m8few, m8many
```

EVOLVE v1 FORTRAN Source.txt Page 18 of 43 Saved: 2003/04/28, 10:14:09 Printed For: Frank Price

SUBROUTINE REPROD

FUNCTION: CONTROLS MATING AND REPRODUCTION OF POPULATION, RETURNS NUMBER OF JUVENILES OF EACH GENOTYPE IN "JUV" ARRAY.

## SYNOPSIS:

1020 | C

|| C 1022

C

С

C

C 1029

 $\mathbf{C}$ 

|| C

C

C 1036

C

C 1039

C 1040

C 1042

C

C

C 1047

 $\mathbf{C}$ 

C 1050

C

( 1052

 $\mathbf{C}$ 1053

C 1054

C 1056

C 1057

C 1060

C 1061

C 1062

C 1063

C 1064

C 1068

C 1071 1072 C

C 1073

C

C

1021

1023

1024 C C 1025

1026 1027 C

1028

1030 C

1031 C

1032

1033 1034

1035

1037

1038

1041 C

1043

1044

1045

1046 C

1048

1049

1051

1055

1058

1059  $\mathbf{C}$ 

1065

1066  $\mathbf{C}$  $\mathbf{C}$ 

1067

1069

1070 C

1074 C

1075 C

1076

1077

1078 1079 C

- REPROD ASSIGNS THE ADULTS OF THE PREVIOUS GENERATION TO THE "NOADLT" ARRAY.
- 2) AFTER INITIALIZING VARIABLES, "REPROD" CALLS 1 OR 2 OTHER ROUTINES ("M8FEW" OR "M8MANY") WHICH RANDOMLY MATE INDIVIDUALS FROM "NOADLT" AND RETURN THE NUMBER OF EACH OF THE 6 POSSIBLE COMBINATIONS OF MATES IN THE "MATES" ARRAY.
- 2) REPROD THEN CALCULATES THE NUMBER OF YOUNG FROM EACH TYPE OF MATING FROM THE REPRODUCTIVE RATES ("REPR8") OF THE GENOTYPES AND PLACES THEM IN THE "NOYONG" ARRAY).
- 3) FINALLY, THE NUMBER OF YOUNG OF EACH GENOTYPE ARE DETERMINED FROM "NOYONG" ACCORDING TO MENDELIAN RATIOS AND PLACED INTO THE "JUV" ARRAY WHICH IS RETURNED TO MAIN.

## NOTES:

- 1) TWO METHODS ARE USED TO CALCULATE THE NUMBER OF EACH COMBINATION OF MATES BECAUSE NEITHER WORKS WELL OVER THE FULL RANGE OF POPULATION SIZES:
  - A) SUBROUTINE "M8FEW" USES THE PROPORTION OF A GENOTYPE AMONG THE ADULTS TO DETERMINE ITS PROBABILITY OF BEING PICKED AS A PARENT. PAIRS OF PARENTS ARE RANDOMLY SELECTED WITHOUT REPLACEMENT FROM THE "NOADLT" ARRAY AND PLACED INTO THE APPROPRIATE CELL OF THE "MATES" ARRAY UNTIL "NOADLT" IS EMPTY OR N = "CUTOFF" INDIVIDUALS HAVE BEEN PAIRED.
  - B) SUBROUTINE "M8MANY" USES "GNTFR", THE PROPORTION OF EACH GENOTYPE REMAINING IN "NOADLT", TO DIRECTLY CALCULATE THE PROBABLE NUMBER OF EACH MATING COMBINATION.
- 2) METHOD A) IS A MONTE CARLO ALGORITHM TO SIMULATE STOCHASTIC COMBINATIONS IN SMALL POPULATIONS. HOWEVER, ITRESULTS IN PROHIBITIVELY LONG RUNNING TIMES WHEN USED WITH LARGE POPULATIONS. THE RUNNING TIME OF METHOD B) IS ESSENTIALLY CONSTANT REGARDLESS OF POPULATION SIZE, BUT CANNOT PROPERLY SIMULATE MATING IN SMALL POPULATIONS.
- 3) IN SMALL POPULATIONS ONLY "M8FEW" IS CALLED. FOR POPULATIONS LARGER THAN "CUTOFF" INDIVIDUALS, "M8FEW" IS CALLED TO MATE CUTOFF INDIVIDUALS AND "M8MANY" IS CALLED TO MATE THE REST. THE RUN TIME OF EVOLVE COULD BE REDUCED SOMEWHAT BY REDUCING THE SIZE OF "CUTOFF" IN BLOCK DATA (CURRENTLY SET = 200). SUBROUTINE "M8FEW" COULD BE ELIMINATED IF ONLY DETERMINISTIC MATING IS DESIRED; SUBROUTINE M8MANY COULD BE ELIMINATED IF POPULATION LIMITS WERE REDUCED, RUN TIME WERE NOT IMPORTANT &/OR MUCH DRIFT IS DESIRED.
- 4) THE "MATES" ARRAY HOLDS THE NUMBER OF INDIVIDUALS INVOLVED IN EACH OF THE 6 COMBINATIONS OF PARENTS: MATES(1) = # OF WW-WW MATINGS, MATES(2) = # OF WW-WM MATINGS, MATES(3) = # OF WW-WW MATINGS, MATES(6) = # OF WW-WM MATINGS,
  - MATES(5) = # OF WW-WW MATINGS, MATES(6) = # OF WW-WM MATINGS,
- 5) THE "NOYONG" ARRAY HOLDS THE NUMBER OF YOUNG FROM EACH OF THE 6 COMBINATIONS OF PARENTS. ARRAY ELEMENTS CORRESPOND TO THOSE IN THE MATES ARRAY.
- 6) ALTHOUGH THE DATA ARE INTEGER, VARIABLES "JUV" AND "NOYONG" ARE DECLARED TO BE REAL TO AVOID POSSIBLE INTEGER OVERFLOW IN 16-BIT MACHINES. THE VARIABLES POTENTIALLY COULD REACH VALUES OF (REPR8 \* MXPOP1) = (10.0 \* 5000) = 50,000.

```
1080 || C
1081 C
       CALLED FROM: MAIN
        CALLS: M8FEW - RANDOMLY MATES N = "CUTOFF" INDIVIDUALS, RETURNS
1082 || C
                        "MATES" ARRAY
1083
    C
    C
               M8MANY - RANDOMLY MATES REMAINDER OF POPULATION, ADDS THEM
1084
                        TO THE "MATES" ARRAY.
   || C
1085
               GAUSS - RETURNS "V", A PSEUDORANDOM NUMBER WITH GIVEN MEAN
    | C
1086
1087
                        AND STANDARD DEVIATION TO DETERMINE REPRODUCTIVE
                        SUCCESS OF EACH COMBINATION OF MATES.
1088
1089
1090
          INCLUDE 'COMMON.f'
1091
    C LOCAL DECLARATIONS
1092
          INTEGER PGEN
1093
1094
          REAL
   1095
          Type *, 'In Reprod'
    C
1096
1097
    C
      INITIALIZE
1098
          PGEN = IGEN - 1
1099
          D0 10 I=1,6
1100
             MATES(I) = 0
1101
             NOYONG(I) = 0.
1102
      10 CONTINUE
1103
          TOADLT = RSLT(1,PGEN) + RSLT(2,PGEN) + RSLT(3,PGEN)
1104
          D0 20 I=1.3
1105
             JUV(I) = 0.
1106
             NOADLT(I) = RSLT(I,PGEN)
1107
       20 CONTINUE
1108
1109 | C
1110 || C
          TEST FOR ZERO ADULTS. RETURN IF TRUE, CONTINUE IF FALSE
          IF (TOADLT .LE. 0) GO TO 999
1111
1112
1113 || C
          DETERMINE NUMBER OF EACH COMBINATION OF PARENTAL GENOTYPES
1114 || C
1115
          CALL M8FEW
1116
          IF (TOADLT .LT. 1) GO TO 30
1117
          CALL M8MANY
1118
1119
    C-
1120 || C
          DETERMINE NUMBER OF YOUNG FROM EACH TYPE OF MATING: SUBROUTINE
1121 || C
1122 C GAUSS IS CALLED AND RETURNS A PSEUDORANDOM NUMBER ("V") WITH A MEAN
       ("AM") EQUAL TO THE AVERAGE NUMBER OF YOUNG PER PARENT (REPR8(3)).
1123
       THE MEAN NUMBER OF YOUNG IS MULTIPLIED BY THE NUMBER OF PARENTS
1124
       (MATES(6)) TO DETERMINE THE TOTAL NUMBER OF YOUNG (NOYONG(6))
    C
    C
       PRODUCED BY THOSE PARENTS.
1126
   ∥ (
1127
      RNOTE: A BETTER METHOD OF SETTING STANDARD DEVIATION OF GAUSS ("S")
1128 C
      MAY BE NEEDED. THE CURRENT METHOD (S = AM / SQRT(X)) IS ESSENTIALLY
      ANALOGOUS TO THE STANDARD ERROR OF THE MEAN AND ATTEMPTS TO REDUCE
1130 C
       THE VARIATION OF GAUSS' OUTPUT AS THE NUMBER OF INDIVIDUALS
1131 || C
       INCREASES. THIS NEEDS THOUGHT.
1132
   || C
    C
1133
      THIS SECTION OF CODE (DOWN TO STATEMENT 90) COULD BE PUT INTO A
1134 || C
      SUBROUTINE ("NUMYNG") TO IMPROVE PROGRAM CLARITY.
1135
   || C
1136 || C.....
1137 C
          NUMBER OF YOUNG FROM WW-WW MATINGS
1138 | C
       30 IF (MATES(1) .LT. 1) GO TO 40
```

```
AM = REPR8(1)
1140
1141 || D
           WRITE (OUTSCR, 170)
    D 170 FORMAT(100('*'))
1142
    D 100 FORMAT ('0ENTER REPRODUCTION BLOCK WITH: MATES(1-6) = ',6(16,1X),
          1'REPR8(1-3) = ',3(F5.2,1X))
1144
    D
           WRITE (OUTSCR, 100) MATES(1), MATES(2), MATES(3), MATES(4),
1145
          1MATES(5), MATES(6), REPR8(1), REPR8(2), REPR8(3)
1146
1147
           X = MATES(1)
           IF (X .LT. 1.) X = 1.
1148
           S = AM / SQRT(X)
1149
1150
           CALL GAUSS
           NOYONG(1) = MATES(1) * V + 0.5
1151
    D
           WRITE (OUTSCR, 110) NOYONG(1), V, S, X, AM
1152
    D 110 FORMAT('0NOYONG(1) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
          1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(1) = MATES(1) * V + 0.5',
1154
          25X, 'S = AM / SQRT(X)', 5X, 'X = MATES(1) OR 1.0', 5X,
    D
1155
          3' AM = REPR8(1)')
1156
    C......
1157
    C
1158
    C
           NUMBER OF YOUNG FROM WW-WM MATINGS
1159
        40 IF (MATES(2) .LT. 1) GO TO 50
1160
           AM = (REPR8(1) + REPR8(2)) / 2
1161
           X = MATES(2)
1162
           IF (X .LT. 1.) X = 1.
1163
           S = AM / SORT(X)
1164
           CALL GAUSS
1165
           NOYONG(2) = MATES(2) * V + 0.5
1166
           WRITE (OUTSCR, 120) NOYONG(2), V, S, X, AM
1167
    D 120 FORMAT('0NOYONG(2) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1168
          1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(2) = MATES(2) * V + 0.5',
1169
          25X, 'S = AM / SQRT(X)', 5X, 'X = MATES(2) OR 1.0', 5X,
    D
1170
          3' AM = (REPR8(1) + REPR8(2)) / 2')
1171
    C.....
1172
    l (
1173
           NUMBER OF YOUNG FROM WW-MM MATINGS
1174
    C
        50 IF (MATES(3) .LT. 1) GO TO 60
1175
           AM = (REPR8(1) + REPR8(3)) / 2
1176
           X = MATES(3)
1177
           IF (X .LT. 1.) X = 1.
1178
           S = AM / SQRT(X)
1179
           CALL GAUSS
1180
           NOYONG(3) = MATES(3) * V + 0.5
1181
           WRITE (OUTSCR, 130) NOYONG(3), V, S, X, AM
1182
    D 130 FORMAT('0N0Y0NG(3) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1183
          1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(3) = MATES(3) * V + 0.5',
1184
          25X, 'S = AM / SQRT(X)', 5X, 'X = MATES(3) OR 1.0', 5X,
1185
    D
          3' AM = (REPR8(1) + REPR8(3)) / 2')
1186
1187
    l C
1188
           NUMBER OF YOUNG FROM WM-WM MATINGS
1189
        60 IF (MATES(4) .LT. 1) GO TO 70
1190
           AM = REPR8(2)
1191
           X = MATES(4)
1192
           IF (X .LT. 1.) X = 1.
1193
           S = AM / SQRT(X)
1194
1195
           CALL GAUSS
           NOYONG(4) = MATES(4) * V + 0.5
1196
           WRITE (OUTSCR, 140) NOYONG(4), V, S, X, AM
1197
    D 140 FORMAT('0N0YONG(4) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1198
          1'X = ', F5.2, 2X, 'AM = ', F5.2, /, 5X, 'NOYONG(4) = MATES(4) * V + 0.5',
1199
    D
```

```
1200 | D
         25X, 'S = AM / SQRT(X)', 5X, 'X = MATES(4) OR 1.0', 5X,
    D
         3' AM = REPR8(2)
1201
    C.....
1202
1203
    C
          NUMBER OF YOUNG FROM WM-MM MATINGS
1204
       70 IF (MATES(5) .LT. 1) GO TO 80
1205
          AM = (REPR8(2) + REPR8(3)) / 2
1206
          X = MATES(5)
1207
          IF (X .LT. 1.) X = 1.
1208
          S = AM / SQRT(X)
1209
1210
          CALL GAUSS
          NOYONG(5) = MATES(5) * V + 0.5
1211
1212 || D
          WRITE (OUTSCR, 150) NOYONG(5), V, S, X, AM
1213 D 150 FORMAT('0NOYONG(5) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
         1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(5) = MATES(5) * V + 0.5',
1214
    D
         25X, 'S = AM / SQRT(X)', 5X, 'X = MATES(5) OR 1.0', 5X,
1215
    D
         3' AM = (REPR8(2) + REPR8(3)) / 2')
1216
    C.....
1217
    C
1218
    C
          NUMBER OF YOUNG FROM MM-MM MATINGS
1219
       80 IF (MATES(6) .LT. 1) GO TO 90
1220
          AM = REPR8(3)
1221
          X = MATES(6)
1222
          IF (X .LT. 1.) X = 1.
1223
          S = AM / SORT(X)
1224
          CALL GAUSS
1225
          NOYONG(6) = MATES(6) * V + 0.5
1226
   ∥ D
          WRITE (OUTSCR, 160) NOYONG(6), V, S, X, AM
1227
   D 160 FORMAT('0NOYONG(6) = ',F6.0,2X,'V = ',F5.2,2X,'S = ',F5.2,2X,
1228
         1'X = ',F5.2,2X,'AM = ',F5.2,/,5X,'NOYONG(6) = MATES(6) * V + 0.5',
    D
1229
         25X, 'S = AM / SQRT(X)', 5X, 'X = MATES(6) OR 1.0', 5X,
    D
1230
         3' AM = REPR8(3)'
    D
1231
          WRITE (OUTSCR, 170)
1232
    (-
1233
   || C
1234
           CALCULATE NUMBER OF YOUNG OF EACH GENOTYPE FROM THE MENDELIAN
1235
    C
          RATIOS ASSOCIATED WITH PARTICULAR COMBINATIONS OF PARENTAL
1236
    \mathbf{C}
          GENOTYPES
1237
1238
          NUMBER OF WW YOUNG
1239
       90 JUV(1) = NOYONG(1) + 0.5*FLOAT(NOYONG(2)) + 0.25*FLOAT(NOYONG(4))
1240
         1+ 0.5
1241
    C
1242
    C
          NUMBER OF WM YOUNG
1243
          JUV(2) = 0.5*FLOAT(NOYONG(2)) + NOYONG(3) + 0.5*FLOAT(NOYONG(4))
1244
1245
         1 + 0.5 * FLOAT(NOYONG(5)) + 0.5
1246
    C
          NUMBER OF MM YOUNG
1247
          JUV(3) = 0.25*FLOAT(NOYONG(4)) + 0.5*FLOAT(NOYONG(5)) + NOYONG(6)
1248
         1+ 0.5
1249
    C
1250
          type *, 'leaving repro'
    \mathcal{C}
1251
          pause
1252
      999 RETURN
1253
          END
1254
1255
    C
1256
    1257
    C
1258
1259
          SUBROUTINE M8FEW
```

```
1260 || C
       FUNCTION: RANDOMLY PICKS UP TO N = "CUTOFF" INDIVIDUALS AS PAIRS OF
1261 C
          MATES BY A MONTE CARLO ALGORITH AND ADDS THEM TO THE MATES ARRAY.
    C
1262
1263
    C
    C
       NOTES:
1264
          1) SEE NOTES IN REPROD FOR A DISCUSSION OF GENERAL METHOD AND
    C
1265
    C
             RATIONALE.
1266
1267
        CALLED FROM: REPROD
    C
1268
    C
        CALLS: RAN
                    - FORTRAN LIBRARY FUNCTION THAT RETURNS "YFL", A
1269
1270
    C
                      PSEUDORANDOM NUMBER UNIFORMLY DISTRIBUTED 0.0->1.0
1271
    C
                      USED TO DETERMINE WHETHER A GENOTYPE IS CHOSEN FOR A
    C
                      PAIR OF MATES.
1272
    1273
1274
          INCLUDE 'COMMON.f'
1275
    C
      LOCAL DECLARATIONS
1276
          INTEGER COUNT, CUTOFF, IPAR(2), MATE, MAXPRS
1277
          REAL
                   PROB, YFL
1278
          COUNT = 0
1279
          cutoff = 200
1280
    1281
    C
          Type *, 'InM8few'
1282
    C
1283
       INITIALIZE
1284
          MAXPRS = (CUTOFF / 2) + 0.5
1285
          D0 10 I=1,2
1286
1287
             IPAR(I) = 0
       10 CONTINUE
1288
    C
1289
1290
    C
        PICK CUTOFF/2 PAIRS OF MATES
1291
    (-
1292
    C
1293
          Type *, MAXPRS
1294
    C
          Pause 'Before loop'
1295
          D0 500 I=1, MAXPRS
1296
             FORMAT ('0ENTERED M8FEW PICK-A-PAIR LOOP WITH:')
    D 900
1297
    D
             WRITE (OUTSCR,900)
1298
             WRITE (OUTSCR, 910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
    D
1299
    D
         1
            CUTOFF, MAXPRS, COUNT, IPAR(1), IPAR(2), MATES(1), MATES(2),
1300
    D
         2
             MATES(3), MATES(4), MATES(5), MATES(6)
    D 910
             FORMAT (' NOADLT(1,2,3) = ',3(I4,1X),2X,'TOADLT = ',I5,5X,
1302
             'CUTOFF = ', I5, 2X, 'MAXPRS = ', I4, 5X, 'COUNT = ', I4, /,
    D
         1
1303
             ' IPAR(1,2) = ',2(I1,1X),2X,'MATES(1---6) = ',6(I5,1X))
    D
1304
1305
             IF (MAXPRS .LT. 1) GO TO 999
1306
             IF (TOADLT .LE. 0) GO TO 999
       20
1307
    C
1308
1309
            PICK A PAIR
1310
    C.....
1311
1312
             D0 110 J=1,2
1313
                FORMAT ('0IN PICK-A-PAIR LOOP:')
    D 930
1314
1315
    D
                WRITE (OUTSCR, 930)
                IF (TOADLT .EQ. 0) GO TO 110
1316
                IF ((TOADLT .EQ. 1) .AND. (J .EQ. 1)) GO TO 60
1317
1318
    C
1319
                YFL = RAN(IY)
```

```
PROB = FLOAT(NOADLT(1)) / FLOAT(TOADLT)
1320
                  WRITE (OUTSCR, 940) YFL, PROB
1321
    D
                  FORMAT (' YFL = ',F11.9,5X,'PR0B = ',F11.9)
1322
    D 940
                  IF (YFL .GT. PROB) GO TO 30
1324
                  IPAR(J) = 1
                  NOADLT(1) = NOADLT(1) - 1
1325
                  G0 T0 50
1326
     C
1327
        30
                  PROB = PROB + FLOAT(NOADLT(2)) / FLOAT(TOADLT)
1328
    D
                  WRITE (OUTSCR,940) YFL, PROB
1329
1330
                  IF (YFL .GT. PROB) GO TO 40
                  IPAR(J) = 2
1331
                 NOADLT(2) = NOADLT(2) - 1
1332
                  G0 T0 50
    C
1334
        40
                  IPAR(J) = 3
1335
    D
                  WRITE (OUTSCR, 940) YFL, PROB
1336
                  NOADLT(3) = NOADLT(3) - 1
1337
        50
                  TOADLT = TOADLT - 1
1338
                  GO TO 110
1339
1340
1341
                  MATE SOLITARY INDIVIDUAL WITH ITSELF. THIS SECTION
    C
1342
                  OF CODE WAS ADDED TO IMPLEMENT "FACULTATIVE INBREEDING" AND
1343
    C
                  PREVENT STEADY DECLINE OF POPULATION SIZE IN SMALL
1344
                  POPULATIONS WITH ODD NUMBERS OF INDIVIDUALS AND LOW ABSOLUTE
    C
1345
                  FITNESSES. EARLIER VERSIONS OF THE PROGRAM USED "OBLIGATE
1346
    C
                  OUTBREEDING" AND STUDENTS FOUND THE RATHER STEADY DECLINE IN
    C
1347
    C
                  POPULATION SIZE IN SOME RUNS TO BE CONFUSING.
1348
     (
1349
                  IF (NOADLT(1) .NE. 1) GO TO 70
        60
1350
                  MATES(1) = MATES(1) + 1
1351
                  NOADLT(1) = NOADLT(1) - 1
1352
                  G0 T0 90
1353
    C
1354
                  IF (NOADLT(2) .NE. 1) GO TO 80
1355
                 MATES(4) = MATES(4) + 1
1356
                 NOADLT(2) = NOADLT(2) - 1
1357
                  GO TO 90
1358
    C
1359
        80
                  IF (NOADLT(3) .NE. 1) GO TO 100
1360
                  MATES(6) = MATES(6) + 1
1361
                  NOADLT(3) = NOADLT(3) - 1
1362
                  TOADLT = TOADLT - 1
        90
1363
                  G0 T0 120
1364
1365
       100
                  ST0P
1366
    C...
    C
1367
       110
              CONTINUE
1368
              CONTINUE
1369
       120
1370 D 920
              FORMAT ('0EXIT PICK-A-PAIR LOOP WITH:')
              WRITE (OUTSCR, 920)
1371
              WRITE (OUTSCR, 910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
    D
1372
    D
          1
              CUTOFF, MAXPRS, COUNT, IPAR(1), IPAR(2), MATES(1), MATES(2),
1373
    D
              MATES(3), MATES(4), MATES(5), MATES(6)
1374
1375
    C
1376
    C
              ADD PAIR TO MATES ARRAY.
1377
1378
    C
1379
              VALUES OF IPAR ARRAY ARE MULTIPLIED AND THE PRODUCT (MATE)
```

```
1380 C
             IS USED TO DETERMINE WHICH ELEMENT OF THE MATES ARRAY TO
    C
             INCREMENT.
1381
    C
1382
             IF ((IPAR(1) .EQ. 0) .OR. (IPAR(2) .EQ. 0)) GO TO 999
1383
1384
             MATE = IPAR(1) * IPAR(2)
             GO TO (300,310,320,330,210,340,210,210,350), MATE
1385
      210
             ST0P
1386
1387
    C
             WW-WW (1*1=1)
1388
             MATES(1) = MATES(1) + 2
      300
1389
1390
             G0 T0 360
1391
    C
    C
             WW-WM (1*2=2)
1392
      310
             MATES(2) = MATES(2) + 2
1393
             G0 T0 360
1394
    C
1395
    C
             WW-MM (1*3=3)
1396
      320
             MATES(3) = MATES(3) + 2
1397
             G0 T0 360
1398
    C
1399
    C
             WM-WM (2*2=4)
1400
      330
             MATES(4) = MATES(4) + 2
1401
             GO TO 360
1402
    C
1403
             WM-MM (2*3=6)
1404
      340
             MATES(5) = MATES(5) + 2
1405
             G0 T0 360
1406
    C
1407
             MM-MM (3*3=9)
1408
      350
             MATES(6) = MATES(6) + 2
1409
    C
1410
      360
             CONTINUE
1411
             COUNT = COUNT + 1
1412
             D0 400 K = 1,2
1413
1414
                IPAR(K) = 0
      400
             CONTINUE
1415
1416 D 950
             FORMAT ('0FINISHED ADD-A-PAIR WITH MATES(1---6) = ',6(I4,1X))
             WRITE (OUTSCR, 950) MATES(1), MATES(2), MATES(3), MATES(4),
1417
             MATES(5), MATES(6)
1418
      500 CONTINUE
1419
    C--
1420
1421
   || C
      999 CONTINUE
1422
          WRITE (OUTSCR, 960)
1423
    D 960 FORMAT ('0EXIT FROM M8FEW WITH:')
1424
          WRITE (OUTSCR, 910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
             CUTOFF, MAXPRS, COUNT, IPAR(1), IPAR(2), MATES(1), MATES(2),
1426
    D
         2
             MATES(3), MATES(4), MATES(5), MATES(6)
1427
          RETURN
1428
          END
1429
    C
1430
1431
    1432
    C
1433
          SUBROUTINE M8MANY
1434
    C
1435
        FUNCTION: CALCULATES THE NUMBER OF EACH TYPE OF MATES BY METHOD 2
1436
    C
    C
           AND THEN ADDS THEM TO THE MATES ARRAY. SEE NOTES IN REPROD.
1437
    C
1438
    C
1439
        NOTES:
```

EVOLVE v1 FORTRAN Source.txt

Page 25 of 43
Saved: 2003/04/28, 10:14:09

Printed For: Frank Price

```
1440 || C
           1) THE PROBABILITY OF EACH TYPE OF MATING IS DETERMINED BY
1441 C
             MULTIPLYING THE FREQUENCIES OF THE APPROPRIATE GENOTYPES
              "GNTFR" ARRAY, BY EACH OTHER. THE PROBABILITY OF EACH
1442
    C
              COMBINATION OF MATES IS THEN MULTIPLIED BY THE NUMBER OF
1443
              ADULTS LEFT IN "NOADLT", ROUNDED TO THE NEAREST
1444
    C
              INTEGER AND ADDED TO THE NUMBER OF MATES ESTABLISHED BY
    (
1445
              SUBROUTINE "M8FEW".
    С
1446
1447
    C
        CALLED FROM: REPROD
1448
    \mathbf{C}
        CALLS: NOTHING
1449
1450
          INCLUDE 'COMMON.f'
1451
1452
    C LOCAL DECLARATIONS
1453
          INTEGER TMATES
1454
          REAL
                   GNTFR(3)
1455
    C===========
1456
                              ______
    C
          Type *,'In M8Many'
1457
1458
    D 900 FORMAT ('0ENTERED M8MANY WITH:')
1459
    D
          WRITE (OUTSCR, 900)
1460
          WRITE (OUTSCR, 910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
1461
            MATES(1), MATES(2), MATES(3), MATES(4), MATES(5), MATES(6)
1462
    D 910 FORMAT (' NOADLT(1-3) = ',3(I5,1X),' TOADLT = ',I5,
1463
         1 'MATES(1---6) = ',6(I5,1X))
    D
1464
    C
1465
          CALCULATE FREQUENCY OF EACH GENOTYPE
    C
1466
1467
          D0 10 I=1,3
             GNTFR(I) = FLOAT(NOADLT(I)) / TOADLT
1468
       10 CONTINUE
1469
    C
1470
1471
      CALCULATE NUMBER OF EACH TYPE OF MATING.
1472
1473
          MATES(1) = MATES(1) + (GNTFR(1)**2) * TOADLT + 0.5
1474
          MATES(2) = MATES(2) + (2 * (GNTFR(1)*GNTFR(2))) * TOADLT + 0.5
1475
          MATES(3) = MATES(3) + (2 * (GNTFR(1)*GNTFR(3))) * TOADLT + 0.5
1476
          MATES(4) = MATES(4) + (GNTFR(2)**2) * TOADLT + 0.5
1477
          MATES(5) = MATES(5) + (2 * (GNTFR(2)*GNTFR(3))) * TOADLT + 0.5
          MATES(6) = MATES(6) + (GNTFR(3)**2) * TOADLT + 0.5
1479
   l C
1480
         TMATES = MATES(1) + MATES(2) + MATES(3) + MATES(4) + MATES(5) +
1481 || D
           MATES(6)
   ∥ D
1482
          LGEN = IGEN - 1
    D
1483
          ATOT = RSLT(1, LGEN) + RSLT(2, LGEN) + RSLT(3, LGEN)
    D
1484
1485
          IF (TMATES .EQ. ATOT) GO TO 930
1486
          WRITE (OUTSCR,920) TMATES, ATOT
    D 920 FORMAT ('0ERROR: TMATES = ', I5, 1X, 'ATOT = ', I5)
1487
    D 930 WRITE (OUTSCR,940)
    D 940 FORMAT ('0EXIT FROM M8MANY WITH:')
1489
    D
          WRITE (OUTSCR, 910) NOADLT(1), NOADLT(2), NOADLT(3), TOADLT,
1490
            MATES(1), MATES(2), MATES(3), MATES(4), MATES(5), MATES(6)
    D
1491
1492
    C
1493
          RETURN
1494
1495
          END
1496
    1497
1498
1499
```

EVOLVE v1 FORTRAN Source.txt Page 26 of 43 Saved: 2003/04/28, 10:14:09 Printed For: Frank Price

```
1500 C file: SR.SU
                    vrsn: 48
                               rvsd: 27 Aug 81
1501
    C
                    subroutines srvivl, sumrys
1502
    1503
          SUBROUTINE SRVIVL
1504
    (
1505
    С
        FUNCTION: TAKES THE NUMBER OF JUVENILES IN THE CURRENT GENERATION
1506
1507
           AND RETURNS THE NUMBER OF ADULTS IN THE CURRENT GENERATION
    C
           "RSLTS(1-3, IGEN)"
1508
    C
1509
        SYNOPSIS:
1510
    C
1511
   || C
           1) THIS SUBROUTINE USES THE RATES OF EMIGRATION ("OUTFLOW") TO
   || C
              DETERMINE THE NUMBER OF JUVENILES OF EACH GENOTYPE EMIGRATING
1512
1513 C
              ("NOOUT") AND REMOVES THEM FROM THE JUVENILE POPULTION.
1514 | C
           2) IT THEN DETERMINES THE NUMBER OF JUVENILES DYING FROM THE
              RATES OF SURVIVAL ("SURVR8") AND REMOVES THEM FROM THE
   || C
1515
    C
              POPULATION TO DETERMINE THE NUMBER SURVIVING TO ADULTHOOD
1516
               ("ADLT" ARRAY).
1517
    C
           3) THE NUMBER OF IMMIGRANTS ("INFLO") OF EACH GENOTYPE IS ADDED
1518
              TO "ADLT".
    ∥ (
1519
           4) IF THE NUMBER OF ADULTS EXCEEDS THE CARRYING CAPACITY
    ll C
1520
              ("MXPOP1"), THE TOTAL IS REDUCED TO THE POST-CRASH SIZE
1521
    C
              ("MXP0P2").
1522
    C
1523
          NOTES:
    C
1524
             1) VAR. "NOOUT" ADDED & PLACED IN COMMON FOR USE IN DEBUG
    C
1525
                OUTPUT FROM SUBROUTINE SUMRYS. IT WILL NOT BE MEANINGFUL
   Ш С.
1526
   || C
                IF SUMRYS IS CALLED TO PRINT A GENERATION OTHER THAN THE
1527
   || C
                CURRENT ONE. IT SHOULD PROBABLY BE REMOVED IN THE PRODUCTION
1528
    C
                VFRSTON.
1529
             2) ALTHOUGH VARIABLES "ATOT" & "ADLT" ARE CONCEPTUAL INTEGERS,
    \mathbf{C}
1530
    C
                THEY ARE DECLARED REAL TO AVOID POSSIBLE INTEGER OVERFLOW ON
1531
                16-BIT MACHINES. THEORETICALLY, "ADLT" COULD =
    C
1532
                JUV * SURVR8/100 + INFL0 = 50,000 * 100/100 + 2000 = 52,000
    \mathbf{C}
1533
                "ATOT" COULD = 3 * ADLT = 3 * 52,000 = 156,000
1534
    C
1535
        CALLED FROM: MAIN
    C
1536
        CALLS: NOTHING
1537
    C=============
1538
          INCLUDE 'COMMON.f'
1539
1540
    C LOCAL DECLARATIONS
                ADLT(3), ATOT, CRSH
          REAL
1542
    1543
          Type *,'In Survival'
    C
1544
1545
    C
       INITIALIZE
1546
    C
1547
          D0 10 I=1,3
1548
             ADLT(I) = 0
1549
       10 CONTINUE
1550
          ATOT = 0
1551
1552
    C
1553
       EMIGRATION OF JUVENILES
1554
    l C
1555
    C
          RNOTE: COULD CALL GAUSS HERE TO GET RANDOM VARIATION IN %
1556
    C
          EMIGRATING.
1557
1558
    С.
1559
```

```
IF ((OUTFLO(1) .EQ. 0) .AND. (OUTFLO(2) .EQ. 0) .AND.
1560
              (OUTFLO(3) .EQ. 0)) GO TO 100
1561
    C
1562
          D0 40 I=1,3
1563
              IF (OUTFLO(I) .GT. 0.) GO TO 20
1564
             NOOUT(I) = 0
1565
              GO TO 40
1566
    C
1567
              IF (JUV(I) .GT. 0.) GO TO 30
       20
1568
              NOOUT(I) = 0
1569
1570
              GO TO 40
    || C.
1571
       30
             NOOUT(I) = ((JUV(I))*(OUTFLO(I) / 100)) + 0.5
1572
              JUV(I) = JUV(I) - NOOUT(I)
       40 CONTINUE
1574
    C
1575
1576
       CALCULATE TOTAL NUMBER OF ADULTS
1577
    C-
1578
          MULTIPLY NUMBER OF YOUNG OF EACH GENOTYPE BY ITS SURVIVAL RATE,
    C
1579
    C
          ADD IMMIGRANTS
1580
1581
          RNOTE: COULD CALL GAUSS HERE TO GET RANDOM VARIATION IN #
    C
1582
          IMMIGRATING.
1583
1584
    100 DO 110 I=1,3
1585
              ADLT(I) = ((SURVR8(I) / 100) * JUV(I)) + .5 + INFLO(I)
1586
1587
              ATOT = ATOT + ADLT(I)
      110 CONTINUE
1588
   || C
1589
1590
       TEST FOR ADULTS > CARRYING CAPACITY, CRASH IF NECESSARY
1591
1592
           RNOTE: COULD CALL GAUSS HERE TO GET RANDOM VARIATION IN POST-
    \mathbf{C}
1593
          CRASH POPULATION SIZE.
1594
1595
1596
          IF (ATOT .LE. MXPOP1) GO TO 300
1597
          WRITE (OUTSCR, 35) ADLT(1), ADLT(2), ADLT(3), MXPOP2, ATOT
1598
       35 FORMAT ('0ENTER CRASH BLOCK IN SRVIVL WITH: ',5X, 'ADLT(1-3) = ',
    D
1599
            1X,3(15,1X),5X,'MXPOP2 = ',1X,15,5X,'ATOT = ',1X,F6.0
    D
1600
    С
1601
           CRSH = FLOAT(MXPOP2) / FLOAT(ATOT)
1602
           DO 200 I=1,3
1603
              ADLT(I) = ADLT(I) * CRSH + .5
1604
1605
       200 CONTINUE
          WRITE (OUTSCR, 36) ADLT(1), ADLT(2), ADLT(3), MXPOP2, ATOT
1606
       36 FORMAT ('0 EXIT CRASH BLOCK IN SRVIVL WITH: ',5X, 'ADLT(1-3) = ',
1607
    D
            1X,3(15,1X),5X,'MXPOP2 = ',1X,15,5X,'ATOT = ',1X,F6.0
1608
1609
1610
       PUT ADULTS INTO RESULTS ARRAY
    C
1611
1612
    C
1613
      300 DO 310 I=1,3
1614
              RSLT(I,IGEN) = ADLT(I)
1615
      310 CONTINUE
1616
    || C
1617
           RETURN
1618
1619
           END
```

EVOLVE v1 FORTRAN Source.txt

Page 28 of 43
Saved: 2003/04/28, 10:14:09

Printed For: Frank Price

```
1620 || C
1621
    1622
          SUBROUTINE SUMRYS
1624
    C
1625
        FUNCTION: PRINTS A SUMMARY OF FIRST AND LAST GENERATIONS, AND
1627
           OF OTHER GENERATIONS IF REQUESTED. OUTPUT IS DIRECTED TO CRT OR
    C
          LINE PRINTER BY VAR. "PRINT2".
1628
    C
1629
1630
    C
        NOTES:
1631
    C
          1) THIS ROUTINE WAS MUCH USED DURING DEBUGGING AND THERE IS A
   II C
             LOT OF DEBUG CODE. THERE ARE TWO MAJOR SECTIONS OF "D" CODE;
1632
   || C
             THE FIRST IS FOR GENERATIONS WHEN SELECTION & FITNESS
1633
1634
             COEFFICIENTS CAN BE CALCULATED, THE SECOND FOR GENERATIONS
   C
             WHEN THE COEFFICIENTS CANNOT BE CALCULATED.
1635
    C
           2) DEBUG CODE HAS NOT BEEN USE FOR A LONG TIME & MAY NOT BE
1636
             RELIABLE. IN THE PRODUCTION VERSION, DEBUG CODE SHOULD
1637
    C
             PROBABLY BE REMOVED, ALTHOUGH CONDUIT MAY WISH TO CLEAN IT
1638
             UP FOR PURCHASERS' USE.
    (
1639
           3) COULD PROBABLY REMOVE JUVENILES AND GENE FLOW FROM PRODUCTION
    C
1640
             OUTPUT; MIGHT BE SIMPLER FOR STUDENTS.
1641
           4) PUT IN PAGESIZE & #LINESPRINTED VARS. ETC. TO PAGE OUTPUT
    C
1642
    C
1643
        CALLED FROM: MAIN
1644
        CALLS: ALFIT - RETURNS ALLELE FITNESS COEFFICIENTS (DEBUG ONLY)
    C
1645
              ALFREQ - RETURNS ALLELE FREQUENCY OF M ALLELE
1646
   II C.
1647
   I C
              ALSEL - RETURNS ALLELE SELECTION COEFFICIENTS (DEBUG ONLY)
              GNTFIT - RETURNS GENOTYPE FITNESS COEFFICIENTS (DEBUG ONLY)
1648
    C
              GNTSEL - RETURNS GENOTYPE SELECTION COEFFICIENTS (DEBUG ONLY)
1649
1650
         INCLUDE 'COMMON.f'
1651
1652
         LOCAL DECLARATIONS
1653
          INTEGER ATOT, ATOTP, INTOT
1654
                  FRQM, FRQW, OUTTOT, TOJUV
1655
    1656
    C
1657
      PRINT HEADER(S)
1658
    C
1659
         Type *,'In Sumrys'
    C
1660
1661
         WRITE (PRINT2, 10)
1662
       10 FORMAT (/,17X,'#
                             GENE FLOW #
                                                  GNOT.',10X,'ALLELE',/,
1663
           1X, GEN. GNOT. YOUNG # IN # OUT ADULTS FREQ. ALLELE
1664
         $
                      ')
            ' FREQ.
1665
         $
1666
         WRITE (OUTLP, 15)
    D
1667
            FORMAT (50X, 'GNOT. GNOT. ',17X, 'ALL.
    D
                                                       ALL.
1668
                                    #
                                         GNOT. SURV. FITN. SELN.',10X,
1669
                       GENE FLOW
                    SURV. FITN. SELN.',/,1X,'GEN. GNOT. YOUNG # IN ',
1670 D
             ' # OUT ADULTS FREQ. RATES COEF. COEF. ALLELE FREQ.
1671
   ∥ D
            'RATES COEF. COEF.
                                   **** MEANS A COEF.',/,110X,
    D
1672
            'CANNOT BE CALCULATED')
    D
1673
    \mathbf{C}
1674
1675
      INITIALIZE VARIABLES, TEST WHETHER COEFFICIENTS CAN BE CALCULATED
1676
1677
       20 FRQM = ALFREQ(SUMGEN)
1678
1679
         FRQW = 1 - FRQM
```

```
IF ((RSLT(1,SUMGEN) .EQ. 0) .AND. (RSLT(2,SUMGEN) .EQ. 0))
1680
              FROW = 0.
1681
           TOJUV = JUV(1) + JUV(2) + JUV(3)
1682
1683
           ATOT = RSLT(1, SUMGEN) + RSLT(2, SUMGEN) + RSLT(3, SUMGEN)
           INTOT = INFLO(1) + INFLO(2) + INFLO(3)
1684
           OUTTOT = NOOUT(1) + NOOUT(2) + NOOUT(3)
1685
           D0 25 I=1,3
1686
             IF (ATOT .GT. 0) GO TO 24
1687
             OBSFR(I) = 0.
1688
             G0 T0 25
1689
1690
        24
             OBSFR(I) = FLOAT(RSLT(I,SUMGEN)) / FLOAT(ATOT)
        25 CONTINUE
1691
1692
    C
1693
1694
           PRINT SUMMARY
           WRITE(PRINT2,30) SUMGEN, JUV(1), INFLO(1), NOOUT(1), RSLT(1, SUMGEN),
1695
               OBSFR(1), FRQW, JUV(2), INFLO(2), NOOUT(2), RSLT(2, SUMGEN), OBSFR(2),
1696
               JUV(3), INFLO(3), NOOUT(3), RSLT(3, SUMGEN), OBSFR(3), FRQM,
1697
              TOJUV, INTOT, OUTTOT, ATOT
1698
1699
        30 FORMAT(' ',I3,5X,'WW',4X,F6.0,2X,I4,2X,F6.0,3X,I4,3X,F5.3,4X,
1700
               'W',5X,F5.3,/,9X,'WM',4X,F6.0,2X,I4,2X,F6.0,3X,I4,3X,F5.3,
1701
               /,9X,'MM',4X,F6.0,2X,I4,2X,F6.0,3X,I4,3X,F5.3,4X,'M',5X,
1702
              F5.3,/,7X,'TOTAL',3X,F6.0,1X,I5,2X,F6.0,3X,I4,/)
1703
     D
           G0 T0 35
1704
           G0 T0 99
1705
    C
1706
1707
    D-
        CALCULATE ADDITIONAL INITIAL VARS FOR DEBUG OUTPUT;
1708
    D
           TEST FOR CALCULATABILITY OF COEFFICIENTS
1709
    D-
1710
    D
        35 IF (SUMGEN .EQ. 1) GO TO 60
1711
    D
           LGEN = SUMGEN - 1
1712
    D
           ATOTP = RSLT(1, LGEN) + RSLT(2, LGEN) + RSLT(3, LGEN)
1713
    D
           FRQMP = ALFREQ(LGEN)
1714
           IF ((FRQMP .EQ. 0) .OR. (FRQM .EQ. 0) .OR. (ATOTP .EQ. 0)
1715
               .OR. (FRQMP .EQ. 0.) .OR. (FRQMP .EQ. 1.)) GO TO 60
   ∥ D
1716
           CALL GNTFIT (SUMGEN)
1717
    D
           CALL ALFIT (SUMGEN)
    D
1718
           CALL GNTSEL (SUMGEN)
1719
    D
           CALL ALSEL (SUMGEN)
1720
    D
    D
1721
1722
    D
        PRINT DEBUG DATA WHEN FITNESS & SEL. COEFFICIENTS CAN BE CALCULATED
1723
1724
     D-
1725
    D
        40 WRITE(OUTLP,50) SUMGEN,JUV(1),INFLO(1),NOOUT(1),RSLT(1,SUMGEN)
               OBSFR(1),GNTSUR(1),FITGNT(1),GSELCO(1),FRQW,ALSUR(1),FITAL(1),
1726
    D
               ASELCO(1), JUV(2), INFLO(2), NOOUT(2), RSLT(2, SUMGEN), OBSFR(2),
    D
          $
1727
               GNTSUR(2),FITGNT(2),GSELCO(2),JUV(3),INFLO(3),NOOUT(3),
1728
    ∥ D
          $
              RSLT(3, SUMGEN), OBSFR(3), GNTSUR(3), FITGNT(3), GSELCO(3), FRQM,
1729
    D
              ALSUR(2), FITAL(2), ASELCO(2), TOJUV, INTOT, OUTTOT, ATOT
1730
    D
    C
1731
        50 FORMAT('0', I3, 5X, 'WW', 4X, F5.0, 2X, I4, 2X, I5, 3X, I4, 3X, 4(F5.3, 2X),
1732
    D
    D
             2X,'W',3X,4(2X,F5.3,/,9X,'WM',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,
1733
             4(F5.3,2X),/,9X,'MM',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,4(F5.3,2X),
1734
    D
             2X, 'M', 3X, 4(2X, F5.3), /, 7X, 'TOTAL', 3X, F5.0, 2X, I4, 2X, I5, 3X,
1735
    D
          $ I4,/))
1736
    D
           G0 T0 99
    D
1737
    D
1738
1739
     D.
```

EVOLVE v1 FORTRAN Source.txt

Saved: 2003/04/28, 10:14:09

Printed For: Frank Price

```
1740 | D
       PRINT DEBUG DATA WHEN FITNESS & SEL. COEFFICIENTS CAN'T BE CALCULATED
1741
       60 WRITE(OUTLP,70) SUMGEN,JUV(1),INFLO(1),NOOUT(1),RSLT(1,SUMGEN),
1742
    D
1743
    D
            OBSFR(1), FRQW, JUV(2), INFLO(2), NOOUT(2), RSLT(2, SUMGEN), OBSFR(2),
            JUV(3), INFLO(3), NOOUT(3), RSLT(3, SUMGEN), OBSFR(3), FRQM, TOJUV,
1744
    D
   ∥ D
         $ INTOT,OUTTOT,ATOT
1745
   ∥ D
1746
1747
       70 FORMAT('0',I3,5X,'WW',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,F5.3,2X,
         $ '*****',' ***** *****',4X,'W',5X,F5.3,2X,
1748 || D
         $ '**** **** ***** *****',/,9X,'WM',4X,F5.0,2X,I4,2X,I5,3X,I4,3X,
1749
           F5.3,2X,'**** ***** ','*****',/,9X,'MM',4X,F5.0,2X,I4,2X,
1750
    D
1751
    D
           I5,3X,I4,3X,F5.3,2X,'*****','***** *****',4X,'M',5X,F5.3,2X,
           '**** **** ***** *****',/,7X,'TOTAL',3X,F5.0,2X,I4,2X,I5,3X,I4)
1752 || D
1753 | C-
1754 C
       99 RETURN
1755
          FND
1756
1757
    1758
1759
    1760
                           vrsn: 43
                                       rvsd: 27 Aug 81
1761
                     Subroutine graph & miscellaneous subroutines
1762
1763
    1764
          SUBROUTINE GRAPH
1765
1766
   ∥ C.
        FUNCTION: PRINTS GRAPHS OF RESULTS TO LINE PRINTER
1767
   II C.
   || C
1768
    C
        SYNOPSIS:
1769
           1) THE MAIN PART OF THE ROUTINE IS DIVIDED INTO TWO MAJOR
    \mathbf{C}
1770
              SECTIONS WHICH DO GRAPHS OF FLOATING POINT AND INTEGER DATA
1771
              RESPECTIVELY. SUBROUTINE GRAPH FIRST DETERMINES WHETHER THE
    \mathbf{C}
1772
              CURRENT GRAPH IS OF INTEGER OR REAL DATA AND CONTROL JUMPS TO
   II ۲
1773
              THE APPROPRIATE SECTION
1774 || C
           2) WITHIN EACH SECTION AN INITIAL PORTION IS DEVOTED TO FILLING A
1775 | C
              DATA ARRAY WITH THE APPROPRIATE DATA TO BE GRAPHED, AND A
1776 | C
              SECOND PORTION PLOTS THE DATA FROM THE DATA ARRAY.
   ∥ C.
1777
           3) PLOTTING CONSISTS OF COMPOSING A LINE OF PRINT FROM THE DATA
1778
              ARRAY, THEN PRINTING THE LINE.
1779
           4) FROM EACH MAJOR SECTION, CONTROL JUMPS TO ONE SECTION THAT PRINTS
    C
1780
              THE ABSCISSA AND LABEL, THEN RETURNS TO THE TOP OF THE ROUTINE
   || C
1781
              TO REPEAT THE PROCESS.
1782
   l C
   || C
1783
    C
       NOTES:
1784
          1) FACTR IS USED TO SCALE VERTICAL AXIS FOR 50 LINES PER PAGE
1785
          2) "-9" IS USED WHEN COEFFICIENTS CANNOT BE CALCULATED; IT IS
1786
    (
    \mathbf{C}
             USED TO SET PRINTED CHARACTER TO "?".
1787
1788
       NOTES ON POSSIBLE CHANGES:
1789
          1) CHANGE DIMENSIONS FOR GRAPHS TO VARIABLES THAT CAN BE SET BEFORE
1790 | C
             GRAPH IS CALLED. E.G., PGLNTH, PGWDTH, HSCALE, VSCALE
    C
1791
          2) MODIFY TO FLIP FROM ONE GRAPH TO ANOTHER & BACK FOR SCREEN USE
1792
    C
             (USE "PRINT2").
    C
1793
             COULD DISPLAY CHARACTER GRAPH TO SCREEN DIMENSIONS, ASK IF USER
1794
    C
             WANTS HARD COPY TO LINE PRINTER.
1795
    C
          3) MODIFY TO USE SIGGRAPH CORE-79 ROUTINES ON TERAK.
1796
    C
          4) MODIFY TO PRINT OUT LENGTH-WISE FOR TELETYPES?
1797
    C
          5) COULD CLARIFY LOGIC BY BREAKING INTO SUBROUTINES TO, FOR E.G.,
1798
    C
1799
             FILL DATA ARRAYS, COMPOSE A LINE, ETC.
```

EVOLVE v1 FORTRAN Source.txt
Page 31 of 43
Saved: 2003/04/28, 10:14:09
Printed For: Frank Price

```
1800 || C
          6) THIS ROUTINE IS SLOW, IT CANNOT DRIVE THE PRINTRONIX AT FULL
            SPEED. WILL HAVE TO TRY TO SPEED IT UP. THIS MAY CONFLICT
1801 C
    C
            WITH NOTE 5.
1802
1803
    C
       CALLED FROM: MAIN
1804
    C
    C
       CALLS: ALFREQ - RETURNS FREQUENCY OF M ALLELE
1805
    С
              GNTFIT - RETURNS FITNESS COEFFICIENT OF EACH GENOTYPE
1806
1807
    C
              ALFIT - RETURNS FITNESS COEFFICIENT OF EACH ALLELE
    C
              GNTSEL - RETURNS SELECTION COEFFICIENT OF EACH GENOTYPE
1808
              ALSEL - RETURNS SELECTION COEFFICIENT OF EACH ALLELE
1809
1810
1811
         INCLUDE 'COMMON.f'
1812
1813
1814
      LOCAL DECLARATIONS
          INTEGER ATOT, BLNK, CHAR, CHARH, CHARM, CHARW,
1815
            DASH, IDATA(125), IDATUM, LNBTM, LINE(125), LNSIZ, MAX,
1816
            Ν
1817
         REAL DATA(3,125), DATUM1, DATUM2, DATUM3, FACTR, FLNBTM,
1818
         $ FLNSIZ, FLNTOP, FMAX, IFACTR
1819
                  BLNK/''/,
                                  CHAR/'@'/,
                                                     CHARH/'H'/,
         DATA
1820
                  CHARM/'M'/,
                                  CHARQ/'?'/,
                                                     CHARW/'W'/,
1821
                  DASH/'-'/,
                                  IDATA/125*0/
1822
    1823
         Type *,'In Graph'
1824
1825
    C
          INITIALIZE, DETERMINE WHETHER ALL GRAPHS HAVE BEEN FINISHED.
    C
1826
    C
         IF NOT, DETERMINE WHETHER CURRENT GRAPH IS OF FLOATING POINT OR
1827
         OF INTEGER DATA.
1828
1829
         DO 6000 I=1,10
1830
            IF (GRF(I) .EQ. ' ') GO TO 9999
1831
            IF ((GRF(I) .EQ. 'PO') .OR. (GRF(I) .EQ. 'WW') .OR.
1832
            (GRF(I) .EQ. 'WM') .OR. (GRF(I) .EQ. 'MM')) GO TO 2000
1833
    C
1834
    1835
    C FLOATING POINT GRAPHS
1836
1837
    1838
            INITIALIZE, ZERO DATA MATRIX
1839
            FMAX = 1.0
1840
            FACTR = FMAX/50
1841
            D0 20 J=1.3
1842
               DO 10 K=1,125
1843
                  DATA(J,K) = 0.
1844
               CONTINUE
1845
       10
       20
            CONTINUE
1846
    C
1847
            DETERMINE GRAPH REQUESTED
1848
            IF (GRF(I) .EQ. 'AF') GO TO 100
1849
            IF (GRF(I) .EQ. 'GF') GO TO 200
1850
            IF (GRF(I) .EQ. 'GW') GO TO 300
1851
            IF (GRF(I) .EQ. 'AW') GO TO 400
1852
            IF (GRF(I) .EQ. 'GS') GO TO 500
1853
            IF (GRF(I) .EQ. 'AS') GO TO 600
1854
1855
            WRITE (OUTLP,30) I
            FORMAT (1X, '****ERROR: GRAPH REQUEST #', I2, ' IS INCORRECT')
1856
            GO TO 9999
1857
    \mathbf{C}
1858
1859
```

```
1860 || C
           SET PARAMETERS FOR FLOATING POINT GRAPHS
     C.
1861
     C
1862
     C
               SET PARAMETERS FOR ALLELE FREQUENCY GRAPH
1863
       100
               DO 110 K=1, IGEN
1864
                  DATA(2,K) = ALFREQ(K)
1865
                  DATA(1,K) = 1. - DATA(2,K)
1866
                  IF((RSLT(1,K) .EQ. 0) .AND. (RSLT(2,K) .EQ. 0))
1867
          $
                     DATA(1,K) = 0.
1868
                  IF((RSLT(3,K) .EQ. 0) .AND. (RSLT(2,K) .EQ. 0))
1869
1870
                     DATA(2,K) = 0.
       110
               CONTINUE
1871
               GO TO 1000
1872
1873
1874
               SET PARAMETERS FOR GENOTYPE FREQUENCY GRAPH
1875
       200
               DO 230 J=1, IGEN
1876
                  ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
1877
                  IF (ATOT .GT. 0) GO TO 210
1878
                  DATA(1,J) = 0.
1879
                  DATA(2,J) = 0.
1880
                  DATA(3,J) = 0.
1881
                  G0 T0 230
1882
     C
1883
                  DO 220 K=1,3
       210
1884
                     IDATUM = RSLT(K,J)
1885
                     IF (IDATUM .GT. 0 ) GO TO 215
1886
                         DATA (K,J) = 0.
1887
                         G0 T0 220
1888
       215
                     DATA(K,J) = FLOAT(IDATUM) / ATOT
1889
       220
                  CONTINUE
1890
       230
               CONTINUE
1891
               GO TO 1000
1892
1893
     C
1894
               SET PARAMETERS FOR GENOTYPE FITNESS GRAPH
1895
       300
               DATA(1,1) = -9.
1896
               DATA(2,1) = -9.
1897
               DATA(3,1) = -9.
1898
               DO 320 J=2, IGEN
1899
                  LGEN = J - 1
1900
                  FRQMP = ALFREQ(LGEN)
1901
                  ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
1902
                  IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
1903
                  (ATOT .GT. 0)) GO TO 310
1904
     C
1905
                  DATA(1,J) = -9.
1906
                  DATA(2,J) = -9.
1907
                  DATA(3,J) = -9.
1908
                  G0 T0 320
1909
     C
1910
                  CALL GNTFIT(J)
       310
1911
                  DATA(1,J) = FITGNT(1)
1912
                  DATA(2,J) = FITGNT(2)
1913
                  DATA(3,J) = FITGNT(3)
1914
1915
       320
               CONTINUE
1916
               GO TO 1000
1917
1918
1919
     C
```

```
1920 C
               SET PARAMETERS FOR ALLELE FITNESS GRAPH
       400
               DATA(1,1) = -.9
1921
               DATA(2,1) = -.9
1922
     C
1923
               DO 420 J=2, IGEN
1924
                  LGEN = J - 1
1925
                  FRQMP = ALFREQ(LGEN)
1926
                  ATOT = RSLT(1, LGEN) + RSLT(2, LGEN) + RSLT(3, LGEN)
1927
                  IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
1928
                  (ATOT .GT. 0)) GO TO 410
          1
1929
1930
     C
1931
                  DATA(1,J) = -9.
                  DATA(2,J) = -9.
1932
1933
                  G0 T0 420
1934
       410
                  CALL ALFIT(J)
1935
                  DATA(1,J) = FITAL(1)
1936
                  DATA(2,J) = FITAL(2)
1937
1938
       420
               CONTINUE
1939
1940
               GO TO 1000
1941
     C
1942
               SET PARAMETERS FOR GENOTYPE SELECTION COEF. GRAPH
1943
       500
               DATA(1,1) = -9.
1944
               DATA(2,1) = -9.
1945
               DATA(3,1) = -9.
1946
               DO 520 J=2, IGEN
1947
                  LGEN = J - 1
1948
                  FRQMP = ALFREQ(LGEN)
1949
                  ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
1950
                  IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
1951
                  (ATOT .GT. 0)) GO TO 510
          $
1952
     C
1953
                  DATA(1,J) = -9.
1954
                  DATA(2,J) = -9.
1955
                  DATA(3,J) = -9.
1956
                  G0 T0 520
1957
1958
       510
                  CALL GNTSEL (J)
1959
                  DATA(1,J) = GSELCO(1)
1960
1961
                  DATA(2,J) = GSELCO(2)
                  DATA(3,J) = GSELCO(3)
1962
       520
               CONTINUE
1963
               GO TO 1000
1964
1965
1966
     C
     C
               SET PARAMETERS FOR ALLELE SELECTION COEF. GRAPH
1967
       600
               DATA(1,1) = -9.
1968
               DATA(2,1) = -9.
1969
               DO 620 J=2, IGEN
1970
                  LGEN = J - 1
1971
1972
                  FRQMP = ALFREQ(LGEN)
                  ATOT = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
1973
                  IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND.
1974
                  (ATOT .GT. 0)) GO TO 610
1975
     C
1976
                  DATA(1,J) = -9.
1977
                  DATA(2,J) = -9.
1978
1979
                  G0 T0 620
```

```
1980 C
       610
                  CALL ALSEL (J)
1981
                  DATA(1,J) = ASELCO(1)
1982
1983
                  DATA(2,J) = ASELCO(2)
       620
               CONTINUE
1984
              GO TO 1000
1985
1986
     C
1987
               PLOT FLOATING POINT DATA
1988
1989
1990
     C
1991
      1000
              WRITE (OUTLP,5000) TITLE
              WRITE (OUTLP,5002)
1992
1993
     C
1994
               PRINT 50 LINES
               IFACTR=FACTR/2
1995
              DO 1060 J=1,51
1996
1997
     C
                  COMPOSE A LINE
1998
                  FLNSIZ = FMAX - ((J-1)*FACTR)
1999
                  FLNTOP = FLNSIZ + FACTR/2
2000
                  FLNBTM = FLNSIZ - FACTR/2
2001
                  DO 1010 K=1,125
2002
                     LINE(K) = BLNK
2003
      1010
                  CONTINUE
2004
                  IF ((GRF(I) .NE. 'AF') .AND. (GRF(I) .NE. 'AW') .AND.
2005
                  (GRF(I) .NE. 'AS')) GO TO 1030
2006
2007
2008
     C
                  COMPOSE ALLELE COEFFICIENT LINES
2009
                  D0 1020 K = 1, IGEN
2010
                     DATUM1 = DATA(1,K)
2011
                     DATUM2 = DATA(2,K)
2012
                     LINE(K) = BLNK
2013
                     IF (DATUM1 .GE. 0.) GO TO 1011
2014
                     LINE(K) = CHARQ
2015
                     GO TO 1012
2016
                     IF ((DATUM1 .GT. FLNBTM) .AND. (DATUM1 .LE.
      1011
2017
2018
                     FLNTOP)) LINE(K) = CHARW
                     IF (DATUM2 .GE. 0.) GO TO 1013
2019
      1012
                     LINE(K) = CHARQ
2020
2021
                     GO TO 1020
      1013
                     IF ((DATUM2 .GT. FLNBTM) .AND. (DATUM2 .LE.
2022
                     FLNTOP)) LINE(K) = CHARM
2023
      1020
                     CONTINUE
2024
                  IF (J .LT. 51) GO TO 1050
2025
2026
                     DO 1025 K=1,125
                         IF ((DATA(2,K) .LE. 0.) .OR.
2027
2028
                            (DATA(1,K) . LE. 0.)) LINE(K) = DASH
      1025
                         CONTINUE
2029
                     GO TO 1050
2030
2031
2032
                  COMPOSE GENOTYPE COEFFICIENT LINES
     C
2033
                  NOTE THAT H OVERPRINTS M WHICH OVERPRINTS W
2034
     C
      1030
2035
                  D0 1040 K = 1, IGEN
                     DATUM1 = DATA(1,K)
2036
                     DATUM2 = DATA(2,K)
2037
                     DATUM3 = DATA(3,K)
2038
2039
                     LINE(K) = BLNK
```

```
IF (DATUM1 .GE. 0.) GO TO 1031
2040
                   LINE(K) = CHARO
2041
                   G0 T0 1032
2042
                   IF ((DATUM1 .GT. FLNBTM) .AND. (DATUM1 .LE.
2043
     1031
                   FLNTOP)) LINE(K) = CHARW
2044
                   IF (DATUM3 .GE. 0. ) GO TO 1033
     1032
2045
                   LINE(K) = CHARQ
2046
2047
                   G0 T0 1034
     1033
                   IF ((DATUM3 .GT. FLNBTM) .AND. (DATUM3 .LE.
2048
                   FLNTOP)) LINE(K) = CHARM
2049
         $
2050
     1034
                   IF (DATUM2 .GE. 0. ) GO TO 1035
2051
                   LINE(K) = CHARQ
                   GO TO 1040
2052
2053
     1035
                   IF ((DATUM2 .GT. FLNBTM) .AND. (DATUM2 .LE.
2054
                   FLNTOP)) LINE(K) = CHARH
     1040
                   CONTINUE
2055
                IF (J .LT. 51) GO TO 1050
2056
                   DO 1045 K=1,125
2057
                       IF ((DATA(2,K) .LE. 0.) .AND. ((DATA(1,K)
2058
                          .LE. 0.) .OR. (DATA(3,K) .LE. 0.))) LINE(K) = DASH
2059
     1045
2060
                       CONTINUE
2061
    C............
    C
2062
                PRINT A LINE
2063
                WRITE (OUTLP, 1052) FLNSIZ, LINE
     1050
2064
                FORMAT (1X,F5.3,'+',125A1)
     1052
2065
     1060
             CONTINUE
2066
    C
2067
             GO TO 4000
2068
    C
2069
    C
2070
2071
      INTEGER PLOTS (POP, WW, WM, MM)
2072
    2073
    C
2074
             ZERO DATA MATRIX
2075
     2000
             DO 2010 K=1,125
2076
                IDATA(K) = 0
2077
     2010
             CONTINUE
2078
             IF (GRF(I) .NE. 'PO') GO TO 2200
2079
    C
2080
2081
    C-
          SET PARAMETERS FOR INTEGER PLOTS
2082
    C
    C-
2083
    C
2084
2085
    C
             SET PARAMETERS FOR POPULATION SIZE GRAPH
2086
    C
    C
             DETERMINE MAXIMUM POPULATION SIZE DURING RUN, IF < 50, SET = 50
2087
             (I.E. ONE INDIVIDUAL PER GRAPH LINE)
2088
    C
             MAX = RSLT(1,1) + RSLT(2,1) + RSLT(3,1)
2089
             IDATA(1) = MAX
2090
             DO 2100 J=1, IGEN
2091
                IDATA(J) = RSLT(1,J) + RSLT(2,J) + RSLT(3,J)
2092
                IF (MAX .LT. IDATA(J)) MAX = IDATA(J)
2093
     2100
             CONTINUE
2094
             FACTR = FLOAT(MAX)/50
2095
             IF (MAX .GT. 50) GO TO 2110
2096
             FACTR = 1.
2097
             MAX = 50
2098
2099
     2110
             GO TO 3000
```

```
2101
               SET PARAMETERS FOR GENOTYPE NUMBERS GRAPHS
    C
2102
    C
2103
     C
              ZERO DATA MATRIX
2104
     2200
              DO 2210 K=1,125
2105
2106
                  IDATA(K) = 0
2107
     2210
              CONTINUE
    C
2108
              SET GENOTYPE FOR GRAPH
2109
2110
              N = 0
2111
              IF (GRF(I) .NE. 'WW') GO TO 2220
              N = 1
2112
2113
              GO TO 2250
2114
              IF (GRF(I) .NE. 'WM') GO TO 2230
     2220
2115
              N = 2
2116
              G0 T0 2250
2117
2118
     2230
              IF (GRF(I) .NE. 'MM') GO TO 2240
2119
2120
              N = 3
              G0 T0 2250
2121
     2240
              WRITE (OUTLP,30) I
2122
              G0 T0 9999
2123
2124
              DETERMINE MAXIMUM NUMBER OF GENOTYPE DURING RUN, IF MAX < 50,
2125
     C
              SET = 50 (I.E., ONE INDIVIDUAL PER GRAPH LINE)
2126
     C
     2250
              MAX = RSLT(N,1)
2127
              DO 2260 J=1, IGEN
2128
                  IDATA(J) = RSLT(N,J)
2129
                  IF (MAX LT. IDATA(J)) MAX = IDATA(J)
2130
      2260
               CONTINUE
2131
               FACTR = FLOAT(MAX)/50
2132
     C
2133
               IF (MAX .GT. 50) GO TO 3000
2134
               FACTR = 1.
2135
              MAX = 50
2136
     C
2137
2138
              PLOT INTEGER DATA
2139
     C
     C-
2140
2141
     C
     3000
              WRITE (OUTLP,5000) TITLE
2142
              WRITE (OUTLP,5002)
2143
     C
2144
               PRINT 50 LINES
2145
     C
              DO 3040 J=1,51
2146
     C
2147
                  COMPOSE A LINE
2148
                  LNSIZ = MAX - ((J-1)*FACTR) + 0.5
2149
                  LNBTM = LNSIZ - FACTR/2
2150
                  DO 3010 K=1,125
2151
2152
                     LINE(K) = BLNK
      3010
                  CONTINUE
2153
                  DO 3020 K=1,IGEN
2154
                     IF (IDATA(K) .GT. LNBTM) LINE(K) = CHAR
2155
      3020
                  CONTINUE
2156
                  IF (J .LT. 51) GO TO 3030
2157
                     DO 3025 K=1,125
2158
2159
                        IF (IDATA(K) .EQ. 0) LINE(K) = DASH
```

```
3025
                       CONTINUE
2160
2161 | C
    C
                PRINT A LINE
2162
                WRITE (OUTLP, 3032) LNSIZ, LINE
2163
     3030
     3032
                FORMAT (1X, I4, '+', 125A1)
2164
     3040
             CONTINUE
2165
             GO TO 4000
2166
2167
    2168
    C PRINT ABSCISSA & LABEL FOR ALL PLOTS
2169
2170
    2171 | C
     4000
             WRITE(OUTLP,5002)
2172
2173
             WRITE (OUTLP,5003)
2174
             WRITE (OUTLP,5004)
             WRITE (OUTLP,5009)
2175
             IF (GRF(I) .NE. 'AF') GO TO 4010
2176
             WRITE (OUTLP,5010)
2177
             GO TO 4110
2178
     4010
             IF (GRF(I) .NE. 'GF') GO TO 4020
2179
             WRITE (OUTLP,5020)
2180
             WRITE (OUTLP,5110)
2181
             GO TO 4110
2182
     4020
             IF (GRF(I) .NE. 'GW') GO TO 4030
2183
             WRITE (OUTLP,5030)
2184
             WRITE (OUTLP,5110)
2185
             WRITE (OUTLP,5120)
2186
2187
             GO TO 4110
     4030
             IF (GRF(I) .NE. 'AW') GO TO 4040
2188
             WRITE (OUTLP,5040)
2189
             WRITE (OUTLP,5120)
2190
             GO TO 4110
2191
     4040
             IF (GRF(I) .NE. 'GS') GO TO 4050
2192
             WRITE (OUTLP,5050)
2193
2194
             WRITE (OUTLP,5110)
             WRITE (OUTLP,5120)
2195
             GO TO 4110
2196
     4050
             IF (GRF(I) .NE. 'AS') GO TO 4060
2197
             WRITE (OUTLP,5060)
2198
             WRITE (OUTLP,5120)
2199
             GO TO 4110
2200
             IF (GRF(I) .NE. 'PO') GO TO 4070
     4060
2201
             WRITE (OUTLP,5070)
2202
             GO TO 4110
2203
             IF (GRF(I) .NE. 'WW') GO TO 4080
     4070
2204
2205
             WRITE (OUTLP,5080)
2206
             GO TO 4110
     4080
             IF (GRF(I) .NE. 'WM') GO TO 4090
2207
2208
             WRITE (OUTLP,5090)
             GO TO 4110
2209
     4090
             IF (GRF(I) .NE. 'MM') GO TO 4100
2210
     4100
             WRITE (OUTLP,5100)
2211
2212
     4110
             WRITE (OUTLP,5009)
    C
2213
     5000
             FORMAT ('1',20X,40A2)
2214
             FORMAT (6X,'+',12('----+'),'----+')
     5002
2215
             FORMAT (16X,'1',9X,'2',9X,'3',9X,'4',9X,'5',9X,'6',9X,'7',9X,
     5003
2216
                 '8',9X,'9',9X,'10',8X,'11',8X,'12')
2217
             FORMAT (45X, 'GENERATION (X10)',/)
     5004
2218
2219
     5009
             FORMAT (20X,100('*'))
```

```
5010
            FORMAT (25X, 'FREQUENCY OF W AND M ALLELES')
2220
     5020
            FORMAT (25X, 'FREQUENCY OF EACH GENOTYPE')
2221
            FORMAT (25X, 'RELATIVE FITNESS OF EACH GENOTYPE')
2222
     5030
            FORMAT (25X, 'RELATIVE FITNESS OF W AND M ALLELES')
2223
     5040
            FORMAT (25X, 'SELECTION COEFFICIENTS OF GENOTYPES')
2224
     5050
     5060
            FORMAT (25X, 'SELECTION COEFFICIENTS OF W AND M ALLELES')
2225
            FORMAT (25X, 'POPULATION SIZE')
     5070
2226
     5080
            FORMAT (25X, 'NUMBER OF WW GENOTYPE')
2227
            FORMAT (25X, 'NUMBER OF WM GENOTYPE')
     5090
2228
            FORMAT (25X, 'NUMBER OF MM GENOTYPE')
     5100
2229
2230
     5110
            FORMAT ('+',10X,'W = WW, H = WM, M = MM')
2231
     5120
            FORMAT ('+',', ? = UNDEFINABLE COEFF')
   || C.
2232
    6000 CONTINUE
2233
2234
         WRITE (OUTLP,6010)
    6010
            FORMAT (1H1)
2235
     9999 RETURN
2236
          END
2237
2238
2239
    2241
          SUBROUTINE GNTFIT (DUMY)
2242
    C
2243
        FUNCTION: RETURNS THE RELATIVE FITNESS COEFFICIENT OF EACH GENOTYPE
2244
2245
    C
          ("FITGNT').
2246 C
       NOTES:
   || C
2247
2248 C
          1) THE METHODS USED FOR THIS AND OTHER COEFFICIENT SUBROUTINES
   C
             ARE DETAILED ON PAGE 52 IN: WILSON, E. O., AND W. H. BOSSERT
2249
             (1971) "A PRIMER OF POPULATION BIOLOGY". STAMFORD, CONN.,
    C
2250
    C
             SINAUER ASSOC.
2251
    C
2252
       CALLED FROM: SUMRYS, GRAPH, GNTSEL
    \mathbf{C}
2253
       CALLS: ALFREQ - RETURNS FREQUENCY OF M ALLELE
2254
2255
    INCLUDE 'COMMON.f'
2256
2257
           LOCAL DECLARATIONS
2258
         REAL
               EXPFRM, EXPFRW,
                                EXPFR(3)
2259
         INTEGER
                        ATOT,
                                DUMY
2260
    2261
         Type *,'In Gntfit'
    C
2262
    C
2263
         TEST THAT COEFFICIENTS CAN BE CALCULATED
2264
2265
          IF (IGEN .NE. 1) GO TO 20
2266
         WRITE (OUTLP, 10)
       10 FORMAT (' *** WARNING: ENTRY INTO GNTFIT WITH IGEN = 1')
2267
         G0 T0 999
2268
       20 ATOT = (RSLT(1,DUMY)+RSLT(2,DUMY)+RSLT(3,DUMY))
2269
         LGEN = DUMY - 1
2270
         FRQMP = ALFREQ(LGEN)
2271
         IF ((FRQMP .GT. 0.) .AND. (FRQMP .LT. 1.) .AND. (ATOT .GT. 0))
2272
            GO TO 40
2273
         WRITE (OUTLP,30)
2274
       30 FORMAT (' *****WARNING: ENTRY INTO GNTFIT WITH M ALLELE FREQ.'
2275
             ' IN PREVIOUS GEN = 0 OR 1, OR TOTAL # ADULTS IN PRESENT GEN',
2276
            ' = 0')
2277
         GO TO 999
2278
2279
```

```
2280 C
           INITIALIZE VARIABLES
2281
        40 DO 50 J=1,3
2282
2283
              FITGNT(J) = 0.
2284
              OBSFR(J) = 0.
              EXPFR(J) = 0.
2285
        50 CONTINUE
2286
2287
    C
    C
           CALCULATE OBSERVED GENOTYPE FREQUENCY
2288
           D0 60 J=1,3
2289
2290
              OBSFR(J) = FLOAT(RSLT(J,DUMY))/(RSLT(1,DUMY)+
              RSLT(2,DUMY)+RSLT(3,DUMY))
2291
        60 CONTINUE
2292
2293
   || C
2294
    C
           CALCULATE EXPECTED FREQUENCY OF EACH ALLELE (BY THE HARDY-WEINBERG
    C
           LAW, THE EXPECTED ALLELE FREQUENCIES EQUAL THE FREQUENCIES IN THE
2295
    \mathbf{C}
           PREVIOUS GENERATION). THE EXPECTED GENOTYPE FREQUENCIES (BY THE
2296
     C
           HARDY-WEINBERG LAW) = P*P, 2PQ, & Q*Q FOR WW, WM & MM
2297
     C
           RESPECTIVELY.
2298
           EXPFRM = 0.
2299
           EXPFRW = 0.
2300
           LGEN = DUMY - 1
2301
    C
2302
           EXPFRM = ALFREQ(LGEN)
2303
           EXPFRW = 1 - EXPFRM
2304
     C
2305
           EXPFR(1) = EXPFRW * EXPFRW
2306
           EXPFR(2) = 2 * EXPFRW * EXPFRM
2307
           EXPFR(3) = EXPFRM * EXPFRM
2308
    C
2309
     C
           CALCULATE GENOTYPE SURVIVAL RATE, OR ABSOLUTE FITNESS OF EACH
2310
           GENOTYPE
2311
           D0 70 J=1.3
2312
              GNTSUR(J) = OBSFR(J) / EXPFR(J)
2313
        70 CONTINUE
2314
    C
2315
           DETERMINE WHICH GENOTYPE HAS THE HIGHEST SURVIVAL RATE, CALCULATE
    C
2316
    \mathbf{C}
           RELATIVE FITNESS COEFFICIENTS BY DIVIDING SURVIVAL RATE OF EACH
2317
           GENOTYPE BY THE HIGHEST SURVIVAL RATE
2318
           IF ((GNTSUR(2) .GE. GNTSUR(1)) .AND. (GNTSUR(2) .GE. GNTSUR(3)))
2319
              GO TO 90
2320
           IF ((GNTSUR(3) .GE. GNTSUR(1)) .AND. (GNTSUR(3) .GE. GNTSUR(2)))
2321
              GO TO 100
2322
    C
2323
    C
           WW HAS HIGHEST SURVIVAL RATE
2324
2325
           ((GNTSUR(1) .GE. GNTSUR(2)) .AND. (GNTSUR(1) .GE. GNTSUR(3)))
2326
        80 FITGNT(1) = 1.0
           FITGNT(2) = GNTSUR(2) / GNTSUR(1)
2327
           FITGNT(3) = GNTSUR(3) / GNTSUR(1)
2328
           G0 T0 999
2329
    C
2330
           WM HAS HIGHEST SURVIVAL RATE
2331
        90 \text{ FITGNT}(2) = 1.0
2332
           FITGNT(1) = GNTSUR(1) / GNTSUR(2)
2333
           FITGNT(3) = GNTSUR(3) / GNTSUR(2)
2334
2335
           G0 T0 999
2336
           MM HAS HIGHEST SURVIVAL RATE
2337
       100 \text{ FITGNT}(3) = 1.0
2338
2339
           FITGNT(1) = GNTSUR(1) / GNTSUR(3)
```

```
FITGNT(2) = GNTSUR(2) / GNTSUR(3)
2340
    C
2341
      999 RETURN
2342
2343
          END
2344
    2345
    2347
          SUBROUTINE ALFIT (DUMY)
2348
    C
2349
2350
        FUNCTION: RETURNS THE RELATIVE FITNESS COEFFICIENT OF EACH ALLELE
2351
   C
           ('FITAL").
2352 C
2353 | C
        NOTES:
2354 C
           1) FOR METHODOLOGY, SEE NOTE IN SUBROUTINE GNTFIT
   C
2355
    C
        CALLED FROM: SUMRYS, GRAPH, ALSEL
2356
        CALLS: ALFREQ - RETURNS FREQUENCY OF M ALLELE
2357
2358
          INCLUDE 'COMMON.f'
2359
2360
    C LOCAL DECLARATIONS
2361
          INTEGER ATOT, DUMY
2362
    C==========
2363
          Type *,'In Alfit'
2364
          TEST THAT COEFFICIENT CAN BE CALCULATED
2365
          IF (IGEN .GT. 1) GO TO 10
2366
2367
          WRITE (OUTLP,1)
             FORMAT (' ***WARNING: ENTRY INTO ALFIT WITH IGEN = 1')
2368
          GO TO 99
2369
       10 LGEN = DUMY - 1
2370
          ATOT = RSLT(1, LGEN) + RSLT(2, LGEN) + RSLT(3, LGEN)
2371
          ALFRMP = ALFREQ(LGEN)
2372
          IF ((ALFRMP .NE. 0.) .AND. (ALFRMP .NE. 1.) .AND. (ATOT .NE. 0))
2373
            G0 T0 30
2374
          WRITE (OUTLP, 20)
2375
       20 FORMAT (' ***WARNING: ENTRY INTO ALFIT WITH FREQ. OF M IN '
2376
             'PREVIOUS GENERATION = 0. OR 1., OR TOTAL # ADULTS = 0.')
2377
          G0 T0 99
2378
2379
          INITIALIZE VARIABLES
2380
       30 ALSUR(1) = 0.
2381
          ALSUR(2) = 0.
2382
          FITAL(1) = 0.
2383
          FITAL(2) = 0.
2384
2385
          CALCULATE ALLELE SURVIVAL RATES
2386
          ALSUR(1) = (1-ALFREQ(DUMY)) / (1-ALFREQ(LGEN))
2387
          ALSUR(2) = (ALFREQ(DUMY)) / (ALFREQ(LGEN))
2388
2389
    C
          DETERMINE WHICH ALLELE HAS THE HIGHEST SURVIVAL RATE
2390
          IF (ALSUR(1) .LT. ALSUR(2)) GO TO 40
2391
    C
2392
          ALLELE W HAS THE HIGHEST SURVIVAL RATE
2393
          FITAL(1) = 1.0
2394
          FITAL(2) = ALSUR(2) / ALSUR(1)
2395
          G0 T0 99
2396
   || C
2397
          ALLELE M HAS THE HIGHEST SURVIVAL RATE
2398
2399
       40 \text{ FITAL}(2) = 1.0
```

```
FITAL(1) = ALSUR(1) / ALSUR(2)
2400
   C
2401
      99 RETURN
2402
2403
        END
   C
2404
   2405
   2407
        SUBROUTINE ALSEL (DUMY)
2408
   C
2409
2410
      FUNCTION: RETURNS THE SELECTION COEFFICIENT OF EACH ALLELE
2411
   C
         ("ASELCO").
2412 | C
2413 C
      NOTES:
2414 | C
         1) FOR METHODOLOGY, SEE NOTE IN SUBROUTINE GNTFIT
  || C
         2) THIS SUBROUTINE COULD BE INCORPORATED INTO ALFIT
2415
   C
2416
   C
      CALLED FROM: SUMRYS, GRAPH
2417
      CALLS: ALFIT - RETURNS FITNESS COEFFICIENT OF EACH ALLELE
2418
2419
   INCLUDE 'COMMON.f'
2420
2421
   C LOCAL DECLARATIONS
2422
         INTEGER DUMY
2423
2424
        Type *,'In Alsel'
2425
   C
   II C
2426
        ALLELE SELECTION COEFFICIENT = 1 - ALLELE FITNESS COEFFIENT
2427
        CALL ALFIT (DUMY)
2428
        DO 10 J=1,2
2429
           ASELCO(J) = 1 - FITAL(J)
2430
      10 CONTINUE
2431
        RETURN
2432
        END
2433
   \mathbf{C}
2434
   2435
   2436
   C
2437
        SUBROUTINE GNTSEL (DUMY)
2438
2439
   C
      FUNCTION: RETURNS THE SELECTION COEFFICIENT OF EACH GENOTYPE
   C
2440
  || C
         ("GSELCO").
2441
2442
   l C
   C
      NOTES:
2443
   C
         1) FOR METHODOLOGY, SEE NOTE IN SUBROUTINE GNTFIT
2444
         2) THIS SUBROUTINE COULD BE INCORPORATED INTO GNTFIT
2445
   C
2446
   C
   C
      CALLED FROM: SUMRYS, GRAPH
2447
      CALLS: GNTFIT - RETURNS FITNESS COEFFICIENT OF EACH GENOTYPE
2449
   INCLUDE 'COMMON.f'
2450
2451
   C LOCAL DECLARATIONS
2452
        INTEGER DUMY
2453
   2454
        Type *,'In Gntsel'
2455
   C
2456
        GENOTYPE SELECTION COEFFICIENT = 1 - GENOTYPE FITNESS COEFFICIENT
   C
2457
        CALL GNTFIT (DUMY)
2458
2459
        D0 10 J=1,3
```

```
GSELCO(J) = 1 - FITGNT(J)
2460
      10 CONTINUE
2461
         RETURN
2462
2463
         END
2464
    2465
    C
2466
2467
         FUNCTION ALFREQ (DUMY)
    C
2468
    C
       FUNCTION: FREQUENCY OF MUTANT ALLELE M (ALFREQ) IN SPECIFIED
2469
2470
    C
          GENERATION
2471
    C
   | C
       NOTES:
2472
2473
   || C
          1)
2474
2475
   C
       CALLED FROM: SRVIVL, GRAPH, GNTFIT
    \mathbf{C}
       CALLS: NOTHING
2476
2477
                           ______
         INCLUDE 'COMMON.f'
2478
2479
     LOCAL DECLARATIONS
2480
         INTEGER DUMY
2481
    2482
         Type *,'In Alfreg'
    C
2483
2484
         VARIABLE ALFREO IS THE ALLELE FREOUENCY OF THE M ALLELE
2485
    C
         IF THERE ARE NO WM OR MM ADULTS, SET ALFREQ=0, OTHERWISE COMPUTE
    C
2486
    C
2487
         ALFREQ
         IF ((RSLT(2,DUMY) .EQ. 0) .AND. (RSLT(3,DUMY) .EQ. 0)) GO TO 10
2488
    C
2489
         ALFREQ = (RSLT(2,DUMY) + (2*FLOAT(RSLT(3,DUMY)))) /
2490
        $ (2*((RSLT(1,DUMY)+RSLT(2,DUMY)+RSLT(3,DUMY))))
2491
         G0 T0 20
2492
2493
      10 ALFREQ = 0.
2494
      20 RETURN
2495
         END
2496
    \mathbf{C}
2497
2498
    2499
2500
         SUBROUTINE GAUSS
2501
    C
2502
       FUNCTION: RETURNS FLOATING POINT NUMBER "V" WITH MEAN = AM AND
    C
2503
    C
          STD. DEV. = S
2504
2505
    C
       NOTES:
2506
    C
          1) THIS ALGORITHM WAS PROVIDED BY A COMPUTER PERSON WHO HAS
    \mathcal{C}
2507
2508
   C
             SINCE MOVED. HE DID NOT PROVIDE DOCUMENTATION, OTHER THAN
             A NOTE TO SEE VOL. 2 OF KNUTH'S "ART OF COMPUTER PROGRAMMING"
2509
   C
          2) BEYOND AN AWARENESS THAT RANDOM NUMBER ALGORITHMS ARE A
2510
             TARPIT FOR THE UNWARY, I AM NOT KNOWLEDGABLE IN THIS AREA.
   C
2511
             I CANNOT VOUCH FOR THE VALIDITY OR TRANSPORTABILITY OF THIS
    C
2512
             ALGORITHM. I HOPE CONDUIT CAN HELP BY VALIDATING THIS
    C
2513
             ALGORITHM OR DONATING A MORE TRANSPORTABLE ONE.
    C
2514
2515
   l C
       CALLED FROM: REPROD
2516
   C
    C
       CALLS: RAN
                   - FORTRAN LIBRARY FUNCTION, RETURNS RANDOM NUMBERS
2517
    C
2518
                     UNIFORMLY DISTRIBUTED FROM 0.0->1.0
    C
2519
              SQRT - FORTRAN LIBRARY FUNCTION, RETURNS SQUARE ROOT
```

```
ALOG - FORTRAN LIBRARY FUNCTION, RETURNS NATURAL LOGARITHM
2520 C
INCLUDE 'COMMON.f'
2522
2523
2524 C
        LOCAL DECLARATIONS
        REAL Q, U1, U2, V1, V2, X1
2525
Type *,'In Gauss'
2527 | C
2528 C
        Type *, IY
2529 | C
        Type *, IZ
2530 C
        Pause
2531 C
2532
     100 \text{ U1} = RAN(IY)
2533
        U2 = RAN(IY)
2534
        V1 = 2 * U1 - 1
        V2 = 2 * U2 - 1
2535
        Q = V1 * V1 + V2 * V2
2536
        IF (Q .GE. 1) GO TO 100
2537
        X1 = V1 * SQRT(-2 * ALOG(Q) / Q)
2538
        V = X1 * S + AM
2539
        RETURN
2540
        END
2541
2542 C
2544
2545
        file: COMN vrsn: 14 rvsd: 13 Aug 81
2546 C
2547 || C-----
2548 C GLOBAL DECLARATIONS
        COMMON /A/ ALSUR(2), AM, ASELCO(2), FITAL(2),
2549
          FITGNT(3), FRQMP, GNTSUR(3), GRF(10), GSELCO(3), IGEN,
2550
           IN, INFLO(3), IY, IZ, JUV(3), LGEN, MATES(6), MXPOP1, MXPOP2, NEXT, NOADLT(3), NOGENS, NOOUT(3), NOYONG(6),
2551
2552
          OBSFR(3), OUTFLO(3), OUTLP, OUTSCR, PRINT2, REPR8(3),
2553
        5 RSLT(3,125), S, SUMGEN, SUMRY, SURVR8(3), TITLE(40)
2554
        COMMON /B/ TOADLT, V
2555
2556 C
        INTEGER IGEN, IN, INFLO, IY, IZ,
2557
        1 LGEN, MATES, MXPOP1, MXPOP2, NOADLT, NOGENS,
2558
           OUTFLO, OUTLP, OUTSCR, PRINT2, RSLT, SUMGEN,
2559
        3 SUMRY, TOADLT
2560
2561 C
        REAL ALSUR, AM, ASELCO, FITAL, FITGNT, FRQMP, GNTSUR,
2562
        1 GSELCO, JUV, NOOUT, NOYONG, OBSFR, REPR8, S, SURVR8, V
2563
2564
        STRING*2 ANS, GRF, NEXT
2565
2566
        STRING*80 TITLE
2567
2568
```