UNITED STATES DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

PROGRAM LIST OF ANALYST,
A COMPUTER PROGRAM FOR CONTROL OF AN
ISOMASS 54E THERMAL-IONIZATION, SINGLE-
COLLECTOR MASS-SPECTROMETER

by

K. R. LUDWIG

OPEN-FILE REPORT 85-142

This report is preliminary and has not been reviewed for
conformity with U.S. Geological Survey editorial standards and
nomenclature. Any use of trade names is for descriptive
purposes only, and does not imply endorsement by the U.S.
Geological Survey.
PROGRAM LIST OF ANALYST,
A COMPUTER PROGRAM FOR CONTROL OF AN ISOMASS 54E
THERMAL-IONIZATION, SINGLE-COLLECTOR
MASS SPECTROMETER

INTRODUCTION

ANALYST is a computer program for controlling an Isomass 54E mass-spectrometer. The program is written in Hewlett-Packard EXTENDED BASIC 2.1 for a Series 200 Hewlett-Packard computer (HP-9836, HP-9816, or HP-9817). The program requires the Advanced Programming and Graphics Extensions binary programs and at least 1 megabyte of RAM. The required keyboard is either the standard HP-9836 keyboard or the Option 805 ASCII extended keyboard for the HP-9816/17 computers.

Documentation for the program is given in a separate U.S.G.S. Open-File Report (Ludwig, 1985), which explains in detail how to use the program, as well as the general philosophy behind it. This report is intended to provide the program listing and use as a reference for those wishing to excise or modify the various subprograms in ANALYST.

A copy of the current version of the program (debugging and modification is expected to be a long-term process) is available from the author on 5-1/4" or 3-1/2" disks on request.

REFERENCES

APPENDIX: PROGRAM LIST OF ANALYST

3 !************************************************************************************** ANALYST **************************************************************************************

6 ! MAR 22 1985 05:50 pm
9 ! HP-9836 Program for ISOMASS 5AC control
12 ! Requires 400 Kbytes net memory, BASIC 2.0, AP2.1 and GRAPH2.1, Printer at 701
15 \ OPTION BASE 1
18 ! Printer_model=2631
21 Printer_model=2225 ! Thinkjet
24 Canump=(Printer_model=2631) ! can the printer take graphics-dumps?
27 MESSAGE 48,42
30 DUMP DEVICE IS 701 \ but disabled with 2631 printer
33 OUTPUT KBD;"SCRRTCH KEY ":CHR$(255)&CHR$(88); ! scratch typing-aid softkey definitions
36 OFF KEY
39 CONTROL 2,1:0 \ cancel PRINTALL
42 CONTROL 1,4:0 \ cancel DISPLAY FUNCTIONS
45 PRINTER IS CRT
48 I_t=2 ! default integration time is 0.2 seconds
51 ! (if I_t=1, integration time is 1 sec; I_t=2 indicates .2 sec)
54 Stripchart=0 ! exit graphics-monitor node
57 IF Run=0 THEN Run=1 \ in case of invalid program-start
60 OUTPUT KBD;CHR$(255)&CHR$(75);
63 PRINT USING "/,K,/,12K,K,K":CHR$(131)&RPT$(" ",80),RPT$(" ",80); ***** THE PRINTER ISN'T RESPONDING ***** "RPT$(" ",15),RPT$(" ",83),CHR$(128)
66 PRINT USING "/,K H ;PRESS THE "CHR$(132)&"CLR I/0"CHR$(128) &" KEY (UPPER RIGHT), CORRECT THE PROBLEM, THEN PRESS "CHR$(132)&"CONTINUE"CHR$(128)
69 PRINTER IS 701:WIDTH <80
72 PRINT USING "/,K":CHR$(27)&"BADS" \ normal print-size
75 PRINT USING "/,K":CHR$(27)&"B&80" \ 1/8" line spacing
78 IF Printer_model=2225 THEN
81 PRINT USING "/,K":CHR$(27)&CHR$(15) \ cancel bold print
84 PRINT USING "/,K":CHR$(27)&"B&80" \ cancel underline
87 END IF
90 PRINTER IS CRT
93 OUTPUT KBD;CHR$(255)&CHR$(75);
96 IF Debug THEN \ skip to BMC if not 1st RUN
99 IF Full_auto THEN \ printout message if illegal full-auto exit
102 \ PRINTER IS Prtr(2)
105 PRINT USING "/,K,/:"****** PAUSE exit from auto-running, ", &TIME$(TIME) &" ******
108 Full_auto=0
111 END IF
114 GOTO Skey_bmc
117 END IF
120 \!
123 !
126 Message:IMAGE "$OFJ",42
129!
132 DIM Input%(63),Prompt%(18),Response%(18),ET01(50),ET101,Time(5,2),PrTr%501
135 INTEGER Pr,Data daily,Escape
138 INTEGER Uncentered_pks(8),Singlefocus(8),Triplefocus(8),Iso(8)
141 REAL Val(8),Use(18),Range(18,2),Flag(2),Resistor_const(6)
144 !
PRINT USING "K,/,16X,K,/,"; "F3 AMPLIFIER: TURN ON, SET AMPS-FULL-SCALE AT IE-5, GAIN AT 1,"; "RESPONSE TIME AT 30 mS ." 
PRINT USING "K,/,8X,K,/,"; "TURN ON ELECTROMAGNET SUPPLY, MULTIPLIER SUPPLY, MOTOR CONTROL,"; "SYSTEM MONITOR, DIGITAL INTEGRATOR."

PRESS BRANDENBURG MAINS, THEN RESET BUTTONS.

PRINT USING "K,/,8X,K,/,"; "TURN ON ELECTROMAGNET SUPPLY, MULTIPLIER SUPPLY, MOTOR CONTROL,
SYSTEM MONITOR, DIGITAL INTEGRATOR."

PRESS BRANDENBURG MAINS, THEN RESET BUTTONS.

PRINT USING "K,/,8X,K,/,"; "TURN FILAMENT KNOBS TO RESET AND AUTO."

PRINT USING "K,/,8X,K,/,"; "TURN PROGRAMMABLE DEFLECTION UNIT & FOCUS UNIT TO AUTO."

PRINT USING "K,/,8X,K,/,"; "TURN PROGRAMMABLE FOCUS UNIT POWER AND OUTPUT TO ON, RU TO 7, 100U TO 0."

PRINT "PUT "@FN$#("SYSTEM"&" DISK IN RIGHT-HAND DRIVE, "@FN$#("DATA"&" DISK IN LEFT-HAND DRIVE."

DISP @$#&("press "@FN$#("CONTINUE")&" when ready)"

PAUSE

DISP

PRINT USING "18/,9X,K,9/"; "IS THE BEAM VALUE OPEN? (press CONTINUE when checked)"

PAUSE

Check_elements(Elements(*),0)

DISP

Run=Auto=0

Full_auto=0

ASSIGN @Path1 TO "FOCUS-INTERNAL"

ENTER @Path1,1;Singlefocus(*)! single-filament focus settings

ENTER @Path1,2;Triplefocus(*)! triple-

ASSIGN @Path1 TO *

GOSUB Daily

Input=RV4(DATE$TIME$DATE$)! check clock if date not in 1980's, => computer turned off

IF VAL(Input$2)<891 THEN CALL Clockset

GOSUB Element

GOSUB Autel

GOSUB Mfind

Shiftlabel(Stripchart,Sample_name$(*),Sample,Run,Full_auto,File$)

Debug=1

GOTO Bnc_1

Daily:CALL Daily_enable(Daily)

OFF KEY

IF Daily AND Debug AND NOT Ze(1) THEN

Zero(filament(*),0,0,Daily,Magnet(1,2),Peak_inter, Mu,Mu$(*),Collector$(*))

IF Subflag THEN WAIT 3

END IF

RETURN

Element:Dispel(Elements(*))

A=0

INPUT "ENTER THE NUMBER OF THE ELEMENT (or press CONTINUE to escape)",A

IF A=0 THEN Skey_bnc

Type=A

RETURN

A=0

Autel:Getel(Type,Runclass$,Nuclide$(*),Normal0(*),Run_name$(Sample),Nu,RF,HV,Magnet0(*),Coarsebin(*),Ref)

IF Subflag THEN

IF Auto THEN Do_next_run

IF NOT Auto THEN Change_element

END IF

Miniso=0

Spike=0

Changed_el=1

IF NOT Auto THEN

LOOP

CALL Hv(Mu$(*),Hv,P,Mu,I_1)
EXIT IF P OR (ABS(Hu-Hu0)<4) ! Hu must be within 4U of std value
PRINT USING "K,K,":"ADJUST HU TO ",Hu0
Clunk
END LOOP
PRINT
END IF

Mu=0
Correct(ff$(*),Mu$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsemag(L(0),Foc(*)))
MAT Daily ok= (0)
B=Magnet(L,2)
Jump_dir=1-2*(L=Niso)
RETURN

! CONTROL KBD:O ! caps lock OFF
! OFF KNOB
! OFF ERROR
Correct(ff$(*),Mu$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsemag(L(0),Foc(*)))
Stripchart=0
Bnc_l:PRINTER IS CRT ! Alternate entrance-line
TRCE OFF
ON KEY 0 LABEL "0= CENTER PERK" GOTO Calcent
ON KEY 1 LABEL "1: FOCUS BEAM" GOTO Calfoc
ON KEY 2 LABEL "2: CNTR BARREL" GOTO Calbar
ON KEY 3 LABEL "3: BEAM CHART" GOTO Chartmenu
ON KEY 4 LABEL "4: FIL. CURRS" GOTO Calfil
ON KEY 5 LABEL "5: CUP/DAILY" GOTO Bnc_collchange
ON KEY 6 LABEL "6: SCAN MAGNET" GOTO Magscan
ON KEY 7 LABEL "7: CHANGE ELEM" GOTO Change_element
ON KEY 8 LABEL "8: NEW SAMPLE" GOTO Do_new
ON KEY 9 LABEL "9: TAKE DATA" GOTO Mandat
ON KEY 10 GOTO Magenu
ON KEY 11 GOTO Opticsmenu
ON KEY 12 GOTO Barrelnenu
ON KEY 13 GOTO Collinesu
ON KEY 14 GOTO Diagondata
ON KEY 15 GOTO Spikedata
ON KEY 16 GOTO Status
ON KEY 17 GOTO Calrunvar
ON KEY 18 GOTO Auto
ON KEY 19 GOTO Calknob
ON KEY 20 GOTO Keybranch !"Live" keyboard for branching
ON KBD .15 GOSUB Fil_knob ! KNOB rotated
CLoop
OUTPUT 8 USING "4A,42":"SUMU",Magnet(L,1)
IF L THEN CALL Enter_beam(0,Mu,L,I_t,Pr,0,0,1)! non-187
IF L=0 THEN CALL Enter Beam(0,Mu,L,I_t,Pr)! Re-187
IF Pr=3 AND L>0 THEN ! if beam>10 volts, jump to next isotope (if L>0)
Over_iso=1
Over_iso
IF Over_iso>Niso THEN
Enter Beam(0,Mu,L,I_t,Pr)
ELSE
PRINT TAB(20,18);CHR$(131);": ***:Magnet(L,1):" ; PERK IS >10 VOLTS *** "CHR$(128)
L=L+Jump_dir
IF L>Niso THEN L=1
IF L=0 THEN L=Niso
B=Magnet(L,2)
663  OUTPUT 8 USING Mag:O
666  Beep(3000, .02, .001, 15)
669  END IF
672  ELSE
675  Over_iso=0
678  END IF
681  IF NOT Mu THEN Yd=GROUND(Mu,1+(Mu)>10)/(Mu)>1000))! rounded beamsize
684  IF Mu THEN Yd=GROUND(Mu,1+(Mu)>1000))
687  Running_time=((TIMELAPSE-time0)/80 time since sample rotated into position(min.)
690  IF Stripchart THEN
693  IF Running_time=MAX THEN
696  IF Dump_stripchart AND Cndump THEN DUMP GRAPHICS
699  GOTO Monitor
702  END IF
705  IF (NOT Lognon) THEN DRRU Running_time= (graphic bean-monitor)
708  IF Lognon AND (Mu<>1) THEN DRRU Running_time,LOG(flY)+NOT tlu)
711  END IF
714  DISP USING 720;Ci$, Magnet(L,1), Cn$, Nuclide$(L), Pk_offset$(10), B, Filament(1), Filament(2), YD$(Yd), Collector$(Mu)
717  END LOOP
723  !
726  !**************************************************************************
729  !**************************************************************************
732  !
735  Fil_knob:STATUS 2,18; K ! KNOB rotated; K=1 if shift-key pressed also,
738  ! 2 if control-key, 3 if both
741  IF K<2 OR K=3 THEN RETURN
744  Curr_change=INT(1000 .*KNBXY/2000)/iGOO
747  Filnum=K-1 ! change center-fil if CONTROL-KNOB, side-fil if CTRL-SHIFT KNOB
750  f=Filenum(Filnum)>Curr_change
753  f=f(.0)*f(FX(.0))?.f(FX(.0))
756  filament(Filnum)=f
759  OUTPUT 8 USING "4R,4Z";F#(0,Filnum), FNF(F)
762  H=(F(.0)*100)+F(.0)*100
765  IF H>4000 THEN H=4000
768  IF TIMELAPSE-1knob>.15 THEN BEEP H, .02
771  Knob=TIMELAPSE I don't beep any more often than every .2 sec
774  MAT Daly_ok= (0)
777  Daly_ok(L)=Mu
780  RETURN
783  !
786  Softkeys: put softkey-branches here
789  Calften:Center_peak(1,L)
792  GOTO Bnc_1
795  Calfocus:Focus(4,1,1, focusnum)
798  Stripchart=0
801  GOTO Bnc_1
804  CalBar:center_barrel(Sample)
807  IF Subflag=1 THEN
810  GRAPHICS OFF
813  PRINT USING "18/4X,K,6/":FMB$("********** CAN'T KEEP FILAMENT-CONTACTS FOR THIS SAMPLE **********")
816  Superclink
819  END IF
822  Stripchart=0
825  GOTO Bnc_1
828  Magscan=IDOSUB Clearall
831  IF Minsio=0 THEN
834  Minsio=Magnet(1,1)
837  Maxiso=Magnet(1,1)
840  Mass_speed=.2
843       Max_pk=10000
844       Logscan=0
849       END IF
852     Scan(Stripchart,Minise,Maxise,Max_pk,Mass_speed,Logscan,Coarsebin(*),Niso,I_t)
855     GOTO Bmc
858      !
861     Caltest:GOSUB Clearall
864     Contact_test(filament(*),Ime0,E,Sample_name(*),Run_name(*),Canunup,Estab(*),Sample,Nfils)
867     ON E GOTO Bmc,SKey,bmc
870      !
873     Do_mfind:GOSUB Mfind
876     GOTO Bmc_1
879      !
882     Bmc_collchange:GOSUB Collchange
885     GOTO Bmc_1
888      !
891     Collchange: ! Toggle from Faraday Cup to Daly Detector or vice versa
894     IF M THEN ! go from Daly to Faraday Cup
897     PRINT TAB(18):FNUM($(" ON FARADAY CUP ",1),60)
899     BEEP 200,1 ! Switch from Daly to Faraday Cup as collector
903     M=0
906     Correct(Ff$(*),M$(*),filament(*),Magnet(L,2),0,I_t,Coarsemag(L<>0),Fac(*))
909     WAIT 5
912     ELSE ! Switch from Faraday Cup to Daly
915     A=Daly_ok(L)
918     Mtest ! check if Daly can be used for this peak, use if OK
921     IF A THEN WAIT .2
924     END IF
927     RETURN
930      !
933     Calfil:GOSUB Clearall ! change filament-current
936     DATA FILAMENT-NUMBER,("1=center-sample, 2=side-sample", "3=center-preheat, 4=side-preheat"),",",NEW CURRENT (amperes),",",MAT
939     DATA 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,01,001,1
942     RESTORE 936
945     REDIM Prompt$(8),Response$(8),Use(*),Range(*)
948     READ Prompt$(*),Range(*)
951     MAT Use= (0)
954     Use(3)=1
957     Use(5)=1
960     Use(8)=1
963     MAT Response$= ("")
966      ! pass last filament# to menu
969     IF Lastfil=0 THEN Response$(3)=URL$(Filnum)
972     IF Lastfil THEN Response$(3)=URL$(Lastfil)
975     IF Nfils=1 THEN
978      Response$(3)="1"
981    Prompt$(6)="(present CF current is "URL$(Filanent(1))" amps)"
984     END IF
987     IF Nfils=2 THEN Prompt$(6)="(present CF="URL$(Filament(1))", SF="URL$(Filament(2))")"
990     Response$(5)="?"
993     IF Lastrate=0 THEN Response$(8)="10"
996     IF Lastrate THEN Response$(8)=URL$(Lastrate)
999    Form(Prompt$(5),Response$(8),Use(*),Range(*),8,(Nfils=2)+5*(Nfils=1),E)
1002 IF E THEN SKey,bmc
1005     Lastfil=URL(Response$(3))
1008 Lastrate=URL(Response$(8))
1011 CALL Filament(Lastrate,URL(Response$(5)),S,Lastfil)
1014 WHoop
1017 OUTPUT KBD:Clear$;
1020 IF F<3 THEN Skeyjnc
1023 PRINT "PREHEAT: CF";filament(3);" SF";filament(9);CHR$(13)&CHR$(10)
1026 GOTO Bnc
1029 !
1032 CalfknobsCALL Filamentknob(Stripchart,Logmon,Time(D),Collector(*),I_t,L,Magnet(*,)),I_t
1035 ON 1+Stripchart GOTO Skey_bnc,Bnc
1039!
1041 Magmenu1: General menu for magnet-related functions
1044 GOSUB Clearall
1047 REDIM Response$(10)
1050 MAT Response$=""
1053 Response$(1)="SCAN MAGNET (AUTO)"
1056 Response$(3)="CHECK PEAKSHAPE"
1059 Response$(4)="DISPLAY MAGNET & OTHER ELEMENT-DATA"
1062 Response$(5)="CHANGE ELEMENT (increments RUN#)"
1065 Response$(2)="SCAN MAGNET USING "M"NU" (KNOB)"
1068 Response$(6)="DEFINE DATA FOR A NEW ELEMENT"
1071 Response$(7)="ADJUST MAGNET-VALUES FOR SLIGHT DRIFT"
1074 Response$(10)="RETURN TO BMC"
1077 CALL Keymenu(Response$(*)
1080 ON KEY 0 LABEL "0: SCAN MAGNET" GOTO flag-scan
1083 ON KEY 2 LABEL "2: PEAK SHAPE" GOTO Calshape
1086 ON KEY 3 LABEL "3: DISP VALUES" GOTO Element_print
1089 ON KEY 4 LABEL "4: CHANGE EL" GOTO Change_element
1092 ON KEY 1 LABEL "1: MAGNET KNOB" GOTO Magnet_knob
1095 ON KEY 5 LABEL "5: NEW ELEMENT" GOTO Cal_newel
1098 ON KEY 6 LABEL "6: DRIFTJUST" GOTO Cal driftedjust
1101 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bnc
1104 GOTO 1104
1101
calshape:M=Mu ! do a graphic check of peakshape
1113 I_t=L_t
1116 GOSUB Clearall
1119 PRINT USING "2/,K,3/'Y GRAPhICS PEAK-SHAPe CHECK: ";
1122 Center_peak(L)
1125 OUTPUT 8:M$(M,T) ! restore original integration-time
1128 IF M>Mu THEN WAIT 6
1131 Mu=M
1134 I_t=1
1137 IF Subflag=4 THEN Skey_bnc
1140 CALL Shape(Mu,Nuclide(*),I_t)
1143 OUTPUT K60:Clear$;
1146 GOTO Bnc
1149!
1152 Magnet_knob:CALL Magknob(Coarse,B,Coarsebin(*),Mu)
1155 GOTO Skey_bnc
1159!
1161 Change_element:GOSUB Element
1164 GOSUB Autel
1167 GOTO Bnc
1170 !
1173 Cal_newel:CALL Newel(Type,Interfere(*),Normal(*),Nuclide(*),Element(*),Runclass$E,Coarsebin(*),Niso,Ref,Rf,Inu,Hu0)
1176 IF NOT E THEN GOSUB Autel
1179 ON 1+E GOTO Skey_bnc,Change_element
1182 !
1185 Cal driftedjust:Drift_adjust(Type,Runclass$E,Normal(*),NormO,Nas,Niso,Nf,Hu0,Magnet0(*),Coarsebin(*),Ref)
1188 GOTO Skey_bnc
1191 !
1194 Opticsmenu!: Menu for functions related to ion optics
1197 Stripchart0=Stripchart
1200 GOSUB Clearall
1203 REDIM Response$(10)
1206 MAT Response$= ('')
1209 Response$(1)="AUTOFOCUS ION-OPTICS"
1212 Response$(2)="FOCUS ION-OPTICS USING KNOB"
1215 Response$(3)="SCAN FOCUS-POTENTIALS"
1218 Response$(4)="DISPLAY FOCUS VALUES"
1221 Response$(5)="RESTORE FOCUS USING STANDARD FOCUS-VALUES"
1224 Response$(6)="DEFINE & STORE PRESENT FOCUS-VALUES AS STANDARD"
1227 Response$(7)="QUERY HIGH-VOLTAGE"
1230 Response$(8)="TYPE IN NEW FOCUS-SETTINGS"
1233 Response$(9)="COMPLETE BEAM TUNE-UP"
1236 Response$(10)="RETURN TO BNC"
1239 CALL Keymenu(Response$(*))
1242 ON KEY 0 LABEL "0: AUTO FOCUS" GOTO Calfoc
1245 ON KEY 1 LABEL "1: MANUAL FOC." GOTO Calfocfocus
1248 ON KEY 2 LABEL "2: SCAN FOCUS" GOTO Calfocscan
1251 ON KEY 3 LABEL "3: DISP VALUES" GOTO Focprint
1254 ON KEY 4 LABEL "4: USE STD" GOTO Do_stdfocus
1257 ON KEY 5 LABEL "5: DEFINE STD" GOTO Defstd
1260 ON KEY 6 LABEL "6: HU?" GOTO Calhy
1263 ON KEY 7 LABEL "7: TYPE IN FOC" GOTO Calfocshift
1266 ON KEY 8 LABEL "8: BEAM TUNEUP" GOTO 2494
1269 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bnc
1272 ON KEY 10 GOTO Call_hchange
1275 PRINT USING "2/,1/K","V\" Change Standard-HU Values for Several Elements"
1278 GOTO 1278
1281 !
1284 Calfocshift:=CALL Focshift(Foc(*)) ' change focus settings to arbitrary values
1287 GOTO Skey_bnc
1290 !
1293 Callfocus:=GOSUB Clearall ' focus the ion optics "manually" using KNOB
1296 CALL Mcfocus(Stripchart0,Lognon,line0,i,t,Foc(*),Z)
1299 Stripchart=Stripchart0
1302 ON 1+Stripchart GOTO Skey_bnc
1305 !
1308 Calfocscan:=GOSUB Clearall ' scan focus-settings
1311 Focscan(Foc(*),i,t)
1314 GOTO Skey_bnc
1317 !
1320 Call_hchange:=Change_hv
1323 GOTO Skey_bnc
1326 !
1329 Do_stdfocus:=OUTPUT KBD:Clear$;
1332 GOSUB Stdfocus
1335 PRINT USING "/,K","press "& amp;"\"CONTINUE\"")" when ready."
1338 PAUSE
1341 GOTO Skey_bnc
1344!
1347 Defstd:=ON ERROR GOTO Bad_stored ' store current focus-settings on disk
1350 ASSIGN @Pathl TO "FOCUS\EXTERNAL"
1353 OUTPUT @Pathl,Mfiles,Foc(*)
1356 PRINT USING "/,K","CURRENT FOCUS-SETTINGS NOW STORED ON DISK AS STANDARD SETTINGS."
1359 PRINT USING "/,K","FOCUSING VALUES =",1,2,3,4,5,6,7,8,Foc(*)
1362 IF Mfiles=1 THEN MAT Singlefocus=Foc
1365 IF Mfiles=2 THEN MAT Triplefocus=Foc
1368 GOTO Bnc_1
1371 !
1374 Barrelenum:= Menu for barrel-related functions
1377 GOSUB Clearall
1380 RE DIM Response$(10)
1383 MAT Response$ = ("")
1386 Response$(1) = "AUTO-ADJUST BARREL-POSITION FOR BEST BEAM"
1389 Response$(2) = "MANUAL BARREL-ADJUST"
1392 Response$(3) = "TEST CONTACTS FOR ALL SAMPLES (turns fils OFF)"
1395 Response$(4) = "RETURN TO BMC"
1398 CALL Keymenu(Response$(*))
1401 ON KEY 0 LABEL "0: AUTO-ADJUST" GOTO Calbar
1404 ON KEY 1 LABEL "1: MAN.-ADJUST" GOTO Calbar
1407 ON KEY 2 LABEL "2: CALIBRATE" GOTO Caltest
1410 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bmc
1413 GOTO 1413
1416!
1419 Calbar: GOSUB Clearkey
1422 CALL Mbar(Mbar(*),Barrel_position,Barrel_pos0,Min_barrel,Max_barrel,l_t)
1425 GOTO Skey_bmc
1428!
1431 Ratio data! Menu for accessing isotope-ratio data on disk
1434 GOSUB Clearall
1437 RE DIM Response$(10)
1440 MAT Response$ = ("")
1443 Response$(1) = "START TAKING ISOTOPE-RATIO DATA"
1446 Response$(2) = "ENTER NAMES FOR ALL SAMPLES IN BARREL"
1449 Response$(3) = "PRINT SUMMARY OF RUNS"
1452 Response$(4) = "DISPLAY SUMMARY OF RUNS"
1455 Response$(5) = "CALCULATE WEIGHTED AVERAGES FOR A RUN"
1458 Response$(6) = "LOCATE RUN-DATA ON DISK"
1461 Response$(7) = "SHOW LIST OF CURRENTLY-DEFINED SAMPLE-NAMES"
1464 Response$(10) = "RETURN TO BMC"
1467 CALL Keymenu(Response$(*))
1470 ON KEY 0 LABEL "0: TAKE DATA" GOTO Mandat
1473 ON KEY 1 LABEL "1: ENTER NAMES" GOTO Calnames
1476 ON KEY 2 LABEL "2: PRINT-RS" GOTO Prt_result
1479 ON KEY 3 LABEL "3: CRT-RESULTS" GOTO Crt_result
1482 ON KEY 4 LABEL "4: WTO AVERAGE" GOTO Calav
1485 ON KEY 5 LABEL "5: LOCATE DATA" GOTO Catalog
1488 ON KEY 6 LABEL "6: SHOW NAMES" GOTO Cal_shownames
1491 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bmc
1494 GOTO 1494
1497!
1500 Calav: CALL Average(0,Normal(*),Randump,I,(Run),Prtr(*))
1503 ON I = E GOTO Bnc_1.Skey_bmc
1506 Catalog:CALL Cat(Prtr(*))! printout catalog of data on disk
1509 GOTO Skey_bmc
1512 Prt_result:P=2
1515 GOSUB Result
1518 GOTO Skey_bmc
1521 Crt_result:P=1
1524 GOSUB Result
1527 GOTO Bnc_1
1530 Calnames:CALL Name(Sample_name(*),Estbar(*))
1533 GOTO Skey_bmc
1536 Cal_shownames:Show_names(Sample_name(*))
1539 GOTO Bnc_1
1542!
1545 Spikedata! Menu for accessing spike-related functions
1548 GOSUB Clearall
1551 RE DIM Response$(10)
1554 MAT Response$ = ("")
1557 GRAPHICS OFF
1560 Response$(1)="$DISPLAY SPIKE-DATA STORED ON DISK"
1563 Response$(2)="$DEFINE A NEW NORMALIZABLE-ELEMENT SPIKE"
1566 Response$(10)="$RETURN TO BMC"
1569 CALL Keymenu(Response$(*))
1572 ON KEY 0 LABEL "0: DISPLAY" GOTO Showspikes
1575 ON KEY 1 LABEL "1: DEFINE NEW" GOTO Calnewspike
1578 ON KEY 9 LABEL "9: ESCAPE" GOTO Skey_bmc
1580 GOTO 1581
1584
1587 Showspikes:CALL Whichspike(0,0,Subflag) ! show #s & names of spikes defined on disk
1590 ON 1+Subflag GOTO bnc_1,Skey_bnc
1593
1596 Calnewspike:CALL Mspike
1599 GOTO Skey_bmc
1602
1606
1608 Status: ! Menu for accessing mass-spec status functions
1611 GOSUB Clearall
1614 REDIM Response$(10)
1617 MAT Response$= ("")
1620 Response$(1)="$SOURCE/TUBE PRESSURE (SINGLE QUERY)"
1623 Response$(2)="$SOURCE/TUBE PRESSURE (GRAPHICS)"
1626 Response$(3)="$HIGH VOLTAGE"
1629 Response$(4)="$QUERY TIME/DATE"
1632 Response$(5)="$SET TIME/DATE"
1635 Response$(6)="$TEST CONTACTS FOR BARREL & URL$Sample"
1638 Response$(7)="$TEST CONTACTS FOR ALL SAMPLES (turns fils OFF)"
1641 Response$(8)="$MAGNET & RUNNING DATA (current)"
1644 Response$(9)="$FOCUS VALUES (current)"
1647 Response$(10)="$RETURN TO BMC"
1650 CALL Keymenu(Response$(*))
1653 ON KEY 0 LABEL " PRESSURE-1" GOTO Calpres
1656 ON KEY 1 LABEL "PRESSURE-GRAPH" GOTO Calpgraph
1659 ON KEY 2 LABEL " HI VOLTAGE" GOTO Calhv
1662 ON KEY 3 LABEL " READ CLOCK" GOSUB Querytime
1665 ON KEY 4 LABEL " SET CLOCK" GOSUB Calset
1668 ON KEY 5 LABEL " CONTACTS-1" GOTO Calflag
1671 ON KEY 6 LABEL " CONTACTS-16" GOTO Caltest
1674 ON KEY 7 LABEL "ELEMENT VALUES" GOTO Element_print
1677 ON KEY 8 LABEL " FOCUS VALUES" GOTO Focprint
1680 ON KEY 9 LABEL " ESCAPE" GOTO Skey_bmc
1683 GOTO 1683
1686
1689 Calpgraph:CALL Press_graph(Ms$(*),Candump)
1692 GOTO Bmc
1695 Calset:Clockset
1698 RETURN
1701 Querytime:PRINT USING "/,K,5K,K,";FMClock_12B((TIME$(TIME_DATE)),DATE$(TIME_DATE))
1704 RETURN
1707 Calpres:GOSUB Clearall
1710 CALL Pressure(Mu,M$(*),,(I_t),1,1)
1713 GOTO Bmc_1
1716 Calhv:GOSUB Clearall ! query accelerating voltage
1719 CALL Hu(M$(*),Mu,0,Mu,I_t) ! query accelerating voltage
1722 GOTO Bmc_1
1725 Element_print:Disp_elvals(Runclass#,Nuclide#(*),Normal(*),Interfer(*),Coarse,Ref,HvO,Magnet(*),Niso,Inv)
1728 GOTO Bmc_1
1731 Focprint:GOSUB Clearall
1734 PRINT USING "K,2/,B(3D,3K),/B(4D,2K),/":"FOCUSING VALUES":1,2,3,4,5,6,7,8,Foc(*)
1737 GOTO Bmc_1
1740 1
1743 Calflag: 1 check filament flags: if .2 amps on filament, temporarily put .2 amps thru to check
1746 FOR I=0 TO 1 1 sample or preheat
1749 FOR J=1 TO 2 I center or side
1752 IF Filament(I+2*HX.2 THEN OUTPUT 8;Ff$(I,J),200
1755 NEXT J
1758 NEXT I
1761 OUTPUT KBD:Clear$;
1764 CALL Flag(Flag(*),Mu)
1767 FOR I=0 TO 1 ! restore fil-currents if necessary
1770 FOR J=1 TO 2
1773 IF Filament(I+2*HX.2 THEN OUTPUT 8;Ff$(I,J),FNF(Filament(I+2*HX.2))
1776 NEXT J
1779 NEXT I
1782 GOTO Bnc
1785 1
1788 Collmenus: menu for accessing collector functions
1791 GOSUB Clearall
1794 REDIM Response$(*)
1797 MAT Response$= ("")
1800 IF Mu THEN Response$= "USE FARADAY CUP AS COLLECTOR"
1803 IF Daly AND NOT Mu THEN Response$= "USE DAILY-DETECTOR AS COLLECTOR"
1806 Response$= "TAKE ZEROES FOR BOTH COLLECTORS"
1809 Response$= "CALIBRATE DAILY"
1812 Response$= "CHANGE ENABLE/DISABLE STATUS OF DAILY"
1815 Response$= "RETURN TO BMC"
1818 CALL Keymenu(Response$(*))
1821 IF Mu THEN OR KEY 0 LABEL "0: CUP" GOTO 1845
1824 IF Mu THEN OR KEY 0 LABEL "0: DAILY" GOTO 1845
1827 OR KEY 1 LABEL "1: ZEROES" GOTO Calzer
1830 OR KEY 2 LABEL "2: DAILYCAL" GOTO DalyCal
1833 OR KEY 3 LABEL "3: DAILYSTATUS" GOTO DalyStat
1836 OR KEY 9 LABEL "9: ESCAPE" GOTO Sky_bnc
1839 GOTO 1839
1842 GOSUB Collchange
1845 GOTO Sky_bnc
1848 GOTO 1845
1851 1
1854 Calzer:OUTPUT KBD:Clear$: 1 determine zeroes for Faraday Cup & Daly
1857 CALL Zero(Filament(*),0,0,Daly,R,Peak_inter,Mu,Mn$(*),Collector$(*)) 1 determine zeroes for Faraday Cup & Daly
1860 IF Subflag THEN
1863 DISP "(press "&FNM"("CONTINUE")&" to continue)"
1866 PAUSE
1869 END IF
1872 GOTO Bnc
1875 1
1878 DalyCal:Center_peak(I,L) 1 Calibrate Daly gain
1881 IF Subflag=4 THEN Sky_bnc
1884 OUTPUT 8;Mn$(0,1)
1887 IF Mu THEN WRT 6
1890 Calibrate(Filament(*),Magnet(I,2),Peak_inter,Daly,I_t,Mu,Mn$(*),Collector$(*))
1893 GOTO Bnc
1896 1
1899 DalyStat:GOSUB Daly
1902 GOTO Sky_bnc
1905 1
1908 Chartmenus: menu for accessing graphics-stripchart functions
1911 GOSUB Clearall
1914 REDIM Response$(*)
1917 MAT Response$= ("")
2097 CASE -162,-169,-170,-172,-418,-426,-425,-428
2100 | change fil-current by .01 amps: unshifted U/D arrow for sample-center, shifted U/D arrow for sample-side, ditto but .001 amp if CTRL used
2103 MSfil:=Keycode
2109 IF F<2 AND Mfil=1 THEN Bnc_1
2112 Filnum=F
2115 SELECT K
2118 CASE -162,-169 | up-arrow, SHIFT up-arrow
2121 Change=.01
2124 CASE -170,-172 | down-arrow, SHIFT down-arrow
2127 Change-.01
2130 CASE -418,-425 | CTRL up-arrow, CTRL-SHIFT up-arrow
2133 Change=.001
2136 CASE -426,-428 | CTRL down-arrow, CTRL-SHIFT down-arrow
2139 Change-.001
2142 END SELECT
2145 F=PROUNO(Filament(Filnum)+Change,-4)
2148 Filament(Filnum)=(F=0)*(F*(F=7)+7*(F>7))
2151 OUTPUT 8 USING "Filaent(Filnum)”,Filnum=Filnum2 OR Filnum=4,filnum(Filnum)
2154 BEEP 220*(Change<0)+660*(Change>0),.05-.03*(Change<.01)
2157 M=fl Daly_ok(0) | zero the multiplier-permitted elements for all isotopes
2160 Daly_ok(0)= Mu | but restore the original value for present isotope
2163 GOTO Bnc_1
2166 END SELECT
2169 |
2172 OFF KEY
2175 SELECT Keycode
2178 CASE 24 | (CTL X)
2181 ALPH A OFF
2184 CALL Pattern
2187 CASE 17,18,19,20 | (CTL 1,2,3, or 4)
2190 GOSUB Filoff
2193 CASE 15 | (CTL /)
2196 I_t=1 | 1-second integration time
2199 PRINT TABXY(1,18);" *** 1 SECOND INTEGRATION TIME *** | (press CTL ^ to return to 0.2 seconds)
2202 BEEP 200,.1
2205 GOTO Bnc
2208 CASE 30 | (CTL )
2211 I_t=2 | 0.2-second integration time
2214 PRINT TABXY(1,18);" ** 0.2 second integration time **
2217 BEEP 800,.1
2220 GOTO Bnc
2223 CASE 42 | (CTL-0)
2226 GOTO Dalystat
2229 CASE 63,47 | (? or /)
2232 Shiftlabel(Stripchart,Sample_name$(*),Sample,Run,Full_auto,Mfil)
2235 CASE -179 | (ALPHA key)
2238 OUTPUT KBD;CHR$(255)&CHR$(-179);
2241 Stripchart=0
2244 CASE -180 | (GRAPHICS key
2247 OUTPUT KBD;CHR$(255)&CHR$(180);
2250 CASE 8 | (CTL-H)
2253 GOSUB Querytime | Read clock
2256 CASE 7 | (CTL-G)
2259 GOTO Calgraph | pressure-graphics
2262 CASE 22 | (CTL-H)
2265 GOTO Calhv | query HV
2268 CASE 6 | (CTL-F)
2271 Stripchart=Stripchart
1920 Response$(1)="START "BNUM#("LINEAR")&" GRAPHICS BEAM-MONITOR"
1923 Response$(2)="START "BNUM#("LOGARITHMIC")&" GRAPHICS BEAM-MONITOR"
1926 IF Candump AND NOT Dump_stripchart THEN Response$(5)="ENABLE AUTO PTR-DUMP OF GRAPHICS BEAM-MONITOR"
1929 IF Candump AND Dump_stripchart THEN Response$(5)="DISABLE AUTO PTR-DUMP OF GRAPHICS BEAM-MONITOR"
1932 Response$(3)="DOUBLE X-Axis TIME-PERIOD (now "URL$(Xtime)" mins)"
1935 Response$(3)="VALUE X-Axis TIME-PERIOD (now "URL$(Xtime)" mins)"
1938 Response$(9)="DOUBLE Y-Axis HI (now "URL$(Yheight)" initial beam)"
1941 Response$(9)="VALUE Y-Axis HI (now "URL$(Yheight)" initial beam)"
1944 Response$(10)="RETURN TO BMC"
1947 CALL Keymenu(Response$(*))
1950 ON KEY 0 LABEL "START LIN-GM" GOTO Limonitor
1953 ON KEY 1 LABEL "START LOG-GM" GOTO Logmonitor
1956 IF Candump AND NOT Dump_stripchart THEN ON KEY 1 LABEL "ON AUTO-DUMP" GOTO Autodump
1959 IF Candump AND Dump_stripchart THEN ON KEY 1 LABEL "OFF AUTO-DUMP" GOTO Offdump
1962 ON KEY 2 LABEL "DOUBLE Y-Axis" GOTO Double_y
1965 ON KEY 7 LABEL "VALUE Y-Axis" GOTO Halve_y
1968 ON KEY 3 LABEL "DOUBLE X-Axis" GOTO Double_x
1971 ON KEY 8 LABEL "VALUE X-Axis" GOTO Halue_x
1974 On KEY 9 LABEL "ESCAPE" GOTO Skey_bmc
1977 IF Dump_stripchart THEN DISP FNH$("AUTO PRINT-DUMP")&FNH$("ENABLED")
1980 IF NOT Dump_stripchart THEN DISP FNH$("AUTO PRINT-DUMP DISABLED")
1983 GOTO 1983
1986
1989 Limonitor:Logmonitor=0
1992 GOTO Monitor
1995 Logmonitor:Logmonitor=1
1998 GOTO Monitor
2001 Autodump:Dump_stripchart=1
2004 BEEP 660,.1
2007 GOTO Chartmenu
2010 Offdump:Dump_stripchart=0
2013 BEEP 220,.1
2016 GOTO Chartmenu
2019 Double_y:Xtime=2*Xtime
2022 BEEP 660,.05
2025 GOTO Chartmenu
2028 Halve_y:IF Xtime>2 THEN Xtime=Xtime/2
2031 BEEP 220,.05
2034 GOTO Chartmenu
2037 Double_x:Yheight=2*Yheight
2040 BEEP 660,.05
2043 GOTO Chartmenu
2046 Halve_x:Yheight=Yheight/2
2049 BEEP 220,.05
2052 GOTO Chartmenu
2055 !
2058 !
2061 Calrunvar=CALL Runvariables(Runvar(*),Run_name(*),Run_type(*),Sample_name(*),Run_order(*),Run_iso(*),0) ! manipulate run variables
2064 GOTO Skey_bmc
2067!
2070!
2073!***************************************************************************
2076 Keybranch=CALL Kbrd(KBD$,Keycode) ! branch from non-softkey keystrokes
2079 OFF KBD
2082 OFF KNOB
2085 SELECT Keycode ! branching to avoid blinking of key-display by avoiding OFF KEY
2088 CASE 32,94,40,41,43,45,-16,-194,-163 10 -455,-414 10 -406
2091 IF L THEN DOSUB Magswitch
2094 GOTO Bmc_1
2274  GOTO Calnfocus ! manual focus
2277  CASE 3 ! (C-T-C)
2280  Check_elements(Element$(*),0) ! recheck element-types on disk
2283  CASE -192 ! (RECALL)
2286  IF NOT Auto THEN Bnc_1
2289  Resume_auto=OFF KEY ! resume automatic running after interrupt
2292  ON KEY 0 LABEL " BMC" RECOVER Recover_bnc
2295  ON KEY 1 LABEL " NEXT RUN" RECOVER Recover_quit
2298  OFF KNOB
2301  Full_auto=1
2304  OUTPUT KBD:Clear$;
2307  GRAPHICS OFF
2310  Nruns=1+(Runvar(Run,1))
2313  Spike=Runvar(Run,26)
2316  IF Spike THEN
2319  Whichspike(Spike,1,Subflag)
2322  IF Subflag THEN autospike_error
2325  END IF
2328  Daily=Runvar(Run,5)*(Daily<>0)
2331  ON 1 Where GOTO Autcont,Autfind,Preheat,Filup,Takeup_cf,alignment_over,Current,Wait1,Wait2,Start,Retake,Rel,Set
2411,Gu,Take_data
2334 ! (SHIFT-RECALL)
2337  CASE -192 ! (SHIFT-RECALL)
2340  IF NOT Auto THEN Bnc_1
2343  INPUT "WHICH RUN DO YOU WANT TO CONTINUE AUTOMATIC-RUNNING FROM? (0 to escape)",&
2346  IF R=0 THEN Bnc
2349  IF Runvar(R,1)>0 THEN
2352  PRINT "NO RUN VARIABLES DEFINED FOR RUN";R
2355  Clunk
2358  GOTO 2343
2361  END IF
2364  Run=R
2367  Full_auto=1
2370  GOTO Autcont
2373  CASE 1 ! (C-T-A)
2376  Average0,Normal(*),Cumum,E,(Run),Ptrt(*) ! calculate weighted averages
2379  IF E THEN Sky_bnc
2382  CASE 2 ! (C-T-A)
2385  GOTO Caltest ! graphic contact-test for all samples
2388  CASE 13 ! (C-T-
2391  IF Magnet(0,1) AND Magnet(0,2) AND L THEN
2394  From_iso=1 ! switch magnet to Re-187
2397  L=0
2400  Mu=0
2403  Correct(Ff$(*),Mf$(*),Filament(*),Magnet(0,2),Mu,L,t,Coarsemag(0),Foc(*))
2406  B=Magnet(L,2)
2409  END IF
2412  CASE 11 ! (C-T+)
2415  L=From_iso ! switch magnet from 187 to normal peak
2418  Mu=0
2421  Correct(Ff$(*),Mf$(*),Filament(*),Magnet(L,2),Mu,L,t,Coarsemag(1),Foc(*))
2424  B=Magnet(L,2)
2427  CASE 16 ! (C-T-P)
2430  GOTO Calshape ! do a graphics check of peakshape
2433  CASE 14 ! (C-T-N)
2436  Show_names(Sample_name$(*)) ! display all defined sample names
2439  CASE 12 ! (C-T-L)
2442  GOTO Catalog ! locate run-data on disk
2445  CASE 26 ! (C-T-Z)
2448  GOTO Calzer
2151 CASE 10  !  (CIL->)
2154 Stripchart=0  !  do a complete center/focus/barrel/focus
2157 Center_peak(1,L)
2160 IF Subflag=1 THEN Skey_bnc
2163 Focus(3,1,1,Focal)
2166 IF Subflag=1 THEN Skey_bnc
2169 Center_peak(1,L)
2172 Center_barrel(Sample)
2175 IF Subflag=1 THEN 887
2178 IF Subflag=1 THEN Skey_bnc
2181 Focus(2,1,1,Focal)
2184 CASE 61  !  equals sign; calc. 207/206 age
2187 Mu=0
2190 OUTPUT &M&$(Mu,1_t)  !  put on cup just in case
2192 R=0
2195 INPUT R
2198 CASE -177  !  (DUMP ALPHA)
2201 DUMP ALPHA
2204 CASE -178  !  (DUMP GRAPHICS)
2207 IF Candump THEN DUMP GRAPHICS
2210 CASE -195  !  (RESULT)
2213 P=2  !  printout isotope-ratio data for run(s)
2216 GOSUB Result  !  using printer (RESULT) or CRT (SH-RESULT)
2219 GOTO Skey_bnc
2222 CASE -163  !  (SHIFT-RESULT)
2225 P=1  !  display isotope-ratio data for run(s)
2228 GOSUB Result
2231 CASE -176,-223  !  (PAUSE, STOP)
2234 PAUSE  !  pause the program
2237 CASE -181,-221  !  (CLR LN, CLR SCR)
2240 OUTPUT KBD;
2243 CLEAR$,;
2246 PRINT USING ";10/,K,/,K";"PRESS kO TO COMPLETELY RESTART THE PROGRAM"," (this will turn off all filaments & reset the barrel!!)
2249 PRINT USING ";2/,K";"PRESS k9 TO RETURN TO BMC"
2252 OFF KEY
2254 ON KEY 0 LABEL "RESTART" GOTO 2574
2257 ON KEY 9 LABEL "ESCAPE" GOTO Skey_bnc
2260 GOTO 2571
2263 GOTO 3
2266 CASE -214  !  INS LN key
2269 OUTPUT KBD;Clear$;  !  dump graphics with a label at the bottom
2272 Input$=""
2275 INPUT "GRAPHICS-DUMP WITH LABEL AT BOTTOM---- ENTER LABEL (? to escape)";Input$[1,60]
2278 IF Input$[1,13]="?" THEN Skey_bnc
2281 CSIZE 4
2284 VIEWPORT 0,100,RATIO,0,100
2287 WINDOW 0,100,0,100
2290 LORG 1
2293 MOVE 5,15
2296 LABEL Input$[1,52]
2299 DUMP GRAPHICS
2302 GCLEAR
2305 Stripchart=0
2308 GOTO Skey_bnc
2311 CASE ELSE
2805 GOTO Bmc_1
2806 !
2811 !
2814 !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
2817 !
2820 Magswitch:  ! switch magnet to another position or isotope
2823 Key#=CHR$(Keycode)
2826 K=Keycode
2829 SELECT K
2832 CASE -43 TO -455  ! CTRL-softkey pressed: jump up # of masses of softkey
2833 Jump=K+454
2836 L=L+Jump
2841 IF L>Liso THEN L=Liso
2844 B=FNisomag(Mcoef(*),Magnet(L,1))
2847 Beep(2000, .03, .02, 5)
2850 Jump_dir=1
2853 JD=3
2856 CASE -414 TO -406  ! CTRL-SHIFT softkey pressed: jump down # of pk's of key
2859 Jump=(K+415)
2862 L=L+Jump
2865 IF L<1 THEN L=1
2868 B=FNisomag(Mcoef(*),Magnet(L,1))
2871 Beep(220, .03, .02, 5)
2874 Jump_dir=-1
2877 JD=3
2880 CASE -196  ! left-arrow: switch magnet 1/2 mass down
2883 B=FNisomag(Mcoef(*),Magnet(L,1)-.5)
2886 JD=1
2889 Beep(440, .03, .02, 3)
2892 CASE -194  ! right-arrow: switch magnet 1/2 mass up
2895 B=FNisomag(Mcoef(*),Magnet(L,1)+.5)
2898 JD=5
2901 Beep(1000, .03, .02, 3)
2904 CASE 32, 94  ! space-bar or ': switch to peak-top
2907 JD=3
2910 B=Magnet(L,2)
2913 Beep(660, .03, .02, 3)
2916 CASE 40  ! left-parenthesis key: switch magnet to lower-side of peak
2919 B=Magnet(L,2)-Aside
2922 JD=2
2925 Beep(550, .03, .02, 3)
2928 CASE 41  ! right-parenthesis key: switch magnet to upper-side of pk
2931 B=Magnet(L,2)+Aside
2934 JD=4
2937 Beep(880, .03, .02, 3)
2940 CASE 43  ! + key: switch magnet up to next-defined isotope
2943 Jump_dir=1
2946 L=L+(L>Miso)
2949 B=Magnet(L,2)
2952 JD=3
2955 BEEP 660, .05
2958 CASE 45  ! - key: switch magnet down to next-defined isotope
2961 Jump_dir=-1
2964 L=L-(L>Miso)
2967 B=Magnet(L,2)
2970 JD=3
2973 BEEP 220, .05
2976 END SELECT
2979 Nu=Mu*Daly_ok(L)
2982 OUTPUT 8 USING Mag:n
DISP CHR$(Keycode);"<code "':Keycode:'"> ?? - UNDEFINED KEY" ! undefined key pressed during BMC

Clunk

WAIT .

END SELECT

GOTO Bnc_1

Result:OFF KEY

IF NOT Full_auto THEN

PRINT USING "12;Y,2;K":"STARTING I, ENDING I RUNS FOR PRINTOUT? (CONTINUE = current run ['"BURL1(Run)"], 0=escape);" / (for use negative $s for file $s)

LINPUT Input$

IF Minputs=0 THEN

R=Run

Z=Run

ELSE

R=Val(I)

Z=Val(1+(Minputs))

END IF

IF R=0 THEN RETURN

E=0

OUTP KBD;Clear$

IF MND Z<0 THEN ! request file $s, not run $s

Printres(Ptr(P),A,E,ABS(R),ABS(Z))

IF E THEN RETURN

RETURN

ELSE

fl=1

IF Full.auto THEN P=2

END IF

FOR Run_num=fl TO Z

IF NOT fluto OR NOT POS(UPC$(Runtype$(Run)),"OUTGRS"> THEN CALL Printres(Ptr(P),(Run_num),E)

IF E THEN RETURN

NEXT Run_num

RETURN

Monitor:IF NOT Log«on THEN ! draw axes & start graphics bean-Monitor

IF Mu*(ruK.l> OR (NOT fluto*(Hu<2) THEN

Y«ax=Hu/10+(HOT Mu>*2 ! snail bean

ELSE

Y«ax=Hu*Yheight ! not-snall bean

IF YnaxMOGOO THEN Ynax=10000

ELSE

Ynax=INT(LGT(Hu*Yheight)+3 ! loganttaic Monitor

IF Y«ax)1 THEN Y«ax=1

IF Ynax<l THEN Ynax=l

END IF

StripcharH

Runriing_title=Tit1EOflTE

Tnin=(RunningLtiMe-TineO)/60

Running_ti«e=Tnin

Tnax=INT(Tnin)+Xtine

flxes(0,100,Z5,100,INT(Tfin),Tfin,-Lognon*(l+2),(Ynax),(Ynax),"lIUTES SINCE STflRI OF RUH","nJ BEfit1 K ,0,t1u,(Log M on))

NOUE Tnin,t1y*(NOT Log«on)+LGI(!1v*(t1u)0)*(My(=0)*.00001)*Lognon

OUTPUT KBD;Clear$;
2985 PRINT TAB(1,10);APR$(" ",80)  
2988 RETURN  
2991  
2994  
2997 Fileoff: turn a filament off  
3000 Fnum=Keycode-16 ! =1,2,3, or 4 for filaments 1,2,3,4  
3003 Preheat=(Keycode-18)  
3006 ON DELAY .5 GOTO 3066 ! refuse to comply if keystroke not repeated within .5 seconds  
3009  
3012 ON KBD GOTO 3027  
3015 BEEP 220,.03  
3018 WAIT .01  
3021 GOTO 3015  
3024  
3027 Keyboard(KBD$,Keycode)  
3030 IF (Keycode=17 OR Keycode=20) THEN 3066  
3033  
3036 OFF DELAY  
3039 OFF KBD  
3042 OUTPUT 8 USING "5",4;fff$(Preheat,1*Fnum=2 OR Fnum=4),0  
3045 IF NOT Preheat THEN PRINT "SAMPLE ";  
3048 IF Preheat THEN PRINT "PREHEAT ";  
3051 PRINT fil$(Fnum=2*Preheat):" OFF"  
3054 filament(Fnum)=0  
3057 Mu=0  
3060 GOTO Bmc_1  
3063  
3066 PRINT USING "5/5,2/5:";"TO TURN OFF A FILAMENT, PRESS THE KEY DURING THE CHIRP."  
3069 OFF DELAY  
3072 OFF KBD  
3075 RETURN  
3078  
3081 !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  
3084 Mandate!GOSUB Clearall ! take data "manually" (e.g. by keyboard-request)  
3087 IF Auto THEN  
3090 PRINT USING "5/5,2/5:";"YOU MUST EXIT THE AUTOMATIC-RUNNING MODE", "IF YOU WANT TO TAKE DATA IN THE MANUAL MODE. (press SHIFT  
1-k8 during BMC)."  
3093 Clunk  
3096 WAIT 5  
3099 GOTO Skey BMC  
3102 END IF  
3105 DATA 0,0,0,0,0  
3108 RESTORE 3105  
3111 READ Re,Auto,Full_auto,Beam_window,Limit_growth  
3114 OFF KEY  
3117 IF Run=0 THEN Run=1  
3120 Start_time=Time  
3123 IF Run_name$(Run)>"" AND NOT Changed el THEN Manualdata_menu  
3126!  
3129! first block(s) of data requested for this run  
3132 IF NOT Changed el THEN  
3135 MAT Lastblock_isor=0  
3138 MAT Peak=0  
3141 MAT Last_aver=0 IF no previous blocks of data  
3144 Last_isotopes"="  
3147 END IF  
3150 Dump_datagraf=0  
3153  
3156 IF Normal0(1) AND Normal0(2) THEN  
3159 Fract_normal=1 ! normalize for fractionation using a natural ratio
3162 ELSE
3163  Fract_nornal=0
3164 END IF
3165
3166 IF Fract_nornal AND TRIM$(UPC$(Runclass$))="U" THEN Manualdata_menu
3167 ! include U element in case of 233-236 double-spiked run
3168 !
3169 ! more info needed if a normalizable element
3170 DATA Sample Name,",Spine Number (0 if an unspiked run),",Normalize data to ratios of," First Block? (Y/N),??,","??,",",",
3171 DATA -1,-1,0,0,-1,-1,0,0,0,0,-2,-2
3172 RESTORE 3171
3173 REDIM Prompt$(6),Range(6,2),Use(6),Response$(6)
3174 READ Prompt$(6),Response$(6),Range(6)
3175 MAT Use=(1)
3176 Use(4)=0
3177 Use(5)=0
3178 Use(6)=0
3179 IF Sample_name$(Sample)="" THEN Response$(1)=Sample_name$(Sample)
3180 Form(Prompt$(6),Response$(6),Use(*),Range(*),6,1,1,2)
3181 IF Not THEN Skeu bec
3182 IF Sample_name$(Sample)=Response$(1)
3183 IF NUM(Response$(3))<5 THEN Response$(3)=58 THEN
3185 Spike=UFL$(Response$(3)>0)
3186 ELSE
3187 DISP FHN$("THE SPIKE NUMBER MUST BE A DEFINED SPIKE WITH A NUMBER BETWEEN 1 AND 20")
3188 Clunk
3189 END IF
3190 IF Spike THEN
3191 Whichspike(Spike,1,Subflag)
3192 IF Subflag THEN
3194 DISP FNH$("*** CAN'T GET SPIKE " & Spike & " FROM EITHER DISK ****")
3195 Clunk
3196 END IF
3197 END IF
3198 HorHai=(Response$(6))
3199 IF HorHai THEN HorHai=0
3200 !
3201 !
3202 DATA Sample Name,Isotopes,Number of Sets in a block,Number of Blocks,Beam Window (most-intense peak),", Minimum Beam (volts),
3203 DATA ", Maximum Beam (volts)
3204 DATA ", Maximum Filament-Current (amps),", Maximum Beam-Growth (%/minute),Final Filament-Currents (amps)," (sa-Cen,sa-Si,preh-Cen,preh-Si)
3205 DATA "Daily Status (0,1,2):", (0: Disabled 1: OK for data 2: Beam-tuneup only)," Dump Graphics each Block? (Y/N)
3206 DATA ??,??,","??,",0",",10",",6",",100",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",","NO"
3207 DATA -1,-1,0,0,4,40,1,80,0,0,0,0,10,0,10,0,6,0,100,0,0,0,0,0,0,0,0,0,0,2,-2,-2
3208 DATA Manualdata_menu=RESTORE 3208 ! Construct Data-Taking Form
3209 REDIM Prompt$(15),Response$(15),Use(15),Range(15,2)
3210 READ Prompt$(15),Response$(15),Range(15)
3211 IF Sample_name$(Sample)="" THEN Response$(1)=Sample_name$(Sample)
3212 MAT Use=(1)
3213 DATA 0,0,0,0
3214 READ Use(5),Use(10),Use(12),Use(13)
3215 IF Daly THEN only allow Daly-enable if Daly not completely disabled
3216 Response$(14)=UFL$(Daly)
ELSE

Promp$[12]=""

Promp$[13]=""

Promp$[14]=""

Response$(14)=""

ENDIF

IF Candum=0 THEN

Use$(15)=0

Response$(15)=""

ELSE

Response$(15)=CHR$(89*Dump_data1graf+78*(NOT Dump_data1graf))

ENDIF

IF Normal0[1]=0 OR Normal0[2]=0 THEN Fract_normal=0 ! shouldn't be necessary, but sometimes an invalid fract_normal of 1 has appeared anyway

IF ! Fract_normal AND Trim$(UPC$(Runclas$))="U" AND Block=0 THEN Response$(1)=Sample_name$(Sample)

IF ! Fract_normal AND Trim$(UPC$(Runclas$))="U" THEN

IF Block=0 AND Sample_name$(Sample)="" THEN Response$(1)="Sample_name$(Sample)"

IF Block=0 AND Sample_name$(Sample)="" THEN Response$(1)="??"

ENDIF

!

IF Block THEN

Response$(1)=Run_name$(Run)

Response$(2)=Last_isotopes

Response$(3)=URL$(Nsets)

Response$(4)=URL$(Numberofblocks)

Response$(5)=URL$(Unin)

Response$(7)=URL$(Unax)

Response$(9)=URL$(Maxgrowth)

Response$(11)=URL$(Sample_name$(Sample))

Response$(14)=URL$(Sample_name$(Sample)"

END IF

! Spike THEN

Promp$[2]=Promp$[2]="" (include "$URL$(Spkdrun_ref)\$, "$URL$(Spikdrun_iso(1)\$", "$URL$(Spikdrun_iso(2)\")\$"

IF Block=0 THEN

Response$(2)=URL$(Spkdrun_ref)\$, "$URL$(Spikdrun_iso(1)\$", "$URL$(Spikdrun_iso(2)\$"

IF Block=0 THEN Response$(2)=URL$(Ref)\$, "$URL$(Normal0[1])\$

END IF

END IF

!


If Spike AND Fract_normal THEN

Promp$[2]=Promp$[2]="" (must include "$URL$(Ref)\$" and "$URL$(Normal0[1])\$"

IF NOT Block THEN Response$(2)=URL$(Ref)\$, "$URL$(Normal0[1])"

END IF

IF Block=0 AND Spike=0 THEN

IF UPC$(Runclas$[1,2])="PB" THEN Response$(2)="206,207,208,209"

IF UPC$(Runclas$[1,2])="PB" THEN Response$(2)="82,83"

IF UPC$(Runclas$[1,2])="PB" THEN Response$(2)="238,235"

IF UPC$(Runclas$[1,2])="PB" THEN Response$(2)="270,267"

IF UPC$(Runclas$[1,2])="PB" THEN Response$(2)="96,98,97"

IF UPC$(Runclas$[1,2])="PB" THEN Response$(2)="144,146,143"

IF UPC$(Runclas$[1,2])="PB" THEN Response$(2)="160,162,159"

IF UPC$(Runclas$[1,2])="PB" THEN Response$(2)="39,41"

END IF

END IF

! filament THEN

URL$(Filament(1))\", "$URL$(Filament(2))\", "$URL$(Filament(3))\", "$URL$(Filament(4))

21
3510 Invoke_menuForm(Prompt$(*)}.Response$(*}.Use(*}.Range(*}.15.1>(Response$<1>)"??"}.E.1)
3513 IF E THEN Skip_bmc
3516 OTT KEY
3519 Run_name$(Run)=Response$<1)
3522 Sample_name$(Sample)=Response$(1)
3525 Nset=VAL(Response$<3>
3528 Number_of_blocks=VAL(Response$<4>)
3531 Uin=VAL(Response$<6>)
3534 Umax=VAL(Response$<7>)
3537 IF Umax<1.2*Umin THEN | beam-window must be at least 20% wide
3540 DISP F6H"MAX. BEAM MUST BE AT LEAST 20% GREATER THAN MIN. BEAM"
3543 Clunk
3546 WAIT 3
3549 GOTO Invoke_menu
3552 END IF
3555 Beam_window=(Uin<0 OR Umax<10)
3558 IF Daily THEN Daily=VAL(Response$<1>)
3561 Fmax=VAL(Response$<8>)
3564 Max_growth=VAL(Response$<9>)
3567 Limit_growth=Max_growth<100
3570 Temp$=Response$<11>
3573 Parse(Temp$,Val(*).Ninputs)
3576 MAT Off_file= Filament
3579 Change_curr=0 | change fil-currs after block?
3582 FOR I=1 TO 4
3585 Off_file(I)=Val(I)
3588 Change_curr=Change_curr+ABS(Filament(I)-Off_file(I))
3591 NEXT I
3594 IF Change_curr<.001 THEN Change_curr=0
3597 Dump_datagraf=Candump*FYes(Response$<15>)
3600 IF NOT Candump THEN Response$<15>="
3603 !
3606 FOR I=1 TO 8
3609 Run_iso(Run,I)=0
3612 NEXT I
3615 Parse(*Response$<2>}.Val(*}.Ninputs)
3618 IF Ninputs<2 THEN
3621 Clunk
3624 PRINT TABXY(1.12);C1:="**** YOUR "&Q6"ISOTOPES"&Q6" RESPONSE OF "&Response$(2)&" DOESN'T GIVE ME AT LEAST 2 ISOTOPES **** ":Cn#
3627 GOTO 3654
3630 END IF
3633 MAT Iso<0
3636 FOR I=1 TO Ninputs
3639 FOR J=1 TO Niso
3642 IF Val(I)=Magnet(J.I) THEN 3666
3645 NEXT J
3648 DISP C1:="**** ";Val(I);"ISN'T ON THE LIST OF DEFINED ISOTOPES FOR THIS ELEMENT-SERIES **** ":Cn#
3651 Clunk
3654 WAIT 3
3657 Response$<2>="??"
3660 GOTO Invoke_menu
3663 |
3666 Run_iso(Run.I)=Val(I)
3669 Iso(I)=Val(I)
3672 NEXT I
3675 IF Spike THEN | check compatibility of ELEMENT and SPIKE
3678 Spike0=Spike
3681 CALL Spike_check(Spike0.Iso(*}.Spikedrun_iso(*}.Spkdrun_ref) | check if the isotopes required by the spike are present
3684 IF Spike0=0 THEN
3687 Spike=0
3690 DISP FN$("**** THE ISOTOPES FOR SPIKE" "DUAL$(Spike)" "DON'T MATCH ELEMENT "RUNclass$" ")
3693 Clunk
3696 WAIT 2
3699 GOTO Mandat
3702 END IF
3705 END IF
3708 IF Spike THEN fract_normal=0
3711 IF fract_normal THEN
3714 IF Ulal(I)=Ref THEN ! 1st isotope must be the reference-isotope for fractionation-normalized runs
3717 DISP FN$(" THE FIRST ISOTOPE")&FN$("MUST")&FN$("BE")&CISB&FN$(Ref) "&UN$(
3720 GOTO 3651
3723 IF NOT Spike THEN
3726 FOR I=2 TO Ninputs
3729 IF Ulal(I)=Normal(I) THEN 3753
3732 NEXT I
3735 DISP FN$(" YOU MUST INCLUDE")&FN$(Normal(I))&FN$("AS ONE OF THE ISOTOPES")
3738 GOTO 3651
3741 END IF
3744 END IF
3747 END IF
3750 !
3753 Last_isotopes$=Response$¥(2)
3756 !
3759 Ref=Run_iso(Run,1) ! "Reference" isotope (either numerator or denominator of all ratios)
3762 Roughscan_done=1
3765 FOR I=1 TO 8 ! match the present-block’s isotopes against previous to see if need a new rough-scan
3768 IF (Run_iso(Run,I)<Lastblock_iso(I)) THEN Roughscan_done=0
3771 NEXT I
3774 IF SUM(Peak)=0 THEN Roughscan_done=0
3777 IF Roughscan_done=0 THEN
3780 MAT Peak= (0)
3783 MAT Order= (0)
3786 MAT Lastblock_iso= (0)
3789 GOSUB Mstart
3792 S=0
3795 FOR I=1 TO 8
3798 S=S+Run_iso(Run,I)
3801 NEXT I
3804 Ph_4678=(S>825 AND Ref=206) ! if a Pb-206-207-208-201 block
3807 END IF
3810 !
3813 PRINTER IS Prtr(2)
3816 IF Change_curr OR Beam_window OR Limit_growth THEN PRINT
3822 IF Beam_window THEN PRINT "BEAM SIZE TO BE RESTRICTED BETWEEN":Min=r:Max=r:U (for fil-currents: &Fmax: &R)
3825 IF Limit_growth THEN PRINT "MAXIMUM PERMITTED GROWTH-RATE OF BeAM IS":Maxgrowth:"%\ MINUTE."
3828 PRINTER IS CRT
3831 Changed_el=0
3834 !
3837 Gronk=FOR Blocknumber=1 TO Numberofblocks
3840 IF UPC$(RUNClass$(1,2))="HF" THEN
3843 !
3846 ! Hafnium runs only
3849 OUTPUT Bo$="H",Filenent$(*)
3852 WAIT 6*Mu
3855 L=0 ! Re-187
3858 Mu=0
3861 Correct(fil$(*)$)="HF",Filenent$(*)$)="HF","Magnet(L,2),0,1,Coarse(0),Foc(*)$)="HF")
3864" HF",Filenent$(*)$)="HF","Magnet(L,2),0,1,Coarse(0),Foc(*)$)="HF")
3867 Spike=0
3690 DISP FN$("**** THE ISOTOPES FOR SPIKE" "DUAL$(Spike)" "DON'T MATCH ELEMENT "RUNclass$" ")
3693 Clunk
3696 WAIT 2
3699 GOTO Mandat
3702 END IF
3705 END IF
3708 IF Spike THEN fract_normal=0
3711 IF fract_normal THEN
3714 IF Ulal(I)=Ref THEN ! 1st isotope must be the reference-isotope for fractionation-normalized runs
3717 DISP FN$(" THE FIRST ISOTOPE")&FN$("MUST")&FN$("BE")&CISB&FN$(Ref) "&UN$(
3720 GOTO 3651
3723 IF NOT Spike THEN
3726 FOR I=2 TO Ninputs
3729 IF Ulal(I)=Normal(I) THEN 3753
3732 NEXT I
3735 DISP FN$(" YOU MUST INCLUDE")&FN$(Normal(I))&FN$("AS ONE OF THE ISOTOPES")
3738 GOTO 3651
3741 END IF
3744 END IF
3747 END IF
3750 !
3753 Last_isotopes$=Response$¥(2)
3756 !
3759 Ref=Run_iso(Run,1) ! "Reference" isotope (either numerator or denominator of all ratios)
3762 Roughscan_done=1
3765 FOR I=1 TO 8 ! match the present-block’s isotopes against previous to see if need a new rough-scan
3768 IF (Run_iso(Run,I)<Lastblock_iso(I)) THEN Roughscan_done=0
3771 NEXT I
3774 IF SUM(Peak)=0 THEN Roughscan_done=0
3777 IF Roughscan_done=0 THEN
3780 MAT Peak= (0)
3783 MAT Order= (0)
3786 MAT Lastblock_iso= (0)
3789 GOSUB Mstart
3792 S=0
3795 FOR I=1 TO 8
3798 S=S+Run_iso(Run,I)
3801 NEXT I
3804 Ph_4678=(S>825 AND Ref=206) ! if a Pb-206-207-208-201 block
3807 END IF
3810 !
3813 PRINTER IS Prtr(2)
3816 IF Change_curr OR Beam_window OR Limit_growth THEN PRINT
3822 IF Beam_window THEN PRINT "BEAM SIZE TO BE RESTRICTED BETWEEN":Min=r:Max=r:U (for fil-currents: &Fmax: &R)
3825 IF Limit_growth THEN PRINT "MAXIMUM PERMITTED GROWTH-RATE OF BeAM IS":Maxgrowth:"%\ MINUTE."
3828 PRINTER IS CRT
3831 Changed_el=0
3834 !
3837 Gronk=FOR Blocknumber=1 TO Numberofblocks
3840 IF UPC$(RUNClass$(1,2))="HF" THEN
3843 !
3846 ! Hafnium runs only
3849 OUTPUT Bo$="H",Filenent$(*)
3852 WAIT 6*Mu
3855 L=0 ! Re-187
3858 Mu=0
3861 Correct(fil$(*)$)="HF",Filenent$(*)$)="HF","Magnet(L,2),0,1,Coarse(0),Foc(*)$)="HF")
3864  WAIT 1
3867  Size(5,0,0,6,6,1,100,1)  ! 5 to 8 volts 187
3870  IF Subflag=1 THEN Bnc
3873  L=RF  ! ref-peak
3876  Correct(Ff$(*),Mw$(*),Filament(*),Magnet(L,2),0,1,Coarsemag(I),Foc(*))
3879  WAIT 1
3882  END IF
3885  !
3888  IF Blocknumber>1 THEN Roughscan_done=1
3891  GOSUB Mgo
3894  IF Main1 AND (Block=1) THEN
3897  FOR I=1 TO H-1  ! normalize to 1st-block value
3900  IF POS(Ratio$(I),WAL$(Bnorn» IHEH
3903  Normal(I)=Bnorm
3906  Normal(2)=Rver(I)*Inv
3909  END IF
3912  NEXT I
3915  END IF
3918  NEXT Blocknumber
3921  !
3924  FOR K=1 TO 4  ! change filament-currents to specified after-run settings
3927  IF (K>2 OR Change_curr) AND Filament(K)<>Ff(K) THEN CALL Filament(10,Off_file(K),5,K))
3930  NEXT K
3933  Correct(Ff$(*),Mw$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsemag(L(0),Foc(*))
3936  GOTO Bnc
3939  !++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
3942  !
3945  Mfin=ADD GOSUB Clearall  ! Request a new sample to be rotated into position
3948  IF Run THEN PRINT TAB(1,18);"(Press 'Q'NH""Continue")A" with no response to escape)"
3951  A=0
3954  LINPUT "ENTER THE BARREL# OF THE NEW SAMPLE, NUMBER OF FILARENTS FOR THIS SAMPLE?",Input$
3957  Parse(Input$,Val(*>,Nir,puts)
3960  A=Val(1)
3963  IF A=0 THEN Skey_bnc
3966  IF A<16 THEN 3970
3969  Sample=B
3972  Ni’s=Val(2)=-1+2*(Val(2)=2 OR Val(2)=3)
3975  IF Ni’s=0 THEN
3978  PRINT TAB(11,12);:* PLEASE ENTER 2 VALUES: THE BARREL#, AND THE NUMBER OF FILARENTS (1 OR 3)
3981  Clunk
3984  GOTO 3940
3987  END IF
3990  OUTPUT KBD;clear$;
3993  Find(0,Sample,Ni’s,Subflag,Filament(*),line0)
3996  Changed_el=0
3999  Lastfil=0
4002  IF Daly THEN Daly=1  ! unless Daly broken, don’t inherit last run’s Daly setting
4005  IF Block THEN Run=Run + Run  ! increment Run# if took any data for previous sample
4008  IF Run>32 OR Run=0 THEN Run=1  ! Max. Run# is 32
4011  Run_name$(Run)="
4014  Filnum=1
4017  MAT Normal= Normal0
4020  Block=0
4023  Focnum=0
4026  Lastrate=10
4029  Spike=0
4032  GOSUB Stdfocus
4035  Whoop
4038  Zero(Filament(*),Zero_time,Noise_time,Daly,Magnet(L,2),Peak_inter,Mu,Mw$(*),Collector$(*))
1011 RETURN
1014 !
1047 Stdfocus: ! Restore std-focus values for single or triple filament samples
1050 ON ERROR GOTO Bad_read
1053 ASSIGN @Pathl TO "FOCUS:INTERNAL"
1056 ENTER @Pathl,Nfiles;Foc(*)
1059 OFF ERROR
1062 PRINT "STANDARD FOCUS-SETTINGS RESTORED"
1065 PRINT USING "X,2,6:30,3X,/,8(40,2X),/";"FOCUSING VALUES:";1,2,3,4,5,6,7,8,Foc(*)
1068 Correct(F$(*),M$(*),Filament(*),Magnet(L,2),Mu,I_t,Coarsenag(L<>0),Foc(*))
1071 RETURN
1074 !
1080 OUTPUT KBD;Clear$; \ clear alpha screen
1083 OFF KBD
1086 GRAPHICS OFF
1089 Stripchart=0
1092 RETURN
1095 !
1099 Skcm_bnc: Shiftlabel(stripchart,Sample_name$(*),Sample,Run,Full_auto,Nfiles)
1101 OUTPUT 8;M$$(Hu,I_t)
1103 GOTO Bnc_1
1105 !
1110 Clearkey: ! Undefine all softkeys & return
1113 FOR I=0 TO 19
1116 ON KEY I Call Clunk
1119 NEXT I
1122 RETURN
1125 !
1128 Bad_store:OFF ERROR \ failed disk-storage operation
1131 PRINT USING "2/,K","UNABLE TO STORE DATA ON DISK"
1134 Clunk
1137 WAIT 2
1140 GOTO Bnc_1
1143 Bad_read: OFF ERROR \ failed disk-read operation
1146 PRINT USING "2/,K","UNABLE TO READ DATA FROM DISK"
1149 Clunk
1152 WAIT 2
1155 RETURN
1158 !
1161 Recover_bnc;I_t=2 \ BMC key (k0) pressed during auto-running,
1164 PRINT IS CRT \ so revert to CRT condition
1167 Full_auto=0
1170 From_iso=Rf
1173 A=KNOBK
1176 OFF KEY
1179 Brop
1182 PRINT IS Prtr(2)
1185 PRINT USING ",K,";"++++ Exitd from auto-running (BMC-key), "&NClock_12$(TIME$(TIMEDATE))&" ++++
1188 GOTO Skey_bnc
1191 !
1194 Recover_quit=OUTPUT KBD;Clear$; \ QUIT key (k2) pressed during auto-run,
1197 GRAPHICS OFF \ so quit this auto-run & go on to next
1200 Brop
1203 PRINT IS Prtr(2)
1206 PRINT USING ",K,";"++++ Exitd from auto-running (QUIT-key), "&NClock_12$(TIME$(TIMEDATE))&" ++++
1209 PRINT IS CRT
1212 Whamp
1215 OFF KEY
1218 A=KNOBK
1221 PRINT TAB(19,1):"PRESS k1 TO CONFIRM REQUEST TO QUIT THIS RUN AND GO ON TO NEXT RUN."
1224 PRINT TAB(19,1):"PRESS k9 TO DENY REQUEST AND RESUME AUTOMATIC RUNNING OF THIS SAMPLE."
1227 ON KEY 1 LABEL " QUIT RUN" GOTO Quit
1230 ON KEY 9 LABEL " ESCAPE" GOTO Resume_auto
1233 GOTO 4233
1236 
1239 Auto! ********************AUTOMATIC RUNNING SEGMENT******************************
1242 ! ********************AUTOMATIC RUNNING SEGMENT******************************
1245 GOSUB Clearkey
1248 OFF KNOB
1251 DISP
1254 IF Auto THEN
1257 PRINT USING "18/,K,/,K,";"PRESS k2 TO RESUME AUTOMATIC RUNNING WHERE YOU LEFT OFF."
1260 PRINT USING "K,2/,K,2/,K,";"PRESS kO TO COMPLETELY RE-START AUTOMATIC RUNNING.","PRESS k.1 TO REVERT TO MANUAL RUNNING."
1263 ELSE
1266 PRINT USING "18/,1{)X,U/,1{)X,K,";"PRESS kl TO STflRT AUTOMATIC RUNNING.","PRESS k9 TO ESCflPE."
1269 END IF
1272 GOSUB Clearkey
1275 ON KEY 9 LABEL " ESCAPE" GOTO Skey>mic
1278 IF Auto THEN
1281 ON KEY 2 LABEL " RESUME AUTO" GOTO Resume_auto
1284 ON KEY 0 LABEL " NEW AUTO" GOTO 4323
1287 ON KEY 4 LABEL " MANUAL" GOTO 4305
1290 ELSE
1293 ON KEY 4 LABEL " START AUTO" GOTO 4323
1296 END IF
1299 GOTO 4299
1302 !
1305 Auto=0
1308 Full_auto=0
1311 Spike=0
1314 l_t=2
1317 GOTO Skey_bmc
1320 !
1323 DATA 1,1,1,0,1,0,0
1326 RESTORE 4323
1329 READ Auto,l_t,Run,Repeat_run,Full_auto,Dump_stripchart,Dump_datagraf
1332 MAT Preheated= (D)
1335 GOSUB Clearkey
1338 ON KEY 0 LABEL " DEFINED" GOTO 4365
1341 ON KEY 2 LABEL " NOT SURE" GOTO 4359
1344 ON KEY 4 LABEL " UNDEFINED" GOTO 4359
1347 ON KEY 9 LABEL " ESCAPE" GOTO 4305
1350 PRINT USING "18/,16K,K,B/ K ;""ARE YOUR AUTOMATIC-RUN VARIABLES DEFINED?"
1353 GOTO 4353
1356! 
1359 OFF KEY
1362 CALL Runvariables(Runvar(*),Run_name$(*),Runtype$(*),Sample_name$(*),Run_order(*),Run_iso(*),1)
1365 IF Runvar(1,1)>0 OR Runtype$(1)="" THEN
1368 PRINT USING "2/,K,2/,K,2/";"THERE ARE NO RUN VARIABLES IN MEMORY- NEED TO DEFINE OR GET FROM DISK"
1371 Clunk
1374 ON KEY 9 LABEL " ESCAPE" GOTO 4305
1377 WAIT 3
1380 OFF KEY 9
1383 GOTO 4359
1386 END IF
1389 OUTPUT KBD:Clear$;
1392 !
1395 Just_outgassing=1 ! Is this auto-run sequence just for outgassing? If so
4398 FOR I=1 TO 32  
  4400 I don't check the HU & don't worry about the beam-value.
  4401 IF Runvar(1,1) AND POS(UPC$(Runtype$(1):1:1:1,8),"OUTGRS")=0 THEN Just_outgassing=0
  4402 NEXT I
  4403 !
  4404 IF NOT Just_outgassing THEN
  4405 CALL Hu(Mnu(*),Hnu,O,Hu,I_t)
  4406 ON KEY 0 LABEL " READY" GOTO Autcont
  4407 ON KEY 4 LABEL " RECHECK" GOTO 4365
  4408 ON KEY 9 LABEL " ESCAPE" GOTO 4305
  4409 PRINT TRUE(Y(1,1:4):"CHECK THAT THE HU IS OK FOR ALL OF YOUR RUNS, AND THAT THE BEAM VALUE IS ""&Hnu$("OPEN")"
  4410 GOTO 4428
  4411 END IF
  4412 !
  4413 Autcont: ! Where variable indicates program line where auto-running should be resumed if interrupted
  4414 OFF KEY
  4415 OFF KBD
  4416 WHERE=0
  4417 PRINTER IS CRT
  4418 Outgas=POS(UPC$(Runtype$(Run)),"OUTGRS")  ! just outgassing for this run
  4419 Pregas=Outgas AND (UPC$(Runtype$(Run):E1:1:1:"P")=*P")  ! outgassing in preheat position
  4420 IF NOT Pregas THEN
  4421 Sample=Run_order(Run)  ! sample number (1-16)
  4422 ELSE
  4423 Sample=Sample-1:17<Sample=1)  ! to outgas in preheat position for barrel# N, rotate barrel# n-1 into running position
  4424 END IF
  4425 Nfils=HRunvar(Run,l)>1  ! Nfils is 1 if a single-filament sample, 2 if a triple
  4426 TypeO=Type
  4427 Ceer=0  ! until have a satisfactory beam
  4428 IF Outgas THEN Runtfind
  4429 !
  4430 FOR Type=1 TO 20  ! find appropriate element from disk, assuming original disk is in drive
  4431 IF Element$(Type)="" AND TRIM$(UPC$(Element$(Type))=UPC$(Runtype$(Run)) THEN 4545
  4432 NEXT Type
  4433 ON ERROR GOTO 4518
  4434 ASSIGN IPathl TO "TYPE:INTERNAL"
  4435 FOR Type=1 TO 20  ! look at each element-file on disk (maybe changed?)
  4436 ENTER IPathlJype;Runclass$,Niso,Hcoef<*>,Hagnet(*),Coarsetag(*>,Peak_inter,Hf, Nuclide$(*),Normal(*)NuInterfere(*)
  4437 IF TRIM$(UPC$(Runclass$)=TRIM$(UPC$(Runtype$(Run))) THEN 4515
  4438 NEXT type
  4439 !
  4440 OFF ERROR
  4441 FOR I=2 TO 1 STEP -1
  4442 PRINT Rer Prtr(I)
  4443 PRINT USING "8/",K,DD,K,0:0":"Cn$" ELEMENT ""&Runtype$(Run)*" NOT RECOGNIZED -- RUN",Run,Cn$
  4444 NEXT I
  4445 Superclunk
  4446 Wait(TIME,DATE,20,0,Magnet(L,1),Auto,Full_auto)
  4447 GOTO Do_next_run
  4448 !
  4449 OFF ERROR
  4450 IF (TypeO=Type) OR (Run=l) THEN GOSUB Autel  ! get element-series from disk if not already in memory
  4451 FOR I=1 TO 2  ! check HU twice in case of dropouts
  4452 CALL Hu(Mnu(*),Hnu,O,Hu,I_t)
  4453 IF P OR (ABS(Hu-Hu0)<20> THEN 4581
  4454 NEXT I
  4455 !
  4456 PRINTER IS Prtr(2)
  4457 PRINT USING "8/",K,DD,K,0:0":"Cn$" CAN'T RUN "ELEMENT$(Type)" ELEMENT-SERIES AT HIGH-VOLTAGE OF ",Hnu,Cn$
  4458 Superclunk
Wait(TIME_DATE,20,0,Magnet(L,1),Auto,Full_auto)
GOTO Do_next_run
!
Spike=Runvar(Run,26)
IF NOT Normal0(1) AND NOT Spike AND Ref(Run,1) THEN Ref=Run_iso(Run,1) ! accept arbitrary reference-isotope if possible
IF (Repeat_run=0) AND (Type0(0)Type) THEN GOSUB Stdfocus
Daily=Runvar(Run,5)
IF Rb_like=(Nfils=2 AND Runvar(Run,2)<187 AND Runvar(Run,3) AND Runvar(Run,4)) THEN treat as an "Rb" run if a triple with a non-Re focussing isotope
IF (Type0=0) THEN GOTO Stdfocus
Oaly=Runvar(Run,5)
Rb_like=(Nfils=2 AND Runvar(Run,2)<187 AND Runvar(Run,3) AND Runvar(Run,4)) ! treat as an "Rb" run if a triple with a non-Re focussing isotope
For an "Rb-like" run, the center-fil is taken up until a non-Re beam of specified intensity is found, then left at that current throughout the run
IF Spike THEN

Uhichspike(Spike,1,Subflag)
IF Subflag THEN

PRT(2):PRINTER IS PRTR
PRINT USING "2/,3a),10/";Cb$a" COULDN'T GET SPIKE\";Spike:" FROM DISK. ABORTING RUN ."&n#
Superclunk
WAIT 20
GOTO Do_next_run
END IF
!
PRINT IS CRT
Correct(Ff$(*),Mx$(*),Filament(*),Magnet(Rf,2),O,l,Coarse( &g<0),Foc(*))
WHERE=1
Autfind:OUTPUT XBO:Clear$:
OFF DELAY
PRINTER IS PRT(2)
IF Run=1 THEN PRINT USING "6/,7K,5(K),6/";"AUTOMATIC RUNNING STARTED AT \".FClock_12$(TIME$(TIME_DATE)), \", FNClock_12$(TIME$(TIME_DATE)), \",0\" AT\".TIME$(TIME_DATE), \"""
IF Run=1 AND NOT Outgas THEN PRINT USING "4/,3(K),4/";"STARTED RUN \".Run\", AT \",FClock_12$(TIME$(TIME_DATE)), \",0\" AT\".TIME$(TIME_DATE), \""
PRINTER IS CRT
IF NOT Repeat_run THEN
FOR I=1 TO 2
Find(0,Sample,Nfils,Subflag,Filament(*),find0)
IF Pregas OR Subflag=0 THEN Found_sample
PRINT USING "K,X,D,X,K,";"**** FAILED TO FIND SAMPLE ON TRY\".I
NEXT I
IF Subflag THEN
FOR P=2 TO 1 STEP -1
PRINT IS PRT(P)
PRINT USING "4/,3(K),4/";Cb$a"******** COULDN'T GET CONTACTS FOR SAMPLE AT BARREL\";Sample,"&n$
NEXT P
Superclunk
Wait(TIME_DATE,20,0,Magnet(L,1),Auto,Full_auto)
GOTO Abort
END IF
END IF
FOUND Samples:Serialnum=0
Lastfil=0
Badflag=0 ! no lost contacts yet
Fail_checkflags=0
IF NOT Outgas THEN
Sample_name$(Sample)=Run_name$(Run)
Zero(Filament(*),Zero_time,Noise_time,Daily,Magnet(Rf,2),Peak_inter,Mx,Mx$(*),Collector$(*))
}
IF Subflag THEN ! bad zeroes- can't run
    Superduperclunk
    GOTO Recover_bmc
END IF
END IF
HERE=2
PreheatCALL Flag(*),Nu) ! take up preheat filament-currents
Preheat sample=Sample=1-16*(Sample=16)
F=3*(Runvar(Run,24)>0 AND Runvar(Run,25)=0) OR (Runvar(Run,25)>0) ! triple or single?
IF F=4 AND (Flag(2)=1 OR (Flag(2)=2)) THEN ! triple-filament
    CALL Filament(60-50*(Runvar(Run,24)>1.5),Runvar(Run,24),5,3)
END IF
IF (F=3 AND Flag(2)=1) OR (F=4 AND Flag(2)=1) THEN ! take up sample filament(s)
    U=2+F

    IF Runvar(Run,1.5 THEN CALL Filament(3B,7.1,F))
    R=Runvar(Run,U)-Filament(F) ! remaining amperes to be gained
    CALL Filament(20,Runvar(Run,4)>2.5,(Fill)) ! takeup cf of triple
    CALL Filament((Runvar(Run,7)>>(Runvar(Run,6)>2.5,(F)) i takeup sanple-fil
    WAIT(TIME,Runvar(Run,8)*60,Filanent(Fil),Magnet<L,1>,fluto,Full_auto)
    CALL Filament((Runvar(Run,10)),Runvar(Run,9),.25,(Fil)
    \!
    PRINTER IS Prtr(2)
    PRINT "Sample at Barrel1:";TAB(18):Sample:TAB(22);"Outgassed at":;TAB(34):Filament(fill):TAB(41):"Amps (Center)"
    IF Filament(fill) THEN PRINT ", , :TAB(56):Filament(fill2):TAB(63):"Amps (Side)"
    PRINT USING "4/"
    PRINT IS CRT
    GOTO Do_next_run
END IF
IF Spike THEN MAT Normal= 0
WHERE=3
Filup: ! bring filament-currents to running temperature
    Nornal= Normal0

    GOSUB Key_escape
    GOTO 1818
    Key_escape:OH KEY 0 BBEL " BHC" RECOUER Recouerjnc
    OH KEY 1 LRBEL " RCH RUH" RECOUER Recouer_quit
    RETURN

    IF Outgas THEN ! just outgassing
        Block=0
        Fill=1+2*Pregas / filament $s for outgassing- center
        Fill2=2+2*Pregas / " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " 
    
    END
4920 IF Nfils=3 THEN ! triple-filament run
4923 ! Center-filament take-up for triple-filament runs
4926 IF Runvar(Run,2)=0 THEN L=0
4929 IF Runvar(Run,2) THEN
4932 FOR L=0 TO Niso ! find focussing-peak index
4935 IF Magnet(L,1)=Runvar(Run,2) THEN 4950
4938 NEXT L
4941 GOTO Abort
4944 END IF
4947 !
4950 Cf_iso=L
4953 !akeup Cf_where=4
4956 L=Cf_iso
4959 IF NOT Repeat_run THEN
4962 Barrel_wiggled=0
4965 R=100-90*Ret_like
4968 !
4971 ! take up center-filament to target-current
4974 IF Preheated(Sample,l) THEN CALL filament(3*R,Preheated(Sample,l),.5,1)
4977 CALL filament(3,Runvar(Run,4),.4,1)
4980 IF Subflag AND Barrel_wiggled THEN Abort
4983 IF Subflag AND NOT Barrel_wiggled THEN
4986 Wiggled_barrel(Barrel_position,Nfils,1,flag(*),NwH*)
4989 IF HFFilter(Nfils,flag(l))=0 THEN Abort
4992 GOTO 4977
4995 END IF
4998 END IF
5001 tuneup_cftonyWhere=5
5004 IF Runvar(Run,2) AND Runvar(Run,3) THEN ! work with focusing isotope
5007 IF Rb_like THEN ! a triple-filament rubidium-like run: take up cf to get specified isotope beam from turned-off sides
5010 ! wait 1 minute, get a <non-Re> beam by increasing the center-filament only (sides only)
5013 Wait(TIMEDELTA,60,filament(1),Magnet(Cf_iso,1),Auto,Full_auto)
5016 REPEAT ! center-focus on beam, adjust to required beamsize, pausing
5019 ! every 0.3 amps to re-center & re-focus.
5022 Center_peak(0,Cf_iso) ! center on focusing isotope
5025 Focus(3,2,1,Focnum)
5028 ! (Rb-like triple): get the minimum center-filament-only Rb beam (or up to 2 times the specified beam)
5031 Size(Runvar(Run,3),Runvar(Run,3)*2.0,filament(1)*3,filament(l)*3,1,2,30)
5034 UNTIL Subflag(2 OR filament(l)*8.5 OR filament(l)*1.5*Runvar(l,4)
5037 ! max. permissible CF current is 1.5x initial CF current
5040 IF Subflag(1) THEN Abort
5043 ELSE ! not a triple-fill, Rb-like run
5046 !
5049 REPEAT
5052 Center_peak(0,Cf_iso) ! center on the center-fill-only peak (Re-187)
5055 Focus(3,2,1,Focnum)
5058 Size(Runvar(Run,3)*1.1,Runvar(Run,3)*1.1,0,filament(1)*3,filament(l)*3,1,10,5) ! adjust the cf-only beam size to within +% 10% of spec.
5061 UNTIL Subflag(2 OR filament(l)*6.5
5064 IF Subflag(1) THEN Abort
5067 Focus(3,2,1,0)
5070 END IF
5073 END IF
5076 END IF
5079 !
5082 S=0 ! calculate sum of isotope-values to see if a Pb-4-6-7-78 run
5085 FOR I=1 TO 8
5088 S=S+Run_iso(Run,I)
5091 NEXT I
5094 L=RF ! work with reference-isotope
5277  Order(N)=I  ! order of data-taking isotopes in Magnet array
5280  GOTO 5289
5283  ENDIF
5286  NEXT J
5289  NEXT I
5292  RETURN
5295  
5298  LOOP
5301  CALL Rough(I,Peak(*)E,Data_daly,N,Order(*))
5304  IF E THEN ON I+AUTO GOTO Bnc, Recover_bnc
5307  Rank(Rp,Peak(*),L,N,Mp,Ref,Order(*),Magnet(*))
5310  Center_peak(I,L)
5313  Focus(3,2,1,Focun)
5316  Enter_bean(Counts,Mu,L,L1,0)
5319  IF Mu>10 THEN
5322  CALL Center_barrel(Sample)
5325  IF Subflag=1 THEN Abort
5328  ENDIF
5331  GRAPHICS OFF
5334  EXIT IF Subflag=0 OR Subflag=4
5337  IF filament(Nfils) Runvar(Run,13) THEN Abort
5340  CALL filament(100-99*Rb_like,Filament(Nfils)+.1-.05*Rb_like,25,Nfils) ! if no beam, raise filament current & try again
5343  END LOOP
5346  
5349  CALL Rough(I,Peak(*)E,Data_daly,N,Order(*))
5352  IF E THEN ON I+AUTO GOTO Bnc, Recover_bnc
5355  Rank(Rp,Peak(*),L,N,Mp,Ref,Order(*),Magnet(*))
5358  Center_peak(I,L)
5361  
5364  Retake: ! Pre-data-taking bean tuneup, all blocks
5367  Where=12
5370  Goer=1
5373  I+i
5376  Abort_count=0
5379  Target=Filament(Nfils)+(Block)0*Runvar(Run,18)*Decay(Runvar(Run,23)) ! new sample-fil. current
5382  L=Mp  ! work with the most-intense peak for bean tune-up & beansize checks
5385  Correct(fg(*),Mf(*),Filament(*),Magnet(L,2),Mu,1,Coarsenagi),Foc(*)&
5388  FF=Filament(Nfils)
5391  GOSUB Ertlabel
5394  IF ABS(Target-FF).0001 THEN CALL filament(2,Target,25,Nfils)
5397  GOSUB Checkflags
5400  GOTO Check_re
5403  
5406  Checkflags: FOR I=1 TO 3 ! check flags 3 times in case of GPIQ dropouts
5409  CALL Flag(flag(*),Mu)
5412  Ok=FNFilter(Nfils,Flag(I))
5415  IF Ok THEN
5418  Fail_checkflags=0
5421  RETURN
5424  ENDIF
5427  WAIT 2 ! wait 2 seconds before next check
5430  NEXT I
5433  PRINTER IS Prtr(2)
5436  PRINT USING "$s,13x,K,$s":"***** FAILED "CHECKFLAGS" FILAMENT-FLAG TEST *****"
5439  PRINTER IS CRT
5442  Fail_checkflags=1
5445  GOTO Abort
5448  
5451  Check_re: IF (Nfils=2) AND (Runvar(Run,2)=187) THEN
5097 Pb_4670=(Ref=206)×(S=905) | Pb_4670=1 if a Pb-206/207/208/204 run
5100 Correct(Fn(*),Mn(*),Filament(*),Magnet(L,1),0,1,Coarsemag(L,1),Foc(*))
5103 |
5106 Where=6
5109 CurrrxBarrel_wiggled=0
5112 GRAPHICS OFF
5115 IF Rununar(Run,6)(Filament(Nfils)) THEN (Currr)
5118 IF Rununar(Run,6)(Filament(Nfils)) THEN CALL Filament(30,7,1,Nfils)! take up to .7 amps rapidly if target 1.5 amps
5121 IF Preheated(Sample,Nfils)(Filament(Nfils)) THEN CALL Filament(2xRununar(Run,7),Preheated(Sample,Nfils),9,Nfils)
5124 Where=7
5127 Currr3=CALL Filament((Rununar(Run,7),Rununar(Run,6),.35,Nfils))  first target-current for sample filament(s)
5130 IF Subflag AND Barrel_wiggled THEN Abort
5133 IF Subflag AND NOT Barrel_wiggled THEN
5136 Wiggles_barrel(Barrel_position,Nfils,1,Flag(*),Mn(*))
5139 IF FMFilTest(Nfils,Flag(l))=0 THEN Abort
5142 Barrel_wiggled=1
5145 GO TO 5127
5148 END IF
5151 Start_wait=TIME
5154 |
5157 Where=8
5160 Wait1=Wait(Start_wait,Rununar(Run,8)×60,Filament(Nfils),Magnet(L,1),Auto,full_auto)
5163 |
5166 Where=9
5169 Currr2=Barrel_wiggled=0
5172 IF Rununar(Run,9)(Filament(Nfils)) THEN Start
5175 CALL Filament((Rununar(Run,10),Rununar(Run,9),.25,Nfils))  second target-current for sample filament(s)
5178 IF Subflag AND Barrel_wiggled THEN Abort
5181 IF Subflag AND NOT Barrel_wiggled THEN
5184 Wiggles_barrel(Barrel_position,Nfils,1,Flag(*),Mn(*))
5187 IF FMFilTest(Nfils,Flag(l))=0 THEN Abort
5190 Barrel_wiggled=1
5193 GO TO 5175
5196 END IF
5199 Start_wait=TIME
5202 |
5205 Where=10
5208 Wait2=Wait(Start_wait,Rununar(Run,11)×60,Filament(Nfils),Magnet(L,1),Auto,full_auto)
5211 |
5214 |
5217 Where=11  filaments at running-currents, start tuning-up beam
5220 GO SUB Mstart
5223 IF N=2 THEN
5226 FOR I=1 TO 2
5229 PRINT USE 10,6,
5232 PRINT USING "10,6x,6k,6r;" " PROGRAM ERROR - INVALID ISOTOPES PASSED FROM RUN VARIABLES 
5235 NEXT I
5238 PRINT USE 10,6,
5241 Superduperclunk
5244 GO TO 5127
5247 END IF
5250 GO TO 5298
5253 |
5256 Mstart=Good_blocks=0
5259 Verygood_blocks=0
5262 N=0
5265 FOR I=1 TO N Iso  calculate number & order of data-taking isotopes in the N-array
5268 FOR J=1 TO 8
5271 IF (Run_iso(Run,J)×J) AND (Magnet(I,1)=Run_iso(Run,J)) THEN
5274 N=N+1
5277 END IF
5628 M=Mu
5631 Mu=M+Dailyok(Order(K))
5634 OUTPUT B:##(Mu,I,t)
5637 IF Mu<0 THEN WAIT 5
5640 Center_peak<(K=1),Order(K))
5643 IF Subflag=4 AND NOT Auto THEN Skip_bnc
5646 IF Subflag=1 AND Auto AND K=1 THEN GOSUB Checkflags
5649 Badcenter=IF Subflag=1 AND NOT Auto THEN Uncentered_ks(K)=1 // couldn't center on Kth isotope
5652 IF Not Auto AND K=1 AND (Blocknumber/4=INT(Blocknumber/4) OR Fcnum=0) THEN CALL Focus(2,1,1,Fcnun) // focus every 4th block
or if unfocused
5655 IF Beam_window AND Not Auto AND (K=1) AND (Blocknumber=1) THEN
5658 L=Hip // restrict beamsize to specified window (manual only)
5661 Size(Umin,Unax,0.6,(Fmax),Nfils,0,10,15)
5664 IF Subflag=1 THEN Abort
5667 END IF
5670 IF K=1 AND gSubflag AND Auto THEN
5673 PRINT Nuclide$(Order(K));hagnet(Order(K),1);" NOT EXPLICITLY CENTERED";CHR$(13)aCHR$(10)
5676 Hagnet(K,2)=Hagnet(K)-Hagnet(Flip)+Hagnet(Hip,2)
5679 END IF
5682 NEXT K
5685 !
5688 IF Not Auto AND SUM(Uncentered_ks)=N THEN
5691 PRINT USING "2$,K,2/","伤口("************ Can'T CENTER ANY PEAKS************")
5694 Superclunk
5697 GOTO Bnc
5700 END IF
5703 !
5706 Scan_all=AUTO*(Block=2)*(Not Auto)*(Rough_scan_done=0) OR (SUM(Peak)=0) =1:0 if rough ratios of these isotopes are already known
5709 CALL Rough(Scan_all,Peak(*),E,Data_daily,N,Order(*))
5712 IF E THEN ON 1 Auto GOTO Bnc,Recover_bnc
5715 FOR I=1 TO 8
5718 Lastblock_iso(I)=Run_iso(Run,I)
5721 NEXT I
5724 IF Scan_all THEN ! calculate rough ratios
5727 Rank(Rp,Peak(*),L,N,Mip,Ref,Order(*),Magnet(*))
5730 FOR I=1 TO N
5733 IF I THEN Pkswitch_ratio(I)=Peak(I)/Peak(I)
5736 Data_iso(I)=Magnet(Order(I),1)
5739 NEXT I
5742 IF Pb_4678 AND Data_iso(4)<204 THEN Pb_4678=0 ! don't treat as natural-Pb run if 204 isn't the least-intense peak
5745 ELSE ! calculate beamsizes for all peaks using first rough-scan ratios
5748 FOR I=2 TO N
5751 Peak(I)=Peak(I)/Pkswitch_ratio(I-1)
5754 NEXT I
5757 END IF
5760 IF Peak(1)>(Daily=0)*10*(Daily=0)*2) THEN // if MIP-center didn't recalibrate mag-settings, normalize mag-settings for all other isotopes to MIP
5763 FOR I=1 TO Niso
5766 IF I>Mip THEN Magnet(I,2)=Magnet(I,2)-Magnet(2)-Magnet(Mip)
5769 NEXT I
5772 END IF
5775 !
5778 Take_data:Where=I? ! take isotope-ratio data
5781 GOSUB Printtime
5794 Valid_block=Bnc
5797 IF fil_in=filename(Nfils) AND Data_daily=Daly_out AND NOT Pb_4678 AND (TIME_DATE-Blockend_t(240)) AND ((NOT Auto)*(Blocknumber=1)
OR Auto*(Block=0)) THEN
5790 Share_bkgd=d1
5793 ELSE
! adjust size of Re-187 beam using center-filament
! correct sample-fil-currents for specified beam

WHERE=14
IF Block=0 THEN
REPEAT

! beam-size adjust before 1st-block: re-check for MIP every 1 amp, in
! case of large isobaric interferences
Size(Runvar(Run,2)/1.1, Runvar(Run,3)*1.1,0,10,6,1,10,15)
IF Subflag=1 THEN Abort
Correct(FF(*),Nfils(*),Filament(*),Magnet(2,2),Mu,1,CoarseMag(0),Foc(*))
END IF
WHERE=15
IF Runvar(Run,12)>60, Filet(Nfils), Magnet(L,1), Auto, Full_auto
END IF
WHERE=16
! final peak-centering & quick scan before data-taking block
GRAPHICS OFF
Hsets=Runvar(Run,22)
IF Subflag=1 THEN Abort
END IF
WHERE=20
FOR K=2-(H01 Block) OR (HOI Huto) 10 H
WHERE=15
IF Runvar(Run,12)>60, Filet(Nfils), Magnet(L,1), Auto, Full_auto
END IF
WHERE=14
IF Block=0 THEN
WHERE=15
Subflag=0
WHERE=16
END IF
WHERE=17
! adjust size of Re-187 bean using center-filament
Correct(FF(*),Nfils(*),Filament(*),Magnet(2,2),Mu,1,CoarseMag(0),Foc(*))
WHERE=18
! adjust sample-fil-currents for specified beam
WHERE=19
END IF
WHERE=20
WHERE=21
5961 IF pver(I)=0 THEN protect against zero-to-power error
5967 IF pver(I)=inv(Minrat) THEN Minrat=pver(I)=inv
5970 END IF
5973 NEXT I
5976 IF Minrat(I)=0 THEN Minbeam=Pk
5979 IF Minrat(I)=0 THEN Minbeam=Pk
5982 IF Minbeam<Runvar<Run,19) THEN Verygood_blocks(I)=Verygood_blocks(I) #blocks with good sigmas at MIP-MINBEAM
5985
5988 OUTPUT 0 USING "$H,4 12";"OMU",Good_blocks I display # of good blocks on the system
5991 FOR I=1 TO Good_blocks I tactic once for each good block
5994 IF I>1 THEN WAIT .07
5997 BEEP 660, .05
6000 BEEP 720, .05
6003 BEEP 660, .05
6006 NEXT I
6009 IF Block<Runvar<Run,20) THEN haven't reached MAX # BLOCKS yet
6012 IF (Good_blocks<Runvar<Run,19)) THEN Retake I take more blocks if required
6015 IF Verygood_blocks<Runvar<Run,19) AND Runvar<Run,17)<0 THEN I if default beam <0, then try and get good blocks with beam=MIN
6018 Runvar<Run,17)=Runvar<Run,14) ! default-beam = min beam
6021 Good_blocks<Verygood_blocks ! reset good-block counter
6024 GOTO Retake
6027 END IF
6030 END IF
6033 Quit: ! finished with data-taking for this run
6039 IF Block=0 AND NOT Outgas THEN Abort
6042 IF Block<1 AND NOT Outgas THEN
6045 Printres<Prtr<2),Run,E) ! print out results for this run
6048 Average<1),Normal<*,Landump,E,(Run),Prtr<*) I calculate wtd averages for this run
6051 END IF
6054 !
6057 Do_next_run<Run=32 THEN finish I 32 runs max.
6060 IF Run<order<1<Run) Then finish I if no next-run defined
6063 Repeat_run<Run<order<Run+1<Sample) !=1 if next run will re-use this sample
6066 IF Repeat_run THEN Runvar<1<Run,9)=MAX((Filament<Files)),<Runvar<1<Run,9))) ! don't decrease fil-currents if reusing same sample
6069 Run=1<Run ! go on to next run
6072 IF NOT Repeat_run THEN CALL Zero_fils<Filament<*)
6075 GOTO Autcont
6078 !
6081 !
6084 Abort: ! if flags OK, abort this run, print out machine-status data, & go to next run. But first, put 0.2 amps through any fils with
6087 ! less than 0.2 amps, & check the filament-flags.
6090 GRAPHICS OFF
6093 MAT Orig_fil= Filament
6096 FOR I=0 TO 1
6099 FOR J=1 TO 2
6102 IF Filament<1<2<1=1<.2 THEN OUTPUT 8;F8B(I,J),200
6105 NEXT J
6108 NEXT I
6109 IF NOT fail_checkflags THEN
6110 CALL flag<Flag<*,Mu>
6111 WAIT 2
6112 END IF
6114 Enter_beam<0,Mu,1,1,1,1)
6117 IF Mu<15 AND Fail_checkflags THEN ON 1<Where=12)+2<Where=16) GOTO 6120, Check_re, Badcenter
6120 IF FNFiltest<Files,Flag<1))>0 THEN
6123 Badflag=1+Badflag
5796 Share_bkgrds=0
5799 END IF
5802 ! share backgrounds between blocks if 1) filament-currents remain unchanged, 2) collector is the same, 3) no more than 4 minutes have elapsed
5805 !
5808 Start_block=TIME date! starting-time of block
5811 Fil_int=Filament(Nfils)
5814 !
5817 ! take isotope-ratio data
5820 Data=(Decay, Run_name#(Run), N, Block_number, Number_of_blocks, Share_bkgrds, Dump_data, Bad_pressure, Miso, Block, Data_daily, Pb-46, 78, Run)
5823 !
5826 IF Bnc_out THEN Recover_bnc
5829 IF Next_run THEN Recover_quit
5832 IF AUTO AND Bad_pressure THEN Bnc
5835 Blockend_t=TIME date
5838 Block_time=(<Blockend_t+Start_block>)/2-Start_time/60 ! time of block, in minutes
5841 GOSUB Printline
5844 Daily_out=Data_daily
5847 Reduced_current=0
5850 IF AUTO THEN PRINT USING "2/"
5853 Printer IS CR
5856 IF Block=Valid_block THEN AUTO=1 GOTO Bnc, Resume IF didn't complete block
5859 FOR I=M TO 10 ?
5862 Acc(I)=0
5865 Signa(I)="
5868 Delta(I)="
5871 Ratio(I)="
5874 Rver(I)=0
5877 NEXT I
5880 M=M+(Spike*(I>(5-(Spike-run_iso(I)<0))))) ! add ratios if a spiked run
5883 IF Spike THEN CALL Spikecorr(Rver(*), Last_av(*), Ratio(*), Acc(*), Lacc(*), Last_ratio(*), Delta(*), Prtr(*), (M), Block)
5886 Writedata(Rver(*), Ratio(*), file, Run_name#(Run), Signa(*), Ratio(*), Delta(*), Acc(*), Prtr(*), Block_time, Filament(Nfils), Block, Run, Sample, (M), Data_collect, Prtr(*))
5889 IF AUTO AND Limit_growth AND (Decay=Maxgrowth) THEN CALL Reduce(0, Maxgrowth), Filament(*), Runvar(Run, 12), Nfils
5892 ! max. permissible beam-growth is 1x "MAX GROWTH" for BERR(EMBER 88 block); 2x for BERR(EMBER 88 block); 3x for BERR(EMBER
5895 IF AUTO AND ((Pk<ABS(Runvar(Run, 17))) OR (Decay)>3*Runvar(Run, 23))) THEN
5898 IF (Decay=2*Runvar(Run, 23)) OR (Decay=3*Runvar(Run, 23)) OR (Decay=4*Runvar(Run, 14)) THEN
5899 Reduce(Reduced_current, Runvar(Run, 23), Filament(*), Runvar(Run, 18), Nfils)
5901 END IF
5907 END IF
5910 !
5913 !
5916 Resume: IF (Block=0) AND AUTO THEN Retake
5919 IF AUTO THEN RETURN
5922 !
5925 FOR J=1 TO M-1 ! test to see if precision or ratios for this block is ok
5928 Sigma=MSign2(Signa(J))) ! returns signa, negative if within limits
5931 S=Runvar(Run, 21) ! max. acceptable signa
5934 IF S=0 THEN ! theoretical limits OK in all cases
5937 IF S>(Acc(J) AND Sigma)=S THEN 5988
5940 ELSE ! theoretical limits only OK if within tolerance
5943 IF ABS(S)(Acc(J) THEN 5988
5946 END IF
5949 NEXT J
5952 !
5955 Good_blocks=Good_blocks+1 ! = # blocks with OK signas so far
5958 N=Runvar(9, I)*99 ! find size of MIP
5961 FOR I=1 TO M

38
READ Number_tries,Subflag,Daly_tested,Escape
FOR I=2 TO Full_auto 10
ON KEY I LABEL " " CALL Clunk
NEXT I
OFF KBD
OFF KNOB
ON KEY 9 LABEL "ESCAPE" GOTO Escape_center
L=C order of peak in the Magnet-array
PRINT "CENTERING "CHR$(129);Magnet(L,1);CHR$(129);"(starting value is ";VAL$;"Magnet(L,2));"
Mu=Magnet(L,2)
Lx1=Magnet(L,2)
Lx2=Lx1
B=Magnet(L,2)
IMAGE ";G2F";12
f1u=Mu*0.55
f1u=Magnet(L,2)
L=Magnet(L,2)
IMPRINT "$OFV",12
!
Correct(FFB$(*),Ms$(*),filament(*),B,Mu,2,Coarsemag(L,0),Fac(*))
IF Mu=0 THEN WAIT 6
FOR I=1 TO 2
Enter_beam(K,Mu,L,2,Pr)
IF Pr)1 THEN 6519
NEXT I
IF NOT Daly_tested AND NOT Mu AND Mu<10 AND Daly THEN
Mtest
Daly_tested=1
GOTO 6525
END IF
Number_readings=1+5*(Mu=0)*(Mu<10)*Mu*(Mu<30)*Mu*(Mu<1)*(Mu<.2)) ! number of .2-sec integrations on each side of peak
OUTPUT B;Mu$(Mu,2)
Centered=0
!
REPEAT
MRT Peakside(0)
FOR J=1 TO 2
OUTPUT B USING 6513: B<(2*J-3) As
FOR I=1 TO Number_readings
Enter_beam(K,Mu,L,2,Pr)
IF Pr)1 THEN 6519
NEXT I
IF I>1 THEN Peakside(J)=Peakside(J)+Mu/Number_readings
NEXT J
PRINT ROUND(Peakside(J),3);" ;";
NEXT J
Number_tries=Number_tries+1
Toler=(Mu=0)*.2+Mu*.006
IF (Number_tries>30) OR ((Peakside(1)>Toler) AND (Peakside(2)>Toler)) THEN
PRINT USING "22,3(X);";"**** CAN'T CENTER ",Magnet(L,1),"****"
IF Full_auto THEN CALL Clunk
Subflag=1
SUBEXIT
END IF
!
Sumside=Peakside(1)+Peakside(2)
Diffsides=Peakside(2)-Peakside(1)+1.E-9
IF ABS(Diffsides)<.01*1(Number_readings)*Sumside THEN Centered=1
!
Stepmult=1+1.5*(Sumside<3*(Mu=0)+.5-Mu) AND (Sumside<ABS(Diffsides)-1(.02))
! increase hunting step if difference in peakside intensities is large
Z=Stepmult*.21*As*(Diffsides<Sumside<.01 Sumside)) ! .21 is "hunting" rate
6615 \( B=B+2*|\text{ABS}(Z)|+1*\text{SGN}(Z)*|\text{ABS}(Z)|/(1) \)
6616 \( E=E+x \) \( \text{WHEN} \) \( \text{Centered}=1 \)
6617 \( Lx2=Lx1 \)
6618 \( Lx1=0 \)
6619 \( \text{PRINT TAB}(37);B \)
6620 \( \text{U} \)
6621 \( \text{PRINT TAB}(37);Lx1:B;Cn:CHR$(13);\text{CHR$(13);10}\) \)
6622 \( \text{Magnet}(L,2)=B \)
6623 \( \text{IF Recalibrate \text{AND} L \text{AND} ((Mu-D)*(Mu>10) \text{OR} Mu*(Mu>2)) \text{THEN} \)
6624 \( \text{IF good beam, use new neg-value to estimate neg-setting of all peaks} \)
6625 \( \text{Mcoef}(1)=\text{Mcoef}(1)*(B-\text{FINNmag}(\text{Mcoef}(1),\text{Magnet}(L,1)))) \)
6626 \( \text{FOR} \ I=1 \text{ TO} 24 \)
6627 \( \text{M=Magnet}(I,1) \)
6628 \( \text{IF M} \text{ THEN} \text{Magnet}(I,2)=\text{Mcoef}(1)+\text{M}(\text{Mcoef}(2)+\text{Mcoef}(3)*M) \)
6629 \( \text{NEXT} \ I \)
6630 \( \text{END IF} \)
6631 \( \text{Done:Correct(Ff$(*),hn$(*),Filanent(*),B,Hu,I_tO,Coarseag(L<>0),Foc/*))} \)
6632 \( \text{IF NOT futo} \text{ THEN} \text{BEEP 500, .03} \)
6633 \( \text{SUBEXIT} \)
6634 \( \text{Escape_center} \text{Subflag}=1 \)
6635 \( \text{GOTO Done} \)
6636 \( \text{SUBEND} \) ! --------------------------------------------------------------
6637 \( \text{Focus:SOB Focu5(Hax_cycles,Minjump,Quick,Foc_number)} \) \( \text{do automatic ion-optics focusing} \)
6638 \( \text{OPTION BASE 1 } \)
6639 \( \text{CON /General/ 2%,INTEGRAL Prtr(*),Subflag,Auto,full_auto,Foc(*),I_t0} \)
6640 \( \text{CON /Specs/ Mu(0:1),ions,2e(0:1),Noise(0:1)} \)
6641 \( \text{CON /Daly/ \text{INTEGER Mu,Daly,Mh(0:3,2),Daly_ekt(0:24),FF(0:1,2),(4)\) \)
6642 \( \text{CON /Magnet/ \text{INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse \)
6643 \( \text{CON /Filaments/ \text{INTEGER Ff$(0:1),Fil$(0:1),INTEGRAL Nfils,F8 \)
6644 \( \text{I} \) ! \)
6645 \( \text{INTEGER Tempfoc(0),Lastjump,Lastjump_bul1,Lastjump_bul2,I,Pr,C,Use(0),I_t,} \)
6646 \( \text{DIM Peak(3)} \)
6647 \( \text{DATA 1,1,1,1,1,1,1,1,61,20,0} \)
6648 \( \text{GCLEAR} \)
6649 \( \text{OFF KNOB} \)
6650 \( \text{READ Use(*),OffsetO,Maxtry,Subflag} \)
6651 \( \text{IF Foc_number=0 \ THEN \ First_focus=1} \)
6652 \( \text{FOR I=Auto*full_auto*2 TO 19} \)
6653 \( \text{ON KEY I \ LABEL " " \ CALL Clunk} \)
6654 \( \text{NEXT I} \)
6655 \( \text{OFF KBO} \)
6656 \( \text{OFF KNOB} \)
6657 \( \text{GOTO 7128} \)
6658 \( \text{ON KEY 9 \ LABEL " ESCAPE" \ GOTO 7128} \)
6659 \( \text{M=Daly_ekt(L)} \)
6660 \( \text{M=Daly_ekt(L)=M} \)
6661 \( \text{C=Coarsemag(L<>0)} \)
6662 \( \text{PRINT IS CRK} \)
6663 \( \text{IMAGE "$09",A,AZ} \)
6664 \( \text{MAT Tempfoc=Foc} \)
6665 \( \text{CALL Correct(Ff$(**),Mh(*),Film$(*),Magnet(L,2),Mu,2,C,Foc(*))} \)
6666 \( \text{CALL Enter_bean(Count,Mu,1,2,1)} \)
6667 \( \text{I_t}=\text{Mu(10)}+\text{M(90)} \)
6668 \( \text{IF Mu<20 \ AND NOT Mu AND Daly \ THEN} \)
6669 \( \text{CALL Mtest} \)
6825 IF Mu=0 THEN WAIT 6
6826 GOTO 6813
6831 END IF
6834 IF Mu>0.02 OR (NOT M)*M<0.8) THEN ! Scan trigger at .02 nV/SmV
6837 !
6840 Mu=(Daly>0)
6843 CALL Correct(ff$(*),Mv$(*),Filament(*),Magnet(L,2),Mu,2,Foc(*))
6846 Scan=1 ! if no apparent bean, scan each plate until a bean is found
6849 DATA 3,6,4,5,2,7,1,8 ! order of plates for panic-scan
6852 ON KEY 4 LABEL "SKIP PLATE" GOTO 6894
6855 RESTORE 6849
6858 FOR I=1 TO 6+Mfils
6861 READ Plate
6864 IF Use(Plate) THEN
6867 FOR J=10*(Plate=8> 10 999-159#(Plate=8> SIEP lQ-6*(Plate=1)-8*<Plate=8)
6870 OUTPUT 8 USING 6804:Z4C10-Plate,10-Plate,J
6873 CALL Enter_bead(X,My,L,2,Pr)
6876 DISP "SCANNING TO FIND BEAM"; Plate:"=";J;"AB<45>";"BEAM<";"GROUND(Mu,2)";"nV"
6879 ON Pr GOTO 6882,6942,6810
6882 IF Mu>0.02 OR (NOT Mu)*M<0.8) THEN 6912
6885 NEXT J
6888 OUTPUT 8 USING 6804:Z4C10-Plate,10-Plate,Tempfoc(Plate)
6891 END IF
6894 NEXT I
6897 PRINT USING "/,K,2/":Fm$("UNABLE TO FIND BEAM")
6900 Subflag=1
6903 Clunk
6906 SUBEXIT
6909 !
6912 Foc(Plate)=J+10-6*(Plate=1) ! found a bean during the panic-scan
6915 Tempfoc(Plate)=Foc(Plate)
6918 END IF
6921 !
6924 IF First_focus THEN
6927 Graph_ymax=4
6930 Ymax=10000
6933 Graph_ymin=-2*Mu
6936 Xmax=600
6939 ELSE
6942 Graph_ymax=Mu*(1.3+5*(Foc_number<2))
6945 Ymax=Graph_ymax
6948 IF Graph_ymax<1.E+4 THEN Graph_ymax=1.E+4
6951 Xmax=300
6954 Graph_ymin=0
6957 END IF
6960 Y=Mu
6963 CALL Axes200,100,8,100,0,Ymax,Graph_ymin,Graph_ymax,"STEP (ION OPTICS FOCUS)","nV" "WURLS(Magnet(L,1)),0,Mu,(First_focus))
6966 Foc_number=1+Foc_number
6969 CSIZE 2.8
6972 !
6975 FOR Cycle=1 TO Max_cycles
6978 Change=0
6981 Focetime=Focetime+10
6984 Mu=Mu
6987 IF Cycle=1 OR Scan THEN
6990 CALL Correct(ff$(*),Mv$(*),Magnet(L,2),Mu,L,M,Foc(*))
6993 CALL Enter_beam(X,Mu,L,M,Mu)
6996 IF NOT First_focus AND Mu<1.2*Ymax THEN
6999 Ymax=2*Mu
7002 Graph_ymax=Ymax
7005 IF Vmax>1.E+4 THEN Vmax=1.E+4
7008 Focctime=0
7011 Vmax=Vmax*.7
7014 CALL Axes(20,100,8,100,0,Vmax,0,Vmax,"CYCLE","MU BEAM",0,Mu,0) ; draw plot-box for graphics display
7017 END IF
7020 END IF
7023 OffsetO=OffsetO/(1+(OffsetO/16))
7026 Min_jump=Min_jump/(1+(Min_jump/1)) ; resolution, in focus units
7029 I_t=1+Max*(Mu3)+(NOT Mu)*(Mu15)
7032 IF Mu AND Daily AND (F>10) THEN
7035 Mu=0
7038 Daily_ok(L)=0
7041 END IF
7044 CALL Correct(Ff$(*),Mu$(*),Filament(*),Magnet(L,2),Mu,I_t,C,Foc(*))
7047 IF Mu THEN UNIT 6
7050 ON KEY 4 LABEL " SKIP PLATE" GOTO Plate_done
7053 !
7056 Plates: ! optimize beam for each focusing-potential
7059 FOR Plate=1 TO 10+5*Hfils
7062 IF (Use(Plate)=0) OR (Plate=1)<Cycle=1> OR (NOT Quick)=(Plate<>1) OR (Plate<>1) OR (Plate<>1) OR (Plate<>1) THEN Nextplate
7065 Min_value=60*(Plate=0) ; minimum allowable value for this plate
7068 Max_value=999-459*(Plate=0) ; maximum " " " "
7071 Point_plole=0
7074 Focctime=Focctime+6 ; graphics X-axis (# changes)
7077 Mu=Noise(Mu)*2
7080 G=2*E10-Plate,10-Plate3
7083 Offset=Offset0
7086 IF (Plate=0) AND (Offset=0) THEN Offset=0
7089 IF Plate=1 THEN CALL Correct(Ff$(*),Mu$(*),Filament(*),Magnet(L,2),Mu,I_t,C,Foc(*))
7092 GHR 1,1,0,0,0
7095 RESTORE 7092
7098 READ Num_jumps,Jump_dir,Lastjump,Lastjump_bul1,Lastjump_bul2
7101 MOVE Focctime+2,Y+Graph_ymax/20
7104 PRINT
7107 LABEL Plate
7110 !
7113 Reset_max=DISP FHHR1((URL$1)(Plate))&" "MU ";TAB(30);"CYCLE";Cycle;"OF";Max_cycles;"CHANGE";Change
7116 Updown=0
7119 GOSUB Foc_change
7122 GOTO Next_jump
7125 !
7128 Foc_change:OUTPUT 8 USING 6804:0$,tempfoc(Plate)+Updown*Offset
7131 PRINT URL$(tempfoc(Plate)+Updown*Offset):
7134 FOR J=1 TO I_t
7137 CALL Enter_beam(X,Mu,I_t,Pr)
7140 NEXT J
7143 ON Fr GOTO 7146,7077,6990
7146 Peak(2*Updown)+Mu
7149 PRINT TAB(6):URL$1(OROUND(Mu,3)<(Mu1000)))
7152 Focctime=Focctime+1
7155 IF first_focus THEN
7158 IF Mu=0 THEN
7161 Y=LGT(Mu)
7164 ELSE
7167 Y=-Mu
7170 END IF
7173 ELSE
7176 Y=-Mu
7179 END IF
7182 IF NOT Point_plole THEN
IF Badflag=1 THEN ! re-find sample & try again
FOR Y=2 TO 1 STEP -1
PRINT USING "3/(K,3)","** No FILAMENT-CONTACTS <sample ",Sample,">) -- TRYING TO RECOVER BY RE-FINDING SAMPLE **"
NEXT Y
Superclunk
Find(0,Sample,Nfils,Subflag,Filament(*),0)
IF Subflag THEN Give_up
'FOR I=1 TO 4 ! take up fil-currents to initial values
IF (I=1 AND Nfils=1) OR (I=2 AND Nfils=2) THEN
take up sample-filament rapidly to 85% of orig. curr. then slowly to orig. curr
CALL Filament(200,Orig_fil(I),5,0))
CALL Filament(10,Orig_fil(I),5,0))
ELSE
IF Fil(I) THEN CALL Filament(200,Orig_fil(I),5,0))
END IF
NEXT I
IF (Filament(3)=0) AND (Filament(4)=0) THEN Call Wait(TIME$(DATE),FILAMENT,Nfils,Magnet(L,1),Auto,Full_auto)
GOTO Resume_auto
END IF
END IF
Give_up:FOR Y=1 TO 2
PRINTER IS Prtr(Y)
PRINT USING "3/(K,/)";RPT$<"*",80),RPT$<"*",80),RPT$<"*",80)
PRINT "*<*» BBORTED RUN FOR REPAIR"*Sample: "Run8":Run:**
PRINT "CENTER FIL.=":Filament(1),"SIDE FIL.=":Filament(2),"TIME IS ":FNClock(12$<TIME$(DATE))
PRINT "GROUNDFil(1):";FILANTERT(1),"DEBM. Where=":Where
PRINT USING ",K,2/";FILANTERT(1,"FOCUS VALUES":1,2,3,4,5,6,7,8,Foc(*))
PRINT USING ",K,2/";RPT$<"*",80),RPT$<"*",80),RPT$<"*",80)
IF Y=1 THEN Call Superclunk
NEXT Y
GOSUB Key_escape
FOR I=1 TO 10
PRINT .2
NEXT I
NEXT I
IF Block THEN
Prntres(Prtr(2),Run,E)
IF Block=1 THEN CALL Average(1,Normal(*),Candump,E,Run,Prtr(*))
ELSE
Writedata(Auer(*),Katio(*),Ffile,Run_name(*),Run,Signa(*),Delta(*),Rec(+),Pk,Block_time,Filament:Nfils),Block,Run,Sample,M
Data_collector,Prtr(*))
END IF
IF NOT Outgas AND Goer=0 AND Mv.2 AND (Filament:Nfils<Runvar<Run,13) OR (Filament(1)<6) THEN Abort_count=Abort_count+1 ! # of successive no-beam aborts
IF Abort_count=2 OR Just_outgassing THEN Do_next_run
FOR Y=2 TO 1 STEP -1
PRINT USING "10/(K,3)","** AUTO OPERATION SUSPENDED -- 2 SUCCESSIVE NO-BEAM ABORTS**" :)
PRINT USING ",K,2/";"PRESS RECALL key to resume auto-operation"
NEXT Y
Superduperclunk
Zero_filts(Filament(*))
GOTO Bnc
Finish:PRINT USING "K,^,10fl,/,6X,K,/,40fi,2/,K";CHR$(27)<&"aHS n",RPT$("» H JO)," RUNRTERIC RUNNING FINISHED",RPT$","aHS<CHR$(27)"a"a"aOS
MOVE Focline,Y

Point_plotted=1

ELSE

DRAW Focline,Y

END IF

Num_jumps=Num_jumps+1*(Offset=Min_jump)

IF Num_jumps=Maxtry THEN Plate_done

RETURN

Next_jump=1/(Plate-8)*(FNLinit(Tempfoc(Plate)+2*Jump_dir*Offset,Min_value,Max_value)=0)

B=(Last_jump=Updown*Offset)*(Last_jump=Updown*Offset)*(Last_jump=Updown*Offset)

IF A AND B THEN Offset=2*Offset ! double offset if last 3 offsets stayed the same, in order to converge on true max more rapidly

IF FNLinit(Tempfoc(Plate)+Jump_dir*Offset,Min_value,Max_value) THEN ! new value not within limits- reduce offset

IF NOT Tracking_incr THEN Change_offdir

Offset=Offset/(1+(Offset=Min_jump))

ON 1*(Offset=Min_jump) GOTO 7224, Plate_done ! plate focused if can't reduce offset any more

END IF

Updown=Jump_dir

GOSUB Foc_change

Last_jump_up2=Last_jump_up1

Last_jump_up1=Last_jump

Last_jump=Updown*Offset

IF (Peak(2+Updown)=Peak(2)+FNDiff(Peak(2),Peak(2+Updown),No,I_t,Ions)) AND NOT FNLinit(Tempfoc(Plate)+Updown*Offset,Min_u value,Max_value) THEN

! if new plate-setting gives a beam greater than the previous-max + theor. noise, then define as the new max-setting & beam

Peak(2)=Peak(2+Updown)

Peak(1)=0

Peak(3)=0

Tempfoc(Plate)=Tempfoc(Plate)+Updown*Offset

GOTO Next_jump

END IF

Offset=Offset/(1+(Offset=Min_jump))

IF FNLinit(Tempfoc(Plate)-Jump_dir*Offset,Min_value,Max_value) THEN

! if proposed new setting isn't within permissible limits of this plate, cut offset in half again

Offset=Offset/(1+(Offset=Min_jump))

ON 1*(Offset=Min_jump) GOTO Reset_max, Plate_done

END IF

! if

Updown=Jump_dir ! change offset direction

GOSUB Foc_change

IF (ABS(Peak(3)-Peak(2))/FNDiff(Peak(2),Peak(3),No,I_t,Ions)) AND ABS(Peak(1)-Peak(2)-FNDiff(Peak(2),Peak(1),No,I_t,Ions)

) THEN Plate_done

IF Peak(2+Updown)=Peak(2) THEN

Change_offdir=Tracking_incr=1

Jump_dir=Jump_dir

IF FNLinit(Tempfoc(Plate)+Updown*Offset,Min_value,Max_value) THEN Plate_done

Tempfoc(Plate)=Tempfoc(Plate)+Updown*Offset

IF Peak(1)=0

Peak(3)=0

GOTO Next_jump

END IF

IF Peak(2)=0 THEN Reset_max

IF Offset=Min_jump THEN

Offset=Offset/2

GOTO Reset_max

END IF

!
PlateDone: IF Plate > 8 THEN NextPlate

Change = Change + ABS(Foc(Plate) - tempfoc(Plate)) \( BUT \) sum of changes for this cycle

Foc(Plate) = tempfoc(Plate)

NextPlate = NEXT Plate

IF Change < 2 THEN 7380

BEEP 1000, .05

NEXT Cycle

IF I = 1 TO 7

IF (Foc(I) > 0) THEN Lowfoc = Lowfoc + 1

IF (Foc(I) < -9999) THEN Highfoc = Highfoc + 1

NEXT I

IF Lowfoc > 1 OR Highfoc > 1 OR Lowfoc = Highfoc THEN

PRINT USING "2,/,k":FNH$("***** SUSPECT FOCUS SETTINGS: POSSIBLE HARDWARE PROBLEMS *****")

GRAPHICS OFF

CALL BEEP(160, .03, .03, 60)

END IF

CALL Whoop

OUTPUT 8;KEY(Mu,I, I_10)

PRINT TAB(2, 17):;

PRINT USING "0(0,4X),k":1,2,3,4,5,6,7,8,"-PLATE"

PRINT USING "2X,0(30,2X),k":Foc(*),"-SETTING"

DISP

SUBEDIT

Subflag = 1*(fluto = 0)

BEEP

?13t CALL Correct($f(*),Mu(*), filament(*), Magnet(I,2),Mu,I_tO,C,Foc(*))

GOTO 7110

SUBEND

--------------

Subflag = 1*(fluto = 0)

BEEP

?13t CALL Correct($f(*),Mu(*), filament(*), Magnet(I,2),Mu,I_tO,C,Foc(*))

GOTO 7110

SUBEND

--------------

Barrel: SUB Center_barrel(INTEGER Sample)

OPTION BASE 1

COM /General/ ZK,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_10

COM /Specs/ Mx(I:1),Ions,2e(0:1),Noise(0:1)

COM /Daily/ INTEGER Mu,Daly,Mm(I:3),2e(1),Daly_ok(O:29),Ff(I:0:1,2:1)

COM /Magnet/ Icon(1:1),INTEGER L,Inside,Peak_inter,Coarsemag(0:1),Magnets(0:2,1),Coarse

COM /Barrel/ INTEGER Barrel_position,Barrel_posO,Max_barrel,Min_barrel

COM /Filaments/ filament(*),fils(*),INTEGER Nfils,fB

REAL F1(4),Barrel_ok(100),F1(2),P2(2),Flg(2)

INTEGER I,Pr,B,I_t,Bnin,F_subflag

DATA 0,0,0,0

READ Subflag,Find_flag,f_subflag,f_subcheck \( BUT \) zero flags

BEEP

IF filament(I) < .2 AND filament(I) > .2 THEN

PRINT USING "10,14X,داخل":"Fmh$("SORRY, CANT CENTER THE BARREL WITHOUT A BEAM")

Clunk

SUBEXIT

END IF

OFF KBD

OFF KEY

OFF KNOB

DEG

Redo:Bnin = Min_barrel + 3*(filament(3) - filament(4)) \( BUT \) lower barrel-limit

Bmax = Max_barrel - 7*(filament(3) > 0 OR filament(4) = 0) \( BUT \) upper barrel-limit

MAT Daly_ok = 0

Daly_ok(L) = Mu

Barrelscan = 0
Correct($f$,\mu$), $filament(\mu_l), Mag_{\nu}(L_{\nu}), Coarse_{\nu}(L_{\nu})$)

Bruni:0;Mn\$($2+Mn$,$l$I$I$)

Enter $bean$($0,MnG,U,l$I$)

If HuO(15 AND Daly AND (Mu=0) THEN 1 small beam - use Daly

Mtest

GOTO Brun

END IF

If NOT Auto AND (Mu(0.05) OR (NOT Mu) * (Mu(0.2) THEN

PRINT USING "1D,6K.K,3/";FМ$\"CAN\nI CENTER BARREL ACCURATELY UNLESS A SIGNIFICANT BEAM IS PRESENT"

PRINT USING "1W,6K.K,10/"; "press k9 to escape, or k0 to try anyway."

Clunk

ON KEY 0 LABEL " TRY" GOTO 7599

ON KEY 9 LABEL " ESCAPE" GOTO 7581

GOTO 7575

!

OUTPUT KBD;CHR$(255)$CHR$(75)$

Subflag=1

OUTPUT 8;Mn$($2+Mn$,$l$I$I$) ! restore initial integration time

SUBEXIT

END IF

!

FOR I=0 TO 19

ON KEY I LABEL " CALL Clunk

NEXT I

ON KEY 9 LABEL " ESCAPE" GOTO Exit_barrel

!

I_t=1+Mu*(HuO$>$l)+NOT Mu)*(HuO$>$10)$

OUTPUT 8;Mn$($2+Mn$,$l$I$I$)

ON KEY L LABEL "" CRLL Clunk

NEXT I

ON KEY 9 LABEL " ESCAPE" GOTO Exit_barrel

!

Axes(0,100,25,100,0,Min$,$bmax1.0,Ymax, "BARREL POSITION", "WUR$\"Magnet(L,1)), 1,Mn$,$)

LONG 5

MOVE (bmin$,$bmax)/2,1+Min$,$

CSIZE 4.5

IF NOT Barrels then LABEL "BARREL-FOCUS"

MOVE Barrel_posO,1.76*Mn$,$

LINE TYPE 4

DRAW Barrel_posO,2*Mn$,$

MOVE Barrel_posO,0.55*Mn$,$

LINE TYPE 1

LONG 4

CSIZE 3

LONG 90

LABEL "<ORIGIN>

LABEL "<ORIGIN>

END IF

!

Bgo: CALL $Brln(flag,l)$

GOSUB Check

Barrel_position=bmin

CALL $Brln(Barrel_position,flag,l)$

GOSUB Check

DISP

IF FNFiltest(Nfils,FlagH RNO NOT Refind THEN ! lost contact at lower barrel-range

Error_message(Fnt(*)+1+2*Full_auto, "UNABLE TO KEEP FILAMENT-CONTACT DURING BARREL-ROTATION")

IF f_check=0 THEN
CALL Flag(flag(*),2*Mu)
Flag=flag(I)
F_bchek=1
GOTO 7704
END IF
GOSUB Refind
Refind=1
GOTO Redo
END IF

FOR I=1 TO Bmax-Bmin+1  "scan barrel"
GOSUB Barrelscan
NEXT I
END LOOP
CALL Brl(Barrel_position,Flag,0)
Ok=(NFiltest(Mfiles,Flag)) ! lost contact at upper barrel-range?
EXIT IF Ok OR F_acheck
Error_message(Prtr(*>,1)*2*Full_auto,"UNABLE TO KEEP FILM-CONTACT DURING BARREL-ROTATION")
F_acheck=1
WAIT .2
END LOOP
GOSUB Refind
Refind=2
GOTO Redo
END IF

FOR J=1 TO Bmax-Bmin+1
IF Max_beam<Barrel_pk(J) THEN
Barrel_position=Bmin+J
Max_beam=Barrel_pk(J)
END IF
PRINT
MOUSE Barrel_position-.06,.5*Max
RECTANGLE .12,.85*(Max_bean-.5*Max),FILL ! draw a pointer to new max
CALL Brl(Bmin,Flag,1)
GOSUB Check
CALL Brl(Barrel_position,Flag,1)
OUTPUT 8;Hx$(0,2,1)
Barrelscans=1+Barrelscans
WAIT .5
GOSUB Check
WAIT .4
Enter_bean(0,Renax,1,1_t,l)
IF (Renax<.85*Max_bean) AND (Barrelscans<2) THEN Bgo ! Re-scan if didn't recover at least 85% of the original beam-intensity
IF Barrelscans<2 AND ((Enax-Barrel_position<3) OR (Barrel_position-Bmin<3)) THEN
Barrel_pos0=Barrel_position
CALL Brl(Bmin_barrel=2,Flag,.5)
GOTO Bgo ! reoptimize barrel if position is within 2 units of losing contact.
END IF
Barrel_pos0=Barrel_position
OUTPUT 8;Hx$(0,1_t,10) ! restore integration time of calling environment
DISP
Daly_ok(L)>=Mu
Whoop
SUBEXIT
Barrelscan: Barrel_position = Barrel_position * l

CALL Brl(Barrel_position, -1, 2*(I_t+2), 0)

Enter_beam(0, P2(2), l, l, l)

IF Pr=2 THEN Brun

IF Pr=3 THEN Bnp

IF P2(2)<0 THEN CALL Correct(Fs$, *, M$, *, Filaent(*), Magnet(l, 2), Nu, l, t, Coarsemag(L)0), Foc(*))

Barrel_pk(1) + P2(2)

P2(1) = Barrel_position + 0.5

IF Started THEN CALL thickpen(P1(*), P2(*), Nu/40, (2-Barrels);scans)/2)

Started = 1

MAT P1 = P2

RETURN

Exit_barrel: Barrel_position = Barrel_position

Subflag = 4

Broop

GOTO 7830

! Checks! check for runaway barrel. If found, turn off fills, rotate sample back into position again, take up fills to previous currents, re-center barrel

FOR G=0 TO 1

WAIT .4

IF I_t+1 THEN OUTPUT 8; "$OM?9?;8"

IF I_t=2 THEN OUTPUT 8; "$OM?9?;8"

OUTPUT 8; "$M"  

ENTER 0: P  

IF P=1 THEN barrel stopped moving- everything ok

IF P=0 THEN barrel is in motion.

OUTPUT 8; M$*(Mu*2, I_t)

RETURN

ELSE

BEEP 100, .05

NEXT G

IF Find_flag AND full_auto THEN only permit 1 attempt at recalibration

Rats = Subflag 1  

Rats. Can't keep contact and/or non-runaway barrel.

Error_message(Prtr(*), 3, "UNABLE TO KEEP FILAMENT-CONTACT DURING BARREL-ROTATION")

Zero_fils(Filaent(*))

SUBEXIT

END IF

MAT F1 = Filaent

Zero_fils(Filaent(*))

Superclunk

GRAPHICS OFF

FOR I=1 TO 1 + Auto

PRINTER IS Prtr(I)

PRINT USING "3", 2*(9x, 60, /), 9x, 9x, 6000: $RPT$(*, *, 60), $RPT$(*, *), 22% "RUNWAY BARREL *(RPT$(*, 22), RPT$(*, 60)

PRINT USING "3"

IF I=1 THEN

PRINT "Sorry- something has caused the barrel to rotate on its own."

PRINT USING "2(/, K) /", "Keep calm. The computer will rotate the sample back into position, " and restore the filament-currents (honest!)."

END IF

NEXT I

PRINTER IS CRT

Find(0, Sample, Nfils, F_subflag, Filaent(*), 0)

IF F_subflag THEN 8061
8067 find_flag=1*find_flag
8070 GOSUB Restore_fils
8073 GOTO Redo
8076 !
8079 Refind:IF find_flag AND Full_auto THEN Rats
8082 !
8085 MAT Fl> filament
8088 GRAPHICS OFF
8091 Clunk
8094 IF NOT Auto AND (Max_barrel-Min_barrel)>15) THEN
8097 PRINT "PRESS "$AFN(=k4)=" FOR AN AUTOMATIC CONTACT-RECALIBRATION."
8100 PRINT USING "/",k,2,"/";"PRESS "$AFN(k4)=" TO JUST NARROW THE SCAN-RANGE BY 5 UNITS."
8103 OFF KEY
8106 ON KEY 0 LABEL "RECALIBRATE" GOTO 8139
8109 ON KEY 9 LABEL "SIMPLE ADJUST" GOTO 8118
8112 GOTO 8112
8115 !
8118 IF Lower THEN Min_barrel=Min_barrel+5
8121 IF NOT Lower THEN Max_barrel=Max_barrel-5
8124 CALL Brl((Max_barrel-Min_barrel)/2,-1,1.5)
8127 OFF KEY
8130 RETURN
8133 END IF
8136 !
8139 PRINT USING "18",k,2,"/";FNH("RECALIBRATING CONTACT POSITIONS FOR THIS SAMPLE - PLEASE DON'T INTERFERE.")
8142 PRINT "+(filament-contacts were lost during the barrel-scan)"
8145 Find(I,Sample,Nfiles,f_subflag,filament(*),0)
8149 IF F_subflag THEN 8061
8151 find_flag=1*find_flag
8154 OFF KEY
8157 GOSUB Restore_fils
8160 RETURN
8163 !
8166 Restore_fils:FOR I=1 TO 4  Take up fil-currents to initial values
8169 IF (I=1 AND Nfiles=1) OR (I=2 AND Nfiles=2) THEN
8172 ! take up sample-filament rapidly to 85% of orig. curr, then slowly to orig. curr
8175 CALL filament(200,0.85*FI(I),5.(1))
8178 CALL filament(10,FI(I),5.(1))
8181 ELSE
8184 IF FI(I) THEN CALL filament(200,FI(I),5,(1))
8187 END IF
8190 NEXT I
8193 MAT filament= FI
8196 IF Nfiles=0 THEN Nfiles=1
8199 IF (filament(5)=0) AND (filament(4)=0) THEN CALL Wait(TIME(5),15,Filament(Nfiles).Magnet(L,1),Auto,Full_auto)
8202 RETURN
8205 SUBEND !  ............................................................................
8208 !
8214 filament:SUB filament(Rate,Target,Pcheck_amps,INTEGER Filnumber)
8217 \ change current of specified filament to target-value at specified speed
8220 OPTION BASE 1
8223 COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Fac(*),I_t
8226 COM /Keyboard/ CN$,CSS,CDS,CUR,0,Clear$
8229 COM /Specs/ Mx(0),Ions,Zc(0),Noise(0)
8232 COM /Daily/ INTEGER Mu,Daily,Mx(0,3,2)[0,2],Daily_ok(0:2),Ff(0:1,2)[4]
8235 COM /Magnet/ Mcore(*),INTEGER L,Aside,Peak_int,Coarsemag(0:1),Magnet(0:24,2),Coarse
8238 COM /Filaments/ filament(*),fil(*),INTEGER Nfiles,F8
8241 DIM Flag(2)
8244 !
IF NOT Full_auto THEN OUTPUT X80;Clear$;
Subflag=0
OFF KNOB
IF Pcheck_amps=0 THEN Pcheck_amps=1 ! just in case 0 accidentally passed
Target=GROUND(Target,4) ! because of binary-decimal roundoff errors
FOR I=Auto*Full_auto TO 19
  ON KEY I LABEL "" CALL Clunk
  SUBLABEL
  NEXT I
ON KEY 9 LABEL " ESCAPE" GOTO Exit
ON KEY 3 LABEL " DOUBLE RATE" GOTO Double_rate
ON KEY 8 LABEL " VALUE RATE" GOTO Value_rate
!
Max_pressure=1.E-6 ; Maximum permissible source-pressure
IF Rate/.04 THEN Rate=.04 ; (otherwise, get waits of >30 seconds)
F=Filament(Filnumber)
Start=F
Pcheck_start=F
IF Target1 AND Target=Target1.25 THEN GOSUB Check_pressure
Preheat=(Filnumber)2
IF Target=Start OR Rate<1 THEN Mu=0
OUTPUT 0;Mn$(Mu,1)
!
Max_pressure=1.E-6 ! Maximum permissible source-pressure
W=10*(Rate10)/(Rate1000)/Rate ; step wait (seconds)
Target=INT(Target1000)/1000
IF Target=Start THEN No_press_check=1 ! don't check pressure if reducing current
Met_target=0
!
LOOP
F=F+Increment*SGN(Target-F)
IF NOT Flagchecked AND (F).2 THEN
  FOR P=0 TO 1
    FOR Fct=1 TO 2! Put 0.2 amps at least thru all fils for flag-check
      IF Filament(fct)=0 THEN OUTPUT 8 USING "AH,42":ff$(P,Fct),200
    NEXT Fct
  NEXT P
  CALL Flag(Flag(*),0) ; Check for valid filament contacts
  FOR P=0 TO 1
    FOR Fct=1 TO 2! rezero fils
      IF Filament(Fct)=0 THEN OUTPUT 8;ff$(P,Fct),0
    NEXT Fct
  NEXT P
  OUTPUT 8;Mn$(0,1)
Flagchecked=1
IF (Filnumber=1)*(Flag[1]=1)*(Flag[1]=3) OR (Filnumber=2)*(Flag[1]=1) THEN 8409
Subflag=1
PRINT USING "2,/X";"******** NO FILAMENT-CONTACT - CORRECT PROBLEM & TRY AGAIN ********"
PRINT USING "X,2/" ;(are center and/or side filament switches in the on position?)"
Clunk
WAIT 4
SUBEXIT
END IF
!
OUTPUT 8 USING "AH,42":ff$(Preheat,1+(Filnumber=2 OR Filnumber=4)),FNF(F)
Filament(Filnumber)=F
Met_target=ABS(F-Met_target)<(Increment-.0001)
EXIT IF Met_target
!
DISP C$:Magnet(L,1):Cns:TAB(14):fil$(Filnumber);"=":F:"WMP$";" -TARGET=":target:"WMP$, RATE=":Rate:"mA/SEC."
IF W>.8 THEN ! so can exit with k9 key immediately
FOR I=1 TO W/.2
  WAIT .2
  NEXT I
ELSE
  WAIT W
END IF

! check source pressure if current > amp every Pcheck_amps
IF DROUND(((F-Pcheck_start)/Pcheck_amps,4)=INT(DROUND(((F-Start)/Pcheck_amps,4)) AND F1 AND NOT No_press_check THEN
  !
  DO LOOP
  CALL Pressure(0,Mn$(*),1,Source_pressure)
  EXIT IF Source_pressure=Max_pressure
  Clunk
  PRINT "WAITING FOR SOURCE-PRESSURE TO DECREASE TO (<1E-07..."
  Wait(TIMEOUT,20,Filament(Nfils),0,Auto,full_auto)
  END LOOP
  END IF
END LOOP

IF Metarget THEN
  Filament(Filunumber)=Target
  OUTPUT 8 USING "1R,1Z";Ff$(Preheat,1+(Filunumber=2 OR Filunumber=4)),Ff$(Target)
END IF

! Exit:IF Target)Start OR Rate(1 THEN Daly_ok= (0)
! OUTPUT KBD-,Clear$;
SUBEXIT

No_press_check:No_press_check=1
BEEP 440,.1
PRINT "(no pressure-checks)"
ON KEY 4 LABEL "CHECK PRESSURE" GOSUB Check_pressure
RETURN

Check_pressure:No_press_check=0
IF F)Start THEN BEEP 440,.1
PRINT "(pressure-check every ";Pcheck_amps; "ampere)"
ON KEY 4 LABEL "NO PRESS-CHECK" GOSUB No_press_check
RETURN

Double_rate:Rate=Rate*2
Pcheck_start=F
BEEP 1000,.08
GOTO Takeup_current

Halve_rate:Rate=Rate/2
BEEP 220,.08
GOTO Takeup_current

!-----------------------------------------------------------------------------
SUBEND !                    -  -

Htest:SUB Htest ! Check that peak is small enough for Daly
OPTION BASE 1
COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,full_auto,Foc(*),I_t
COM /Specs/ Hx(0:1),Ions,Ze(0:1),Noise(0:1)
COM /Daly/ INTEGER Mn,Daly,Ms*(0:3,2),Daly_eh(0:24),F1$(0:1,2),E4)
COM /Magnet/ Moeg(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
COM /Filaments/ filament(*),Fil$(0),INTEGER Mn$Fils,F0
8607 OFF KNOB
8610 IF Daly=0 THEN
8613 IF Auto THEN SUBEXIT
8616 Clunk
8619 PRINT TAB(1,18):FNH(""DALY DISABLED")&" (press [CONTROL] 0 during BMC to enable)"
8622 SUBEXIT
8625 END IF
8628 IF Daly_ok(L) THEN Turn_on Daly
8631 !
8634 PRINTER IS CRT
8637 I_t=0=I_t
8640 I_t=2
8643 OUTPUT 8:MN$(0,I_t) ! Faraday cup, .2-sec integration
8646 FOR K=1 TO ? ! scan magnet from -3/8 to +3/8 isotope
8649 GOSUB Check_peak
8652 IF Mu>50 THEN Peak_too_large
8655 IF Mu>Mu_max OR K=1 THEN
8658 Mu_max=Mu
8661 Max_mag=Magnet_value
8664 END IF
8667 NEXT K
8670 !
8673 Turn_on Daly:PRINT USING ",X,K,Mx,K,;"(Magnet(L,1));" PEAK OK FOR DALY",FNH("DALY ON")
8676 IF NOT Auto THEN BEEP 1000, 1
8679 IF Daly_ok(L)=0 AND Mu_max>1 THEN ! check to see if Daly functioning OK
8682 OUTPUT 8 USING "4R,42":"OFJ",Max_mag
8685 Mu=1
8688 Enter_beam(K,Mu,L,2,1,0)
8691 IF Mu>Mu_max/2.5 THEN ! something wrong
8694 PRINT USING "5X,13X,K,4X,K;"; "DAILY-DETECTOR RESPONSE IS INADEQUATE","PLEASE CHECK THAT THE MULTIPLIER SUPPLY AND THREE SWITCHES ARE ON,"
8697 PRINT USING ",X,2,;"; "AND THAT THEIR KNOBS ARE AT THE MARKED POSITIONS."
8700 PRINT USING "X,K,2,;"; "MAKE SURE THAT THE BRANDENBURG IS ON,";"AND THEN PRESS THE BRANDENBURG RESET BUTTON."
8703 Superclunk
8706 IF Auto THEN
8709 WAIT 10
8712 Daly=0
8715 Mu=0
8718 ELSE
8721 PRINT USING "4X,K;";"-- Press CONTINUE to return to the BMC --"
8724 Mu=0
8727 PAUSE
8730 END IF
8733 GOTO 8778
8736 END IF
8739 END IF
8742 Mu=1
8745 Daly_ok(L)=1
8748 GOTO 8793
8751 !
8754 Check_peak=Magnet_value=INT(Magnet(L,2)>Peak_inter*(K-4)/9)
8757 OUTPUT 8 USING "4R,42":"OFJ",Magnet_value
8760 Enter_beam(K,Mu,L,2,1,0)
8763 DISP "CHECKING":Magnet(L,1);"INTENSITY FOR DALY",Magnet_value;K
8766 RETURN
8769 !
8772 Peak_too_large:PRINT TAB(1,18):Magnet(L,1);"PEAK TOO LARGE FOR DALY"
8775 !
8778 IF NOT Auto THEN CALL Clunk
8781 Daly_ok(L)=0 ! this peak must be checked again if Daly is requested
Correct:SUB Correct(FM$,Mu$,INTEGER Magnet,Mu,I_t,Coarsemag,Foc*)
Output 8 USING "(1R,1Z),8BYWJ%Magnetll,2),Mn$(riu,IJ)
0811 ! Change/restore magnet, focus, filament, & collector values
0812 Z$="123156?89"
0817 Output 8 USING "(1R,1Z),8BYWJ%Magnet,40FK",Coarsemag ! magnet values
0820 Output 8 USING "(1R,1Z),8BYWJ%Magnet,40FK",Coarsemag ! magnet values
0822 Output 8 USING "(1R,1Z),8BYWJ%Magnet,40FK",Coarsemag ! magnet values
0826 Output 8;m$$(Mu,I_t) ! collector-type & integration-time
0829 FOR I=1 TO 10 ! focus values
0832 Output 8 USING "(1R,1Z),8BYWJ%Magnet,40FK",Coarsemag ! magnet values
0833 NEXT I
0838 SUBEND ! -----------------------------------------------
0844 Pressure:SUB Pressure(INTEGER Mu,Mn$,I_t,REAL filament(1),OPTI03$H filament(2),INTEGR 1 Magnet,Mu,I_t,Coarsemag,Foc*)
0847 ! Query source-pressure and (optional) tube-pressure
0850 OFF KEY
0853 OFF KNOB
0856 ON KEY 9 LABEL " ESCAPE" GOTO 0919
0859 UM=Mn$$(Mu,1,1|03|
0862 PRINTER IS CRT
0865 PRINT "SOURCE PRESSURE =";
0866 Output 8;"$IOU"; "$IOU"
0871 FOR I=1 TO 10
0874 WAIT .2
0877 NEXT I
0880 Output 8;"$IOU";
0883 ENTER B:PO ! zero-value
0886 Output 8;"$IOU"; "$IOU"
0892 Source_press=GROUND(1.E-8*100*(P-2950)/1020,3)
0895 PRINT Source_press;
0899 IF P>P0 THEN
0901 PRINT TAB(40);"TUBE PRESSURE =";
0904 Output 8;"$IOU";
0907 GOSUB Raw_press
0910 Tube_pressure=GROUND(10*(7.185E-6*(P-PO)-8.969),3)
0913 PRINT Tube_pressure;
0916 END IF
0919 PRINT
0922 Output 8;Mn$$(Mu,I_t)
0925 SUBEXIT
0928 |
0931 Raw_press:FOR I=1 TO 5
0934 WAIT .2
0937 NEXT I
0940 Output 8;"$IOU"
0943 ENTER B:P
0946 IF P>P0 THEN P=PO
0949 RETURN
0952 SUBEND ! -----------------------------------------------
0955 !
0959 !
0961 Contact_test:SUB Contact_test(filament(1),line0,Escape,Sample_name(1),Run_name(1),Candump,INTEGER Estbar(1),Sample,Mn$*1$)
! locate barrel-positions with valid filament-contacts for each sample

OPTION BASE 1

DIM /Barrel/ INTEGER Barrel_position,Barrel_pos0,Mx_barrel,Min_barrel
DIM /Daily/ INTEGER Mn,Daily,Min$(0:3,2),(Daily_ok(0:24),Flag$(0:1,2))

PRINT "PRESS k9 TO CHECK FILAMENT-CONTACTS FOR ALL OF THE SAMPLES IN THE BARREL","(turns filaments off),"
PRINT "PRESS k9 TO ESCAPE (return to BMC)."
GOTO 9012

PRINT USING "18/,17H,K,8/";"PLEASE WAIT WHILE BARREL IS BEING RESET"
Resbar
E=81+18?*(Barpos-1)
Start=E*(Barnum=1)+70*(Barnum>1)
CRLL Br«E-St*(Est>Start),0,2,0>
FOR J=Est-St TO Est*60
C=J-E+Startn
CflLL8rl((J),-l,.10,0)
ISP "SAMPLEl "&H11$(Uml$(Barnum));"Barrel-position: ";J
CflLL Flag(Flag(*),2,1)
Contact(P)«Flag(l)
!
!

FOR F=l TO 2
IF Flag(F) THEN
C=Flag(F)=3,t.5*(Flag(F)=l) ! box-pattern is solid for triple,
AREA COLOR C,C,C ! gray for single-fil ! gray for single,
MODE Barnum+.35*(F-2),l-Est/2 ! blank for sides-only
!
! draw rectangle whose fill-pattern indicates which filaments are
! contacted (to left of barrel# for sample, right for preheat).
RECTANGLE .35,1,FILL,EDGE
END IF
NEXT J
DATA 0,0,0,0,0,0,0
RESTORE 9162
READ S1,S1,T1,T1,S2,T2,
FOR P=1 TO 200
C=Contact(P)
S=S+(C=1 OR (C=3)) ! #successive single-fil contacts
R=R+(C=3) ! triple
IF (S>10 OR R>10) THEN ! count # of successive nocontacts
NEXT J
NEXT P
END IF
END IF
EXIT_testsZero_fils(filament*)
OFF KEY
CALL Br(3080,-1,1,0)
FOR Barpos=1 TO 16 ! find first sample in the barrel to rotate to when done
IF Estbar(FNBarnum(Barpos)),1> THEN 9285
NEXT Barpos
Sample=1 ! default
Sample=FBarnum(Barpos)
OUTPUT KB0:Clear$
Wfils=Estbar(Sample,2)
IF Wfils=0 THEN Wfils=1
IF Candump AND NOT Escape THEN
DUMP GRAPHICS
END IF
DISP CHR$(130)"Dumpling Contact-Test Graphics..."CHR$(128)
END IF
DISP FH$="DO YOU WANT TO ERASE THE PREVIOUSLY-DEFINED SAMPLE NAMES NOW? (Y/N)"$
INPUT Temp$
IF Temp$="Y" THEN
MAT Sample_name$= ("")
9321  MAT Run_name$ = ""
9324  END IF
9327  !
9330  OFF KEY
9333  LOOP
9336  LOOP
9339  INPUT "WHAT BARREL# DO YOU WANT ROTATED INTO RUNNING POSITION?", Sample
9342  EXIT IF Sample$ = 1 AND Sample$ = 16
9345  PRINT USING ":/10X,K,": FNHF$("**** BARREL-NUMBERS ONLY GO UP TO 16 **** ")
9348  Clunk
9351  END LOOP
9354  EXIT IF Establish(Sample, 1) ! did the contact-test find a sample here?
9357  Clunk
9360  GRAPHICS OFF
9363  PRINT TABXY(1, 10): FNHF$("NO SAMPLE AT BARREL# " & Run_number$: & "; PLEASE RE-ENTER BARREL#...")
9366  END LOOP
9369  !
9372  DISP "PRESS CONTINUE TO RESET BARREL, ROTATE BARREL#; Sample INTO POSITION, A ESCAPE TO BNC"
9375  PAUSE
9378  GRAPHICS OFF
9381  OUTPUT KBO; Clear$;
9384  OFF KEY
9387  OFF KBO
9390  Reset
9393  find(0, Sample, files, 0, Filament(*), Time0)
9396  SUBEXIT
9399  Escape = 0
9402  GOTO Exit_test
9405  SUBEND ! -------------------------------------------------------------
9408  !
9411  !
9414  Average = SUB Average(Do, Normal(*), Candump, Escape, INTEGER Run_number, Prtr(*))
9417  OPTION BASE 1
9420  COUNT Keyboard/ Cn$, C1$, C2$, Cu$, C3$, Clear$
9423  Maxblock = 80 ! maximum # of blocks of data that can be brought into memory
9426  DIM Date$(12), Sample_name$(50), Ratio$(80, '7), Block_ratio$(7), Aver_ratio$(7), Run$(20, '7), l1001, Input$(160), Delta$(7), 101, Sign$(7), 103, Hfrag$(10)
9429  REAL Aver$(80, '7), Ok_ratio$(80), Block_average(7)
9432  REAL Acc$(80, '7), Block_acc(7), Ok_acc$(80), Pk, Rejected$(80), Blocktime$(80), Ok_time$(80)
9435  INTEGER Sample, Ref, Block, Data_collector, N_isotopes$(80), Nblock(80), N
9438  Escape = 0
9441  Run = Run_number
9444  Run = Run
9447  First_average = 1
9450  OUTPUT KBO; Clear$;
9453  GRAPHICS OFF
9456  OFF KBO
9459  PRINT USING "16K,K,:FNHF$(" WEIGHTED AVERAGES OF RATIOS: ")
9462  IF Not Auto THEN
9465  IF First_average THEN
9468  PRINTER IS CRT
9471  PRINT USING "B/2K,K": " WHICH RUN#? 
9474  PRINT TABXY(1, 15): "enter 0 to escape, a negative run# (e.g. -" & Run_number$ & ") for CRT-display only,
9477  PRINT TABXY(1, 15): "press CONTINUE for current run (" & Run_number$: "); 
9480  PRINT TABXY(1, 15): "enter 100 to use file numbers rather than Run numbers"
9483  ELSE
9486  DISP " WHICH RUN#? 0 to escape, CON for run# " & Run_number$: ", neg. run# for CRT-display only;
9489  END IF
9492  INPUT Run
9495  Files = 0
9499 IF Rnum=100 THEN
9501 INPUT "ENTER FIRST, LAST FILE #S FOR DATA: ",First_record,Last_record
9504 Files=1
9507 ELSE
9510 IF Rnum=0 THEN
9513 Escape=1
9516 SUREEXIT
9519 END IF
9522 Run_number=ABS(Rnum)
9525 END IF
9528 END IF
9531 Print(Rnum) ! Provide hard-copy printout IF Print=1 (i.e. negative run#)
9534 PRINTER IS CRT
9537 GRAPHICS OFF
9540 IF Run_number\Lastrun OR NOT Got OR Files THEN
9543 IF NOT Files THEN
9546 Lastrun=Run_number
9549 ON ERROR GOTO Fai1dir
9552 ASSIGN SPathl TO "RESOIR:INTERNflL,1,l"
9555 ENTER SPathl,Run_number;Sample,Date$,Samplename$,First_record,Last_record
9558 PRINT USING "18/,K,/,8fi,20,10H,5fl,2D,10H,K,2/,K,3H,i"; "SAMPLE: "Samplename$, "BARREL ",Sample,"RUN ",Run_number,Date$,
9561 ISOIPOE RATIOS"
9561 END IF
9564 ON ERROR GOTO Faildat
9567 DISP "Getting data from disk. Please wait..."
9570 ASSIGN SPathl TO "RESULT:INTERNAL,4,4"
9573 FOR K=First_record TO Last_record*500*(First_record> Last_record)! get data from disk for all blocks of run
9576 I=K-500*(K>500)
9579 ENTER SPathl;I;Nfrag$,Block_ratio$(*),Block_average(*),Sigma$(*),Block_acc(*),Delta$(*),Pk,N,Data_collector.Block,Line,F
9582 curr
9585 Aver_ratio$=""
9588 Got=0! at least 1 block retrieved
9590 N_isotopes(Block)=N
9591 Blocktime(Block)=Time
9594 FOR J=l TO N-1! put data for all blocks of run into arrays
9597 Aver(Block,J)=Block_average(J)
9600 Acc(Block,J)=Block_acc(J)
9603 Ratio$(Block,J)=Block_ratio$(J)
9606 NEXT J
9609 IF Block=Maxblox THEN PRINT USING "2/,3<K>,2/Y**» CRN ONLY IfiKE UP TO ",Maxblox," BLOCKS ***"
9612 IF Block=Maxblox THEN 9618
9615 NEXT K
9618 OFF ERROR
9621 R=0
9624 MAT R$ = ""
9627 FOR I=l TO Block! find out which ratios were taken during the run and store in the R$ array
9630 FOR J=l TO N_isotopes(I)-1
9633 Oldratio=0
9636 FOR K=1 TO R
9639 IF Ratio$(I,J)=R$(K) THEN Oldratio=1
9642 NEXT K
9645 IF NOT Oldratio THEN
9648 R=R+R! Oldratio=0 if a new ratio
9651 R$(K)=Ratio$(I,J)
9654 IF R>20 THEN 9672 ! no more than 20 different ratios can be handled
9657 END IF
9660 NEXT J
9663 NEXT I
9666 END IF
9669 !
IF R=l AND NOT Auto THEN
AVER_Ratio$=Ra$(l)
PRINT
GOTO 9750
END IF
IF NOT Auto THEN
FOR I=1 TO R
PRINT USING "*.*":Ra$(I)
NEXT I
PRINT USING "2/":Rver_ratio$=Ra(d)
PRINT USING "1/":Rver_ratio$="H" THEN Rver_ratio$=Ra$(l)
DISP "WHICH RATIO (xxx/xxx)? (CONTINUE for "AVER_RATIO$"1,73")"
INPUT Rver_ratio$[1,7]
Rver_ratio$="FIRSt"(UPD(Rver_ratio$))
Rct=1 #??????
GOTO 9744
END IF
FOR Rct=1 TO N_isotopes(Block)-1
AVER_Ratio$=Ratio$(Block,Rct)
IF Normal(l) AND POS(Rver_ratio$",URL$(Normal(l))) THEN 10005 ! if an automatic run, don’t average the normalizing ratio
IF Ratio$(Block,N_isotopes(Block)-1)="SRH/SPK" AND (Rver_ratio$"SRH/SPK") AND (NOT POS(Rver_ratio$,"*")) THEN 10005
! don’t average un-normalized spiked-run ratios
PRINT "PRINTER IS Prtr(+auto)"
IF Full.auto THEN PRINT USING "/,8BD,";RPT$(H",";
PRINT USING "/,8BD,";FNUn$(IIBLACK,i),FNUn$(Rucr_ratioS),FNUn$(1'SIGMARMCflNr 1)
Found_ratios=0
FOR I=1 TO Block ! display all values for selected ratio
FOR H TO N.isotopes(Block)
IF Ratio$(J)=Rver_ratio$ THEN
PRINT TRB(l8);I;TRB(27);DRflUNO(Rver_d,J>,6);TRB(11);OROUNO<Rucr(I,J>,3)
Found_ratios=UFind_ratios
GOTO 9780
END IF
NEXT J
NEXT I
IF NOT Found_ratios THEN
PRINT USING "/,2/":FNH$<"THERE fiRE NO "Rver_ratio$&" RRTI05 IN THIS RON"
ON 1+fluto GOTO 9162,10005
END IF
IF NOT Auto THEN PRINT USING "2/"
N=0
Input$=""
IF NOT Auto THEN
DISP "RATIOS TO BE REJECTED? (e.g. 2,5,32) [ press "$FNUn$("CONF")& for none]"
INPUT Inputs
Parsed_input$,Rejected(*),Ninputs)
IF Print THEN PRINTER IS Prtr(2)
N_rej_blocks=Ninputs
IF N_rej_blocks=Found_ratios THEN
Clunk
DISP FNH$("YOU CAN’T REJECT THAT MANY BLOCKS.")
WHIT 3
GOTO 9807
END IF
IF N_rej_blocks THEN
PRINT "BLOCKS NOT INCLUDED: "
FOR I=1 TO N_rej_blocks
PRINT Rejected(I);
END FOR
9852     NEXT I
9855     PRINT
9858     END IF
9861     END IF
9864 FOR I=1 TO Block
9867     FOR K=1 TO N_rej_blocks
9870       IF I=Rejected(K) THEN 9906
9873       NEXT K
9876     FOR J=1 TO N_isotopes(I)-1
9879       IF Ratio$(I,J)=Aver_ratio$ THEN
9882         N=N+1
9885         Ok_ratio(N)=Aver(I,J)
9888         Ok_acc(N)=Acc(I,J)
9891         Ok_time(N)=Blocktime(I)
9894         Nblox(N)=I
9897     GOTO 9906
9900     END IF
9903     NEXT J
9906 NEXT I
9909 IF NOT fluto AND Print THEN
9912 FOR Y=0 TO Print! printout the ratios to be averaged
9915   PRINT USING "80A(2),/K,/,8A(2),10A(5),8A(2),/";RPT$("*,",#00),"SAMPLE: ",Sample,"RUN ",Run_number
9918   PRINT USING "21X,9A,4X,9A,4X,K,/,":FMUn$("BLOCK"),FMUn$(Aver_ratio$),FMUn$("SIGMAHAXE")
9921   FOR J=1 TO N
9924     PRINT TAB(22);Nblox(I);TAB(31);DROUNO(Ok_ratio(I),6);TAB(13);DROUNO(Ok_acc(I),3)
9927     NEXT I
9930     NEXT Y
9933     PRINT
9936 NEXT V
9939     END IF
9942 !
9945 IF N>1 THEN
9948     CALL Rggraf(Ok_ratio(*),N,Aver_ratio*,Ok_acc*,Ok_time*)
9951     Calcav(Ok_ratio(*),Ok_time*,Print,Aver_ratio*,Ok_acc*,Candump,Ptr*)
9954     ALPHA ON
9957     IF Print AND Candump THEN
9960     IF Not Auto THEN
9963       DISP "DO YOU WANT TO DUMP THESE GRAPHICS TO THE PRINTER?"
9966     OFF KEY
9969     ON KEY 0 LABEL "DUMP GRAPHICS" GOTO 9981
9972     ON KEY 4 LABEL "NO DUMP" GOTO 9993
9975     GOTO 9975
9978     END IF
9981     OFF KEY
9984     DISP FMUn$("DUMPING GRAPHICS...")
9987     DUMP GRAPHICS
9990     END IF
9993   First_average=0
9996   OFF KEY
9999 IF NOT Auto THEN 9462
10002 END IF
10005 NEXT Ret
10008 SUBEXIT
10011 !
10014 Faildir=PRINT USING "/,3(K),2,";Cl$("CAN'T READ DATA DIRECTORY FOR RUN ",Run_number," ");Cn$;
10017 ON 1:Auto GOTO 9462,10008
10020 Faildat=PRINT USING "/,5(K),2,";Cl$("CAN'T READ DATA FOR FILE ",K," RUN ",Run_number," ");Cn$;
10023 GOTO 9615
10026 SUBEND !---------------------------------------------
10029!
10032! 10035 Endsave! end of SAVE ed part
10044 DEF FMBar_digits(Barslot) ! Rotational position of barrel
10045 
10047 ! Barslot is the relative position in the barrel (1-16), not the 
10049 ! barrel-number.
10050 Y=-103.3*x185.2*Barslot ! .H03*Barslot A 2
10055 RETURN INT(YWFRRCT(Y>.5)
10059 FNEHO
10062 !
10065 !
10068 Pb_age=SUB Pb_age(R76)
10071 L8=1.55125E-4
10074 L5=9.0965E-4
10077 U=1.37.98
10080 IF R76>.0156 AND R76(1.9 THEN
10083 Trial_t=4500*(SDN(-1/2)*(R76/L5/(L8*U)))*(R>.7)
10086 Change=1
10089 REPEAT
10092 f=1.8*EXP(Trial_t*(L8-L5))/L5
10095 t=(LOG((1+EXP(Trial_t*L5)-1-f*EXP(Trial_t*L5)/1/(1+R76)-f))/L5
10098 Change=HOS(T-Trial_t)
10101 Trial_t=T
10104 UNTIL Change<.01
10107 END IF
10110 IF T THEN
10113 PRINT TRYXY(1,19):
10116 PRINT USING "K,D,M,K:";207/206 AGE FOR ";R76;" = ";ROUND(1,5)
10119 END IF
10122 SUBEND
10125 !
10128 !
10131 Error_message=SUB Error_message(INTEGER Prtr(*),Printer,Message$)
10134 ! If Printer=1, then display on CRT only
10137 ! If Printer=2, then printout only
10140 ! If Printer=3 then do on both.
10143 !
10146 FOR P=1 TO 1+(Printer=3)
10149 Printer IS Prtr(P+(Printer=2))
10152 PRINT USING "4/,k,4/":$"(";MessageS" \\
10155 NEXT P
10158 Printer IS CRT
10161 SUBEND
10164 !
10167 !
10170 Broop=SUB Broop ! make a "droopy-beep" sound
10173 FOR I=500 TO 100 STEP -50
10176 BEEP I,.02
10179 NEXT I
10182 SUBEND
10186 !
10188 !
10191 Drift_adjust=SUB Drift_adjust(Type,Runclass$,NuclideS(*),NormalO(*),NormD,Nu$,INTEGER Miso,Rf,Mu0,MagnetO(*),Coarsebin(*),Ref)
10194 ! Recalibrate appropriate elements for slight drift of magnet-settings
10197 OPTION BASE 1
10200 COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
10203 COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
10206 COM /Daily/ INTEGER Mu,Daily,MuS(0:3,2),(0:1),Daily_ok(0:24),Ff0(0:1,2)Y
10209 COM /Magnet/ Mcoef(*),INTEGER L,Axide,Peak_inter,Coarserang(0:1),Magnet(0:24,2),Coarserange
10212 COM /Filaments/ Filament(*),FilS(*),INTEGER Hflies,F8
10215 DIM /Data1/ Peak(*),Pk,Interfere(*),Normal(*),100,INTEGER Data_iso(*),Data_collector,Inv,Num_out,Next_run,Spike
10218 INTEGER Hu
10220 OUTPUT KB0:CHR$(255)&CHR$(75);
10224 PRINT USING "8/,K,/Y'This function will adjust the stored ELEMENT data for slight drifts in the best magnet-settings for the
element.'"
10227 PRINT USING "K,/,K":"To do this, you must: (1) have the high voltage set within 2 volts of the","default value for the ele
10230 PRINT USING "K,/,K":" (2) be on a peak of at least 10 nll. If both of the above are true, then press the appropriate softk
10233 OFF KEY
10236 ON KEY 0 LABEL " PROCEED" GOTO 10240
10239 ON KEY 9 LABEL " ESCAPE" GOTO 10360
10242 GOTO 10242
10245 I
10248 OFF KEY
10251 OUTPUT KB0:CHR$(255)&CHR$(75);
10254 CALL Hu(MoS(*),Ho,μ,M,I_t) ! query accelerating voltage
10257 IF P THEN SUBEXIT
10260 IF ABS(Hu(Hv-HHo))<2 THEN
10263 PRINT USING "5/,6H,K,K":FNH($(" ***** HIGH-VOLTAGE MUST BE WITHIN 2 VOLTS OF "HV/HV-HHV" ")
10266 Clunk
10269 END IF
10272 SUBEXIT
10275 END IF
10276 H=Ho1
10281 WAIT 1
10284 Enter_beat(0,Mv,L,I_t,0)
10287 IF (Mv<I) AND (Mv<10) OR (NOT (Mv>20) AND (Mv<20) THEN
10290 PRINT USING "3/,6X,r:FNH($(" **** MUST HAVE >10 nV (ONLY) OR >20 nV (CUP) *** ")
10293 Clunk
10296 WAIT 2
10299 SUBEXIT
10302 END IF
10303 L=I
10305 Center_peak(I,L)
10308 IF Subflag THEN
10311 PRINT USING "4/,6H,K":FNH($(" ***** NEED A CENTERABLE PEAK *****")
10314 Clunk
10317 END IF
10318 SUBEXIT
10319 END IF
10320 !
10321 IF filament(I)>4.7 THEN ! If >4.7 amps on center-fil, try to adjust
10322 ! Re-187 magnet-value for drift.
10323 Center_peak(0,0) ! center Re-187, then restore original-peak magnet-values
10324 I=0
10326 OUTPUT 8 USING "2(48,42),88":*DF1",Magnet(2),*DFK",CoarseMag(1)
10327 I Subflag THEN PRINT USING "K,/,K":"(Couldn't center the Re-187 peak)
10328 END IF
10329 !
10330 ON ERROR GOTO 10356
10332 ASSIGN @Pathl TO TYPE=INTERNATIONAL
10335 PRINT @Pathl,Type=Runclass$,Niso,Mcoef(*),Magnet(*),CoarseMag(*),Peak_inter,Aside,Rf,Nuclide(*),Normal(*),Inv,Interfere(*),
10338 Hz0=Hz
10341 OFF ERROR
10342 Whoop
10344 PRINT FNH($(" *** ADJUSTED MAGNET-VALUES FOR "&TRIM$(Runclass$)& " STORED ***")
10347 END IF
10350 SUBEXIT
10353 !
10356 PRINT USING "3/,10X,K";FNH$<"**** UNREADABLE ON DISK ****">
10359 OFF ERROR
10362 Clunk
10365 UNIT 2
10368 SUBEND
10369 !
10370 !
10372 Data:SUB Data(Decay,Na$,Ln,Mb$,$Share_bkgd,Dump_datagrd,Bad_pressure,INTEGER Miss,Block,Oy,Pb_4678,Run)! take a block of
10373 isotope-ratio data
10376 OPTION BASE 1
10379 COMM /General/ &Z,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_l
10385 COMM /Specs/ Mn(0:1),Ions,Ze(0:1),Noise(0:1)
10388 COMM /Magnet/ Mn,Daily,Mb$(0:3,2),Dally_ok(0:24),Ff$(0:1,2),Ce$
10391 COMM /Barrel/ INTEGER Barrel_position,Barrel_pos0,Max_barrel,Min_barrel
10394 COMM /Filaments/ Filament(*),Fill(*),INTEGER Nfilts,F0
10397 COMM /Data/ Peak_in(*),Pk,Interference(*), Normally(0:1),DATA_iso(*),Data_collector,Inv,Out_mout,Next_run,Spk
10400 COMM /Data2/ Acc(*),Lacc(*),Signs(*),Deltas(*),Avr(*),Last_aver(*),Last_ratio(*),Ratios(*)
10403 COMM /Data3/ Bkrd(*),Bkrd_var(*),Resdecay(*),INTEGER Sample,N_sets,Ref,Ref,Bkrd_rdgs(*),Bkrd_sign(*)
10406 COMM /Keyboard/ Slope,Int,Spknumber,Sum_in,Peak_t(0,40),Timensanc_corr(0),Peak_height(0,40),Normrat_slope,Normrat_inter,Norm_interve
10409 COMM /Interference/ Interference(0,3),Interf_nsecs,Interf_mon_cts(4,2),Interf_mon_time(4,2),INTEGER Long(*)
10412 INTEGER Bkrd_posn(12,2),Mag_pos(12),Mag_pos0(12),Ratio_iso(12),Integ_time(0),Wait_time(0),Ko,Pr,Specif_mon_pk(0),Dropout
10415 REAL Interf_corr_err(8),Peak_pk(8),Peak(8),Sum(5)
10418 DIM Mul(2146,1),Mul(2),101,203,184,71,Coi(1,1),Chi_square(40),Temp_aver(7)
10421!
10424 DATA 0,0,MILLIOLTS,CUP,DAILY,2,0,0
10427!
10430 Chisq=DATA 0,0,5.99,7.81,9.49,11.1,12.6,14.1,15.5,16.9,18.3,19.7,21.4,22.3,23.7,25.0,26.3,27.6,28.7,30.1,31.4,32.7,33.9,35.2,36.4,37.6,38.9,40.1,41.3,42.6
10433 DATA 43.8,44.9,46.2,47.4,48.6,49.8,51.2,52.2,53.4,54.6
10436!
10439 GRAPHICS OFF
10442 OFF KNOB
10445 N=N_prine ! don't allow value of N_prine as passed in Data3 to change
10448 OUTPUT 8 USING "44,42:";"SDF",FNIsomag(Mcoef(*),Data_iso(M)-.5) ! put magnet on a bkrd position
10451 FOR I=Auto+Full_auto=2 TO 19
10454 ON KEY I LABEL "" CALL Clunk
10457 NEXT I
10460 ON KEY O LABEL "" Omc GOTO Bmc_out
10463 ON KEY 1 LABEL "" NEXT RUN GOTO Next_run
10466 READ Pb4678_cycle,Daly_discr,UV,Mul(*),Daly_discr,Out_mout,Next_run,Chi_square(*)
10469 MAT Peak= Peak_in ! for subprogram is Peak(*) in main program
10472 MAT Peak_in= (0) ! only restore Peak_in array if complete data-block
10475 Interf_nsecs=N_sets ! seconds to spend on "short" interf-mon pks
10478 Spike_number=SPK!
10481 Number_peaks=N ! # isotopes requested by user (doesn't incl. interf-mon pks)
10484 Number_of_sets=N_sets
10487 MAT Peak_aka Peak
10490 MAT Interf_mon_cts= (0)
10493 MAT Interfere0= Interference determine which isotopes to monitor for isobaric interferences
10496 MAT Long= (0)
10499 MAT Interf_inter_err= (0)
10502 MAT Specif_mon_pk= (0)
10505 MAT Sigma$= (***)
10508 MAT Delta$= (***)
10511!
FOR I=1 TO N
10520 IF Interfere(I,2) = Data_iso(I) THEN 10535
10523 NEXT I
10526 InterfereO(H,1) = 0 ! ignore isobaric interferences if none for the
10529 InterfereO(H,2) = 0 ! specified isotopes.
10532 InterfereO(H,3) = 0
10535 NEXT M
10538 FOR I=1 TO N
10541 Ratio_isotope(I) = Data_iso(I) ! can include interf-non isotopes
10544 NEXT I
10547 FOR M=1 TO 4
10550 FOR I=1 TO N
10553 IF Interfere(I,1) = Ratio_isotope(I) THEN
10556 Long(M) = I ! order of monitor isotope in data-isotope list
10559 InterfereO(M,1) = 0 ! so don't monitor before/after peak tops
10562 Specif_non_pk(I) = 1 ! if a specifically-requested interference-monitor peak
10565 END IF
10568 NEXT I
10571 NEXT M
10574 OUTPUT KBD:CLEAR$;
10577 ON KEY 0 LABEL " ESCAPE" GOTO Exit_data
10580 IF NOT Auto THEN S1$="(auto)"
10583 IF Auto THEN S1$="(AUTO)"
10586 IF Nfiles>2 THEN S1$="SIDE-TIL=\"\";FILAMENT(2)\"\" AMPS"
10589 Tpress=TIMEOBE
10592 PRINTER IS CRT
10595 CALL Pressure(Mu,Mm$(*),1,Source_pressure,Tube_pressure)
10598 IMAGE 3,,08R,,,.08R,.08X,08R,3/
10601 IF (Source_pressure)E-6 OR (Tube_pressure)E-7 THEN
10604 FOR Y=2 TO 1 STE 1
10607 PRINTER IS Prtr(Y)
10610 IF Source_pressure)E-6 THEN
10613 PRINT USING 10598;RPT$(*,80),RPT$(*,80),FNB1$("SOURCE PRESSURE IS TOO HIGH \"\";Source_pressure\"\")",RPT$(*,80)
10616 END IF
10619 IF Tube_pressure)E-7 THEN
10622 PRINT USING 10598;RPT$(*,80),RPT$(*,80),FNB1$("TUBE-PRESSURE IS TOO HIGH \"\";Tube_pressure\"\")",RPT$(*,80),R
10625 END IF
10628 NEXT Y
10631 Superduperclunk
10634 IF Auto THEN
10637 IF (TIMEOBE-Tpress)/60)60 THEN ! wait no more than 1 hour for pressure to improve (auto-running only)
10640 Bad_pressure=1
10643 GOTO Exit_data
10646 END IF
10649 PRINT "WAITING FOR ADEQUATE SOURCE AND TUBE PRESSURE..."
10652 Wait:TIMEOBE,60,Filament(Nfiles),0,Auto,Full_auto
10655 GOTO 10595
10658 ELSE
10661 GOTO Exit_data
10664 END IF
10667 OUTPUT KBD:CLEAR$;
10670 END IF
10673 Bad_pressure=0
10676 FOR Y=1 TO 2
10679 PRINTER IS Prtr(Y)
10682 IF Y=2 THEN PRINT USING "$X,K":CHR$(27)*8180" ! narrow line-spacing
10685 IF Y=2 THEN PRINT "SOURCE PRESSURE =";Source_pressure:LAB(40);"TUBE PRESSURE =";Tube_pressure
10688  **IMAGE** "BARREL", "sh", "14x", "BLOCK", "0.0", "2x", "0.8", "4x", "10x", "2x", "RULE", "0.0", ",/", "CENT. FIL. = "0.30", "AMP", "4x", "20x", "2x", "DD", "SETS ", "/", "SAMPLE : "59A/.
10691  **PRINT USING** 10688;Sample, 1*Block, S1*$CHR$(120) & "%Y", "BLOCK! %DO,2H,8fi,4H,10fl,8H,"RUffft",DO," V/CENT. FIL. =",D.30," flMPS",ffi,20fl,18X,DD l " SEIS ",/,"SflNPLE: ",50fy
10701  ! PRIHI USING 10688;SaflpIe,lt8iock,Sl$,CHR$(128tMV)a i'^lM$(l+Dv)a''^"aCHR$(128),Run/ilaNent(l),S$,Nsets,Ha$
10703  ! if Conzer=1 then take backgrounds only for the least-intense peak
10705  ! - applies if cup data only, max-ratio<20, good pressure, not Pb4670.
10703  IF Peak(N)>0 THEN
10706  Sperclunk
10709  **PRINT USING** "4X,3X,3X,";"************** NULL(2)=0 ERROR IN Data ******NULL******";"*************** PLEASE NOTIFY KEN **********
10712  ! Peak(N)=1
10715  END IF
10719  IF Peak(I)/Peak(N)<20 AND (DY=0) AND (Pb4670=0) AND (Source_pressure<4.0) AND (Tube_pressure<6.0) THEN Conzer=1
10721  FOR I=1 TO N
10724  IF Ratio_isotope(I)=Ref THEN R=I !R is order or ref-isot. in peak-switching
10727  NEXT I
10730  Setup_1: ! if Pb 204-206-207-208 block, take 6/4 data before & after 6/7/8 data
10733  IF Pb4670 THEN
10736  Ratio_isotope(I)=206
10739  Ratio_isotope(2)=204
10742  Hsets=S 7*Number_of_sets ! but take only 70% of specified sets each time
10745  IF Hsets<7 THEN Hsets=7
10748  M=2 ! since only 206 & 204
10751  Number_peaks=2
10754  MAT Peak_a= (0)
10757  Peak_a(1)=Peak(R) ! 206
10760  FOR I=1 TO 4
10763  IF Data_iso(I)<204 THEN Peak_a(2)=Peak(I)
10766  NEXT I
10769  Refpk=1
10772  ELSE
10775  Refpk=R
10778  END IF
10781  Setup_2:CALL Peaktime(Peak_a(*),Noise(*),Isos,Integr_time(*),Wait_time(*),N,DY,(Refpk),Ratio_isotope(*))
10784  GCLERR
10787  FOR J=1 TO 2
10790  PRINTER IS Prtr(J)
10793  IF J=2 THEN PRINT USING ",K","CHR$(27)" & "BID"
10796  PRINT USING "9A,12X,";"ISOTOPE="
10799  FOR I=1 TO N
10802  PRINT USING "3D,3X,";Ratio_isotope(I)
10805  IF Specif_non_pk(I)=1 THEN ! if isotope is a specifically requested isobaric-interference monitor, use wait-t & integrat
10808  "ion-t of 3 seconds
10811  "Wait_time(I)=2
10814  "Integr_time(I)=3
10817  END IF
10817  NEXT I
10820  PRINT USING ",",18X,"=";"INTEGRATION-TIMES:"
10823  FOR I=1 TO N
10826  PRINT USING "3D,3X,";Integr_time(I)
10829  NEXT I
10832  PRINT USING ",",11X,"=";"WAIT-TIMES:"
10835  FOR I=1 TO N
10838  PRINT USING "3D,3X,";Wait_time(I)
10841  NEXT I
10844  PRINT
10847  NEXT J
10850  FOR R=1 TO N
10853  IF Ratio_isotope(R)=Ref THEN 10859 !R is order or ref-isot. in peak-switching
10856  NEXT R
10859 PRINTER IS Prtr(2);WIDTH (132)
10862 PRINT USING "\#K,\#:";CHR$(27)a"ak2$! snail print
10865 IF NOT Pb_4678 OR NOT Pb4678_cycle THEN
10868 PRINT "FOCUS":TAB(19); ! print focus-settings
10871 FOR I=1 TO 6+Nfils
10874 PRINT USING "3(K),3H,";I,"-",FocCD
10877 NEXT I
10880 PRINT
10883 END IF
10886 Nn=Nn+*(Pb_4678*Pb4678_cycle=0) ! Expand to include possible interf-non pks
10889 !
10892 FOR K=1 TO Nn ! determine magnet-settings for peaktops & bkrd
10895 IF K*N THEN Ratio_isotope(K)=Magnet(J,N-1) ! add interference-monitor isotope to Ratio-Isotope list
10898 FOR J=1 TO Niso
10901 IF Ratio_isotope(K)>=Magnet(J,1) THEN
10904 Mag_pos(K)=Magnet(J,2) ! peaktops
10907 |
10910 ! bkrd at 1/2 amu above & below peaktops
10913 Bkrd_posn(K,2)=Mag_pos(K)+(FNIsmag(coef(*)-Ratio_isotope(K)*.5)-FNIsmag(coef(*),<Ratio_isotope(K)>)! mag-setting f or ABOVE bkrd
10916 Bkrd_posn(K,1)=Mag_pos(K)+(FNIsmag(coef(*),Ratio_isotope(K)-.5)-FNIsmag(coef(*),<Ratio_isotope(K)>)! mag-setting for BELOW bkrd
10919 END IF
10922 NEXT J
10925 NEXT K
10928 |
10931 Mn=Mu
10934 OUTPUT B;"M\#(Mu,1) ! 1-second integration
10937 DISP
10940 PRINT USING "\#K,\#:";"MAGNET: ("WURL$(Coarse)" ) " ! print magnet-settings
10943 FOR I=1 TO Nn
10946 IF Ratio_isotope(I)>=0 THEN PRINT USING "3(K),3H,";Ratio_isotope(I),"-",Mag_pos(I)
10949 NEXT I
10952 PRINT USING "/"
10955 K=0
10958 PRINTER IS Prtr(2);WIDTH (132)
10961 FOR I=1 TO Nn ! Print headings for backgrounds
10964 FOR J=Number_peaks+1 TO 1-1
10967 IF (Ratio_isotope(J)>0) THEN PRINT USING "3(K),3H,";Ratio_isotope(J),"-",Mag_pos(J)
10970 NEXT J
10973 K=K+1
10976 IF Ratio_isotope(I) THEN PRINT TAB(10*K-9):Ratio_isotope(I):
10979 NEXT I
10982 PRINT " BACKGROUNDS: (cts per sec./std. dev.)" |
10985 Bkrd=1 take before-peaktop backgrounds
10988 IF NOT Share_bkrgs THEN ! If not sharing the last block's backgrounds
10991 CALL Backgrounds(Cunzer,Number_peaks,Peak(*),R,Pb_4678_cycle,Nn,0,N,Pb_4678,Integr_time(*),Bkrd_posn(*),Ratio_isotope(*),Coarsemag(1))
10994 IF Bmc_out OR Next_run OR Subflag! THEN Exit.data
10997 ELSE
11000 FOR I=1 TO Nn |
11003 FOR J=1 TO 2
11006 Bkrd(J,1)=Bkrd(J,2)
11009 Bkrd_rdg(J,1)=Bkrd_rdg(J,2)
11012 NEXT J
11015 NEXT I
11018 END IF
11021 OUTPUT B;"$ICL"
11024 ENTER &;In
11027 IF Pb_4678=0 OR Pb4678_cycle=0 THEN
11030 \text{MAT Mag_pos0= Mag\_pos}
11033 \text{CALL Interfere(O,In,Bkrd{s}(\*),Mag\_pos0(\*),\langle Number\_peaks\rangle,Subflag,Bnc\_out,Next\_run)}
11036 \text{ON Subflag GOTO 11039,Bkrd,Bkrd,Exit\_data}
11039 \text{IF Bnc\_out OR Next\_run THEN Exit\_data}
11042 \text{END IF}
11045 \text{FOR M=1 TO 4 ! Put monitor isotope with data-isotope list if interference correction will be > .10% and monitor peak is >.5nU/.05nU (Cup/Daly)}
11048 \text{P=Interf\_man\_cts(M,1)/Mix(Mu)+ nU interference-monitor peak}
11051 \text{IF Interfere0(M,1) AND Long(M)=0 AND (Mu\(\star\)\(\star\)) OR (Mu\(\star\))>.05) THEN}
11054 \text{FOR I=1 TO N}
11057 \text{IF Interfere0(M,2)=Ratio\_isotope(I) AND KB THEN}
11060 \text{Checked=0}
11063 \text{FOR J=1 TO M-1! only monitor an interference peak once per set}
11066 \text{IF Interfere(I,J)=Interfere(M,1) AND Long(J) THEN Checked=1}
11069 \text{NEXT J}
11072 \text{Percent\_corr=100*(Interf\_man\_cts(M,1)/Mix(Mu)+Interfere0(M,3)/\text{Peak}(I)) % correction from isobaric interference}
11075 \text{IF Percent\_corr>.10 AND NOT Checked THEN ! if correction >.10 then use "long" interference monitor}
11078!
11081 \text{N=N+1! increment the number of isotopes in the set}
11084 \text{Long(M)=N}
11087 \text{FOR P=1 TO 2! allocate interf-Mon bkrds to correct main-pk order}
11090 \text{Bkrd(P,M)=Bkrd(P,Number\_peaks+M)}
11093 \text{Bkrd\_rdgs(P,M)=Bkrd\_rdgs(P,Number\_peaks+M)}
11096 \text{NEXT P}
11099 \text{Peak(N)=Peak(N-1)}
11102 \text{Ratio\_isotope(N)=Interfere(M,1)}
11105 \text{Wait\_time(N)>2}
11108 \text{Integr\_time(N)=3*(Percent\_corr).2+2*(Percent\_corr).4! integrate for 3 seconds if correction>.2X, 4 if between .2X and .4X, 6 if>.4X}
11111 \text{Interfere0(M,1)=0}
11114 \text{Interfere0(M,2)=0}
11117 \text{FOR K=1 TO Miso}
11120 \text{IF Magnet(K,1)=Ratio\_isotope(N) THEN Mag\_pos(N)=Magnet(k,2)}
11123 \text{NEXT K}
11126 \text{END IF}
11129 \text{END IF}
11132 \text{NEXT I}
11135 \text{END IF}
11138 \text{NEXT M}
11141 \text{Short\_interf=0}
11144 \text{FOR M=1 TO 4 ! count # of "short"-monitored interf-monitor pks}
11147 \text{IF Interfere0(M,1) THEN Short\_interf=1*Short\_interf}
11150 \text{NEXT M}
11153 \text{!}
11156 \text{PKs=PRINTER IS CRT}
11159 \text{OUTPUT KBorClear;};
11162 \text{IF NOT Pb\_4678 OR Pb\_4678\_cycle=1 THEN}
11165 \text{Pmax=1.2*Peak(1)}
11168 \text{ELSE}
11171 \text{Pmax=1.2*Peak_0(1)}
11174 \text{END IF}
11177 \text{Raxes=0,100,36,86,0,(20*Short\_interf*Msets*(SUM(Integr\_time)+SUM(Wait\_time))/60,0,0,\text{"MINUTES","\text{\"nU BEAM"},0,-1,0)}
11180 \text{PRINT USING 10680;Sample,1,0,Block,15,"**"Mix\(\star\)\(\star\)\"**",Run,Filament(I),S\text{e},Msets,Has}
11183 \text{CALL Peaks(R,Peak\_t(\*),N,Peak\_height(\*),Pmax,Temp\_aver(\*),N,CoarseMag(I),Mag\_pos(\*),Integr\_time(\*),Wait\_time(\*),Ratio\_isotope(\*),Dropouts)}
11186 \text{IF Bnc\_out OR Next\_run THEN Exit\_data}
11189 \text{ON Subflag GOTO Bkrd,Over\_10volts,Exit\_data,Exit\_data}
11192!
11195 \text{Bkrd#1 take after-peaktop backgrounds}
11198 \text{PRINTER IS Prtr(2);WIDTH (132)
CALL Backgrounds(Conzer,Number_peaks,Pk(*),R,Pb_1678_cycle,N,2,H,Pb_1678,Integr_time(*),Bkrd_posn(*),Ratio_isotope(*),Coars
eng(1))

PRINT USING "$,K";CHR$(2>aikGS"! nornal print

IF Bnc_out OR Next_run OR Subflag1 THEN Exit_data

FOR M=1 TO 4 ! Shift interf-non bkrds for spec.-non pkks to correct order

IF Long(M) THEN

IF NOT Specif_mon_pk(Long(M)) THEN ! if a not specifically-requested interference-monitor peak

FOR P=1 TO 4

Bkrd(P,Long(M))=Bkrd(P,Number_peaks+M)

Bkrd_rdgs(P,Long(M))=Bkrd_rdgs(P,Number_peaks*H)

END IF

NEXT P

END IF

END IF

NEXT M

PRINT IS CRT

flfil Interf_corr_err= (0)

FOR J=1 TO

FOR J=1 TO

IF (Interfer(J,J)=Interfer(J,J)) AND Long(J) THEN Long(M)=Long(J)

NEXT J

NEXT M

IF NOT Pb_1678*Pb_1678_cycle THEN

CALL Interference(2,In,Bkrd(*),Mag_posO(*),Number_peaks,Subflag,Bnc_out,Next_run)

IF Subflag=2 OR Subflag=3 THEN Bbkrd

IF Bnc_out OR Next_run OR Subflag=4 THEN Exit_data

END IF

DISP

Sum_bkrd_rdgs=0

PRINT IS Prtr(2)

FOR I=1 TO H ! calculate average backgrounds

Bkgd=0

Bkrd_var(I)=0

FOR H TO 1

SUM_bkrd_rdgs=SUM_bkrd_rdgs+Bkrd_rdgs(K,I) ! total bkrd rdgs used for Ith peak

Bkgd=Bkgd+Bkrd(K,I)/H ! average background in counts/second

Bkrd_var(I)=Bkrd_var(I)+Noise(M)/2/(H*Bkrd_rdgs(K,I)) ! Average background variance of the mean, in (mU/sec)^2, for Ith peak

NEXT K

NEXT M

FOR J=1 TO Nsets

Peak_height(I,J)=Peak_height(I,J)-Bkgd ! subtract backgrounds

NEXT J

NEXT I

MAT Timeconst_corr= (0)

FOR M=1 TO 4 ! correct for isobaric interferences

END IF

IF Long(M)=0 THEN

IF Interfer(0,M)=0 THEN 11453 ! ' short' isobaric interfer. monitor (linear interp)

Coef(2,1)=(Interf_mon_cst(M,2)-Interf_mon_cst(M,1))/(Interf_mon_time(M,2)-Interf_mon_time(M,1)) ! slope of interf-monitor peak decay

Coef(1,1)=Interf_mon_cst(M,1)-Coef(2,1)*Interf_mon_time(M,1) ! intercept of interf-monitor peak decay

Coef(3,1)=0

Coef(4,1)=0

ELSE

CALL Cubic(Coef(*),Peak_t(*),Peak_height(*),Nsets,Long(M))! "Long" isobaric-interference monitor (cubic interp)

END IF

FOR I=1 TO H

IF Interfer(M,2)=Ratio_isotope(I) THEN

Nv_monitor_pk=0

1136
FOR J=1 TO Nsets
    T=Peak_t(I,J)
    Mon_pk=Coef(l,l)*T*(Coef(2,l)+T*Coef(3,l))<predicted monitor-pk intensity
    Interf.pk=Mon_pk*Interfere(l,3)<predicted interfering-pk intensity
    Peak_height(I,J)=Peak_height(I,J)-Interf.pk correct for isobaric interference
    Pcor=100*Interf.pk/Peak_height(I,J) average Z correction to Peak_height(I,J)
    IF Peak_height(I,J)<1e-9 THEN PRINT "error in peak height";
END IF
NEXT J
flu_interf_corr=flu_interf_corr/(Hpcor+(Hpcor=0) OR Peak_height(I,J)<Hpcor*100*100)
Npcor=UHpcor
Pcor=Pcor
END IF
NEXT I
Hpcor=0
 FOR J=1 TO Nsets
     CALL Suns=S(E:L,11,(Peak_height(I,J)),Peak_t(I,J),0)
     CALL Slope(Suns,Decay,Int,SO,S9,Hsets)
     Pk=Suns(l)/Hsets/1x(1u)
     Decay=Decay*Hsets*6.65/Suns(l)
     WHOOP
     PRINT DROUND(Pk,3);"MU":Ref;
     FOR I=1 TO N
     IF I<3 THEN Calc_nextratio
     I=M
     dbn't calculate a ratio if peak I is the ref-pk, or if an unspiked, fractionation-normalizable run until the normalizing rat
io has been calculated.

11546 IF (NOT Spike)*(Ratio_count*(Ratio_isotope(IO)=IO_normalizer) OR (Ratio_count))*(Ratio_isotope(IO) = Normal(IO)) THEN Call_ratioscalc

11549 PRINT 15 Prn(2)

11552 CALL Ratioscalc(IO,Calc_inverted,R,Mass_discr,Ratio_count,Pb4678_cycle,Normal(IO),N_sets_used,Decay_sets,Pb_4678,Ref,Inv,M,Ratio_isotope(IO))

11555 !

11558 Final:1n=ABS(Rater(IO))

11561 Temp(N_sets_used-I)/N_sets_used*Noise(Mu)^2*(INV^2)/INTEGRATE(time(IO)+1/INTEGRATE(R))/PK^2*(INTEGRATE(time(IO)+1)/PK)^2

11564 Ts=100+SQR(Temp)

11567 IF Concent=0 THEN Bkrd_uncert=100+SQR(Bkrd_var(IO)+2*INV)/PK

11570 IF Concent=1 THEN Bkrd_uncert=100+SQR(Bkrd_var(N)+2*(INV-1)/PK)

11573 ! Bkrd_uncert * Error in ratio due to background uncertainty

11576 ! Err=SQR(Interf_corr_err(IO)^2+Interf_corr_err(IO)^2) ! Error due to isobaric-interference corrections

11579 Rcc(IO)=SQR(Mean(Sig,2)+Rcc(I))/Recip_yar4/Rcc(l) A 2+l/First61 sigfiean A 2

11580 Rfluer(l)=(Rfluer(l)+Rcc(I))/Recip_yar4/Rcc(l) A 2+l/First61 flier(l),Recip_yar

11583 Sig=1/SQR(1/(Sig^2+First61 sig^2)/2)

11586 Sig=100+SQR(Bkrd_var(IO)+2*(INV-1)/PK)

11589 END IF

11612 !

11615 FOR K=1 TO Ln-1 ! Calculate percent change in ratios, put in parentheses if within theoretical

11618 IF Block RHQ <Last_ratio$<K*,RHQ Last_auer(K) THEH Call Delta(Delta$(I),flaler(I),Last_auer(K),Rcc(K))

11621 NEXT K

11624 Sigmas(I)=TRIM(Sig(K))

11627 Sigmas(I)=TRIM(Sig(K))

11630 Within_theor=Sig/(Sig^2+First61 sig^2)/2

11633 IF Within_theor THEN Sigmas(I)=100+SQR(I)

11636 GRAPHICS OFF

11639 IF NOT Printed THEN OUTPUT XBO;Clear$;

11642 Printed=1

11645 FOR Y=1 TO 2-Pb_4678*(Pb_4678_cycle=l)

11648 PRINT 15 Prn(Y);WIDTH (132)

11651 T=5*Y(2)

11654 PRINT "AVERAGE:"TAB(16);"AVERAGE:"TAB(29);"SIGMA:"TAB(40);"SIGMA:"TAB(50);"SIGMA:"TAB(60);"SIGMA:"TAB(70)

11657 IF Y=2 THEN PRINT USING "#%,**=";CHR$(2798)+"160" ! Normal line-spacing

11660 PRINT "NRatio$(I);Rcc(I)+FInvert$(Ratio$(I));(Ratio$(I)+Rcc(I))";TAB(40)+"Theor.:";TAB(50);"Mean:";TAB(60)+"Mean:";TAB(70)+"DELT$(I)"

11663 IF Y=1 AND N<K THEN PRINT


11669 PRINT TAB(49):GROUND(Rcc(IO),2):TAB(59):GROUND(IO/Iner(K),6):TAB(70):Delta$(IO)

11672 IF Y=2 AND I=1 THEN PRINT USING "#%,**=";CHR$(2798)+"180"

11675 PRINT RTPS("-",80)

11670 NEXT Y

11681 Ratio_count=Ratio_count+1

11684 IF (Ratio_count=l) AND Normal(IO) AND Spike=0 THEN

11687 Normal_slope=Slope ! Linear-regression slope of normalizing ratio

11690 Normal_inter=Int ! " " intercept " "

11693 Normal_inverted=Calc_inverted

11696 Normal_acc=Rcc(IO) ! uncertainty of normalizing ratio

11699 I=0

11702 GOTO Call_ratioscalc

11705 END IF
11708 PRINT \n11709 PRINTER IS CRT \n11711 Calc.nextation=NEXT IO
11714 !
11717 IF Pb_9478 AND Pb_9678_cycle THEN
11720 Center_peak1,k(k) re-center the 206 peak & recalibrate mag-settings
11723 IF Subflag=1 THEN Bmc_out
11726 SELECT Pb_9678_cycle
11729 CASE 1 ! after 1st 206/207 sets
11732 PRINT USING 10688;Sample,1*Block,S1$,**"$Mul$(1+Mk)"**",Run,Fileulant(l),S8,Nsets,Mk$
11735 MAT Peak_a=0;
11738 J=0
11741 FOR I=1 TO 4
11744 IF Data_iso(I)<>204 THEN
11747 J=J+1
11750 Ratio_isootope(J)=Data_iso(I)
11753 Peak_a(J)=Peak(I)
11756 END IF
11759 NEXT I
11762 N=3
11765 Number_peaks=3
11768 Msets=Number of sets
11771 First_64=Aver(1)
11774 First64.sigmean=Act(1)
11777 first64.sig=Sig
11780 first64.taig=Tsig
11783 First64.sets=N_sets_used
11786 FOR I=1 TO 3 find order of 206 & 207 peaks
11789 IF Ratio_isootope(I)=206 THEN Refpk=I
11792 IF Ratio_isootope(I)=207 THEN Peak_207=I
11795 NEXT I
11798 R=Refpk
11801 FOR K=1 TO 2 ! share 206 bkgrds from 6-4 to 6-7-8 if 6/7 ratio not too large
11804 Bkrdx(K,Refpk)=Bkrdx(2*K,1)
11807 Bkrd_rdgs(K,Refpk)=Bkrd_rdgs(2*K,1)
11810 Bkrd_sigmak(K,Refpk)=Bkrd_sigmak(2*K,1)
11813 NEXT K
11816 PRINT IS Prtr(2)
11819 PRINT RPI*$("-",00)
11822 PRINT IS CRT
11825 GOTO Setup_2
11828!
11831 CASE 2! after 206-207-208 sets
11834 FOR K=1 TO 2
11837 Bkrdx(K,1)=Bkrdx(2*K,Refpk)
11840 Bkrd_rdgs(K,1)=Bkrd_rdgs(2*K,Refpk)
11843 Bkrd_sigmak(K,1)=Bkrd_sigmak(2*K,Refpk)
11846 NEXT K
11849 IF Dump_datagraf THEN DUMP GRAPHICS ! only dump 206-207-208 graphics
11852 GOTO Setup_1
11855 END SELECT
11858 END IF
11861
11864 IF Dump_datagraf AND NOT Pb_9478 THEN DUMP GRAPHICS
11867 MAT Last_aver=Aver
11870 MAT Lacc= Acc
11873 MAT Last_ratio= Ratio$
11876 L=R$
11879 Datacollector=Mk$
11882 Block=1*Block
11885 Exit_dataLn=N_prime*Pb_9478+N*(Pb_9678=0) ! include any added interf.-mon
Nset5=Number_of_sets  ! pks if not a Pb-4678 block
Nset4=K-1

SUBEXIT

Over_10volts:IF K<4 THEN Exitjata ! Stop peak-switching if beam > 10V
Nsets'=K-1
GOTO flbkrd

Bnc_out=1

IF Auto AND SUM(Peak_in)=0 AND SUM(Peak)>0 THEN MAT Peak_in= Peak
! to solve early-exit zeroing of Peak array in auto-running
GOTO Exit_data

SUBEND ! -----------------------------------------------------------

SUB Suns(Sums(*),G,V,X,INTEGER N)

If (N<3) OR (G<0) THEN SUBEXIT
Suns(1)=Sums(1)+G*V  ! sum of V
Suns(2)=Suns(2)+G*V*V ! sum of squares of Y
Suns(3)=Suns(3)+G*X  ! sum of X
Suns(4)=Suns(4)+G*X*X ! sum of squares of H
Suns(5)=Suns(5)+G*X*Y ! cross-product sums
N=N-(G<0>

SUBEND ! -----------------------------------------------------------

Slope:SUB Slope(Sums(*),Slope,Intercept,Std_dev_pts,Std_dev_slope,INTEGER H)
Slope=(N*Suns(5)-Sums(3)*Sums(1))/(N*Suns(1)-Suns(3)*2)
Intercept=(Suns(1)-Slope*Sums(3)/N
S=Suns(2)-Intercept*Sums(1)-Slope*Sums(5)/(N-2)
Std_dev_pts=SQR(S*(S>0))  ! std deviation of pts about line
Std_dev_slope=SQR(S/S/L)

SUBEND ! -----------------------------------------------------------

Runvariables:SUB Runvariables(RunvarO(*),Run_name(*),Runtype(*),Sample_name(*),INTEGER Run_order(*),Run_iso(*),Auto)

OPTION BRSE 1

COM /Keyboard/ Cn$,Ci$,Cb$,Cu$,0$,Clear$
COM /Auto_form/ Lastresponse(-4:27)/E50,Resp(-4:27),Last_runtypex(0),Normal(2),Response(-4:27)/E50,Promptx(-4:27)/E50
COM /Auto_form/ Spike,INTEGER Isotope(0),Niso,Rf,Magnet(0:21,2)
DIM Stdvar(32,27),Std_name*(32),Stdtype*(32),Elemt(20),Run*(4:27),S*(9:160),S_name(32):E50,Mgas(32)

INTEGER Std_iso(32,8)
OUTPUT K80;CHR$(255)&CHR$(75);
Got_runvars=0

Got_stdvars=0

flgain:OFF KEY
OFF KBO
OFF KNOB
A=KNOB
GRAPHICS OFF

Last_runtypex=""
12065 Prompt$(1)="DEFINE A NEW SET OF "$RUN$" VARIABLES"
12068 Prompt$(2)="EDIT or VIEW "$RUN$" VARIABLES"
12071 Prompt$(3)="WHAT "$RUN$" ARE "$STD$" VARIABLES?"
12074 Prompt$(4)="HOW TO DEFINE OUTGASSING VARIABLES?"
12077 Prompt$(5)="EDIT or VIEW "$STD$" VARIABLES"
12080 Prompt$(6)="ADD to "$STD$" VARIABLES"
12083 Prompt$(7)="ADD to "$STD$" VARIABLES"
12086 Prompt$(8)="PRINT OUT ALL "$STD$" VARIABLES"
12089 Prompt$(9)="PRINT OUT ALL "$STD$" VARIABLES"
12092 IF NOT Auto THEN Prompt$(10)="RETURN TO BMC"
12095 IF Auto THEN Prompt$(10)="BEGIN AUTOMATIC RUNNING"
12098!
12101 ON KEY 0 LABEL "NEW RUN VARIABLES" GOTO New_runvars
12104 ON KEY 1 LABEL "EDIT RUN VARIABLES" GOTO Edit_runvars
12107 ON KEY 4 LABEL "HELP" GOTO Help
12110 ON KEY 5 LABEL "OUTGAS" GOTO Outgas
12113 ON KEY 6 LABEL "ADD "$STD$" VARIABLES" GOTO Add_stdvars
12116 ON KEY 7 LABEL "ADD "$STD" VARIABLES" GOTO Add_stdvars
12119 ON KEY 8 LABEL "PRINT "$STD$" VARIABLES" GOTO Print_stdvars
12122 ON KEY 9 LABEL "PRINT "$STD$" VARIABLES" GOTO Print_stdvars
12125 IF NOT Auto THEN ON KEY 9 LABEL "ESCAPE" GOTO 12143
12128 IF Auto THEN ON KEY 9 LABEL "BEGIN RUNS" GOTO 12143
12131 CALL Keymenu(Prompt$(*)>
12137 GOTO 12137
12140 !
12143 IF Runchange THEN GOSUB Store_runvars
12146 IF Stdchange THEN GOSUB Store_stdvars
12149 SUBEXIT
12152 !
12155 Help=RESTORE 12158
12158 DATA RUN-variables define how the samples will be run during an automatic run-sequence. You must assign each sample its own RUN variables.
12161 DATA "STD-RUN variables are examples of typical RUN-variables that you can easily assign as the RUN variables. This way, instead"
12164 DATA "of having to fill in all of the entries for the RUN variables for a sample, you can just specify"
12167 DATA "the STD-RUN variables that best match what you want, and then edit just a few of the values."
12170 DATA "To use a particular set of STD-RUN variables as RUN-variables, enter the STD-RUN number as the RUN-ELEMENT, and enclose"
12173 DATA "it in asterisks (you may need a printout of the STD-run variables before you do this)."
12176 DATA "For example, *12*, to use values from STD-RUN number 12."!
12179 DATA "To get a display showing the names & isotopes of the STD-RUN variables, enter *?* as the RUN-ELEMENT."
12182 REDIM $S(6)
12185 READ $S(*)
12188 Helpscreen($S(*)>
12191 DATA "SSSSSSSSSSSSSYou can get specific HELP for any parameter in the RUN VARIABLES form by pressing CTRL-H instead of entering the parameter."
12194 REDIM $S(1)
12197 READ $S(*)
12200 Helpscreen($S(*)>
12203 GOTO Again
12206 !
12209 Outgas=RESTORE 12212
12212 DATA "To define RUN-variables for automatic outgassing (no data-taking), enter the"
12215 DATA "run-type as either OUTGAS or P-OUTGAS. To outgas the sample in front of the"
12218 DATA "source (normal running position), use OUTGAS."
12221 DATA "To outgas a sample in the preheat position, use P-OUTGAS."
12224 DATA "For preheat-position outgassing, enter the barrel-number of the actual sample to be outgassed."
12227 REDIM $S(5)
12230 READ $S(*)

71
Helpscreen($(*)
GOTO Again

Print_stdvars:Std=1

IF NOT Got_stdvars THEN GOSUB Get_stdvars
GOSUB Printvars
GOTO Again
Print_runvars:Std=0

IF NOT Got_runvars THEN GOSUB Get_runvars
GOSUB Printvars
GOTO Again

New_runvars:Run=0 ! define a completely new set of run-variables
Std=0

MAT Runvar0=(0)
MAT Run_order= (0)
MAT Lastresponse$= ("")
Last_runtype$="??"
WHILE Esc=0 AND Run<=32
Run=1+Run
Lastresponse$(0)="??"
Lastresponse$(3)="??"
Auto_form(Run,0,Esc,Stdvar(*),Sample_name$(*),Std_name$(*),Stdtype$(*),Checked_els,Element$(*),Std_iso(*))
IF Esc THEN
Esc=0
Run=Run-1
GOTO 12323
END IF
GOSUB Extract_runvars
ENDWHILE
IF Run=0 THEN figain

Edit_runvars: edit or view the run-variables for a single run
OUTPUT KBD;Clear$;
OFF KEY
INPUT "WHICH RUN-NUMBER DO YOU WISH TO EDIT or VIEW? (press CONTINUE to escape)" ,Run
IF Run<1 OR Run>32 THEN Again
IF Run_order(1)=0 THEN GOSUB Get_runvars
GOSUB Convert_runvars
Auto_form(Run,0,Esc,Stdvar(*),Sample_name$(*),Std_name$(*),Stdtype$(*),Checked_els,Element$(*),Std_iso(*))
IF Esc THEN
Esc=0
GOTO Edit_runvars
END IF
GOSUB Extract_runvars
GOTO Edit_runvars

Edit_stdvars: edit/view the standard-run variables
OUTPUT KBD;Clear$;
OFF KEY
INPUT "WHICH STANDARD-RUN NUMBER TO YOU WISH TO EDIT or VIEW (press CONTINUE to escape)" ,Run
IF Run<1 OR Run>32 THEN Again
12113 IF NOT Get_stdvars THEN GOSUB Get_stdvars
12146 FOR I=1 TO 32
12149 IF Stdvar(I,1)=0 THEN 12125
12152 NEXT I
12154 Stdmax=I-1
12158 IF Run>Stdmax THEN
12161 Clunk
12164 DISP C:$" SORRY, ONLY";Stdmax:"STANDARD RUNS HAVE BEEN DEFINED "&C$#
12167 WAIT 2
12170 GOTO Again
12173 END IF
12176 GOSUB Convert_stdvars
12179 Auto_form(Run,1,Esc,Stdvar(*),Sample_name$(*),Std_name$(*),Std_type$(*),Checked_els,Element$(*),Std_iso(*))
12182 IF Esc THEN
12185 Esc=0
12188 GOTO EdiStduars
12191 END IF
12194 GOSUB Extract_stdvars
12197 GOTO Edit_stdvars
12200 OFF KEY
12203 IF NOT GoUstdvars THEN GOSUB Get_stdvars
12206 Last_response$="??"
12209 Last_response$(-3)="??"
12212 Run=1
12215 WHILE Run_order(Run) AND Run<32
12218 Run=1+Run
12222 END WHILE
12225 IF Run>32 THEN
12228 DISP FNHK"32 POSSIBLE RUNS ARE DEFINED ON THIS DATA-DISK"
12231 Clunk
12234 WAIT 2
12237 GOTO Again
12240 END IF
12243 Add_runvars:Std=0 ! add to a set of existing run-variables
12246 OFF KEY
12249 IF NOT Get_runvars THEN GOSUB Get_runvars
12252 Last_response$="??"
12255 Last_response$(-3)="??"
12258 Run=1
12261 WHILE Run_order(Run) AND Run<32
12264 Run=1+Run
12267 END WHILE
12270 IF Esc THEN
12273 Esc=0
12276 GOTO figain
12279 END IF
12282 GOSUB Extract_runvars
12285 GOTO 12182
12288 OFF KEY
12291 IF NOT Get_stdvars THEN GOSUB Get_stdvars
12294 Run=1
12297 WHILE Stdvar(Run,1) AND Run<32
12300 Run=1+Run
12303 END WHILE
12306 IF Run>32 THEN
12309 DISP FNH("32 POSSIBLE STANDARD RUNS ARE DEFINED ")
12312 Clunk
12315 WAIT 2
12318 GOTO Again
12321 END IF
12324 Add_stdvars:Std=1 ! add to existing standard-run variables
12327 OFF KEY
12330 IF NOT Get_stdvars THEN GOSUB Get_stdvars
12333 Run=1
12336 WHILE Stdvar(Run,1) AND Run<32
12340 Run=1+Run
12343 END WHILE
12346 IF Run>32 THEN
12350 DISP FNH("ALL 32 POSSIBLE STANDARD-RUNS ARE DEFINED ")
12353 Clunk
12356 WAIT 2
12359 GOTO Again
12362 END IF
12365 OFF KEY
12368 MAT Last_response$= ("")
12371 Auto_form(Run,1,Esc,Stdvar(*),Sample_name$(*),Std_name$(*),Std_type$(*),Checked_els,Element$(*),Std_iso(*))
12593 IF Esc THEN
12596   Esc=0
12599   GOTO Again
12602 END IF
12605 GOSUB Extract_stduars
12608 GOTO Again
12611
12614 Printvars:! printout all run or std-run variables
12617   PRINTER IS 701;WIDTH (132)
12620   OFF KEY
12623   ON KEY 9 LABEL "   ESCAPE" GOTO 12887
12626 PRINT USING "2(K)";CHR$(27)$"&16D";CHR$(27)$"&16D" ! small print # 6 lines/inch
12629 IF NOT Std THEN ! runvar printout
12632 PRINT USING "2/,K,2/,?e,^,llfl,3QH,30H,3l :l,/";"RUN VARIABLES: "
12635 FOR Run=1 TO 32
12638   I=Run_order(Run); Barrell# for this run
12641   Ogas(Run)=(PQS(UPC$(Run),1),"OUTGAS")();
12644   IF I=0 THEN 12683
12647   PRINT TAB(2);I;TAB(6):
12650   IF Run_name$(Run)<>"" THEN PRINT Run_name$(Run);
12653   IF Run_name$(Run)<>"" THEN PRINT Sample_name$(I);
12656   IF Ogas(Run) THEN
12659       PRINT TAB(5);"****";
12662   ELSE
12665       FOR J=1 TO 8! printout run isotopes
12668       IF Run_iso(J) THEN PRINT TAB(9+6*(J-1));Run_iso(J);
12671 NEXT J
12674 END IF
12677 PRINT TAB(92);Run$(Run);
12680 NEXT Run
12683 PRINT USING "3/,5H,8fi,17J{,13R"; "STANDARD-RUN VARIABLES: 
12686 ELSE ! standard-runvar printout
12689 PRINT USING "2/,K,2/,l3fi,1H,1B,8K,1B,l1H,8fi,/"; "STANDARD RUN NUMBER"
12692 FOR Srun=1 TO 32
12695 IF Stdvar(Srun,1) THEN
12698   PRINT TAB(6);Srun$(Srun);UPC$(Stdtype$(Srun));TAB(29);Std_name$(Srun);
12701   IF UPC$(Stdtype$(Srun),1,81),"OUTGAS" THEN PRINT TAB(50);"****";
12704   ELSE
12707     FOR J=1 TO 8
12710     IF Std_iso(Srun, J) THEN PRINT TAB(10+6*(J-1));Std_iso(Srun, J);
12713 NEXT J
12716 END IF
12719 END IF
12722 END IF
12725 NEXT Srun
12728 PRINT USING "3/,2K,8fi,21!U:"; "STANDARD-RUN ISOTOPES"
12731 END IF
12734 !
12737 FOR Group=0 TO 1
12740 IF NOT Std AND Group AND NOT Run_order(17) THEN 12887
12743 IF Std AND Group AND Stdvar(17,1) THEN 12887
12746 FOR Run=1+16*Group TO 16+16*Group
12749 R=Run-16*Group
12752 IF NOT Std AND Run_order(R) THEN PRINT TAB(24+6*R);Run_order(R);
12755 IF Std AND Stdvar(R,1) THEN PRINT TAB(24+6*R);Run;
12758 NEXT Run
12761 PRINT USING "/"
12767 DATA SINGLE(1)/TRIPLE(3),FOCUSING ISOTOPE,CENTER-FIL BEAM (U),INITIAL CF-CURR.(A),DAY (0/1/2),CURRENT-1,RATE-1,WAIT-1 (MIN .),CURRENT-2,RATE-2
12770 DATA WAIT-2 (MIN.),DATA WAIT (MIN.),ABORT-CURRENT,MIN BEAM (U),MAX. BEAM (U),DEFAULT CURRENT,DEFAULT BEAM (U),FIL. INCREASE /BLOCK
12773 DATA MIN. #:BLOCKS,MAX. #:BLOCKS,MAX SIGMA MEAN(2),#SETS/BLOCK,MAX. GROWTH (2/MIN.),PREHEAT CF (A),PREHEAT SF (A),NOUSPIKE#,”
12776! RESTORE 12767
12778 FOR J=1 TO 26
12780 PRINT J;TAB(6);#(1,19);TAB(30);
12781 FOR Run=1 TO 16
12784 I=Run+16*Group
12787 U=Run var0(I,J)
12800 IF NOT Std AND Run order(I) THEN
12803 IF ((U=0 AND (J=1 AND J(5)) OR (J=24 AND U=0) OR (J=25 AND U=0) OR (J=26 AND U=0) OR (J=23 AND U=100) OR (Gass(1) AND (J>5 OR J<11)) THEN
12806 PRINT “--“;TAB(30+6*Run);
12809 ELSE
12812 PRINT U;TAB(30+6*Run);
12815 END IF
12818 END IF
12821 IF Std AND Stdvar(I,1) THEN
12824 U=Stdvar(I,J)
12827 IF ((U=0 AND (J=1 AND J(5)) OR (J=24 AND U=0) OR (J=25 AND U=0) OR (J=26 AND U=0) OR (J=23 AND U=100) THEN
12830 PRINT “--“;TAB(30+6*Run);
12833 ELSE
12836 PRINT U;TAB(30+6*Run);
12839 END IF
12842 END IF
12845 NEXT Run
12848 PRINT
12851 NEXT J
12854 PRINT
12857 IF NOT Std THEN
12860 FOR R=1 TO 16
12863 Run=R+16*Group
12866 IF Run_order(Run) THEN PRINT TAB(2+6*R);Run;
12869 NEXT R
12872 PRINT USING “/,.30X,K,”;”RUN-NUMBER”
12875 END IF
12878 PRINT USING “2/”
12881 NEXT Group
12884!
12887 PRINT USING “0,.2(X)”;CHR$(27),“NAME”,CHR$(27),“4180” ! Resume std typesize
12890 PRINTER IS CRT
12893 RETURN
12896!
12899 Store_runvars! store the run-variables on disk
12902 ON ERROR GOTO 12935
12905 ASSIGN #Path1 TO “RUNVAR:INTERNAL,4,1”
12908 GOTO 12917
12911! ON ERROR GOTO 6770
12914! ASSIGN #Path1 TO “RUNVAR:INTERNAL”
12917! ON ERROR GOTO 12935
12920 OUTPUT #Path1:Runvar0(*),Run_order(*),Run_name(*),Run_type(*),Run_iso(*)
12923 OFF ERROR
12926 ASSIGN #Path1 TO *
12929 RETURN
12932!
12935 OFF ERROR

75
IF ERRH=59 THEN RETURN !EOF- no data in file yet
CALL Bad_udisk(Std,1)
GOTO Again
! When storing/retrieving runvars from disk, always try left-hand drive first, then right-hand drive if left doesn't work
GOTO flgain
Store_stdvars: store the std-run variables on left-hand disk
ON ERROR GOTO 12935
ASSIGN @Path1 TO "STDVAR:INTERNAL,4,1"
GOTO 12995
ON ERROR GOTO 6770
ASSIGN @Path1 TO "STDVAR:INTERNAL"
OUTPUT @Path1;Stdvar(*),Std_iso(*),Std_name$(*),Stdtype$(*)
OFF ERROR
ASSIGN @Path1 TO *
RETURN
Get_stdvars: get the std-run variables from left disk
DISP "Loading Std-Run Variables from disk..."
ASSIGN @Path1 TO "STDVAR:INTERNAL,4,1"
GOTO 13001
ON ERROR GOTO 6809
ASSIGN @Path1 TO "STDVAR:INTERNAL"
GOTO 13009
ENTER @Path1;Stdvar(*),Std_iso(*),Std_name$(*),Stdtype$(*)
OFF ERROR
ASSIGN @Path1 TO "STDVAR:INTERNAL"
GOTO 13010
ENDIF
ON ERROR GOTO 13070
ENTER @Path1;Stdvar(*),Std_iso(*),Std_name$(*),Stdtype$(*)
RETURN
Get_runvars: get run variables from left-hand disk
DISP "Loading Run Variables from disk..."
ASSIGN @Path1 TO "RUNVAR:INTERNAL,4,1"
GOTO 13031
ON ERROR GOTO 6809
ASSIGN @Path1 TO "RUNVAR:INTERNAL"
GOTO 13039
ENTER @Path1;Runvar(@*),Run_order(*),Run_name$(*),Runtype$(*),Run_iso(*)
OFF ERROR
ASSIGN @Path1 TO "RUNVAR:INTERNAL"
RETURN
IF ERRH=59 THEN RETURN !EOF- no data in file yet
CALL Bad_udisk(Std,0)
GOTO Again
Extract_stdvars: Std_name$(Run)=Response$(-1) ! convert to MAIN format
Stdtype$(Run)=Response$(2)
FOR J=1 TO 8
Std_iso(Run,J)=Isotope(J)
NEXT J
FOR U=1 TO 27
Stdvar(Run,U)=Nresp(U)
NEXT U
Stdchange=1
RETURN
13118 Extract_runuars: ! convert to MAIN program format
13119 Run_order(Run)=Nresp(-3)
13120 Run_type(Run)=Response(-2)
13121 Run_name(Run)=Response(0)
13122 FOR J=1 TO 8
13123 Run_iso(Run,J)=Isotope(J)
13124 NEXT J
13125 FOR U=1 TO 27 ! extract the 27 run-variables
13126 Runvar(0,Run,U)=Nresp(U)
13127 FOR J=1 TO J
13128 IF Run_order(Run) AND Run_name(Run)="" AND Run_name(Run)="" THEN Sample_name(Run_order(Run))+Run_name(Run)
13129 Run_change=1
13130 RETURN
13131!
13132 Convert_stduars: Last_response(0)=Std_name(Run) ! convert for MENU2 format
13133 Last_response(-2)=Std_type(Run)
13134 Last_response(-1)=URL(Std_iso(Run,1))
13135 FOR J=2 TO 8
13136 IF Std_iso(Run,J) THEN Last_response(-1)=Last_response(-1)="URL(Std_iso(Run,J))
13137 NEXT J
13138 FOR M TO 27 ! extract the 27 run-variables
13139 Last_response(U)=URL(Stdvar(Run,U))
13140 NEXT U
13141 RETURN
13142!
13143 Convert_runuars: ! convert to MENU2 format
13144 Last_response(0)=Run_name(Run)
13145 Last_response(-3)=URL(Run_order(Run))
13146 Last_response(-2)=Run_type(Run)
13147 Last_response(-1)=URL(Run_iso(Run,1))
13148 FOR J=2 TO 8
13149 IF Run_iso(Run,J) THEN Last_response(-1)=Last_response(-1)="URL(Run_iso(Run,J))
13150 NEXT J
13151 FOR U=1 TO 27 ! extract the 27 run-variables
13152 Last_response(U)=URL(Stdvar(Run,U))
13153 NEXT U
13154 RETURN
13155!
13156 Sub Beep(Hertz,Seconds,Wait,Repeat,OPTIONAL Tune(*))
13157! if NFPR=4 then repeat a beep specified by the 1st 3 parameters.
13158! if NFPR=5 then beep a tone-pattern described by the Tune array.
13159 IF NFPR=-1 THEN
13160 FOR I=1 TO Repeat
13161 BEEP Hertz,Seconds
13162 WAIT Wait
13163 NEXT I
13164 ELSE
13165 EXIT
13166 I=1
13167 WHILE Tune(I,1)>0
13168 BEEP Tune(I,1),Tune(I,2)
13169 I=I+1
13170 END WHILE
13171 END IF
13172 Sub END ! ---------------------------------------------------------------------
13173!
13174 Size: SUB Size(REAL Bmin,Bmax,Default_beam,Default_curr,Abort,INTEGER F,REAL Rate,Wait)
adjust filament at rate of Rate \(a/sec\) to keep beam size \(\leq B_{\text{max}}\) \& \(\leq B_{\text{min}}\) (fil.curr. < Default_curr \&), while allowing no more than Abort amps fil-curr. \(F\) indicates center(1) or side (2) filaments.

**OPTION BASE 1**

```sql
13304 COM /General/ Z#,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),I_t
13305 COM /Specs/ Mx(0:1),ions,Ze(0:1),Noise(0:1)
13306 COM /Daly/ INTEGER Mu,Daly,Mn#(0:3,2)(),Daly ok(0:24),Ff#(0:1,2)()!
13307 COM /Magnet/ Mncoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
13308 COM /Filaments/ Filament(*),Fil#(*),INTEGER Nfils,F8
13309 INTEGER Pr
13310 INTGERR Pr
13311 SUBFLRG key: 0 = OK; 1 = beam too large, so reduced current
13312 ! 2 = beam too small and beyond abort-current
13313 ! 3 = failed ENTER_BERM check more than 6 times
13314 ! 4 = failed filament flag-check in CHANGE_CURRENT > 3 times
13315 FOR I=Auto*Full_auto+2 TO 19
13316 ON KEY I LABEL "" CALL Clunk
13317 NEXT I
13318 Subflag=0
13319 Correct(Tt$(*),Mt$(*),Filament(*),Magnet(0,2),Mu,L,Coarsemag(1:10),Foc(*))
13320 WAIT I
13321 Screw=0
13322 !
13323 REPEAT
13324 Enter_beam(X,Mu,L,1,Pr)
13325 Screw=1-Screw
13326 UNTIL Pr=1 OR Screw=6
13327 !
13328 IF Screw=6 THEN
13329 Subflag=3
13330 Error_message(Prtr(*),3,"FAILED ENTER_BERM SUBFLAGS >6 TIMES IN SIZE")
13331 SUBEXIT
13332 END IF
13333 !
13334 W=beam/Mn(1:1,E-9) ! W is size of beam-window, in ratio of max-beam to min-beam
13335 IF W=Bnin AND W=beam THEN SUBEXIT ! beam within window
13336 ! IF W<beam\(B_{\text{min}}\) THEN\n13337 ! beam too large: reduce filament-current
13338 Subflag=1
13339 Pstep=.023-.005*(W<3)-.005*(W<2)-.005*(W<1.5) ! decrease filament-current by 2.3% if beam-window ratio<3, by 1.0% if <2, by 1.3% if <1.5
13340 ! and by 0.8% if <1.5
13341 F5tep=MAX(Pstep*Filament(F),.012) ! but not less than .012 amps
13342 Target=Filament(F)-Fstep
13343 GOTO Change_current
13344 !
13345 END IF
13346 !
13347 IF F=1 AND Nfils=2 THEN Increase_curr ! if center-fil beam for a triple
13348 ! IF Mu=0 AND Daly AND Mu<3 THEN
13349 Mu=1
13350 GOTO 13361
13351 END IF
13352 !
13353 Bu=Bmax/(Bmin*1.E-9) ! size of beam-window, in ratio of max-beam to min-beam
13354 IF W>Bnin AND W=beam THEN SUBEXIT ! beam within window
13355 ! IF W=beam\(B_{\text{max}}\) THEN \n13356 ! beam too large: reduce filament-current
13357 Subflag=1
13358 Pstep=.023-.005*(W<3)-.005*(W<2)-.005*(W<1.5) ! decrease filament-current by 2.3% if beam-window ratio<3, by 1.0% if <2, by 1.3% if <1.5
13359 ! and by 0.8% if <1.5
13360 F5tep=MAX(Pstep*Filament(F),.012) ! but not less than .012 amps
13361 Target=Filament(F)-Fstep
13362 GOTO Change_current
13363 !
13364 END IF
```

78
Increase curr: beam too small- increase filament-current.

IF Flag = Abort THEN
  GOTO Exit_size
END IF

Pstep = 0.15(0.05*<Bw(3)>.005*<Bw(2)>.004*<Bw(1.5)) increase filament-current by 1.9% if beam-window ratio >3, by 1.4% if < 3.

by 0.9% if >1.5 <2, by 0.5% if < 1.5.

Fstep = MAX(PstepFilament(F), 0.01) but not < 0.09 angs

Target = Filament(F) + Fstep

Bad contacts = 0

Change current: REPEAT

CALL Filament(10*TargetFilament(F)+Rate*TargetFilament(F), 1, F)

Bad contacts = Bad contacts + Subflag

UNTIL Subflag = 1 OR Bad_contacts > 2

IF Bad_contacts > 2 THEN
  Subflag = 1
  Error_message(Prtr(*), j, "FAILED CONTACT-TEST 3 TIMES IN SIZE-INVOKED CHANGE_CURRENT")
  SUBEXIT
END IF

WAIT (IMEDIATE, Wait), Filament(F), Magnet(l1), Auto, Full_auto

GOTO 13361

Exit_size: DISP

Spikecorr: SUB Spikecorr(Aver(*), Last_aver(*), Ratio(*), Acc(*), Last_ratio(*), Delta(*), INTEGER Prtr(*), N, Block)

correct observed ratios for spike isotopes and (linear) fractionation. Bodson's algorithm.

OPTION BASE 1

CON /Spikedrun/ Spike$, Spikedrun_ratio(*), Spike_ratio(*), Natural_ratio(*), INTEGER Iso_dif(*), Spkdrun_iso(*), Spkdrun_ratio

DIM Fract(2), R1(2), R2(2), R3(2), Cov(2), Srad(2), U(2), Sf(2), Spkdrun_signa(3), Spkdrun_ratio(3), Rad(2)

R = 0

Spkdrun_ratio = 0

FOR I = 1 TO 3-(Spkdrun_iso(3) = 0 OR N = 3) ! find largest spike-ratio
  IF Spike_ratio(I) = 0 THEN
    GOTO 13536
  END IF
END FOR

R = R + 1 ! add "Spike/SPK" ratio as new ratio (radiogenic)

(N passed in parens, so shouldn't affect N in main program)

K3 = Iso_dif(1) * (Natural_ratio(2)/Spkdrun_ratio(2) - 1) - Iso_dif(2) * (Natural_ratio(1)/Spkdrun_ratio(1) - 1)

K4 = Iso_dif(2) * (Spkdrun_ratio(1) - 1) - Iso_dif(1) * (Spkdrun_ratio(2) - 1)

13652 \[ \text{Za}(I)=\text{Natural\_ratio}(I)+R^{*}\text{Spike\_ratio}(I) \]

13655 NEXT I

13658 \[ \text{Ca}(2H)\text{os}(l)*Za(2)/(r;i*\text{Spkd\_ratio}(2)A 2) \]

13661 \[ 5r=S_gR(R8S(Ca(l) A 2*Spkdrun\_signa(l) A 2tCa(2) A 2*Spkdrun\_signa(2) A 2)) \]

13665 IF N=6 THEN \[ \text{fract}(I)\text{Za}(I)/(\text{Iso\_dif}(I)*\text{Spkdrun\_ratio}(I)\text{*(1+R))} \]

13667 FOR I=1 TO 2

13670 \[ U(I)=\text{Mso\_dif}(3)*\text{Fract}(I) \]

13673 \[ R(I)=(1+R)*U(I) \]

13676 \[ \text{psi}=(I=1)-(I=2) \]

13679 \[ \text{sf}(I)=\sqrt{\text{G}(I)2+S_{r}-2^2(I)2+2*\text{psi}(I)2} \]

13682 \[ \text{Courf}(I)=\sqrt{R(B(I)2*\text{psi}(I)2+\text{iso\_dif}(I)2)2} \]

13685 \[ \text{Srad}(I)=\sqrt{RBS(R(I)2*\text{psi}(I)2+\text{iso\_dif}(I)2)2} \]

13688 \[ \text{Rada}(I)=R(I)^{*}\text{psi}(I)2\text{Iso\_dif}(I)2 \]

13691 \[ \text{Ratio}(I)=\text{Ratio}(I-1)2+\text{psi}(I)2\text{Iso\_dif}(I)2 \]

13694 \[ \text{Ratio}(I)=\text{Ratio}(I-1)2+\text{psi}(I)2\text{Iso\_dif}(I)2 \]

13697 \[ \text{Courf}(I)=\sqrt{R(B(I)2*\text{psi}(I)2+\text{iso\_dif}(I)2)2} \]

13699 \[ \text{Srad}(I)=\sqrt{R(B(I)2*\text{psi}(I)2+\text{iso\_dif}(I)2)2} \]

13701 \[ \text{Srad}(I)=\sqrt{R(B(I)2*\text{psi}(I)2+\text{iso\_dif}(I)2)2} \]

13704 \[ \text{Srad}(I)=\sqrt{R(B(I)2*\text{psi}(I)2+\text{iso\_dif}(I)2)2} \]

13706 \[ \text{psi}=(I=1)-(I=2) \]

13709 \[ \text{psi}=(I=1)-(I=2) \]

13712 \[ \text{psi}=(I=1)-(I=2) \]

13715 \[ \text{psi}=(I=1)-(I=2) \]

13718 \[ \text{psi}=(I=1)-(I=2) \]

13721 \[ \text{psi}=(I=1)-(I=2) \]

13724 \[ \text{psi}=(I=1)-(I=2) \]

13727 \[ \text{psi}=(I=1)-(I=2) \]

13730 \[ \text{psi}=(I=1)-(I=2) \]

13733 \[ \text{psi}=(I=1)-(I=2) \]

13736 \[ \text{psi}=(I=1)-(I=2) \]

13739 \[ \text{psi}=(I=1)-(I=2) \]

13742 \[ \text{psi}=(I=1)-(I=2) \]

13745 \[ \text{psi}=(I=1)-(I=2) \]

13748 \[ \text{psi}=(I=1)-(I=2) \]

13751 \[ \text{psi}=(I=1)-(I=2) \]

13754 \[ \text{psi}=(I=1)-(I=2) \]

13757 \[ \text{psi}=(I=1)-(I=2) \]

13760 \[ \text{psi}=(I=1)-(I=2) \]

13763 \[ \text{psi}=(I=1)-(I=2) \]

13766 \[ \text{psi}=(I=1)-(I=2) \]

13769 \[ \text{psi}=(I=1)-(I=2) \]

13772 \[ \text{psi}=(I=1)-(I=2) \]

13774 \[ \text{psi}=(I=1)-(I=2) \]

13777 \[ \text{psi}=(I=1)-(I=2) \]

13780 \[ \text{psi}=(I=1)-(I=2) \]

13783 \[ \text{psi}=(I=1)-(I=2) \]

13786 \[ \text{psi}=(I=1)-(I=2) \]

13789 \[ \text{psi}=(I=1)-(I=2) \]

13792 \[ \text{psi}=(I=1)-(I=2) \]

13795 \[ \text{psi}=(I=1)-(I=2) \]

13798 \[ \text{psi}=(I=1)-(I=2) \]

13801 \[ \text{psi}=(I=1)-(I=2) \]

13804 \[ \text{psi}=(I=1)-(I=2) \]

13807 \[ \text{psi}=(I=1)-(I=2) \]

13810 \[ \text{psi}=(I=1)-(I=2) \]

13813 \[ \text{psi}=(I=1)-(I=2) \]

13816 \[ \text{psi}=(I=1)-(I=2) \]

13819 \[ \text{psi}=(I=1)-(I=2) \]

13822 \[ \text{psi}=(I=1)-(I=2) \]
J.JUi.-J I

13826 !
13829 DISPIKES:SUB DISPIKES ! show $s & names of spikes defined on disk
13832 OPTION BASE 1
13835 DIM /Spikedrun$(*),Spike_ratio(*),Natural_ratio(*),ISO_iso_diff(*),Spikedrun_iso(*),Spkdrun_ref
13838 DIM Drive$(*),Drive$(*),Drive$(*),Drive$(*),Drive$(*),Drive$(*)
13841 Drive$(*):="LEFT"
13844 Drive$(*):="RIGHT"
13847 OUTPUT KBD:"CHR$(255)&CHR$(75)"
13850 FOR Disk=1 TO 0 STEP -1
13853 ON ERROR GOTO 13865
13856 ASSIGN Pathl TO "SPIKE:INTERNAL,1,"&Disk$(Disk)
13859 PRINT "SPIKES DEFINED ON DISK IN "&Drive$(Disk)&" DRIVE:"
13862 GOTO 13877
13865 NEXT Disk
13866 Clunk
13867 PRINT FNHF8:" NO SPIKE FILE PRESENT IN EITHER DRIVE ")
13874 SUBEXIT
13877 PRINT USING ":,8R,8X,6R,7H,5R,";FNHF8("NAME")
13880 ON ERROR GOTO 13913
13883 FOR I=1 TO 20
13886 ENTER Pathl,I;Spikc$,Spkdrun_ref,Spikedrun_iso(*),Spikedrun_ratio(*),ISO_iso_diff(*),Spike_ratio(*),Natural_ratio(*)
13889 IF TRIN$(Spkdrun_ref) THEN PRINT "I;";Spikc$;"=";Spkdrun_ref;
13892 PRINT I;"=";Spkdrun_iso(J);"=";Spkdrun_iso(J);
13895 FOR J=1 TO 3
13898 IF Spkedrun_iso(J) THEN PRINT Spkedrun_iso(J);"=";
13901 NEXT J
13904 PRINT
13907 NEXT I
13910 NEXT I
13913 PRINT
13916 OFF ERROR
13919 SUBEND ! ---
13922 !
13925 !
13928 Cubic:SUB Cubic(Coef(*),Peak_t(*),Peak_array(*),INTEGER Nsets,!
13931 ! calculate least-squares cubic fit to monitor isotope with time
13934 OPTION BASE 1
13937 DIM Time_array(Nsets),Peak(Nsets),Xa(Nsets),Hb(Nsets),Hc(Nsets),Hd(Nsets)
13940 READ IN time_array(Nsets),Peak(Nsets),Xa(Nsets),Hb(Nsets),Hc(Nsets),Hd(Nsets)
13943 FOR J=1 TO Nsets
13946 Time_array(J,1)=1 ! power-array of X-values (1,1^1/2,1^3)
13949 FOR P=2 TO 4
13952 Time_array(J,P)=Time_array(J,P-1)+Peak_t(J)
13955 NEXT P
13958 Peak(J,1)=Peak_array(I,J) ! vector of Y-values
13961 NEXT J
13964 MAT Ha= TRN(Time_array)
13967 MAT Xa= Ha*Time_array
13970 MAT Xc= INX(Ha)
13973 MAT Xa= Xc*Xa
13976 MAT Coef= Xa*Peakc ! coefficients of best-fit 3rd-order polynomial
13979 SUBEND ! ---
13982 !
13985 Focscan:SUB Focscan(INTEGER Foc(*),I,t) ! scan focus-potentials
13988 OPTION BASE 1
13991 DIM Prompt$(*),Response$(*),Response$(*),Range(*,2)
13994 COM /Specs/ Ms(0,1),Ze(0,1),Noise(0,2)
13997 COM /Daly/ INTEGER Ms,Daly,Mw(0,2),CW,CH,Daly.Ok(0,2),Fr(0,2),C3
14000 COM /Filaments/ filament(*),filb(*),INTEGER Nfiles,filnum

81
INTEGER Pr
14005 2%="123456789"
14009  OFF KNob
14012 ON KEY 9 LABEL "  ESCAPE" GOTO 14114
14015 PRINT USING "8(3D,3X),8(3D,3X),8(3D,3X)";"FOCUSING VALUES:"1,2,3,4,5,6,7,8,Foc(*)
14018 DATA Plate#(1-8),Min. Value,Max. Value,Scan-Step
14021 DATA "1","0","999",1,1,0,0,999,1,999,1,1
14024 READ Prompt$,Response$,Range$(*)
14027 IF Nils=1 THEN ' don't allow plate-b scan if not a triple
14030 Range(I,2)=7
14033 Prompt$(1)(I)(1111)="7"
14036 END IF
14039 MAT Use= (1)
14042 Form(Prompt$,Response$,Use$(*)
14045 IF THEN SUBEXIT
14048 OUTPUT KBO-,CHR$(255)&CHR$(75);
14051 Plate=VAL(Response$(1))
14054 Xmin=VAL(Response$(2))
14057 Xmax=VAL(Response$(3))
14060 Step=VAL(Response$(4))
14063 IF (Plate=0) AND (Xmin<=500) THEN Xmax=500
14066 IF (Plate=0) AND (Xmax>=500) THEN Xmax=550
14069 IF Step<=1 THEN Step=1
14072 Enter_beam(X,Y,Mu,t,It,Fr)
14075 IF Fr>1 THEN 14018
14078 ON KEY 3 LABEL " DOUBLE STEP" GOSUB Double
14081 ON KEY 8 LABEL " HALVE STEP" GOSUB Halve
14084 ON KEY 9 LABEL " ESCAPE" GOTO 14114
14087 Rxs(0,100,8,100,Xmin,Xmax,0,1.3*Mu,"PLx"(Plate),"MU BEAM",0,Mu,0)
14090 I=Xmin
14093 WHILE I<=Xmax
14096 OUTPUT 8 USING "2(M,M):";"OR"(2*M)(10-Plate,10-Plate),I,"#Mv",I
14099 Enter_beam(X,Mu,t,I,It,Fr)
14102 IF I<Xmin THEN PLOT I,Mu
14105 IF I<Xmin THEN MOVe I,Mu
14108 I=I+Step
14111 END WHILE
14114 OUTPUT 8 USING "4M,4Z":"OR"(2*M)(10-Plate,10-Plate),Foc(Plate)
14117 WAIT 2
14120 GRAPHICS OFF
14123 ALPHA ON
14126 GOTO 14042
14129 Double+Step=2*Step
14132 BEEP 880,.07
14135 RETURN
14138 Halve=IF Step>=2 THEN Step=Step/2
14141 BEEP 220,.07
14144 RETURN
14147 SUBEND !-------------------------------------------------------------
14150 !
14153 !
14156 Calibrate:SUB Calibrate(REAL filament(*),INTEGER B,Pkx_inter,Line,It,Mu,Ms(*),Collector(*))
14159 OPTION BASE 1
14162 CM /Specs//Mu(*),Ions,Ze(*),Noise(*)
14165 DIM Mu_beam(3),New_Mu(0:1)
14168 OFF KEY
14171 OFF KNob
14174 ON KEY 9 LABEL "  ESCAPE" GOTO 14135
14177 Mu=0
14180 I,t=1
MAT New_nx= Mx
11186 OUTPUT KBD;CHR$(255)&CHR$(75);
11189 OUTPUT B: "$IDU"
11192 ENTER B;Count
11195 Beamsize=(Count-Ze(0))/New_nx(0)
11198 IF (Beamsize(1) OR (Beamsize(50) OR NOT Daly THEN
11201 PRINT USING "3,k,",";FHN$"("BEAM MUST BE >1 nV AND (<50 nV, Daly")&FHN$(""TO CALIBRATE DAILY GAIN.")")
11204 Clunk
11207 GOTO 11205
11210 END IF
11213 OUTPUT B USIHG "8","Df",B-Peak_inter/2 ! set magnet to -1/2 mass offset for zero-calibration
11216 Zero(Filament(*),0,Noise_time,Daly,B,Peak_inter,Mu,Mw*$(*),Collector$(*))
11219 OUTPUT KBD;CHR$(255)&CHR$(75);
11222 PRINT USING "K,2",";FHN$(""CALIBRATION OF DAILY GAIN")
11225 OUTPUT B USIHG "8","Df",B
11228 FOR I=1 TO 3
11231 Mu=(I*2)
11234 OUTPUT B;Mu$(Mu,i)
11237 WAIT (2*5+(I*2))
11240 Seconds_integr=10*(I*(NOT (Mu*(I>1)\=Beamsize(5)+Beamsize(2)))
11243 ! 10 sec. integration for Daly, 10 sec. for Cup if >50 nV, 20 sec if 2-5 nV, 30 sec if 1-2 nV
11246 FOR J=1 TO Seconds_integr
11249 OUTPUT B: "$IDU"
11252 ENTER B;Beancount
11255 Beam_num=(Beancount-Ze(Mu))/New_nx(Mu)
11258 Mu_beam(I)=Mu_beam(I)+Beam_num/Seconds_integr
11261 DISP Collector$(Mu,J);GROUND(Beam_num,3)
11264 NEXT J
11267 PRINT Collector$(Mu,1);RA(I,12);GROUND(Mu_beam(I),4)
11270 NEXT I
11273 Oldgain=New_nx(1)/New_nx(0)
11276 New_nx(1)=New_nx(I)=Mu_beam(2)/(Mu_beam(I)+Mu_beam(3))/2
11279 PRINT USING "2,5(K),2",";DAILY GAIN IS ",GROUND(New_nx(1))/New_nx(0),3," (LAST CALIBRATION WAS ",GROUND(Oldgain,3),")"
11282 MAT Mu= New_nx
11285 I_t=2
11288 OUTPUT B;Mu$(Mu,I_t)
11291 BEEP
11294 SUBEND !----------------------------------------------------------
11297 !
11300 !
11303 Zero:SUB Zero(REBL Filament(*),Zero_time,Noise_time,INTEGER Daly,8,Peak_inter,Mu,Mw*$(*),Collector$(*))
11306 !take collector zeroes (no more than every hour) and measure dark-noise (no more than every 6 hours)
11309 OPTION BASE 1
11312 COM /Spese/ Mx(*),Ions,Ze(*),Noise(*)
11315 COM /General/ Zn,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*),Itime
11318 COM Sums(5),Reading(20),Count_noise(0:1),New zeroes(0:1),New_noise(0:1)
11321 INTEGER Count time
11324 COM /Keyboard/ Cn$,G$,D$,Ct$,D$,Clear$
11327 MAT New_zeroes= Ze
11330 MAT New_noise= Noise
11333 OFF KEY
11336 OFF KB0
11339 ON KEY 9 LABEL "ESCAPE" GOTO 11663
11342 GRAPHICS OFF
11345 Subflag=0
11349 IF New_zeroes(I)>0 AND Daly THEN
11351 Zero_time=0
11354 Noise_time=0
11357 END IF
11360 IF (TIME(DATE-Zero_time)/3600)<1 THEN SUBEXIT ! zeros taken within last hour

83
PRINT USING "3/"
PRINT USING "K,2/" COLLECTOR ZERO & NOISE CALIBRATION: "X" Please Wait...

Nocurrent=Filament(1)=0 AND (Filament(2)=0)
OUTPUT 8:Mes$(0,2-Nocurrent)
MAT New_zeroes= (0)
Mu=Mu
WAIT (1+5*H)
IF NOT Nocurrent THEN

Scan magnet for lowest zero if filaments have currents
GOSUB Scan_for_min
IF Max_count<10000 THEN
OUTPUT 8;Hn$(0,2-Mu)
Mu=1
GOSUB Scan_for_min
GOTO 14507
ELSE
PRINT USING "10/,K";WT FINO R GOOD ZERO-POSITION"
Clunk
GOSUB Restore
SUBEXIT
END IF
!
Scan_for_min:Zero_location=300
OUTPUT 8 USING "4a,42":"$OFJ",Zero_location=#Peak_inter/8
WAIT 2
Min_count=1.E+99
Max_count=0
FOR C=? TO 0 STEP -1 ! find magnet-setting giving lowest zero
Mag_pos=INT(Zero_location-C*Peak_inter/8)
IF Mag_pos=0 AND Mag_pos=C.E+4 THEN
OUTPUT 8 USING "4a,42":"$OFJ",Mag_pos
OUTPUT 8:"$IDU"
ENTER 8;Count
HISP "Looking for best zero-position (CUP) ";Tag_pos,Count
IF Count<Min_count THEN
Min_count=Count
Mag_min=Mag_pos
END IF
IF Count>Max_count THEN Max_count=Count
END IF
NEXT C
OUTPUT 8 USING "4a,42":"$OFJ",Mag_min
RETURN
END IF
!
Hours=(TIMEDATE-Noise_time)/3600 ! hours since last collector-noise measurement
IF Hours<6 THEN Check_noise=1
FOR M=0 TO Daily
OUTPUT 8:Mes$(M,1)
Count_time=M-4*H+G*Check_noise
MAT Sums= (0)
FOR Ct=-5 TD Count_time
OUTPUT 8:"$IDU"
ENTER 8;Count
DISP Collector$;(" ZERBER";1B$(10);CHR$(12B+Ct));Count$;TAB(40);Count_time-Ct+1
IF Ct=0 THEN
Reading(Ct)=Count
14543 CALL Suns(Sums(0),1,0,Count,0)
14545 END IF
14549 NEXT I
14552 Calc_signal(Count_noise(M),Sums(3),Sums(4),Count_time)
14555 Est_zero=Sums(3)/Count_time
14558 IF I=1 TO Count_time ! 2-sigma rejection of outliers
14561 IF ABS(Est_zero-Reading(I))>2*Count_noise(M) THEN CALL Suns(Sums(0),-1,0,Reading(I),Count_time)
14564 NEXT I
14567 Calc_signal(Count_noise(M),Sums(3),Sums(4),Count_time)
14570 IF Check_noise THEN New_noise=M00_M00(M00(M)/M00(M)) noise in mV
14573 New_zeroes=M00_Sums(3)/Count_time
14576 NEXT M
14579 PRINT USING "33X,7fi,9H,8fi,/&";FHyn$(H\ COP\ "),FHUn$("OfiLY
14582 PRINT FNH$<"ZEROES:"r<counts/second)";TB8(36);im(New_zeroes(0»;Tfl8<51);INT(New_zeroes(l))
14585 IF Check_noise THEN PRINT FNH$<&"DRRHOISE:"r< (mV/second)";TB8(35);DROUHO(New_noise(0),3);TB8(50):DROUHO(New_noise(l),2);
14588 GOSUB Restore
14588 GOSUB Restore
14591 IF New_zeroes(0)>800 OR New_zeroes(1)<200 OR (Daily AND (New_zeroes(0)>900 OR New_zeroes(1)<300)) THEN Bad_readings
14594 IF Check_noise AND New_noise(0).1 OR New_noise(1).01 THEN Bad_readings
14597 IF Check_noise THEN Noise_tine=TIME_DATE
14600 Zero_tine=TIME_DATE
14605 MAT Ze= New_zeroes
14606 MAT Noise= New_noise
14609 SUBEXIT
14612 !
14615 Restore:OUTPUT B;Mn$(Mu,mtime) ! restore collector, integr. time
14618 OUTPUT 8 USING "$A,4;Z1,81;OFJ",B ! restore magnet-position
14621 RETURN
14624 !
14627 Bad_readings=Noise_time=0
14630 Subflag=1
14631 FOR P=1 TO full_auto TO 1 STEP -1
14636 PRINTER IS Prt(P)
14639 IF P=2 THEN PRINT USING "$3,80A,/,80A,/,;RPT$("\",80),RPT$("\",80)
14642 PRINT USING "$3,80A,/,80A,/,;RPT$("\",80),RPT$("\",80)
14645 PRINT USING "$3,80A,/,80A,/,;RPT$("\",80),RPT$("\",80)
14648 PRINT USING "$3,80A,/,80A,/,;RPT$("\",80),RPT$("\",80)
14651 PRINT USING "$3,80A,/,80A,/,;RPT$("\",80),RPT$("\",80)
14654 IF P=2 THEN PRINT USING "$3,80A,/,80A,/,;RPT$("\",80),RPT$("\",80)
14657 NEXT P
14660 Clunk
14663 SUBEND ! --------------------------------------------------------------
14666 !
14669 Mfocus=SUB Mfocus(Stripchart,Logon,tine0,8),INTEGER I_t,Fact(*),28) ! Manual focus-change
14672 OPTION BASE 1
14675 INTEGER Maxplate(8),Minplate(8),Pr
14679 DIM P$(8)
14681 COM /Magnet/ Mcof(*),INTEGER L,Aside,Peak_inter,Coarsemag(0:1),Magnet(0:29,2),Coarse
14684 COM /Keyboard/ Cn$,C1$,C2$,C3$,C4$,Clear$
14687 COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
14690 COM /Daily/ INTEGER Mu,Daily,MW(0:3,2),Daily,uk(0:29),FfB(0:1,2:4)
14693 COM /Filaments/ Filament(*),Fil$(0:*),INTEGER Mf1s,F8
14696 Maxplate= (999)
14699 MAT Minplate= (0)
14702 Maxplate(0)=540
14705 Minplate(0)=960
14708 I_t=2
14711 OUTPUT B;Mn$(Mu,I_t) ! .2 second integr-time
14714 OUTPUT KBD:Clear$;
14717 FOR I=1 TO 8
11720 P$(1)=FNB1$(T)
11723 NEXT I
11726 IF HOI Stripchart THEN
11729 PRINT USING "K,2/,K";"NflNUflL-FOiUS ROUTINE: ","Press a softkey or a number-key to indicate which plate to focus,"
11732 PRINT USING "K,2/,K,/";"u5e "aFNH$"(KNOB)" or up/down arrows, or the + - keys to adjust setting of plate. ";"Press "8
11735 PRINT "(NOTE: You can focus in this way but with real-time ion-beam graphics if"
11738 PRINT "a BEfiH CHfiRT ( k3 during BMC) was in use before the manual-focus request)"
11741 END IF
11744 A+KNOBK
11747 Plate=I
11750 P$(1)=FNB1$("I")
11753 I+=2
11756 FOR I=1 TO 6+Hfils
11759 ON KEY I LABEL " PLflTE "aUflLKD GOTO 11792
11762 NEXT I
11765 ON KEY 9 LABEL " ESCfiPE" GOTO Exit
11768 IF Stripchart THEN
11771 GRAPHICS ON
11774 ELSE
11777 GOSUB Print_plates
11780 PRINT TABXY(1,16);"PLATE"
11783 PRINT TABXY(1,17);"SETTING"
11786 END IF
11789 GOSUB Print_values
11792 GOTO 14652
11795 !
11798 Print_plates;FOR I=1 TO 6+Hfils
11801 IF NOT Stripchart THEN PRINT TABXY((I-1)*7,16);I
11804 NEXT I
11807 RETURN
11810 !
11819 Print_values;IF NOT Stripchart THEN
11822 FOR I=1 TO 6+Hfils
11825 PRINT TABXY((I-1)*7,17);Foc(I)
11828 NEXT I
11831 ELSE
11834 GOSUB Plates
11837 END IF
11840 RETURN
11846 RETURN
11849 !
11852 IF NOT Stripchart THEN GOSUB Intro
11855 Enter_beam(Counts,Mu,L,I,t,Pr)
11858 WAIT .10
11861 ON CYCLE .195 GOSUB Query_beam
11864 ON KBD ALL GOSUB Change_plate
11867 ON KNOB .10 GOSUB Change_setting
11870 GOTO 14870
11873 |
11876 Query_beam=Enter_beam(Counts,Mu,L,2,Pr)
11879 IF Pr=3 THEN Exit
11882 T=(TIME-1)/60
11885 IF Stripchart THEN
11888 IF NOT Lognon THEN DRAW T,Mu
11891 IF Lognon THEN DRAW T,LTG(ABS(Mu*(NOT Mu)))
ELSE
14907 IMAGE 2X,"mV","X","X","","3X","X",
14908 DISP USING 14897:Magnet(L,1),ROUND(Mv,4)
14909 END IF
14910 RETURN
14911 !
14912 Change_plate:CALL Keyboard(KBOS,K)
14913 SELECT K
14914 CASE 49 TO 54:Wfil" ! keys 1 thru 7 or 8 pressed
14915 LASTPLATE=PLATE
14916 PLATE=K-48
14917 CASE -207 TO -202:Wfil" ! keys kl thru k7 or k8 pressed
14918 LASTPLATE=PLATE
14919 PLATE=K+208
14920 CASE -162,43 ! up-arrow or plus key pressed
14921 Change=1
14922 GOTO 15020
14923 CASE -170,45 ! down-arrow or minus key pressed
14924 Change=-1
14925 GOTO 15020
14926 CASE -199 ! k9 pressed
14927 GOTO Exit
14928 CASE -176 ! PAUSE key pressed
14929 PAUSE
14930 CASE ELSE
14931 Clunk
14932 RETURN
14933 END SELECT
14934 BEEP .440, .08
14935 IF NOT Stripchart THEN
14936 GOSUB Print_plates
14937 Intro:PRINT TAB(X+12)+";Ci$;Plate;Cn$
14938 END IF
14939 IF Stripchart THEN
14940 PRINT "OF";$OF";8-PIATE,10-PIATE,Setting,"OFU",Setting
14941 END IF
14942 Setting=Foc(Plate)
14943 RETURN
14944 !
14945 Change_setting:Change=INT(KNOB/K/3)
14946 Setting=Foc(Plate)+Change
14947 IF Setting<MINPLATE(Plate) THEN 15050
14948 Setting=MINPLATE(Plate)
14949 END IF
14950 IF Setting>MAXPLATE(Plate) THEN 15050
14951 Setting=MAXPLATE(Plate)
14952 END IF
14953 GOSUB Plate
14954 END IF
14955 GOSUB Plates
14956 Setting=Foc(Plate)
14957 RETURN
14958 !
14959 Change_setting:Change=INT(KNOB/K/3)
14960 Setting=Foc(Plate)+Change
14961 IF Setting<MINPLATE(Plate) THEN 15050
14962 Setting=MINPLATE(Plate)
14963 END IF
14964 IF Setting>MAXPLATE(Plate) THEN 15050
14965 Setting=MAXPLATE(Plate)
14966 END IF
14967 GOSUB Plate
14968 END IF
14969 GOSUB Plates
14970 EXIT:OFF CYCLE
14971 OFF KNOB
14972 OFF KB0
15074 SUBEND ! -------------------------------
15077 !
15080 !
15083 Nspiker=SUB Nspike
15086 ! define values for a new spike
15089 OPTION BASE 1
15092 DIM /Spikedrun, Spike$,Spikedrun_ratio$(#,Spikedrun_iso$(#),Natural_ratio$(#),INTGER Iso_dif$(#),Spikedrun_iso$(#),Spkdrun_ref
15095 DIM Response$(5),Prompt$(5),Use$(5),Range$(5,2)
15098 CALL Dispike
15101 OFF KEY
15104 OFF KNOB
15107 PRINT TAB(1,1);"ENTER NUMBER (1-20) AND NAME (must include element-symbol) OF NEW SPIKE;"
15110 PRINT TAB(1,1);"(e.g. "SCHR#34" for Sr, "CH#24" for Ch") Enter 0,0 to return to BMC"
15113 INPUT Spike,Spike$[1,81
15116 IF Spike=0 THEN SUBEXIT
15119 IF Spike>20 THEN 15107
15122 REDIM Prompt$(5),Response$(5),Use(5),Range(5,2)
15125 DATA REFERENCE ISOTOPE (eg 86 for Sr),MONORADIOGENIC ISOT 1 (eg 87 for Sr),MONORADIOGENIC ISOT 2 (eg 88 for Sr),RADIOGENIC IS
15128 01 if any; eg 87 for Sr
15128 DATA ??,??,??,??,"",10,300,10,300,10,300,-1,-1
15131 READ Prompt$(5),Response$(5),Range(5,2)
15134 MAT Use=(1)
15137 Form(Prompt$(5),Response$(5),Use(5),Range(5,2),1,1)
15140 IF E THEN SUBEXIT
15143 Spkdrun_ref=UFL(Response$(1))
15146 Spikedrun_iso(1)=UFL(Response$(2))
15149 Spikedrun_iso(2)=UFL(Response$(3))
15152 Spikedrun_iso(3)=0
15155 IF ERROR GOTO 15161
15158 Spikedrun_iso(3)=UFL(Response$(4))
15161 OFF ERROR
15164 FOR I=1 TO 3
15167 Spikedrun_ratio$(I)=UFL(Spikedrun_iso(I))"/"&UFL(Spkdrun_ref)
15170 NEXT I
15173 REDIM Prompt$(5),Response$(5),Use(5),Range(5,2)
15176 Prompt$(1)="Spikedrun_ratio$(1)" OF NATURAL ELEMENT"
15179 Prompt$(2)="Spikedrun_ratio$(2)" OF NATURAL ELEMENT"
15182 Prompt$(3)="Spikedrun_ratio$(3)" OF SPIKE"
15185 Prompt$(4)="Spikedrun_ratio$(4)" OF SPIKE"
15188 IF Spikedrun_iso(3) THEN
15191 N=5
15194 Use$(5)=1
15197 Prompt$(5)="Spikedrun_ratio$(5)" OF SPIKE"
15200 ELSE
15203 N=4
15206 Use$(5)=0
15209 END IF
15212 DATA 0,0E99,0,0E99,0,0E99,0,0E99,0,0E99
15215 READ Range(*)
15218 MAT Response$= ("??")
15221 Form(Prompt$(5),Response$(5),Use(5),Range(5,2),1,1)
15224 IF E THEN SUBEXIT
15227 Natural_ratio$(1)=UFL(Response$(1))
15230 Natural_ratio$(2)=UFL(Response$(2))
15233 Spike_ratio$(1)=UFL(Response$(3))
15236 Spike_ratio$(2)=UFL(Response$(4))
15239 IF N=5 THEN Spike_ratio(3)=UFL(Response$(5))
15242 FOR I=1 TO 3
15245 IF I<3 OR N=5 THEN Iso_dif(I)=Spikedrun_iso(I)-Spkdrun_ref
15248 NEXT I

88
15131 IF Peak(N)=0 THEN Scan_all=1 ! if no value for this peak, must scan all peaks. Should prevent errors from early Data-exits
15137 NEXT I
15140 Wait=1
15143 FOR I=1 TO N-1
15146 ! increase wait-time if change in isotope is more than 10%
15149 IF ABS(Magnet(Order(I),1)-Magnet(Order(I+1),1))>Magnet(Order(I),1)/10 THEN Wait=2
15152 NEXT I
15155 OUTPUT 6;M$#(M,1)
15159 IF M>M THEN WAIT 6
15161 MAT Peak= <0>
15164 FOR J=0 TO 1
15167 FOR I=(HOT J)<N*J 10 N*(HQT J)*J STEP (NOT J):J ! Step-scan up, down over isotopes
15170 IF Scan_all OR I=1 THEN
15173 60SUB 15518
15176 OH Pr GOIO 15182,15113,15113
15179 END IF
15182 NEXT 1
15185 NEXT J
15188 DISP
15191 Q=(HBX(Peak(*))>10)
15194 IF Q=0 AND M>M AND Daly=1 THEN ! use Daly for data-taking if all peaks <35 mV & Daly-variable=1
15197 Mu=1
15200 OUTPUT 6;M$#(M,1)
15203 PRINT "ON DAILY"
15206 GOTO 15161
15209 END IF
15212 Daly=Mu*(Daly=1)
15215 SUBEXIT
15218 Correct(F$(*),M$(*),Filament(*),Magnet(Order(I),2),M,1,Coarsemag(I),Foc(*))
15221 FOR K=1 TO Wait+1
15224 IF J*(K<=Wait)*(I=N) THEN 15539
15227 Interf_beam(0,M,Order(I),1,Pr)
15230 IF Pr) THEN Rough
15239 NEXT K
15242 IF Peak(I)<0.0001 THEN Peak(I)=0.0001 ! Protect against apparent zero-peaks
15245 RETURN
15248 Escape_rough:Scan_all=0
15251 Escape=1
15254 MAT Peak= <0>
15257 SUBEND ! ---------------------------------------------------------------
15260 /
15263 !
15266 Interfere:SUB Interfere(Q,In,Brkds(*),INTEGER MagposD(*),Number_peaks,Subflag,Dnc_out,Next_run)
15269 OPTION BASE 1
15272 COM /Interfere/ Interfere(*),Interf_non_secs,Interf_non_cts(*),Interf_non_time(*),INTEGER Long(*)
15275 COM /Specs/ Mu(0:1),Ions,2e(0:1),Noise(0:1)
15278 COM /Daly/ INTEGER Mu,Daly,M$#(D=3,2)C8,Daly_ok(D=24),F$#(D=1,2)C4
15281 COM /Filaments/ Filament(*),Fils(*),INTEGER Nfils,Filnum
15284 ! monitor for isobaric interferences
15287 DIM Suns(5)
15290 FOR I=0 TO 19
15293 ON KEY I LABEL "" CALL Clunk
15296 NEXT I
15299 IF Auto*Full_auto THEN ON KEY 0 LABEL "" ;DM"" GOTO Dnc_out
15302 IF Auto*Full_auto THEN ON KEY 1 LABEL "" NEXT RUN"" GOTO Next_run
15305 ON KEY 9 LABEL "" ESCAPE"" GOTO 15698
15251  A-POS(Spikedrun_ratio$,3,"\")
15254  IF Spikedrun_iso(3) THEN Spikedrun_ratio$[3][A,\]="x"
15257  FOO Drive=1 TO 0 STEP -1
15260  ON ERROR GOTO 15275
15263  ASSIGN #Pathl TO "SPIKE:INTERNAL,A,"\$URL$(Drive)
15266  OUTPUT #Pathl,Spikedrun,Spikedrun_ref,Spikedrun_iso(*),Spikedrun_ratio(*),Iso_dif(*),Spike_ratio(*),Natural_ratio(*)
15269  ASSIGN #Pathl TO *
15272  SUBEXIT
15275  NEXT Drive
15278  DISP FMB$("UNABLE TO STORE SPIKE-DATA ON EITHER DISK")
15281  Clunk
15284  WAIT 3
15287  SUBEND !---------------------------------------------------------------
15290  !
15293  !
15311  IF Peak(J)<X THEN NextJ
15314  X=Peak(J)
15317  Peak(J)=Peak(I)
15320  Peak(I)=X
15323  E=Order(I)
15326  Order(I)=Order(J)
15329  Order(J)=E
15332  NEXT J: NEXT I
15335  NEXT I
15338  FOR I=1 TO N
15341  IF Magnet(Order(I),1)=Ref THEN Rp=I ! Rp is order of ref-pk. in Magnet-list
15344  NEXT I
15347  Mip=Order(I) ! Mip is order of Most Intense Peak in Magnet-list
15350  L=Mip
15353  OUTPUT 8 USING "$H04",Magnet(L,1)
15356  SUBEND !---------------------------------------------------------------
15359  !
15362  !
15365  Rough:SUB Rough(Scan_all,Peak(*),Escape,INTEGER Data_dalv,N,Order(*))
15368  !
15371  ! rapid peak-scan to determine rough peak-heights & ratios
15374  !
15377  OPTION BASE 1
15380  COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Fac(*),I_t
15383  COM /Spectra/ Mx(0:1),Ions,2(0:1),Noise(0:1)
15386  COM /Daily/ INTEGER Mu,Daly,Mx(0:3,2),Dalv_ok(0:24),FF(0:1)
15389  COM /Magnet/ Mcoef(*),INTEGER I,Aside,Peak_inter,Coarsemag(0:1),Magnet(Gs24,2),Coarse
15392  COM /Filaments/ filament(*),Fil$(0:1),INTEGER Mflis,F8
15395  INTEGER Pr
15398  FOR I=Auto*Full_auto*2 TO 19
15401  ON KEY I LABEL " CALL Clunk
15404  NEXT I
15407  ON KEY 9 LABEL " ESCAPE" GOTO Escape_rough
15410  OFF XBD
15413  M=M
15416  Subflag=0
15419  Escape=0
15422  Daily_count=0
15425  FOR I=1 TO N
15428  IF Daily_ok(Order(I)) THEN Daily_count=1*Daly_count
15608 INTEGER Pr
15611 Subflag=1
15614 FOR M=1 TO 4
15617 IF (Interfere(M,1)=0) OR Long(M) OR Interf_non_cts(M, U/2+1) THEN 15677
15620 OUTPUT 8 USING "2(4F,4Z):";"OF",Magpas0(M+Number_peaks),"OMV",Interfere(M,1)
15623 MAT Sums= <0>
15626 Zero=(Bkrs(M+M+Number_peaks)+Bkrs(2+M+Number_peaks))/2
15629 LOOP K=-7 TO Interfsecs
15632 Enter_bean(Counts,M,1,Subflag)
15635 OUTPUT 8;"OLCI"
15638 ENTER 8;T
15611 IF Subflag) THEN SUBEXIT
15611 IF K>0 THEN CALL Sums(Sums(*),1,Counts,-ln,0)
15617 DISP "INTERFERENC3-MONITOR PK= ;CHR$(129);Interfere(M,1);CHR$(129);TAB(45);CHR$(128+(K>0));Interfsecs-Max;CHR$(128)"; "; ;GROUND(M,3);"nU"
15650 NEXT K
15653 Interf_non_cts(M, U/2+1)=Sums(M)/Interfsecs-zero ! monitor peak-height in cts/sec
15656 Interf_non_time(M, U/2+1)=Sums(3)/Interfsecs ! average time of non-pk
15659 FOR K=M+1 TO 4 ! share monitor peak data for multiple-interferences
15662 IF Interferes(K,1)=Interfere(M,1) THEN
15665 Interf_non_cts(K, U/2+1)=Interf_non_cts(M, U/2+1)
15668 Interf_non_time(K, U/2+1)=Interf_non_time(M, U/2+1)
15671 END IF
15674 NEXT K
15677 NEXT M
15680 SUBEXIT
15683 Bmc_out: Bmc_out=1
15686 SUBEXIT
15689 Next_run: Next run=1
15692 SUBEXIT
15695 1
15698 Subflag=4*(Auto=0)
15701 SUBEXIT! -------------------------------------------------------------
15704 1
15707 1
15710 Writedata: SUB Writedata(Aver(*),Ratio$(*),Ffile,M$,Signal$(*),Ampl$(*),Acc(*),Pk,Time,Filcurr,INTEGER Block,Run,Samp,Se,Data _collector,Prtr(*))
15713 ! store data for block on disk
15716 PRINTER IS Prtr(2)
15719 Date$=DATE$(TIME(2))
15722 ON ERROR GOTO 15785
15725 OFF KBD
15728 SUSPEND INTERACTIVE i Lockout keyboard during data-write operations
15731 ASSIGN @Pathl TO "LRES:INTERNAL, 4,1"
15734 ENTER @Pathl;Lres ! last file used
15737 Lres=Lres+150*{Lres}>999} ! Increment last file used (max# files is 500)
15740 ASSIGN @Pathl TO "LRES:INTERNAL,4,1"
15743 OUTPUT @Pathl;Lres
15746 IF Block=1 THEN Ffile=Lres
15749 ON ERROR GOTO 15785
15752 ASSIGN @Pathl TO "RESULT:INTERNAL,4,1"
15755 OUTPUT @Pathl,Run,Samp,Date$,M$,Ffile,Lres
15758 ON ERROR GOTO 15794
15761 ASSIGN @Pathl TO "RESULT:INTERNAL,4,1"
15764 OUTPUT @Pathl,Lres,M$,101,Ratio$(*),Aver(*),Signal$(*),Ampl$(*),Acc(*),Pk,M,Data _collector,Block,Time,Filcurr
15767 ASSIGN @Pathl TO *
15770 RESUME INTERACTIVE
15773 OFF ERROR
15776 PRINTER IS CRT
15779 SUBEXIT
15782 !
15785 PRINT USING "2/,10X,5(K),2/";**** DATA-DIRECTORY WRITE-ERROR, RUN#, "Run," BLOCK#, "Block," ****
15788 GOTO 15770
15791 !
15794 PRINT USING "2/,10X,5(K),2/";**** UNABLE TO STORE DATA FOR RUN#, "Run," BLOCK#, "Block," ****
15797 GOTO 15770
15800 SUBEND ! ---------------------------------------------------------------
15803 !
15806 !
15809 Printres:SUB Printres(INTEGER Printer,Run,REAL Escnre,OPTIONAL REAL Filel,File2)
15812 ! print out the isotope-ratio results for a run (get from disk)
15815 OPTION BASE 1
15818 INTEGER Sample,N,Data_collector,Block
15821 REAL Pk,Rec(?),Aver(?)
15824 DIM Ratio(?),Date(?),H,D,Delta(?),Sigma(?),Name_frag(10)
15827 !OUTPUT KB:CHR$(255)ACH$(75)
15830 Db"= 0"
15833 OFF KNOB
15836 ON KEY 9 LABEL "ESCAPE" GOTO 15977
15839 PRINTER IS Printer
15842 IF NPAR<4 THEN
15845 ON ERROR GOTO 15953
15848 ASSIGN @Pathl TO "RES,\INTER\,1"
15851 ENTER @Pathl,Run;Sample,Date$,H#,Firstfile,Lastfile
15854 IF Firstfile=0 THEN Firstfile=Lastfile-10
15857 IF Printer<1 THEN PRINT USING "2/,K,/,!(,2/";"**** RUN ERROR ****"
15860 PRINT USING "25k,2D,?X,1fl,2D,?X,";"DATA SUMMARY FOR BARREL",Sample,"RUN",Run,Date$,H#
15863 IF Printer<1 THEN PRINT
15866 ELSE
15869 Firstfile=Filel
15872 Lastfile=File2
15875 END IF
15878 PRINT "BLOCK":TAB(8);"RATIO":TAB(16);"AVERAGE":TAB(26);"SIGMA":TAB(34);"SIGMA MEAN":TAB(42);"DELTA":TAB(50);"M:REF-PS":TAB(65);"File curr"
15881 PRINT USING "28,K":"Minutes"
15884 IF Printer<1 THEN PRINT
15887 FOR K=Firstfile TO Lastfile*500*(Firstfile)Lastfile
15890 I=K-500*(K)500
15893 ON ERROR GOTO 15962
15896 ASSIGN @Pathl TO "RESULT\INTER\,1"
15899 ENTER @Pathl,Name_frag$,Ratio(?),Aver(?),Sigma(?),Rec(?),Delta(?),Pkn,Data_collector,Block,Time,Filecurr
15902 IF Block=0 THEN
15905 PRINT USING "2/,26K,K,2/";**** RUN ABORTED ****
15908 ELSE
15911 FOR J=1 TO N-1
15914 IF J=1 THEN PRINT TAB(2);(VAL$(Block)0@0)1*DATA Collector;
15917 PRINT TAB(7):Ratio(?);TAB(15):ROUND(Aver(J),6);TAB(26):Sigma(?);TAB(34):ROUND(Rec(J),3);TAB(42):Delta(?);TAB(50):Pkn;
15920 IF J=1 THEN PRINT TAB(55):ROUND(Pk,3);TAB(64):Filecurr;TAB(73):ROUND(Price,4);
15923 PRINT
15926 NEXT J
15929 IF Printer<1 THEN PRINT
15932 END IF
15935 NEXT K
15939 IF Printer<1 THEN PRINT USING ";/K,/,K,5/";RP$("K",40),RP$("K",40)
15941 IF Printer<1 THEN PRINT
15944 PRINTER IS CRT
15947 OFF ERROR
15950 SUEXIT
15953 PRINTER IS CRT
15956 PRINT USING ";,3(K),";**** UNABLE TO READ DIRECTORY FOR RUN, "Run," ***
TINECONST_Corr: SUB TINECONST_Corr(Corr(*), REAL Resdecay(*), Raw_peaks(*), INTEGER H, Integr_time(*), Delay_time(*))

1 DO corrections for resistor time-constants, to depth of 3 sets

1 OPTION BASE 1

1 REAL Rvpk(8), Peak(30), Itime(8)

1 MAT Rvpk = RSUM(Raw_peaks) ! average peak-heights for each isotope

1 MAT Itime = Integr_time + Delay_time ! total peak-time for each isotope

1 Total=SUM(Itime) ! set time

1 I=0

1 FOR I=1 TO N ! array of peak-heights for each second, 9 sets deep

1 FOR J=1 TO Itime(I)

1 I=I+1

1 P=Rvpk(I)

1 Peak(I)=P

1 Peak(I+1)=P

1 Peak(I+2)=P

1 IF I+3<301 THEN Peak(I+3)=P

1 NEXT J

1 NEXT I

1 Ct=0

1 FOR I=1 TO H

1 Reach=Ct+1 ! first donor-peak (1st sec. of Ith-pk 2-sets-prior)

1 TINECONST_Corr=0

1 FOR J=1 TO Itime(I)

1 Ct=Ct+1

1 IF J>Delay_time(I) THEN

1 Last=Ct+3*Total-1 ! last donor-peak (previous second)

1 FOR T=Last TO Reach STEP -1

1 Z=Last-T+1

1 IF (Resdecay(Z)=0) OR (Z=80) THEN 16085

1 TINECONST_Corr=TINECONST_Corr+Peak(T)*Resdecay(Z) ! effect of donor peak

1 NEXT T

1 END IF

1 NEXT J

1 NEXT I

1 Resdecay=SUB Resdecay(REAL Resistor_const(*), Resdecay(*)) ! calculate 80 seconds of resistor-decay effects

1 FOR I=2 TO 6 STEP 2

1 IF Resistor_const(I)=0 THEN Resistor_const(I)=1

1 NEXT I

1 FOR I=1 TO 80

1 T=I-.5

1 Resdecay(I)=Resistor_const(I)*EXP(-T/Resistor_const(2))+Resistor_const(3)*EXP(-T/Resistor_const(4))+Resistor_const(5)*EXP(-T/Resistor_const(6))

1 FOR I=1 TO 3 THEN 16128

1 IF Resdecay(I)>1.E-8 THEN SUBEXIT ! ignore DZC tails of <.01 ppm

1 NEXT I

1 SUBEND !---------------------
Print using "K,2D,8X,K,10":"Coarse-Magnet Range: ",Coarse,"Re-107 Magnet-setting: ",Magnet(0,2)

FOR I=0 TO 12 STEP 12
FOR J=1+J TO 12+J
    IF DNiso THEN 16163
    PRINT TRB(6*(J-JH);Magnet(I,1));
    NEXT I
    PRINT
    NEXT J
16163 PRINT
16166 FOR I=+I TO 12+J
    IF I>Niso THEN 16194
    PRINT TRB(6*(I-J)-2);Magnet(I,2);
    NEXT I
    PRINT
16194 NEXT J
16197 SUBEND !-----------------------------------------------------------
16198 !
16199 !
16200 Decay=SUB Decay(Reduced_current,Flashed,Minbeam,Maxgrowth,Abort,INTEGER Block)
16201 !
16202 ! monitor decay/growth of beam, flash if noisy.
16203 OPTION BASE 1
16206 DIM /General/ 2#,INTEGER Prtr(*),Subflag,AutoFull_auto,Foc(*),I_t
16208 DIM /Specs/ Mu(0:1),lons,ze(0:1),Noise(0:1)
16211 DIM /Daly/ INTEGER Mu,Daly,Mn#(0:3,2),Daly_ok(0:24),Ff#(0:1,2)(43)
16214 DIM /Magnet/ Mcoef(*),INTEGER L,Ms,Peak_inter,CoarseMag(0:1),Magnet(0:24,2),Coarse
16217 DIM /Filaments/ Filament(*),Fil#(*),INTEGER Nfil,F8
16219 DIM Suns(*)
16222 !
16223 ON KEY 9 IFiBEL " ESCFlPE" GOTO 16106
16225 !
16226 ! INTEGER Pr
16229 FOR I=Auto*Full_auto*2 TO 19
16232 ON KEY I LABEL " CALL Clunk"
16235 NEXT I
16238 ON KEY 9 LABEL "   ESCFlPE" GOTO 16106
16240 !
16244 !
16249 !
16252 !
16253 !
16256 !
16259 IF Block=0 THEN Reduced_current=0
16262 TO=TIMEOflTE
16265 Mf(5)= (0)
16268 FOR Y=0 TO Nsec
16271 Enter_beam(X,Mu,l,1,Pr)
16274 IF Mu>2 AND Y AND Mu/Last_nv/2 THEN Mu=Last_nw ! crude dropout protection
16277 Last_nv=Mu
16280 DISP "CHECKING BEFlM NOISE AND DECAY ";Msec-Y,DROUND(Mu,4);" Mv"
16283 IF Pr>1 THEN Decay
16286 IF Y THEN
16289 CALL Suns(Suns(*),1,Mu,l,10)
16292 ENDF
16295 NEXT Y
16298 CALL Slope(Sums(*),Slope,Int,Scatter_nv,Slope_s,Int,Msec)
16301 Ru_beam=Sums(l)/Msec ! average beam in millivolts
16304 Decay=Slope*60*100/Ru_beam ! change change in X/minute
16307 Decay_s=Slope*60*100/Ru_beam ! uncertainty in X decay
16310 Noise_percent=100*Scatter_nv/Ru_beam
16313 PRINT USING ";,3(K),8K,3(K),4A,";"BEAFl-CHFlG= ",DROUND(Decay,2),"X/MIN.","NOISE= ":DROUND(Noise_percent,2)," X/SEC."
Theoretical \( \text{noise} = \text{SQR}(\text{Noise}(\text{Mu}^2 + 2 \cdot \text{Av_beam} \cdot \text{Ions}) \times (\text{Scatter}_\text{mu} \times 2.5 \times \text{Theor_noise}) \) \)

beam is defined as noisy if \( \text{noise} > 0.5 \times \text{second noise} \) and \( \text{theoretical noise} \)

\[ \text{If NOT Noisy AND (Decay_copy > Decay_sig)} \rightarrow \text{THEN Check growth} \]

\[ \text{beam either noisy or decaying rapidly} \]

\[ \text{T} = \frac{(\text{TIME}_\text{TRIP} - 60)}{60} \]

\[ \text{PRINT USING} \quad "3(K),/"; "(\text{decay or noise exceeds tolerance): }, \text{TAbort} \times 2 \text{ Minutes to abort or flash)}" \]

\[ \text{IF TAbort THEN 16265} \]

\[ \text{IF Noisy AND NOT Flashed THEN Flash} \]

\[ \text{Error_message(Prtr(*),3,"CANT PROCEED WITH RUN - "}) \]

\[ \text{Subflag}=1 \]

\[ \text{GOTO 16109} \]

\[ \text{Flash:} \]

\[ \text{Error_message(Prtr(*),3,"NOISY BEAM "}) \]

\[ \text{FLASHING TO "}) \text{INT(1150}\cdot\text{Filament(Nfils)/1000)}" \text{ AMPS}" \]

\[ \text{Flashed}=1 \]

\[ \text{CALL Filament(50,1.15,\text{Filament(Nfils)},1,\text{Nfils})} \]

\[ \text{increase sample-filament current by 15%} \]

\[ \text{Wait(TIME}_\text{TRIP},120,\text{Filament(Nfils)},\text{Magnet(L,1),Auto,Full_auto)} \]

\[ \text{wait 2 minutes after flashing} \]

\[ \text{CALL Filament(10,\text{Filament(Nfils)},1.15,\text{Nfils})} \]

\[ \text{restore original sample-filament current} \]

\[ \text{GOTO 16262} \]

\[ \text{Check_growth:} \quad \text{rate of beam growth too large?} \]

\[ \text{IF Block=0 RND Reduced_current=0 AND (Av_beam-Minbeam)>(Growth\times2\times\text{Maxgrowth}) OR (Av_beam-Minbeam)>(Growth\times\text{Maxgrowth})} \]

\[ \text{reduce fil-curr if growthMAX & beam\text{MIN, OR IF growth}=\text{MAX} \& \text{beam\text{MIN}} \quad \text{(before 1st data-block only)} \]

\[ \text{Reduce(Reduced_current,\text{Maxgrowth,Filament(Nfils)},1,\text{Nfils})} \]

\[ \text{beam-growth too large} \]

\[ \text{Wait(TIME}_\text{TRIP},180,\text{Filament(Nfils)},\text{Magnet(L,1),Auto,Full_auto)} \]

\[ \text{ON KEY 9 LABEL " ESCAPE" GOTO 16106} \]

\[ \text{END IF} \]

\[ \text{END IF} \]

\[ \text{OUTPUT KBD;CHR$(255);CHR$(75);} \]

\[ \text{Peaks:} \]

\[ \text{SUB Peaks(}R, \text{Peak_height}, \text{Set,}Tn, \text{Peak_height}, \text{Prax,Ruer(*)}, \text{INTEGER H,CoarseMag,Mag_pos(*)}, \text{Integer_time(*)}, \text{Wait_time(*)}, \text{Isotope(*)}, \text{Dropouts}) \]

\[ \text{OPTION Base 1} \]

\[ \text{ This is the actual peaktop-jumping routine for data-taking, & includes real-time graphics & ratio-calculation} \]

\[ \text{GOTO 16106} \]

\[ \text{GOTO 16262} \]

\[ \text{COM /General/ Z$, INTEGER Prtr(*),Subflag,Auto, Full_auto,Foc(*),I_t} \]

\[ \text{COM /Keyboard/ Cn$,C*,C#,*},\text{Clear}$ \]

\[ \text{COM /Specs/ Hx,B:l>,Ions,Ze(0:l),Noise(0:l)} \]

\[ \text{COM /Baly/ INTEGER Mu,Daly,Mn$(0:3,2)[8],Daly_ok(0:2),FF$(0:1,2)[4]} \]

\[ \text{COM /Filaments/ Filament(1),Fil$(1)El03, INTEGER Nfils,F8} \]

\[ \text{COM /Oatal/ Peak(*),Pk, Interfered(*), Normal(2),100, INTEGER Data_iso(*),Data_collector,Inv,Bnc_out,Next_run,Spoke} \]

\[ \text{COM /Data/ Bkrd(*),Bkrd_ga(*),Resdecay(*),INTEGER Sample,H_prime,Msets,Ref,Rf,Bkrd_rdgs(*),Bkrd_signal(*)} \]

\[ \text{GOTO 16106} \]

\[ \text{GOTO 1606} \]

\[ \text{GOTO 16106} \]

\[ \text{DIM Brat(7,38),Sums(5),Rsums(5),Tsums(5),Asums(7,5),Last_set(2,8),Decay(222)} \]

\[ \text{Decay}="\text{beam}" \]
FOR I=0 TO 19
    IF Auto*Full_auto THEN ON KEY 0 LABEL " BMC" GOTO BMC_out
    IF Auto*Full_auto THEN ON KEY 1 LABEL " NEXT RUN" GOTO Next_run
    ON KEY 9 LABEL " ESCAPE" GOTO Operator_exit
    OFF KBD
    MAT M=0
    MAT Rsun=0
    Dropouts=0 ! # of GPIO-interface dropouts caught
    Dot_interval=9
    PRINTER IS CRY
    Spac=2*(N+8)\+\4*(N/5)
    MAT Peak_height=0
    MOVE 0,0
    OUTPUT 8 USING "AR,42","0DF",Mag_pos(I)
    WAIT 1! allow time for magnet to switch from far-away bkd positions
    Normal_delta=0
    J=0
    MAT Rsun=0
    Ref_bkrd=(Bkrd(1,8)+Bkrd(2,8))/2 ! reference-peak background
    FOR I=1 TO N
        IF I<3 THEN Ratio=(Ref/I)/(SUM(Isotope(I)))
        IF I=3 THEN Ratio=FNInvert(Ratio)
        PRINT Cu;TFLXY((13-Spac)*(I-(I>R))-(10-Spac),17);Raho$;Cu$
        END IF
        NEXT I
    FOR Set=1 TO Nsets
        IF Set<1 THEN ON KEY 1 LABEL " CUT SHORT" GOTO Cutshort
        FOR I=1 TO N
            MAT Sums=0
            FOR Iz=1 TO 5
                Rsun(Iz)=Rsun(I-(I>R),Iz)
                NEXT Iz
            NEXT I
            OUTPUT 8 USING "2(1fl,1Z)""WJ",Tag_pos(D,Isotope(D ! switch magnet to peak-top
            Enter_beam(Counts,ftu,1,1,Subflag,0,Dropouts)
            IF Subflag!=1 THEN SUBEXIT
            OUTPUT 8;"TCI"
            ENTER 8:It
            IF Last_time AND It\(Last_time THEN ! Guard against GPIO dropouts - don't accept any times less than previous time
            FOR Ct=1 TO 3
                BEEP 1500;200;Ct,0.04
                NEXT Ct
            Bad_time=Ct+Bad_time
            Dropouts=1+Dropouts
            IF Bad_time<2 THEN 16606
            END IF
            Last_time=It
            It=It-in
        Bad_time=0
        Otherpk_bkrd=INT((Bkrd(1,8)+Bkrd(2,8))/2) ! non-ref pk backgrd
        L=Wat_tine(I)*1.1*(Wat_tine(I))*Integ_tine(I)-J
        Mu_peak=(Counts*Otherpk_bkrd)/Mx(Mu) ! peak-height in millivolts
        IF J=Wat_tine(I) AND Mu_peak<Last_nv/2 AND Last_nv>(0.6*Mu) THEN ! Guard against GPIO dropouts, trigger being drop of 2x in beamsize
            BEEP 1500,0.05
            BEEP 1300,0.05
        ELSE
16667    Bad_nv=1*Bad_nv
16670    Dropouts=1*Dropouts
16673    IF Bad_nv<3 THEN 16604
16676    END IF
16679    Last_nv=Mv_peak
16682    Bad_nv=0
16685    line=t/6.E+3
16688    IF J=Integrate_time(I)*Wait_time(I) THEN  last count on peak
16691    Last_set(I,I)=Line
16694    Last_set(2,I)=Mv_peak
16697    END IF
16700    IF J<2 THEN DRAW Time,Mv_peak
16703    IF J>2 THEN
16706    IF Set=1 THEN  connect to last pk with dotted line
16709    MOVE Last_set(I,I),Last_set(2,I)
16712    LINE TYPE 4, Dot_interval
16715    DRAW Time,Mv_peak
16718    LINE TYPE 1
16721    ELSE
16724    MOVE Time,Mv_peak
16727    END IF
16730    END IF
16733    Display C#:Isotope(I);Ch#:TAB(15);"SE#":TAB(19);Set:TAB(27);DRAW@Mv_peak,6-3*Mv:"Mv":TAB(45);CH#:128+(J)*Wait_time(I)
16736    IF J=Wait_time(I) THEN CALL Suns(Sums(*),I,(Counts),Tt,0)
16739    NEXT J
16742    BEEP 500,.05
16745    Int_time=Integrate_time(I)
16748    Peak_height(I,Set)=Sums(3)/Int_time ! time of Ith peak for Kth block
16751    IF Int_time<3 THEN  use average pk-height if <3 seconds on peak
16754    Peak_height(I,Set)=Sums(3)/Int_time
16757    ELSE
16760    CALL Slopes(Sums(*),Slope,Inter,Scatter_cnts,Scatter_ints,Int_time)
16763    ! linear regression of pktop counts with time
16766    Peak_height(I,Set)=Slope*Peak_t(I,Set)+Inter ! use mid-value of linear regression with time for peak-height if )=3 seconds on peak
16769    END IF
16772    IF Peak_height(I,Set)<Otherpk_bkrd*(K>R)-Ref_bkrd>(I=R) THEN CALL Correct(Fs(*),Mn(*),Filament(*),Mag_pos(I),Mu,1,Co_arsenq,Foc(*))
16775    ! restore focus if a negative peak-height (to correct for possible arc)
16778    IF (Set)=1 AND (I=0) THEN! calculate real-time ratio
16780    L=Set+(R<I)-1 ! Lth pair of ratio-isotopes
16784    RF1=Peak_height(I,L)-Ref_bkrd I net cps for reference-peak
16787    Temp1=(Peak_t(I,L)-Peak_t(I,Set-1))
16790    Temp2=(Peak_height(I,Set)-Peak_height(I,Set-1))/(Peak_t(I,Set)-Peak_t(I,Set-1))
16793    Temp3=Peak_height(I,Set-1)+Temp1*Temp2*Otherpk_bkrd! interpolated non-ref. peak net-cps
16796    IF Temp3=0 THEN Temp3=1.E-12! Guard against div. by 0
16799    RF2=RF1/Temp3 ! raw ratio (uncorrected for fractionation)
16802    IF NOT Spike AND Normal(I) THEN Rr=RF2/(I)<Isotope(I)>Normal(I)>Normal_delta*(Ref-Isotope(I))/(Ref-Normal(I)))! real-time ratio
16805    ! correct for mass-discrimination (linear)
16808    IF Normal_delta OR Normal(I)=0 OR Spike OR Isotope(I)=Normal(I) THEN
16811    ! calculate & display real-time ratios
16814    Ix=I-1*(Ix)! ratio#
16817    Mrat(Ix)=1/Mrat(Ix) ! Ratios taken for Ith ratio
16820    Drat(Ix,Mrat(Ix))=RF-Inul ratio
16823    CALL Suns(Sums(*),1,0,Drat(Ix),Mrat(Ix)),Mrat(Ix))
16826    Aver(Ix)=Rsums(3)*Mrat(Ix) I no-outliers average
16829    IF Mrat(Ix)=1 THEN
16832    Calc_sigmat(Sigma,Rsums(3),Rsums(4),Mrat(Ix))
16835 IF Nrat(Ix) > 1 THEN ! reject outliers at 1.7-sigma level
16838 HUT Tsuns = Rsuns
16841 Nratios = Nrat(Ix)
16844 FOR Iz = 1 TO Nrat(Ix)
16847 IF ABS(Drat(Ix, Iz) - Aver(Ix)) > 1.7*Sigma THEN CALL Sums(Izsums(*), -1, 0, Drat(Ix, Iz), Nratios)
16850 NEXT Iz
16853 Aver(Ix) = Tsuns(Izsums(Ix))/Nratios
16856 Calc_sigma(Sigma, Tsuns(Ix), Tsuns(Iz), Nratios)
16859 END IF
16862 END IF
16865 IF NOT Spike AND Isotope(I) = Normal(I) THEN Normal_delta = Aver(Ix) / Inv_Normal(2) - 1 ! difference w. normalizing ratio
16868 PRINT TABXY(Ix*(13-Spac)-10-Spac)-2*(Hx(5)); VAL$(ORD Szum(Aver(Ix), 5));
16871 IF MS THEN ! include std-deviations in realtime ratio-printout
16874 IF Nrat(Ix) > 1 THEN PRINT "-" + VAL$(ORD Szum(10 + Sigma/Aver(Ix), 2)) & 
16877 ELSE
16878 PRINT " "
16880 END IF
16883 END IF
16886 FOR Iz = 1 TO 5
16889 Asums(Ix, Iz) = Rsuns(Iz)
16892 NEXT Iz
16895 END IF
16898 END IF
16901 IF Set > 6 THEN ! display beam-growth/decay (if room)
16904 PK1 = (Peak_height(I, Set) - Other_pk) * Mx(Mu)
16907 PK0 = (Peak_height(I, Set) - Other_pk) / Mx(Mu)
16910 IF PK1 > 9 * Mu THEN
16911 G1 = 1 + 60 * (PK1 * PK0 - 1) / (Peak_t(I, Set) - Peak_t(I, Set - 1))
16916 IF ABS(G1) < 0.1 THEN G = 0
16919 IF G > 0 THEN Decay$ = "growth"
16922 IF G < 0 THEN Decay$ = "decay"
16925 Decay$ = VAL$(ABS(ORD Szum(G, 2))) & 
16928 PRINT TABXY(58, 18): Decay$
16931 END IF
16934 END IF
16937 NEXT I
16940 NEXT Set
16943 SUBEXIT
16946 |
16949 Inc_out = Inc_out + 1
16952 SUBEXIT
16955 Next_run = Next_run + 1
16958 SUBEXIT
16961 |
16964 Cutshort: ! k4 key pressed: stop taking data & go on to bkrs
16967 Set = Set - 1
16970 Nsets = Set
16973 Bkrop
16976 SUBEXIT
16979 |
16982 Operator_exit: Subflag = 1
16985 SUBEND |---------------------------------------------------------------------------------------------------------------------
16988 |
16991 !
16994 Calcavg: SUB Calcavg(Ratio(*), Time(*), Print(*), R#, Percent_error(*), Cndump, INTEGER M, Prtr(*))
16997 OPTION BASE 1
17000 ! calculate weighted averages of ratios in a block
17003 DIM Inverse_var(80)
17006 Nset = 9
17009 Count = 0
17012 Recalc: Ext sigma = 0
17015 Count=Count+1
17018 Weight=0
17021 Sum_wtd_ratios=0
17024 Q=0
17027 FOR I=1 TO N
17030 IF Ratio(I) THEN
17033 Inverse_var(I)=(100/(Percent_error(I)*Ratio(I)))^2
17036 Weight=Weight+Inverse_var(I)
17039 Sum_wtd_ratios=Sum_wtd_ratios+Inverse_var(I)*Ratio(I)
17042 Q=Q+Inverse_var(I)*Ratio(I)^2
17045 END IF
17049 NEXT I
17051 Nu=Nn-1 ! degrees of freedom
17054 I=NST(Nn)
17057 Wtd_aver_int=Sum_wtd_ratios/Weight ! "internal" error of wtd average
17060 Wtd_aver=Wtd_aver_int
17063 Sums=Q-Weight*Wtd_aver_int^2 ! sums of squares of weighted deviates
17066 IF Sums=0 THEN Sums=0
17069 Msud=Sums/Nu ! mean square of weighted deviates
17072 Percent_int_sig=100/Wtd_aver_int/SQR(Weight)
17075 Percent_totsig=Percent_int_sig*SQR(Msud)
17078 IF Msud<0 THEN
17081 Probability=0
17084 ELSE
17087 !
17090 Q=1 ! calculate the probability that internal errors alone will give the observed scatter, using cumulative chi-square distribution
17093 R=1
17096 IF NOT NF(Nu) THEN
17099 I=0 ! nu odd
17102 REPEAT
17105 I=I+2
17108 D=D*Sums/(Nu-I)
17111 A=A+D
17114 UNTIL A<1.E-6
17117 D=SQR(Pi)
17120 FOR I=1/2 TO Nu/2
17123 D=D*I
17126 NEXT I
17129 Probability=1-EXP(-Sums/2)*(Sums/2)^(Nu/2)*A/D
17132 ELSE
17135 !
17138 FOR I=1/2 TO Nu/2-1! Nu even
17141 D=D*Sums/(2*I)
17144 A=A+D
17147 NEXT I
17150 Probability=A*EXP(-Sums/2)
17153 END IF
17156 END IF
17159 !
17162 Percent_err_95=Percent_int_sig*(Nn<=6)*1.5*SQR(Sums/Nu)+(Nn(6)*1.96)
17165 !
17168 IF Probability<.2 THEN ! calculate excess variance (Troutman's equation)
17171 Ext_sigsq=SQR((Percent_totsig)^2-Percent_int_sig^2)*Nu
17174 Wtd_aver_ext=Wtd_aver_int! wtd average taking into account external variance
17177 G=0
17180 MD=0
17183 SI=0
17186 !
17189 LOOP
ON ERROR GOTO No_convergence
MD=100
DATA 0,0,0,0,0,0,0
RESTORE 17198
READ M,K,Q,R,0,E,L
FOR I=1 TO N
  IF Ratio(I) THEN
    R=6+1/Inverse_var(I)
    L=(Ratio(I)-Utd_aver_ext)^2
    K=K+Ratio(I)/R
    M=M+1/R
    Q=Q+L/R^2
    E=E+L/R^3
  END IF
  FOR N=1 TO N
    S1=6-(Q-M)/(Q-2*E)
    Utd_aver_ext=K/S1
    OISP "iteration":NO:" mean":S1:" ext. var.";S1,3)
    EXIT IF (ABS(S1-SI)/(1,E-4) AND (S1)=0)
  ELSE 3D THEN ! if hasn't converged within 30 iterations, test for convergence
    IF (ABS(S1-SI)>=2) OR (SI=0) THEN
      No_convergence:GFF ERROR
    END IF
    Percent_err_95=SQR((1.96*Percent_int_sig)*2+1/2*(Percent_intsig)^2)
    GOTO Reject
  END IF
  END IF
  ZO=flBS(S1-6)
  G=S1
  LOOP!
  Ext_5igMa=100*SQR((N*Hn/Nu)/Utd_aver_int
  Percent_errJ5=SQR((1.96*Percent_int_sig)^2+(Ext_5igMa)^2/Nn)
  Utd_aver=Utd_aver_ext
  !
  Reject:HO=Nn ! reject outliers
  FOR I=1 TO N
    IF Ratio(I) AND Nn>.?*N THEN ! don't reject more than 30% of ratios
      Tolerance=I*(Count-1)/8)*SQR((1.1*Ratio(I)*Percent_error(I)))^2+0.01*Utd_aver*Ext_5igMa
      IF Rej_print=0 THEN
        IF Rej_print=1 THEN
          PRINT "REJECTED: ";
        END IF
      ELSE 3D THEN ! recalculate utd average
        IF Nu<HO THEN
          Nu=1
          AISP
        END IF
        FOR I=1 TO N
          IF Nu<HO THEN
            AISP
          END IF
        NEXT I
        IF Nu<HO THEN recalculate utd average
      END IF
      DISP
17369 PRINT
17372 GOTO Recalc
17375 END IF
17378 END IF
17381
17384 Abs_err_95*GROUND(.01*Percent_err_95*Utd_aver,2)
17387 IF (Abs_err_95<0) OR (Utd_aver<0) THEN 0=6
17390 IF (Abs_err_95<0) AND (Utd_aver>0) THEN 0=1 INT(INT(INT(Abs_err_95*(Abs_err_95=0)/Utd_aver)))
17393 FOR Y=0 TO 10 PRINT
17396 PRINTER IS PrtrROM
17402 IF Y=0 THEN PRINT TAB(1,15)
17405 IF Y THEN PRINT RTPH("","00")
17408 PRINT USING "3(X),K","UTD OVERAGE = ",DROUND(Utd_aver,0)," +/- 
17411 IF Y THEN PRINT RTPH("","00")
17414 IF Nu=0 THEN SUBEXIT
17417 PRINT USING "3(X),10X,3(K)";"EST. TOTAL SIGMA = ",DROUND(Percent_int_sig,2),"/;"EST. TOTAL SIGMA = ",DROUND(Percent_int_sig,2),"/;
17420 IF Ext_signa THEN PRINT "EXTERNAL SIGMA = ",DROUND(Ext_signa,2)
17423 PRINT "M.S.D." =":DROUND(Msd,3):TAB(37):"PROBABILITY =":DROUND(Probability(Probability).001),2
17426 IF Y THEN PRINT USING "00A,2","RPIH(">",80)
17429 NEXT Y
17432 PRINTER IS CRT
17435 Tspred=Time(N)-Time(l)
17438 MOVE Time(l)-1.2-Tspred/10,Utd_aver
17441 DRAW Time(N)+1.2-Tspred/10,Utd_aver
17444 SUBEND !
17447 !
17450 !
17453 Peakint=SUB Peakint(REAL Peak(*),Noise(*),Ions,INTEGER Integr_tine(*),Wait_time(*),N,Collector,Refpk,Isotope(*))
17456 ! calculate optimum (in the sense of precision in least time for a perfectly-stable beam) integration & wait times for each peak,
17459 ! using the algorithms of Ludwig, USGS Prof. Paper ????
17462 OPTION BASE 1
17465 REAL Optim_tineratio(0),Integr_tine(0)
17468 MAT Optim_tineratio= (0)
17471 MAT Integr_tine= (0)
17474 Mu_av_pk=.001*SUM(Peak)/N ! average peak-height in volts
17477 A=1+99*Collector
17480 Au_integr_tine=A*(Mu_av_pk(0/4)+(Mu_av_pk(1/8)+(Mu_av_pk(.1/8)+(Mu_av_pk(.01/8)+(Mu_av_pk(.001/8)
17483 !
17486 FOR I=1 TO N ! theoretically optimum ratio of integration-times
17489 Optim_tineratio(I)=SQR(((Noise(Collector)/Peak(I))**2+1/Ions/Peak(I))/((Noise(Collector)/Peak(Refpk))**2+1/Ions/Peak(Refpk)))
17492 NEXT I
17495 !
17498 ! don't nest this loop within above loop!!! must have all Optim_tineratio!!
17501 FOR I=1 TO N ! theoretically optimum integration times
17504 Integr_tine(I)=Optim_tineratio(I)*Au_integr_tine*N/SUM(Optim_tineratio)
17507 NEXT I
17510 IF Integr_tine(I)<2 THEN
17513 IF N=4 AND Optim_tineratio(N)=5 THEN Au_integr_tine=1.5*Au_integr_tine
17516 END IF
17519 !
17522 ! for 3 isotopes, Least-Intense Peak IT no longer than next-LIP
17525 ! for 3 isotopes, LIP II no more than 2.1 times next-LIP
17528 IF N=3 THEN Integr_tine(N)=Integr_tine(N-1)
17531 IF (N=3) AND (Integr_tine(3)<2.1*Integr_tine(2)) THEN Integr_tine(3)=2.1*Integr_tine(2)
17534 ! renormalize for average integration time
17537 MAT Integr_tine= Integr_tine*N/SUM(Integr_tine)
17540 !
17549 !
FOR I=1 TO N
17546  Integr_time(I)=INT(Integr_time0(I)>(FRAC(Integr_time0(I)))=.5)
17549  IF Integr_time(I)>=2 THEN Integr_time(I)=2
17552  IF Integr_time(I)>=4 THEN Integr_time(I)=4
17555  NEXT I
17558  
17561  FOR I=1 TO N
17564  Peakratio=ABS(Peak(I)+W(I))/Peak(I))
17567  Mass_jump=ABS((Isotope(I)+W(I))/Isotope(I))
17570  Wait_time(I)=1+(Peakratio2.5)*(Peakratio2)+(Peakratio100)+(Peakratio500)+(Peakratio2000)
17573  IF (Mass_jump.15) AND (Wait_time(I)<3) THEN Wait_time(I)=3 ! wait at least 3 sec for large mass-jumps
17576  NEXT I
17579  SUDENT I
17582  
17585  
17588  Hu=SUB Hu(M]()%Hu,Pass,INTEGER Mu,I_t) ! query accelerating voltage
17591  OFF KEY
17594  OFF KNOB
17597  ON KEY 9 LABEL "ESCAPE" GOTO 17600
17600  Pass=0
17603  U=+M(Mu[I])
17606  PRINT "HV =";
17609  !
17612  ! read zero
17615  OUTPUT 8;"$QW:"
17618  FOR I=1 TO 10
17621  WAIT .2
17624  NEXT I
17627  OUTPUT 8;"$QV"
17630  ENTER 0;Zero
17633  !
17636  ! read gross value
17639  OUTPUT 8;"$QW:"
17642  FOR I=1 TO 10
17645  WAIT .2
17648  NEXT I
17651  OUTPUT 8;"$QV"
17654  ENTER 0;0
17657  !
17660  Hu=INT(U-Zero)/100
17663  OUTPUT 8;$QW:Mu[I_t] ! reset system monitor to ion-collector
17666  IF Hu/I.E+4 THEN ! impossible value
17669  Clunk
17672  DISP "GAIN MUST BE SET ON X1 -PRESS CONTINUE WHEN READY"
17675  PAUSE
17678  GOTO 17606
17681  END IF
17684  PRINT Hu:"VOLTS"
17687  SUBEXIT
17690  Pass=1
17693  SUDENT I
17696  
17699  
17702 Spike_check:SUB Spike_check(Spike,INTEGER Iso(*),Spikedrun_iso(*),Spkdrun_ref) ! check if the run-variables specify the isotopes required by the spike
17705  FOR I=1 TO 8
17708  Q=Iso(I)
17711  IF Q THEN ! check only for nonzero isotopes
17714  IF <Q>Spikedrun_iso(2)> AND <Q>Spikedrun_iso(2)> AND <Q>Spikedrun_iso(2)> AND <Q>Spikedrun_ref THEN Spike=0
17717  IF Q=Spkdrun_ref THEN Spkdrun_ref<1 ref. isotope is present
17720 IF (Q=Spikedrun_iso(D) OR (Q=Spikedrun_iso(2)) THEN Norniso50=l+Norniso! 8 of normalizing isotopes present
17723 END IF
17726 NEXT I
17729 IF (Spkdrun_refiso=0) OR (Norniso<2) IHEN Spike=0
17732 SUBEND ! -----------------------------
17735 !
17738 !
17741 Wait:SUB Wait(Tzero,Waitsec,Filcurr,INTEGER Isotope,Fiuto,Full_auto) ! wait for Waitsec seconds
17744 FOR I=2*Fiuto*Full_auto 10 19
17747 ON KEY I LABEL "" CALL Clunk
17750 NEXT I
17753 ON KEY 3 LABEL " HALVE WAIT" GOTO Halve
17756 ON KEY 4 LABEL " DOUBLE WAIT" GOTO Double
17759 ON KEY 9 LABEL " ESCAPE" GOTO 17831
17762 ON TIME (Tzer0+Waitsec) MOD 86400 GOTO 17834
17765 !
17768 Iso$="("aUfiL$(l5otope)&")"
17771 IF Isotope=0 THEN Iso$=""
17777 T$="SECOHDS"
17780 Divide=1
17783 ELSE
17786 T$="MINUTES"
17789 Divide=60
17792 END IF
17795 IMAGE "WAITING",4,40.20,2X,7A,BX,K,"%6S","10K,K
17798 T=Waitsec-TIMEDATE+Tzero
17801 IF T<0 THEN 17831
17804 DISP USING 17795;T/Divide,"#",/Filcurr,Isos$
17807 WAIT .1
17810 GOTO 17798
17813 |
17816 Waitsec=Waitsec/2
17819 GOTO 17762
17822 Double=Waitsec=Waitsec*2
17825 GOTO 17762
17828 |
17831 Tzero=TIMEDATE
17834 DISP
17837 SUBEND ! -----------------------------
17840 !
17843 !
17846 Ratiocalc:SUB Ratiocalc(I,IO,Calc_inverted,R,Ralcy_discr,Ratio_count,Pb1478_cycle,Normal(*),Ns,Decay,INTEGER Nsets,Pb_4678,Ref,
17849 INU,M,Ratio_isotope(*))
17852 OPTION BASE 1
17855 CNM /Data2/Acc(*),Lacc(*),SingaS(*),DeltaS(*),Aver(*),Last_aever(*),Last_ratioS(*),RatiO(*)
17858 CNM /sloa,Slope,Int,Spike,Sig,Pee_k_t(*),finecon_pkcorr(*),Peak_height(*),Normrat_slope,Normrat_inter,Norm_inverted
17861 INTEGER NratiO_used
17864 OFF KEY
17867 OFF KBD
17870 H=KNDBX
17873 Printsets=0 ! don't printout the individual set-ratios
17876 Ratio$=URL$(Ref$,"$URL$(Ratio_isotope(I))
17879 IF Inv=-1 THEN Ratio$=FNInvert$(Ratio$)
17882 P=(R(0)
17885 H=P*(NOT P)*I0
17888 B=P*I0*(NOT P)*R
17891 FOR J=1 TO Nsets-l ! Dodson's interpolation algorithm
17894 T=(Peak_t(A,J)+Peak_t(R,J)+Peak_t(B,J)+Peak_t(B+,J))/4 ! time of interpolation
17897 Ratio$=FNInvert$(Ratio$)
17900 END FOR
17897 \[ \text{Pa} = \text{Peak_height}(R,J) \times (\text{I-Peak_t}(R,J)_{i+1} - \text{Peak_t}(R,J)) / (\text{Peak_t}(R,J+1) - \text{Peak_height}(R,J)) \] ! interpolation peak A

17900 \[ \text{Pb} = \text{Peak_height}(B,J) \times (\text{I-Peak_t}(B,J)_{i+1} - \text{Peak_t}(B,J)) / (\text{Peak_height}(B,J+1) - \text{Peak_t}(B,J)) \] ! interpolation peak B

17903 \[ \text{Iso_ratio}(J) = (\text{Pa/Pb}) \times (2^{P-1}) \] ! ratio of reference-pk to Ith-pk

17906 \[ \text{Ratio_time}(J) = (\text{I-Peak_t}(J,1)) / 100 \] ! time for this ratio, in seconds from the first peak

17909 NEXT J

17912 \[ \text{Linecorr} = (1 + \text{Lineoncorr}(R)) / (1 + \text{Lineoncorr}(ID)) \] ! Correction for resistor time-constants

17915 \[ \text{MAT} \text{Iso_ratio} = \text{Iso_ratio}(\text{Lineoncorr}) \]

17918 \[ \text{Nratios_used} = \text{Nsets} - 1 \]

17921 \[ \text{Ns} = \text{Nratios_used} \]

17924 DATA 0,0,0,0,0

17927 RESTORE 17924

17930 READ F, Audiscr, z, Alpha, Normalized

17933 \[ \text{MAT Sums} = (0) \]

17936 IF NOT Spike AND Normal(1) AND Ratio_count > 0 THEN

17939 ! correct for mass-dependent fractionation using the exponential law of Russell \\
17941 ! & others, Geoch. Cosmoch. Acta, v. 42, p. 1075-
17943 ! 1090, 1978

17945

17948 Normalized = 1

17951 \[ D = \text{Ref} \times \text{LOG}(\text{Normal}(1)/\text{Ref}) \]

17954 FOR J = 1 TO Ns

17957 \[ F = ((\text{Normrat_slope} \times \text{Ratio_time}(J)_{\text{Normrat_inter}} \times \text{Norm inverted}) / \text{Normal}(2)) \]

17960 IF F < 0 THEN Alpha = LOG(0)/0

17963 Audiscr = Audiscr + Alpha/Ns

17966 \[ M = (\text{Ref} / \text{Ratio_isotope}(ID)) \times (\text{Alpha} \times \text{Ref}) \]

17969 \[ z = 2 \times M/Ns \]

17972 \[ \text{Iso_ratio}(J) = \text{Iso_ratio}(J) \times M \]

17975 NEXT J

17978 Rs* = Ratio*(1)

17981 IF Inv < -1 THEN Rs* = FInvert*(Rs*)

17984 END IF

17987 !

17990 Calc_inverted = 1 - 2 * (Iso_ratio(J) / 1) ! ratios > 1?

17993 \[ ik = \text{Calc_inverted} + \text{Inv} \]

17996 \[ \text{PRINT} \text{USING} "W(24,2);CHRN(27:"M";25",CHR(27:"X";8160") \] 1 small print, 1/16" line-spacing

18002 FOR J = 1 TO Ns \ calculate std deviation of ratios, reject outliers

18005 IF Printsets THEN PRINT GROUND(Iso_ratio(J)*Inv, 6):

18008 Iso_ratio(J) = Iso_ratio(J) / Calc_inverted \ convert to reciprocal, raw ratios > 1

18011 CALL Sums(Sums(*), 1, Iso_ratio(J), Ratio_time(J, 0))

18014 NEXT J

18017 Rej_tolerance = 2 \ rejection tolerance, in sigma

18020 Nnumber_passes = 1 \ how many passes thru set-ratios for rejection

18023 Nrej = 0 \ # rejected ratios

18026 !

18029 \ ratio-rejection routine --------------------------------------------------

18032 LOOP

18035 K1 = Nratios_used

18038 Calc_sigma(Sl, Sums(1), Sums(2), Nratios_used)

18041 \[ Rr = \text{Sums}(1)/\text{Nratios_used} \]

18044 FOR J = 1 TO Ns

18047 IF (ABS(Rr - Iso_ratio(J))) \& Rej_tolerance > Sl AND Iso_ratio(J) THEN

18050 \[ Nrej = 1 \& Nrej \]

18053 IF Printsets AND Nratios_used > Ns THEN

18056 PRINT

18059 PRINT "REJECTED ";

18062 END IF

18065 IF Printsets THEN PRINT GROUND(Iso_ratio(J) / Ix, 6);
18068 CALL Sums(Sums(*),-1,Isoratio(J),Ratio_time(J),Hratios_used)
18071 Isoratio(J)=0
18074 END IF
18077 NEXT J
18080 IF Numpasses>2 THEN Rej_tolerance=Rej_tolerance+.3! increase rejection-tolerance by .3-sigma with each pass after the second pass
18083 Numpasses=Numpasses+1
18086 EXIT IF Hratios_used=1 OR Hratios_used=.7*Numpasses OR Numpasses=10
18089 ! don't reject more than 30% of ratios
18092 END LOOP
18095 ! ---------------------------------------------------------------

18104 Ave(I)=Rr*Ik  ! correct for Daly discrimination here, if desired
18107 Ns=Hratios_used+1
18110 IF Pb_4678 AND (Pb4678_cycle<>2) THEN PRINT "(partial-block 206/204 =:GROUNDS(R)=Ik,6):"; sigmaA obs. =:GROUNDS(Sig,3);"%"
18113 CALL Slope(Sums(*),Slope,Int,50,59,Hratios_used)
18116 PRINT USING "K,lt";CHR$(<2>a.l&8"1/6" line spacing
18119 PRINT "REJECTED":Rraj:"RATIO(S) OUT OF":Nsets-1:TAB(58):"TIME-CONSTANT CORRECTION =":GROUNDS(1,1+6*(timecon_ratcorr*Inv-1,3): "PPM":
18122 IF (I=1 AND NOT Pb_4678 OR (I=1 AND Pb_4678) THEN
18125 PRINT TAB(103):"BEAM":;
18128 IF Decay<>0 THEN PRINT "GROWTH":
18131 IF Decay=0 THEN PRINT "DECAY":
18134 PRINT " "=":GROUNDS(ABS(Decay),2):"/minute"
18137 ELSE
18140 PRINT
18143 END IF
18146 IF Normalized THEN
18149 PRINT "MSS-DISCRIMINATION CORRECTION WAS:";GROUNDS(100*(discr.3);"%/M.U. ":
18152 PRINT "((discr.-raw ratio was:";GROUNDS(Rr*Calc_inverted)^Ik,6);"%"
18155 END IF
18158 IF Daly_disc THEN PRINT "AVERAGE CORRECTED FOR DAILY DISCRIM. OF:";Daly_disc;"%/M.U. (RAW =";GROUNDS(Rr*Ik,6);")"
18161 IF ABS(Slope)>59 THEN PRINT USING "J,6X":"<<<<<<<< RATIO CHANGE DURING BLOCK OF ",GROUNDS(Slope*Ik/Rr,2), % PER MINUT E >>>>>)
18164 PRINT USING "N,2(X):";CHR$(<27)&8"&180"  ! normal print
18167 IF NOT Pb_4678 OR Pb4678_cycle<>1 THEN PRINT
18170 SUBEND ! ---------------------------------------------------------------

18179 Backgrounds:SUB Backgrounds(Conzer,Number_peaks,Pek(*),R,Pb4678_cycle,Nm,AFTER,INTEGER M,Pb_4678,Integr_time(*),Bkrd_posn(*),
18182 Ratio_isotope(*),Coarseflag)
18185 OPTION BASE 1
18188 COM /General/ Z$,INTGR Prtr(*),Subflag,Auto,Full_2uto,Full_2uto,Fac(*),F_t
18190 COM /Keyboard/ Cons,Cs&$,Dob,Cx&$,Clear$
18191 COM /specs/ Mx(0:1),Ionz,Ze(0:1),Noise(0:1)
18194 COM /Daily/ INTEGR Nu,Dayy,Mx(0:1),Dayy_ok(0:29),FBB(0:1,2Y4I
18197 COM /Filaments/ Filament(*),Fb&$YX101,INTEGR Films,FB
18200 COM /Data/ Peak_in(*),Pk,Interfer(*),Normal(*),TDD,INTEGR Data jsf(*),Data_collector,Inv,Bck_out,Next_run,Spke
18203 COM /Data& Bkrd(*),Bkrd var(*),Resdecay(*),INTEGR Sample,H prime,Nsets,Ref,Rr,Bkrd rdgs(*),Bkrd signal(*)
18206 REAL Bkrd_cst(120),Sums(S)
18209 INTEGER Bkrd_integ,Bkrd_time(12)
18212 Reprint=0 ! don't print rejected bkgrds
18215 FOR I=0 TO 19
18218 ON KEY I LABEL " CALL Clunk
18221 NEXT I
18224 IF Auto*Full_2uto THEN ON KEY D LABEL " Bac" GOTO Bck_out
18227 IF Auto*Full_2uto THEN ON KEY 1 LABEL " NEXT RUN" GOTO Next_run
18230 ON KEY 9 LABEL " ESCAPE" GOTO 18656
18233 Correct(Fb(*),Mx(*),Filament(*),Bkrd_posn(l,J),Mu,l,Coarseflag,Fac(*))
FOR I=I TO Mn
IF Pb_4678=0 OR Pb_4678_cycle(I)=0 THEN
Bkgds(1+After(I),I)=0  
ELSE
Bkgds(2+After(I),I)=0
END IF
NEXT I
MST Bkgd_time(I)= (Nsets/2)  
! calculate bkgd times, use Nsets/2 for interf-monitor peaks
IF Conzer THEN
A=0  
! calculate optimum background for shared positions
FOR I=1 TO Number_peaks
IF I>R THEN
Ratio=Integr_time(I)/Integr_time(R)
C=Nsets*(Peak(R)-Peak(I))/(Peak(R)+Peak(I))*SQR(1+Ratio)
D=1/Peak(R)*2/(1/Peak(R)+1/(Ratio*Peak(I)))/(Ions*Noise(Mu)^2)
Acon=MAK((R,C)/SQR(D))
Bcon=con
END IF
NEXT I
END IF
FOR I=1 TO Number_peaks
IF Pb_4678 AND Pb_4678_cycle(I) AND I=2 THEN
Bkgd_iso=4
ELSE
Bkgd_iso=1
END IF
! optimum background times
Bkgd_time(I)=Conzer*(Bcon+Integr_time(I))/((NOT Conzer)*((Nsets*Integr_time(I)/4)*SQR(Noise(Mu)^2)/(Noise(Mu)^2)+ABS(Peak(Bkgd_iso))))
IF Mu AND Peak(Bkgd_iso)<4 AND Bkgd_time(I)<7 THEN Bkgd_time(I)=7  
! at least 7 seconds for small Daly peaks
NEXT I
Rejected=0
Bct=0
FOR I=1 TO Mn
! take the background data
IF Conzer=0 OR I=Number_peaks THEN  
ELSE
FOR P=1+After(I) TO 2+After(I) After=0 for before-peaktop bkgds, 2 for after above
Above=(P=2)<(P=1)  
!1 for above bkgds, 0 for below bkgds
MAT Sums(0)
POST=GRND(Ratio_isotope(1)*((2*Above-1)/2,4)) mass-posn of this bkgd
IF (Ratio_isotope(1))=Bkgd_iso=0 THEN Next_p  
! skip if this position shared or if no isotope specified
Bkgds(P,I)=0
L=0
OUTPUT 8 USING "$H,1Z,",Bkgd_posn(I,Above+1)
Bct=1+Bct  
! to tell when the first mag-switch has occurred
U=4+(Bct)*3*(After(I))  
! bkgd wait-time (<3 or 4 for lst reading)
! Bkgd_integr=Bkgd_time(I)
FOR Si=-1 TO 1 STEP 2  
! look up & down isotope for shared bkgd-posns
FOR J=1 TO Mn
L=*(Ratio_isotope(I)=Ratio_isotope(J))  
! L is position in isotope list of sharing isotope
IF L THEN  
! if found a shared position
Bkgd_integr=MAX(Bkgd_time(I),((NOT Above)*(Si=1)+Above*(Si=1))*L=0)*Bkgd_time(L+(NOT L)))
GOTO 10104
END IF
NEXT J
NEXT Si
IF Bkgd_integr=120 THEN Bkgd_integr=120  
! no more than 120 seconds on this background
IF Bkgd_integr<5 THEN Bkgd_integr=5  
! no less than 5 seconds
IF (Bkrd_integr<6) THEN Bkrd_integr=fii >=6 seconds if a shared position
FOR J=1 TO U*Bkrd_integr
Enter_be3f!(Counts,t1u,l,l?Subflag)
IF Subflag) THEN SUBEXIT
DISP Ci$;Ratio_isotope(I)+(Z*Bboue-l)/2-,Cn$;TflB(20);Counts;TflB(30);CHR$(128t(J>U));U+(J>U)*Bkrd_integr-J+l;Cn$
IF J)0 THEH
Bkrd.ctsU-UKounts
COLL Suns(Suns(*),l,(Counts),0,Bkrd_integr)
END IF
HEKT J
Calc_signa(Signa,Sutts(l),Suns(2),Bkrd_integr)
FOR J=1 TO Bkrdjntegr
! reject background readings at a 2-sigma tolerance
IF (ABS(Sums(1)/Bkrd_integr-Bkrd_ct3(J»>2*SigNa> fiND Signa THEH
IF NOT Rejected AND Reprint THEN
PRINT "REJECTED: ";
Rejected=1
END IF
IF Reprint THEN PRINT URL&"(Bkrd_cts(J))">";("BURL$(Ratio_isotope(I)+(2*Above-1)/2)");":
CALL Suns(Sums(*),-l,(Bkrd_cts(J)),0,0)
END IF
HEHT J
Bkrds(P,I)=Suri5(l)/Bkrd_integr ! auerage bkrd cts/sec
Calc_signa(Signa,Sutts<l),Suns(2),Bkrd_integr)
Bkrd_signa<P,I)=Signa ! std dev. of bkrd counts/sec
Bkrd_rdg5<P,I)=Bkrd_integr ! # of readings used for std-dev-counts/sec
FOR J=1 TO Mn
!share bkrd s with other pks that have bkrd s at same mass-position
FOR K=1 TO 2! search other pk above & below posns
IF DROUHD(Ratio_isotope(J)-K-1.5,1)=Posn THEH GOSb'B Share.bkrd
HEHT K
NEXT J
NEXT P
IF Comzer THEN
FOR J=1 TO 10 Number_peaks ! share bkrd s for least-intense peak for Comzer=1 (non-interf-men-pks only)
Bkrd(J+After,I)=Bkrd(JtRfter,Number_peaks)
Bkrd_signa(J+After,I)=Bkrd_signa(J+flfter,Number_peaks)
Bkrd_rdg5(J+After,I)=Bkrd_rdg5(J+After,Number_peaks)
NEXT J
NEXT I
EN IF
END IF
FOR J=0 TO 1 ! print bkrd s & sigmas for all isotopes
L=0
FOR I=1 TO Mn
FOR K=Number_peaks+1 TO 1-1 ! but don’t print redundant interf-men. isotopes
IF (Ratio_isotope(K)=Ratio_isotope(I)) OR (Ratio_isotope(I)=0) THEN 18572
NEXT K
NEXT I
NEXT J
NEXT P
IF J=0 THEN PRINT "below, ";
IF J=1 THEN PRINT "above, ";
IF NOT After THEN PRINT "before pk-tops";
IF After THEN PRINT "after pk-tops"
FOR I=1 TO Nn  ! are above/below counts significantly different?
IF Ratio_isotope(I) THEN
   M=1+After
   Qa=(FNSt((Bkrd_rdgs(M,I)))*Bkrd_sigfia(M,I))/Bkrd_rdgs(M,I)
   Qb=(FNSt((Bkrd_rdgs(M,I)))*Bkrd_sigfia(M+1,I))/Bkrd_rdgs(M+1,I)
   Qc=ABS(Bkrd(M,I)-Bkrd(M+1,I))
   IF NOT Hu flNB Qc>1.5*SQR(Qa+Qb) THEN PRINT "STEPPEB";Ratio_isotope(I); " BF,
   NEXT I
   SUBEXIT
   Share_bkrd: Bkrd(K^fter,J)=Bkrd(P,I)  ! share backgrounds for equivalent magnet-positions
   Bkrd_sigfia(K+fter,J)=Bkrd_sigfia(P,I)+(HOT Bkrd_.sigrta<P,I>/10
   Bkrd_rdgs(K+fter,J)=Bkrd_rdgs(P,I)
   RETURN
   Bnc_out:Bnc_out=1
   SUBEXIT
   Next_run:Next_run=1
   SUBEXIT
   !
   Subflag=1  ! requested escape
   SUBEND ! ---------------------------------------------------------------
   Calc_signa:SUB Calc_signa(Sigsa,Sums,Sums_of_squares,INTEGER H)
   S=N*Sums_of_squares-Sums  A2 I calculate standard deviation
   Sigsa=SQ.R(S*(S>Q>/(N*(H)»  ! guard against negative square-roots
   SUBEND ! ---------------------------------------------------------------
   Invert:DEF FNInvert$(R$)  ! invert the ratio-string
   RETURN "EPOS(R$,"+"/"+"/S&1,EPOS(R$,"-/]"
   FNEND ! ---------------------------------------------------------------
   Getel:SUB Getel(Type,Runclass#,Nuclide#*,Normol0(*),Normo,Wg#,INTEGER Niso,Rf,HuO,MagnetO(*),Coarsebin(*),Ref)
   OPTION BASE 1
   ON ERROR GOTO 18794
   RSSJGN SPathl TO "TYPEsIHTERHflL" ! get element-series data fron disk in right-drive
   ENTER SPathl,Type;Runclass$,Niso,Hcoef(*),Hagnet(*),Coarse(iag(*),Peak_inter,Side,Rf,Nuclide#*,Normal(*),Inu,Bnc_out,Next_run,Spike
   HflT HornalO='Homal
   FOR 1=1 TO 21
      MagnetO(D=Hagnet(I,2)
   NEXT I
   OFF ERROR
   PRINTER IS CRT
   PRINT USING "1/
   L=RF  ! Order of reference-isotope in A-list
18767 Norm=Normal(1) ! Normalizing isotope
18770 IF NOT Auto THEN Na=""
18773 FOR Coarserange=0 TO 10 ! Coarse magnet-range
18776 IF Coarsebin(Coarserange)=Coarsemag(1) THEN 18772
18779 NEXT Coarserange
18782 Ref=Magnet(Rf,1) ! Reference isotope
18785 Mag(Daly.ok)=0
18788 Disp_elements(ElementClass$,Element$(*),Normal(*),Interfere(*),Coarserange,Ref,Magnet(*),Miso,Inv)
18791 SUBEXIT
18794 PRINT USING "2/,3(i0,/";Ci$a" ELEMENT-SERIES ",Type," NOT DEFINED OK DISK IN RIGHT-HAND DRIVE "&c$"
18797 Subflag=1
18800 Clunk
18803 WAIT 2
18806 SUBEND ! -----------------------------------------------
18809 !
18812 !
18815 Axis=SUB Axis(Xmin,Xmax,Ymin,Ymax,Xmin,Xmax,Ymin,Ymax,X$,Y$,Unit_xtick,INTERFERENCE & Plot,Logplot) ! draw plot-box for graphics display
18818 ! if Unit_xtick is 1, draw a tick for each integral X-axis value (used for weighted-average graphics). If Daly is 1, add "DAILY" to Y-axis label.
18821 ! if Daly is 0, add "CUP". & if Daly is negative, don't add nothing
18824 ! if Logplot is 1, use a logarithmic Y-axis
18827 ! Xmin,Xmax,Ymin,Ymax define windows of CRT (0-100)
18830 DIM Y$n#1241
18833 OFF KEY
18836 OFF KBD
18839 GINIT
18842 CSIZE 3.3
18845 DEG
18848 GCLEAR
18851 GRAPHICS ON
18854 OUTPUT KBD;CHR$(255);CHR$(75);:
18857 VIEWPORT Xmin*RATIO,Xmax*RATIO,Ymin,Ymax
18860 Xspred=Xmax-Xmin
18863 Yspred=Ymax-Ymin
18866 IF Unit_xtick THEN
18869 Xtick=1
18872 ELSE
18875 Xtick=ROUND(Xspred/5,1)
18878 !
18881 FOR I=1 TO 12! force lower-bound X value to have a min # of sig-figs
18884 X=FNOR(X(min),I)
18887 IF ABS(X[min]-X) THEN 18893
18890 NEXT I
18893 IF X[min]<0 OR ABS(X[min])>X THEN
18896 X[min]=SIGN(X[min])*X
18899 ELSE
18902 X[min]=-X*Xtick
18905 END IF
18908 FOR X=X[min] TO Xmax STEP Xtick ! force Xmax to lie on tick
18911 IF X<Xtick THEN X=Xtick
18914 Xmax=X*Xtick
18917 GOTO 18932
18920 END IF
18923 NEXT X
18926 END IF
18929 !
18932 IF Logplot THEN
18935 Ytick=1
18938 ELSE
18941 Ytk=10*INT(LGT(ABS(Yspred<(NOT Yspred)))))/8
18944 IF ABS(Yspred/Ytk))12 THEN
18947 Ytk=2*Ytk
18950 GOTO 18944
18953 END IF
18956 W=ABS(Ytk)/10*INT(LGT(ABS(Ytk<(NOT Ytk)))))
18959 IF W<INT(W) THEN
18962 Ytick=INT(W)*10*INT(LGT(ABS(Ytk<(NOT Ytk)))))
18965 ELSE
18968 Ytick=Ytk
18971 END IF
18974 FOR I=1 TO 12 ! force lower-bound Y to have a min # of sig-figs
18977 Y=FNDflBS<Ynin>,I>
18980 IF flBS(flBS<Ynin)-Y=X=Ytick IHEH 18986
18983 NEXT I
18986 IF Ymn>0 OR ABS(Ynin)<Y THEN
18989 Ynin=SSN(Ynin)*Y
18992 ELSE
18995 Ynin=-Y-Ytick
18998 END IF
19001 FOR Y=Ynin TO Ymax STEP Ytick ! force Ymax to lie on tick
19004 IF Y<Ytick)Y=Ynin THEN
19007 Ymax=Y<Ytick
19010 GOTO 19025
19013 END IF
19016 NEXT Y
19019 END IF
19022 !
19025 Xspred=Xmax-Ymin
19028 Yspred=Ymax-Ymin
19031 WINDOW Xmin-Xspred/8, Xmax, Ymin-Yspred/10-(<Ymax-Ymin>(60)), Ymax-Yspred/20
19034 CLIP Xmin, Xmax, Ymin, Ymax ! Use Ymin-Ytick/10 if interference
19037 XRES Xtick/(5<(NOT Unit_xtick)+Unit_xtick),Ytick,Xmin,Ymin,5-4*Unit_xtick,1,3+2*Logplot
19040 XRES Xtick/(5<(NOT Unit_xtick)+Unit_xtick),Ytick, Xmax,Ymax,5-4*Unit_xtick,1,3+2*Logplot
19043 IF Logplot THEN ! draw extra Y-ticks
19046 FOR A=0 TO INT(Ymax-Ymin)-1
19049 FOR S=2 TO 9
19052 MOVE Xmin+Xspred/80,Ymin+B-LGT(S)
19055 IDRAW -Xspred/80,0
19058 IMOVE 99*Xspred/80,0
19061 IDRAW Xspred/80,0
19064 NEXT S
19067 NEXT A
19070 END IF
19073 CLIP OFF
19076 LONG 6
19079 FOR I=Ymin TO Ymax STEP Xtick*((NOT Unit_xtick)+Unit_xtick*1/(<Xmax>16)>2*(<Xmax>30))
19082 IF I<Ymax THEN ! don't label tick if at right-edge of box
19085 MOVE I,Ymin
19088 LABEL I
19091 END IF
19094 NEXT I
19097 E=(2-Logplot)*Ytick
19100 LONG 8
19103 FOR I=Ymin TO Ymax STEP E
19106 MOVE Xmin+(Xspred/80+40*Logplot),I-Logplot*(Yspred/80)*(I=Ymax)+Yspred/70*(I=Ymin)
19109 IF (NOT Logplot) THEN LABEL I
19112 IF Logplot THEN LABEL 10"1;" "
19115 NEXT I
19118 LONG 6
19298 IF Minmag<0 OR Minmag>0 THEN
19301 Minmag=0
19304 Minmag=0
19307 END IF
19310 IF Maxmag>9999 OR Maxmag<9999 THEN
19313 Maxmag=1.0
19316 Maxmag=9999
19319 END IF
19322 Max_pk=WAL(Response$(3))
19325 IF NOT Coarsechange THEN Mass_speed=WAL(Response$(4))
19328 IF Coarsechange THEN Mass_speed=WAL(Response$(4))
19331 Logscan=(UPC$(Response$(5))="LIM")
19334 IF Logscan AND Max_pk(10 THEN Max_pk=10
19337 Coarse=WAL(Response$(6))
19340 IF Coarse()Coarse0 THEN
19343 Coarsechange=1
19346 Coarse0=Coarse
19349 Coarsemag(1)=Coarsebin(Coarse)
19352 OUTPUT 8 USING "AR,42":;"QFF",Coarsemag(1)
19355 Prompt$(1)="Start scan at magnet-value (0-9999)"
19358 Prompt$(2)="End scan at magnet-value (0-9999)"
19361 Prompt$(4)="Scan speed (magnet-units/sec)"
19364 Range(4,1)=1
19367 Range(4,2)=2000
19370 Response$(1)="0" 
19373 Response$(2)="9999"
19376 Response$(4)="100"
19379 GOTO Fillout_form
19382 END IF
19385 OUTPUT 8 USING "AR,42":;"QFF",Minmag
19388 IF NOT Logscan THEN
19391 Ynin=0
19394 Ymax=Max_pk
19397 ELSE
19400 M=GET(Max_pk)
19403 Ymax=NEXT(M)N(M MOD INT(M))/10
19406 Ynin=1-M-(Ymax/3)M
19409 END IF
19412 ON KEY 9 LABEL "ESCAPE" GOTO 19574
19415 ON KEY 0 LABEL "DOUBLE SPEED" GOSUB Double
19418 ON KEY 1 LABEL "HALVE SPEED" GOSUB Halve
19421 Axes(0,100,25,100,Minmag,Maxmag,Ynin,Ymax, "MAGNET UNITS", "HUG BEAM", O, Nu, Logscan))
19424 Yspeed=Ymax-Ynin
19427 Xspeed=Maxmag-Minmag
19430 IF NOT Coarsechange THEN
19433 Mass_inter=FMWmag(Moef(*),(Miso))-FMWmag(Moef(*),1),(Miso-1)
19436 Mass_speed=Mass_speed*Mass_inter
19439 FOR I=1 TO Miso
19442 IF Minmag(Magnet(I,2)) AND (Maxmag(Magnet(I,2)) THEN
19445 LINE TYPE 4
19448 MOVE Magnet(I,2),Ynin
19451 IDRAW 0,.5*Yspeed
19454 MOVE Magnet(I,2),Ynin+Yspeed*5
19457 LINE TYPE 1
19460 IF FMWmag(I) OR Xspeed/Mass_inter<13 THEN
19463 LONG 5
19466 CSIZE 3
19469 LABEL Magnet(I,1)
19472 END IF
19475 END IF
MOVE Xmax-Ymax/2,Ymin-Ymax/2,(Xmax-Ymin)/GO; (0.05+0.03*(Wynax-Wynin))/(GO); 

LABEL X

LOER 90

MOVE Xmax-Ymax/2,Ymin-Ymax/2

IF Daly>0 AND YS="**" THEN LABEL YS="**" (CUP)"

IF Daly>0 AND YS="**" THEN LABEL YS="**" ( ONLY)

IF Daly>0 THEN LABEL YS

LOFR 0

CLR Xnin,Xmax,Ynin,Ymax

Magnet_scansub Scan(Stripchart,Miniso,Maxiso,Max_pk,Mass_speed,Logscan,INTEGER Coarsebin(*),Iso,I_t)

OPTION BASE 1

COM /Magnet/ Mcoef(*),INTEGER L,Aside,Peak_inter,Coarsemag(D:2),Magnet(D:24,2),CoarseO

COM /Specs/ nX(D:1),nions,Peak_inter,Coarsemag(D:1)

COM /Paks/ Mx(D:1),Ions,nX(0:1),nnoise(0:1)

COM /Says/ Mx(D:1),Ions,nX(0:1),nnoise(0:1)

COM /Filants/ filament(*),fil(*),INTEGER Ofils,filnum

INTGER Pr

DIM Prompt$(6),Response$(6),Use$(6),Range(*>,6,1,Escape,3)

EDIT "Start scan at isotope","End scan at isotope","Max. beam on graph (mV)","Scan-speed (mass-units/second)","Linear or Log Scan (Lin/Log)"

OFF "Coarse-magnet range (0-10)"

DATA "",",","",","",","",","",","",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",","
19658  COM /Filaments/ Filament(*),Fil(*),INTEGER Nfiles,Filnum
19661  INTEGER Pr
19664  OFF KNOB
19667  ON KEY 9 LABEL "ESCAPE" GOTO 19720
19670  O=INT(.9*W@side)
19673  Max_inc=.01
19676  X=Magnet(L,2)-0
19679  N=H
19682  C=Magnet(L,2)-0
19685  Ynin=GROUND(Peaksize*.92,2)
19688  Start_graph=Ymax=Peaksize*(1+Max_inc)
19691  Yspred=Ymax-Ynin
19694  OUTPUT 8 USING "49,47:""091",X
19697  Rxes(1,100,25,100,-0,0,Ynin,Ymax,"MAGNET","H",0,0,0)
19700  CSIZE 3.3
19703  LONG 5
19706  MOVE 0,Ymin=Yspred/10
19709  LABEL TRIME(Nuclide(L))&"=""QUAD"(Magnet(L,1)) &"" PERK-SHAPED"
19712  MOVE 0,Ymin=Yspred/10
19715  LABEL "SCAN UP"
19718  FOR P=1 TO -1 STEP -2
19721  MOVE -0,Peaksize/2
19724  FOR I=1 TO (P<0)*C*(P<0)*R STEP P
19727  OUTPUT 8 USING "49,47:""091",I
19730  Enter_beam(C,Mu,L,L_C,P)
19733  IF Mu<Ymax THEN ! exceeds Y-max of graph- increase Ymax & redo
19736  Max_inc=Max_inc*.03
19739  GOTO Start_graph
19742  END IF
19745  IF P>0 THEN ORMU(I-Magnet(L,2),Mu
19748  IF P<0 THEN ORMU(Hagnet(L,2)-I0,.99*Mu
19751  NEXT I
19754  K=C
19757  MOVE 0,Ymin=Yspred/4
19760  LABEL "SCAN DOWN"
19763  MOVE 0,Mu
19766  NEXT P
19769  SUBEND ! ----------------------------------------------------------
19772  !
19775  !
19778  Avgraf=SUB Avgraf(Ratio(*),N,Mu,REAL Percent_error(*),Line(*)) weighted-average graphics
19781  OPTION BASE 1
19784  DIM Err(80)
19787  FOR I=1 TO N
19790  Err(I)=2*Ratio(I)*Percent_error(I)/100
19793  NEXT I
19796  Ymax=.9.E+99
19799  Ynin=5.E+99
19802  Boxwidth=2.5
19805  FOR I=1 TO N
19808  IF Ymax(Ratio(I)+Err(I)) THEN Ymax=Ratio(I)+Err(I)
19811  IF Ymin(Ratio(I)-Err(I)) THEN Ymin=Ratio(I)-Err(I)
19814  NEXT I
19817  Yspred=ABS(Max-Ynin)
19820  Xmax=Time(N)-(Time(N)-Time(1))/(5+3*(N+5))
19823  IF Xmin<0 THEN Xmin=0
19826  Xmax=Time(N)+Boxwidth/2
19829  Yspred=Vmin-Ynin
19832  Rxes(0,100,45,100,Ynin,Ymax,Ynin-Yspred/10,Ymax-Yspred/10,"TIME (minutes) ["X"]","R",0,-1,0)
19835  CSIZE 3.3
19478 NEXT I
19481 END IF
19484 LINE TYPE 1
19487 Ex_ymin=Logscan*10^"Ymin
19490 MOVE Minmag,0
19493 A=INT(Mag_speed/(t1 t-3))
19496 C=0
19499 Started=0
19502 P=Ax(NOT A)
19505 P2(1)=Minmag0-F/2
19514 WHILE P2(1)<Minmag0
19517 Mpos=INT(P2(1))+P/2
19520 IF Mpos=999 THEN Mpos=9999
19523 OUTPUT & USING "4A.42":"40.4F",Mpos
19526 Enter_beam(0,M,1,1 t,Pr)
19529 IF Coarsechange THEN OISP USING 19508;P2(1),Mag_speed,GROUND(0u,3)
19532 IF NOT Coarsechange THEN OISP USING 19511;P2(1),Mag_speed,GROUND(0u,3)
19535 IF Pt=1 THEN 19526
19538 IF NOT Logscan THEN
19541 Mu=Mu+Yspeed/200
19544 P2(2)>Mu
19547 ELSE
19550 IF Mu=Ex_ymin THEN Mu=1.05*Ex_ymin
19553 P2(2)>LGT(Mu)
19556 END IF
19559 IF Started THEN CALL Thickpen(P1(*),P2(*),Yspeed/150,1)
19562 P=PI=P2
19565 P2(1)=P+P(2)
19568 Started=1
19571 END WHILE
19574 MOVE 0,0
19577 Stripchart=0
19580 Coarse0=Coarse_in
19583 Coarsemag(1)=Coarsebin(Coarse0)
19586 SUBEXIT
19589 Double:IF Mag_speed>2000 THEN
19592 P=P*2
19595 BEEP 1000,.1
19598 Mag_speed=Mag_speed*2
19601 Mass_speed=Mass_speed*2
19604 END IF
19607 RETURN
19610 Value:IF Mag_speed>1 THEN
19613 P=P/2
19616 Mass_speed=Mass_speed/2
19619 Mass_speed=Mass_speed/2
19622 BEEP 100,.1
19625 END IF
19628 RETURN
19631 SUBEND!---------------------------------------------------------------
19634 !
19637 !
19640 PeakshapesUB Shape(Peaksize,Nuclide$("r"),INTEGER I t)
19643 ! look at peakshape graphics
19646 OPTION BASE 1
19649 COM /Magnet/ M=fit(4),INTEGER I,Rate,Peak_inter,Coarsemag(0t1),Magnet(0t24),Coarse
19652 COM /Specs/ Mu(B1),Ions,ze(0t1),Noise(0t1)
19655 COM /Daily/ INTEGER Mu,Daly,MmK(0t3,2),DB(0t24),Daly_ok(0t24),FfK(0t1,2),E3
AREA COLOR .5, .5, .5
FOR I=1 TO N
MODU I: I>I, BOXWIDTH/2, RATIO(I)-ERR(I)
RECTANGLE BOXWIDTH, 2*ERR(I), FILL, EDGE
NEXT I
SUBEND !--------------------------------------------------------------

19862 Flag: SUB Flag(Flag+1), INTEGER Flag nu, OPTIONAL Numprint) ! check filament flags
19865 OPTION BASE 1
19868 COM /General/ %, INTEGER Prt+1, Subflag, Auto, Full_auto, Foc+1, I
19871 COM /Daily/ INTEGER Mu, Daily, Mn$(0:3, 2)E8, Daily_ok(0:2), F$(0:1, 2)E1
19874 DIM F$(1:20)
19877 DATA "NO FILMENTS", "SINGLE-FILMENT", "SIDE-FILMENT ONLY", "TRIPLE-FILMENT"
19880 READ F$(*)
19883 FOR I=2 TO 1 STEP -1
19886 OUTPUT B: "COM" "RUN:(1-1)" ";28"
19889 OUTPUT B: "FIN"
19892 ENTER 8: F
19895 Flag(I)=F
19898 IF NPRR=2 THEN
19901 IF I=2 THEN PRINT USING "2/,K/,K,/,Y"; "PREHEAT SAMPLE: "
19904 IF I=1 THEN PRINT "RUNNING SAMPLE: "
19907 PRINT USING "K,/,/;FHR$(F$(F*)")
19910 IF ((I=1) AND (F=0)) OR (F=2) THEN CALL Clunk
19913 END IF
19916 OUTPUT B: Mt$(Flag nu, I)
19919 NEXT I
19922 SUBEND !--------------------------------------------------------------
19925 !
19928 !
19931 Cat: Sub Cat(INTEGER Prt+1) ! Print Run-Directory
19934 DIM Date$(121), Mn$(50), Mn$(101, 50), Pb$(101, 50)
19937 INTEGER Sample
19940 OFF KEY
19943 OFF KNOB
19946 MAT Rs=" (")"
19949 Rs$(1)="SEARCH THE DATA-DISK FOR A RUN WITH A PARTICULAR SAMPLE-NAME"
19952 Rs$(2)="SHOW THE DATA-DISK DIRECTORY FOR RUNS 1 TO 32"
19955 Rs$(4)="DISPLAY THE DISK-INFORMATION ON THE CRT ONLY (default)"
19958 Rs$(5)="PRINTOUT ' ' ' ' ' PRINTER"
19961 Rs$(10)="RETURN TO THE BHC"
19964 CALL Keymenu(Rs$(*)
19967 ON KEY 0 LABEL "SAMPLE-NAME" GOTO 20009
19970 ON KEY 1 LABEL "DIRECTORY" GOTO 20159
19973 ON KEY 3 LABEL "CRT DISPLAY" GOSUB Crt
19976 ON KEY 4 LABEL "PRINTOUT" GOSUB Printout
19979 ON KEY 9 LABEL "ESCAPE" GOTO 20249
19982 GOTO 19982
19985 !
19988 Crt: Printout=0
19991 PRINT " CRT DISPLAY ONLY"
19994 RETURN
19997 Printout: Printout=1
20000 PRINT "PRINTOUT SEARCH ON PRINTER"
20003 RETURN
20006 !
20009 OFF KEY
20012 Sp=""
20015 PRINT USING "10/,/K/,/K,/": "ENTER THE CHARACTERS THAT YOU WANT MATCHED IN THE FIRST 10 CHARACTERS", "OF THE RUN-NAME (or pres
20192 NEXT P
20195 NEXT I
20198 IF NOT Printout THEN
20201 DISP "PRESS CONTINUE WHEN READY TO GO ON, USE ARROWS TO SCROLL DISPLAY."
20204 PAUSE
20207 END IF
20210 GOTO 19946
20213 !
20216 RETRN:GOTO 20132
20219 PRINTER IS 1
20222 PRINT USING "K,K";"CANT ACCESS DATA FOR RUN " , I
20225 RETURN
20228 PRINTER IS 1
20231 PRINT USING "Z/,K";"UNABLE TO ACCESS THE RESULT FILE ON THE LEFT-HAND DISK."
20234 GOTO 20213
20237 PRINTER IS 1
20240 PRINT USING "Z/,K";"UNABLE TO ACCESS THE RESULT FILE ON THE LEFT-HAND DISK."
20243 PRINT USING "Z/,K";"PRESS CONTINUE TO RETURN TO BMC."
20246 PAUSE
20249 SUBEND ! -----------------------------------------------------------
20252 !
20255 Facshift:SUB Facshift(INTEGER Fac(*)
20258 PRINT USING "18/,K,2H,8(30,3H)");Foc<*)
20261 PRINT USING "18H,8(3C,3K),5/";Foc<*)
20267 OFF KEY
20269 OFF KNOB
20272 LOOP
20275 Plate=0
20279 LOOP
20282 INPUT "PLATE, HEU VALUE (0-999)? (press CONT to escape)",Plate,F
20285 EXIT IF (Plate(8)*(F>=0 BNO FH99) OR (Plate=8)*(FH60 flND F(=510)
20288 Clunk
20291 END LOOP
20294 EXIT IF Plate(1 OR Plate=0
20297 Foc(Plate)=F
20300 PRINT TAB((18-(Plate-1)*6,13);FMHR$(URL$(F))&"
20303 END LOOP
20306 SUBEND ! -----------------------------------------------------------
20309 !
20312 !
20315 find:SUB Find(INTEGER Redo,Barnum,Nfiles,Subflag,REAL Filament(*),line0)
20318 OPTION BASE 1
20321 COM /Keyboard/ Cn$,Ct$,Cv$,Cc$,C8$,Clear$ COM /Barrel/ INTEGER Barrel_position,Barrel_pos0,Max_barrel,Min_barrel COM /Daily/ INTEGER Mu,Daly,Min$(0:3,2)EO,Daily_ok(0:24),ff$(0:1,2)EO
20327 ! Rotate a new sample into position
20333 DIM Fil_flags(2)
20336 OFF KEY
20339 OFF KNOB
20342 MAT Daily_ok= (0)
20345 Default_barrel=O1+107*(FMBarpos((Barnum))-1) ! Rough estimate of correct barrel position for this sample
20348 Mu=0
20351 Subflag=0
20354 MAT Filament= (0)
20357 OUTPUT 8:"000000000"
20360 Zero_fil(Filament(*))
20363 IF Redo=0 THEN PRINT USING "3(K),/";"ROTATING BARREL ",Barnum," INTO POSITION <default position "URL$(Default_barrel)"
20366 Oct=1 ! try# counter
20369 !
Barpos=Default_barrel
Barrel_posO=Default_barrel
Barrel_position=Default_barrel-70
Barrel_position=Barrel_position*(Barrel_position>0)
Resbar
DISP CHR$(130)&"ROmiN6 10 BflRREL
Barrel_position=Barrel_position*(Barrel_position>=0)
RES
OUT B;"%M9978"
OUT B;"%354"
ENTER B;E
IF E THEN
WAIT .2
GOTO 20393
END IF
!
OUTPUT B;"%M0200"! put 0.2 amp through the sample filaments to check the filament flags
WAIT .1
!
LOOP
Barrel_position=Barrel_position+1
CALL Brl(Barrel_position,F,.09) ! Rotate barrel until get filament contact
EXIT IF FNFiltest(Nfils,F) OR (Barrel_position-Default_barrel)>70 OR Barrel_position>Default_barrel-70
END LOOP
!
IF Barrel_position-Default_barrel>70 THEN Default_exit
!
Min_barrel=Barrel_position ! Min_barrel is the position where contact is just made (positive rotation direction!)
REPEAT
Barrel_position=Barrel_position+1
Barrel_delta=Barrel_position-Default_barrel
CALL Brl(Barrel_position,F,.09) ! rotate in a positive direction until contact is lost
UNTIL FNFiltest(Nfils,F)=0 OR Barrel_delta>70
IF Barrel_delta>70 THEN
Subflag=1
PRINT USING "3/,3(X),2":Ci$a" OPEN CIRCUIT IN BflRREL-URLUES" (Is Filament Supply "0F2V("ON")>FMM("??")"
GOTO Default_exit
!
Max_barrel=Barrel_position ! Max_barrel is the position where filament contact is just lost (positive rotation)
Barrel_position=Max_barrel
Min_barrel=Min_barrel/2
IF ABS(Barrel_posO-Barrel_position)>34 OR Max_barrel-Min_barrel<9 THEN
!
shift the default position if the "best" pos'n differs from the default pos'n by more than 35 units.
Default_exit:Barrel_position=Default_barrel ! default barrel-value
Max_barrel=Barrel_position ! default "above" barrel-value
Min_barrel=Barrel_position-8 ! default "below" barrel-value
PRINT FNBl$("***USING DEFAULT BARREL-VALUES***"&" (should check manually)"
Subflag=2
Beep(160,.05,.03,.30)
END IF
!
Barrel_posO=Barrel_position
Barrel_position=(Barrel_position-70)*(Barrel_position>70)
CALL Brl(Barrel_position,F,1) ! rotate barrel to the final position
REPEAT
CALL Brl(Barrel_position+1,F,1,09)
Barrel_position=Barrel_position+1
UNTIL Barrel_position=Barrel_posO
!
Found_sample:PRINT DEF IS CRT
s CONTINUE to escape)."

PRINT USING "K,/,K:/";'"ANALYST WILL THEN SEARCH THE DATA-DISK FOR ALL RUNS WITH THOSE CHARACTERS,." AND SHOW WHICH FILE-NUMBERS THEY ARE LISTED UNDER."

PRINT USING "K,/,K:/";'"YOU CAN THEN USE THESE FILE-NUMBERS TO ACCESS DATA-BLOCKS WHOSE DIRECTORY"'"HAS BEEN OVER-Written."

INPUT S$

IF S$="" THEN 19916

S$=UPC$(TRIM$(S$))

DISP

ON ERROR GOTO 20237

ASSIGN IPathl TO "RESULT:INTERNAL,4,1"

OUTPUT KBD;CHR$(255)&CHR$(75);

ON ERROR GOTO 20195

ON KEY 9 LABEL " ESCAPE" GOTO 19946

FOR I=1 TO 500

ENTER IPathl,I;N$El,10

PRINT I;TFLB(6);N$

IF POS<UPC$(N$),S$) THEN

IF Fill THEN Fill

Fi2=I

ELSE

IF Fill THEN

FOR P=Printout TO 1 STEP -1

PRINTER IS Prtr(P)

PRINT USING "3/,2(U>,2(K),/Y'FOUNO RUN UITH "aCHR$(31)aS$aCHR$(31>a" IN FILES iF,Fil,"TO ",Fi2

NEXT P

ENDIF

ENDIF

OFF KEY

ON KEY 0 LABEL " CONTINUE" GOTO 20108

ON KEY 1 LABEL " QUIT" GOTO 19946

GOTO 20102

Fi2=0

Fi2=0

Fi2=0

Fi2=0

OFF KEY

ON KEY 9 LABEL " ESCAPE" GOTO 19946

GOTO 20132

END IF

OPTION OFF ERROR

ON KEY 9 LABEL " NO SUCH RUN-NAME EXISTS ON THE DISK. DO YOU WANT TO TRY AGAIN (Y/N)?"

OFF KEY

ON KEY 0 LABEL " YES" GOTO 20009

ON KEY 1 LABEL " NO" GOTO 20249

GOTO 20153

ON KEY 9 LABEL " ESCAPE" GOTO 19946

GOTO 20162

ON KEY 9 LABEL " ESCAPE" GOTO 19946

OUTPUT KBD;CHR$(255)&CHR$(75);

ON ERROR GOTO 20231

ASSIGN IPathl TO "RESULT:INTERNAL,4,1"

ON ERROR GOTO 20195

FOR I=1 TO 32

ENTER IPathl,I;Sample,Date$,Na$,Fil,Fi2

FOR P=Printout TO 1 STEP -1

PRINTER IS Prtr(P)

PRINT USING "8R,20,SK,5R,20,8K,12R,8K,4(K),/,K:/";'"BARREL # "Sample,"RUN # ",I,Date $,"File $s ",Fil," to ",Fi2,Na$
I put .2 amps through preheat filaments.

If Max_barrel - Min_barrel < 15 THEN PRINT "(this amount of contact is marginal. Take care)"

Temp = TIME - Time0 

\[ T = \frac{1}{1 + \mu t} \]

\[ \text{RETURN} \ 2.2 \times \text{SG}.R\left(\frac{2 \times \text{Dark} - (\text{Peak1} + \text{Peak2})}{\text{Ions}}\right) \]

\[ \text{LIMIT(DEF FNLinit(K, l, L2)) = (for FOCUS) Is value within focus-limit bounds?} \]

\[ \text{RETURN} \ \langle \text{K(L1) OR K(L2)} \rangle \]

\[ \text{FNST (H) = Student's-t approximation (H-l d.f.) (from Andy Turek)} \]

\[ \text{RETURN} \ \frac{12.7 \times (H-2) \times N \times 1.96 \times (H-1)}{SQR((H-1) - 2.13 \times (H-1) + 1.696)} \]

Enter beam:SUB Enter_beam(REAL Counts, Mu, INTEGER I, I_t, Pr, OPTIONAL REAL Wait, INTEGER Dropouts, Bnc)

\[ \text{get counts from the DVM of the spectrometer, convert to millivolts, protect against oversize beans} \]

\[ \text{OPTION BASE 1} \]

\[ \text{COM /Specs/ Mu(0:1), Ions, Ze(0:1), Noise(0:1)} \]

\[ \text{COM /Daly/ Mu, Daly, Mu(0:3,2)Dx(0:24), Ff(0:1,2)Dx(4)} \]

\[ \text{COM /filaments/ Filament(*),Filament(*), INTEGER Nfils, Filnum} \]

\[ \text{Bad_counts=0} \]

\[ \text{IF NPRR<8 THEN Turndown_ok=1} \]

\[ \text{IF NPRR>5 THEN ! set integration time & collector } \]

\[ \text{OUTPUT 8;Mu(I_t)} \]

\[ \text{EXIT IF Bad_counts<3 OR Counts*(1+\times(I_{t-1}))<Ze(Mu)-150} \]

\[ \text{guard against GPIO dropouts- don't accept any counts less than 15 below accepted zero} \]

\[ \text{Bad_counts=1+Bad_counts} \]

\[ \text{IF NPRR=0 THEN Turndown_ok=NOT Bnc} \]

\[ \text{IF NPRR<>S THEN} \]

\[ \text{END IF} \]

\[ \text{END LOOP} \]

\[ \text{END LOOP} \]

\[ \mu = ((\mu t-3)\times\text{Counts}\times\text{Ze(Mu)})/\text{Mu} \]! millivolts beam

\[ \text{IF Mu=0 AND Mu(1,E+4) OR Mu(0,5) THEN SUBEXIT} \]

\[ \text{IF Mu=0 AND Mu(1,E+4) THEN} \]! Faraday cup beam must be <10 v.
20726 IF Turndown_ok THEN ! If not called from the BMC, then
20729 ! reduce sample-filament current by 35%,
20732 F=N fila/(6)<(NOT L) ! but turn down the center-fil if on
20735 IF F=2 AND filament(2)=0 THEN F=1? Re-187 but a non-Re run.
20738 Target=.97*filament(F) ! isn't any current in the sides, turn
20741 M$=""$OM0"$OM1" ! down the center filament.
20744 OFF KEY
20747 ON KEY 9 LABEL " ESCAPE" GOTO 20019
20750 WHILE filament(F)>Target
20753 DISP " DEMO10: REDUCING FILAMENT-CURRENT BY 35% - ":filament(F),"(PRESS 9 TO ESCAPE)"
20756 OUTPUT 8 USING "4A,42;F8(0,F),FM(filenent(F)-.001)
20759 filament(F)=filament(F)-.001
20762 WAIT .02
20765 END WHILE
20768 DISP
20771 END IF
20774 END IF
20777 !
20780 IF Mu fiNO Mu>=50 THEN ! Daly beam must be |50 nV)
20783 ! switch from Daly to Faraday cup, wait 6 seconds
20786 Mu=0
20789 Daily_OK(L)=0
20792 Pr=2
20795 OUTPUT 8;M$=O,I_t)
20798 DISP USING ";K";"BEAM TOO INTENSE FOR DALLY- NOW ON CUP (6 seconds before beam will reappear)"
20801 Beep(400,.03,.02,20)
20804 WAIT 5.7
20807 DISP
20810 SUBEXIT
20813 END IF
20816 !
20819 SUBEND !---------------------------------------------------------------
208221
208251
208283 Clockset=SUB Clockset
20831 ON ERROR GOTO Bad_date
20834 A$=""
20837 DISP "Enter the current date, in the format "DATE$TIME$(TIME$(TIME$)
20840 IF NOT POS(DATE$,"1900") THEN DISP "("$FM$="CONTINUE"$" if OK)"
20843 INPUT A$
20846 IF A$="" THEN 20858
20849 SET TIME DATE(A$)
20852 A$=DATE$(TIME$(TIME$)
20855 IF VAL(A$)<(1984 OR VAL(A$))>2000 THEN Bad_date
20856 ON ERROR GOTO Bad_date
20861 A$=""
20864 DISP "Enter the correct time, in the format "AFNClock_12$(TIME$(TIME$)" ["FM$="CONTINUE"$" if OK]"
20867 INPUT A$
20870 IF A$="" THEN SUBEXIT
20873 C=POS(A$,":")
20876 R=POS(UPC$(A$),"R")
20879 P=POS(UPC$(A$),"P")
20882 H=VAL(A$)
20885 IF C=0 OR (R=0 AND P=0) THEN Bad_date
20888 H=H+12*(P AND H=12)
20891 M=VAL(A$[1+C])
20894 IF H=24 AND M=0 THEN H=0
20897 SET TIME TIME(VAL$(H)"$:"VAL$(M))
20900 OFF ERROR
20903 SUBEXIT
20001 ' !
20002 BAD_DATE: CLUNK
20005 PRINT AB: "* Isn't in the requested format. Please try again."
20008 GOTO 20063
20009 BAD_CLOCK: CLUNK
20012 PRINT AB: "* Isn't in the requested format. Please try again."
20014 GOTO 20061
20017 SUBEND ! -----------------------------
20019 !
20020 "Clunk:SUB Clunk I error-sound
20023 BEEP 220, .1
20024 BEEP 100, .2
20027 SUBEND ! -----------------------------
20029 !
20030 "Whoop:SUB Whoop I up-beat sound indicating process finished
20033 OPTION BASE 1
20035 DIM T(12,2)
20038 DATA 244,325,107,488,570,730,980,1140,1380,1630,0
20041 MAT T = (.01)
20044 FOR I = 1 TO 12
20047 READ T(I,1)
20050 NEXT I
20053 BEEP(0,0,0,0,T(*))
20056 SUBEND ! -----------------------------
20058 !
20060 "Superclunk:SUB Superclunk I something-is-very-wrong sound
20063 OPTION BASE 1
20065 DIM T(11,2)
20068 MAT T = (.04)
20071 T(11,1) = 0
20074 FOR J = 0 TO 1
20077 FOR I = 1 TO 20
20080 T(I + 20 * J,1) = 1000 - I * 40
20083 NEXT I
20086 NEXT J
20089 BEEP(0,0,0,0,T(*))
20092 SUBEND ! -----------------------------
20095 !
20097 "Br1:SUB Br1(INTEGER Barrel_pos,REAL Flag,Wait,OPTIONAL Display)
20099 I rotate barrel to Barrel_pos, wait Wait seconds, then check filament
20102 flag.
20105 '!
20108 DIM F$(3)C213
20111 READ F$(*)
20114 OUTPUT 8 USING "4R,42":"$88",Barrel_pos
20117 WAIT Wait
20120 IF Flag = 0 THEN ! query filament-flags if flag variable not negative
20123 OUTPUT 8:"$NM40:20"
20126 OUTPUT 8:"$SIMU"
20129 ENTER 8:Flag
20132 IF MPAR(4 THEN DISP "BARREL":Barrel_pos;TAB(20);F$#Flag)
20135 END IF
20138 SUBEND ! -----------------------------
20140 !
20143 !
21086 RESBAR:SUB Resbar !reset barrel to microswitch (old US routine?)
21089 OUTPUT B;"$00000000$10000000$00000000$010000"
       \ ! turn off all files
21092 DISP CHR$(130);"RESETTING BARREL";CHR$(120)
21095 OUTPUT B;"$0000-8"
21098 REPEAT
21101 OUTPUT B;"$0BB88000"
21104 WAIT 1
21107 OUTPUT B;"$0BB88000"
21110 OUTPUT B;"$OBB8008"
21113 WAIT .4
21116 OUTPUT B;"$SIMW"
21119 ENTER B: Barrel_stopped
21122 UNTIL Barrel_stopped
21125 DISP
21128 SUBEND!  " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " }
21266 Yes:DEF FMYes(Input$)
21269 RETURN 1 for input of Y, 0 for N, -1 for space, -2 for any other input
21272 - looks at only first character of input -
21275 DIM Temp$[11]
21278 Temp$[11,13]=UPC$(TRIM$(Input$))
21281 IF Temp$="Y" THEN RETURN 1
21284 IF Temp$="N" THEN RETURN 0
21287 IF Temp$=" " THEN RETURN -1
21290 RETURN -2
21293 ! END ---------------------------------------------
21296!

21302 Form:SUB Form(Prompt$(*) , Response$(*) , Use(*) , Range(*) , Nchoices , firstchoice , Escape , OPTIONAL Help Sub)
21306 2/28/81
21309 ! General-purpose Form fill-out screen.
21311 ! The Prompt$ array contains the prompts or labels of the parameters,
21314 ! the Response$ array as passed to the subprogram contains the default
21317 ! responses, and returns any changed responses.
21320 ! The Use array determines whether a given parameter can be accessed and
21323 ! changed.
21326 ! The Range array contains the minimum & maximum permissible values for a
21329 ! parameter (0,0 for any range,-1,-1 for any string, -2,-2 for V/H).
21332 ! Parameters whose values must be freshly entered must be indicated with a
21335 ! passed response of "??".
21338 ! Prompt$ and Response$ must be dimensioned for string-lengths of 36 & 37-50.
21341 ! No more than 15 choices can fit on the screen at one time.
21344 ! The cursor is controlled by either the KNOB, by CONTINUE (increment parameter)/CONTROL CONTINUE
21347 ! (decrement parameter) or by the UP/DOWN arrows.
21350 DIM ABC$(50), DEF$(38), GHI$(50)
21353 COM /Keyboard/ Cn$, Ci$, Cb$, Cu$, Qi$, Clear$
21356 Escape=0
21359 B$=RPT$(" ", 38)
21362 C$=RPT$(" ", 50)
21365 G$=Ci$ & Cn$
21368 OFF KEY
21371 Begin:CONTROL 1,1,0 ! HOME key
21374 CONTROL KBD:1 ! put CAPS LOCK on
21377 Capslock=1
21380 OUTPUT KBD:Clear$;
21383 PRINT TAB(3):FNUn$("PARAMETER"):TAB(13):FNUn$("RESPONSE")
21386 FOR I=1 TO Nchoices
21389 PRINT TABXY(1,12):Prompt$(I):TAB(42):Response$(I)
21392 NEXT I
21395 PRINT TABXY(1,10):"Use ARROWS, (CTRL) CONTINUE or KNOB to move cursor";
21398 IF NPAR=0 THEN PRINT " to different parameters."
21401 IF NPAR>0 THEN PRINT ", (CTRL) Help) for help."
21404 IF NOT Helped THEN
21407 FOR P=firstchoice TO Nchoices
21410 IF Use(?) THEN 21416
21413 NEXT P
21416 Np=P+1 "new" parameter$
21419 END IF
21422 Helped=0
21425 !
21428 Change_params:IF Np>Nchoices THEN Np=1
21431 IF Np>1 THEN Np=Nchoices
21434 IF Use(Np)=0 THEN ! goto next param if on a forbidden one
21437 Np=Np-1
21440 GOTO 21428
21443 END IF
2146 PRINT TABXY(42,P*2);3# ! blank out response area
2149 PRINT TABXY(1,P*2);Prompt$(P);TAB(42);Response$(P)[I,36] ! restore display format for previous parameter
2152 11=LEN(Prompt$(Np)) ! for adding pointer-dashes
2155 PRINT TABXY(2,I,Np*2);PI*"-",38-LL ! add dash-pointer
2158 Inverse:PRINT TABXY(I,Np*2);FNM$(Response$(Np)[I,37]) ! print response to be changed in inverse video
2161 !
2169 DISP " ENTER new value (press EXECUTE when all parameters defined, k9 to escape)"
2170 J=0 ! counter for position in response-string
2174 R$=Response$(Np)
2176 Get_keystroke:ON KBD.fill GOTO 21159
2178 ON KNOB .05 GOTO 21185
2179 GOTO 21479
2182 !
2195 IF KNOBD>0 THEN
2198 ON KNOB .1 GOSUB Retrn ! dummy
2201 GOTO 21551
2205 ELSE
2207 ON KNOB .1 GOSUB Retrn ! dummy
2209 GOTO 21503 END IF
2214 !
2219 CALL Kbd(KBD$) !
2222 ON KBD fill GOSUB Retrn ! dummy
2225 SELECT K
2228 CASE -199 ! k9 key pressed - escape.
2230 Escape=1
2232 CASE -187 ! ENTER key pressed
2233 Exec=0
2235 CASE -168 ! EXECUTE key pressed
2236 Exec=1
2238 CASE -189,-170 ! CONTINUE or DOWN-ARROW key pressed
2240 P=Np
2243 CASE -107 ! ENTER key pressed
2245 CASE -13 ! UP-ARROW key pressed
2248 CASE -16 ! CONTROL-CONTINUE or UP-ARROW keys pressed
2250 P=Np
2253 CASE -152 ! CONTROL-CONTINUE or UP-ARROW keys pressed
2255 P=Np
2258 CASE -211 ! DEL CHR
2260 IF J=1 AND J=LEN(TRIM$(R$)) THEN
2263 ASC(J+1)=ASC(J+1)+28 " 
2266 PRINT TABXY(42,I,Np*2);Ci$&ABC(J+1,J+1)&ABC(J+1,J+1)Ci$&ABC(J+1,J+1) 
2269 END IF
2272 CASE -31 ! DEL CHR
2274 CASE -94,-194 ! Rt arrow or Lft arrow pressed- move cursor indicating position in response-string
2276 IF J=0 THEN 21467
2279 J=J+195+K
2282 IF J=0 THEN J=0
2285 IF J>34 THEN J=34
21626 PRINT TABXY(42,Np+2);"\&I\&C;\&H\&J;+1,J+13&J&\&J;1+2,36) ! move the cursor
21629 GOTO Get_keystroke
21632 CASE -221 ! CLR LN key pressed: clear the response area
21635 PRINT TABXY(42,Np+2);CHR$(127)&RPT$(" ",38)&Cn$
21636 GOTO 21467
21641 CASE -176 ! PAUSE key
21644 PAUSE
21647 CASE -177 ! dump-alpha key pressed
21650 DUMP ALPHA
21653 GOTO Get_keystroke
21656 CASE -171 ! CAPS LOCK key pressed
21659 Capslock=NOT Capslock
21662 CONTROL KBD:Capslock
21665 BEEP 300+300*Capslock,.1
21668 GOTO Get_keystroke
21671 CASE 0,15,31 ! CTRL-H, CTRL-/, CTRL-?
21674 IF NP=8 THEN
21677 SELECT Helpsub
21680 CASE 1
21683 Manual_help(Np)
21686 CASE 2
21689 Fract_help(Np)
21692 CASE 3
21695 Magscan_help
21698 END SELECT
21701 Helped=1
21704 P=Np
21707 GOTO Begin
21710 ELSE
21713 CALL Clunk
21716 GOTO Get_keystroke
21719 END IF
21722 CASE (32,126) ! invalid key
21725 Clunk
21728 GOTO Get_keystroke
21731 CASE ELSE
21734 !
21737 J=J+1 ! increment string-position
21740 IF J=50 THEN J=50
21743 A[B,J]=CHR$(K) ! define this character of the response-string
21746 IF J=1 THEN
21749 PRINT TABXY(41,Np+2);RPT$(" ",10) ! clear response area
21752 R=CHR$(K)&Cn$[1,193
21755 END IF
21758 IF J>36 THEN PRINT TABXY(41,Np+2);CHR$(J+1);CHR$(J+1)$&Cn$ ! print the new character in the response area, followed by a cursor
21761 GOTO Get_keystroke
21764 END SELECT
21767 ELSE
21770 Response$(Np)=TRIN$(\&R)
21773 P=Np
21776 Numeric=0
21779 ON ERROR GOTO 21791
21782 Value=VAL(Response$(P)) ! if no error, string is a valid number
21785 OFF ERROR
21788 Numeric=1
21791 R1=Range(P,1) ! minimum allowable numeric response
21794 R2=Range(P,2) ! maximum " " "
21797 IF R1=-1 AND R2=-1 THEN 21857 ! String response OK
21800 IF R1=-2 AND R2=-2 THEN ! must be Y/N
21803  R=A(Ne=Response$(P))
21806  IF A(1) AND A(3) THEN
21809  DISP FN$("MUST ENTER")&FN$("Y")&FN$("OR")&FN$("N")&FN$("FOR THIS PARAMETER")
21812  GOTO Clunkout
21815  ELSE
21818  GOTO 21857
21821  END IF
21824  END IF
21827  IF NOT Numeric THEN
21830  DISP FN$("YOU MUST ENTER A NUMERIC RESPONSE TO THIS PARAMETER")
21833  GOTO Clunkout
21836  END IF
21839  IF R1<0 OR R2<0 THEN
21842  IF Value(R1 OR Value(R2) THEN
21845  DISP FN$("URL\$\$Value\$G" IS NOT WITHIN THE ACCEPTABLE RANGE ("URL\$\$R1\$G - "URL\$\$R2\$G") &"FOR THIS PARAMETER")
21848  GOTO Clunkout
21851  END IF
21854  END IF
21857  IF Exec THEN Done
21860  !
21863  Special lines to handle spike-query input from first MANDAT form.
21866  IF NP=0 THEN
21869  IF Helpsub=2 AND P=3 AND Response$(P)="?" THEN
21872  Whichspike(0,0,0) ! show user which spikes are defined
21875  Response$(P)="O"
21878  Helped=1
21881  P=Np
21884  GOTO Begin
21887  END IF
21890  END IF
21893  !
21896  PRINT TABAY(8,P,2);"9"
21899  Np=P+1
21902  BEEP 660,.06
21905  GOTO Change_params
21908 !
21911  Done:FOR I=1 TO Mchoices
21914  IF Response$(I),1,23="??" THEN Incomplete
21917  NEXT I
21920  OUTPUT KBD;Clear$;
21923  SUBEXIT
21926  !
21929  Incomplete! Check for undefined but essential parameters
21932  DISP FN$("SORRY- YOU MUST ENTER A VALUE FOR EVERY ")&FN$("??")&FN$("PARAMETER.")
21935  Clunk
21938  WAIT 2.5
21941  Np=I
21944  GOTO Change_params
21947  Return:RETURN
21950  Clunkout:Clunk
21953  WAIT 3
21956  Response$(P)="??&" !
21959  GOTO Inverse
21962  SUBMODI! -------------------------------
21965  !
21968  !
21971  Mbar:SUB Mbar\$(\$\$,INTEGER Barrel_position,Barrel_pos0,Min_barrel,Max_barrel,I_t) ! manual barrel-rotation
21974  OUTPUT KBD:CHR$(255)&CHR$(75);
21977  PRINT TABAY(1,4):"USE KBD OR ARROWS TO ROTATE BARREL."
21980  PRINT TABAY(1,6):"TO ROTATE BEYOND LIMITS OF FILAMENT-CONTACTS, PRESS THE "&FN$("SHIFT")&" KEY ALSO."
21983 FOR I=0 TO 19
21984 ON KEY I LABEL "" CALL Clunk
21985 NEXT I
21986 ON KEY 9 LABEL " " ESCAPE GOTO Exit
21989 R=KNOB
21991 OUTPUT 8;Int$(2*Mn$,I_t)
22000 DISP "BARREL: ";Barrel_position
22004 ON KNOB .1 GOTO 22016
22007 ON KBD GOTO 22040
22010 GOTO 22010
22013 #
22016 STATUS 2,10:L
22019 Limit=(L()I)
22022 IF KNOBX0 THEN
22025 GOSUB Increment
22028 ELSE
22031 GOSUB Decrement
22034 END IF
22037 #
22040 CALL Kybrd(KB$,K)
22043 IF K=-176 THEN PAUSE
22046 Incr=(K=-105 OR K=-169 OR K=-162 OR K=-194)
22049 Decr=(K=-104 OR K=-172 OR K=-196 OR K=-170)
22052 IF Incr=0 AND Decr=0 THEN 22010
22055 #
22058 Limit=(K=-162 OR K=-170 OR K=-196 OR K=-194)
22061 IF Incr THEN GOSUB Increment
22064 IF Decr THEN GOSUB Decrement
22067 GOTO 22010
22070 #
22073 Increment:IF (Limit AND Barrel_position)=Max_barrel-3) OR (NOT Limit AND Barrel_position=2900) THEN
22076 BEEP 1000, .05
22079 GOTO 22097
22082 ELSE
22085 Barrel_position=Barrel_position+1
22088 CALL Br(Barrel_position,-1,0,0)
22091 DISP "BARREL: ";Barrel_position
22094 END IF
22097 RETURN
22100 #
22103 Decrement:IF (Limit AND Barrel_position)=Min_barrel+3) OR (NOT Limit AND Barrel_position=0) THEN
22106 BEEP 1000, .05
22109 GOTO 22127
22112 ELSE
22115 Barrel_position=Barrel_position-1
22118 CALL Br(Barrel_position,-1,0,0)
22121 DISP "BARREL: ";Barrel_position
22124 END IF
22127 RETURN
22130 #
22133 Exit:OFF KBD
22136 OFF KNOB
22139 OUTPUT 8;Int$(Mn$,I_t)
22142 OUTPUT KBD:CHR$(255)\CHR$(75);
22145 SUBEND! ---------------------------------------------------------------
22148;
22151!
22154 Filamentknob:SUB Filamentknob(Stripchart,Lognorm,Limit,Collector$(#),INTEGER I_t,L,Isotope)
22157! manually change filament-currents, using knob or keyboard-keys
22160 OPTION BASE 1
COM /Specs/ Mx(0:1),Ions,Ze(0:1),Noise(0:1)
COM /Filemsgs/ Filemt(1),Fil$(*),INTEGR Nfils,Filnum
COM /Daly/ INTEGER Mux,Daly,Mn$(*,2:0),dniy ok(0:24),fil$(*,1:2:4)
COM /Keyboard/ Cn$,Cl4,C18,Cu$,Clear$

REAL Flag(2),Ok(4)
INTEGER Pr

OFF KEY
MAT Ok= (1)
MAT Daly ok= (0)
Daly ok(L)=Mu

OUTPUT KBD;Clear$;
a=KNOBX

I= Daly ok(L)

IF Mant(I$ 2:1)<.1 THEN OUTPUT 8;fil$(I,J),200
NEXT J
NEXT I

CALL Flag(Flag(*),Mu,1)

FOR I=1 TO 4
| re zero fil-currents if necessary
IF Flag(I)<2 OR Nfils=1 THEN Ok(I)=0
NEXT I

OUTPUT B;Mn$(Mu,I)

IF Flag(I)<2 OR Flag(I)<0 THEN Ok(I)=0
IF Flag(I)<2 OR Nfils=1 THEN Ok(I)=0
IF (Flag(I)<1 AND Flag(I)<3) THEN Ok(I)=0
IF Flag(I)<2 THEN Ok(I)=0

IF filanent(I)<.1 AND Flag(I)<2 THEN Ok(I)=0

| no filament contacts at all
IF SuM(Ok)=0 THEN ! no filament contacts at all

PRINT USING "18.4X,K,6/";Fil$(I)
Clunk
WAIT 2
SUBEDIT
END IF

I = PRINT TABBY(14,10);"Use "$fil$("KNOBX")" to change filament-currents."
FOR I=1 TO 4
IF Ok(I) THEN PRINT TABBY(14+(I-1)*17,j4);fil$("K"|"I"|"I"|"I")

NEXT I

IF Ok(I) THEN PRINT TABBY(10,15);"CENTER-SAMPLE"
IF Ok(2) THEN PRINT TABBY(27,15);"SIDE-SAMPLE"
IF Ok(3) THEN PRINT TABBY(44,15);"CENTER-PREHEAT"
IF Ok(4) THEN PRINT TABBY(61,15);"SIDE-PREHEAT"
GRAPHICS OFF
END IF

FOR I=0 TO 19
ON KEY I CALL Clunk
NEXT I

IF SuM(Ok)>1 AND Ok(I) THEN ON KEY 1 LABEL "CENTER-SAMPLE" GOSUB 22403
IF Ok(2) THEN ON KEY 2 LABEL "SIDE-SAMPLE" GOSUB 22409
IF NOT Stripchart THEN
IF Ok(3) THEN ON KEY 3 LABEL "CENTER-PREHEAT" GOSUB 22415
IF Ok(4) THEN ON KEY 4 LABEL "SIDE-PREHEAT" GOSUB 22421
END IF
ON KEY 9 LABEL " ESCAPE" GOTO Exit
GOSUB Dispcurrs
22366  _t=2
22367  Enter_beam(Counts,Mu,L,2,Pr)
22369  WAIT .1
22372  ON CYCLE .195 GOSUB Query_beam
22375  ON KNOB .10 GOSUB Knob_change
22378  ON KBD GOSUB Keyboard
22381  GOTO 22361
22364  !
22367  Query_beam=Enter_beam(Counts,Mu,L,2,Pr)
22370  IF Pr>1 THEN Exit
22373  T=(TImeD-TinE0)/60
22376  IF Stripchart THEN
22379  IF NOT Lognon THEN ORflU T,Mu
22382  IF Lognon THEN ORflU T,LGT(BBS(Mu+NOT Mu)))
22385  ELSE
22388  IMAGE 10k," nU",n,k,", 3k,K
22391  DISP USING 22388;Isotope,ROUND(Mu,4)
22394  END IF
22397  RETURN
22400  !
22403  Filnum=1
22406  GOTO Change_filnum
22409  Filnum=2
22412  GOTO Change_filnum
22415  Filnum=3
22418  GOTO Change_filnum
22421  Filnum=4
22424  !
22427  Change_filnum=BEEP 440,1
22430  Dispcurr=IF NOT Stripchart THEN
22433  FOR I=1 TO 4
22436  IF OK(I) THEN
22439  US=CHR$(128*KFilnun=I))&" "&URL$(Filament(I))&" "&En$
22442  PRINT TBLKY(12+0*(Filnun=I-1))+(1-1)*17,16);US&RTS"",10-LEN(U$)
22445  END IF
22448  NEXT I
22451  ELSE
22457  DISP USING 22454;CHR$(128+3*(Filnum=1)),Filament(I),Isotope,CHR$(128+3*(Filnum=2)),Filament(2),Collector$;Mu
22460  END IF
22463  Curr=Filament(Filnum)
22466  RETURN
22469  !
22472  Keyboard=CALL Kvbrd(KBD$,K)
22475  SELECT K
22478  CASE -162,43           ! up-arrow or plus key pressed
22481  Change=.01
22484  GOTO Change_current
22487  CASE -170,45           ! down-arrow or minus key pressed
22490  Change=-.01
22493  GOTO Change_current
22496  CASE -176              ! PAUSE key pressed
22499  PAUSE
22502  CASE ELSE
22505  Clunk
22508  RETURN
22511  END SELECT
22514  !
22517  Knob_change=Change=INT(1000.*KNOBX/2900)/1000
22520  Change_current=Curf=Filament(Filnum)+Change
22523  SELECT Curr
22526  CASE 0
22529   Curr=0
22532  END SELECT
22535  CASE ?
22538   Curr=?
22541  END
22544  GOSUB Dispcurrs
22547  P=(Filnum>2)
22553  GOSUB Dispcurrs
22556  RETURN
22562
22565  Exit:OFF CYCLE
22568  OFF KNOB
22571  OFF KBD
22574  OUTPUT KBD;CHR$(255>aCHR$<75>;)
22577  SUBEND! -     -        .-..  ..
22580
22583
22586  RutofoiwSUB Buto_forn(Run,Std,Escape,Stduar(*),Saflple.naiel(*),Std_nane$<*),Stdtype$(*),Checked_els,Element$(*),INTEGER Std.
5Q(*»
22589  OPTION BRSE 1
22592  COM /Keyboard/ Ln#,Ci$,Cu$,O#,Cl$;Clear$
22595  COM /Auto_form/ Lastresponse$(*),Nresp(*),Last_runtype$,Normal(*),Response$(*),Prompt$(*)
22598  COM /Auto_form/ Spike,INTERG 1ss(*),Niss,RF,Magnet(*)
22601  Variables 1 thru 27 are the usual run/std variables, variables -3 thru 0 are the barrel#, run-type, isotopes, & sample name.
22604  All variables but -2 and 0 must be numeric.
22607  Variable -1 (isotopes) remains a string which must contain the isotopes, separated by commas.
22610
22613  ! The Prompt$ array contains the prompts or labels of the parameters.
22616  ! The Response$ array as passed to the subprogram contains the default
22619  ! responses, and returns any changed responses.
22622  ! The Use array determines whether a given parameter can be accessed and
22625  ! changed.
22628  ! Parameters whose values must be freshly entered must be indicated with a
22631  ! passed response of "??".
22634  ! Move cursor with either KNOB, CONTINUE, or CONTROL-CONTINUE, or UP/DOWN arrows.
22637  ! Use right- or left-arrows to move cursor within a response, use the
22640  ! CLR LN key to blank-out a response.
22643  !
22646  ! OFF KEY
22649  Not_captioned=1
22652  DIM P(501,(-3:0,2),Lr(-3:0),D(-3:0),Disp$(0G),R#(103),Mu#(0:24)163,S#(101),Isot(0),P#(501),Runtype$(101),Mcoef(3)
22655  INTEGER Co(0:1),Peak_inter,As,Use(-4:27),S_iso(3),Sref
22658  MAT Use=(1)
22661  Use(-4)=0
22664  FOR I=2 TO 4 ! don't access cf-only pars if a single-fill run
22667  IF Response$(I)="I" THEN Use(I)=0
22670  NEXT I
22673
22676  DATA 47,1,14,2,47,2,19,3  ! columns, rows for response of params -3 to 0
22679  DATA 4,12,31,50  ! lengths of response-strings for params -3 to 0
22682  DATA 38,1,38,1  ! columns for prompts of params -3 to 0
22685!
22688  DATA RUN#,BARREL#,ELEMENT .....ISOTOPES,SAMPLE NAME
22691  DATA STD-RUN,",",ELEMENT,ISOTOPES,STD-RUN NAME
22694  DATA SINGLE() = TRIPLE(3),FOCUSING ISO TOE (CF),CENTER-FIL BEAM (U),INITIAL CF (amps)
22697  DATA "DAILY ENABLE (0,1,2)",CURRENT-1 (amps),RATE-1 (nA/SEC),UNIT-1 (min.),CURRENT-2 (amps),RATE-2 (nA/sec)
DATA WAIT-2 (min.), DATA-WAIT (min.), ABORT CURRENT (amps)
DATA MIN. BEAM (u), MAX. BEAM (u), DEFAULT CURR (amps), DEFAULT BEAM (u)
DATA FIL. INCH/BLOCK (amps), MIN. BLOCKS, MAX. BLOCKS, MIN. SUMMATION (u), #SETS/BLOCK, MAX. GROWTH (#/minute), PREHEAT CT (amps), PR
HEAT SF (amps), NORMSPIKE

READ C(*), Lr(*), D(*)
IF Std THEN
RESTORE 22691
Lr(0)=10
ELSE
RESTORE 22690
END IF
FOR I=4 TO 0
READ Prompt$(*)
NEXT I
FOR I=1 TO 26
READ Prompt$(*)
NEXT I
IF Lastresponse$(I)="" THEN
MRT Response$= Lastresponse$
ELSE
RESTORE 22696
READ Response$(*)
END IF
IF Not Std THEN
Response$[*]=URL$<Run>
ELSE
Use(-3)=0
Response$(-3)="
END IF
Disp$="ENTER value (EXECUTE when done, K to escape. KNOB, (CTRL)CONTINUE moves cursor)"
CONTROL 1,4:0 / HOME key
Capslock=1
CONTROL KBD:Capslock ! CAPS LOCK on
Beginn=OUTPUT KBD;Clear$;
print the prompts & responses for the run#, barrel#, run-type, isotopes, and sample-name
PRINT TABXY(1,1);Prompt$(-1);TAB(4);Prompt$(-1);TAB(6);Response$(-1)
PRINT TABXY(1,2);Prompt$(-2);TAB(17);Response$(-2);TAB(38);Prompt$(-1);TAB(17);Response$(-1)
PRINT TABXY(1,3);Prompt$(-3);TAB(17);Response$(-3)
END IF
IF Not Std THEN
Hot_copied=0
P=0
Hp=0
FOR I=1 TO 13
FOR J=0 TO 1
IF Response$(-1)*1" OR J OR (1<2 OR I>4) THEN
Col=I+1*39
Row=I+4
Param=I+J*13
PRINT TABXY(Col,Row);Prompt$;TAB(24+Col);Response$;Param
END IF
NEXT J
NEXT I
IF Not Std AND Not_copied=0 THEN
P=0
Hp=0

IF P<1 THEN
    Last_response=P<0>(P)
    Last_promptcol=D(P)
    Last_responsecol=C(P,1)
    Last_row=C(P,2)
ELSE
    Last_response=P<12
    Last_promptcol=1+39*(P>13)
    Last_responsecol=21+39*(P>13)
    Last_row=P+4+13*(P>13)
ENDIF

IF Hp<1 THEN
    Response=len(Prmpt$(P))
    Promptcol=0(Prmpt$(P))
    Nrcol=C(Prmpt$(P),1)
    Row=C(Hp,2)
ELSE
    Response=P+2
    Promptcol=1+39*(P>13)
    Nrcol=21+39*(P>13)
    Row=Hp+4+13*(P>13)
ENDIF

IF Np=-2 THEN ! run-type or std-run type; show detailed prompt
    PRINT TABXY(1,3);"RP$(P)"
    IF NOT Std THEN
        PRINT TABXY(1,3);"(e.g. \$FH$("Sr")<\$Rc") To use std runs, enclose std-run in asterisks, e.g. \$FH$("*12*")"
    ELSE
        PRINT TABXY(1,3);"must match an element defined on the DATA disk"
    END IF
ENDIF

IF Use(P) THEN
    PRINT TABXY(Prmptcol*1+Li,Response$(P));"",80)
    PRINT TABXY(Prmptcol*1+Li,Response$(P));
    IF Use(P) THEN
        PRINT TABXY(Prmptcol*1+Li,Response$(P));
        PRINT TABXY(Prmptcol*1+Li,Response$(P));
        PRINT TABXY(Prmptcol*1+Li,Response$(P));
    END IF
ENDIF
23051 R$=RPT$(" ", Resp_len)
23054 Get_keystroke:ON KBD ALL GOTO 23090
23057 ON KNOB .07 GOTO 23066
23060 GOTO 23060
23063 :=
23066 IF KNOB=0 THEN
23069 ON KNOB=0 GOSUB Retrn ! to disable knob while processing knob-rotation
23072 GOTO Movecursor_down
23075 ELSE
23078 ON KNOB=0 GOSUB Retrn ! ditto
23081 GOTO Movecursor_up
23084 END IF
23087 :=
23090 CALL Keybd(KBD$, K)
23093 ON KBD ALL GOSUB Retrn ! disable live keyboard while processing keystroke
230961
23099 SELECT K
23102 CASE -199 ! k9 key pressed, so escape
23105 Escape=1
23108 OUTPUT KBD:Clear$;
23111 SUBEXIT
23114 CASE -187 ! ENTER key pressed
23117 Exec=0
23120 GOTO Check_response
23123 CASE -160 ! EXECUTE key pressed
23126 Exec=1
23129 GOTO Check_response
23132 CASE -189,-170 ! CONTINUE or DOWN-ARROW key pressed
23135 Movecursor_down:=Hp
23138 REPENT
23141 Hp:=Hp+1
23144 IF Hp>26 THEN Hp=-3
23147 UNTIL Use(Hp) AND (Response$(1)<"1" OR (Hp<2) OR (Hp)=4))
23150 GOTO Change_params
23153 CASE -1,-162 ! CONTROL-CONTINUE or UP-ARROW keys pressed
23156 Movecursor_up:=Hp
23159 REPENT
23162 Hp:=Hp-1
23165 IF Hp<-3 THEN Hp=26
23168 UNTIL Use(Hp) AND (Response$(1)<"1" OR (Hp<2) OR (Hp)=4))
23171 Go_on=1
23174 GOTO Change_params
23177 CASE -196,-194 ! rt arrow or lt arrow pressed- move cursor indicating position in response-string
23180 J=J+195*K
23183 IF J>26 THEN J=0
23186 IF JResp_len-1 THEN JResp_len-1
23189 PRINT TABRARY(Ncol1,Rw):ASC1,ASC1 \&ASC1,J=178\&ASC1,J=2,Resp_len
23192 GOTO Get_keystroke
23195 CASE -221 ! CLR LN key pressed: clear response area
23198 PRINT TABRARY(Ncol,Rw):CHR$(127)\RPT$(\", Resp_len)\Cn$
23201 GOTO Start_response
23204 CASE -176
23207 PAUSL
23210 CASE -177 ! dump alpha key pressed
23213 DUMP ALPHA
23216 CASE -171 ! CAPS LOCK key pressed
23219 Capslock=NOT Capslock
23222 CONTROL KBD:Capslock
23225 BEEP 300*300*Capslock,1
23228 GOTO Get_keystroke
23231 CASE 0,15,31 • CTL-H, CTL-?, CTL-
23234 CALL Autohelp(Np,Std)
23237 Redo_screen=1
23240 P=Np
23243 GOTO Begin
23246 CASE (22,126) • non-alphanumeric key, so ignore
23249 DEEP 200,1
23252 GOTO Get_keystroke
23255 CASE ELSE • valid alphanumeric keystroke
23258!
23261!
23264 IF J=0 AND TRIM$(R$)="" THEN PRINT TABXY(Hrcol,Row);RPT$="
",Resp_len) • clear response area
23267 J+=1 • process character to put in response-string & increment string-position
23270 IF J>Resp_len THEN J=Resp_len
23273 ARG1,2=CHR$(K) • define this character of the response-string
23276 PRINT TABXY(Hrcol+J,Row);ARG1,2) • print the new character in the response area
23279 GOTO Get_keystroke
23282 CASE -3
23285!
23288!
23291!
23294 IF Np(-2) AND Np(0) AND Np(0) AND (NOT Outgas OR Response$(Np)="**") THEN
23297 ON ERROR GOTO Non_numeric
23300 Hresp(Np)=URL(Response$(Np))
23303 OFF ERROR
23306 END IF
23309!
23315 N=Nresp(Np)
23318 P$=Response$(Np)
23321 SELECT Np • check values for valid ranges
23324 CASE -3
23327 IFN1 OR N16 THEN
23330 DISP Ci$;" THE BARRELS MUST BE IN THE RANGE OF 1 TO 16 ":Ci$
23333 GOTO Clunkout
23336 END IF
23339 IF TRIM$(UPC$(Response$(-2))="OUTGAS" THEN Outgas=1 • in case EXECUTE key pressed just for barrel# input
23342 IF Outgas AND (Response$(D)="??") OR Response$(D)="**" THEN Response$(D)="**"
23345 IF Sample_name$(N)="" THEN
23348 IF NOT Outgas THEN Response$(N)=Sample name$(N)
23351 Response$(N)=Sample_name$(N)
23354 PRINT TABXY(0,1),C(0,2);Response$(D);RPT$=" ",Lr$(D)=LEN(A$)) • print the sample name
23357 END IF
23360 CASE -2
23363 Response$(D)=TRIM$(Response$(-2)[1,0]) • trim to correct length
23366 R$=UPC$(Response$(-2))
23369 IF R$="" THEN
23372 DISP FNH$("YOU MUST ENTER AN ELEMENT-TYPE")
23375 GOTO Clunkout
23378 END IF
23381 Outgas=POS(R$,"OUTGAS") • indicates outgassing only
23384 IF NOT Outgas AND Was_outgas THEN
23387 FOR I=-3 TO 26 • change all "**" responses to ?? if not an outgas run
23390 IF Response$(I)="**" THEN
23393 Response$(I)="??"
23396 IF (I<2 OR I>4) AND Response$(I)="1" THEN
23399 !resp(I)=1
23402 Use(I)=1
23405 END IF
23408 END IF
23411
23111 NEXT I
23114 Redo_screen=1
23117 Np=1
23120 Was_outgas=0
23123 GOTO Begin
23126 END IF
23133 IF Preheat=(R$El,11="P")*Outgas ! outgas preheat filaments only
23135 Sample=(NOT Preheat) ! outgas sample filaments only
23138 DATA -3,-2,1,4,6,7,8,10,11
23141 FOR I=-3 TO 26
23144 FOR J=1 TO 10
23147 IF H=0 THEN 23171
23150 NEXT J
23153 NEXT I
23155 REDO SCREEN=1
23158 Uas_outgas=1
23161 GOTO Begin
23164 END IF
23169 !
23172 IF P$="*?m" THEN ! request display of defined std-run variables
23175 GOSUB Load_stduars:
23178 OUTPUT KBD:Clear$;
23180 FOR 1=1 TO 32
23183 IF Std_run(I,1) THEN
23186 IF 1=17 THEN
23189 DISP "PRESS "&CHR$(13001)+" TO SEE MORE STANDARD RUNS"
23192 PAUSE
23195 NEXT I
23198 PRINT TAB(3);TAB(12);UPC$(Std_type$(I));TAB(21);Std_name$(I);
23201 FOR J=1 TO 8
23204 IF Std_iso(I,J) THEN PRINT TAB(33)+S$(J-1);Std_iso(I,J);
23207 NEXT J
23210 PRINT TAB(69);Std_run(I,1)
23213 END IF
23216 NEXT I
23219 DISP "PRESS "&CHR$(13001)+" TO RETURN TO RUN-VARIABLE SCREEN"
23222 PAUSE
23225 GOTO Begin
23228 END IF
23231 !
23234 IF NOT Std AND NOT Outgas THEN ! test for std-run input
23237 Stdcopy=0
23240 P1=POS(P$,"w")
23243 P2=POS(P$1+P1,"x")+P1
23246 IF P1 AND P2 AND P2>P1 THEN Stdcopy=URL$(P$1+P2-1)
23249 IF Stdcopy THEN
23252 GOSUB Load_stduars
23255 Response$(-2)=Std_type$(Stdcopy) ! transfer std-run vars to main
23258 FOR K=1 TO 26
23261 Response$(K)=URL$(Std_run(Stdcopy,K))
23264 NEXT K
23267 !

FOR K=2 TO 10
IF Std_iso<Std_copy,K> THEN Response$(-1)=Response$(-1:1)
FOR I=2 TO 4
Use(I)=0
NEXT I
CLS
FOR I=2 TO 4
Use(I)=1
NEXT I
END IF
GOTO Begin
ELSE
Runtype$=R$
END IF
END IF
CASE -1 ! isotopes input
IF NOT Outgas THEN
Parse((Response$<Hp>),Isot(x),Diso)
Hfil Isot= Isot
IF Diso<2 THEN
DISP C#;" NEED AT LEAST TWO ISOTOPES!! ";Cn$
GOTO Clunkout
END IF
ELSE
CASE 0 ! name
IF Std AND NOT Outgas THEN Response$<D(0)=TRIM(Responses$<DI:10>)
CASE 1 ! #filaments: blank out or restore params 2-4 as required
IF K>1 AND M>3 THEN
DISP FN$="SINGLE-FIL.=1 TRIPLE-FIL.=3"
GOTO Clunkout
END IF
CASE 3 ! Clunk out
CASE 0 ! name
IF Use(2)=I OR M=1 THEN
FOR I=2 TO 4
Use(I)=0
Response$<I>="0"
PRINT TABXY(1,4);Prmt$(" ",39)
NEXT I
P=Np
Np=5*<Use(5)=0>
GOTO Change_Params
END IF
ELSE
IF Use(2)=0 THEN
Outer=(POS(UPC$(Response$<2:2>),"OUTGAS") AND M>3) ! indicates an outgas run for a triple-filament
IF Outer THEN Np=3
FOR I=2 TO 4
IF I=4 OR Outer=0 THEN
Use(I)=1
PRINT TABXY(1,4);Prmt$(I);TAB(25);Response$(I)
END IF
END IF
NEXT I
END IF
23774 END IF
23777 CASE 17 ! default beam
23780 IF NOT Outgas AND (ABS(N)<0 OR ABS(N)>10) THEN Out_of_range
23783 CASE 14, 15 ! beam sizes in volts
23789 IF NOT Outgas AND (N<.0001 OR N>10) THEN Out_of_range
23791 CASE 3 ! CF-only beam
23792 IF NOT Outgas AND (N<0 OR N>10) THEN Out_of_range
23795 CASE 9, 13, 16, 24, 25 ! filament currents in amperes
23800 IF N<0 OR N>10 THEN Out_of_range
23801 CASE 5 ! daily-enable code
23804 IF N<0 OR N>10 THEN Out_of_range
23807 CASE 7, 10 ! filament-current take-up rates in mA/second
23810 IF N<.01 OR N>1000 THEN Out_of_range
23813 CASE 9, 11, 12 ! waits, in minutes
23816 IF N<0 OR N>480 THEN Out_of_range
23819 CASE 18 ! spike current increase/block
23822 IF N<0 THEN Out_of_range
23825 CASE 19, 20 ! blocks
23828 IF NOT Outgas AND (N3 OR N>80) THEN Out_of_range
23831 CASE 22 ! sets/block
23834 IF NOT Outgas AND (N6 OR N>80) THEN Out_of_range
23837 CASE 26 ! spike number
23840 IF POS(Response$(26), "?") THEN ! if a question-mark response, let
23843 Whichspike(0, 0, Q) ! user see which spikes are defined
23846 Response$(26)="0"
23849 Redo_screen=1
23852 P=Np
23855 GOTO Begin
23860 END IF
23861 IF N<0 OR N>10 THEN Out_of_range
23864 END SELECT
23867 !
23870 IF Exec THEN Done
23873 BEEP 660, .06
23876 GOTO Movecursor_down
23879 !
23882 !-------------------------------------------------------------
23885 Done:=FOR P=-3 TO 26
23888 IF Response$(P, 1, 2)="???" AND Use(P)=1 THEN Incomplete
23891 ON ERROR GOTO 23897
23894 Nresp(P)=URL(Response$(P))
23897 OFF ERROR
23900 NEXT P
23903 Check_values1: check that the run-variables are internally consistent
23906 IF NOT Outgas AND Nresp(1)=3 AND Nresp(4)=0 THEN
23909 DISP C1$;" IF YOU\'RE RUNNING A TRIPLE, YOU MUST WANT \"BC\"S SOME \"AC\"S\" CENTER-FI L CURRENT ! "
23912 Clunk
23915 PO=0
23920 GOTO Curswitch
23921 END IF
23924 IF NOT Outgas AND (Nresp(15)/(Nresp(14)+1.0E-9)-1)<.1 THEN
23928 DISP C1$;" MAX. BEAM MUST BE AT LEAST 10% GREATER THAN MIN. BEAM \":En$
23930 PO=15
23933 GOTO Curswitch
23936 END IF
23939 IF Nresp(17)<Nresp(14) THEN
23942 DISP C1$;" DEFAULT BEAM MUST NOT BE MORE THAN MIN. BEAM \":E$
23945 PO=17
23948 GOTO Curswitch

137
ENO IF

IF NOT Outgas AND Nresp(13)<Nresp(9) THEN
    DISP C1$: " AND OR CURRENT MUST BE GREATER THAN CURRENT-2 "; Cn$
    PO=13
    GOTO Curswitch
ENO IF

IF NOT Outgas AND (Nresp(20)<Nresp(19)) THEN
    DISP CHR$(129);" MAX# BLOCKS MUST NOT BE LESS THAN MIN# BLOCKS "; Cn$
    PO=20
    GOTO Curswitch
ENO IF

IF NOT Outgas THEN
    Parse((Response$(-1)),Isot(*),Iso$)
    ENO IF

IF Std THEN 21260
    ! check that the specified isotopes are valid for the specified element
    ! and spike (run variables only- pass for std variables).
    IF NOT Outgas THEN
        ! get element-data from disk
        Runtype$=TRIM$(Response$(-2)[1,8])
        ENTER Path$,$;Elenent$(I),Hiso,Hcoef(I),Hagnet(*),Co(*),Peak_inter,fl5,Rf 1Nu$(*),Nornal(*)
        Last_runtype$=Runtype$
        GOTO 21071
    END IF
    FOR 1=1 TO 20
        IF TRIM$(UPC$(ne(I)=yPC$(Runtype$) THEN
            ON ERROR GOTO 21053
            fiSSIGN 8Pathl TO "TYPE:INTERNH"
            ENTER $Pathl,$;Elenent$(I),Hiso,Hcoef(I),Hagnet(*),Co(*),Peak_inter,fl5,Rf 1Nu$(*),Nornal(*)
    NEXT I
ENDIF

OFF ERROR
CALL Rad_udsk(0,0)
PO=2
GOTO Curswitch
END IF

IF NOT Outgas AND (Last_runtype$="" OR ((UPC$(TRIM$(Response$(-2)[1,8]))<UPC$(Last_runtype$)))) THEN ! get element-data from disk
    Runtype$=TRIM$(Response$(-2)[1,8])
ENDIF

IF NOT Checked_els THEN ! recheck existing elements on system-disk
    Check_elements(Elenent$(I),1)
    Checked_els=1
ENDIF

IF NOT Outgas THEN
    FOR 1=0 TO Oiso ! check that each isotope is valid for specified element
        FOR I=1 TO Niso
            IF Iso(I)=Nagnet(I,1) THEN 21122
        NEXT I
        OISP Ci$;" ISOTOPE ";Iso(J);" ISN'T DEFINED FOR ELEMENT ";Runtype$;" ";Cn$
        PO=-1
        GOTO Curswitch
    NEXT J
ENDIF

IF NOT Outgas THEN
    FOR 1=1 TO Disc; check that each isotope is valid for specified element
    FOR I=1 TO Niso
        IF Iso(I)=Magnet(I,1) THEN 24122
        NEXT I
        DISP Ci$;" :ISO TO ";Iso(J);" ISN'T DEFINED FOR ELEMENT ";Runtype$;" ";Cn$
        PO=1
        GOTO Curswitch
    NEXT J
ENDIF

IF NOT Outgas THEN
    FOR 1=1 TO Disc; check that each isotope is valid for specified element
    FOR I=1 TO Disc
        IF Iso(I)=Magnet(I,1) THEN 24122
        NEXT I
        DISP Ci$;" :ISO TO ";Iso(J);" ISN'T DEFINED FOR ELEMENT ";Runtype$;" ";Cn$
        PO=1
        GOTO Curswitch
    NEXT J
ENDIF

138
21128  Spike=Hresp(26)
21130  IF Normal(1) AND NOT Spike THEN
21134    A=0
21137    B=0
21140  FOR J=1 TO Disk: check that isotopes include both the reference isotope and the normalizing isotope
21142    IF Iso(J)=Magnet(Rf,l) THEN A=1
21146    IF Iso(J)=Normal(1) THEN B=1
21149 NEXT J
21152  IF A=0 OR B=0 THEN
21155    DISP C1;" NEED BOTH: Magnet(Rf,l) AND NORMAL(1): "IN LIST OF ISOTOPES FOR ELEMENT "@runtype@" "@Cn#
21158    PO=-2
21161 GOTO Curswitch
21164  END IF
21167  END IF
21170
21173  IF Spike THEN ! check for spike on both disks
21176  IF Lastspike$=Spike$ OR Lastspike$="" THEN
21179    ON ERROR GOTO 21206
21182 FOR K=1 TO 2
21185    FOR 1=1 TO 10
21188      ASSIGN @Path1 TO "SPIKE:INTERNAL,1, W*(2-K)
21191    ON ERROR GOTO 21203
21194 ENTER @Path1,Spiket$,Sref,S_i5o(*)
21197 OFF ERROR
21200 GOTO 21221
21203 NEXT K
21206 NEXT I
21209 DISP C1;" CAN'T FIND DATA FOR SPIKE ":Spike;" ON EITHER DISK. ";@Cn#
21212 PO=-26
21215 GOTO Curswitch
21218 !
21221  Lastspike$=Spike$
21224  END IF
21227 !
21230  CALL Spike_check(Spike,Iso(*),S_i5o(*),Sref)
21233  IF Spike=0 THEN
21236    DISP C1;" SPIKE:"@resp@;" REQUIRES DATA FOR ISOTOPES ":Sref:S_i5o(1);S_i5o(2);" ";@Cn#
21239    PO=-1
21242 GOTO Curswitch
21245  END IF
21248 END IF
21251 END IF
21254 !
21257 !
21260 MAT Lastresponse$= Response$
21263 OUTPUT KBD:Clear$;
21266 SUBEXIT
21269 !
21272 Incomplete! Check for undefined but essential parameters
21275 DISP C1;" SORRY- YOU MUST ENTER A VALUE FOR EVERY "@fn@;"@fn@;"PARAMETER"
21278 Clunk
21281 WAIT 2.5
21284 PO=P
21287 P=Np
21290 Np=PO
21293 GOTO Change_params
21296 !
21299 Out_of_range=Clunk
21302 DISP C1;" VALUE IS NOT WITHIN ACCEPTABLE RANGE FOR THIS PARAMETER ";@Cn#
21305 WAIT 2.5
24300 GOTO Change_params
24311 Curswitch:F=Hp
24314 Np=Po
24317 Clunkout:Clunk
24320 WAIT 2.5
24323 Response$(Np)="??"
24326 GOTO Change_params
24329 Non_numeric:OFF ERROR  ! non-numeric response
24332 DISP FNAME("MUST HAVE A NUMERIC VALUE FOR THIS PARAMETER")
24335 Response$(Np)="??""
24338 GOTO Clunkout
24341 Return:RETURN
24344:
24347 OFF ERROR
24350 CALL Dad_udisk(1,0)
24353 GOTO Begin
24356:
24359 Load_stdvars:  ! load std-run variables from disk if not already in memory
24362 IF Stdvar(1,1)=0 THEN  ! load std-vars into memory
24365 ON ERROR GOTO 24374
24368 ASSIGN @Path1 TO "STOUflR:INTERNAL,1,1"  ! try left drive 1st
24371 GOTO 24380
24374 ON ERROR GOTO 24374
24377 ASSIGN @Path1 TO "STOUflR:INTERNAL"  ! then right drive
24380 ON ERROR GOTO 24374
24382 ENTER @Path1:Stdvar(*),Std_iso(*),Std_name(*),Stdtype(*)
24386 OFF ERROR
24389 END IF
24392 RETURN
24395 SUBEND!  ----------------------------------------------------------
24398 !
24401 !
24404 Press_graph:SUB Press_graph(MvP(*),Candump)
24407 OPTION BASE 1
24410 DIM P1(2),P2(2),I(2),Started(2),L(2),Po(2,2),Gauge$(2)
24413 OFF KEY
24416 OFF KNOB
24419 OUTPUT KOD:CHAR$(255)&CHAR$(75)
24422 GCLEAR
24425 GRAPHICS OFF
24428 DATA GO,4,9,1,625
24431 READ Xtime,Yheight,Ymin,L(*)
24434 DIM Response$(10),363
24437 MAT Response$=""
24440 Response$(1)="START GRAPHICS PRESSURE-MONITOR"
24443 IF Candump THEN Response$(2)="ENABLE AUTO AUTO-DUMP OF GRAPHICS"
24446 Response$(3)="DISABLE"
24449 Response$(4)="DOUBLE X-AXIS TIME (now "BURL$(Xtime)" mino"
24452 Response$(5)="HALVE"
24455 Response$(6)="INCREASE MAX. PRESS. (NOW IF "BURL$(9-Yheight)"
24458 Response$(7)="DECREASE"
24461 Response$(10)="RETURN TO BMC"
24464 CALL Keymenu(Response$(*)
24467 ON KEY 0 LABEL " START" GOTO Start
24470 IF Candump THEN ON KEY 1 LABEL "ON AUTO-DUMP" GOSUB Autdump
24473 IF Candump THEN ON KEY 2 LABEL "OFF AUTO-DUMP" GOSUB Offdump
24476 IF KEY 3 LABEL "DOUBLE TIME" GOTO Double_x
24479 IF KEY 4 LABEL "HALVE TIME" GOTO Halve_x
24482 IF KEY 5 LABEL "INCREASE MAX-P" GOTO Incr_p
24485 IF KEY 6 LABEL "DECREASE MAX-P" GOTO Decr_p
ON KEY 9 LABEL "9: ESCAPE" GOTO 24731
GOTO 24991
24991
24997 Rut-dump: If d-1
25000 DISP FNAME("AUTO PRINTER-DUMP")
25003 BEEP 440, .1
25006 RETURN
25009 Off-dump: If d-0
25012 DISP FNAME("CANCEL AUTO PRINTER-DUMP")
25015 BEEP 220, .1
25018 RETURN
25021 Double_x: Xtime=2*Xtime
25024 BEEP 660, .1
25027 GOTO 24440
25030 Halve_x: If Xtime>=2 THEN Xtime=INT(Xtime/2)
25033 BEEP 220, .1
25036 GOTO 24440
25039 Incr_p: Yheight=Yheight+1
25042 BEEP 660, .1
25045 GOTO 24440
25048 Decr_p: If Yheight>=1 THEN Yheight=Yheight-1
25051 BEEP 220, .1
25054 GOTO 24440
25057 R=Yheight/C100
25060 OFF KEY
25063 ON KEY 9 LABEL " ESCAPE" GOTO 21731
25066
25069 Keys(0,100,8,100,0,Xtime,Ymin,Ymax,"MINUTES","PRESSURE (Mbar)",0,-1,0)
25072 US=M3(M1(7)
25075 Gauge$1("%DM100")="US"
25078 Gauge$2("%DM20")="US"
25081 TIME=100
25084 MAT T= (D)
25087 MAT P2= (D)
25090 MAT Started= (D)
25093 R=Yheight/C100
25096 CSIZE 3
25099 FOR I=0 TO 1
25102 GOSUB Legend
25105 NEXT I
25108 OUTPUT 8;"SIMUL4":"BUS"
25111 WAIT 2
25114 FOR I=1 TO 2
25117 OUTPUT 8;"IDU"
25120 ENTER 0;Zero ! zero-value, integrated for 2 seconds
25123 Zero=Zero+Zero/2
25126 NEXT I
25129 LOOP
25132 IF P2(G1)>Xtime THEN
25135 IF G1 THEN DUMP GRAPHICS
25138 GOTO 24569
25141 END IF
25144 FOR Gauge=1 TO 2
25147 OUTPUT 8;Gauge$Gauge
25150 WAIT 1/(NOT Started(1))
25153 OUTPUT 8;"IDU"
25156 ENTER 8;P ! source pressure-reading
25159 Press=1.0E-9
25162 ON ERROR GOTO 24704
25165 IF P(Zero THEN P=Zero
21668 IF Gauge=1 THEN P2(2)<(P-Zero)<(2950)<(1020)<(8! log(source-pressure))
21671 IF Gauge=2 THEN P2(2)<7.155E-6*(P-Zero)*0.969 <(tube-
21674 time in minutes)
21677 P1(1)=P0(Gauge,1)
21680 P1(2)=P0(Gauge,2)
21683 IF Started(Gauge) THEN CALL Thickpen(P1(*),P2(*),R,L(Gauge))
21686 P0(Gauge,1)=P2(1)
21689 P0(Gauge,2)=P2(2)
21692 Started(Gauge)=1
21695 NEXT Gauge
21698 END LOOP
21701 !
21704 Legend<>AREA COLOR L(I+1),L(I+1),L(I+1)
21707 MOVE .85*Xtime*Height*.9-.08*Ymin
21710 RECTANGLE .1*Xtime,Y*Height,.85*Xtime,.85*Height
21713 MOVE .5*Xtime,.5*Height
21716 LOG R 4
21719 IF NOT I THEN LABEL "SOURCE"
21722 IF I THEN LABEL "TUBE"
21725 RETURN
21728 !
21731 OUTPUT K0D;CHR$(255)|CHR$(75):
21734 SUBEND! ---------------------------------------------------------------
21737 !
21740 !
21742 Bad_disk:SUB Bad_disk(Std,Write) ! Bad run- OR std-run variable disk-read/write
21746 DIM #K(1:3,0:1):I23
21749 DATA READ, STORE, RUN, STANDARD, FROM, ON
21752 READ #K(*)
21755 DISP FNH$ <" UNABLE TO " #K(1,Write):" #K(2,Std):" VARIABLE FILE "#K(3,Write):" DISK ">
21758 Clunk
21761 UNIT 3
21764 SUBEND ! ---------------------------------------------------------------
21767 !
21770 !
21773 Check_elements:SUB Check_elements(Element$(*),INTEGER Auto)
21776 MAT Element$=""
21779 ON ERROR GOTO 24806
21782 ASSIGN #Path1 TO "TYPE:INTERNAL"
21785 ON ERROR GOTO 24800
21788 FOR I=1 TO 30
21791 ENTER #Path1,I;Element$I()
21794 IF NOT Auto AND Element$I(I)<" THEN PRINT TAB(9);I;TAB(14);": Element$I(I)
21797 Z=Z+1
21800 NEXT I
21803 IF Z<0 THEN
21806 OFF ERROR
21809 Clunk
21812 PRINT FNH$ <" CAN'T SET ELEMENT DATA --- ">;CHR$(10)
21815 PRINT FNH$ <"PRESS">;FNH$<"CONTINUE">;FNH$<"WHEN THE SYSTEM DISK IS IN THE RIGHT-HAND DRIVE">
21818 PAUSE
21821 GOTO 24779
21824 END IF
21827 OFF ERROR
21830 SUBEND! ---------------------------------------------------------------
21833 !
21836 !
21839 Thickpen:SUB Thickpen(P1(*),P2(*),Thickness,Density)
21842 ! draw a line from point P1(x,y) to P2(x,y) with thickness defined in terms of Y-units & line-density from 0 to 1
21845 OPTION BASE 1
24048   DIM Pgran(1,2)
24051   )
24054   Pgran(1,1)=Pi(1)
24057   Pgran(1,2)=Pi(2)-Thickness/2
24060   Pgran(2,1)=Pi(1)
24063   Pgran(2,2)=Pi(2)-Thickness/2
24066   Pgran(3,1)=Pz(1)
24069   Pgran(3,2)=Pz(2)-Thickness/2
24072   Pgran(4,1)=Pz(1)
24075   Pgran(4,2)=Pz(2)-Thickness/2
24078   !
24081   AREA COLOR Density, Density, Density
24084   PEN 1
24087   MODE Pi(1), Pi(2)
24090   ON ERROR GOTO 24096
24093   PLOT Pgran(*),FILL
24096   SUBEND !  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
24099   !
24102   !
24105   Daly_enable: SUB Daly_enable(INTEGER Daly)
24108   COMM Keyboard: Cn$, Cn$, Cn$, Cn$, Cn$, Cn$, Clear$
24111   OUTPUT KBD; Clear$
24114   GRAPHICS OFF
24117   OFF KEY
24120   PRINT TABXY(1,3);"SELECT STATUS OF DALY-DETECTOR:"
24123   PRINT TABXY(1,6);FNUM$("ENABLE- DATA")
24126   PRINT TABXY(20,6);"Use when necessary for beam tuning with small beams."
24129   PRINT TABXY(1,9);FNUM$("ENABLE- TUNEUP")
24132   PRINT TABXY(20,9);"Use when necessary for beam tune-up."
24135   PRINT TABXY(20,10);"but &FNUM$("not")&" for data-taking."
24138   PRINT TABXY(1,12);FNUM$("DISABLE DALY")
24141   PRINT TABXY(20,12);"Don’t use Daly under any circumstances."
24144   PRINT TABXY(20,13);"(Use this ONLY if the Daly is malfunctioning)"
24147   OFF KEY
24150   ON KEY 0 LABEL "ENABLE- DATA" GOTO 24962
24153   ON KEY 1 LABEL "ENABLE- TUNEUP" GOTO 24968
24156   ON KEY 2 LABEL "DISABLE" GOTO 24974
24159   ON KEY 7 LABEL "ESCAPE" GOTO 24977
24166   GOTO 24956
24169   !
24172   Daly=1
24175   GOTO 24977
24178   Daly=2
24181   GOTO 24977
24184   Daly=0
24187   GOTO 24977
24190   OUTPUT KBD; Clear$
24193   SUBEND !  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
24196   !
24199   ShiftLabel: SUB ShiftLabel(Stripchart,Sample_name$(*),INTEGER Sample,Run,Auto,Nfiles) ! label the GMD-defined shifted-softkeys
24202   DIM Keys$(9)
24205   DATA MAGNET,ION OPTICS, BARREL, COLLECTORS, ISOTOPE-RATIO DATA, SPIKES, "MASS- SPEC STATUS (pressure, HU, time, contacts...)";
24208   AUTOMATIC-RUNNING VARIABLES
24211   DATA "", "CHANGING ANY FILMCHART CURRENT USING KNOB"
24214   READ Keys$(9)
24217   IF Auto THEN Keys(0)="REVERT TO MANUAL RUNNING"
24220   IF NOT Auto THEN Keys(0)="START AUTOMATIC RUNNING"
24223   OUTPUT KBD;CHR$(255)&CHR$(75); 
24226   GRAPHICS OFF
24229   PRINT XCRT
24232   PRINT TABXY(1,1); "USE THESE &FNUM$("SHIFTED")& SOFTKEYS (k0-k9) TO OBTAIN OTHER FUNCTIONS FOR:"
24235   FOR Key=0 TO 9
25025 PRINT TABXY(1,Key+3);FN$("HURLS$(Key))&" ----- ;Keys$(Key)
25028 NEXT Key
25031 PRINT TABXY(1,19):"Use KNOB with CTRL-key to change center-fil current."
25034 IF Nfils>2 THEN PRINT TABXY(1,15):" " " "CTRL-SHIFT keys to change side-fil current."
25037 PRINT TABXY(1,ISfNtis);"Barrel# " "HURLS$(Sample)" , Run# " "HURLS$(Run)" ;TAB(SD):FN$(Sample_name$(Sample))
25040 Stripchart=0
25043 END! ---- ---------------------------------------------------------------
25046 !
25049 !
25052 Gain=DEF FNGain(Counts,Constant(*),INTEGER Daly)! #### use when Daly nonlinearity is calibrated
25055 ! corrects for collector-nonlinearity
25058 IF NOT Daly THEN RETURN Counts/100
25061 IF Daly THEN RETURN Counts*Constant(1)+Counts*Constant(2)
25064 FNEND
25067 !
25070 !
25073 Locate=SUB Locate(Ok,Nuclide(*),INTEGER Niso,Coarsebin(*))! find peaks in a given magnet-interval, find peak-to-peak & half-peak intervals
25076 !
25079 OPTION BASE 1
25082 COM /General/ 2*,INTEGER Prt(*),Subflag,Auto,full_auto,Fac(*),I_t
25085 COM /Keyboard/ Cn$,CI$,Cu$,Co$,OK,STR,RETURN
25088 COM /Specs/ nx(0:1),Ions,Ze(0:1),Noise(0:1)
25091 COM /Daly/ INTEGER Mu,Daly,Mw(0:3,2);Daly_ok(0:29),Ff$(0:1,2);E(i)
25094 COM /Magnet/ Mcoef(*),INTEGER l,Reside,Peak_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse_in
25097 COM /Filaments/ Filament(*),Flil$(0:1),INTEGER Nfils,Filnum
25100 DIM Peak(3),Isotope(24),Calibr_mag(24),Calibr_isot(24),Peak_nu(100)
25103 DIM Mu(0:29),Mw(3,3),Mx(3,3),My(3,3),Mcoef(3,1),Marray(24,3),Vector(24,1)
25106 DIM Peakszize(24),Tr(3,2),Mw$(25),Mw$(24),Mw$(24)
25109 INTEGER Pr,Halfpeak_mag(0:1),Isonew(24),Magnet(0:24,2)
25112 M=IMAGE "$OFJ",42
25115 I_t=2
25118 MM Magnet= (0)
25121 Define scans;OUT KB$;Clear$;
25124 PRINT TABXY(1,9);"ENTER THE MAGNET COARSE-RANGE (0-10) AND THE MAGNET-INTERVAL (<300-9700)"
25127 PRINT TABXY(1,5);"IN WHICH YOU EXPECT TO FIND PEAKS."
25130 INPUT Coarse,Mag$Coarse,Mag$Ok
25133 IF Coarse<0 OR Coarse>10 OR (Mag$Ok-Mag$Coarse)<300 OR Mag$Ok>30000 THEN 25121
25136 OUTPUT 8;Mw$(Mu,I_t)
25139 Mu=Mu
25142 OUTPUT 0 USING "$24A,42$";"$OFK";Coarsebin(Coarse),"$OFJ";Magnet
25145 OUTPUT KB$;Clear$;
25148 IF Daly THEN
25151 PRINT "PLEASE CHOOSE THE COLLECTOR FOR THE PEAK-SEARCH SCAN..."
25154 ON KEY 0 LABEL " CUP" GOTO 25163
25157 ON KEY 1 LABEL " DAILY" GOTO 25169
25160 GOTO 25160
25163 Mu=0
25166 GOTO 25160
25169 Mu=1
25172 ELSE
25175 Mu=0
25178 END IF
25181 OFF KEY
25194 Trigger_nu=1*Mu*(2-Daly)<MUNot Mu) ! peak-height indicating a valid peak
25187 DATA 0,0,0,0,0
25190 RESTORE 25187
25193 READ n_ph,Cal_num,Ok,Mw$Onpk_num,Max_pksze
25196 Ymin=-1-Mu ! min. Y is 1.0V for cup, 0.1V for daily
25199 Ymax=2-Mu ! max. Y is 10V for cup, 100mV for daily
FOR MinmagO TO MaxmagO STEP 3  
1 scan for all peaks >trigger
220 Enter_beam(Y,Mv,1,1,1,Pr)
223 IF Pr>1 THEN 25187
225 PLOT M,LGTABS(Mv+(NOT Mu)),1-(M-MinmagO)
228 Peak(3)=Peak(2)
229 Peak(2)=Peak(1)
231 Peak(1)=Mv
232 IF On_pk THEN
233 IF On_pk THEN
235 On_pk_nufl=On_pk
236 Peak(On_pk_nufl)=Peak(1)
237 END IF
238 END IF
239 END IF
240 IF D=1 THEN Nexmagnag
241 !
242 IF NOT On_pk THEN
243 Startpk_nag=M
244 ELSE
245 Endkag:OUTPUT 8 USING M;H+2
246 NEXT H
247 ON KEY 0 LABEL "SCRH OK" GOTO Identify.iso
248 ON KEY 1 LABEL "REPEAT SCRn" GOTO Definescan
249 GOTO 25352
250 !
251 Identify.iso:0rr KEY
252 Xspred=Maxmag-Minmag
253 Yspred=Ymax-Ymin
254 W=Xspred/50  ! rectangle width
255 H=Yspred/8  ! height
256 Tr(1,1)=W  ! triangle (arrow-point) array
257 Tr(1,2)=0
258 Tr(2,1)=.W4
25382 \text{Tr}(2,2)=W/2
25385 \text{Tr}(3,1)=-1.6W
25388 \text{Tr}(2,2)=0
25391 \text{ARER PEN 1}
25394 \text{FOR} \text{I}=1 \text{TO Cal_num}
25397 \text{IF} \text{I} \text{THEN} ! \text{erase old arrow}
25400 \text{P}=1
25403 \text{ARER PEN -1}
25406 \text{GOSUB Draw Arrow}
25409 \text{ARER PEN 1}
25412 \text{END IF}
25415 \text{P}=I
25418 \text{GOSUB Draw Arrow}
25421 \text{BEER 300,1}
25424 \text{DISP} "ENTER THE MASS & NUCLIDE OF THIS PEAK (e.g. "88\text{Sr}^0 88\text{Sr}^0" 160\text{Nd}^0 160\text{Nd}^0" "87\text{Rb}^0 87\text{Rb}^0")";
25427 \text{INPUT} M,\text{Nu}(I);I1,63
25430 \text{Calibr_isot}(I)=\text{GROUNDD.INT}(M+1,3)
25433 \text{NEXT} \text{I}
25436 \text{GOSUB Peak_interval}
25439 \text{GOTO} \text{Halfpeak}
25442 !
25445 \text{Draw Arrow:} \text{Ystart}=\text{Ymin}+1.15*(\text{LET}(\text{Peaksize}(P)>0)\text{LET}(\text{Peaksize}(P)<0))\text{LET}(\text{Ypred}-\text{Ymin})! \text{start of crotch of arrow}
25448 \text{IF} \text{Ysstart-\text{Ymin}}>.8*\text{Ypred} \text{THEN} \text{Ystart}=\text{Ymin}+.2*\text{Ypred} ! \text{if arrow too high}
25451 \text{MOVE Calibr_mag(P)-W/2,Ystart+Tr(2,2)} ! \text{draw an arrow-pointer to the peak}
25454 \text{RECTANG}\text{LE} U,H,FILL
25457 \text{MOVE Calibr_mag(P),Ystart}
25460 \text{IPL0T} \text{Tr}(\text{let}),FILL
25463 \text{RETURN}
25466 !
25502 \text{Peak intervals}\text{Peak_inter}=0 ! \text{calculate the average interval for 1 a.m.u.}
25505 \text{TEXT PEN 1}
25508 \text{P}=\text{Cal_num}
25511 \text{ARER PEN 1}
25514 \text{FOR} \text{I}=2 \text{TO} \text{Cal_num}
25517 \text{Peak_inter}=\text{Peak_inter}<(\text{Calibr_mag}(I)-\text{Calibr_mag}(I-1))/(\text{Calibr_isot}(I)-\text{Calibr_isot}(I-1))
25520 \text{NEXT I}
25523 \text{Peak_inter}=\text{Peak_inter}/(\text{Cal_num} \text{1})
25526 \text{RETURN}
25529 !
25532 \text{Halfpeak}\text{Above}=0 ! \text{determine half-peak offset}
25535 \text{Mip}=0
25538 \text{CSIZE} 2
25541 \text{DISP}
25544 \text{OUTPUT 8 USING M;Calibr_mag(Maxpk_num)}
25547 \text{FOR} \text{K}=1 \text{TO} 12
25550 \text{Enter_bean(Y,Mv,1,I_t,Pr)}
25553 \text{IF} \text{Pr}=1 \text{THEN} \text{Halfpeak}
25556 \text{IF} \text{K}=1 \text{THEN} \text{Mip}=\text{Mip}+8
25559 \text{NEXT} \text{K}
25562 \text{LOGR 5}
25565 \text{FOR} \text{I}=0 \text{TO} \text{Peak_inter}
25568 \text{Halfpeak_mag}((\text{Above})=\text{Calibr_mag(Maxpk_num) Peak_inter/2+I}
25571 \text{OUTPUT 8 USING M;Halfpeak_mag(Above)}
25574 \text{WAIT} (((\text{NOT} \text{I}) \text{OR} (I=\text{Peak_inter/2})))
25577 \text{Enter_bean(Y,Mv,1,I_t,Pr)}
25580 \text{IF} \text{Pr}=1 \text{THEN} \text{25498}
25583 0=Mv-Mip/2
25586 Y=LGT(HABS(Mv),(\text{NOT} \text{Mv}))
25589 \text{MOVE} \text{Halfpeak_mag}((\text{Above}),Y<(Y)(\text{Ymin})+\text{Ymin}*(Y>\text{Ymin}))
25562  LABEL "1"
25563  DISP "FINDING HALF-PEAK LOCATIONS: *
25565  "Halfpeak_mag(above),Magnet(above),Magnet(below)
25566  IF (Above=0 AND (T(D) OR ((Above=1 AND (T(D)
25567  THEN 25563
25571  Above=1+Above
25574  DEEP
25577  IF Above=2 THEN 25586
25580  IF Above=1 THEN I=Peak_inter/2-1
25583  NEXT I
25586  DISP
25590  Rside=(Halfpeak_mag(1)-Halfpeak_mag(0))/2
25592  COUNT=0
25595  L=1
25598  MAT Daly_ok= (0)
25601  Magnet0(1,2)=Calibr_mag(Maxpk_num)
25604  Magnet0(1,1)=Calibr_isot(Maxpk_num)
25607  Coarsenag1=Coarsenag1(Coarse)
25610  REPEAT ! do a more precise measurement of half-peak setting
25613  Rside=Rside
25616  Center_peak(0,1)
25619  Enter_beam(Cts,Mag,1,1,1,1,2,2,0) ! peaktop mV
25622  OUTPUT 8 USING M=Magnet0(1,2)+Rside ! switch to above-side
25625  Enter_beam(Cts,Mag,1,1,1,1,2,2,0) ! above mV
25628  OUTPUT 8 USING M=Magnet0(1,2)-Rside ! switch to below-side
25631  Enter_beam(Cts,Mag,1,1,1,1,2,2,0) ! below mV
25634  R=(Max-Min)/2/Mag ! ratio of half-pk to pktop mV
25637  Rside=Rside-(R(1.1)(R.6)
25640  Count=1+Count
25643  UNTIL Rside=Rside0 OR Count12
25646  FOR I=1 TO Cal_num ! center all isotopes found during scan
25649  Magnet0(1,2)=Calibr_mag(I)
25652  Magnet0(1,1)=Calibr_isot(I)
25655  Coarsenag1=Coarsenag1(Coarse)
25658  M=M
25661  Mu=0
25664  OUTPUT 8:Mag(0,2)
25667  IF M THEN WAIT 6
25670  Center_peak(0,11)
25673  Calibr_mag(I)=Magnet0(1,2)
25676  NEXT I
25679  OUTPUT 8:Mag(Mu,2)
25682  GOSUB Peak_interval
25685  DISP
25688  ! solve for either linear or least-squares quadratic curve to describe magnet-settings as a function of isotope
25691  IF Cal_num2 THEN ! linear solution if only two peaks
25694  Mcoef(2,1)=Calibr_mag(2)-Calibr_mag(1)/Calibr_isot(2)-Calibr_isot(1)
25697  Mcoef(1,1)=Calibr_mag(1)-Mcoef(2,1)*Calibr_isot(1)
25700  Mcoef(3,1)=0
25703  ELSE #peaks=3
25706  ! solve for least-squares quadratic curve to describe magnet-settings as a function of isotope
25709  REDIM Xarray(Cal_num,3), Yvector(Cal_num,1)
25712  FOR I=1 TO Cal_num
25715  Xarray(I,1)=I
25718  Xarray(I,2)=Calibr_isot(I)
25721  Xarray(I,3)=Calibr_isot(I)2
25724  Yvector(I,1)=Calibr_mag(I)
25727  NEXT I
25730  MAT ka= TRN(Xarray)
25733  MAT kb= ka*Yarray
25736  MAT xc= INV(kb)
25739  MAT xd= xc*ka
MAT Ncoef= Xd¥vector
END IF

FOR I=1 TO Cal_nu fluctuations
check for improperly identified nuclides by checking residuals
Residual=Mcoef(1,I)+Mcoef(2,I)*Calibr_isot(I)+Mcoef(3,I)*Calibr_isot(I)^2*Calibr_mag(I)

IF ABS(Residual)>C THEN
  DISP FNH:="AT LEAST 1 ("B=Val=(Calibr_isot(I)>A)" OF YOUR ISOTOPE IDENTIFICATIONS IS WRONG")
  Clunk
  WAIT 1
  GOTO Identify_iso
END IF

NEXT I

Add_iso: add isotopes that weren't found during scan
OUTPUT KBD: Clear$;
GRAPHICS OFF
MAT Isonew= (0)
PRINT USING "H/,((,/,(="ISOIOPE, NUCLIOE OF PEAKS IFHN$(1)"
PRINT TABXY(1,0); "(press CONTINUE when done)"
PRINT TABXY(1,1);:
FOR New_iso=0 TO 23-Cal_nu;
  I=0
  II=New_iso;1
  DISP "NEW ISOTOPE, NUCLIOE #:B=Val=(II)";
  INPUT I,NuO$(II)
  IF I=0 THEN 25853
    M=ncoef(1,I)+Hcoef(2,I)*I+Hcoef(3,I)*2^I
    IF M>300 OR M<700 THEN
      OISP "ISOTOPE ";I;" DOESN'T FALL WITHIN THE SPECIFIED CORRECTION RANGE";
      Clunk
      WAIT 3
      GOTO 25805
    ELSE
      Isonew(I)=II
      PRINT "NEW ISOTOPE, " I; NuO$(II);!
    END IF
  NEXT I
  NEXT New_iso

FOR J=1 TO Cal_nu ! eliminate duplicate isotopes
  MINI=23-Cal_nu
  FOR K=I TO New_iso ! find lowest new-isotope
    IF Isonew(K) flNO Isonew(KXHiniso THEN
      INI50=K
      Nuclide$(I)=NuO$(K)
      MINI=K
    END IF
  NEKT K
  NEKT J
  NEKT New_iso

MAT Nuclide$= ""
Mise=0
FOR I=1 TO 24 ! sort the new & scanned isotopes & place in magnet array
  Miniso=1.000
  Knin=0
  FOR K=1 TO New_iso ! find lowest new-isotope
    IF Isonew(K) AND Isonew(K)<Miniso THEN
      Miniso=Isonew(K)
      Nuclide$(I)=NuO$(K)
      Knin=K
    END IF
  NEKT K
  NEKT New_iso
  IF Calibr_isot(J) AND Calibr_isot(J)<Miniso THEN
    Miniso=Calibr_isot(J)
    Nuclide$(I)=Calibr_isot(J)
  NEKT J

148
26096 PRINT USING "2/,K,/,K,";FNH$<"*** Can't get spike-data from disk ***")
26099 Clunk
26102 WAIT 2
26105 GOTO 26033
26108 END IF
26111 SUBEND ! -----------------------------------------------
26114 !
26117 !
26120 Signa=IF FNSigna(Sig$) ! Extract signa-value from signa-string- negative if in parens
26123 Paren=POS(Signa,"(")  
26126 ON ERROR GOTO 26132
26129 RETURN SGN(5-Paren)*VR(Signa(Paren+1))
26132 RETURN 0
26135 FNEND ! -----------------------------------------------
26138!
26141!
26144 Magknob: SUB Magknob(INTEGER Coarse0, Mag0, Coarsebin(*), On_daly) ! scan magnet with KNOB
26147 OPTION BASE 1
26150 INTEGER Mag
26152 CM /Keyboard/ Cn$;Ci$;Cb$;Cu$;Q$;Clear$
26156 OUTPUT 8 USING "1fi,1Z";$OFK",Coarsebin(Coarse)
26158 GOSUB Display
26159 IF On_daly THEN
26162 PRINT USING "18/,K,/,K,?/";"Sorry, you can't do knob-controlled scanning while on the daily. ","use the semi-auto magnet-scan routine (k6) instead."
26165 Clunk
26166 WAIT 3
26171 GOTO Escape
26174 END IF
26177 Speed=100
26180 Coarse=Coarse0
26183 Mag=Mag0
26186 GOSUB Display
26189 PRINT TAB(1,0);"Magnet-control knob enabled. Use softkeys to escape, to change coarse-range."
26192 PRINT TAB(1,10);"Or to change response of knob."
26195 OFF KEY
26199 ON KEY 9 LABEL "_escape" GOTO Escape
26201 ON KEY 0 LABEL " coarse up" GOSUB Upcoarse
26204 ON KEY 1 LABEL " coarse down" GOSUB Downcoarse
26207 ON KEY 3 LABEL " speed up" GOSUB Speedup
26210 ON KEY 4 LABEL " slow down" GOSUB Slowdown
26213 A=KNOBK
26216 GOTO Knob
26219 !
26222 Upcoarse=Coarse+1\Coarse
26225 IF Coarse\11 THEN Mag=0
26228 IF Coarse\10 THEN Coarse=10
26231 OUTPUT 8 USING "4A,4Z";"ok",Coarsebin(Coarse)
26234 GOSUB Display
26237 BEEP 1000, .05
26240 RETURN
26243 Downcoarse=Coarse-Coarse\1
26246 IF Coarse\10 THEN Mag=9999
26249 IF Coarse\0 THEN Coarse=0
26252 OUTPUT 8 USING "4A,4Z";"ok",Coarsebin(Coarse)
26255 GOSUB Display
26258 BEEP 150, .05
26261 RETURN
26264 Speedup=Speed*2*Speed
26267 GOSUB Display
26270 BEEP 150, .05
Nuclide(I) = Nu(I)

END IF

NEXT J

IF Knin AND NOT Jnin THEN Isonew(Knin) = 0

IF Miniso > 1000 THEN

Magnet(I,1) = Miniso

Niso = I + Niso

END IF

NEXT I

FOR I = 1 TO Niso ! define Magnet values from quadratic curve

Magnet(I,2) = Hcoef(3,1) * Magnet(I,1) + Hcoef(2,1) * Magnet(I,1) + Hcoef(1,1)

NEXT I

FOR I = 1 TO 3

Mcoef0(I) = Hcoef(I,1)

NEXT I

GRAPHICS OFF

OUTPUT KB0; Clear$;

MagnetO = Magnet

MAT MagnetO = Magnet

Ok = 1

SUBEND !

SUBEND !

Whichspike: SUB Whichspike(INTEGER Spike, Get_spike, Subflag) ! get spike data from disk

OPTION BASE 1

COM /Spikedrun, /Spikedrun_ratio(*), Spike_ratio(1), Natural_ratio(1), INTEGER Idf(*), Spikedrun_iso(*), Spikedrun_ref

OFF KEY

OFF KNOB

Subflag = 0

IF NOT Get_spike THEN CALL Dispikes

FOR Drive = 1 TO 0 STEP -1

ON ERROR GOTO 26024

ASSIGN @Pathl TO "SPIKE:INTERNAL,1,"; " elle"(Drive)

GOTO 26030

NEXT Drive

GOTO 26087

IF NOT Get_spike THEN

Spikc = 0

PRINT "WHICH SPIKE DO YOU WANT TO LOOK AT? (press CONTINUE to escape)", Spike

IF Spike = 0 THEN SUBEXIT

END IF

ON ERROR GOTO 26087

ENTER @Path1, Spike, Spike_ref, Spikedrun_ref, Spikedrun_iso(*), Spikedrun_ratio(*), Idf(*), Spike_ratio(*), Natural_ratio(*)

OFF ERROR

OUTPUT KB0; CHAR$(255) & CHAR$(75);

PRINT USING "/4,k,0,0,2a,b,k,k,2a,4": "SPIKE", Spike, " NAME: ", Spike,

PRINT USING "/10x,k,2a,3,k,bx,k,bx,k,10": "REFERENCE ISOTOPE: ", Spikedrun_ref, " RATIO ISOTOPES", "SPIKE-RATIO", "NAT

URAL-RATIO"

FOR I = 1 TO 3

IF (Spikedrun_iso(I) > 0) THEN PRINT TAB(3),Spikedrun_ratio(I); TAB(23), Spike_ratio(I);

IF I < 3 THEN PRINT TAB(3), Natural_ratio(I)

NEXT I

PRINT

IF Get_spike THEN SUBEXIT

GOTO 26033

OFF ERROR

If NOT Get_spike THEN
26276 Slowdown=Speed/2
26279 \$SUB Display
26282 BEEP 200,.05
26286 RETURN
26291 Display:"COARSE-RANGE":Coarse;TAB(20);"MAGNET =":Mag;TAB(55);"SPEED =":INT(Speed)
26294 RETURN
26297!
26300 Knob:ON Knob .05 GOTO Scan
26303 GOTO 26303
26306 Scan:S*Knob
26309 Mag=Mag*S-Speed/200
26312 IF Mag<0 THEN Mag=0
26315 IF Mag>9999 THEN Mag=9999
26318 IF flag<0 THEN Mag=0
26321 GOSUB Display
26324 IF Mag=0 THEN BEEP 100, .05
26327 IF Mag>9999 THEN BEEP 2500, .05
26330 GOTO 26303
26333!
26336 Escape:OUTPUT 8 USING "2(4R,42);""OFK",Coarsebin(CoarseO),"OFJ",Mag0
26339 OUTPUT KBD;Clear$!
26342 SUBEND!---------------------------------------------
26345!
26349!
26351 Dispose:SUB Dispose(Runclass$,Nuclide$(x),Normal(*),Interfered(*),INTEGER Coarse,Ref,Huo,Magnet(*),Miso,Inv)
26354 OPTION BASE 1
26357 COM /Keyboard/ Cn$,Cl$,Cn$,Cu$,Cn$,Cn$,Cn$,Clear$
26360 OUTPUT KBD;Clear$;
26363 FOR J=0 TO 24 STEP 8 ! display defined isotopes & magnet-values
26366 FOR I=J+1 TO 80
26372 IF DHiso THEN 26381
26375 PRINT TRIH$(Huclide$(I))a"U="UflL$(Hagnet(I,1)=" ";
26378 PRINT
26381 NEXT J
26384 IF (Niso/SOINHNiso/S)) THEN PRINT
26387 CALL Magprt(Coarse,Magnet(*),Miso)
26390 PRINT"NORMALIZING RATIO: ";
26393 IF Normal(1)=0 THEN PRINT "NONE"
26399 IF Normal(1) THEN PRINT URL$(Ref&"","URL$(Normal(1))&"","Normal(2)
26402 PRINT USING ",k,k,k,k,k,k,k,k,k,k,k;\"MONITOR ISOT.\",\"INTERFERING ISOT.\",\"RATIO \"&n$
26405 FOR I=1 TO 1
26408 IF I=1 AND Interfered(I,1)=0 THEN PRINT TAB(4);"NONE";TAB(21);"NONE";TAB(45);"-----
26411 IMAGE 4#30,15#30,15#,:","","","","","","","","","","","","",K
26414 IF Interfered(I,1) THEN PRINT USING 26411:Interfered(I,1),Interfered(I,2),Interfered(I,2),Interfered(I,1),Interfered(I,3)
26417 NEXT I
26420 PRINT
26423 SUBEND!---------------------------------------------
26426!
26429!
26432 Superduperclunk:SUB Superduperclunk ! panic sound
26435 ON KEY 9 LABEL " ESCAPE" GOTO 26456
26438 FOR I=1 TO 5
26441 Superclunk
26444 FOR J=1 TO 5
26447 Clunk
26450 NEXT J
26453 NEXT I
26456 SUBEND !---------------------------------

26459 !
26462 !
26465 Tconvert:DEF FNclock_12$<time$> ! convert from 24-hour to 12 hour clock
26468 H=URL(time$) | hour
26471 C1=POS(time$,":")
26474 C2=POS(time$[1];C1,"-"">C1
26477 time$=time$[1];C2-1
26480 SELECT H
26483 CASE [12
26486 RETURN URL$(H-12);time$[C1];" P.M."
26489 CASE 12
26492 RETURN time$" P.M."
26495 CASE >0
26498 RETURN time$" A.M."
26501 CASE 0
26504 RETURN "12";time$[C1];" A.M."
26507 CASE ELSE
26510 RETURN "????????"
26513 END SELECT
26516 FNEND !-----------------------------------

26519 !
26522 !
26525 Isomag:DEF FNIsomag(Mcoef*,Isotope) ! return fine-magnet value for any isotope
26528 RETURN Mcoef(1)+Isotope*Mcoef(2)
26531 FNEND !-----------------------------------

26534 !
26537 !
26540 Newel:SUB Newel(Type,Interference*,Normal*,Nuclide*,Element*,Runclass$,Early_exit,INTEGER Coarsebin*,Niso,Ref,Rf,Inv,

26548 OPTION BASE 1
26546 COM /General/ Z$,INTEGER Prtr(*),Subflag,Auto,Full_auto,Foc(*)
26549 COM /Specs/ Mix(0:1),Ions(0:1),Noise(0:1)
26552 COM /Magnet/ Mcoef(*),INTEGER L,Resid,Pek_inter,Coarsemag(0:1),Magnet(0:24,2),Coarse
26555 COM /Daly/ INTEGER Mu,Mx,Mx$(0:3),Daly ok(0:20),Ff$(0:1,2)
26558 COM /Keyboard/ Cn$,Cl$,Cb$,Cu$,Q$,Clear$
26561 COM /Filaments/ Filaent(*)
26564 DIM Runclass$_[10]$,Nuclide$[0:21]$,Mcoef(3),Normal(2),Interference(1,3)
26567 INTERGER Niso,Magnet(0:24,2),Coarsemag(0:1),Peak_inter,Resid,Ref,Inv,Mx0

26569 OFF KEY
26570 OFF KBO
26576 OUTPUT KBD;Clear$;

26579 Early_exit=0
26582 Type$=Type
26585 PRINT "PRESS "$CH$("x")" TO CHANGE MAGNET VALUES ONLY,";CHR$(10)
26588 PRINT "PRESS "$CH$("x")" TO CHANGE RUNNING-DATA ONLY (normalization, interferences...),";CHR$(10)
26591 PRINT "PRESS "$CH$("x")" TO CHANGE BOTH.";
26594 PRINT USING "3","PRESS "$CH$("x")" TO RETURN TO EMS"

26597 Magonly=0
26600 Datenly=0
26603 ON KEY 8 LABEL " MAGNET ONLY" GOTO 26610
26606 ON KEY 1 LABEL "RUN-DATA ONLY" GOTO 26621
26609 ON KEY 2 LABEL " DATA" GOTO 26627
26612 ON KEY 9 LABEL " ESCAPE" GOTO Early_exit
26615 GOTO 26615
26618 Magnonly=1
26621 GOTO 26627
26624 Datenly=1
26627 OUTPUT KBD;Clear$;
26630 OFF KEY
26633 ON KEY $ LABEL " ESCAPE" GOTO Early_exit
26636 IF NOT Detonly THEN
26639 CALL Locate(Ok,Nuclide$(0),Niso,Coarsebin(*)! peak-locating subprogram
26642 IF NOT Ok THEN Newel
26645 CALL Nu(Ans$,0),Hv,0,Mu,I1;
26648 FOR I=1 TO 20
26651 IF UPC$<element$(I)(1,2)="RE" THEN 26720
26654 NEXT I
26657 Magnet(0,1)=0
26660 Nuclide$(0)=""
26663 GOTO 26756
26666
26669 ASSIGN @Path1 TO "TYPE=INTERNAL" ! get Re-element data from disk in right-drive
26672 ENTER @Path1,ISRunclass$,Niso,INcof$(0),Magnet$(0),Coarsemag$(0),Peak_inter,Ref_1,Re

26675 IF HBS(Hv-HvO)>6 THEN
26678 FOR I=1 TO Niso
26681 IF Magnet(I,1)=187 THEN Magnet(0,2)=Magnet(I,2)
26683 Coarsemag(0)=Coarsemag(I)
26684 Nuclide$(0)="Re"
26687 GOTO 26756
26691 END IF
26694 END IF
26702 NEXT I
26705 Magnet(0,1)=0
26708 Magnet(0,2)=0
26711 Nuclide$(0)=""
26714 END IF
26717 END IF
26720 OFF KEY
26723 INPUT "ENTER REFERENCE ISOTOPE FOR AUTOMATIC RUN? (0 to escape)",Ref
26726 IF Ref=0 THEN Early_exit
26729 FOR RF=1 TO Niso
26732 IF Magnet(Rf,1)=Ref THEN 26750
26735 NEXT RF
26738 PRINT USING "K,K,";Ref," ISN'T DEFINED IN YOUR LIST OF ISOTOPE- PLEASE TRY AGAIN.",Ref
26741 Clunk
26744 GOTO 26720
26747
26750 L=Ref
26753
26756 IF NOT Magonly THEN
26759 MAG Normal= (0)
26762 Dispel(Instance$(0))
26765 LOOP
26768 PRINT USING "K,K,";"ENTER THE NUMBER (1-20) AND NAME (currently-defined elements shown above)","FOR THE NEW ELEMENT-TYPE (enter 0,0 to escape)
26771 INPUT Type,Runclass$(0,10)
26774 EXIT IF Type>=0 AND Type<=20
26777 PRINT USING "/,1DK,K,";"ERROR** ELEMENT-TYPE NUMBER MUST BE (10-20)
26780 Clunk
26783 END LOOP
26786 IF Type=0 THEN Early_exit
26789 Type0=Type
26792 OUTPUT KBD;Clear$;
26795 DISP "CALCULATE RATIOS AS ";$URL$(Ref);"/"$ISOTOPE (+1) OR ISOTOPE/"$URL$(Ref);" (-1);
26798 INPUT Inu
26801 IF Inu=0 THEN Early_exit

153
26801 IF (INuOl) AND (INuOl-1) THEN 26795
26807 MAT Normal = 0
26810 PRINT TAB(1,10); "WHAT ISotope, IF ANY, DO YOU WANT TO USE TOGETHER WITH "VAL$(Ref)" FOR"; TAB(1,11); "FRACTIONATION NOR-
MALIZATION?"
26813 PRINT TAB(1,13); "(EXAMPLE: 88 FOR Sr; TAB(1,15); "(press "FINISH"("CONTINUE")" if fractionation-normalization isn’t poss-
ible)"
26816 INPUT Normal(1)
26819 OUTPUT RBB; Clear$
26822 IF Normal(1) THEN
26825 FOR I=1 TO Niso
26828 IF Normal(I)>=Magfctd(I,1) THEN 26816
26831 NEXT I
26834 Clunk
26837 DISP "*** Normal(I):" ISN’T IN YOUR LIST OF ISOTOPES ***
26840 WAIT 3
26843 GOTO 26810
26846 DISP "ENTER THE STANDARD VALUE FOR THE NATURAL "VAL$(Ref)="/"VAL$(Normal(I))" RATIO";
26849 INPUT Normal(2)
26852 END IF
26855 Element$(Type) = Runclass$
26858 MAT Interfer$ = 0
26861 PRINT USING "1d,K,/,K"; "ENTER THE NUMBER (UP TO 4) OF ISOBARIC INTERFERENCES", FOR THE DATA-TAKING ISOTOPES:"
26864 INPUT Isobar_interf
26867 IF Isobar_interf=0 THEN 26545
26870 IF Isobar_interf>4 THEN 26661
26873 PRINT USING "1d,K,/,K"; "(THE MONITOR isotope is the isotope whose peak-size is used to calculate the", "peak-size of the I-
TERFERING isotope."
26876 PRINT USING "2d,K,/,K"; "(THE INTERFERING isotope is an isotope whose mass coincides with the mass of the", "isotope that you’re
really interested in."
26879 PRINT USING "K,/,K"; "For example, in Strontium runs, Rb-85 is a MONITOR isotope for the INTERFERING", "isotope Rb-87, which
interferes with Sr-87."
26882 FOR I=1 TO Isobar_interf
26885 DISP "MONITOR ISOTOPE "VAL$(I)"", INTERFERING ISOTOPE "VAL$(Interfere(I,2))" ?
26888 INPUT Interferes(I,1),Interferes(I,2)
26891 A=0
26894 B=0
26897 FOR J=1 TO Niso
26900 IF Interferes(J,1)=Magfctd(J,1) THEN A=1
26903 IF Interferes(J,2)=Magfctd(J,1) THEN B=1
26906 NEXT J
26909 IF A=0 OR B=0 THEN
26912 DISP "*** BOTH ISOTOPES MUST EXIST IN THE LIST OF ISOTOPES FOR THIS ELEMENT ***"
26915 Clunk
26918 WAIT 3
26921 GOTO 26885
26924 END IF
26927 DISP "ENTER THE EXPECTED VALUE FOR THE "VAL$(Interferes(I,2))"="/"VAL$(Interferes(I,1))" RATIO";
26930 INPUT Interferes(I,2)
26933 NEXT I
26936 ELSE
26939 Type=Type0
26942 END IF
26945 IF NOT Only$ THEN HuO=Hu
26948 !
26951 ASSIGN @Path1 TO "TYPE:INTERNAL"
26954 OUTPUT @Path1,Type=Runclass$,Niso,Mass$( ),Covarag$( ),Peak_inter,Aside,Rf,Huclidea( ),Normal( ),INu,Interferes( ),
Hu0
26957 ASSIGN @Path1 TO *
26960 SUBEXIT
26963 !
26966 Early_exit:OUTPUT KB0:Clear$;
26969 Early_exit=1
26972 PRINT USING "7/,K,/,K";"THE FOLLOWING DISPLAY, YOU MUST REQUEST A NEW ELEMENT TO RETURN TO THE BMC, WITHOUT PROBLEMS.
26975 PRINT USING "3/,18X,K";"Press CONTINUE when ready."
26978 PAUSE
26981 SUBEND! __________________________________________
26984 !
26987 !
26990 Display=Sub Display(Element$(x)); ! display all element-types defined on disk
26993 OUTPUT KB0:CHR$(255)\CHR$(13);\GRAPHICS OFF
26996 PRINT TABXY(7,1);"NUMBER";TAB(20);"TYPE"
27000 ON ERROR GOTO 27032
27005 FOR I=1 TO 30
27008 IF Element$(I)="" THEN
27010 COUNT=COUNT+1
27013 Side=(Count>15>)
27015 IF Count>15 THEN PRINT TABXY(17,1);"NUMBER";TAB(55);"TYPE"
27020 XPOS=9+39*Side
27023 YPOS=COUNT/2 IS*Side
27026 PRINT TABXY(XPOS,YPOS);I;TAB(4*YPOS);" ";Element$(I)
27029 END IF
27032 NEXT I
27035 OFF ERROR
27038 PRINT TABXY(1,10);
27041 SUBEND! __________________________________________
27044 !
27047 !
27050 H=DEF FNH$(Input$) ! put a string in inverse-video
27053 RETURN CHR$(129)\"Input$"\CHR$(129)
27056 FNEND ! __________________________________________
27059 !
27062 !
27065 Un=DEF FNN$(Input$,OPTIONal P) ! underline a string
27068 IF NRPR=2 THEN
27071 Printer=2
27074 ELSE
27077 Printer=0
27080 END IF
27083 IF Printer THEN RETURN CHR$(27)\"bD\"Input$\CHR$(27)\"bD"
27086 IF NOT Printer THEN RETURN CHR$(132)\Input$\CHR$(128)
27089 FNEND ! __________________________________________
27092 !
27095 !
27098 Bl=DEF FNBl$(Input$) ! put a string in inverse, blinking video
27101 RETURN CHR$(131)\"Input$"\CHR$(129)
27104 FNEND ! __________________________________________
27107 !
27110 !
27113 Barnum=DEF FNBarnum(Barpos) ! return the barrel# for a barrel-position
27116 IF Barnum=1 THEN RETURN 1 ! (the barrel# is the number stamped on the
27119 Barnum=27-5*Barpos ! barrel: the barrel-position is the sequen-
27122 WHILE Barnum<16 ! tial position of a filament-block location
27125 Barnum=Barnum+16 ! in the barrel.
27128 END WHILE
27131 RETURN Barnum
27134 FNEND ! __________________________________________
27137 !
27140 !
27143 Barpos=DEF FNBarpos(Barnum) ! return barrel-position for a barrel#
tcal = DEF FNM(Analog_curr) ! Change analog current in amps to the
 corresponding digital value necessary to to obtain that current
 10/22/81 test indicates: True current = 0.88*(meter current)
 so (True Current) = (Digital Output) * 1.12

RETURN MAX(INF(Analog_curr=1000:100),0)

RETURN Barpos

CHANGE analog current in atips to the corresponding digital value necessary to obtain that current.

10/22/81 test indicates: True current = 0.88*(meter current)
so (True Current) = (Digital Output) * 1.12

RETURN HflX <HflX<flanalog_curr>1000-180),0)

IF Number THEN

RETURN INT(Number) * H

ELSE

RETURN H

ENDIF

SUB Wiggle_barrelINTEGER Barrel_position,fil[,REAL Flag(*),M$(*))

! wiggle the barrel a bit to try & regain contacts

CALL Brl(Barrel_position,G,1.0,G) ! rotate barrel up G units, wait .G sec
CALL Brl(Barrel_position-B,1.0,B) ! " " down B units, wait .B sec
CALL Brl(Barrel_position,Flag(Fil),1.0) ! " " up 0 units to original
CALL Brl(Barrel_position,Flag(Fil),1.0) ! " " position, wait .G sec, check contacts

SUBEND

SUB Mag_iso:DEF FNM_iso(Mcoef(*),INTEGER tiagnet)

RETURN isotope as function of fine-magnet setting

IF Mcoef(3) THEN

RETURN (<Mcoef(2)>SQR(Mcoef(2)^2-4*Mcoef(3)*Mcoef(1)-Magnet)))/2*Mcoef(3))

ELSE

RETURN (Magnet-Mcoef(1))/Mcoef(2)

ENDIF

SUBEND

SUB Name:DEF Name(Sanple_name$(*),INTEGER Estbar<*>)

! If the barrel-contacts were tested, enter the run-names for all of the samples in the barrel

SUBEND

SUB END

OK TO enter new names, OK TO return to BMC.

SUBEND

ON KEY 0 LABEL " START" GOTO 27329
ON KEY 9 LABEL " ESCAPE" GOTO 27393
27320 MAT Sample_name$ = ("")
27332 OUTPUT KBD:Clear$;
27335 CONTROL KBD:1 ! caps lock ON
27338 FOR I=1 TO 16
27341 IF (Sample(I,1)) THEN
27344 Count=Count
27347 C(I)=Count
27350 GOSUB Enter_name
27353 END IF
27356 NEXT I
27359 Edit:1=0
27362 PRINT TABBY(1,17);"PLEASE ENTER THE NAME TO BE ASSIGNED TO THE RUN FOR BARREL# "&$1688" :": "$18765
27365 PRINT TABBY(1,18);"(press "AFROM"("CONTINUE")&" when done) "
27368 INPUT I
27371 IF DO BNB K17 THEN
27374 GOSUB Enter_name
27378 GOTO Edit
27380 END IF
27383 SUBEDIT
27386 Enter_name:PRINT TABBY(1,17);"PLEASE ENTER THE NAME TO BE ASSIGNED TO THE RUN FOR BARREL# "&$1688" :": "$18765
27389 PRINT TABBY(1,18);"(press "AFROM"("CONTINUE")&" to escape)"
27392 LINPUT Sample_name$[I](1,50)
27395 Sample_name$[I]=TRIM(Sample_name$[I])
27398 IF Sample_name$[I]="" THEN Edit
27401 PRINT TABBY(1,C(I))="&
27404 IF Count=I THEN PRINT TABBY(1,C(I));"BARREL#"
27407 PRINT TABBY(9,C(I));URL$[I]="";TAB(12):Sample_name$[I]
27410 RETURN
27413 SUBEND ! ----------------------------------------------- ---------------
27416 !
27419 !
27422 Show_names:SUB Show_names(Sample_name$(*))
27425 OUTPUT KBD;CHR$(255);CHR$(75);
27428 FOR I=1 TO 16
27431 IF Sample_name$[I]="" THEN
27434 Count=Count
27437 IF Count=I THEN PRINT "BARREL#";
27440 PRINT TAB(9):I:TAB(15):Sample_name$[I]
27443 END IF
27446 NEXT I
27449 IF Count=0 THEN PRINT USING "?/1X,K:";"*** NO SAMPLE NAMES HAVE BEEN DEFINED ***"
27452 SUBEND
27455 !
27458 !
27461 Autohelp:SUB Autohelp(Param,Standard) ! July 17, 1981, 12:56 pm
27464 OPTION BASE 1
27467 DIM S$[10](160)
27470 SELECT Param
27473 !
27476 CASE -1
27479 RESTORE 27492
27482 DATA "$88$8$These are the isotopes that you wish to take isotope-ratio data for, excluding any isotopes used only for"
27485 DATA "isobaric interference monitoring.$The first isotope in the list must be the reference isotope.$If the element that"
27488 DATA "you choose for this run has a ratio defined for fractionation-normalization, then your list of isotopes must includ e the"
27491 DATA "two isotopes used for the normalization.$You may request up to 8 isotopes."
27494 N=4
at a rate of \( R_{\text{fl}} \) = 30 nA/second, followed by a wait of 1 minute. Exceptions: (1) the first 0.7 Amps may be assigned a \( R_{\text{fl}} \) of 30 nA/second, (2) if the filament was already preheated to a given current, the assigned \( R_{\text{fl}} \) will be doubled until that current is reached. The sample-filaments are then taken to a current of \( C_{\text{fl}} \) at a rate of \( R_{\text{fl}} \) = \( C_{\text{fl}} \) nA/second, followed again by a wait of 1 minute. The beam is then centered and focused several times, followed by a wait of \( D_{\text{fl}} \) minutes just before the start of data-taking.

The sample-filaments are then taken to a current of \( C_{\text{fl}} \) at a rate of \( R_{\text{fl}} \) = \( C_{\text{fl}} \) nA/second, followed again by a wait of 1 minute. The beam is then centered and focused several times, followed by a wait of \( D_{\text{fl}} \) minutes just before the start of data-taking.

If \( R_{\text{fl}} \) reaches the abort current on the sample-filament(s) [center if a single, sides if a triple] while trying to achieve either the MIN.-BEAM or the DEFAULT BEAM, the run is aborted and automatic-running sequence goes on to the next run.

The intent of the DEFAULT parameters is to allow you to specify both a desired beam-size (MIN.-BEAM to MAX.-BEAM), and a "not the best quality but I'll take it anyway" beam-size (DEFAULT-BEAM to MAX.-BEAM)." Sticky note: To force \textsc{rnrlyst} to try and get the MIN. BLOCKS at a beam-size of greater than the MIN. BEAM instead of just the...

After each block, the sample-filament current (center-fil if a single-fil run, side-fils if a triple-fil run) will be increased by \( \text{FIL INCR./BLOCK} \) amperes.

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)

\( \text{MIN.-MAX blocks, sigmas} \)
27500 CASE -3
27503 IF Standard THEN SUBEXIT
27506 RESTORE 27509
27509 DATA "The BARREL # is the location in the barrel (1-16) of the sample to be run."
27512 N=1
27515 !
27518 CASE 2
27521 RESTORE 27524
27524 DATA "$The ELEMENT is the element for running, such as Pb, Sr, U, NdO...$It must be defined on the DATA disk and include the isotopes that"
27527 DATA "you wish to take isotope-ratio data for.$To find out which elements are defined, press the CHANGE ELEMENT key during the BMC."
27530 DATA "$To use a set of STD-RUN variables, enter the number of the STD-RUN enclosed in asterisks"'
27533 DATA "(for example, *12*). If you don't remember which STD-RUN you want, enter *?* for a summary of the"
27536 DATA "defined STD-RUN variables.$If you just want to outgas the sample, though, enter either OUTGAS (to outgas in the"
27539 DATA "running position) or P-OUTGAS (to outgas in preheat position)."
27542 N=6
27545 !
27548 CASE 0
27551 IF Standard THEN
27554 RESTORE 27557
27557 DATA "$$$$$$$The STD-RUN NAME is just the name (up to 10 characters) that you wish to assign to this standard-run."
27560 ELSE
27563 RESTORE 27566
27566 DATA "$$$$$$$The RUN NAME is just the name (up to 50 characters long) that will be assigned to this run."
27569 END IF
27572 N=1
27575 !
27578 CASE 1! single/triple
27581 RESTORE 27584
27584 DATA "$$$$$$$Enter 1 if the sample is a single-filament load, 3 if a triple (or double) filament load."
27587 N=1
27590 !
27593 CASE 2,3,1! foc-iso, CF-only beam, initial CF
27596 RESTORE 27599
27599 DATA "$#FOR TRIPLE FILAMENT RUNS ONLY#$"'
27602 DATA "The center filament is taken to an initial current of INITIAL CF AMS, then the FOCUSING ISOTOPE (if nonzero) is centered "
27605 DATA "and focused.$If nonzero, the FOCUSING ISOTOPE beam is then adjusted to the intensity"
27608 DATA "corresponding to the CENTER-FIL. Beam by varying the center-filament current.$"
27611 DATA "If the FOCUSING ISOTOPE is 137 (Rhenium), its intensity will be maintained at the CENTER-FIL. Beam throughout the run.$"
27614 DATA "$If the FOCUSING ISOTOPE is NOT Rhenium, then no attempt to control its beam$"
27617 DATA "$size will be made once the side-filaments are turned on.$"
27620 N=7
27623 !
27626 CASE 5! Daly
27629 RESTORE 27632
27632 DATA "$An entry of 0 indicates that the Daly is nonfunctional, and should not be used under any circumstances.$"
27635 DATA "$An entry of 1 indicates that the Daly can be used for both beam tuneup and data-taking.$"
27638 DATA "$Data will be taken with the Daly in all cases where all of the isotopes for data-taking are less than 35 nU in size.$"
27641 DATA "$An entry of 2 indicates that the Daly can be used for beam tuneup, but not under any circumstances for data-taking.$"
27644 N=4
27647 !
27650 CASE 6,7,8,9,10,11,12! current-rate-wait
27653 RESTORE 27656
27656 DATA "The sample filament(s) (center if a single-filament, sides if a triple-filament) are taken to a current of CURRENT-1
DATA "$SSS$S$S$IF the growth-rate of the beam exceeds MAX. GROWTH (%/MIN.) and the MIP (most intense peak) beam is more than n"
DATA "the MIN. BEAM, the sample-filament current will be reduced by 2.5%. SSI $S$flf the MIP beam is less than the MIN. BEAM"
DATA "value, ANALYST will tolerate up to twice the MIN. BEAM value.$Typical values are 2% - 3% per minute for$
DATA "Sr and Nd. A value of 100%/minute is equivalent to ignoring the beam growth."
N=1
CASE 24,25
RESTORE 27827
DATA "$SSSSSS$Just after the running sample is rotated into position, the filaments of the preheat sample (at a barrel of$
DATA "one greater than the running sample) will be taken to currents of PREHEAT CF (R) amperes (center filament) and$
DATA "PREHEAT SF (R) amperes (side filament)."
N=3
CASE 26
RESTORE 27840
DATA "$SSSSSS$NORMSPIKE8 is the number of the spike that the sample was spiked with (possible only for elements with an internal"
DATA "ratio for fractionation-normalization, such as Sr, Nd, Sn...). For a valid input, you must have already defined$
DATA "a spike with this number.$Enter a value of 0 if the element is fractionation-nornalizable, or if you haven't defined$
DATA "such a spike. If you enter a valid NORMSPIKE8, both the spike:sample ratio and the radiogenic-isotope ratio (if an$
DATA "$v, corrected for$
DATA "$SSSSSS$SET is a single sequence of step-scanning over the data-taking isotopes during a block.$Vou can specify fron 1 to 10 SETS in a block.$
DATA "$SSSSSS$BLOCK is a complete cycle of data-taking, including peak-centering, backgrounds, and isobaric-interference monitoring$"
27962  DATA "$You may specify from 1 to 80 blocks."
27965  N=2
27968  CASE 6,7,8
27971  RESTORE 27971
27974  DATA "$The BEAM WINDOW indicates the allowable beamsize for the Most Intense Peak (MIP). If the MIP beamsize is found to be"
27977  DATA "not within the specified BEAM WINDOW after a block of data, ANALYST will turn the filament current up or down until"
27980  DATA "the beamsize does fall within the BEAM WINDOW limits. The BEAM WINDOW is defined by the MINIMUM BEAM and MAXIMUM BEAM"
27983  DATA "values. If MINIMUM BEAM=0 and MAXIMUM BEAM=10, then no restrictions will be placed on the beam size."
27986  DATA "$ANALYST will not take the sample-filament current past the MAXIMUM FILAMENT CURRENT however, regardless of the BEAM WINDOW values."
27989  N=5
27992  CASE 9
27995  RESTORE 27995
27998  DATA "$After a block of data-taking, ANALYST determines that the rate of beam growth exceeded the MAXIMUM BEAM-GROWTH value."
28001  DATA "value, ANALYST will turn the sample-filament current down by about 2.1%. A value of 100 is equivalent to disregarding the rate of beam growth. Typical useful values for Sr or Nd runs are 2% - 3% per minute."
28004  N=5
28007  CASE 11
28010  RESTORE 28010
28013  DATA "$After the last block requested by the NUMBER OF BLOCKS value, ANALYST will turn the sample and preheat filament currents down to the values specified here. The default values are simply the filament-currents in effect at this time, so"
28016  DATA "if you don't change the values, no change in the filament currents will occur when the data blocks are done."
28019  DATA "$The 4 values pertain to the Sample/Center-filament, Sample/Side-filaments, Preheat/Center-filament, and Preheat/Side-filaments, respectively."
28022  N=6
28025  CASE 14
28028  DATA "$If the Daly-Enable Code is 0, the Daly will be used for neither data-taking, nor beam tune-up under any circumstances. This code should be used ONLY if the Daly is malfunctioning. A Daly-Enable Code of 1 allows for beam-tuneup of small beams, and also for data-taking when all of the data-taking isotopes are less than 35 millivolts. A Daly-Enable Code of 2 allows the Daly to be used for beam tune-up of small beams, but NOT, under any circumstances, for data-taking."
28031  N=5
28034  CASE 15
28037  DATA "$If you answer Y(es), the peak-jump graphics that appear during data-taking will be dumped to the printer at the end of each block."
28040  DATA "$If you answer N(o), no such graphics-dump will occur."
28043  N=2
28046  END SELECT
28049  !
28052  REDIM $S%(N)
28055  READ $S%(x)
28058  Helpscreen($S%(*)
28061  SUBEND !
28091 |
28094 |
28097 Pattern=SUB Pattern | Hewlett-Packard "Pen" program from demo disk.
28100 INTEGER Polygon, Polygons, Side, Sides, Pen
28103 OFF KMOD
28106 Polygons=20
28109 Sides=3
28112 Pen=0
28115 ALLOCATE INTEGER X(0:Polygons-1,1:Sides), Y(0:Polygons-1,1:Sides)
28110 ALLOCATE INTEGER Dx(Sides), Dy(Sides)
28121 RANDOMIZE
28124 GINIT
28127 PLOTTER IS 3,"INTERNAL"
28130 GRAPHICS ON
28133 WINDOW 0,511,0,389
28136 PEN Pen
28139 ON KBO ALL GOTO Escape
28142 OFF KEY
28145 OFF KMOD
28148 a=0
28151 FOR Side=1 TO Sides
28154 X(0,Side)=RND*512
28157 Y(0,Side)=RND*389
28160 PLOT X(0,Side), Y(0,Side)
28163 NEXT Side
28166 IF Sides<2 THEN PLOT X(0,1), Y(0,1)
28169 GOSUB Define_deltas
28172 FOR Polygon=1 TO Polygons-1
28175 PENUP
28178 FOR Side=1 TO Sides
28181 Temp=X(Polygon-1,Side)+Dx(Side)
28184 IF Temp<511 THEN
28187 Dx(Side)=0
28190 ELSE ! (it's not off right side)
28193 IF Temp<0 THEN Dx(Side)=-Dx(Side)
28196 END IF ! (off right side?)
28199 X(Polygon,Side)=X(Polygon-1,Side)+Dx(Side)
28202 Temp=Y(Polygon-1,Side)+Dy(Side)
28205 IF Temp<389 THEN
28208 Dy(Side)=0
28211 ELSE ! (it's not off top)
28214 IF Temp<0 THEN Dy(Side)=-Dy(Side)
28217 END IF ! (off the top?)
28220 Y(Polygon,Side)=Y(Polygon-1,Side)+Dy(Side)
28223 PLOT X(Polygon,Side), Y(Polygon,Side)
28226 NEXT Side
28229 IF Sides<2 THEN PLOT X(Polygon,1), Y(Polygon,1)
28232 NEXT Polygon
28235 New=0
28238 ON CYCLE 10 GOSUB Define_deltas
28241 LOOP
28244 IF New=0 THEN
28247 Previous=Polygons-1
28250 ELSE
28253 Previous=(Previous+1) MOD Polygons
28256 END IF ! (new=0?)
28259 PENUP
28262 PEN -Pen
28265 DISABLE
28268 FOR Side=1 TO Sides
! PLOT X(New,Side),Y(New,Side)
! NEXT Side
! IF Sides>2 THEN PLOT X(New,1),Y(New,1)
! PEN Pen
! FOR Side=1 TO Sides
! Temp=X(Previous,Side)+Dx(Side)
! IF Temp>511 THEN
! Dx(Side)=Dx(Side)
! ELSE
! IF Temp<0 THEN Dx(Side)=Dx(Side)
! END IF
! X(New,Side)=X(Previous,Side)+Dx(Side)
! TCnp=Y(Previous,Side)+Dy(Side)
! IF TCnp>389 THEN
! Dy(Side)=Dy(Side)
! ELSE
! IF TCnp<0 THEN Dy(Side)=Dy(Side)
! END IF
! Y(New,Side)=Y(Previous,Side)+Dy(Side)
! END LOOP
! END LOOP
! FOR Side=1 TO Sides
! Dx(Side)=RND+3/2
! IF RND<.5 THEN Dx(Side)=Dx(Side)
! Dy(Side)=RND+3/2
! IF RND<.5 THEN Dy(Side)=Dy(Side)
! NEXT Side
! RETURN
! Intercept:OFF KBD
! GRAPHICS OFF
! SUBEND
! Escape:OFF KBD

! Change_hv=SUB Change_hv ! change some or all of the element HV values
! OPTION BASE 1
! DIM Mcoor(2),Normal(2),Interferes(4),Runclass(0:21),Run(0:21),
! INTEGER Nclass(0:21),Magnet(0:21,2),Coarsemag(0:1),Peak_inter,Side,HU,Inv,Old_hv,New_hv
! ! INPUT KBD:CHR$(255)&CHR$(75):
! OFF KEY
! OFF KBD
! PRINT TAB(1,0):"This subprogram will change the High-Voltage settings for all elements whose"
! PRINT USING "K,":"present HV settings are within some specified range of a specified new value."
! PRINT "Please enter the new High-Voltage value and the RANGE from this value that"
! PRINT USING "K,":"the old HV values must fall within in order to be acceptable for the change."
! PRINT USING "K,":"Example: 0010,20 --- the HV of all elements whose present HV lies between","7000 and 8030 volts"
! PRINT USING "K,":"will be changed to 8010."
! PRINT USING ",K":"(enter 0,0 to escape to CMC)"
! ! INPUT New_hv,Range
! IF New_hv=0 AND Range=0 THEN SUBEXIT
! IF New_hv>9999 OR Range>200 THEN
CLUNK
28143 DISP "*** THE NEW HU SHOULD BE <10,000 AND THE RANGE SHOULD BE <200 ***"
28147 \  "WAIT 3"
28151 \  "GOTO 28439"
28155 \  "END IF"
28160 \  "ON ERROR GOTO Bad_disk"
28164 \  "ASSIGN #Path1 TO ""TYPE:INTERNAL."
28168 \  "OFF ERROR"
28172 \  "PRINT "Searching system disk..."
28176 \  "FOR Type=1 TO 20"
28180 \  "ENTER Path��Type;Runclass$;Niso,Mcconf(*);Magnet(*);Coarsemag(*);Peak_inter,Raide,Rl,Carbon(*);Normal(*);Inv,Interfere(*)
28184 \  ,Old_hu"
28188 \  "Count=1:Count"
28192 \  "IF Old_hu<>New_hu AND ABS(New_hu-Old_hu)<Range THEN"
28196 \  "OUTPUT Path身Type;Runclass$;Niso,Mcconf(*);Magnet(*);Coarsemag(*);Peak_inter,Raide,Rl,Carbon(*);Normal(*);Inv,Interfere(*)
28200 \  ,New_hu"
28204 \  "PRINT "ELEMENT ";Runclass$;Type(20);"HU Changed from ";Old_hu;"to";New_hu"
28208 \  ELSE"
28212 \  "PRINT "ELEMENT ";Runclass$;Type(20);"HU = ";Old_hu;"-- no change"
28216 \  END IF"
28220 \  "NEXT Type"
28224 \  "OFF ERROR"
28228 \  "IF Count=0 THEN Bad_disk"
28232 \  "WHooop"
28236 \  "PRINT "&FNH$""CONTINUE"&" when ready to return to BMC..."
28240 \  "PAUSE"
28244 \  "GOTO Exit"
28248 \  "good"
28252 \  "Bad_disk:Clunk"
28256 \  "PRINT USING ""2/JOH,K"";FNH$""******** CAN'T READ SYSTEM DISK *******"
28260 \  "WAIT 4"
28264 \  "GOTO Exit"
28268 \  "good"
28272 \  "Exit;ASSIGN #Path1 TO x"
28276 \  "SUBEND"
28280 \  "Helpscreen:SUB Helpscreen(Input_string$(*)*,OPTIONAL In)
28284 \  "prints out a message on the CRT with linefeeds at appropriate breaks"
28288 \  "between words. Also inserts one linefeed for each $ symbol, and in-
28292 \  "dents each line after a linefeed by indent spaces. Tricky, huh?"
28296 \  "DIM Large_string$(1600),Line_string$(80),Rev_line$(80)
28300 \  "good"
28304 \  "OUTPUT KBD;CHAR$(255)&CHAR$(75):"
28308 \  "PRINTER IS EXT"
28312 \  "IF NFPR=2 THEN Indent=In"
28316 \  "IF NFPR=1 THEN Indent=0"
28320 \  "Large_string$="'
28324 \  "build a single large string out of the input-string array"
28328 \  "FOR I=1 TO SIZE(Input_string$,1))
28332 \  "Large_string$=Large_string$TRIM$(Input_string$(I))&"'
28337 \  "NEXT I"
28341 \  "Maxlen=80"
28622 If Linefeed THEN
28623 PRINT USING "K,,,R$T$",LH,Maxlen)\Line_string$(L,Linefeed-L)
28624 IF L0 <= 80 THEN Done
28625 Maxlen=80
28626 END IF
28627 ELSE
28628 Rev_line$=RCUS(Line_string$) \ find place for wordbreak
28629 Sr=POS(Rev_line$," ")
28630 IF L0<=Maxlen THEN
28631 S=L-Sr
28632 ELSE
28633 S=0
28634 END IF
28635 PRINT RPJ$","",BO-Maxlen)&Line_string$(S,]i
28638 END IF
28641 UNTIL L=0
28644 Done=DISP "press CONTINUE to go on, DUMP ALPHI for a hard copy"
28645 PAUSE
28712 \ help screen for first manual-data request for
28715 a fractionation-normalizable element.
28718 OPTION BASE 1
28721 QIM S$(?X160)
28724 !
28727 SELECT Param
28730 \ Sample name
28733 RESTORE 28736
28736 DATA "%%The SAMPLE NAME is just the name (up to 50 characters) that will be assigned to this run."
28739 M=1
28742 CASE 3 \ Spike Number
28745 RESTORE 28748
28748 DATA "%%The SPIKE NUMBER will apply only if you have spiked your sample with a spike that has been defined on the DATA d
isk.$"
28751 DATA "If you have done so, then enter the number of that spike (the number that you assigned to it when you defined it on t
he disk)."
28754 DATA "Your isotope-ratio data will then be calculated corrected for both fractionation and spike isotopes (if there is a
radiogenic isotope)"
28757 DATA "for this element), and the Spike Sample ratio will also be calculated.$"
28760 DATA "If this is not such a spiked sample, though, just enter 0 as the spike number.$"
28763 DATA "To see which spikes are defined on the DATA disk, enter a Question-Mark instead of a spike-number.$"
28766 DATA "To define a new spike, exit to the BMC and then press SHIFT KS."}
28769 M=7
28772 CASE 6 \ Normalize ratios to 1st block?
28775 RESTORE 28778
28778 DATA "%%If the isotope ratios of this sample will not have the natural ratios for the normalizing isotopes (for example, if the sample"
28781 DATA "is a spike, or if it's a sample spiked with an undefined spike), then you should answer Y(es)."
28784 DATA "The data will then use the ratio for the normalizing isotopes of the first block for fractionation-normalization of a
11 subsequent"
DATA "Blocks. You will then be able to do weighted averages on all the blocks from within ANALYST, without being bothered by the"
DATA "drift of fractionation. Of course, the normalizing ratio will be wrong, but you can correct just the average rather
than each"
DATA "block for this bias. For unspiked samples or for samples spiked with a spike that you have defined on the DATA disk ("
SHIFT-h5 during the"
DATA "BMC), though, you should answer Y(a)."
DATA "N=7"
END SELECT
RCDIH S$<N)
RERO S$<*)
Helpscreen<S$(»
SUBEND
SUBEND
SUBEND
Magscan_help:SUB Magscan_help
OPTION BASE 1
DIM S$(6)$([16])
DAPR"The magnet-scan starts at the mass-position of the Start Scan isotope, and goes up-mass at the Scan Speed to the E
nd Scan"
DATA "isotope. The scan is shown by real-time graphics, where the maximum beam that will fit on the graph is given by the M
ax. Beam on"
DATA "Graph value. The Y-axis of the graph can be either LINEar or LOGarithmic."
DATA "If you want to scan over a region that does not lie within the coarse-magnet range for the element currently in use,"
DATA "enter a different coarse-magnet range (0-10) for the Coarse-Magnet Range. The FORM screen will then ask you to enter
the."
DATA "starting and ending positions of the scan in magnet units (0-9999) rather than mass units."
READ S$(*)
Helpscreen(S$(»
SUBEND
SUBEND
SUBEND
Delta=SUB Delta(Delta$,Mean,Last_mean,Signan,Last_signan)
! determine if difference in ratios is significant & construct Delta$
Percent_change=100*(Mean/Last_mean-1)
Within_theor=xABS(Percent_change)*2*SQRT(Sigmean^2+Last_signan^2))
Delta$[1,0]=TRIIM(VAL#(0#))%TRIM(Deltas)
Delta$=TRIM(Deltas)
IF Percent_change>0 THEN Delta$=""&Delta$[1,7]
IF Within_theor THEN Delta$=""&Delta$[1,9]"
SUBEND
RETURN
SUBEND
FiltestsDEF FHIltest(INTEGER Mi1s,REAL Filflag) ! valid-contact test?
SELECT Mi1s
CASE 1
RETURN Filflag=1 OR Filflag=3
CASE 2
RETURN Filflag=3
CASE 1
RETURN 0
END SELECT
FNEND
RETURN
FNEND
DEF FHIltest(N! is N even? return 1 if even, 0 if odd
RETURN ABS(ABS(N!)/2-INT(ABS(N!)/2)))<1.E-20
FNEND
SUBEND