```
c ########################################################################
c
c                           Dierk Raabe
c                             Maerz 20119
c
c               Probabilistischer Zellulaerer Automat
c            Simulation der Erholung und Rekristallisation
c
c                           2 D   VERSION
c
c           Zyklische oder Nicht-Zyklische Randbedingungen
c
c                      Eulerwinkel kontinuierlich
c
c ########################################################################
c
c                         Modellannahmen
c
c   Berueecksichtigung der Misorientierungen (Eulerraum - 3D)
c
c   Implementierte Keimbildungsmodelle
c   a) Orientierte stetige Keimbildung
c   b) Regellose stetige Keimbildung
c   c) Regellose ortsgesaettigte Keimbildung bei t=0s
c   d) Einheitlich orientierte Keimbildungsfront bei t=0s
c
c   Ratengleichung MESO  - echtzeit - zaehler
c   Nicht-Metropolis Monte Carlo Integration
c
c   Dimension:  "D
c
c   Korngrenzenmobilitaet:
c   Isotropes und Orientierungs/Achsen abhaengiges Keimwachstum
c   Bei unterschiedlichen Aktivierungsenergien wird das delta_t fuer
c   die langsamste Korngrenze berechnet werden, alle schnelleren
c   Korngrenzen mit kleineren Aktivierungsenergien erhalten dann
c   einen Verlaengerungsfaktor bei der wahrscheinlichkeitsberechnung
c   der sich aus dem Verhaeltnis der delta_t berechnet
c
c ########################################################################

      program rx_konti1

      use MSFLIB
      use PORTLIB

c ########################################################################
c                             PARAMETER - Feld
c ########################################################################
      implicit integer (f)

      parameter (nachbar= 18)
      parameter (i_raus=20)
      parameter (i_spez=2)
      parameter (i_null=0)
      parameter (x_null=0.)
      parameter (feldrandx = 805)   !473 X 473 X 65   !805
      parameter (feldrandy = 280)   !473 X 473 X 65   !280
      parameter (feld_3d = 5)
      parameter (laufzeit=9999999999)
c ----------------------------------------------------------------------
c i_raus       Anzahl der rauszuschreibenden Gefuege
c nachbar      Anzahl der Nachbarn, die bei der Transformation berueecksichtigt wird
c feld         Feldgroesse-1
c feldrand     genaue Abmessung des Feldes bis einschlieslich Rand
c parameter    feld = rand min
c parameter    feldrand = rand
c ########################################################################

c ########################################################################
c                             PARAMETER - Units fuer Eingabe und Ausgabe
c ########################################################################
      implicit integer (u)

      parameter (uni_boun=966)
      parameter (uni_ein1=969)
      parameter (uni_ein2=970)
      parameter (uni_au0=967)
      parameter (uni_keim_ori=994)
      parameter (uni_anf_ori=996)
      parameter (uni_anf_rho=998)
      parameter (uni_end_ori=986)
```

```
      parameter (uni end rho=988)
      parameter (uni kw3=987)
      parameter (uni kin1=990)
      parameter (uni kin2=991)
      parameter (uni ch=992)
c      parameter (uni hv=993)
c -------------------------------------------------------------------
c uni ein1     969 : unit fuer allgemeine, physikalische Eingabedaten
c              Name : rx konti phys.txt
c uni ein2     970 : unit fuer alle ortsdiskreten Daten (Mapping der verformten
c                    Ausgangsprobe)
c              Name : rx konti map.txt
c uni au0      967    : physikalische Ausgabedaten allgemeiner Art
c              Name : rx konti para.txt
c uni keim     938    : Ausgabedaten mit Keimbildungsgefuege direkt nach Keimbildung
c              Name : rx konti keim.txt
c uni au1      971   : Ausgabedaten mit Gefuege
c              Namen : rx konti gef1.txt
c uni au2      972   : Ausgabedaten mit Gefuege
c              Namen : rx konti gef2.txt
c uni au3      973   : Ausgabedaten mit Gefuege
c              Namen : rx konti gef3.txt
c uni au4      974   : Ausgabedaten mit Gefuege
c              Namen : rx konti gef4.txt
c uni au5      975   : Ausgabedaten mit Gefuege
c              Namen : rx konti gef5.txt
c uni end ori      986   : letzte Ausgabedaten mit Gefuege (ori)
c uni end rho      986   : letzte Ausgabedaten mit Gefuege (rho)
c              Namen : rx konti end.txt
c uni kin1     990 : Ausgabedaten der Kinetik (Zeit, umgewandelter Volumenbruchteil)
c              Name : rx konti kin1.txt
c uni kin2     992 : Ausgabedaten der Kinetik (Zeit, umgewandelter Volumenbruchteil)
c              Name : rx konti kin2.txt
c uni korn     991 : Ausgabedaten der Korngroessenverteilung (auch Aspect ratio etc.)
c              Name : rx konti korn.txt
c uni ch       992  : Ausgabedaten fuer Cahn-Hagel Analyse S(X)
c              Name : rx konti ch.txt
c uni hv       993  : Ausgabedaten fuer Haertekurve hv(t)
c              Name : rx konti hv.txt
c ######################################################################


c ######################################################################
c                            CHARACTER
c ######################################################################
c ######################################################################
      CHARACTER(9) today
c ######################################################################

c ######################################################################
c                            INTEGER - Skalar
c ######################################################################
c ######################################################################
      integer i randx
      integer i randy
      integer ra3d aus
      integer ra3d min
      integer ra3d tat
      integer rand tatx
      integer rand minx
      integer rand ausx
      integer rand taty
      integer rand miny
      integer rand ausy
      integer bildung
      integer keimzahl
      integer iwurf
      integer ix,iy,iz
      integer jx,jy,jz
      integer i
      integer ii
      integer rho rx
      integer rho situ
      integer*4 iseed
      integer zeitmax
      integer z shiftx
      integer z shifty
      integer*2 ihr, imin, isec, i100th
      integer zyklisch
c -------------------------------------------------------------------
c zyklisch     zyklische Randbedingungen - ja oder nein
c z_shift      z-Verschiebung beim Zeichnen des Gefueges
```

```
c iwurf        Integer Wurf
c rho max      Die maximal auftretende Versetzungsdichte
c keimzahl     Anzahl der Keime bei ortsgesaettigter Keimbildungs
c ori rx       Orientierung einer Zelle nach der Rekristallisation
c ori um       Umgebungsorientierung
c rho rx       Versetzungsdichte innerhalb einer Zelle nach der Rekristallisation
c rho situ     Versetzungsdichte nach der in-situ Rekristallisation
c x 0,y 0      Offset bei der Graphik
c zeitmax      maximale Laufzeit in willkuerlichen Einheiten
c ######################################################################


c ######################################################################
c                            INTEGER - Feld
c ######################################################################
      integer*2  rho alt(feldrandx,feldrandy,feld 3d)
      integer*2  rho neu(feldrandx,feldrandy,feld_3d)
      integer*2  zaa(i raus)
      integer    iachs(3)
      integer    a s(i spez,3)
      integer    uni au(i raus)
      integer    uni rho(i raus)
      integer    vek(i spez)
c ---------------------------------------------------------------------
c uni au()     units fuer raus-gefuege   ORIENTIERUNGEN
c uni rho()     units fuer raus-gefuege   RHO
c a s1         Drehachse der 1. speziellen Korngrenze
c a s2         Drehachse der 2. speziellen Korngrenze
c iachs        Normierte integer-gerundete Drehachse zum Min-Winkel zwischen 2 Orientierungen
c rho alt      Versetzungsdichte nach vorangegangenem Durchlauf
c ori alt      Orientierung nach vorangegangenem Durchlauf
c rho neu      Versetzungsdichte nach Transformation
c ori neu      Orientierung nach Transformation
c ######################################################################

c ######################################################################
c                            REAL - Skalar
c ######################################################################
      real keimvol
      real keimstreu
      real fi1keim,fikeim,fi2keim
      real dukbt
      real kb
      real ev
      real aengst
      real mikro
      real giga
      real erg max
      real rho offset
      real ge max
      real v max
      real wurf
      real temp
      real temp sch
      real schubmod
      real keim erg
      real wach erg
      real vol rx
      real fak 2d
      real fak 3d
      real log fak
      real faktor
      real burgers
      real sb,sbb,bb
      real mob0
      real lambda
      real hoch13
      real rho max
      real p max
      real deltarho
      real querkon
      real pi
      real sekunden
      real frac alt
      real frac rel
      real frac abb
      real frac ab
      real frac_rx
      real zeit
      real delta t
      real vor wahr
      real wahrsch
```

```
      real wink_ab
      real winkel
      real keim_mis
      real gam_mis
      real vol_inv
      real d0_self
      real q_self
      real w_spez
      real therm_g
      real therm_b
      real freq0
      real q_min
      real mobq_min
      real sigma
      real abw
      real rad
c ----------------------------------------------------------------------
c fi1keim,fikeim,fi2keim   Keimorientierung
c abw          unschaerfe bezgl. RX vol.anteil beim rausschreiben der zwischengefuege
c q_min und mobq_min  die minimal auftretende aktivenergie nebst mobil.
c                     zur berechnung der geber-frequenz
c freq0        anschwingfrequenz
c therm_g      Temperaturkoeffizient des G-Moduls in GPa/K
c therm_b      Temperaturkoeffizient des b-Vektors in 1/K
c w_spez       Wahrscheinlichkeitsfaktor fuer spezielle Korngrenzen
c d0_self      Vorfaktor der Selbstdiffusion
c q_self       Aktivierungsenergie der Selbstdiffusion
c vol_inv      inverses normiertes halbiertes Gesamtvolumen *lambda
c keim_mis     kleinste Misorientierung, bei der Keimbildung auftritt - zwischen 10 und 20 Gra
d
c gam_mis      Read-Shockley Energie zum Winkel keim_mis
c gam_lok      lokale Read-Shockley Energie aus der lokalen tatsaechl. Misorient.
c wink_ab      unterer Abschneidewinkel fuer die Mobilitaet
c              - bei kleineren Werten ist die Mob. Null
c winkel       Misorientierungswinkel
c umgebung     Anzahl der Nachbarzellen mit anderer Orientierung,
c              ist relevant fuer Kruemmung und Korngrenzenbegradigung
c querkon      Querkontraktionszahl v
c ge_max       Maximal auftretende Geschwindigkeit eines aus einer der
c              Nachbarzellen in die betrachtete Zelle hineinwachsenden Keimes, d.h.
c              diejenige Nachbarzelle, die bei Umgebungsabfrage die maximale
c              Geschwindigkeit aufweist, wird in die momentan beobachtete Zelle hineinwachsen
c wurf         gewuerfelte, also regellose Zahl
c volumen      wahres volumen in der rechnung: n(x)*n(y)*n(z)*zellgroe**3
c temp         Temperatur in K
c inv_temp     inverse Temperatur in 1/K
c temp_sch     Schmelztemperatur in K
c schubmod     Schubmodul in Giga Pascal
c keim_erg     einheitliche Aktivierungsenergie der Keimbildung in eV
c wach_erg     einheitliche Aktivierungsenergie der Korngrenzenmoblitaet in eV
c vers_erg     gespeicherte Versetzungsenergie in eV
c wahrsch      Wahrscheinlichkeitsfaktor
c vol_rx       rekristallisiertes Volumen
c fak_2d       2D Geometriefaktor zur Beruecksichtigung der flaechendiagonalen
c              Transformation
c fak_3d       3D Geometriefaktor zur Beruecksichtigung der raumdiagonalen
c              Transformation
c burgers      Burgersvektor in Aengstroem
c a0
c log_a0          r3eduzierte exper. mobilitaet
c deltarho     Aenderung der Versetzungsdichte ueber die Korngrenze
c frac_rx      rekristallisierter Volumenbruchteil
c ######################################################################


c ######################################################################
c                           REAL - Feld
c ######################################################################
      real geschw(nachbar)
      real fi1(2),fi(2),fi2(2)
      real fi1_alt(feldrandx,feldrandy,feld_3d)
      real fi_alt(feldrandx,feldrandy,feld_3d)
      real fi2_alt(feldrandx,feldrandy,feld_3d)
      real fi1_neu(feldrandx,feldrandy,feld_3d)
      real fi_neu(feldrandx,feldrandy,feld_3d)
      real fi2_neu(feldrandx,feldrandy,feld_3d)
      real dr1(3,3),dr3(3,3)
      real xpp(3,3),xppr(3,3)
      real alpu_s(i_spez),alpo_s(i_spez)
      real wach_s(i_spez)
      real mob0s(i_spez)
      real mob_exp(i_spez)
```

```
      real gef(i raus)
c ---------------------------------------------------------------------
c rod_u(i spez)  Rodriguesvektorbetraege spez. KG unten
c rod_o(i spez)  Rodriguesvektorbetraege spez. KG oben
c gef()       Gefuege RX bruchteil als kriterium zum rausschreiben
c wach_s(),log_a0s(),korn_s(),a0_s(),mob0s()
c dr           Drehmatrizen fuer Euler-Operationen insbesondere bei orient. KB
c xpp          dito
c geschw       Geschwindigkeit des aus den Nachbarzellen in die betrachtete
c              Zelle hineinwachsenden Keime
c fi1,fi,fi2   Eulerwinkel
c #####################################################################

c #####################################################################
c                            COMMON - Feld
c #####################################################################
      common sym(3,3,24)
c ---------------------------------------------------------------------
c sym()        Symmetriematrizen zur Berechnung der Misorientierung
c #####################################################################

c #####################################################################
c                            DATA
c #####################################################################
      data eulermax   /90./
      data kb         /1.38E-23/
      data ev         /1.60218E-19/
      data aengst     /1.0E-10/
      data giga       /1.0E+9/
      data centi      /1.0E-2/
      data mikro      /1.0E-6/
      data hoch13     /1.0E+13/
      data pi         /3.14159265/
      data frac_rx    /x null/
c #####################################################################

c #####################################################################
c                            FORMATE
c #####################################################################
100   format(a)
101   format(f12.3)
102   format(a2,i12)
104   format(a2,i12,a)
105   format(5i12)
106   format(i12)
108   format(f18.7,a)
109   format(i12,a)
111   format(a,i7)
115   format(2f12.3)
118   format(3i6,3f8.2,a)
119   format(f17.7,f12.3)
128   format(f20.11,a)
139   format(3f12.3)
155   format(a2,f12.3)
157   format(f12.5)
179   format(E15.6,a)
198   format(e16.6,a)
207   format(f12.6)
298   format(3i5)
919   format(f11.3,2f10.3,f12.5,f6.2)
917   format(3i7,a,f7.2,a,f16.2,a)
1119  format(2f17.7)
1129  format(i4,3f8.3)
1509  format(a,i10)
c #####################################################################

c Laufzeit und Datum abfragen - START
      call GETTIM(ihr, imin, isec, i100th)
      sek_start=float(ihr)*3600.+float(imin)*60.+float(isec)
      call date(today)


C ERSTELLEN DER SYMMETRIEMATRIZEN
C ***********************************************************
       DO 219 I=1,3
       DO 219 J=1,3
       DO 219 K=1,24
219      SYM(I,J,K)=0.
C
       SYM(1,1,1)=1.
       SYM(2,2,1)=1.
       SYM(3,3,1)=1.
```

```
C
      SYM(1,1,2)=1.
      SYM(2,3,2)=-1.
      SYM(3,2,2)=1.
C
      SYM(1,1,3)=1.
      SYM(2,2,3)=-1.
      SYM(3,3,3)=-1.
C
      SYM(1,1,4)=1.
      SYM(2,3,4)=1.
      SYM(3,2,4)=-1.
C
      SYM(1,3,5)=-1.
      SYM(2,2,5)=1.
      SYM(3,1,5)=1.
C
      SYM(1,3,6)=1.
      SYM(2,2,6)=1.
      SYM(3,1,6)=-1.
C
      SYM(1,1,7)=-1.
      SYM(2,2,7)=1.
      SYM(3,3,7)=-1.
C
      SYM(1,1,8)=-1.
      SYM(2,2,8)=-1.
      SYM(3,3,8)=1.
C
      SYM(1,2,9)=1.
      SYM(2,1,9)=-1.
      SYM(3,3,9)=1.
C
      SYM(1,2,10)=-1.
      SYM(2,1,10)=1.
      SYM(3,3,10)=1.
C
      SYM(1,2,11)=-1.
      SYM(2,3,11)=1.
      SYM(3,1,11)=-1.
C
      SYM(1,3,12)=1.
      SYM(2,1,12)=-1.
      SYM(3,2,12)=-1.
C
      SYM(1,2,13)=-1.
      SYM(2,3,13)=-1.
      SYM(3,1,13)=1.
C
      SYM(1,3,14)=-1.
      SYM(2,1,14)=1.
      SYM(3,2,14)=-1.
C
      SYM(1,2,15)=1.
      SYM(2,3,15)=-1.
      SYM(3,1,15)=-1.
C
      SYM(1,3,16)=-1.
      SYM(2,1,16)=-1.
      SYM(3,2,16)=1.
C
      SYM(1,2,17)=1.
      SYM(2,3,17)=1.
      SYM(3,1,17)=1.
C
      SYM(1,3,18)=1.
      SYM(2,1,18)=1.
      SYM(3,2,18)=1.
C
      SYM(1,2,19)=1.
      SYM(2,1,19)=1.
      SYM(3,3,19)=-1.
C
      SYM(1,1,20)=-1.
      SYM(2,3,20)=1.
      SYM(3,2,20)=1.
C
      SYM(1,3,21)=1.
      SYM(2,2,21)=-1.
      SYM(3,1,21)=1.
C
      SYM(1,1,22)=-1.
```

```
       SYM(2,3,22)=-1.
       SYM(3,2,22)=-1.
C
       SYM(1,3,23)=-1.
       SYM(2,2,23)=-1.
       SYM(3,1,23)=-1.
C
       SYM(1,2,24)=-1.
       SYM(2,1,24)=-1.
       SYM(3,3,24)=-1.


      rad=pi/180.

c Units fuer die Gefuegefiles (Orientierungen)
      do 196 i=1,i_raus
      uni_au(i)=7770+i
196   continue


c Units fuer die Gefuegefiles (Rho)
      do 197 i=1,i_raus
      uni_rho(i)=7870+i
197   continue


c Initialisierung und Warmlaufen des Random Generators *****************
      sekunden=secnds(0.0)
      iseed=ifix(sekunden)
      do 272 i=1,20
272   wurf = ran(iseed)

c Vorbelegung fuer Gefuegefiles (nur ein Gefuege mit t=const. pro file)
      do 3272 i=1,i_raus
3272  zaa(i)=0

c Vorbelegung fuer Gesamtsimulationszeit
      zeit=0.0


c Oeffnen der beiden Ausgangsdateien ***********************************
c rx_konti_phys.txt : Physikalische Startdaten
c rx_konti_map.txt  : Ausgangsgefuege
      open(unit=uni_ein2,
     $file='c:\for\rx_konti2d\rx_konti_map2d.txt')
      open(unit=uni_ein1,status='unknown',
     $file='c:\for\rx_konti2d\rx_konti_phys2d.txt')
      open(unit=uni_boun,status='unknown',
     $file='c:\for\rx_konti2d\rx_konti_boun2d.txt')
      open(unit=uni_au0,status='unknown',
     $file='c:\for\rx_konti2d\rx_konti_para2d.txt')

c Textinfo bei Programmaufruf ******************************************
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '########################################'
      write(uni_au0,fmt=100) '    ORTS- UND ZEITDISKRETE SIMULATION'
      write(uni_au0,fmt=100) '         DER ERHOLUNG, KEIMBILDUNG'
      write(uni_au0,fmt=100) '           UND REKRISTALLISATION'
      write(uni_au0,fmt=100) '                MITTELS EINES'
      write(uni_au0,fmt=100) '  PROBABILISTISCHEN ZELLULAEREN AUTOMATEN'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '       Eulerraum kontinuierlich'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '         ++  2D  VARIANTE   ++ '
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '      Zyklische Randbedingungen'
      write(uni_au0,fmt=100) '                 oder'
      write(uni_au0,fmt=100) '    Nicht-Zyklische Randbedingungen'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '---------------------------------------'
      write(uni_au0,fmt=100) '         Dierk Raabe, Maerz 99'
      write(uni_au0,fmt=100) '########################################'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) ' Startfile: rx_konti_phys2d.txt'
      write(uni_au0,fmt=100) ' Startfile: rx_konti_map2d.txt'
      write(uni_au0,fmt=100) '                          (Win95/98/NT)'

      write(*,fmt=100)
      write(*,fmt=100) '########################################'
      write(*,fmt=100) '    ORTS- UND ZEITDISKRETE SIMULATION'
      write(*,fmt=100) '         DER ERHOLUNG, KEIMBILDUNG'
```

```
      write(*,fmt=100) '           UND REKRISTALLISATION'
      write(*,fmt=100) '                MITTELS EINES'
      write(*,fmt=100) '  PROBABILISTISCHEN ZELLULAEREN AUTOMATEN'
      write(*,fmt=100)
      write(*,fmt=100) '       Eulerraum kontinuierlich'
      write(*,fmt=100)
      write(*,fmt=100) '         ++  2D  VARIANTE   ++ '
      write(*,fmt=100)
      write(*,fmt=100) '       Zyklische Randbedingungen'
      write(*,fmt=100) '                 oder'
      write(*,fmt=100) '    Nicht-Zyklische Randbedingungen'
      write(*,fmt=100)
      write(*,fmt=100) '---------------------------------------'
      write(*,fmt=100) '         Dierk Raabe, Maerz 99'
      write(*,fmt=100) '#######################################'
      write(*,fmt=100)
      write(*,fmt=100) ' Phys. Daten: rx_konti_phys2d.txt'
      write(*,fmt=100) ' Gefuegedaten: rx_konti_map2d.txt'
      write(*,fmt=100) '                          (Win95/98/NT)'


c Einlesen der physikalischen Daten von rx_konti_phys **********************
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
c wahre Versuchstemperatur [K]
      read(uni_ein1,fmt=101) temp
c Schmelztemperatur [K]
      read(uni_ein1,fmt=101) temp_sch
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
c Burgersvektor, Schubmodul, Querkontraktion, theoretische Obergrenze der Versetzungsdichte im
 betroff. Material
      read(uni_ein1,fmt=101) burgers
      read(uni_ein1,fmt=101) therm_b
      read(uni_ein1,fmt=101) schubmod
      read(uni_ein1,fmt=157) therm_g
      read(uni_ein1,fmt=101) querkon
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=101) lambda
      read(uni_ein1,fmt=101) sigma
      read(uni_ein1,fmt=207) fak_2d
      read(uni_ein1,fmt=207) fak_3d
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=157) frac_abb
      read(uni_ein1,fmt=157) frac_ab
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=106) zyklisch
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=101) abw
      read(uni_ein1,fmt=101) gef(1)
      read(uni_ein1,fmt=101) gef(2)
      read(uni_ein1,fmt=101) gef(3)
      read(uni_ein1,fmt=101) gef(4)
      read(uni_ein1,fmt=101) gef(5)
      read(uni_ein1,fmt=101) gef(6)
      read(uni_ein1,fmt=101) gef(7)
      read(uni_ein1,fmt=101) gef(8)
      read(uni_ein1,fmt=101) gef(9)
      read(uni_ein1,fmt=101) gef(10)
      read(uni_ein1,fmt=101) gef(11)
      read(uni_ein1,fmt=101) gef(12)
      read(uni_ein1,fmt=101) gef(13)
      read(uni_ein1,fmt=101) gef(14)
      read(uni_ein1,fmt=101) gef(15)
```

```
      read(uni_ein1,fmt=101) gef(16)
      read(uni_ein1,fmt=101) gef(17)
      read(uni_ein1,fmt=101) gef(18)
      read(uni_ein1,fmt=101) gef(19)
      read(uni_ein1,fmt=101) gef(20)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=106) i_ch
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=106) bildung
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=101) keim_mis
      read(uni_ein1,fmt=101) keim_erg
      read(uni_ein1,fmt=101) fi1keim
      read(uni_ein1,fmt=101) fikeim
      read(uni_ein1,fmt=101) fi2keim
      read(uni_ein1,fmt=101) keimstreu
      read(uni_ein1,fmt=101) erg_max
      read(uni_ein1,fmt=101) rho_offset
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=100)
      read(uni_ein1,fmt=106) i_erhol
      read(uni_ein1,fmt=101) d0_self
      read(uni_ein1,fmt=101) q_self
      read(uni_ein1,fmt=106) i_hv


c Einlesen der Korngrenzendaten von rx_konti_boun ***********************
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
c Daten allgeneine Korngrenze
      read(uni_boun,fmt=101) wink_ab
      read(uni_boun,fmt=101) wach_erg
      read(uni_boun,fmt=101) mob0
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
c Daten erste spezielle Korngrenze
      read(uni_boun,fmt=101) wach_s(1)
      read(uni_boun,fmt=101) mob0s(1)
      read(uni_boun,fmt=101) alpu_s(1)
      read(uni_boun,fmt=101) alpo_s(1)
      read(uni_boun,fmt=106) a_s(1,1)
      read(uni_boun,fmt=106) a_s(1,2)
      read(uni_boun,fmt=106) a_s(1,3)
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
      read(uni_boun,fmt=100)
c Daten zweite spezielle Korngrenze
      read(uni_boun,fmt=101) wach_s(2)
      read(uni_boun,fmt=101) mob0s(2)
      read(uni_boun,fmt=101) alpu_s(2)
      read(uni_boun,fmt=101) alpo_s(2)
      read(uni_boun,fmt=106) a_s(2,1)
      read(uni_boun,fmt=106) a_s(2,2)
      read(uni_boun,fmt=106) a_s(2,3)

C Eingabefiles schliessen
      close(uni_ein1)
      close(uni_boun)


c ######################################################################
c ################   Kontrolle der Eingangsdaten    ###################
c ######################################################################
c Kontrollieren der physikalischen Daten  ******************
      if(temp.ge.temp_sch) then
      write(*,*) '==>  Versuchstemperatur zu hoch'
```

```
      goto 999
      end if

      if(sigma.gt.100.) then
      write(*,*) 'STATISTISCHER FEHLER'
      write(*,*) sigma, 100.
      write(uni_au0,fmt=100) 'STATISTISCHER FEHLER'
      pause
      end if

      if(erg_max.lt.rho_offset) then
      write(*,*) 'ALLE ZELLEN ERFUELLEN DAS THERMODYNAMISCHE
     $ KEIMBILDUNGSKRITERIUM  (ERG_MAX = RHO_OFFSET)'
      write(uni_au0,fmt=100)'ALLE ZELLEN ERFUELLEN DAS
     $THERMODYNAMISCHE KEIMBILDUNGSKRITERIUM  (ERG_MAX = RHO_OFFSET)'
      end if
c-----------------------------------------------------------------------
c-----------------------------------------------------------------------



c Nicht-Normalisierte Zahl in Anlehnung an
c Rodriguesvektorbetraege zu den speziellen Korngrenzen
c ausrechnen zur Bewaeltigung der Vorabfrage in der Hauptschleife
c echter Vektor:(volle Berechnung sparen) R = (r(ij)/Betr(r(ij)))*tan(w/2)
      do 292 i=1,i_spez
      vek(i)=a_s(i,1)*a_s(i,1)+a_s(i,2)*a_s(i,2)+a_s(i,3)*a_s(i,3)
c      write(*,*) vek(i)
292   continue


c ########################################################################
c ######   Umrechnen der physikalischen Daten in SI Einheiten   ########
c ########################################################################
c Umrechnen der physikalischen Daten in SI Einheiten ********************
c Netzmaschengroesse  [m]
      lambda=lambda*mikro
c Burgers Vektor [m]
      b_ohne=burgers
      burgers=burgers*aengst
c thermischer Ausdehnungskoeffizient des Burgers Vektors
      therm_b=therm_b*mikro
c Temperatur-Korrektur des des Burgers Vektors
      burgers=burgers*(1.+therm_b*(temp-300.))
c Temperatur-Korrektur des Schubmoduls noch in Gigapascal
      schubmod=schubmod-therm_g*(temp-300.)
c Schubmodul von Giga- in MegaPascal umrechnen [N/(m*m)]
      schubmod=schubmod*giga
c Aktivierungsenergie fuer die Keimbildung [J]
      keim_erg=keim_erg*ev
c Korngrenzenmobilitaeten [m*m*m/(N*s)]
      mob0=mob0*mikro
      do 1196 imo=1,i_spez
      mob0s(imo)=mob0s(imo)*mikro
1196  continue

c vorgegebene mittlere statistische Varianz fuer die weitere Rechnung
      sigma=sigma/100.

c Zusammengesetzte Groessen berechnen **********************************
c thermische Energie
      dukbt=1./(kb*temp)
c verschiedene hilfreiche Produkte
      sb=schubmod*burgers
      sbb=schubmod*burgers*burgers
      bb=burgers*burgers
c Vorfaktor fuer Korngrenzenenergie nach Read-Shockley in SI Einheiten
      gam_vor=sb/(4.*pi*(1.-querkon))
c Korngrenzenenergie nach Read-Shockley bei der zuerst Keimbildung
c auftritt minimaler kritischer Winkel, SI Einheiten
      keim_mis=keim_mis*rad
      if(keim_mis.lt.0.00001) keim_mis=0.00001
      gam_mis=gam_vor*sin(keim_mis)*(1.-log(sin(keim_mis)))
c SI Einheiten fuer d0_self
      d0_self=d0_self*centi*centi
c SI Einheiten fuer q_self
      q_self=q_self*ev
c Eichwerte fuer die Anschwingfrequenz ******************************
c Kleinste Aktivierungsenergie finden
      q_min=wach_erg
c     mobq_min=exp(log_a0)*mikro*mikro/korn_erg
      mobq_min=mob0
```

```
      do 279 ikg=1,i spez
      if(wach s(ikg).le.q_min) then
      q min=wach s(ikg)
c     mobq min=exp(log a0s(ikg))*mikro*mikro/korn_s(ikg)
      mobq min=mob0s(ikg)
      write(*,*)    'wach s(ikg),q min',wach_s(ikg),q_min
      write(*,*)    'mob0',mob0s(ikg)
      end if
279   continue

c Aktivierungsenergie fuer das Wachstum [J] ****************************
      wach erg=wach erg*ev
c Richtige Einheiten fuer die speziellen Korngrenzen
c spezielle KG
      do 278 ikg=1,i spez
      wach s(ikg)=wach_s(ikg)*ev
278   continue

c Neues Feld belegen als Produkt von Vorfaktor und Boltzmannterm
c der Mobilitaet fuer alle speziellen Korngrenzen
c Das Feld wird spaeter in der Wachstumsschleife benoetigt
      do 288 ikg=1,i spez
      mob exp(ikg)=mob0s(ikg)*exp(-wach_s(ikg)*dukbt)
288   continue

c Eichwerte fuer die Anschwingfrequenz in richtiger SI Einheiten *******
c q_min: minimal auftretende Aktivierungsenergie
      q min=q min*ev
c-------------------------------------------------------------------------
c-------------------------------------------------------------------------



c Einlesen der ASCII - Startdaten von rx_konti_map.txt ******************
c des verformten Startgefueges
      read(uni ein2,fmt=100)
      read(uni ein2,fmt=100)
      read(uni ein2,fmt=100)
      read(uni ein2,fmt=102) dum,rho rx
      read(uni ein2,fmt=102) dum,rho_situ
      read(uni ein2,fmt=100)
      read(uni ein2,fmt=102) dum,i randx
      read(uni ein2,fmt=102) dum,i randy
      read(uni ein2,fmt=102) dum,i_3d
      read(uni ein2,fmt=100)
      read(uni ein2,fmt=100)
      read(uni_ein2,fmt=100)

      z shiftx=i randx+1
      z_shifty=i_randy+1

c Kontrollieren der Feldgroesse auf Zulaessigkeit **********************
      if(i 3d.gt.(feld 3d-2)) then
      write(*,*) '==>  zulaessige Feldgroesse (3D) ueberschritten'
      goto 999
      end if
      if(i randx.gt.(feldrandx-2)) then
      write(*,*) '==>  zulaessige Feldgroesse ueberschritten'
      goto 999
      end if
      if(i randy.gt.(feldrandy-2)) then
      write(*,*) '==>  zulaessige Feldgroesse ueberschritten'
      goto 999
      end if

c Verschiebungsvektor wegen zyklischer Randbedingungen *****************
      i_shift=1

      rand tatx=i randx+1
      rand minx=i randx-i shift+1
      rand_ausx=i_randx+i_shift+1

      rand taty=i_randy+1
      rand miny=i randy-i shift+1
      rand_ausy=i_randy+i_shift+1

      ra3d tat=i 3d+1
      ra3d min=i_3d-i_shift+1
      ra3d_aus=i_3d+i_shift+1
```

```
c Einlesen der Verformungsdaten ***************************************
      rho_max=x_null
c bei einlesen : zuerst bleibt z, dann x und dann erst y konstant ******
c k : z-Koordinate
c i : x-Koordinate
c j : y-Koordinate
      do 210 k=2,ra3d_tat
      do 210 i=2,rand_tatx
c      write(*,*) 'x=', i
      do 210 j=2,rand_taty
      read(uni_ein2,fmt=1129) rho_alt(i,j,k),
     $fi1_alt(i,j,k),fi_alt(i,j,k),fi2_alt(i,j,k)

      if(float(rho_alt(i,j,k)).gt.rho_max) rho_max=float(rho_alt(i,j,k))
      rho_neu(i,j,k)=rho_alt(i,j,k)
      fi1_neu(i,j,k)=fi1_alt(i,j,k)
      fi_neu(i,j,k)=fi_alt(i,j,k)
      fi2_neu(i,j,k)=fi2_alt(i,j,k)
210   continue


c Einlesen der Verformungsdaten ***************************************
c BINAER
c k : z-Koordinate
c i : x-Koordinate
c j : y-Koordinate
c      do 211 k=2,ra3d_tat
c      do 211 i=2,rand_tatx
c      do 211 j=2,rand_taty
c
c      read(uni_ein2) xxx
c      rho_alt(i,j,k)=ifix(xxx)
cc      write(*,*) 'Versetzungen ',i,j,k,rho_alt(i,j,k)
c
c      read(uni_ein2) xxx
c      ori_alt(i,j,k)=ifix(xxx)
cc      write(*,*) 'Orientierung ',i,j,k,ori_alt(i,j,k)
c
c      if(rho_alt(i,j,k).gt.rho_max) rho_max=float(rho_alt(i,j,k))
c211   continue


      if(rho_max.gt.1500.) then
      write(*,*) '==>  Versetzungsdichte zu hoch'
      goto 999
      end if

c Einlesen der Verformungsdaten beenden
      close(uni_ein2)



c Die maximal auftretende Versetzungsdichte in SI Einheiten umrechnen
      rho_max=rho_max*hoch13
c maximal auftretende treibende Kraft [N/(m*m)] ***********************
      p_max=sbb/(4.*pi*(1.-querkon))*rho_max*
     $(0.5-log(sqrt(rho_max)*burgers))

c maximal auftretender Vorfaktor aus der treibenden Kraft [N/(m*m)] ******
c      rhovor=rho_max*(0.5-log(sqrt(rho_max)*burgers))
c maximale Mobilitaet
c      xmobvor=mobq_min*exp(-q_min*dukbt)
c Vorfaktor fuer die Loops
c     vor=sigma/(rhovor*xmobvor)
c     write(*,*) 'vor',vor

c Vorfaktor KB energie bei orientierter KB: p_max-Quadrat*16 * pi / 3
      erg_vor=16.*pi/(3.*p_max*p_max)
c Netzanschwingfrequenz (meso) [1/s] **********************************
      freq0=(mobq_min*p_max*exp(-q_min*dukbt))/(lambda*sigma)
      write(*,*) 'freq0',freq0
      write(*,*) '1./freq0',1./freq0
      write(*,*) ' v_max',p_max*mobq_min*exp(-q_min*dukbt)

C RANDBEDINGUNGEN ****************************************************

      if(zyklisch.gt.0)  then
c Vorbelegung fuer zyklische Randbedingung ****************************
c Vorbelegung der Aussen- und Innenbereiche (Fortschreibung) ***********
c Alle drei Seitenflaechen getrennt behandeln **************************
      do 1377 k=1,ra3d_aus
      do 1377 i=1,rand_ausx
```

```
      fi1_alt(i,1,k)=fi1_alt(i,rand_taty,k)
      fi_alt(i,1,k)=fi_alt(i,rand_taty,k)
      fi2_alt(i,1,k)=fi2_alt(i,rand_taty,k)

      fi1_alt(i,rand_ausy,k)=fi1_alt(i,2,k)
      fi_alt(i,rand_ausy,k)=fi_alt(i,2,k)
      fi2_alt(i,rand_ausy,k)=fi2_alt(i,2,k)

      fi1_neu(i,1,k)=fi1_neu(i,rand_taty,k)
      fi_neu(i,1,k)=fi_neu(i,rand_taty,k)
      fi2_neu(i,1,k)=fi2_neu(i,rand_taty,k)

      fi1_neu(i,rand_ausy,k)=fi1_neu(i,2,k)
      fi_neu(i,rand_ausy,k)=fi_neu(i,2,k)
      fi2_neu(i,rand_ausy,k)=fi2_neu(i,2,k)

      rho_alt(i,1,k)=rho_alt(i,rand_taty,k)
      rho_alt(i,rand_ausy,k)=rho_alt(i,2,k)
      rho_neu(i,1,k)=rho_neu(i,rand_taty,k)
      rho_neu(i,rand_ausy,k)=rho_neu(i,2,k)
1377   continue

      do 1477 k=1,ra3d_aus
      do 1477 i=1,rand_ausy

      fi1_alt(1,i,k)=fi1_alt(rand_tatx,i,k)
      fi_alt(1,i,k)=fi_alt(rand_tatx,i,k)
      fi2_alt(1,i,k)=fi2_alt(rand_tatx,i,k)

      fi1_alt(rand_ausx,i,k)=fi1_alt(2,i,k)
      fi_alt(rand_ausx,i,k)=fi_alt(2,i,k)
      fi2_alt(rand_ausx,i,k)=fi2_alt(2,i,k)

      fi1_neu(1,i,k)=fi1_neu(rand_tatx,i,k)
      fi_neu(1,i,k)=fi_neu(rand_tatx,i,k)
      fi2_neu(1,i,k)=fi2_neu(rand_tatx,i,k)

      fi1_neu(rand_ausx,i,k)=fi1_neu(2,i,k)
      fi_neu(rand_ausx,i,k)=fi_neu(2,i,k)
      fi2_neu(rand_ausx,i,k)=fi2_neu(2,i,k)

      rho_alt(1,i,k)=rho_alt(rand_tatx,i,k)
      rho_alt(rand_ausx,i,k)=rho_alt(2,i,k)
      rho_neu(1,i,k)=rho_neu(rand_tatx,i,k)
      rho_neu(rand_ausx,i,k)=rho_neu(2,i,k)
1477   continue


      do 1577 i=1,rand_ausx
      do 1577 j=1,rand_ausy

      fi1_alt(i,j,1)=fi1_alt(i,j,ra3d_tat)
      fi_alt(i,j,1)=fi_alt(i,j,ra3d_tat)
      fi2_alt(i,j,1)=fi2_alt(i,j,ra3d_tat)

      fi1_alt(i,j,ra3d_aus)=fi1_alt(i,j,2)
      fi_alt(i,j,ra3d_aus)=fi_alt(i,j,2)
      fi2_alt(i,j,ra3d_aus)=fi2_alt(i,j,2)

      fi1_neu(i,j,1)=fi1_neu(i,j,ra3d_tat)
      fi_neu(i,j,1)=fi_neu(i,j,ra3d_tat)
      fi2_neu(i,j,1)=fi2_neu(i,j,ra3d_tat)

      fi1_neu(i,j,ra3d_aus)=fi1_neu(i,j,2)
      fi_neu(i,j,ra3d_aus)=fi_neu(i,j,2)
      fi2_neu(i,j,ra3d_aus)=fi2_neu(i,j,2)

      rho_alt(i,j,1)=rho_alt(i,j,ra3d_tat)
      rho_alt(i,j,ra3d_aus)=rho_alt(i,j,2)
      rho_neu(i,j,1)=rho_neu(i,j,ra3d_tat)
      rho_neu(i,j,ra3d_aus)=rho_neu(i,j,2)
1577   continue
      end if
c Ende Vorbelegung fuer zyklische Randbedingungen



      if(zyklisch.lt.1)  then
c Vorbelegung der Raender fuer statische Randbedingung *****************
c Alle drei Seitenflaechen getrennt behandeln ***************************
      do 3377 k=1,ra3d_aus
```

```
      do 3377 i=1,rand_ausx

      fi1_alt(i,1,k)=fi1_alt(i,2,k)
      fi_alt(i,1,k)=fi_alt(i,2,k)
      fi2_alt(i,1,k)=fi2_alt(i,2,k)

      fi1_alt(i,rand_ausy,k)=fi1_alt(i,rand_taty,k)
      fi_alt(i,rand_ausy,k)=fi_alt(i,rand_taty,k)
      fi2_alt(i,rand_ausy,k)=fi2_alt(i,rand_taty,k)

      fi1_neu(i,1,k)=fi1_neu(i,2,k)
      fi_neu(i,1,k)=fi_neu(i,2,k)
      fi2_neu(i,1,k)=fi2_neu(i,2,k)

      fi1_neu(i,rand_ausy,k)=fi1_neu(i,rand_taty,k)
      fi_neu(i,rand_ausy,k)=fi_neu(i,rand_taty,k)
      fi2_neu(i,rand_ausy,k)=fi2_neu(i,rand_taty,k)

      rho_alt(i,1,k)=rho_alt(i,2,k)
      rho_alt(i,rand_ausy,k)=rho_alt(i,rand_taty,k)
      rho_neu(i,1,k)=rho_neu(i,2,k)
      rho_neu(i,rand_ausy,k)=rho_neu(i,rand_taty,k)
3377   continue

      do 3477 k=1,ra3d_aus
      do 3477 i=1,rand_ausy

      fi1_alt(1,i,k)=fi1_alt(2,i,k)
      fi_alt(1,i,k)=fi_alt(2,i,k)
      fi2_alt(1,i,k)=fi2_alt(2,i,k)

      fi1_alt(rand_ausx,i,k)=fi1_alt(rand_tatx,i,k)
      fi_alt(rand_ausx,i,k)=fi_alt(rand_tatx,i,k)
      fi2_alt(rand_ausx,i,k)=fi2_alt(rand_tatx,i,k)

      fi1_neu(1,i,k)=fi1_neu(2,i,k)
      fi_neu(1,i,k)=fi_neu(2,i,k)
      fi2_neu(1,i,k)=fi2_neu(2,i,k)

      fi1_neu(rand_ausx,i,k)=fi1_neu(rand_tatx,i,k)
      fi_neu(rand_ausx,i,k)=fi_neu(rand_tatx,i,k)
      fi2_neu(rand_ausx,i,k)=fi2_neu(rand_tatx,i,k)

      rho_alt(1,i,k)=rho_alt(2,i,k)
      rho_alt(rand_ausx,i,k)=rho_alt(rand_tatx,i,k)
      rho_neu(1,i,k)=rho_neu(2,i,k)
      rho_neu(rand_ausx,i,k)=rho_neu(rand_tatx,i,k)
3477   continue


      do 3577 i=1,rand_ausx
      do 3577 j=1,rand_ausy

      fi1_alt(i,j,1)=fi1_alt(i,j,2)
      fi_alt(i,j,1)=fi_alt(i,j,2)
      fi2_alt(i,j,1)=fi2_alt(i,j,2)

      fi1_alt(i,j,ra3d_aus)=fi1_alt(i,j,ra3d_tat)
      fi_alt(i,j,ra3d_aus)=fi_alt(i,j,ra3d_tat)
      fi2_alt(i,j,ra3d_aus)=fi2_alt(i,j,ra3d_tat)

      fi1_neu(i,j,1)=fi1_neu(i,j,2)
      fi_neu(i,j,1)=fi_neu(i,j,2)
      fi2_neu(i,j,1)=fi2_neu(i,j,2)

      fi1_neu(i,j,ra3d_aus)=fi1_neu(i,j,ra3d_tat)
      fi_neu(i,j,ra3d_aus)=fi_neu(i,j,ra3d_tat)
      fi2_neu(i,j,ra3d_aus)=fi2_neu(i,j,ra3d_tat)

      rho_alt(i,j,1)=rho_alt(i,j,2)
      rho_alt(i,j,ra3d_aus)=rho_alt(i,j,ra3d_tat)
      rho_neu(i,j,1)=rho_neu(i,j,2)
      rho_neu(i,j,ra3d_aus)=rho_neu(i,j,ra3d_tat)
3577   continue
      end if
C Ende Vorbelegung fuer nicht-zyklische Randbedingungen



c Rausschreiben der Startdaten ***************************************
      write(uni_au0,fmt=100)
```

```
 write(uni au0,fmt=100) 'Datum'
 write(uni au0,fmt=100) today
 write(uni au0,fmt=100) 'Uhrzeit,  h  min  sec'
 write(uni au0,fmt=298) ihr, imin, isec
 write(uni au0,fmt=100)
 if(zyklisch.lt.1)
$write(uni au0,fmt=100) 'Nicht-Zyklische Randbedingungen'
 if(zyklisch.gt.0)
$write(uni au0,fmt=100) 'Zyklische Randbedingungen'
 write(uni au0,fmt=100)
 write(uni au0,fmt=100) 'Maximal auftretende Versetzungsdichte'
 write(uni au0,fmt=100) '(in 10 hoch 13 pro Quadratmeter)'
 write(uni au0,fmt=108) rho_max/hoch13,' rho_max  [10^13 /m*m]'
 write(uni au0,fmt=100)
 write(uni au0,fmt=100) 'Versetzungsdichte nach Rekristallisation'
 write(uni au0,fmt=100) '(in 10 hoch 13 pro Quadratmeter)'
 write(uni au0,fmt=104) '# ',rho_rx,'   rho_rx  [10 hoch 13 /m*m]'
 write(uni au0,fmt=100)
 write(uni au0,fmt=100) 'Versetzungsdichte nach RX in-situ'
 write(uni au0,fmt=100) '(in 10 hoch 13 pro Quadratmeter)'
 write(uni au0,fmt=104) '# ',rho_situ,' rho_situ [10 hoch 13/m*m]'
 write(uni au0,fmt=100)
 write(uni au0,fmt=100) 'Quasi-Zellgroesse vorgegeben'
 write(uni au0,fmt=198) lambda,'   lambda [m]'
 write(uni au0,fmt=100)
 write(uni au0,fmt=100) 'Feldgroesse, x-Richtung'
 write(uni au0,fmt=104) '# ',rand_tatx-1,' rand_tatx-1 [1]'
 write(uni au0,fmt=100)
 write(uni au0,fmt=100) 'Feldgroesse, y-Richtung'
 write(uni au0,fmt=104) '# ',rand_taty-1,' rand_taty-1 [1]'
 write(uni au0,fmt=100)
 write(uni au0,fmt=100) 'Feldgroesse, z-Richtung'
 write(uni au0,fmt=104) '# ',ra3d_tat-1,' ra3d_tat-1 [1]'
 write(uni au0,fmt=100)
 v_max=p_max*mobq_min*exp(-q_min*dukbt)
 write(uni au0,fmt=108) lambda/mikro,' lambda    [um]'
 write(uni au0,fmt=108) lambda*sigma/mikro,' lambda*sigma [um]'
 write(uni au0,fmt=179) exp(-q_min*dukbt),' Boltzmann, v_max [1]'
 write(uni au0,fmt=108) v_max/mikro,'  v_max [um/s]'
 write(uni au0,fmt=108) keim_erg/ev,'  keim_erg, Aktiv.en der kb'
 write(uni au0,fmt=108) wach_erg/ev,' Aktiv.energie wach_erg  [eV]'
 write(uni au0,fmt=108) temp,'  Temperatur  temp   [K]'
 write(uni au0,fmt=108) temp_sch,'  Schmelzpunkt temp_sch  [K]'
 write(uni au0,fmt=108) burgers/aengst,'  b(T)   [Aengstroem]'
 write(uni au0,fmt=108) mobq_min/mikro,' min. mob0, 10E-6 m*m*m/Ns'
 write(uni au0,fmt=108) mob0/mikro,' Mobil. mob0, 10E-6  m*m*m/Ns'
 write(uni au0,fmt=108) schubmod/giga,' Schubmodul(T)    [GPa]'
 write(uni au0,fmt=108) freq0,'  attack freq in Hz'
 write(uni au0,fmt=108) 1./freq0,' Zeite., inv.attack freq in [s]'
 write(uni au0,fmt=108) q_min/ev,' min. auftr. aktiv.energ.'
 write(uni au0,fmt=108) p_max,'  maximal auftretende p    [Pa]'
 write(uni au0,fmt=108) querkon,'  Querkontraktion querkon [1]'
 write(uni au0,fmt=108) frac_abb,'  frac abb, Delta-RX - Abbruch'
 write(uni au0,fmt=108) frac_ab,'  frac ab, Absolut-RX - Abbruch'
 write(uni au0,fmt=108) keimstreu,'  Gauss - Keimstreuwinkel'
 write(uni au0,fmt=108) erg_max,'  krit. Vers.Schwellwert f. KBdg.'
 write(uni au0,fmt=108) rho_offset,'  rho_offset, Mindestvers.wert'
 write(uni au0,fmt=108) gef(1),'  gef(1) - rausschreiben'
 write(uni au0,fmt=108) gef(2),'  gef(2) - rausschreiben'
 write(uni au0,fmt=108) gef(3),'  gef(3) - rausschreiben'
 write(uni au0,fmt=108) gef(4),'  gef(4) - rausschreiben'
 write(uni au0,fmt=108) gef(5),'  gef(5) - rausschreiben'
 write(uni au0,fmt=108) gef(6),'  gef(6) - rausschreiben'
 write(uni au0,fmt=108) gef(7),'  gef(7) - rausschreiben'
 write(uni au0,fmt=108) gef(8),'  gef(8) - rausschreiben'
 write(uni au0,fmt=108) gef(9),'  gef(9) - rausschreiben'
 write(uni au0,fmt=108) gef(10),'  gef(10) - rausschreiben'
 write(uni au0,fmt=108) gef(11),'  gef(11) - rausschreiben'
 write(uni au0,fmt=108) gef(12),'  gef(12) - rausschreiben'
 write(uni au0,fmt=108) gef(13),'  gef(13) - rausschreiben'
 write(uni au0,fmt=108) gef(14),'  gef(14) - rausschreiben'
 write(uni au0,fmt=108) gef(15),'  gef(15) - rausschreiben'
 write(uni au0,fmt=108) gef(16),'  gef(16) - rausschreiben'
 write(uni au0,fmt=108) gef(17),'  gef(17) - rausschreiben'
 write(uni au0,fmt=108) gef(18),'  gef(18) - rausschreiben'
 write(uni au0,fmt=108) gef(19),'  gef(19) - rausschreiben'
 write(uni au0,fmt=108) gef(20),'  gef(20) - rausschreiben'
 write(uni au0,fmt=109) nachbar,' Nachbarschaft'
 write(uni au0,fmt=109) bildung,' Bildung'
 write(uni au0,fmt=108) wink_ab,' unterer Mob.Abschn.-winkel: m=0 '
 write(uni au0,fmt=108) keim_mis/rad,' Mind.wink. Keimbildung'
 write(uni_au0,fmt=108) gam_mis,' R-S KG Energie zu keim_mis in SI'
```

```
      write(uni_au0,fmt=100)


c Rausschreiben der Anfangstextur und -versetzungsverteilung *********
      open(unit=uni_anf_ori,recl=4,
     $file='c:\for\rx_konti2d\ori_anf.bin',
     $form='unformatted',status='unknown',access='direct')
      open(unit=uni_anf_rho,recl=4,
     $file='c:\for\rx_konti2d\rho_anf.bin',
     $form='unformatted',status='unknown',access='direct')

      do 1620 iiiz=2,ra3d_tat
      do 1620 iiix=2,rand_tatx
      do 1620 iiiy=2,rand_taty
      xxx=float(rho_alt(iiix,iiiy,iiiz))
      write(uni_anf_rho) xxx

      xxx=fi1_alt(iiix,iiiy,iiiz)
      write(uni_anf_ori) xxx
      xxx=fi_alt(iiix,iiiy,iiiz)
      write(uni_anf_ori) xxx
      xxx=fi2_alt(iiix,iiiy,iiiz)
      write(uni_anf_ori) xxx
1620  continue
      close(uni_anf_ori)
      close(uni_anf_rho)


c ######################################################################
c ############################  Keimbildung ############################
c ######################################################################
      wurf=ran(iseed)
      lauf=ifix(wurf*200.)
      write(*,*) 'Keimbildungsphase'

c Rausschreiben der Keimbildungstextur ****************************
      open(unit=uni_keim_ori,
     $file='c:\for\rx_konti2d\rx_keim2d.ori')


c ######################################################################
c ###############  Regellose ortsgesaettigte Keimbildung #############
c ###############  Regellos in Ort und Eulerraum   #############
c ######################################################################
c Regellose, ortsgesaettigte Keimbildung
c Berechnung der Keimzahl aus thermischer Wahrcheinlichkeit
c unter Beruecksichtigung der Anzahl der vorhandenen Gitterplaetze
c die Orientierung des Keims ist nicht an die des Mutterkorns gebunden
      if(bildung.eq.0) then
c Rausschreiben der Keimdaten aus rx4_geo
      write(uni_au0,fmt=100) '        Keimdaten'
      write(uni_au0,fmt=100) 'Keimbildung im Einkristall'
      write(uni_au0,fmt=100) 'ortsgesaettigte Keimbildung'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) 'Keimenergie   Temperatur'
c Keimzahl zur Zeit t=0 berechnen
      write(uni_au0,fmt=115) keim_erg/ev,temp
      wahrsch=exp(-keim_erg*dukbt)
c     write(*,*) wahrsch

      write(uni_au0,fmt=198) wahrsch,' Keimbildwahrscheinlichkeit'
      keimzahl=ifix(float(i_randx*i_randy*i_3d)*wahrsch)
      write(*,fmt=109) keimzahl,' Keimzahl bei Ortssaettigung'
      write(uni_au0,fmt=109) keimzahl,' Keimzahl bei Ortssaettigung'

      probe=lambda*lambda*lambda
      xdichte=keimzahl/probe
      write(*,*) 'Keimdichte in 1/(m*m*m)',xdichte
      write(uni_au0,*) 'Keimdichte in 1/(m*m*m)',xdichte

      write(uni_au0,fmt=109) i_randx,' i_randx - Feldgroesse in x'
      write(uni_au0,fmt=109) i_randy,' i_randy - Feldgroesse in y'
      write(uni_au0,fmt=109) i_3d,' i_3d - Feldtiefe '
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '      Art und Position neuer Keime'
      write(uni_au0,fmt=100) 'x-Koord  y-Koord  z-Koord  Keimorientier.'


      keimvol=100.
      if(keimzahl.gt.0) keimvol=100./keimzahl

      do 212 i=1,keimzahl
```

```
      wurf=ran(iseed)
      lauf=ifix(wurf*200.)

      do 2212 ia=1,lauf
2212  wurf=ran(iseed)


      wurf=ran(iseed)
      iwurf=ifix(wurf*float(i_randx))
      if(iwurf.lt.1) iwurf=1
      ix= iwurf+i_shift
c     write(*,*) wurf,ix

      do 2213 ia=1,lauf
2213  wurf=ran(iseed)
      wurf=ran(iseed)
      iwurf=ifix(wurf*float(i_randy))
      if(iwurf.lt.1) iwurf=1
      iy= iwurf+i_shift
c     write(*,*) wurf,iy

      do 2214 ia=1,lauf
2214  wurf=ran(iseed)
      wurf=ran(iseed)
      iwurf=ifix(wurf*float(i_3d))
      if(iwurf.lt.1) iwurf=1
      iz= iwurf+i_shift
c     write(*,*) wurf,iz

      do 2614 ia=1,lauf
2614  wurf=ran(iseed)


c Neue Orientierung wuerfeln ****************************************
c hat keinerlei Beziehung zum bestehenden Gefuege,
c Keimbildung ist also auch bezgl. Kristallorientierung voellig regellos
      do 2215 ia=1,lauf
2215  wurf=ran(iseed)
      wurf=ran(iseed)
      fi1_neu(ix,iy,iz)=wurf*eulermax
      fi1_alt(ix,iy,iz)=fi1_neu(ix,iy,iz)

      do 3215 ia=1,lauf
3215  wurf=ran(iseed)
      wurf=ran(iseed)
      fi_neu(ix,iy,iz)=wurf*eulermax
      fi_alt(ix,iy,iz)=fi_neu(ix,iy,iz)

      do 4215 ia=1,lauf
4215  wurf=ran(iseed)
      wurf=ran(iseed)
      fi2_neu(ix,iy,iz)=wurf*eulermax
      fi2_alt(ix,iy,iz)=fi2_neu(ix,iy,iz)

      rho_neu(ix,iy,iz)=rho_rx
      rho_alt(ix,iy,iz)=rho_rx


      write(uni_au0,fmt=118) ix-i_shift,iy-i_shift,iz-i_shift,
     $fi1_neu(ix,iy,iz),fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),
     $'  [ x y z fi1 Fi fi2]'


      write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu

212   continue
      end if
c Ende ortsgesaettigte Keimbildung ***********************************




c ####################################################################
c ################ Keimbildungsmodell ################################
c  NUR die Zellen mit kritischer Versetzungsdichte
c ####################################################################
c ####################################################################
c ####################################################################
```

```
      if(bildung.eq.10) then

      keimvol=0.099

c Rausschreiben der Keimdaten
      write(uni_au0,fmt=100) '        Keimdaten'
      write(uni_au0,fmt=100) 'Keimbildung im Einkristall'
      write(uni_au0,fmt=100) 'ortsgesaettigte Keimbildung'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) 'Keimenergie   Temperatur'


      write(uni_au0,fmt=109) i_randx,' i_randx - Feldgroesse in x'
      write(uni_au0,fmt=109) i_randy,' i_randy - Feldgroesse in y'
      write(uni_au0,fmt=109) i_3d,' i_3d - Feldtiefe '
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '      Art und Position neuer Keime'
      write(uni_au0,fmt=100) 'x-Koord  y-Koord  z-Koord  Keimorientier.'

      wink_ab=keim_mis/rad

      do 6311 iz=2,ra3d_tat
      do 6311 ix=2,rand_tatx
      do 6311 iy=2,rand_taty


c erst ueberpruefen, ob die Versetzungsdichten eine kritische Grenze ueberschreitet
      if(rho_neu(ix,iy,iz).gt.erg_max) then

c dann alle Punkte absuchen, ob die Nachbarorientierung anders ist
      if(fi1_neu(ix,iy,iz).ne.fi1_neu(ix+1,iy,iz).and.
     $fi_neu(ix,iy,iz).ne.fi_neu(ix+1,iy,iz).and.
     $fi2_neu(ix,iy,iz).ne.fi2_neu(ix+1,iy,iz)) then

c jetzt die tatsaechliche Misorientierung zwischen beiden
c Orientierungen berechnen
      fi1(1)=fi1_neu(ix,iy,iz)
      fi(1)=fi_neu(ix,iy,iz)
      fi2(1)=fi2_neu(ix,iy,iz)

      fi1(2)=fi1_neu(ix+1,iy,iz)
      fi(2)=fi_neu(ix+1,iy,iz)
      fi2(2)=fi2_neu(ix+1,iy,iz)

      call misori(fi1,fi,fi2,winkel,iachs)

c Abfragen, ob Misorientierung groesser ist als 15 Grad
      if(winkel.ge.wink_ab) then

      if(rho_neu(ix,iy,iz).ge.rho_neu(ix+1,iy,iz)) then
c Neue orientierung beiden zellen zuordnen
            fi1_neu(ix,iy,iz)=fi1(1)
            fi_neu(ix,iy,iz)=fi(1)
            fi2_neu(ix,iy,iz)=fi2(1)

            fi1_neu(ix+1,iy,iz)=fi1(1)
            fi_neu(ix+1,iy,iz)=fi(1)
            fi2_neu(ix+1,iy,iz)=fi2(1)

            fi1_alt(ix,iy,iz)=fi1(1)
            fi_alt(ix,iy,iz)=fi(1)
            fi2_alt(ix,iy,iz)=fi2(1)

            fi1_alt(ix+1,iy,iz)=fi1(1)
            fi_alt(ix+1,iy,iz)=fi(1)
            fi2_alt(ix+1,iy,iz)=fi2(1)

            rho_neu(ix,iy,iz)=rho_rx
            rho_neu(ix+1,iy,iz)=rho_rx

            rho_alt(ix,iy,iz)=rho_rx
            rho_alt(ix+1,iy,iz)=rho_rx

      end if




      if(rho_neu(ix,iy,iz).lt.rho_neu(ix+1,iy,iz)) then
c Neue orientierung beiden zellen zuordnen
            fi1_neu(ix,iy,iz)=fi1(2)
            fi_neu(ix,iy,iz)=fi(2)
```

```
            fi2_neu(ix,iy,iz)=fi2(2)

            fi1_neu(ix+1,iy,iz)=fi1(2)
            fi_neu(ix+1,iy,iz)=fi(2)
            fi2_neu(ix+1,iy,iz)=fi2(2)

            fi1_alt(ix,iy,iz)=fi1(2)
            fi_alt(ix,iy,iz)=fi(2)
            fi2_alt(ix,iy,iz)=fi2(2)

            fi1_alt(ix+1,iy,iz)=fi1(2)
            fi_alt(ix+1,iy,iz)=fi(2)
            fi2_alt(ix+1,iy,iz)=fi2(2)

            rho_neu(ix,iy,iz)=rho_rx
            rho_neu(ix+1,iy,iz)=rho_rx

            rho_alt(ix,iy,iz)=rho_rx
            rho_alt(ix+1,iy,iz)=rho_rx

      end if

      write(uni_au0,fmt=118) ix-i_shift,iy-i_shift,iz-i_shift,
     $fi1_neu(ix,iy,iz),fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),
     $'  [ x y z fi1 Fi fi2]'

      write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu

      end if
      end if
      end if

6311   continue
      end if









c ######################################################################
c ################ Keimbildungsmodell ##################################
c  2 Zellen mit kritischer Versetzungsdichte
c    und kritsicher Missorientierung
c ######################################################################
c ######################################################################
c ######################################################################

      if(bildung.eq.9) then

      keimvol=0.099

c Rausschreiben der Keimdaten
      write(uni_au0,fmt=100) '        Keimdaten'
      write(uni_au0,fmt=100) 'Keimbildung im Einkristall'
      write(uni_au0,fmt=100) 'ortsgesaettigte Keimbildung'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) 'Keimenergie   Temperatur'


      write(uni_au0,fmt=109) i_randx,' i_randx - Feldgroesse in x'
      write(uni_au0,fmt=109) i_randy,' i_randy - Feldgroesse in y'
      write(uni_au0,fmt=109) i_3d,' i_3d - Feldtiefe '
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '      Art und Position neuer Keime'
      write(uni_au0,fmt=100) 'x-Koord  y-Koord  z-Koord  Keimorientier.'

      wink_ab=keim_mis/rad

      do 4311 iz=2,ra3d_tat
      do 4311 ix=2,rand_tatx
      do 4311 iy=2,rand_taty


c erst ueberpruefen, ob die Versetzungsdichten eine kritische Grenze ueberschreitet
      if(rho_neu(ix,iy,iz).gt.erg_max) then
```

```
c dann alle Punkte absuchen, ob die Nachbarorientierung anders ist
      if(fi1_neu(ix,iy,iz).ne.fi1_neu(ix+1,iy,iz).and.
     $fi_neu(ix,iy,iz).ne.fi_neu(ix+1,iy,iz).and.
     $fi2_neu(ix,iy,iz).ne.fi2_neu(ix+1,iy,iz)) then

c jetzt die tatsaechliche Misorientierung zwischen beiden
c Orientierungen berechnen
      fi1(1)=fi1_neu(ix,iy,iz)
      fi(1)=fi_neu(ix,iy,iz)
      fi2(1)=fi2_neu(ix,iy,iz)

      fi1(2)=fi1_neu(ix+1,iy,iz)
      fi(2)=fi_neu(ix+1,iy,iz)
      fi2(2)=fi2_neu(ix+1,iy,iz)

      call misori(fi1,fi,fi2,winkel,iachs)

c Abfragen, ob Misorientierung groesser ist als 15 Grad
      if(winkel.ge.wink_ab) then

      if(rho_neu(ix,iy,iz).ge.rho_neu(ix+1,iy,iz)) then
c Neue orientierung beiden zellen zuordnen
            fi1_neu(ix,iy,iz)=fi1(1)
            fi_neu(ix,iy,iz)=fi(1)
            fi2_neu(ix,iy,iz)=fi2(1)

            fi1_neu(ix+1,iy,iz)=fi1(1)
            fi_neu(ix+1,iy,iz)=fi(1)
            fi2_neu(ix+1,iy,iz)=fi2(1)

            fi1_alt(ix,iy,iz)=fi1(1)
            fi_alt(ix,iy,iz)=fi(1)
            fi2_alt(ix,iy,iz)=fi2(1)

            fi1_alt(ix+1,iy,iz)=fi1(1)
            fi_alt(ix+1,iy,iz)=fi(1)
            fi2_alt(ix+1,iy,iz)=fi2(1)

            rho_neu(ix,iy,iz)=rho_rx
            rho_neu(ix+1,iy,iz)=rho_rx

            rho_alt(ix,iy,iz)=rho_rx
            rho_alt(ix+1,iy,iz)=rho_rx

      end if



      if(rho_neu(ix,iy,iz).lt.rho_neu(ix+1,iy,iz)) then
c Neue orientierung beiden zellen zuordnen
            fi1_neu(ix,iy,iz)=fi1(2)
            fi_neu(ix,iy,iz)=fi(2)
            fi2_neu(ix,iy,iz)=fi2(2)

            fi1_neu(ix+1,iy,iz)=fi1(2)
            fi_neu(ix+1,iy,iz)=fi(2)
            fi2_neu(ix+1,iy,iz)=fi2(2)

            fi1_alt(ix,iy,iz)=fi1(2)
            fi_alt(ix,iy,iz)=fi(2)
            fi2_alt(ix,iy,iz)=fi2(2)

            fi1_alt(ix+1,iy,iz)=fi1(2)
            fi_alt(ix+1,iy,iz)=fi(2)
            fi2_alt(ix+1,iy,iz)=fi2(2)

            rho_neu(ix,iy,iz)=rho_rx
            rho_neu(ix+1,iy,iz)=rho_rx

            rho_alt(ix,iy,iz)=rho_rx
            rho_alt(ix+1,iy,iz)=rho_rx

      end if

      write(uni_au0,fmt=118) ix-i_shift,iy-i_shift,iz-i_shift,
     $fi1_neu(ix,iy,iz),fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),
     $'  [ x y z fi1 Fi fi2]'

      write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu
```

```
      end if
      end if
      end if

4311   continue
      end if




c ######################################################################
c #############    MODELL B  DER KEIMBILDUNG    ###############
c Einfach alle Zellen mit einer Versetzungsdichte oberhalb der kritischen
c Versetzungsdichte als RX einstufen. Jede Zelle hat dann ja die Chance
c bei ausreichender kinetischer Instabilitaet der Umgebung
c weiterzuwachsen
c ######################################################################
c ######################################################################
c ######################################################################

      if(bildung.eq.11) then

      keimvol=0.099

c Rausschreiben der Keimdaten
      write(*,*) '    Keimbildung nach Modell B (bildung eq 11)'

      write(uni_au0,fmt=100) '    Keimdaten Modell B (11)'
      write(uni_au0,fmt=100) 'Keimbildung im Einkristall'
      write(uni_au0,fmt=100) 'ortsgesaettigte Keimbildung'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) 'Keimenergie   Temperatur'


      write(uni_au0,fmt=109) i_randx,' i_randx - Feldgroesse in x'
      write(uni_au0,fmt=109) i_randy,' i_randy - Feldgroesse in y'
      write(uni_au0,fmt=109) i_3d,' i_3d - Feldtiefe '
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '      Art und Position neuer Keime'
      write(uni_au0,fmt=100) 'x-Koord  y-Koord  z-Koord  Keimorientier.'

      do 5311 iz=2,ra3d_tat
      do 5311 ix=2,rand_tatx
      do 5311 iy=2,rand_taty


c Ueberpruefen, ob die Versetzungsdichte eine kritische
c Grenze erreicht oder ueberschreitet
      if(rho_neu(ix,iy,iz).ge.erg_max) then

            rho_neu(ix,iy,iz)=rho_rx
            rho_alt(ix,iy,iz)=rho_rx

      write(uni_au0,fmt=118) ix-i_shift,iy-i_shift,iz-i_shift,
     $fi1_neu(ix,iy,iz),fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),
     $'  [ x y z fi1 Fi fi2]'

      write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu

      end if

5311   continue
      end if

C Modell B  (11)  zuende




c ######################################################################
c ###############  ortsgesaettigte Keimbildung #############
c ###############  regellos im Eulerraum, aber an Inhomogenitaeten ####
c ###############  und mit bestimmter minimal Misorientierung ####
c ######################################################################
c Regellose, ortsgesaettigte Keimbildung
c Berechnung der Keimzahl aus thermischer Wahrcheinlichkeit
c unter Beruecksichtigung der Anzahl der vorhandenen Gitterplaetze
```

```
c die Orientierung des Keims ist nicht an die des Mutterkorns gebunden
      if(bildung.eq.8) then

      keimvol=0.1

c Rausschreiben der Keimdaten aus rx4_geo
      write(uni_au0,fmt=100) '        Keimdaten'
      write(uni_au0,fmt=100) 'Keimbildung im Einkristall'
      write(uni_au0,fmt=100) 'ortsgesaettigte Keimbildung'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) 'Keimenergie   Temperatur'
c Keimzahl zur Zeit t=0 berechnen
c      write(uni_au0,fmt=115) keim_erg/ev,temp
c      wahrsch=exp(-keim_erg*dukbt)
c     write(*,*) wahrsch

c      write(uni_au0,fmt=198) wahrsch,' Keimbildwahrscheinlichkeit'
c      keimzahl=ifix(float(i_randx*i_randy*i_3d)*wahrsch)
c      write(*,fmt=109) keimzahl,' Keimzahl bei Ortssaettigung'
c      write(uni_au0,fmt=109) keimzahl,' Keimzahl bei Ortssaettigung'

c      probe=lambda*lambda*lambda
c     write(*,*) 'Gesamtvolumen der Probe in m*m*m',probe
c     write(uni_au0,*) 'Gesamtvolumen der Probe in m*m*m',probe
c     xdichte=keimzahl/probe
c     write(*,*) 'Keimdichte in 1/(m*m*m)',xdichte
c     write(uni_au0,*) 'Keimdichte in 1/(m*m*m)',xdichte

c     write(*,*) 'Keimdichte in 1/(um*um*um)',wahrsch
c     write(uni_au0,*) 'Keimdichte in 1/(um*um*um)',wahrsch


      write(uni_au0,fmt=109) i_randx,' i_randx - Feldgroesse in x'
      write(uni_au0,fmt=109) i_randy,' i_randy - Feldgroesse in y'
      write(uni_au0,fmt=109) i_3d,' i_3d - Feldtiefe '
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) '      Art und Position neuer Keime'
      write(uni_au0,fmt=100) 'x-Koord  y-Koord  z-Koord  Keimorientier.'

      wink_ab=keim_mis/rad


      do 3311 iz=2,ra3d_tat
      do 3311 ix=2,rand_tatx
      do 3311 iy=2,rand_taty


         if(fi1_neu(ix,iy,iz).ne.fi1_neu(ix+1,iy,iz)) then

         if(rho_neu(ix,iy,iz).gt.erg_max) then

      fi1(1)=fi1_neu(ix,iy,iz)
      fi(1)=fi_neu(ix,iy,iz)
      fi2(1)=fi2_neu(ix,iy,iz)

      fi1(2)=fi1_neu(ix+1,iy,iz)
      fi(2)=fi_neu(ix+1,iy,iz)
      fi2(2)=fi2_neu(ix+1,iy,iz)

      call misori(fi1,fi,fi2,winkel,iachs)

            if(winkel.gt.wink_ab) then

c Neue Orientierung um den halben Winkel verdreht
c Winkel halbieren
      wink_h=(winkel/2.)*rad
c Drehachse
      xa1=float(iachs(1))
      xa2=float(iachs(2))
      xa3=float(iachs(3))
c Drehmatrix zu Achse - Halber-Winkel berechnen
      call rotmat(wink_h,xa1,xa2,xa3,dr3)
c matrixorientierung zu drehmatrix umformen
      call euldreh(fi1(1),fi(1),fi2(1),dr1)
c matrixorientierung um den halben winkel drehen
c dreht dr1 um dr3
c dr1 :matrix
c dr3 :rotation
      call drehen(dr1,dr3)
c dr1 ist nun die neue, gedrehte matrix/keimorientierung in der mitte
c zwischen beiden
      call dreheul(dr1,x1,x2,x3,fi1(1),fi(1),fi2(1),xpp,xppr)
```

```
C    Berechnung der Eulerwinkel aus der Dreh-Matrix dr2
C    jeweils reduziert in den 1. Unterraum
C    mit:  0 < Fi,Fi2 < 90ø und 0 < Fi1< 360ø (allg. Eulerwinkel)
C    und:  0 < FiR1,FiR,FiR2 < 90ø  (reduzierte Eulerwinkel )
c




c Neue orientierung beiden zellen zuordnen
            fi1_neu(ix,iy,iz)=fi1(1)
            fi_neu(ix,iy,iz)=fi(1)
            fi2_neu(ix,iy,iz)=fi2(1)

            fi1_neu(ix+1,iy,iz)=fi1(1)
            fi_neu(ix+1,iy,iz)=fi(1)
            fi2_neu(ix+1,iy,iz)=fi2(1)

            fi1_alt(ix,iy,iz)=fi1(1)
            fi_alt(ix,iy,iz)=fi(1)
            fi2_alt(ix,iy,iz)=fi2(1)

            fi1_alt(ix+1,iy,iz)=fi1(1)
            fi_alt(ix+1,iy,iz)=fi(1)
            fi2_alt(ix+1,iy,iz)=fi2(1)

            rho_neu(ix,iy,iz)=rho_rx
            rho_neu(ix+1,iy,iz)=rho_rx

            rho_alt(ix,iy,iz)=rho_rx
            rho_alt(ix+1,iy,iz)=rho_rx
c Hier wurden beide Zellen rekristallisert

      write(uni_au0,fmt=118) ix-i_shift,iy-i_shift,iz-i_shift,
     $fi1_neu(ix,iy,iz),fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),
     $'  [ x y z fi1 Fi fi2]'

      write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu

      end if
      end if
      end if

3311   continue
      end if



c ######################################################################
c ######  Keimbildungsfront mit einheitlicher Orientierung #########
c ######################################################################
c Einheitliche Keimbildungsfront fuer KG-wanderungstest ****************
      if(bildung.eq.2) then

      keimvol=100./(float(rand_minx+1)*float(ra3d_tat+1))

      do 254 iz=1,ra3d_tat+1
      do 254 iy=1,rand_miny+1

      rho_neu(rand_tatx,iy,iz)=rho_rx
      rho_alt(rand_tatx,iy,iz)=rho_rx

      fi1_neu(rand_tatx,iy,iz)=fi1keim
      fi1_alt(rand_tatx,iy,iz)=fi1keim
      fi_neu(rand_tatx,iy,iz)=fikeim
      fi_alt(rand_tatx,iy,iz)=fikeim
      fi2_neu(rand_tatx,iy,iz)=fi2keim
      fi2_alt(rand_tatx,iy,iz)=fi2keim

      write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu

254   continue


      end if
c Ende Keimbildungsfront ***********************************************


c ######################################################################
c ##########################  Nur ein Keim ##########################
```

```
c ######################################################################
      if(bildung.eq.4) then

      keimvol=100.

c      wurf=ran(iseed)
c      wurf fi1=wurf*eulermax
c      wurf=ran(iseed)
c      wurf fi=wurf*eulermax
c      wurf=ran(iseed)
c      wurf_fi2=wurf*eulermax

            iz=2+ifix((float(ra3d tat)-float(2))/2.)
            ix=2+ifix((float(rand_tatx)-float(2))/2.)
            iy=2+ifix((float(rand_taty)-float(2))/2.)

c keimbildung

      fi1 neu(ix,iy,iz)=fi1keim
      fi1 alt(ix,iy,iz)=fi1keim
      fi neu(ix,iy,iz)=fikeim
      fi alt(ix,iy,iz)=fikeim
      fi2 neu(ix,iy,iz)=fi2keim
      fi2_alt(ix,iy,iz)=fi2keim

            rho neu(ix,iy,iz)=rho rx
            rho_alt(ix,iy,iz)=rho_rx


      write(uni keim ori,fmt=919) fi1 neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu

      end if
c Ende Ein Keim ****************************************************


c ######################################################################
c ##########################  Vier Keime ###############################
c ######################################################################
      if(bildung.eq.5) then

      keimvol=0.25

            iz=2+ifix((float(ra3d tat)-float(2))/2.)
            ix=2+ifix((float(rand_tatx)-float(2))/4.)
            iy=2+ifix((float(rand_taty)-float(2))/4.)

c keimbildung
c Keim Nr 1
      wurf=ran(iseed)
      wurf fi1=wurf*eulermax
      wurf=ran(iseed)
      wurf fi=wurf*eulermax
      wurf=ran(iseed)
      wurf_fi2=wurf*eulermax

      fi1 neu(ix,iy,iz)=wurf fi1
      fi1 alt(ix,iy,iz)=wurf fi1
      fi neu(ix,iy,iz)=wurf fi
      fi alt(ix,iy,iz)=wurf fi
      fi2 neu(ix,iy,iz)=wurf fi2
      fi2_alt(ix,iy,iz)=wurf_fi2

      rho neu(ix,iy,iz)=rho rx
      rho_alt(ix,iy,iz)=rho_rx

      write(uni keim ori,fmt=919) fi1 neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu

c Keim Nr 2
      wurf=ran(iseed)
      wurf fi1=wurf*eulermax
      wurf=ran(iseed)
      wurf fi=wurf*eulermax
      wurf=ran(iseed)
      wurf_fi2=wurf*eulermax

      fi1 neu(3*ix,iy,iz)=wurf fi1
      fi1 alt(3*ix,iy,iz)=wurf fi1
      fi neu(3*ix,iy,iz)=wurf fi
      fi alt(3*ix,iy,iz)=wurf fi
      fi2_neu(3*ix,iy,iz)=wurf_fi2
```

```
      fi2_alt(3*ix,iy,iz)=wurf_fi2

      rho_neu(3*ix,iy,iz)=rho_rx
      rho_alt(3*ix,iy,iz)=rho_rx

      write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu


c Keim Nr 3
      wurf=ran(iseed)
      wurf_fi1=wurf*eulermax
      wurf=ran(iseed)
      wurf_fi=wurf*eulermax
      wurf=ran(iseed)
      wurf_fi2=wurf*eulermax

      fi1_neu(ix,3*iy,iz)=wurf_fi1
      fi1_alt(ix,3*iy,iz)=wurf_fi1
      fi_neu(ix,3*iy,iz)=wurf_fi
      fi_alt(ix,3*iy,iz)=wurf_fi
      fi2_neu(ix,3*iy,iz)=wurf_fi2
      fi2_alt(ix,3*iy,iz)=wurf_fi2

      rho_neu(ix,3*iy,iz)=rho_rx
      rho_alt(ix,3*iy,iz)=rho_rx

      write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu


c Keim Nr 4
      wurf=ran(iseed)
      wurf_fi1=wurf*eulermax
      wurf=ran(iseed)
      wurf_fi=wurf*eulermax
      wurf=ran(iseed)
      wurf_fi2=wurf*eulermax

      fi1_neu(3*ix,3*iy,iz)=wurf_fi1
      fi1_alt(3*ix,3*iy,iz)=wurf_fi1
      fi_neu(3*ix,3*iy,iz)=wurf_fi
      fi_alt(3*ix,3*iy,iz)=wurf_fi
      fi2_neu(3*ix,3*iy,iz)=wurf_fi2
      fi2_alt(3*ix,3*iy,iz)=wurf_fi2

      rho_neu(3*ix,3*iy,iz)=rho_rx
      rho_alt(3*ix,3*iy,iz)=rho_rx

      write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
     $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu

      end if
c Ende Vier Keime ***************************************************


c ###################################################################
c ##################  Keimbildungsfront mit Textur  #################
c ##################    verschiedene Komponenten    #################
c ##################    fuer Schaufelexperimente    #################
c ###################################################################
c Heterogene Keimbildungsfront fuer Wachstumsauslesetest ***********
      if(bildung.eq.7) then

      keimvol=100./(float(rand_miny)*float(rand_tatx))

      mit=0
      i_breite=30

      wurf=ran(iseed)
      wurf_fi1=wurf*eulermax
      wurf=ran(iseed)
      wurf_fi=wurf*eulermax
      wurf=ran(iseed)
      wurf_fi2=wurf*eulermax

      do 1254 iy=2,rand_miny
        imit=mod(mit,i_breite)
            if(imit.lt.1) then
            wurf=ran(iseed)
            wurf_fi1=wurf*eulermax
            wurf=ran(iseed)
```

```
            wurf_fi=wurf*eulermax
            wurf=ran(iseed)
            wurf_fi2=wurf*eulermax
            end if
          mit=mit+1

         do 1254 iz=2,ra3d_tat
               rho_neu(rand_tatx,iy,iz)=rho_rx
               rho_alt(rand_tatx,iy,iz)=rho_rx

         fi1_neu(rand_tatx,iy,iz)=wurf_fi1
         fi1_alt(rand_tatx,iy,iz)=wurf_fi1
         fi_neu(rand_tatx,iy,iz)=wurf_fi
         fi_alt(rand_tatx,iy,iz)=wurf_fi
         fi2_neu(rand_tatx,iy,iz)=wurf_fi2
         fi2_alt(rand_tatx,iy,iz)=wurf_fi2

         write(uni_keim_ori,fmt=919) fi1_neu(ix,iy,iz),
        $fi_neu(ix,iy,iz),fi2_neu(ix,iy,iz),keimvol,keimstreu

1254   continue

       end if
c Ende Keimbildungsfront ********************************************

c ------ ENDE DER STATISCHEN ORTSGESAETTIGTEN KEIMBILDUNG -------------
       write(*,*) 'Keimbildungsphase abgeschlossen'

       close(uni_keim_ori)



c Oeffnen der Ausgabefiles mit dem Wachstumsgefuege zu verschiedenen Zeiten
c im Verlauf der RX


C Binaer     - ORIENTIERUNG
       open(unit=uni_au(1),recl=4,
      $file='c:\for\rx_konti2d\ori01.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(2),recl=4,
      $file='c:\for\rx_konti2d\ori02.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(3),recl=4,
      $file='c:\for\rx_konti2d\ori03.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(4),recl=4,
      $file='c:\for\rx_konti2d\ori04.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(5),recl=4,
      $file='c:\for\rx_konti2d\ori05.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(6),recl=4,
      $file='c:\for\rx_konti2d\ori06.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(7),recl=4,
      $file='c:\for\rx_konti2d\ori07.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(8),recl=4,
      $file='c:\for\rx_konti2d\ori08.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(9),recl=4,
      $file='c:\for\rx_konti2d\ori09.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(10),recl=4,
      $file='c:\for\rx_konti2d\ori10.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(11),recl=4,
      $file='c:\for\rx_konti2d\ori11.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(12),recl=4,
      $file='c:\for\rx_konti2d\ori12.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(13),recl=4,
      $file='c:\for\rx_konti2d\ori13.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(14),recl=4,
      $file='c:\for\rx_konti2d\ori14.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(15),recl=4,
      $file='c:\for\rx_konti2d\ori15.bin',
      $form='unformatted',status='unknown',access='direct')
```

```
       open(unit=uni_au(16),recl=4,
      $file='c:\for\rx_konti2d\ori16.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(17),recl=4,
      $file='c:\for\rx_konti2d\ori17.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(18),recl=4,
      $file='c:\for\rx_konti2d\ori18.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(19),recl=4,
      $file='c:\for\rx_konti2d\ori19.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_au(20),recl=4,
      $file='c:\for\rx_konti2d\ori20.bin',
      $form='unformatted',status='unknown',access='direct')


C Binaer     - RHO
       open(unit=uni_rho(1),recl=4,
      $file='c:\for\rx_konti2d\rho01.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(2),recl=4,
      $file='c:\for\rx_konti2d\rho02.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(3),recl=4,
      $file='c:\for\rx_konti2d\rho03.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(4),recl=4,
      $file='c:\for\rx_konti2d\rho04.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(5),recl=4,
      $file='c:\for\rx_konti2d\rho05.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(6),recl=4,
      $file='c:\for\rx_konti2d\rho06.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(7),recl=4,
      $file='c:\for\rx_konti2d\rho07.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(8),recl=4,
      $file='c:\for\rx_konti2d\rho08.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(9),recl=4,
      $file='c:\for\rx_konti2d\rho09.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(10),recl=4,
      $file='c:\for\rx_konti2d\rho10.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(11),recl=4,
      $file='c:\for\rx_konti2d\rho11.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(12),recl=4,
      $file='c:\for\rx_konti2d\rho12.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(13),recl=4,
      $file='c:\for\rx_konti2d\rho13.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(14),recl=4,
      $file='c:\for\rx_konti2d\rho14.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(15),recl=4,
      $file='c:\for\rx_konti2d\rho15.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(16),recl=4,
      $file='c:\for\rx_konti2d\rho16.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(17),recl=4,
      $file='c:\for\rx_konti2d\rho17.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(18),recl=4,
      $file='c:\for\rx_konti2d\rho18.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(19),recl=4,
      $file='c:\for\rx_konti2d\rho19.bin',
      $form='unformatted',status='unknown',access='direct')
       open(unit=uni_rho(20),recl=4,
      $file='c:\for\rx_konti2d\rho20.bin',
      $form='unformatted',status='unknown',access='direct')


C ######################################################################
```

```
C                                KEIMWACHSTUM
C ######################################################################

c Echt-Zeitschritt ermitteln ****************************************
      delta_t=1./freq0
      write(*,*) delta_t

      write(uni_au0,fmt=100)
      write(uni_au0,fmt=100) 'Geschwindigkeitskonstante der Simulation'
      write(uni_au0,fmt=128) delta_t,' Zeitschritt     [s]'
      write(uni_au0,fmt=100)

c Vorfaktor fuer die stetige, statische Erholung **********************
c Formeln aus Humphreys und Hatherly, Seite 140 ff
c phenomenologische Kinetik erster Ordnung
c d rho/ d t = - erhol * rho
      vor_er=6.*d0_self*exp(-q_self*dukbt)*sb*bb*
     $bb*dukbt/(lambda*lambda)

c Vorfaktor fuer die treibende Kraft der primaeren RX *****************
      faktor=sbb/(4.*pi*(1.-querkon))

      write(uni_au0,fmt=198) faktor,' Vorfaktor Vers.energie'
      write(uni_au0,fmt=100)

c Wahrscheinlichkeits-Vorfaktor berechnen ****************************
      vor_wahr=1./(freq0*lambda)
      write(uni_au0,fmt=198) vor_wahr,' Vorfaktor Wahrscheinlichkeit'
      write(uni_au0,fmt=100)


c Gesamt-Vorfaktor berechnen *****************************
      vor_fak=vor_wahr*faktor

c Faktor fuer allgem. KG
      w_allg=mob0*exp(-wach_erg*dukbt)
      write(uni_au0,fmt=179) w_allg/mob0,' NUR Boltzm.fakt.'
      write(uni_au0,fmt=179) w_allg,' mo * Boltzm.fakt. fuer allgem. KG'

c Faktor fuer allgem. KG
      log_fak=sqrt(rho_max)*burgers
      deltarho=rho_max
      wahrsch=w_allg*vor_fak*deltarho*(0.5-log(log_fak))
      write(uni_au0,fmt=179) wahrsch,' Vor-Faktor mit Vers.erg. all. KG'
      write(uni_au0,fmt=100)

c Kinetik- Ausgabefiles oeffnen ***************************************
      open(unit=uni_kin1,
     $file='c:\for\rx_konti2d\rx_konti_kin1.txt')
      write(uni_kin1,fmt=100) '  Zeit[Sek]       X(t)[Proz]'

c Cahn-Hagel-Analyse - Ausgabefile oeffnen und vol_inv berechnen *********

c def. cahn-hagel parameter nach vandermeer (1989)
c gesamte grenzflaeche zwischen RX und nicht-RX pro gesamtvolumen
c normiert auf volumen
      open(unit=uni_ch,
     $file='c:\for\rx_konti2d\rx_konti_ch.txt')
      write(uni_ch,fmt=100) '  X[percent]         S(X)[1/cellsize]'
c auf lambda normiert - spart rechenzeit
c fuer 3D
      x_vol=float(rand_tatx-1)*float(rand_taty-1)*float(ra3d_tat-1)
c 0.5, da sonst grenzflaeche doppelt gezaehlt
c *lambda  - weil das lambda-quadrat sich herauskuerzt
      vol_inv=0.5/x_vol

c Haertekurve - Ausgabefile oeffnen und Vorfaktor berechnen ************
c      open(unit=uni_hv,
c     $file='c:\for\rx_konti2d\rx_konti_hv.txt')
c      write(uni_hv,fmt=100) '# t[s]      Tau[MPa]     rho [10**13/m**2]'
c     tau_vor=0.5*sb/mega

c aenderung um die .le. durch eine .lt. abfrage zu ersetzen
      rho_situ=rho_situ+1

c ######################################################################
c        Haupt - Zeitschleife fuer Keimwachstum bis Zusammenstoss
c ######################################################################
      do 310 ii=1,laufzeit
      vol_rx=0.
      frac_alt= frac_rx
```

```
c ########################################################################
c               Ortsschleife - Keimwachstum - Nachbarn
c ########################################################################
      do 311 iz=2,ra3d tat
      do 311 ix=2,rand tatx
      do 311 iy=2,rand taty
c Erholung ************************************************************
c Alle Zellen koennen von Erholung betroffen sein
c Vorsicht: Die Dichten sind in Einheiten von 10^13 pro qm gerechnet

C ERHOLUNGSABFRAGE RAUS ZUR PROGRAMMBESCHLEUNIGUNG
c      if(i erhol.gt.0) then
c      rho alt(ix,iy,iz)=rho alt(ix,iy,iz)-
c     $ifix(delta t*vor er*float(rho alt(ix,iy,iz))*
c     $hoch13*float(rho alt(ix,iy,iz)))
c      if(rho alt(ix,iy,iz).lt.rho rx) rho_alt(ix,iy,iz)=rho_rx
c      rho_neu(ix,iy,iz)=rho_alt(ix,iy,iz)
c
c
c
c
Cc      xxx=delta t*vor er*float(rho alt(ix,iy,iz))*
Cc     $float(rho_alt(ix,iy,iz))*hoch13+0.5
Cc
Cc    write(*,*) rho alt(ix,iy,iz),xxx,ii,
Cc    $ifix(delta t*vor er*float(rho alt(ix,iy,iz))*
Cc     $hoch13*float(rho_alt(ix,iy,iz))+.5)
c     end if


      do 312 i=1,nachbar
      geschw(i)=x_null
312   continue

c Euler-Orientierung der Komponente bei ix,iy,iz
      fi1(1)=fi1 alt(ix,iy,iz)
      fi(1)=fi_alt(ix,iy,iz)
      fi2(1)=fi2_alt(ix,iy,iz)

c 3D Wachstum
c absuchen der noch nicht RX zellen
c symmetrisierte Ratengleichung



c betrachtete-Zelle  ##########################################
      if(rho_alt(ix,iy,iz).gt.rho_situ) then


c Nachbar Nr.1 ################################################

          if(rho_alt(ix-1,iy,iz).lt.rho_situ) then
          geschw(1)=x null
          w_spez=w_allg

      if(fi1 alt(ix,iy,iz).ne.fi1 alt(ix-1,iy,iz).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix-1,iy,iz).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix-1,iy,iz)) then

      fi1(2)=fi1 alt(ix-1,iy,iz)
      fi(2)=fi_alt(ix-1,iy,iz)
      fi2(2)=fi2_alt(ix-1,iy,iz)


          call misori(fi1,fi,fi2,winkel,iachs)
c Nur Kleinwinkelkorngrenze
          if(winkel.lt.wink ab) goto 701
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8201 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8201  continue
      if(ikon.lt.1) goto 7001

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8001 ik=1,i_spez
```

```
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
c es handelt sich um eine spezielle KG
          w_spez=mob_exp(ik)
          goto 7001
          end if
          end if
8001      continue
7001      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
           deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix-1,iy,iz))*hoch13
           log_fak=sqrt(deltarho)*burgers
           wahrsch=w_spez*vor_fak*deltarho*(0.5-log(log_fak))

c          write(*,*) 'wahrsch',wahrsch
c          wahrsch2=w_spez*deltarho*(0.5-log(log_fak))*vor
c          write(*,*) 'wahrsch2',wahrsch2
c          pause

           wurf= ran(iseed)
c Die groesstmoegliche wahrscheinlichkeit ist fuer die lokale Situation
c gleich dem Wert  wahrsch, d.h. geschw(1)_max =  wahrsch
c          geschw(1)=gesch_min-wurf

c             if(wurf.gt.wahrsch) goto 701
c Merke:  geschw(1) kann negative werden - spart eine if - Zeile
           geschw(1)=wahrsch-wurf
           end if
701        continue
           end if


c Nachbar Nr.2 ###################################################
           if(rho_alt(ix+1,iy,iz).lt.rho_situ) then
           geschw(2)=x_null
           w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix+1,iy,iz).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix+1,iy,iz).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix+1,iy,iz)) then

       fi1(2)=fi1_alt(ix+1,iy,iz)
       fi(2)=fi_alt(ix+1,iy,iz)
       fi2(2)=fi2_alt(ix+1,iy,iz)

           call misori(fi1,fi,fi2,winkel,iachs)
           if(winkel.lt.wink_ab) goto 702
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
       ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
       ikon=0
       do 8202 ik=1,i_spez
       if(ivek.eq.vek(ik)) ikon=ikon+1
8202   continue
       if(ikon.lt.1) goto 7002

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
           ia1=abs(iachs(1))
           ia2=abs(iachs(2))
           ia3=abs(iachs(3))
       do 8002 ik=1,i_spez
       i_ja=0
       if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
       if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
       if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
```

```
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i ja=1
          if(i ja.gt.0) then
          if(winkel.ge.alpu s(ik).and.winkel.le.alpo_s(ik)) then
          w spez=mob_exp(ik)
          goto 7002
          end if
          end if
8002      continue
7002      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          deltarho=float(rho alt(ix,iy,iz)-rho_alt(ix+1,iy,iz))*hoch13
          log_fak=sqrt(deltarho)*burgers
          wahrsch=w_spez*vor_fak*deltarho*(0.5-log(log_fak))

c         write(*,*) 'wahrsch',wahrsch
c         wahrsch2=w spez*deltarho*(0.5-log(log_fak))*vor
c         write(*,*) 'wahrsch2',wahrsch2
c         pause

          wurf= ran(iseed)

c             if(wurf.gt.wahrsch) goto 702
c         geschw(2)=gesch min-wurf
          geschw(2)=wahrsch-wurf
          end if
702       continue
          end if


c Nachbar Nr.3 ################################################

          if(rho_alt(ix,iy-1,iz).lt.rho_situ) then
          geschw(3)=x null
          w_spez=w_allg


      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix,iy-1,iz).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix,iy-1,iz).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix,iy-1,iz)) then

       fi1(2)=fi1_alt(ix,iy-1,iz)
       fi(2)=fi_alt(ix,iy-1,iz)
       fi2(2)=fi2_alt(ix,iy-1,iz)

          call misori(fi1,fi,fi2,winkel,iachs)
          if(winkel.lt.wink_ab) goto 703
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
       ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
       ikon=0
       do 8203 ik=1,i_spez
       if(ivek.eq.vek(ik)) ikon=ikon+1
8203   continue
       if(ikon.lt.1) goto 7003

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
       do 8003 ik=1,i_spez
       i ja=0
       if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
      $i ja=1
       if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
      $i ja=1
       if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
      $i ja=1
       if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
      $i ja=1
       if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
      $i ja=1
       if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
      $i ja=1
          if(i ja.gt.0) then
          if(winkel.ge.alpu s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
```

```
         goto 7003
         end if
         end if
8003     continue
7003     continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix,iy-1,iz))*hoch13
          log_fak=sqrt(deltarho)*burgers
          wahrsch=w_spez*vor_fak*deltarho*(0.5-log(log_fak))

c         write(*,*) 'wahrsch',wahrsch
c         wahrsch2=w_spez*deltarho*(0.5-log(log_fak))*vor
c         write(*,*) 'wahrsch2',wahrsch2
c         pause

          wurf= ran(iseed)

c             if(wurf.gt.wahrsch) goto 703
c         geschw(3)=gesch min-wurf
          geschw(3)=wahrsch-wurf
          end if
703       continue
          end if



c Nachbar Nr.4 ##################################################

          if(rho_alt(ix,iy+1,iz).lt.rho_situ) then
          geschw(4)=x_null
          w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix,iy+1,iz).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix,iy+1,iz).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix,iy+1,iz)) then

      fi1(2)=fi1_alt(ix,iy+1,iz)
      fi(2)=fi_alt(ix,iy+1,iz)
      fi2(2)=fi2_alt(ix,iy+1,iz)

          call misori(fi1,fi,fi2,winkel,iachs)
          if(winkel.lt.wink_ab) goto 704
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8204 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8204  continue
      if(ikon.lt.1) goto 7004

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8004 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7004
          end if
          end if
8004      continue
7004      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix,iy+1,iz))*hoch13
          log_fak=sqrt(deltarho)*burgers
          wahrsch=w_spez*vor_fak*deltarho*(0.5-log(log_fak))
```

```
c         write(*,*) 'wahrsch',wahrsch
c         wahrsch2=w_spez*deltarho*(0.5-log(log_fak))*vor
c         write(*,*) 'wahrsch2',wahrsch2
c         pause

          wurf= ran(iseed)

c             if(wurf.gt.wahrsch) goto 704
c         geschw(4)=gesch min-wurf
          geschw(4)=wahrsch-wurf
          end if
704       continue
          end if


c Nachbar Nr.5 ################################################

          if(rho alt(ix,iy,iz-1).lt.rho_situ) then
          geschw(5)=x null
          w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix,iy,iz-1).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix,iy,iz-1).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix,iy,iz-1)) then

      fi1(2)=fi1_alt(ix,iy,iz-1)
      fi(2)=fi_alt(ix,iy,iz-1)
      fi2(2)=fi2_alt(ix,iy,iz-1)

          call misori(fi1,fi,fi2,winkel,iachs)
          if(winkel.lt.wink_ab) goto 705
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8205 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8205  continue
      if(ikon.lt.1) goto 7005

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8005 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7005
          end if
          end if
8005      continue
7005      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix,iy,iz-1))*hoch13
          log_fak=sqrt(deltarho)*burgers
          wahrsch=w_spez*vor_fak*deltarho*(0.5-log(log_fak))


c         write(*,*) 'wahrsch',wahrsch
c         wahrsch2=w_spez*deltarho*(0.5-log(log_fak))*vor
c         write(*,*) 'wahrsch2',wahrsch2
c         pause

          wurf= ran(iseed)

c             if(wurf.gt.wahrsch) goto 705
```

```
c         geschw(5)=gesch min-wurf
          geschw(5)=wahrsch-wurf
          end if
705       continue
          end if



c Nachbar Nr.6 ################################################

          if(rho_alt(ix,iy,iz+1).lt.rho_situ) then
          geschw(6)=x_null
          w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix,iy,iz+1).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix,iy,iz+1).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix,iy,iz+1)) then

      fi1(2)=fi1_alt(ix,iy,iz+1)
      fi(2)=fi_alt(ix,iy,iz+1)
      fi2(2)=fi2_alt(ix,iy,iz+1)

          call misori(fi1,fi,fi2,winkel,iachs)
          if(winkel.lt.wink_ab) goto 706
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8206 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8206  continue
      if(ikon.lt.1) goto 7006

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8006 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7006
          end if
          end if
8006      continue
7006      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix,iy,iz+1))*hoch13
          log_fak=sqrt(deltarho)*burgers
          wahrsch=w_spez*vor_fak*deltarho*(0.5-log(log_fak))

c         write(*,*) 'wahrsch',wahrsch
c         wahrsch2=w_spez*deltarho*(0.5-log(log_fak))*vor
c         write(*,*) 'wahrsch2',wahrsch2
c         pause

          wurf= ran(iseed)

c             if(wurf.gt.wahrsch) goto 706
c         geschw(6)=gesch min-wurf
          geschw(6)=wahrsch-wurf
          end if
706       continue
          end if


c Diagonalbeitraege ************************************************
```

```
c Nachbar Nr.7 ################################################

          if(rho_alt(ix+1,iy+1,iz).lt.rho_situ) then
          geschw(7)=x_null
          w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix+1,iy+1,iz).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix+1,iy+1,iz).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix+1,iy+1,iz)) then

      fi1(2)=fi1_alt(ix+1,iy+1,iz)
      fi(2)=fi_alt(ix+1,iy+1,iz)
      fi2(2)=fi2_alt(ix+1,iy+1,iz)

          call misori(fi1,fi,fi2,winkel,iachs)
          if(winkel.lt.wink_ab) goto 707
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8207 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8207  continue
      if(ikon.lt.1) goto 7007

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8007 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7007
          end if
          end if
8007      continue
7007      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix+1,iy+1,iz))
     $*hoch13
          log_fak=sqrt(deltarho)*burgers
       wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
          wurf= ran(iseed)

c             if(wurf.gt.wahrsch) goto 707
c         geschw(7)=gesch_min-wurf
          geschw(7)=wahrsch-wurf
          end if
707       continue
          end if


c Nachbar Nr.8 ################################################

            if(rho_alt(ix+1,iy-1,iz).lt.rho_situ) then
            geschw(8)=x_null
            w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix+1,iy-1,iz).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix+1,iy-1,iz).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix+1,iy-1,iz)) then

      fi1(2)=fi1_alt(ix+1,iy-1,iz)
      fi(2)=fi_alt(ix+1,iy-1,iz)
      fi2(2)=fi2_alt(ix+1,iy-1,iz)
```

```
            call misori(fi1,fi,fi2,winkel,iachs)
            if(winkel.lt.wink_ab) goto 708
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8208 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8208  continue
      if(ikon.lt.1) goto 7008

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8008 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7008
          end if
          end if
8008      continue
7008      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix+1,iy-1,iz))
     $*hoch13
            log_fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
            wurf= ran(iseed)

c               if(wurf.gt.wahrsch) goto 708
c         geschw(8)=gesch_min-wurf
          geschw(8)=wahrsch-wurf
            end if
708       continue
            end if

c Nachbar Nr.9 ##################################################

            if(rho_alt(ix-1,iy+1,iz).lt.rho_situ) then
            geschw(9)=x_null
            w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix-1,iy+1,iz).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix-1,iy+1,iz).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix-1,iy+1,iz)) then

      fi1(2)=fi1_alt(ix-1,iy+1,iz)
      fi(2)=fi_alt(ix-1,iy+1,iz)
      fi2(2)=fi2_alt(ix-1,iy+1,iz)

            call misori(fi1,fi,fi2,winkel,iachs)
            if(winkel.lt.wink_ab) goto 709
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8209 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8209  continue
      if(ikon.lt.1) goto 7009

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
```

```
      do 8009 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7009
          end if
          end if
8009      continue
7009      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix-1,iy+1,iz))
     $*hoch13
            log_fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
            wurf= ran(iseed)

c               if(wurf.gt.wahrsch) goto 709
c         geschw(9)=gesch_min-wurf
          geschw(9)=wahrsch-wurf
            end if
709       continue
            end if

c Nachbar Nr.10 ###################################################

            if(rho_alt(ix-1,iy-1,iz).lt.rho_situ) then
            geschw(10)=x_null
            w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix-1,iy-1,iz).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix-1,iy-1,iz).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix-1,iy-1,iz)) then

      fi1(2)=fi1_alt(ix-1,iy-1,iz)
      fi(2)=fi_alt(ix-1,iy-1,iz)
      fi2(2)=fi2_alt(ix-1,iy-1,iz)

            call misori(fi1,fi,fi2,winkel,iachs)
            if(winkel.lt.wink_ab) goto 710
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8210 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8210  continue
      if(ikon.lt.1) goto 7010

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8010 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
```

```
          w spez=mob_exp(ik)
          goto 7010
          end if
          end if
8010      continue
7010      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix-1,iy-1,iz))
     $*hoch13
            log fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
            wurf= ran(iseed)

c               if(wurf.gt.wahrsch) goto 710
c         geschw(10)=gesch min-wurf
          geschw(10)=wahrsch-wurf
            end if
710       continue
            end if

c Nachbar Nr.11 ##################################################

          if(rho_alt(ix+1,iy,iz+1).lt.rho_situ) then
          geschw(11)=x null
          w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix+1,iy,iz+1).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix+1,iy,iz+1).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix+1,iy,iz+1)) then

      fi1(2)=fi1_alt(ix+1,iy,iz+1)
      fi(2)=fi_alt(ix+1,iy,iz+1)
      fi2(2)=fi2_alt(ix+1,iy,iz+1)

          call misori(fi1,fi,fi2,winkel,iachs)
          if(winkel.lt.wink_ab) goto 711
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8211 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8211  continue
      if(ikon.lt.1) goto 7011

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8011 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7011
          end if
          end if
8011      continue
7011      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix+1,iy,iz+1))
     $*hoch13
            log_fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
          wurf= ran(iseed)

c             if(wurf.gt.wahrsch) goto 711
c         geschw(11)=gesch min-wurf
          geschw(11)=wahrsch-wurf
```

```
          end if
711       continue
          end if


c Nachbar Nr.12 ################################################

            if(rho_alt(ix+1,iy,iz-1).lt.rho_situ) then
            geschw(12)=x_null
            w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix+1,iy,iz-1).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix+1,iy,iz-1).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix+1,iy,iz-1)) then

      fi1(2)=fi1_alt(ix+1,iy,iz-1)
      fi(2)=fi_alt(ix+1,iy,iz-1)
      fi2(2)=fi2_alt(ix+1,iy,iz-1)

            call misori(fi1,fi,fi2,winkel,iachs)
            if(winkel.lt.wink_ab) goto 712
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8212 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8212  continue
      if(ikon.lt.1) goto 7012

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8012 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7012
          end if
          end if
8012      continue
7012      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix+1,iy,iz-1))
     $*hoch13
            log_fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
            wurf= ran(iseed)

c               if(wurf.gt.wahrsch) goto 712
c         geschw(12)=gesch_min-wurf
          geschw(12)=wahrsch-wurf
            end if
712       continue
            end if

c Nachbar Nr.13 ################################################

            if(rho_alt(ix-1,iy,iz+1).lt.rho_situ) then
            geschw(13)=x_null
            w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix-1,iy,iz+1).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix-1,iy,iz+1).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix-1,iy,iz+1)) then

      fi1(2)=fi1_alt(ix-1,iy,iz+1)
```

```
      fi(2)=fi_alt(ix-1,iy,iz+1)
      fi2(2)=fi2_alt(ix-1,iy,iz+1)

            call misori(fi1,fi,fi2,winkel,iachs)
            if(winkel.lt.wink_ab) goto 713
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8213 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8213  continue
      if(ikon.lt.1) goto 7013

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8013 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
         if(i_ja.gt.0) then
         if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
         w_spez=mob_exp(ik)
         goto 7013
         end if
         end if
8013     continue
7013     continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix-1,iy,iz+1))
     $*hoch13
            log_fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
            wurf= ran(iseed)

c               if(wurf.gt.wahrsch) goto 713
c         geschw(13)=gesch_min-wurf
          geschw(13)=wahrsch-wurf
            end if
713       continue
            end if

c Nachbar Nr.14 ################################################

            if(rho_alt(ix-1,iy,iz-1).lt.rho_situ) then
            geschw(14)=x_null
            w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix-1,iy,iz-1).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix-1,iy,iz-1).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix-1,iy,iz-1)) then

      fi1(2)=fi1_alt(ix-1,iy,iz-1)
      fi(2)=fi_alt(ix-1,iy,iz-1)
      fi2(2)=fi2_alt(ix-1,iy,iz-1)

            call misori(fi1,fi,fi2,winkel,iachs)
            if(winkel.lt.wink_ab) goto 714
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8214 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8214  continue
      if(ikon.lt.1) goto 7014

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
```

```
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
       do 8014 ik=1,i_spez
       i_ja=0
       if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
      $i_ja=1
       if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
      $i_ja=1
       if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
      $i_ja=1
       if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
      $i_ja=1
       if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
      $i_ja=1
       if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
      $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7014
          end if
          end if
8014      continue
7014      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix-1,iy,iz-1))
      $*hoch13
            log_fak=sqrt(deltarho)*burgers
       wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
            wurf= ran(iseed)

c               if(wurf.gt.wahrsch) goto 714
c         geschw(14)=gesch_min-wurf
          geschw(14)=wahrsch-wurf
            end if
714       continue
            end if

c Nachbar Nr.15 ##################################################

          if(rho_alt(ix,iy+1,iz+1).lt.rho_situ) then
          geschw(15)=x_null
          w_spez=w_allg

       if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix,iy+1,iz+1).or.
      $fi_alt(ix,iy,iz).ne.fi_alt(ix,iy+1,iz+1).or.
      $fi2_alt(ix,iy,iz).ne.fi2_alt(ix,iy+1,iz+1)) then

       fi1(2)=fi1_alt(ix,iy+1,iz+1)
       fi(2)=fi_alt(ix,iy+1,iz+1)
       fi2(2)=fi2_alt(ix,iy+1,iz+1)

          call misori(fi1,fi,fi2,winkel,iachs)
          if(winkel.lt.wink_ab) goto 715
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
       ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
       ikon=0
       do 8215 ik=1,i_spez
       if(ivek.eq.vek(ik)) ikon=ikon+1
8215   continue
       if(ikon.lt.1) goto 7015

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
       do 8015 ik=1,i_spez
       i_ja=0
       if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
      $i_ja=1
       if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
      $i_ja=1
       if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
      $i_ja=1
       if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
      $i_ja=1
       if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
      $i_ja=1
       if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
```

```
     $i ja=1
         if(i ja.gt.0) then
         if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
         w_spez=mob_exp(ik)
         goto 7015
         end if
         end if
8015     continue
7015     continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
          deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix,iy+1,iz+1))
     $*hoch13
          log_fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
          wurf= ran(iseed)

c             if(wurf.gt.wahrsch) goto 715
c         geschw(15)=gesch_min-wurf
          geschw(15)=wahrsch-wurf
          end if
715       continue
          end if

c Nachbar Nr.16 ################################################

            if(rho_alt(ix,iy+1,iz-1).lt.rho_situ) then
            geschw(16)=x_null
            w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix,iy+1,iz-1).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix,iy+1,iz-1).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix,iy+1,iz-1)) then

      fi1(2)=fi1_alt(ix,iy+1,iz-1)
      fi(2)=fi_alt(ix,iy+1,iz-1)
      fi2(2)=fi2_alt(ix,iy+1,iz-1)

            call misori(fi1,fi,fi2,winkel,iachs)
            if(winkel.lt.wink_ab) goto 716
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8216 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8216  continue
      if(ikon.lt.1) goto 7016

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8016 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
         if(i_ja.gt.0) then
         if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
         w_spez=mob_exp(ik)
         goto 7016
         end if
         end if
8016     continue
7016     continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix,iy+1,iz-1))
     $*hoch13
            log_fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
            wurf= ran(iseed)
```

```
c               if(wurf.gt.wahrsch) goto 716
c         geschw(16)=gesch min-wurf
          geschw(16)=wahrsch-wurf
            end if
716       continue
            end if

c Nachbar Nr.17 ################################################

            if(rho alt(ix,iy-1,iz+1).lt.rho_situ) then
            geschw(17)=x null
            w_spez=w_allg

      if(fi1 alt(ix,iy,iz).ne.fi1 alt(ix,iy-1,iz+1).or.
     $fi alt(ix,iy,iz).ne.fi alt(ix,iy-1,iz+1).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix,iy-1,iz+1)) then

      fi1(2)=fi1 alt(ix,iy-1,iz+1)
      fi(2)=fi alt(ix,iy-1,iz+1)
      fi2(2)=fi2_alt(ix,iy-1,iz+1)

            call misori(fi1,fi,fi2,winkel,iachs)
            if(winkel.lt.wink_ab) goto 717
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8217 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8217  continue
      if(ikon.lt.1) goto 7017

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8017 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7017
          end if
          end if
8017      continue
7017      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix,iy-1,iz+1))
     $*hoch13
            log_fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
            wurf= ran(iseed)

c               if(wurf.gt.wahrsch) goto 717
c         geschw(17)=gesch_min-wurf
          geschw(17)=wahrsch-wurf
            end if
717       continue
            end if

c Nachbar Nr.18 ################################################

            if(rho_alt(ix,iy-1,iz-1).lt.rho_situ) then
            geschw(18)=x_null
            w_spez=w_allg

      if(fi1_alt(ix,iy,iz).ne.fi1_alt(ix,iy-1,iz-1).or.
     $fi_alt(ix,iy,iz).ne.fi_alt(ix,iy-1,iz-1).or.
     $fi2_alt(ix,iy,iz).ne.fi2_alt(ix,iy-1,iz-1)) then
```

```
      fi1(2)=fi1_alt(ix,iy-1,iz-1)
      fi(2)=fi_alt(ix,iy-1,iz-1)
      fi2(2)=fi2_alt(ix,iy-1,iz-1)

            call misori(fi1,fi,fi2,winkel,iachs)
            if(winkel.lt.wink_ab) goto 718
c Abfrage auf spezielle Korngrenzen - - - - - - - - - - - - - -

c Grobabfrage
      ivek=iachs(1)*iachs(1)+iachs(2)*iachs(2)+iachs(3)*iachs(3)
      ikon=0
      do 8218 ik=1,i_spez
      if(ivek.eq.vek(ik)) ikon=ikon+1
8218  continue
      if(ikon.lt.1) goto 7018

c Detailabfrage wenn Vektorwinkelbetrag uebereinstimmt
          ia1=abs(iachs(1))
          ia2=abs(iachs(2))
          ia3=abs(iachs(3))
      do 8018 ik=1,i_spez
      i_ja=0
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,1).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,3).and.ia3.eq.a_s(ik,1))
     $i_ja=1
      if(ia1.eq.a_s(ik,2).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,3))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,1).and.ia3.eq.a_s(ik,2))
     $i_ja=1
      if(ia1.eq.a_s(ik,3).and.ia2.eq.a_s(ik,2).and.ia3.eq.a_s(ik,1))
     $i_ja=1
          if(i_ja.gt.0) then
          if(winkel.ge.alpu_s(ik).and.winkel.le.alpo_s(ik)) then
          w_spez=mob_exp(ik)
          goto 7018
          end if
          end if
8018      continue
7018      continue
c - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
            deltarho=float(rho_alt(ix,iy,iz)-rho_alt(ix,iy-1,iz-1))
     $*hoch13
            log_fak=sqrt(deltarho)*burgers
      wahrsch=fak_2d*w_spez*vor_fak*deltarho*(0.5-log(log_fak))
            wurf= ran(iseed)

c               if(wurf.gt.wahrsch) goto 718
c         geschw(18)=gesch_min-wurf
          geschw(18)=wahrsch-wurf
            end if
718       continue
            end if


c ######################################################################
c #########  die schnellste der umgebungsgrenzflaechen ermitteln #####
c ######################################################################
      ge_max=x_null
      do 433 i=1,nachbar
      if(geschw(i).gt.ge_max) ge_max=geschw(i)
433   continue

      if(ge_max.gt.x_null) then
          if(geschw(1).ge.ge_max) then
          rho_neu(ix,iy,iz)=x_null
          fi1_neu(ix,iy,iz)=fi1_alt(ix-1,iy,iz)
          fi_neu(ix,iy,iz)=fi_alt(ix-1,iy,iz)
          fi2_neu(ix,iy,iz)=fi2_alt(ix-1,iy,iz)
          end if
          if(geschw(2).ge.ge_max) then
          rho_neu(ix,iy,iz)=x_null
          fi1_neu(ix,iy,iz)=fi1_alt(ix+1,iy,iz)
          fi_neu(ix,iy,iz)=fi_alt(ix+1,iy,iz)
          fi2_neu(ix,iy,iz)=fi2_alt(ix+1,iy,iz)
          end if
          if(geschw(3).ge.ge_max) then
          rho_neu(ix,iy,iz)=x_null
          fi1_neu(ix,iy,iz)=fi1_alt(ix,iy-1,iz)
```

```
fi_neu(ix,iy,iz)=fi_alt(ix,iy-1,iz)
fi2_neu(ix,iy,iz)=fi2_alt(ix,iy-1,iz)
end if
if(geschw(4).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix,iy+1,iz)
fi_neu(ix,iy,iz)=fi_alt(ix,iy+1,iz)
fi2_neu(ix,iy,iz)=fi2_alt(ix,iy+1,iz)
end if
if(geschw(5).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix,iy,iz-1)
fi_neu(ix,iy,iz)=fi_alt(ix,iy,iz-1)
fi2_neu(ix,iy,iz)=fi2_alt(ix,iy,iz-1)
end if
if(geschw(6).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix,iy,iz+1)
fi_neu(ix,iy,iz)=fi_alt(ix,iy,iz+1)
fi2_neu(ix,iy,iz)=fi2_alt(ix,iy,iz+1)
end if
if(geschw(7).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix+1,iy+1,iz)
fi_neu(ix,iy,iz)=fi_alt(ix+1,iy+1,iz)
fi2_neu(ix,iy,iz)=fi2_alt(ix+1,iy+1,iz)
end if
if(geschw(8).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix+1,iy-1,iz)
fi_neu(ix,iy,iz)=fi_alt(ix+1,iy-1,iz)
fi2_neu(ix,iy,iz)=fi2_alt(ix+1,iy-1,iz)
end if
if(geschw(9).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix-1,iy+1,iz)
fi_neu(ix,iy,iz)=fi_alt(ix-1,iy+1,iz)
fi2_neu(ix,iy,iz)=fi2_alt(ix-1,iy+1,iz)
end if
if(geschw(10).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix-1,iy-1,iz)
fi_neu(ix,iy,iz)=fi_alt(ix-1,iy-1,iz)
fi2_neu(ix,iy,iz)=fi2_alt(ix-1,iy-1,iz)
end if
if(geschw(11).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix+1,iy,iz+1)
fi_neu(ix,iy,iz)=fi_alt(ix+1,iy,iz+1)
fi2_neu(ix,iy,iz)=fi2_alt(ix+1,iy,iz+1)
end if
if(geschw(12).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix+1,iy,iz-1)
fi_neu(ix,iy,iz)=fi_alt(ix+1,iy,iz-1)
fi2_neu(ix,iy,iz)=fi2_alt(ix+1,iy,iz-1)
end if
if(geschw(13).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix-1,iy,iz+1)
fi_neu(ix,iy,iz)=fi_alt(ix-1,iy,iz+1)
fi2_neu(ix,iy,iz)=fi2_alt(ix-1,iy,iz+1)
end if
if(geschw(14).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix-1,iy,iz-1)
fi_neu(ix,iy,iz)=fi_alt(ix-1,iy,iz-1)
fi2_neu(ix,iy,iz)=fi2_alt(ix-1,iy,iz-1)
end if
if(geschw(15).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix,iy+1,iz+1)
fi_neu(ix,iy,iz)=fi_alt(ix,iy+1,iz+1)
fi2_neu(ix,iy,iz)=fi2_alt(ix,iy+1,iz+1)
end if
if(geschw(16).ge.ge_max) then
rho_neu(ix,iy,iz)=x_null
fi1_neu(ix,iy,iz)=fi1_alt(ix,iy+1,iz-1)
fi_neu(ix,iy,iz)=fi_alt(ix,iy+1,iz-1)
fi2_neu(ix,iy,iz)=fi2_alt(ix,iy+1,iz-1)
end if
if(geschw(17).ge.ge_max) then
```

```
          rho neu(ix,iy,iz)=x null
          fi1 neu(ix,iy,iz)=fi1 alt(ix,iy-1,iz+1)
          fi neu(ix,iy,iz)=fi alt(ix,iy-1,iz+1)
          fi2 neu(ix,iy,iz)=fi2_alt(ix,iy-1,iz+1)
          end if
          if(geschw(18).ge.ge_max) then
          rho neu(ix,iy,iz)=x null
          fi1 neu(ix,iy,iz)=fi1 alt(ix,iy-1,iz-1)
          fi neu(ix,iy,iz)=fi alt(ix,iy-1,iz-1)
          fi2 neu(ix,iy,iz)=fi2_alt(ix,iy-1,iz-1)
          end if

      end if



c ######################################################################
c Auf Abfrage stetige, regellose Keimbildungswahrscheinlichkeit beruecksichtigen
c      if(bildung.eq.1) then
c         if(rho neu(ix,iy,iz).gt.rho situ) then
c         wahr st=delta t*exp(-keim_erg*dukbt)
c         wurf= ran(iseed)
c            if(wurf.lt.wahr_st) then
c
c      wurf=ran(iseed)
c      wurf fi1=wurf*eulermax
c      wurf=ran(iseed)
c      wurf fi=wurf*eulermax
c      wurf=ran(iseed)
c      wurf_fi2=wurf*eulermax
c
c
c           rho neu(ix,iy,iz)=rho rx
c     fi1 neu(rand tatx,iy,iz)=wurf fi1
c     fi1 alt(rand tatx,iy,iz)=wurf fi1
c     fi neu(rand tatx,iy,iz)=wurf fi
c     fi alt(rand tatx,iy,iz)=wurf fi
c     fi2 neu(rand tatx,iy,iz)=wurf fi2
c     fi2_alt(rand_tatx,iy,iz)=wurf_fi2
c
c           end if
c        end if
c     end if
c ---------------------------------------------------------------------




      end if


c Ende der Ortsschleife innerhalb der Zeitschleife
311   continue


c Gefuege neu auf alt umschreiben
c im zeitschritt - nicht im ortsschritt
      do 511 jz=2,ra3d tat
      do 511 jx=2,rand tatx
      do 511 jy=2,rand taty
      rho alt(jx,jy,jz)=rho neu(jx,jy,jz)
      fi1 alt(jx,jy,jz)=fi1 neu(jx,jy,jz)
      fi alt(jx,jy,jz)=fi neu(jx,jy,jz)
      fi2_alt(jx,jy,jz)=fi2_neu(jx,jy,jz)


c RX Bruchteil
      if(rho alt(jx,jy,jz).lt.rho_situ) then
      vol rx=vol rx+1.
      frac rx=100.*vol_rx/(float(i_randx*i_randy*i_3d))
      end if
511   continue

      if(zyklisch.gt.0) then
c Zyklische Randbedingung innnerhalb der Zeitschleife ******************
c Vorbelegung fuer zyklische Randbedingung *****************************
c Vorbelegung der Aussen- und Innenbereiche (Fortschreibung) ************
c Vorbelegung fuer x und y (Flaeche) ***********************************
      do 2377 k=1,ra3d aus
      do 2377 i=1,rand ausx
      fi1_alt(i,1,k)=fi1_alt(i,rand_taty,k)
```

```
      fi_alt(i,1,k)=fi_alt(i,rand_taty,k)
      fi2_alt(i,1,k)=fi2_alt(i,rand_taty,k)

      fi1_alt(i,rand_ausy,k)=fi1_alt(i,2,k)
      fi_alt(i,rand_ausy,k)=fi_alt(i,2,k)
      fi2_alt(i,rand_ausy,k)=fi2_alt(i,2,k)

      fi1_neu(i,1,k)=fi1_neu(i,rand_taty,k)
      fi_neu(i,1,k)=fi_neu(i,rand_taty,k)
      fi2_neu(i,1,k)=fi2_neu(i,rand_taty,k)

      fi1_neu(i,rand_ausy,k)=fi1_neu(i,2,k)
      fi_neu(i,rand_ausy,k)=fi_neu(i,2,k)
      fi2_neu(i,rand_ausy,k)=fi2_neu(i,2,k)

      rho_alt(i,1,k)=rho_alt(i,rand_taty,k)
      rho_alt(i,rand_ausy,k)=rho_alt(i,2,k)
      rho_neu(i,1,k)=rho_neu(i,rand_taty,k)
      rho_neu(i,rand_ausy,k)=rho_neu(i,2,k)
2377   continue

      do 2477 k=1,ra3d_aus
      do 2477 i=1,rand_ausy
      fi1_alt(1,i,k)=fi1_alt(rand_tatx,i,k)
      fi_alt(1,i,k)=fi_alt(rand_tatx,i,k)
      fi2_alt(1,i,k)=fi2_alt(rand_tatx,i,k)

      fi1_alt(rand_ausx,i,k)=fi1_alt(2,i,k)
      fi_alt(rand_ausx,i,k)=fi_alt(2,i,k)
      fi2_alt(rand_ausx,i,k)=fi2_alt(2,i,k)

      fi1_neu(1,i,k)=fi1_neu(rand_tatx,i,k)
      fi_neu(1,i,k)=fi_neu(rand_tatx,i,k)
      fi2_neu(1,i,k)=fi2_neu(rand_tatx,i,k)

      fi1_neu(rand_ausx,i,k)=fi1_neu(2,i,k)
      fi_neu(rand_ausx,i,k)=fi_neu(2,i,k)
      fi2_neu(rand_ausx,i,k)=fi2_neu(2,i,k)

      rho_alt(1,i,k)=rho_alt(rand_tatx,i,k)
      rho_alt(rand_ausx,i,k)=rho_alt(2,i,k)
      rho_neu(1,i,k)=rho_neu(rand_tatx,i,k)
      rho_neu(rand_ausx,i,k)=rho_neu(2,i,k)
2477   continue


      do 2577 i=1,rand_ausx
      do 2577 j=1,rand_ausy
      fi1_alt(i,j,1)=fi1_alt(i,j,ra3d_tat)
      fi_alt(i,j,1)=fi_alt(i,j,ra3d_tat)
      fi2_alt(i,j,1)=fi2_alt(i,j,ra3d_tat)

      fi1_alt(i,j,ra3d_aus)=fi1_alt(i,j,2)
      fi_alt(i,j,ra3d_aus)=fi_alt(i,j,2)
      fi2_alt(i,j,ra3d_aus)=fi2_alt(i,j,2)

      fi1_neu(i,j,1)=fi1_neu(i,j,ra3d_tat)
      fi_neu(i,j,1)=fi_neu(i,j,ra3d_tat)
      fi2_neu(i,j,1)=fi2_neu(i,j,ra3d_tat)

      fi1_neu(i,j,ra3d_aus)=fi1_neu(i,j,2)
      fi_neu(i,j,ra3d_aus)=fi_neu(i,j,2)
      fi2_neu(i,j,ra3d_aus)=fi2_neu(i,j,2)

      rho_alt(i,j,1)=rho_alt(i,j,ra3d_tat)
      rho_alt(i,j,ra3d_aus)=rho_alt(i,j,2)
      rho_neu(i,j,1)=rho_neu(i,j,ra3d_tat)
      rho_neu(i,j,ra3d_aus)=rho_neu(i,j,2)
2577   continue
      end if


      if(zyklisch.lt.1) then
c Statische Randbedingung innnerhalb der Zeitschleife ******************
c Vorbelegung fuer x und y (Flaeche) ***********************************
      do 5377 k=1,ra3d_aus
      do 5377 i=1,rand_ausx
      fi1_alt(i,1,k)=fi1_alt(i,2,k)
      fi_alt(i,1,k)=fi_alt(i,2,k)
      fi2_alt(i,1,k)=fi2_alt(i,2,k)

      fi1_alt(i,rand_ausy,k)=fi1_alt(i,rand_taty,k)
```

```
      fi_alt(i,rand_ausy,k)=fi_alt(i,rand_taty,k)
      fi2_alt(i,rand_ausy,k)=fi2_alt(i,rand_taty,k)

      fi1_neu(i,1,k)=fi1_neu(i,2,k)
      fi_neu(i,1,k)=fi_neu(i,2,k)
      fi2_neu(i,1,k)=fi2_neu(i,2,k)

      fi1_neu(i,rand_ausy,k)=fi1_neu(i,rand_taty,k)
      fi_neu(i,rand_ausy,k)=fi_neu(i,rand_taty,k)
      fi2_neu(i,rand_ausy,k)=fi2_neu(i,rand_taty,k)

      rho_alt(i,1,k)=rho_alt(i,2,k)
      rho_alt(i,rand_ausy,k)=rho_alt(i,2,k)
      rho_neu(i,1,k)=rho_neu(i,2,k)
      rho_neu(i,rand_ausy,k)=rho_neu(i,2,k)
5377   continue

      do 5477 k=1,ra3d_aus
      do 5477 i=1,rand_ausy
      fi1_alt(1,i,k)=fi1_alt(2,i,k)
      fi_alt(1,i,k)=fi_alt(2,i,k)
      fi2_alt(1,i,k)=fi2_alt(2,i,k)

      fi1_alt(rand_ausx,i,k)=fi1_alt(rand_tatx,i,k)
      fi_alt(rand_ausx,i,k)=fi_alt(rand_tatx,i,k)
      fi2_alt(rand_ausx,i,k)=fi2_alt(rand_tatx,i,k)

      fi1_neu(1,i,k)=fi1_neu(2,i,k)
      fi_neu(1,i,k)=fi_neu(2,i,k)
      fi2_neu(1,i,k)=fi2_neu(2,i,k)

      fi1_neu(rand_ausx,i,k)=fi1_neu(rand_tatx,i,k)
      fi_neu(rand_ausx,i,k)=fi_neu(rand_tatx,i,k)
      fi2_neu(rand_ausx,i,k)=fi2_neu(rand_tatx,i,k)

      rho_alt(1,i,k)=rho_alt(2,i,k)
      rho_alt(rand_ausx,i,k)=rho_alt(2,i,k)
      rho_neu(1,i,k)=rho_neu(2,i,k)
      rho_neu(rand_ausx,i,k)=rho_neu(2,i,k)
5477   continue


      do 5577 i=1,rand_ausx
      do 5577 j=1,rand_ausy
      fi1_alt(i,j,1)=fi1_alt(i,j,2)
      fi_alt(i,j,1)=fi_alt(i,j,2)
      fi2_alt(i,j,1)=fi2_alt(i,j,2)

      fi1_alt(i,j,ra3d_aus)=fi1_alt(i,j,ra3d_tat)
      fi_alt(i,j,ra3d_aus)=fi_alt(i,j,ra3d_tat)
      fi2_alt(i,j,ra3d_aus)=fi2_alt(i,j,ra3d_tat)

      fi1_neu(i,j,1)=fi1_neu(i,j,2)
      fi_neu(i,j,1)=fi_neu(i,j,2)
      fi2_neu(i,j,1)=fi2_neu(i,j,2)

      fi1_neu(i,j,ra3d_aus)=fi1_neu(i,j,ra3d_tat)
      fi_neu(i,j,ra3d_aus)=fi_neu(i,j,ra3d_tat)
      fi2_neu(i,j,ra3d_aus)=fi2_neu(i,j,ra3d_tat)

      rho_alt(i,j,1)=rho_alt(i,j,2)
      rho_alt(i,j,ra3d_aus)=rho_alt(i,j,ra3d_tat)
      rho_neu(i,j,1)=rho_neu(i,j,2)
      rho_neu(i,j,ra3d_aus)=rho_neu(i,j,ra3d_tat)
5577   continue
      end if



c Berechnung der Echtzeit in [s] ***************************************
      zeit=zeit+delta_t

c Rausschreiben der RX - Kinetik ***************************************
      write(uni_kin1,fmt=119) zeit,frac_rx


c wahlweise Cahn-Hagel-Analyse *****************************************
      if(i_ch.gt.0) then
      i=0
c hier wird jede Grenzflaeche zunaechst doppelt gezaehlt
      do 573 jz=2,ra3d_tat
```

```
      do 573 jx=2,rand tatx
      do 573 jy=2,rand taty
      if(rho alt(jx,jy,jz).le.rho situ) then
      if(rho alt(jx-1,jy,jz).gt.rho situ) i=i+1
      if(rho alt(jx+1,jy,jz).gt.rho situ) i=i+1
      if(rho alt(jx,jy-1,jz).gt.rho situ) i=i+1
      if(rho alt(jx,jy+1,jz).gt.rho situ) i=i+1
      if(rho alt(jx,jy,jz+1).gt.rho situ) i=i+1
      if(rho_alt(jx,jy,jz-1).gt.rho_situ) i=i+1
      end if
573   continue
c
      s=float(i)*vol inv
c s in 1 / mikrometer angegeben (siehe berechnung von vol_inv)
      write(uni_ch,fmt=1119) frac_rx,s
      end if

      zeitmax=ii

c Zwischengefuege rausschreiben
      do 374 igf=1,i raus
      if(frac rx.ge.(gef(igf)-abw).and.frac rx.le.(gef(igf)+abw)) then
C Kontrolle, ob auf diesen File schon geschrieben wurde
      if(zaa(igf).gt.98) goto 374
      zaa(igf)=99
      write(*,*)
      write(*,*) 'Schreibe Orientierungen binaer auf ',uni au(igf)
      write(*,*) 'Schreibe Versetzungsdichten binaer auf ',uni_rho(igf)
      write(*,*) 'RX Bruchteil ',frac_rx

      write(uni_au0,fmt=100)
      write(uni_au0,fmt=1509) 'Schreibe ori binaer auf',uni_au(igf)
      write(uni_au0,fmt=1509) 'Schreibe rho binaer auf',uni_rho(igf)
      write(uni_au0,fmt=917) rand_tatx-i_shift,rand_taty-i_shift,
     $ra3d_tat-i_shift,
     $' feldgroe.',frac_rx,' frac',zeit,' zeit'


c Zwischengefuege BINAER rausschreiben: Feldgroesse
c      xxx=float(rand_tat-i_shift)
c     write(uni au(igf)) xxx
c      xxx=float(ra3d_tat-i_shift)
c     write(uni_au(igf)) xxx

c Zustandsvariable(n rho und) Orientierung

c Zustandsvariablen Orientierung und Rho rausschreiben

      do 1534 iiiz=2,ra3d tat
      do 1534 iiix=2,rand tatx
      do 1534 iiiy=2,rand_taty

      xxx=float(rho alt(iiix,iiiy,iiiz))
      write(uni_rho(igf)) xxx

      xxx=fi1_alt(iiix,iiiy,iiiz)
      write(uni_au(igf)) xxx
      xxx=fi_alt(iiix,iiiy,iiiz)
      write(uni_au(igf)) xxx
      xxx=fi2_alt(iiix,iiiy,iiiz)
      write(uni_au(igf)) xxx

1534   continue

c      do 534 iiiz=2,ra3d tat
c      do 534 iiix=2,rand tat
c      do 629 iiiy=2,rand tat
c      write(uni_au(igf),fmt=103) iiix-i_shift,iiiy-i_shift,
c     $iiiz-i_shift,rho_alt(iiix,iiiy,iiiz),ori_alt(iiix,iiiy,iiiz)
c629   continue
c      write(uni_au(igf),fmt=100)
c534   continue
c

      end if

C File schliessen, damit nicht zu viele gleichzeitig offen sind
      if(zaa(igf).gt.98) close(uni_au(igf))
      if(zaa(igf).gt.98) close(uni_rho(igf))

374   continue
```

```
c Abbruchkriterium fuer die Zeitschleife ******************************
      frac_rel=abs(frac_rx-frac_alt)
c      if(frac_rel.le.frac_abb) goto 512
c      if(frac_rel.frac_rx.gt.frac_ab) then
c        i_frac=i_frac+1
c        if(i_frac.gt.2) goto 512
c     end if

      if(frac_rel.lt.frac_abb.and.frac_rx.gt.frac_ab-10.) goto 512
      if(frac_rx.gt.frac_ab) goto 512

c Ende der Haupt - Zeitschleife fuer Keimwachstum bis Zusammenstoss
      write(*,*) frac_rx,' prozent'
310   continue

c Sprungmarke Abbruchbedingung
512   continue

c Kinetik-, Haerte-, und Cahn-Hagel Dateien schliessen
      close(uni_kin1)
      close(uni_ch)
c     close(uni_hv)

c BINAER

      open(unit=uni_end_ori,recl=4,
     $file='c:\for\rx_konti2d\ori_end.bin',
     $form='unformatted',status='unknown',access='direct')

      open(unit=uni_end_rho,recl=4,
     $file='c:\for\rx_konti2d\rho_end.bin',
     $form='unformatted',status='unknown',access='direct')

c Endgefuege BINAER rausschreiben: Feldgroesse
c     write(uni_end,fmt=817) rand_tat-i_shift,ra3d_tat-i_shift
c817   format(2i10)

      do 620 iiiz=2,ra3d_tat
      do 620 iiix=2,rand_tatx
      do 620 iiiy=2,rand_taty
      xxx=float(rho_alt(iiix,iiiy,iiiz))
      write(uni_end_rho) xxx

      xxx=fi1_alt(iiix,iiiy,iiiz)
      write(uni_end_ori) xxx
      xxx=fi_alt(iiix,iiiy,iiiz)
      write(uni_end_ori) xxx
      xxx=fi2_alt(iiix,iiiy,iiiz)
      write(uni_end_ori) xxx
620   continue
      close(uni_end_ori)
      close(uni_end_rho)


c Laufzeit abfragen - Ende
      CALL GETTIM(ihr, imin, isec, i100th)
      sek_end=float(ihr)*3600.+float(imin)*60.+float(isec)

      write(*,*)
      write(*,*) 'Programmlaufzeit ', sek_end-sek_start,' Sekunden'
      write(uni_au0,fmt=100)
      write(uni_au0,fmt=108) sek_end-sek_start,' Programmlaufzeit [s]'

      goto 1999
999   continue
      write(*,*) ' FEHLER BEI DEN PHYSIKLAISCHEN PARAMETERN'
      write(*,*) ' FEHLER BEI DEN PHYSIKLAISCHEN PARAMETERN'
1999   continue



c Ausgabefile schliessen
      close(uni_au0)

c Programmende
      end
```

```
c XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C               UNTERPROGRAMME
c XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX


c ######################################################
      Subroutine ROTMAT(Omega,U,V,W,DREH3)
C
C bestimmt Drehmatrix DREH3 fuer Drehung um Omega um Achse (u,v,w)
C
      DIMENSION DREH3(3,3)
      BETRAG=SQRT(U**2+V**2+W**2)
      S=SIN(OMEGA)
      C=COS(OMEGA)
      U=U/BETRAG
      V=V/BETRAG
      W=W/BETRAG
      DREH3(1,1)=(1-U**2)*C+U**2
      DREH3(1,2)=U*V*(1-C)+W*S
      DREH3(1,3)=U*W*(1-C)-V*S
      DREH3(2,1)=U*V*(1-C)-W*S
      DREH3(2,2)=(1-V**2)*C+V**2
      DREH3(2,3)=V*W*(1-C)+U*S
      DREH3(3,1)=U*W*(1-C)+V*S
      DREH3(3,2)=V*W*(1-C)-U*S
      DREH3(3,3)=(1-W**2)*C+W**2
      RETURN
      END


      Subroutine EULDREH (P1,P,P2,ROTA)
C
C Best. Drehmatrix ROTA fuer Euler-Winkel
C
      DIMENSION ROTA(3,3)
C
      Pi=3.1415927
      FAKT=Pi/180.
      XP1=P1*FAKT
      XP=P*FAKT
      XP2=P2*FAKT
      C1=COS(XP1)
      C=COS(XP)
      C2=COS(XP2)
      S1=SIN(XP1)
      S=SIN(XP)
      S2=SIN(XP2)
      ROTA(1,1)=C1*C2-S1*S2*C
      ROTA(1,2)=S1*C2+C1*S2*C
      ROTA(1,3)=S2*S
      ROTA(2,1)=-C1*S2-S1*C2*C
      ROTA(2,2)=-S1*S2+C1*C2*C
      ROTA(2,3)=C2*S
      ROTA(3,1)=S1*S
      ROTA(3,2)=-C1*S
      ROTA(3,3)=C
      RETURN
      END







C
C **********************************************************************
C
     Subroutine DREHEN (DREH,DREH3)
C
C  Dreht Drehmatrix DREH um Matrix DREH3
C
      DIMENSION DREH(3,3),DREH3(3,3),ROTA(3,3)
        DO 1 I=1,3
         DO 1 J=1,3
1         ROTA(I,J)=0.0
C
        DO 4 I4=1,3
          DO 5 I5=1,3
            DO 6 I6=1,3
            ROTA(I4,I5)=ROTA(I4,I5)+DREH3(I4,I6)*DREH(I6,I5)
```

```
6            CONTINUE
5          CONTINUE
4        CONTINUE
C
         DO 9 I=1,3
          DO 9 J=1,3
9           DREH(I,J)=ROTA(I,J)
C
      RETURN
      END
C


C
C ***************************************************************************
C

      Subroutine DREHEUL (XMATR, Fi1,Fi,Fi2, FiR1,FiR,FiR2, PP, PPR)

C    Berechnung der Eulerwinkel aus der Dreh-Matrix XMATR
C    jeweils reduziert in den 1. Unterraum
C    mit:  0 < Fi,Fi2 < 90ø und 0 < Fi1< 360ø (allg. Eulerwinkel)
C    und:  0 < FiR1,FiR,FiR2 < 90ø  (reduzierte Eulerwinkel )


      DIMENSION ROTA(3,3), XMATR(3,3), PP(3,3), PPR(3,3), DREH3(3,3)
C
      Pi = 3.1415927
      FAKT = 180./Pi

      DO 100 I=1,3
        DO 100 J=1,3
          ROTA(I,J)=XMATR(I,J)           ! umspeichern
100   CONTINUE

C   Drehmatrix fuer 120ø<111>
      Omega = 120./FAKT
      X=1.
      Y=1.
      Z=1.
      CALL ROTMAT (OMEGA,X,Y,Z,DREH3)

C   Anfang der Schleife ueber die 3 Unterr„ume

C --- Drehen der Orientierung 120ø<111> ------------------
      DO 500 I=1,3
        CALL DREHEN (ROTA,DREH3)

C   Berechnen der Eulerwinkel
        A = ROTA(3,3)
        IF (ABS(A).GE..999999) GOTO 510

C --- Phi
        Fi = Pi/2.-ASIN(ROTA(3,3))
        SINFI= SIN(Fi)

C --- Phi 1
        IF (ROTA(3,2).NE.0) THEN
          Fi1 = -ATAN(ROTA(3,1)/ROTA(3,2))
          IF ((ROTA(3,2)/SINFI).GT.0) Fi1 = Pi+Fi1
        ELSE
          Fi1 = Pi/2 * ROTA(3,1) / ABS(ROTA(3,1))
        ENDIF

C --- Phi 2
        IF (ROTA(2,3).NE.0.) THEN
          Fi2 = ATAN(ROTA(1,3)/ROTA(2,3))
          IF ((ROTA(2,3)/SINFI).LT.0.) Fi2 = Pi+Fi2
        ELSE
          Fi2 = Pi/2 * ROTA(1,3) / ABS(ROTA(1,3))
        ENDIF

        GOTO 520

510     Fi = 0.
        Fi2 = 0.
        Fi1 = ASIN(ROTA(1,2))
520     CONTINUE

C    negative Winkel ?
        if (Fi.LT.0.)  Fi = (2.*Pi)+Fi
        if (Fi1.LT.0.)  Fi1 = (2.*Pi)+Fi1
```

```
        if (Fi2.LT.0.)  Fi2 = (2.*Pi)+Fi2

C ***UMRECHUNG AUFGRUND KUBISCHER KRISTALLSYMMETRIE: 0 <   Fi1  < 360 ***
C ***                                                0 < Fi,Fi2 <  90 ***

550     if (Fi.GT.(Pi/2.)) then
          Fi1 = Fi1+Pi
          Fi2 = -Fi2+Pi
          Fi = -Fi+Pi
          if (Fi.LT.0.) then
            Fi1 = Fi1+Pi
            Fi = -Fi
            Fi2 = Fi2+Pi
          endif
        endif
        if ((Fi.LT.0.) .OR. (Fi.GT.(Pi/2.))) GOTO 550

        if (Fi1.LT.0.)  Fi1 = Fi1+2.*Pi*(int(-Fi1/(2.*Pi))+1)
        if (Fi1.GT.(2.*Pi))  Fi1 = Fi1-2.*Pi*int(Fi1/(2.*Pi))

        if (Fi2.LT.0.)  Fi2 = Fi2+(Pi/2.)*(int(-Fi2/(Pi/2.))+1)
        if (Fi2.GT.(Pi/2.))  Fi2 = Fi2-(Pi/2.)*int(Fi2/(Pi/2.))


C   RUECKUMWANDLUNG IN DEG.
        Fi1 = Fi1*180./Pi
        Fi  = Fi *180./Pi
        Fi2 = Fi2*180./Pi
C   Reduzierung der Eulerwinkel in H1

C   0ø < Fi1 <= 90ø
        FiR1 = Fi1
        FiR  = Fi
        FiR2 = Fi2

C   90ø < Fi1 <= 180ø
        if ((Fi1.GT.90.) .AND. (Fi1.LE.180.)) then
          FiR1 = 180.-Fi1
          FiR2 = 90.-Fi2
        endif

C   180ø < Fi1 <= 270ø
        if ((Fi1.GT.180.) .AND. (Fi1.LE.270.))  FiR1 = Fi1-180.

C   270ø < Fi1 <= 360ø
        if (Fi1.GT.270.) then
          FiR1 = 360.-Fi1
          FiR2 = 90.-Fi2
        endif

C       Abspeichern im Feld des aktuellen Unterraums
        PP(i,1) = Fi1
        PP(i,2) = Fi
        PP(i,3) = Fi2
        PPR(i,1) = FiR1
        PPR(i,2) = FiR
        PPR(i,3) = FiR2

500   CONTINUE        ! Ende der Schleife, naechster Unterraum

C PP-Felder nach kleinerem Fi ordnen

      IMIN=1
600   DO 650 I=IMIN,3
        IF (PP(IMIN,2).GE.PP(I,2)) GOTO 650
        DO 610 J=1,3
          ZW=PP(IMIN,J)
          PP(IMIN,J)=PP(I,J)
          PP(I,J)=ZW
          ZW=PPR(IMIN,J)
          PPR(IMIN,J)=PPR(I,J)
          PPR(I,J)=ZW
610     CONTINUE
650   CONTINUE
      IMIN=IMIN+1
      IF (IMIN.LT.3) GOTO 600

      RETURN
      END
```

```
c ----------------------------------------
C subroutine ZUR BERECHNUNG VON ORIENTIERUNGSBEZIEHUNGEN ZWISCHEN
C ZWEI VORGEGEBENEN ORIENTIERUNGEN
      subroutine misori(p1,p,p2,al_min,ac_min)
       character meu,isy
       integer ac min(3)
       DIMENSION D(3,3,2),P1(2),P(2),P2(2)
     $,DM(3,3),DRR(3,3,24),A(3),IA(3),DR(3,3)

      common sym(3,3,24)

       PI=3.14159265

      pidurch=180./pi
      pimal=pi/180.

      al min=360.
      ac min(1)=0
      ac min(2)=0
      ac_min(3)=0


C *********************************
C ERSTELLEN DER BEIDEN DMATRIZEN
c 205    FORMAT(A1)

      meu='E'
      isy='E'
      I31MAX=4


       DO 31 I31=1,I31MAX
       DO 1 I1=1,2
       PP1=P1(I1)*pimal
       PP=P(I1)*pimal
       PP2=P2(I1)*pimal
       C1=COS(PP1)
       C=COS(PP)
       C2=COS(PP2)
       S1=SIN(PP1)
       S=SIN(PP)
       S2=SIN(PP2)
       D(1,1,I1)=C1*C2-S1*S2*C
       D(1,2,I1)=S1*C2+C1*S2*C
       D(1,3,I1)=S2*S
       D(2,1,I1)=-C1*S2-S1*C2*C
       D(2,2,I1)=-S1*S2+C1*C2*C
       D(2,3,I1)=C2*S
       D(3,1,I1)=S1*S
       D(3,2,I1)=-C1*S
       D(3,3,I1)=C
       GOTO 1
1      CONTINUE
C *********************************

       DO 24 I24=1,3
       DO 25 I25=1,3
       DM(I24,I25)=D(I24,I25,1)
       DR(I24,I25)=D(I24,I25,2)
25     CONTINUE
24     CONTINUE
        f=pimal
C
C ***************************************************
C BESTIMMUNG DER INVERSEN MATRIX ZUR ORIENTIERUNG 1:DM
C ***************************************************
       DO 3 I=1,3
       DO 3 J=1,3
3      DM(I,J)=D(J,I,1)
C ************************
C FELDER GLEICH NULL SETZEN
C ************************
       DO 4 I=1,3
       DO 4 J=1,3
       DO 4 K=1,24
       DR(I,J)=0.
4      DRR(I,J,K)=0.
C ***********************************************************
C MATRIZENMULTIPLIKATION DER MATRIZEN D(I,J,2) UND DM=DR(I,J)
C ***********************************************************
       DO 5 I=1,3
       DO 5 J=1,3
```

```
       DO 6 K=1,3
6      DR(I,J)=D(I,K,2)*DM(K,J)+DR(I,J)
5      CONTINUE
C ********************************************
C MATRIZENMULITIPLIKATION DER MATRIZEN SYM UND DR
C ********************************************
       DO 7 IZ=1,24
       DO 8 I=1,3
       DO 8 J=1,3
       DO 9 K=1,3
9      DRR(I,J,IZ)=SYM(I,K,IZ)*DR(K,J)+DRR(I,J,IZ)
8      CONTINUE
C ****************************
C BESTIMMUNG DES ROTATIONSWINKELS
C ****************************
       SPUR=0.
       DO 10 I=1,3
10     SPUR=SPUR+DRR(I,I,IZ)
       SP=(SPUR-1.)*0.499999

c       IF(SP.GE.1.)  SP=0.9999
c       IF(SP.LE.-1.) SP=-0.9999
       OMEGA=PI*0.5-ASIN(SP)
c       IOMEGA=INT(OMEGA*pidurch+0.5)
       ALPHA=OMEGA*pidurch
c       IF(IOMEGA.gt.179.and.iomega.lt.181) GOTO 12
c       IF(IOMEGA.gt.-1.and.iomega.lt.1) GOTO 12

       X=2.*SIN(OMEGA)
         if(x.lt.0.001) x=0.001
      xx=1.0/x


C ******************************************************
C BESTIMMUNG DER ROTATIONSACHSE, INCL SONDERFALL OMEGA = 180
C ******************************************************

       A(1)=100.*(DRR(2,3,IZ)-DRR(3,2,IZ))*xx
       A(2)=100.*(DRR(3,1,IZ)-DRR(1,3,IZ))*xx
       A(3)=100.*(DRR(1,2,IZ)-DRR(2,1,IZ))*xx


       GOTO 13
12     A(1)=SQRT((DRR(1,1,IZ)+1.)*0.5)*100.
       A(2)=SQRT((DRR(2,2,IZ)+1.)*0.5)*100.
       A(3)=SQRT((DRR(3,3,IZ)+1.)*0.5)*100.
13     DO 11 I=1,3
       IF(abs(A(I)).lt.0.001) GOTO 19
       IA(I)=INT(A(I)+0.5*A(I)/ABS(A(I)))
       GOTO 11
19     IA(I)=0
11     CONTINUE

      if(abs(alpha).lt.al_min) then
      al min=abs(alpha)
      ac min(1)=ia(1)
      ac min(2)=ia(2)
      ac min(3)=ia(3)
      end if

7      CONTINUE
       IF(I31.EQ.1) P1(2)=180.+P1(2)
       IF(I31.EQ.2) P1(2)=360.-P1(2)
       IF(I31.EQ.2) P2(2)=90.-P2(2)
       IF(I31.EQ.3) P1(2)=180.+P1(2)


31     CONTINUE
      i min=100
      if(abs(ac min(1)).lt.i_min.and.abs(ac_min(1)).gt.0)
     $i min=abs(ac min(1))
      if(abs(ac min(2)).lt.i_min.and.abs(ac_min(2)).gt.0)
     $i min=abs(ac min(2))
      if(abs(ac min(3)).lt.i_min.and.abs(ac_min(3)).gt.0)
     $i_min=abs(ac_min(3))

      do 5501 ii=1,3
5501  ac_min(ii)=nint(float(ac_min(ii))/float(i_min))

      return
      end
```