```
algol<
Multifit
with outer zone , homogeneous 10.10.68



begin
real pi, mod, recmo, G, k, m0, a, c, Ls, Rs, Ms, lTes, L, Te, R, lTe, lL, lR, L0, LV,
     eps, kappa, arb, arb1, U1, U3, cnsqm,epsp,epsn,Lneu, eps4,epsg, delt, deltg, del
     testg, testn, q, T, rho,  P, T4, Xny, qk, Xk, qy, Xy, beta, age, M, X, Y, Z,
     XomH, YomHe, ZomA, eomua, eomue, tb, rb, sec,
     A0, A1, A2, A3, A4, A5, A6, A01, A02, A08, A09, Atim4, A1p5,
     delta, delatm, dd, ddr, dda, gr, alder, fsum, gsum, F, dpar, step, sres, mf,
     gacc, alfa, l, Hp, U, si, t1, ltb, lps, slT, slP,
     C1, C2, C3, C4,dummy,exp30;
integer s, t, i, j, nQ, t4, tælqx, ar, tgr, tgrn, argr, grgæt,energi, nfp, fit, nm, l
        UNRW, UNWW, UNWP, UNWD,
        g, intil, autm, mtx4, ccore, nul, to, tre, fire, fem, num, rat, lat, ka,
        counts, mode, ztype, ilt, ilp, tsg, tsd;
integer array zonenum[-1:20];

array k1, y[1:5], fstep,Q, XQ[0:99], XM[1:4], km[1:1,0:17,0:20],
      fp, lHJU, ddaa, lkap[0:30, 0:24], ara, rpfac, alter[1:4];
comment punch off;
integer procedure lyn; lyn:=0;
integer procedure read integer; begin integer a;input(a);read integer:=a end;
real procedure read real; begin real a; input(a); read real:=a end;

boolean procedure kbon; kbon:= false;
procedure writechar(a); value a; integer a; if a=0 then writetext(‡< ‡);
integer procedure select(a); value a;integer a; select:=a;

comment punch on;*

real procedure antlog(r); value r; real r; antlog:= exp(r/mod);

real procedure log(r); value r; real r; log:= mod×ln(r);

procedure LLGAUSS(n,R,A,U,delta); value n,delta; integer n; real delta;
array R,A; label U;
begin integer i,j,k,h; real a,b;
k:= 0; Do it again: k:= k+1; a:= A[k,k]; h:= k; for i:= k step 1 until n-1 do
begin b:= A[i+1,k]; if abs(a)>abs(b) then go to S; h:= i+1; a:= b; S: end;
if abs(a)<delta then go to U; if h=k then go to eliminate;
for j:= k step 1 until n+1 do begin b:= A[k,j];A[k,j]:= A[h,j]; A[h,j]:= b end;
eliminate: for i:= k+1 step 1 until n do
begin b:= -A[i,k]/a; for j:= k+1 step 1 until n+1 do A[i,j]:= A[i,j]+A[k,j]×b end;
if k<n then go to Do it again; next unknown: b:= 0; for j:= n step -1 until k+1 do
b:=R[j]×A[k,j]+b; R[k]:= (A[k,n+1]-b)/A[k,k]; k:= k-1;
if k>0 then go to next unknown
end LLGAUSS;




begin integer k;
real t, p, t1; procedure fpline(A); array A;
        begin for j:= 0 step 1 until (if k<ilp then k else ilp)  do A[i,j]:= read rea
        for j:= ilp+1 step 1 until k do dummy:= read real end;
UNRW:= 16; UNWW:= 17; UNWD:=9; UNWP:= 33; A02:= 0.2;
nul:= 0;  to:= 2; tre:= 3; fire:= 4; fem:= 5; Atim4:= 10-4;
```

```
NEWFP: dummy:=  select(UNWW); writetext(‡<
  t1, ilt:=  ‡); t1:= read real; arb:= read real;
dummy:=
select(UNRW);  ltb:=read real; ilt:=read integer;  lps:=read real; ilp:=read integer;
             X:= read real; Y:= read real; Z:= read real;
             slT:= read real; dummy:=  read real; slP:= read real; dummy:=  read real
label 1: t:= read real; p:= read real;
         if t < t1-Atim4 then begin
         ilp:= read integer; for j:=1 step 1 until 4×(ilp+1) do dummy:= read real;
         goto label 1 end;
ltb:= t; lps:= p; ilt:=arb; ilp:=24;
for i:= 0 step 1 until ilt do
begin  if i≠0 then begin t:= read real; p:= read real;end;
k:= read integer;
if abs(t-ltb-slT×i)+abs(p-lps-slP×i)>Atim4 then begin
dummy:=
select(UNWW); writetext(‡<
Error in fp-table‡); dummy:=  lyn;  go to NEWFP end;fpline(fp); fpline(lkap); fpline(
end reading of fp tables;

ny tab: dummy:=  select(UNWW); dummy:=  lyn; dummy:=  select(UNRW); tb:= read real; r
ka:= read integer; t:= read integer; nm:= read integer;
for i:= 1 step 1 until nm do begin XM[i]:= read real;
for j := 0 step 1 until t do begin
if abs(read real-tb-j×A02)>Atim4 then begin writetext(‡<
TABELFEJL‡); go to ny tab end;
dummy:=
read real; dummy:=  read real; for k:= 0 step 1 until ka-1 do
km[i,j,k]:= read real end end læs kappa;
end tablereading; dummy:=  select(UNWW);
writetext(‡<
STELLAR MODELS
MULTIFIT WITH OUTER ZONE
l/Hp:=  ‡); alfa:= read real; dummy:= select(UNRW);


nfp:= read integer;
begin
array korr, rltp, rltpg, rltpgg[0:nfp-1,1:4], res[1:nfp,1:4],
 fmas[0:nfp], drdp[1:nfp,1:4,1:4];

switch SW:= frem, choose, ldef, tdef;

procedure MERSN;
begin real x0, ho3, eps, eq,gstep,tstop; boolean last;
array y0, k3, k4, k5 [1:5];

procedure CRGR;
begin integer t; real gtg, teste, qe, qg, qn;
t:= nul;
gtg:= -testg; qe:= q;  teste:= testn; qg:= x0; writechar(29);
iter: if abs(testn)>Atim4∧t<15 then
begin if t>12 ∨ kb on  then
begin if ztype<tre then WATM else   SKRIV; writecr; write(‡-dd.dddd‡, testn, teste, t
qn:=(teste×qg-testg×qe)/(teste-testg);
for i:= 1 step 1 until nm do y[i]:= y0[i]; q:= x0; DIFF(k1);
RKM(qn-x0); t:= t+1; if t=14 then writechar(62); DIFF(k4);
if sign(testn)=sign(teste) then begin qe:= qn; teste:= testn end
else begin qg:= qn; testg:= testn end; go to iter end; if ztype<tre then WATM else  S
if mode = 1 then fstep[counts]:= qn-x0;
if abs(q-fmas[intil])>2₁₀-7 then last:= false;
testg:= testn:= abs(testn)×sign(gtg);
end procedure CRGR;
```

```
procedure RKM (h); value h; real h;
begin ho3:= h/3;
for i:= 1 step 1 until nm do y[i]:= k1[i]×ho3+y0[i]; q:= x0+ho3;
DIFF(k3); for i:= 1 step 1 until nm do y[i]:= (k1[i]+k3[i])/2×ho3+y0[i];
DIFF(k3); for i:= 1 step 1 until nm do y[i]:= (k1[i]×.375+k3[i]×1.125)×ho3+y0[i];
q:=x0+h/2; DIFF(k4);
for i:= 1 step 1 until nm do y[i]:= (k1[i]×A1p5-k3[i]×4.5+k4[i]×6)×ho3+y0[i];
q:= h+x0; DIFF(k5);
for i:= 1 step 1 until nm do y[i]:=((k1[i]+k5[i])/2+k4[i]×2)×ho3+y0[i];
end RKM;


procedure RKMER(h); value h; real h;
begin x0:= q; for i:= 1 step 1 until nm do y0[i]:= y[i];
RKM(h);

eps:= A0; for i:= 1 step 1 until (if nm>4 then 4 else nm) do
begin eq:= abs(k1[i]×A02-k3[i]×A09+k4[i]×A08-k5[i]×A01)×rpfac[i];
if eq>eps then eps:= eq end; eps:= eps×abs(ho3);
if delta≥eps then tsg:= tsg+1 else  begin comment step unacceptable;
if kb on then writetext(‹<.›);  tsd:= tsd+1;
for i:= 1 step 1 until nm do y[i]:= y0[i]; q:= x0; go to QR end;
end RKMER;

if ztype<tre then begin  tstop:= fmas[nfp]/mod;
TOT6: RKMER(step);  DIFF(k1); if sign(testn)= sign(testg)∧abs(testn)>Atim4 then
testg:= testn else CRGR; if kb on ∨ mode = 4 then begin WATM; if ztype = to∧T>7₁₀5  th
if ztype=1 then go to (if y[1]<A01 then QR else finis);
if ztype=2 then go to (if y[tre]< tstop then QR else TILQ);

TILQ: delta:= delta/Atim4;
IGEN: if abs(y[tre] - y0[tre])>₁₀-7 then begin
   DIFF(k1); if kb on then WATM;
   RKMER((x0-q)×(y[tre]- tstop)/(y[tre] - y0[tre]));
 if abs(y[tre] - tstop)>5₁₀-7 then go to IGEN
 end;
delta:= delta×Atim4;
go to finis;
end integration with ztype<3;

nm:= if mode=4 then fem else fire; go to R;
try last: last:= true; gstep:=step; step:= fmas[intil]-q; go to R1;
R: if abs(step)>.05 then step:= sign(step)×.05; if abs(fmas[intil]-q)≤abs(step) then
                            last:= false;
R1:
RKMER(step);

if mode = 1 then begin counts:= counts+1; fstep[counts]:= step end;
DIFF(k1); if sign(testn)= sign(testg)∧ abs(testn)>Atim4 then
testg:= testn else CRGR;
if kb on ∨ mode=4  then begin SKRIV; end;
if last then begin step:=gstep; go to finis; end;
if eps = A0 then begin step:= A2×step; goto R end;
QR: step:= (delta/eps)↑A02×A08×step; go to (if ztype<tre then TOT6 else R);
finis: end MERSN;

procedure WIZ;
begin

y[1]:= ln(y[1]×Rs×R); arb:= y[to];
y[to]:= antlog(lL)×Ls/L0;
y[fire]:= q;  F:= A1; q:= arb; ztype:= tre; DIFF(k1); if mode=4 then s:=select(UNWP);
write(‹-d.dddd›, log(rho), log(kappa), log(ddr), F); if mode=4 then dummy:=  select(s
```

```
y[1]:= exp(y[1])/Rs/R;
y[to]:= q;
q:= y[fire];
ztype:= to; DIFF(k1);
end WIZ;


procedure DIFF(K); array K;
begin integer mt,mr,tad;
if ztype<tre then

begin integer t,p;
real tm, pn, m, n, cmm1, cm0, cm1, cnm1, cn0, cn1, x2, FP, PmPr;
real procedure INTP(A); array A;
INTP:= (A[t-1, p-1]×cnm1+A[t-1, p]×cn0+A[t-1, p+1]×cn1)×cmm1+(A[t, p-1]×cnm1+A[t, p]×
        +A[t, p+1]×cn1)×cm0+(A[t+1, p-1]×cnm1+A[t+1, p]×cn0+A[t+1,p+1]×cn1)×cm1;


procedure MIXING;
begin real U8,U16,fsi,fm,hs,s2;
l:= alfa×Hp; U:= C3/eomua×Hp×T4/(T×exp(arb/mod)×FP×P/PmPr×kappa×rho×l×l);
U8:= 8×U/9; U16:= 2×U8; hs:= U8×(ddr-dda);
la: s2:= si×si; fsi:= s2×si+U8×s2+U16×U×si-hs;
   fm:= 3×s2+U16×(si+U); si:= si-fsi/fm;
if abs(fsi/hs)>1_10-6 then go to la; dd:= dda+si×si+2×U×si;
end mixing procedure;
 P:= exp(q);
if ztype=1 then begin T:=Te×(0.5+0.75×y[1])↑0.25;y[3]:=ln(T) end else T:=exp(y[3]);
tm:=(y[3]×mod-ltb)/slT; t:= entier(tm);
pn:= (q×mod - (lps + tm×slP))/slP; p:= entier(pn);
if ilp > 2×p then p:= p+1;
m:= tm -t; n:= pn - p ;
if t < 1 ∨ t > ilt -1 ∨ p < 1 ∨  p > ilp - 1 then CONTROL;
cmm1:= m/2×(m-1); cm0:= 1-m×m; cm1:= m/2×(m+1);
cnm1:= n/2×(n-1); cn0:= 1-n×n; cn1:= n/2×(n+1);
T4:=T×T; T4:= T4×T4; PmPr:= P-a/3×T4; FP:= INTP(fp); rho:= PmPr/FP/C2/T;
kappa:= exp(INTP(lkap)/mod);
dd:= ddr:= C1×kappa×P/(y[2]×T4);
if ztype=1 then K[1]:= kappa×P/gacc else
begin x2:= y[1]×y[1];
Hp:= x2×P/(y[2]×rho×gacc);
dda:=INTP(ddaa); testn:=A1/ddr -A1/dda;
if ddr>dda then begin arb:= INTP(lHJU); MIXING; end;
K[1]:= -Hp/(R×Rs);
K[2]:= -C4×x2×rho×Hp;
K[3]:= dd
end end
else begin real V,r3,T616,T626,T646,T6,t,r,f0,f1,n,p,p1,mnul,m1,Pc,ks;
X:= Xk;
if X<0 then X:= 0; Y:= 1-X-Z;
XomH:= X/1.00801; YomHe:= Y/4.0028; ZomA:= Z/16.942;
eomue:= XomH + A2 × YomHe + Z/A2;
eomua:= XomH + YomHe + ZomA;
T:= exp(y[tre]); P:= exp(y[fire]); V:= T↑3; T4:= V × T;
beta:= A1 - a/A3 × T4/P;

t:= P×beta/(T×k/m0);arb:= sqrt(V); rho:= t/(eomua+eomue×F);
ks:= X+7×Z+A3; r3:= cnsqm×ks↑3;
DEGL: Pc:= sqrt(r3×rho/T)×rho;
t:= (P×beta+Pc)/(T×k/m0); f1:= rho;
f0:= rho/(9.0476_10-9×arb)×eomue; r:= (f0 - 30)/30;
F:= ((((0.109744×r-0.165376)×r+0.151420)×r-0.462698)×r+3.296236)×r+5.187420;
rho:= t/(eomua + eomue×F);
```

```
if abs((f1-rho)/rho)>10-5 then go to DEGL;

V:= V/rho; dda:= A1/(A4-A1p5×beta×beta/(A4-A3×beta)); t:= mod×y[tre];
if X<110-7 then eps:= eps4:= Xny:= 0 else
begin T6:= T×10-6; T616:= T6↑(1/6); T626:= T616↑2; T646:= T626↑2;
ks:= 0.133210 9×sqrt(ks/V);
arb:= 2.37810-16×(1+0.00922×T626)×exp(-A3×ks+99.96/T626 -30)×exp30×(X/(1-X-Z))↑2;
arb1:= sqrt(1+arb); arb:= (arb1+arb/A2-1)/arb;
epsp:= 4.19910 6×arb×(1+0.0123×T626+0.00781×T646+0.00067×T6)/T646×exp(ks-33.809/T626)×r
if T6<12 then eps4:= 0 else
eps4:= 4.45810 27×(1+0.00274×T626)/T646×exp(7×ks-152.31/T626)×rho×X×Z;
Xny:= 0;
arb:= 1/(1+2.42810 16×X/(1+X)×(1+0.0041×T626)/T616×exp(A4×ks-102.64/T626));
eps:= epsp×(0.980+(0.964×arb-1.044)/(arb1+A3))+eps4×0.936; epsn:=epsp+eps4-eps;
if T6<7 then eps:= if T6>6.5 then eps×(T6-A6) else eps/A2; end;
epsg:= 0; eps:= eps+epsg;
r:= mod×ln(rho); arb:= (r-rb-A3×(t-tb))×A5; arb1:= (t-tb)×A5;
mt:= entier(arb1); mr:= entier(arb);
if mr<nul then begin mr:= nul; KTABS end;
if mr>ka-2 then begin mr:= ka-to; KTABS end;
tad:= 0; p:= 1;
f0:= +km[tad+1,mt,mr]×p;
f1:=+km[tad+1,mt,mr+1]×p;
n:= arb-mr; mnul:= f0+n×(f1-f0);
f0:=+km[tad+1,mt+1,mr]×p;
f1:=+km[tad+1,mt+1,mr+1]×p;
m1:= f0+n×(f1-f0); n:= arb1-mt;
kappa:= exp((mnul+n×(m1-mnul))/mod);
r:= exp(y[1]); r3:= r↑3;
K[1]:= U1×M/(r3×rho); K[to]:= Ms×M×eps/L0; K[fem]:= Ms×M×epsn/L0; K[fire]:= -G×(Ms×M/
ddr:= U3/M×kappa×y[to]×L0×P/(q×T4); testn:= A1/ddr-A1/dda;
K[tre]:= K[fire]×(if ddr<dda then ddr else dda);
end diff for innerzone;
end procedure DIFF;


procedure UD(A); array A;
begin integer i ; writecr;
write(‹-ddd.ddddd›,A[1],A[2],A[3],A[4]);
end;
procedure UD2(A,j); value j; integer j; array A;
begin integer i;
for i:= 1 step 1 until fire do ara[i]:= A[j,i]; UD(ara)
end for UD2;

procedure KTABS; begin s:=select(UNWW); writetext(‹<
   ext›); dummy:=  select(s); end;

procedure CONTROL;
begin dummy:= select(16); writetext(‹<
  fp table too small  logT = ›); write(‹-d.dddd›,y[3]×mod);
      writetext(‹<  logP = ›); write(‹-d.dddd›, q×mod); writetext(‹<  input new atm d
      go to rep
end;

procedure WATM;
begin if mode=4 then s:=select(UNWP); writecr; if ztype=2 then
 write(‹-dd.dddd›, q×mod, log(T), log(rho), log(kappa), log(ddr), log(dda), log(dd),
else write(‹-dd.dddd›, q×mod,log(T), log(rho),log(kappa), log(ddr),y[1]); if mode=4 t
end WATM;


procedure ATMOF(lTe, lL); value lTe, lL; real lTe, lL;
begin WTS;
```

```
lR:= A2×(lTes - lTe) + lL/A2;  Te:= exp(lTe/mod); R:= exp(lR/mod); L:= exp(lL/mod);
gacc:= G×M×Ms/(R×Rs)↑2;y[2]:= 1; C1:= U3×Ls×L/M;C4:= 4×pi/gacc×G;
X:= Xy; Y:= 1-X-Z;
eomua:= X/1.00801+Y/4.0028+Z/16.942;C2:= k×eomua/m0;
if kb on then begin
writecr; write(╞-dd.dddd╡, M, lTe, lL, lR, log(gacc), 5040/Te×2↑0.25); end;
nm:= ztype:= 1;  q:=(lps + (log(Te×0.5↑0.25) - ltb)/slT×slP)/mod +0.01; y[1]:= 0;

ad1P: DIFF(k1); if kb on then WATM;
if k1[1]<0.01 then begin q:= q+1; go to ad1P end;

step:= 0.1; MERSN;
if kb on then begin DIFF(k1); WATM end;

ztype:= to; nm:= tre;y[3]:= ln(Te×(0.5+0.75×y[1])↑0.25); y[1]:= A1; DIFF(k1); testg:=
if kb on then WATM;
step:= 0.05; MERSN;

if kb on then begin DIFF(k1); WATM end;

y[1]:= ln(y[1]×Rs×R); arb:= y[to];
y[to]:= antlog(lL)×Ls/L0;
y[fire]:= q;  F:= A1; q:= arb; ztype:= tre; Lneu:= y[fem]; y[fem]:=0; DIFF(k1);
if kb on then SKRIV;

testg:= testn; step:= -0.0002; WTS
end ATMOF;
procedure WTS;
begin if kb on then begin
   writecr; write(╞ddddd╡, tsg, tsd);
end end WTS;
procedure CKUGL(Tc,rhoc); value Tc,rhoc; real Tc,rhoc;
begin
X:= Xk;
XomH:= X/1.00801; YomHe:= (1-X-Z)/4.0028; ZomA:= Z/16.942; eomue:= XomH + A2 × YomHe

q:= fmas[nul]; y[1]:= A1; y[to]:= Atim4;
ztype:= tre; y[tre]:= Tc;
T:= exp(Tc); rho:= exp(rhoc);
arb:= F:= rho/(9.165₁₀-9×exp(A1p5×Tc))×eomue;
F:= (F-30)/30; F:= ((((0.109744×F-0.165376)×F+0.151420)×F-0.462698)×F+3.296236)×F+5.1
if kb on then begin write(╞-ddd.dddd╡,arb,F); writecr end;
y[fire]:= ln(k/m0×rho×T×(XomH+YomHe+ZomA+eomue×F)+a/A3×exp(Tc×A4)
  -sqrt(cnsqm×(X+7×Z+A3)↑3×rho/T)×rho);
DIFF(k1); y[fem]:= k1[fem];
if kb on ∨  mode=4  then SKRIV;
step:= q:= fmas[nul];
y[1]:= ln(A3×M×Ms×q/(A4×pi×rho))/A3;
ddr:= U3×Ms×kappa×eps×P/T4;
arb:= -G×M×Ms×q/A2×rho/(exp(y[1])×P); y[to]:= eps×M×Ms×q/L0;
y[tre]:= Tc+arb×(if ddr<dda then ddr else dda); y[fire]:= y[fire]+arb; DIFF(k1); y[fe
testg:= testn; if kb on ∨ mode=4  then SKRIV;
end CKUGL;



procedure SKRIV;
begin if mode=fire then go to udskriv; writecr; write(╞-dd.dddd╡,q,exp(y[1]-lR/mod)/R
mod×y[tre], mod×y[fire], mod×ln(rho), ddr,mod×ln(kappa), eps); go to ud;
udskriv:s:= select(UNWP); writecr;
write(╞-d.dddd╡, q, exp(y[1]-lR/mod)/Rs,y[to]×L0/(antlog(lL)×Ls),
mod×y[tre], 0.1×mod×y[fire], mod×ln(rho), ddr, mod×ln(kappa),
X, beta, eps4×0.936₁₀-4, eps×Atim4); dummy:=  select(s);
```

```
ud: end procedure SKRIV;


autm:= mode:= zonenum[-1]:=  nul; comment punch off; autm:=2; comment punch on;* mtx4
A0:= gsum:= 0; A1:= mf:= 1;
A2:= 2; A3:= 3; A4:= 4; A5:= 5; A6:= 6;
A01:= 0.1; A08:= 0.8; A09:= 0.9; A1p5:= 1.5;
pi:= 3.14159265; mod:= 0.434294482; recmo:= 2.30258509;
G:= 6.668₁₀-8; m0:= 1.66026₁₀-24; k:= 1.38046₁₀-16; a:= 7.5641₁₀-15; c:= 2.997929₁₀10;
Ls:= 3.90₁₀33; Rs:= 6.9598₁₀10; Ms:= 1.989₁₀33; lTes:= log(Ls/(a×pi×c×Rs↑2))/A4;
U1:= Ms/(A4×pi); U3:= A3/(16×pi×a×c×G×Ms); cnsqm:= pi/k/8×(4.8029↑3/m0/3₁₀30)↑2/m0; ex
C3:= A6×sqrt(2)×a×c×m0/k; go to ldef1;

rep: writecr;  writecr; write(⊰-ddddd⊱,num); write(⊰-ddd.ddd⊱, M, Xy, Z); writecr;
lR:= A2×(lTes-lTe) + lL/A2;
write(⊰-dd.dddd⊱,lR,lTe,lL,rltp[0,tre]×mod,rltp[0,fire]×mod);

if autm = 0 ∨ kb on then go to SW[lyn];
fsum:= 0; tsg:= tsd:= 0;
for lfp:= 0 step 1 until nfp-1 do begin

intil:= if lfp<fit then lfp+1 else nfp-1-lfp+fit;
if lfp = 0 then CKUGL(rltp[0,tre], rltp[0,fire]);
if lfp = fit then begin si:=0; ATMOF(lTe,lL) end;

DIFF(k1); testg:= testn; MERSN; if mode = 1 then zonenum[lfp]:= counts;
if kb on then SKRIV; if autm ≠ 1 ∧ lfp≠fit-1 then writecr;

if lfp = fit-1 then
     begin for i:= 1 step 1 until fire do res[fit,i]:= -y[i] end else
if lfp = nfp-1 then
     begin for i:= 1 step 1 until fire do begin res[nfp,i]:= y[i]; arb:= y[i]+res[fit
                    fsum:= fsum+abs(arb);
                    if autm ≠ 1 then begin s:= select(UNWW); write (⊰-dd.dddd⊱,arb);
                    end end end
else begin comment normal case follows; if kb on then begin
UD 2 (rltp,intil); writecr end;
for i:= 1 step 1 until fire do begin arb:= rltp[intil,i]; res[1+lfp,i]:= arb1:= arb-y
y[i]:= arb;
if autm ≠ 1 then begin s:= select(UNWW); write (⊰-dd.dddd⊱,arb1); dummy:=  select(s)e
fsum:= fsum + abs(arb1);
end end normal case; sres:= sres + fsum
end integration statement;
if mode=4 then goto frem1;


writecr; write(⊰-dd.dddd⊱,fsum); write(⊰-dddd⊱,tsg,tsd); if autm = 0 then begin i:= 1
if i = 7 then go to matrix else go to SW[i] end else begin
mtx4:= mtx4-1; ccore:= -to; if fsum<0.0005 then go to frem;
if fsum<0.0050 then begin ccore:= -tre; go to forbedring end;
if fsum>0.2×gsum∧mtx4<1 then go to matrix end;
gsum:= fsum;


forbedring: begin
procedure FORBEDRING;
begin integer i,k;  real rm1,rm2,rm3,rm4;
array A,B,C,A1,B1,C1[1:4], KO[1:4,1:5];
procedure ELIM (beg, slut); value beg, slut; integer beg, slut;
begin
for i:= 1 step 1 until fire do begin A[i]:= res[beg,i]; B[i]:= drdp[beg,i,tre];
    C[i]:= drdp[beg,i,fire]; end;
for i:= 1+beg step 1 until slut do begin
for k:= 1 step 1 until fire do begin
```

```
rm1:= drdp[i,k,1]; rm2:= drdp[i,k,to]; rm3:= drdp[i,k,tre]; rm4:= drdp[i,k,fire];
A1[k]:= -rm1×A[1]-rm2×A[to]-rm3×A[tre]-rm4×A[fire]+res[i,k];
B1[k]:= -rm1×B[1]-rm2×B[to]-rm3×B[tre]-rm4×B[fire];
C1[k]:= -rm1×C[1]-rm2×C[to]-rm3×C[tre]-rm4×C[fire] end;
for k:= 1 step 1 until fire do begin A[k]:= A1[k]; B[k]:= B1[k]; C[k]:= C1[k]; end;
end end ELIM;

ELIM(1,fit);
for k:= 1 step 1 until fire do begin
KO[k,fem]:= A[k]; KO[k,tre]:= B[k]; KO[k,fire]:= C[k];
end elimination from centre;
ELIM(1+fit,nfp);
for k:= 1 step 1 until fire do begin
KO[k,fem]:= KO[k,fem]+A[k]; KO[k,1]:= B[k]; KO[k,to]:= C[k];
end elimination from surface;

LLGAUSS(fire,A,KO,EXIT,10-6);
for i:= 1 step 1 until fire do korr[0,i]:= A[i];
for i:= 1 step 1 until fit-1 do begin
for k:= 1 step 1 until  fire do korr[i,k]:= res[i,k]-(if i = 1 then 0 else drdp[i,k,1
        +drdp[i,k,to]×korr[i-1,to])-drdp[i,k,tre]×korr[i-1,tre]-drdp[i,k,fire]×korr[i
end solution from centre;
for k:= 1 step 1 until fire do korr[nfp-1,k]:= res[fit+1,k]
        -drdp[fit+1,k,1]×korr[0,1]-drdp[fit+1,k,to]×korr[0,to];
for i:= nfp-to step -1 until fit +1 do begin
real procedure drgk;
begin s:= s+1;
drgk:= drdp[t,k,s]×korr[i+1, s]
end drgr;
t:= nfp-i+fit;
for k:= 1 step 1 until fire do begin s:= 0;  korr[i,k]:=
       res[t,k] - drgk - drgk - drgk - drgk
end end  solution from surface;
for i:= 1 step 1 until fire do korr[fit,i]:= 0;
go to END;
EXIT:writetext(‹<
No solution›);
END: end procedure FORBEDRING;

FORBEDRING;
if autm≠0 then  mf:= if fsum>A3 then 0.3 else if fsum>1 then 0.6 else 1;
for i:= nul step 1 until nfp-1 do begin if kb on then begin UD2(rltp,i); UD2(korr,i);
for j:= 1 step 1 until fire do rltp[i,j]:= rltp[i,j] - korr[i,j]×mf; end forbedring;
lTe:= rltp[nul,1]; lL:= rltp[nul,to];
end forbedringsblock;


if ccore = -tre then go to frem;
if autm = 0 then go to SW[lyn]; go to rep;


matrix: writecr; mtx4:= fire;
begin real GF, gstep; integer from , i;
procedure FORST(s,t); value s,t; integer s,t;
for i:= 1 step 1 until fire do drdp[lfp+1,i,s]:= drdp[lfp+1,i,s+to ]:=
     (rltp[intil,i]-y[i]-res[lfp+1,i])/alter[t];


for lfp:= 0 step 1 until nfp-1 do begin
intil:= if lfp<fit then lfp + 1 else nfp-1-lfp + fit;
from:= if lfp<fit then lfp else intil + 1;
if lfp = 0 then begin CKUGL(rltp[0,tre]+alter[tre],rltp[0,fire]); MERSN; FORST(1,tre)
        CKUGL(rltp[0,tre],rltp[0,fire]+alter[fire]); MERSN; FORST(to,fire); gstep:=
end else
```

```
if lfp = fit then begin ATMOF(lTe+alter[1],lL); MERSN; FORST(1,1);
         ATMOF(lTe,lL+alter[to]); MERSN; FORST(to,to); gstep:= step; GF:= F;
end else
begin comment normal case follows;
for i:= 1 step 1 until fire do begin for j:= 1 step 1 until fire do y[j]:= rltp[from,
q:= fmas[from]; y[i]:= y[i] + alter[i]; F:= GF; step:= gstep; DIFF(k1);
testg:= testn; MERSN;
for j:= 1 step 1 until fire do drdp[lfp+1,j,i]:= (if lfp=fit-1 then-y[j]-res[fit,j] e
if lfp=nfp-1 then y[j]-res[nfp,j] else rltp[intil,j]-y[j]-res[lfp+1, j])/alter[i];
end testalterations;
end one zone;
GF:= F; gstep:= step;
end lfp forstatement;
end matrixblock;
if autm = 0 then go to SW[lyn]; go to forbedring;

ldef: dummy:=  select(UNWW); writetext(ɬ<
Input definitionɭ); dummy:=  lyn; dummy:= select(UNRW); nfp:= read integer;
ldef1: fit:= read integer;
begin
procedure LPGG;
begin real Rt,Lt;
for i:= nul step 1 until nfp-1 do begin
arb:=readreal;
if arb ≠ fmas[i] then begin dummy:=  select(17); writetext (ɬ<
Definition error, read new definitionɭ); dummy:= lyn; go to ldef end;
rltpgg[i,1]:= read real; rltpgg[i,to]:= read real;
rltpgg[i,tre]:= read real/mod; rltpgg[i,fire]:= read real/mod;
if i = nul then begin
Rt:= antlog(rltpgg[nul,to]/A2+A2×(lTes-rltpgg[nul,1]))×Rs;
Lt:= antlog(rltpgg[nul,to])×Ls;
if L0 = 0 then L0:= Lt/LV end
else begin comment i ≠ 0, normal case follows;
rltpgg[i,1]:= ln(rltpgg[i,1]×Rt);
rltpgg[i,to]:= rltpgg[i,to]×Lt/L0;
end end
end procedure LPGG;

procedure TPT(A); array A;
for i:= nul step 1 until nfp-1 do
for j:= 1 step 1 until fire do A[i,j]:= rltpgg[i,j];


num:= read integer; age:= read real; delt:= read real; deltg:= read real;
M:= read real; qk:= read real; Xk:= read real; qy:= read real; Xy:= read real;
Z:= read real; LV:= read real; delta:= read real; dpar:= read real;
for i:= 1 step 1 until fire do begin rpfac[i]:= read real; alter[i]:= read real end;
nQ:= read integer; tgr:= read integer; energi:= read integer;
for i:= nQ step -1 until nul do begin Q[i]:= read real; dummy:=  read real;
XQ[i]:= read real/mod end;
for i:= nul step 1 until tgr do begin arb:= read real; arb:= read real end;
for i:= nul step 1 until nfp do fmas[i]:= read real;
for i:= 1 step 1 until nfp do
for j:= 1 step 1 until fire do for s:= 1 step 1 until fire do drdp[i,j,s]:= read real
L0:= 0; LPGG; TPT(rltp); LPGG; TPT(rltpg); LPGG; lTe:= rltp[nul,1]; lL:= rltp[nul,to]
end inputblock; dummy:=  select(UNWW);
go to rep;



tdef: s:=select(UNWD);
begin
comment remember punch off / on around print in call of lineprint;
comment punch off  ;
```

```
procedure lineprint(n,a,b,c,d,e); value n,a,b,c,d,e;
integer n; real a,b,c,d,e;
begin switch SW1:= l1,l2,l3,l4,l5;
go to SW1[n];
l1: write(ǂ-dd.ddddd╞,a,writetext(ǂ<,╞));
goto END;
l2: write(ǂ-dd.ddddd╞,a,writetext(ǂ<,╞),b,writetext(ǂ<,╞));
goto END;
l3: write(ǂ-dd.ddddd╞,a,writetext(ǂ<,╞),b,writetext(ǂ<,╞),c,writetext(ǂ<,╞));
goto END;
l4: write(ǂ-dd.ddddd╞,a,writetext(ǂ<,╞),b,writetext(ǂ<,╞),c,writetext(ǂ<,╞),
    d,writetext(ǂ<,╞));
goto END;
l5: write(ǂ-dd.ddddd╞,a,writetext(ǂ<,╞),b,writetext(ǂ<,╞),c,writetext(ǂ<,╞),
    d,writetext(ǂ<,╞),e,writetext(ǂ<,╞));
END: end procedure lineprint;
comment punch on * ;

procedure line (n,a,b,c,d,e); value n,a,b,c,d,e;
integer n; real a,b,c,d,e;
begin integer i; real r;
i:=0; writecr;
for r:=a,b,c,d,e do begin i:=i+1; if i≤n then begin write(ǂ-dd.ddddd╞,r);
writetext(ǂ<,╞); end;end;end procedure line;

procedure WPAR(A); array A;
begin real Rt,Lt;
writetext(ǂ<
╞); Rt:= antlog(A[nul,to]/A2+A2×(lTes-A[nul,1]))×Rs; Lt:= antlog(A[nul,to])×Ls;
line  print * (5,fmas[0],A[0,1],A[0,2],A[0,3]×mod,A[0,4]×mod);
for i:= 1 step 1 until nfp-1 do
line  print * (5,fmas[i],exp(A[i,1])/Rt,A[i,2]×L0/Lt,A[i,3]×mod,A[i,4]×mod);
end procedure WPAR;

line  print*  (2,nfp,fit,dummy,dummy,dummy);
line  print*  (4,num,age,delt,deltg,dummy);
line  print*  (3,M,qk,Xk,dummy,dummy);
line  print*  (3,qy,Xy,Z,dummy,dummy);
line  print*  (3,LV,delta,dpar,dummy,dummy);
fori:=1 step 1 until fire do
line  print*  (2,rpfac[i],alter[i],dummy,dummy,dummy);
line  print*  (3,nQ,tgr,energi,dummy,dummy);
writecr;
for i:=nQ step -1 until 0 do
line  print*  (3,Q[i],exp(XQ[i]),mod×XQ[i],dummy,dummy);
writecr;
for i:=nul step 1 until tgr do
line  print*  (2,arb,arb,dummy,dummy,dummy);
for i:=nul step 1 until nfp do
line  print*  (1,fmas[i],dummy,dummy,dummy,dummy);
for i:=1 step 1 until nfp do begin writetext(ǂ<
╞);
for j:=1 step 1 until fire do
line  print*  (4,drdp[i,j,1],drdp[i,j,2],drdp[i,j,3],drdp[i,j,4],dummy);end;
WPAR(rltp);   WPAR(rltpg);  WPAR(rltpgg);
writetext(ǂ<




                    ╞);
end af tdef;
```

```
go to rep;
choose:t:= select(UNWW); writetext(⊰<
choose by letter⊱); i:= lyn; if i = 53 then energi:= read integer;
if i= 54 then begin fmas[0]:= read real; fmas[nfp]:= read real end;
if i = 39 then dpar:= read real;
if i = 18 then go to stop; if i = 50 then begin
for j:= 1 step 1 until fire do alter[j]:= read real end;
if i = 36 then mf:= read real;
if i = 49 then autm:= read integer;
if i = 52 then delta:= read real;
dummy:=
select(t);
go to SW[lyn]; go to rep;

frem: dummy:=  select(49); writetext(⊰<
   Te,  Mbol  =  ⊱); write(⊰ddddddd⊱, antlog(lTe));
write(⊰-ddddd.ddd⊱, 4.72 - 2.5×lL);  mode:= fire; t4:= 1; go to tdef;
frem1: arb1:= antlog(lL)×Ls; dummy:=  select(49); writecr;
writetext(⊰<   Lneu/L =  ⊱); write(⊰dd.dddd⊱, (Lneu-y[fem])×L0/arb1); dummy:=  select
 M:=M+dpar;
for i:= nul step 1 until nfp-1 do
    for  j:= 1step1 until fire do begin
     arb:= 2×rltp[i,j] - rltpg[i,j];
     rltpg[i,j]:= rltp[i,j]×(if i>0∧j=2 then L0/arb1 else 1 ); rltp[i,j]:= arb×(if i>
mode:= nul; gsum:= A4; L0:=arb1;
lTe:= rltp[nul, 1]; lL:= rltp[nul, to];
dummy:=
select(UNWW); go to rep;

stop: end end
t<



procedure fpline(A);...

begin. for j:=0,1,2,3,4 do read real;..

(if. k-5<ilp then k-5. n k.

real;.
for j:=k-5+1 step 1 until ilp do A[i,j]:=A[i,j-1]×1.5-A[i,j-2]×0.5;..

until k.-5..

lps:= p.+5×slP;.;.

p-lps.+5×slP..



l/Hp:=  ⊱...

nfp:= read. real.integer.

ldef...

nfp:= read. real.integer.

fit:= read. real.integer.

num:= read. real.integer.

end af tdef;.
```

```
dummy:=select(s);
go to rep;

.rep;.
```