Bulletin of the Chemists and Technologists of Macedonia, Vol. **16**, No. 2, pp. 149–160 (1997)
GHTMDD – 299
Received: August 6, 1997
Accepted: December 19, 1997
ISSN 0350 – 0136
UDC: 548.3 : 681.5
Computer application

CRYMCALC – A COMPUTER PROGRAM FOR CRYSTAL MORPHOLOGY CALCULATIONS

Vladimir M. Petruševski[1], Vladimir G. Ivanovski[1] and Kostadin G. Trenčevski[2]

[1]*Institute of Chemistry, Faculty of Natural Sciences and Mathematics, The "Sv. Kiril & Metodij" University, POB 162, 91001 Skopje, Republic of Macedonia*
[2]*Institute of Mathematics, Faculty of Natural Sciences and Mathematics, The "Sv. Kiril & Metodij" University, POB 162, 91001 Skopje, Republic of Macedonia*

Correct indexing of single-crystal faces is vital in many areas of spectroscopy, crystallography and solid state physics and chemistry in general. Precise measurements of the angles between the crystal faces, combined by knowledge of the unit-cell parameters, allow positive identification of the types of faces. An equation for calculation of the angle between two planes, defined by their Müller indices, was derived for the most general case (oblique coordinate system). A computer program (*CRYMCALC*) written in Qbasic is used to calculate the theoretical values of the angles between different pairs of crystal planes which are then compared to the measured values. The value which is closest to the measured one reveals the pairs of planes in question.

Key words: crystal face indexing; Müller indices; crystal planes; calculation of angles

INTRODUCTION

Single-crystal spectroscopy (Raman, IR, UV-VIS etc.) has become an important tool for material research in the last few decades [1–3]. The interest in the application of these methods increases since much information obtainable in this way remains obscured when polycrystalline samples are studied. Single crystals of both natural and synthetic origin may easily be studied [4–5].

The first step, after the sample has been selected, is a proper alignment of the crystal with respect to the laboratory coordinate system. This orientation implies knowledge of the crystal morphology; that is, the crystal faces have to be correctly indexed. This task is not as trivial as it may look. A systematic method for face indexing is, therefore, desirable. The purpose of the present paper is to point to a possible solution to problems of this kind.

DEFINITION OF THE PROBLEM

Single crystal faces correspond to certain crystal planes. The easiest way to describe these planes is in terms of their Müller indices. Only the simplest representatives of the form ($\pm h$, $\pm k$, $\pm l$) are important, since all other planes of the form (nh, nk, nl) where n is an integer are parallel with them. In special cases (typically in crystals of high symmetry, or crystals with developed pinacoidal faces only), one may reveal the crystal morphology by simple inspection. Ordinarily, however, this is not possible. In such cases, perhaps the easiest approach for the correct indexing of crystal faces is the measurement

of the angle between a pair of faces sharing a common edge. This could be done by means of a goniometer or some other instrument of comparable precision. Providing the unit cell parameters are known, one could calculate the (theoretical) value of the angle between any two crystal planes. Then, by simple comparison of the calculated values with the measured one (allowing for the error of the goniometer/instrument), one may deduce the pair of planes in question. In this way three measurements of angles (typically angles built by three planes having a common vertex), will result in unambiguous face indexing.

In principle, one might use various available computer programs/packages to accomplish this task. There is a program *CRYSCALC* written by one of the authors some time ago [6], that can be used to complete the job. This program, however, was designed for routine calculations in crystallography, where the user defines a plane by a set of three atoms (i.e. the input data are the unit-cell parameters and the fractional coordinates of the atoms in question). For the present work, where systematic calculations of a number of angles are necessary, it is totally impractical. The program *CRYMCALC* that we present here may be considered as an attractive and time-saving alternative. A few comments on its structure and use will follow.

THE COMPUTER PROGRAM

The question of principal importance is: How can the angle between two crystal faces/planes be calculated? As mentioned, the easiest way to define a crystal face/plane is by means of its Müller indices. This implies the use of a modified segment form of an equation of plane. Note that the problem is not trivial, because in a general case the coordinate system may not be rectangular. The results of the solution to the problem are given in Appendix 1. As can be seen, the angle depends only on the unit cell parameters and the Müller indices of the two intersecting planes. This is of crucial significance to the problem, since the indices may be easily varied in a systematic way (using loops).

The computer program is given in Appendix 2, hence only a brief description will follow. The program is composed of several parts. The first part of the program (VariablesInUse) declares all variables that are to be used. The next part (Intro) gives the user all necessary information. UnitCellParameters is the third part, where a, b, c, α, β and γ are entered. Indices is a very important part, where the user picks lower and upper limits for the Müller indices of the two sets of planes. Integers from –4 to 4 are the legal input. It is highly recommended not to use the entire range (–4 to 4) for all indices of the two sets of planes, for this will eventually result in an overlong calculation. MainProgram is the central part of the program. It consists of six nested loops (two triplets of h, k, l indices) within which a subroutine for angle calculation is called. In this way, all relevant pairs of planes are taken into account. A little trick is used to skip unnecessary calculations for sets of planes that are parallel to what we call the simplest representatives. The *sums of cubes* : $|h|^3 + |k|^3 + |l|^3$ are calculated for both planes. Whenever any of these is equal to 8, 16, 24, 27, 54, 64, 72, 80, 81, 128, 136 or 192, the calculation is avoided. The value of 8 corresponds to a (200), (020) or (002) plane (or their symmetry equivalents) and these are, of course, parallel to the already included (100), (010), and (001) planes. The other numbers are explained similarly. The reason for using sum of cubes rather than squares is that the sum of squares is not unique. For example, both (330) and (411) planes have the same sum of squares – 18, but only the first one is to be avoided. Evidently, the sum of squares gives no criterion of redundancy. On the other hand, the sums of cubes for these two planes are 54 and 66, respectively.

Throughout the whole program most input errors are intercepted (the user being returned to the same data entry). The output results may (optionally) be saved to a disk sequential file under the name 'ANGLES.CMC'. Lots of comments are used in the program, to facilitate understanding its structure and the function of various parts (cf. Appendix 2).

Acknowledgement: The authors wish to express their sincere thanks to one of the referees who pointed out to a more compact and faster subroutine for the angle calculation. Although the gain in speed is tiny (and is limited mainly by the relatively slow print-to-disk routine, during sequential file write operation), the suggestion was adopted since it is the correct approach from the programmers point of view.

REFERENCES

[1] P. M. A. Sherwood, *Vibrational Spectroscopy of Solids*, Cambridge University Press, Cambridge, 1972.

[2] D. A. Long, *Raman Spectroscopy*, McGraw-Hill, New York–London, 1977.

[3] J. C. Decius, R. M. Hexter, *Molecular Vibrations in Crystals*, McGraw-Hill, New York, 1977.

[4] H. Takahashi, I. Maehara, N. Kaneko, *Spectrochim. Acta*, **39A**, 449 (1983).

[5] A. Goypiron, J. de Villepin, A. Novak, *J. Raman Spectrosc.* **9**, 297 (1980).

[6] V. M. Petruševski, *CRYSCALC – a computer program for various calculations in crystallography*, Institute of Chemistry, Faculty of Science, Skopje, 1990 (unpublished).

[7] K. Trenčevski, unpublished work.

APPENDIX 1

The crystal in question may belong to any of the six crystal systems. Let us consider an oblique coordinate system, as shown in Fig. 1. The equations of the two planes may be written as:

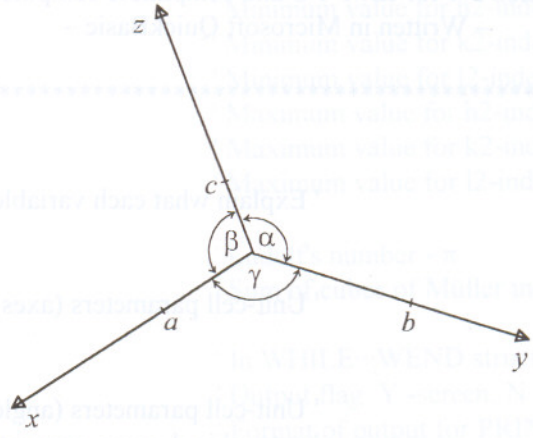

Fig. 1. The oblique coordinate system in use

$$\Sigma_1: h_1x + k_1y + l_1z = 1$$
$$\Sigma_2: h_2x + k_2y + l_2z = 1$$

where h_i, k_i and l_i are the Müller indices of the planes. After some lengthy manipulations, one arrives [7] at the following formula for calculation of the angle between the planes:

$$\phi = \arccos\left(\frac{A}{\sqrt{BC}}\right)$$

where ϕ is the angle, and the quantities A, B, C are defined below:

$$A = a^2b^2l_1l_2\sin^2\gamma + b^2c^2h_1h_2\sin^2\alpha + c^2a^2k_1k_2\sin^2\beta + ab^2c(l_1h_2 + l_2h_1)(\cos\gamma\cos\alpha - \cos\beta) +$$
$$+ bc^2a(k_1h_2 + k_2h_1)(\cos\alpha\cos\beta - \cos\gamma) + ca^2b(k_1l_2 + k_2l_1)(\cos\beta\cos\gamma - \cos\alpha)$$

$$B = a^2b^2l_1^2\sin^2\gamma + b^2c^2h_1^2\sin^2\alpha + c^2a^2k_1^2\sin^2\beta + 2ab^2ch_1l_1(\cos\gamma\cos\alpha - \cos\beta) +$$
$$+ 2bc^2ak_1h_1(\cos\alpha\cos\beta - \cos\gamma) + 2ca^2bl_1k_1(\cos\beta\cos\gamma - \cos\alpha)$$

$$C = a^2b^2l_2^2\sin^2\gamma + b^2c^2h_2^2\sin^2\alpha + c^2a^2k_2^2\sin^2\beta + 2ab^2ch_2l_2(\cos\gamma\cos\alpha - \cos\beta) +$$
$$+ 2bc^2ak_2h_2(\cos\alpha\cos\beta - \cos\gamma) + 2ca^2bl_2k_2(\cos\beta\cos\gamma - \cos\alpha)$$

APPENDIX 2

LISTING OF THE COMPUTER PROGRAM

```
REM ************************************************************************
REM *                                                                      *
REM *     C R Y M C A L C  -  CRYSTAL MORPHOLOGY CALCULATIONS              *
REM *                                                                      *
REM *                                by                                    *
REM *                                                                      *
REM *          Vladimir M. Petrusevski & Vladimir G. Ivanovski             *
REM *                Department of Physical Chemistry,                     *
REM *          Institute of Chemistry, Faculty of Science,                 *
REM *              Arhimedova 5, 91000 Skopje, Macedonia                   *
REM *                                                                      *
REM *          Version 2.1 for IBM-PC and compatible computers             *
REM *                - Written in Microsoft QuickBasic -                   *
REM *                                                                      *
REM ************************************************************************

VariablesInUse:                    ' Explain what each variable means     '
  a = 0
  b = 0                            ' Unit-cell parameters (axes)          '
  c = 0

  alpha = 0                        ' Unit-cell parameters (angles:        '
  betta = 0                        ' entered in degrees, converted        '
  gamma = 0                        ' into radians)                        '

  ca = 0: cb = 0: cg = 0           ' Cosines of angles α, β and γ         '
  sa = 0: sb = 0: sg = 0           ' Sines of angles α, β and γ           '

  aabb = 0: bbcc = 0: ccaa = 0     ' Products: aabb = a*a*b*b etc.        '
  abbc = 0: bcca = 0: caab = 0     ' Products: abbc = a*b*b*c etc.        '
  sasa = 0: sbsb = 0: sgsg = 0     ' Products: sasa = sa*sa etc.          '
  cgab = 0: cabg = 0: cbga = 0     ' Products: cgab = cg*ca-cb etc.       '

  Aa = 0: Bb = 0: Cc = 0           ' Variables to calculate cosfi         '

  Counter = 0                      ' Number of angles calculated          '

  cosfi = 0                        ' Cosine of angle between planes,      '
                                   ' defined as cosfi = Aa/SQR(Bb*Cc)     '
  fi = 0                           ' Angle between the planes             '
  Fl = 0                           ' Output flag: 1- screen, 2- disk      '
```

```
h1 = 0
k1 = 0                          ' Müller indices of first plane
l1 = 0
h2 = 0
k2 = 0                          ' Müller indices of second plane
l2 = 0
Limit = 0                       ' Index limit (between 1 and 4)

H1min = 0                       ' Minimum value for h1-index
K1min = 0                       ' Minimum value for k1-index
L1min = 0                       ' Minimum value for l1-index
H1max = 0                       ' Maximum value for h1-index
K1max = 0                       ' Maximum value for k1-index
L1max = 0                       ' Maximum value for l1-index

H2min = 0                       ' Minimum value for h2-index
K2min = 0                       ' Minimum value for k2-index
L2min = 0                       ' Minimum value for l2-index
H2max = 0                       ' Maximum value for h2-index
K2max = 0                       ' Maximum value for k2-index
L2max = 0                       ' Maximum value for l2-index

Pi = 4 * ATN(1)                 ' Ludolf's number - π
Sc1 = 0: Sc2 = 0                ' Sum of cubes of Müller indices

Answer$ = ""                    ' In WHILE···WEND structures: Y/N
Flag$ = ""                      ' Output flag: Y -screen; N -disk
Format$ = ""                    ' Format of output for PRINT USING
Limit$ = ""                     ' String-value of variable Limit
Sp$ = ""                        ' In WHILE···WEND structures: wait
                                ' for SPACE, Q or R to be pressed

Intro:                          ' User information

CLS
PRINT
PRINT "   C R Y s t a l   M o r p h o l o g y   C A L C u l a t i o n s"
PRINT : PRINT
PRINT "          The program calculates automatically the angle between"
PRINT "   two planes, given by their Müller indices.  No calculations are"
PRINT "   done for parallel planes except for  the simplest representati-"
PRINT "   ve. That is (100), (010), (001), (110), (101) etc. are included"
PRINT "   but (n00), (0n0), (00n), (nn0), (n0n) etc. are not.  The result"
PRINT "   is printed on the screen."
PRINT "          The input data are the unit cell parameters (a,b,c may"
PRINT "   be entered in arbitrary units, angles are entered in degrees °;"
PRINT "   the inequalities α < β + γ; β < γ + α; γ < α + β must all hold)"
PRINT "   and the limiting values of the  Müller indices  (accounting for"
PRINT "   the planes to be included in the calculation)."
PRINT "          The results may be optionally saved to disk, under the"
```

```
PRINT "    name 'ANGLES.CMC',  in the current directory.  Rename this file"
PRINT "    (if you want to keep it) immediately after output is redirected"
PRINT "    to disk, for it may be invariably lost."
GOSUB ContinueOrQuit

UnitCellParameters:                    ' Enter a, b, c, α, β, γ                '

CLS
PRINT : PRINT
PRINT "  Enter unit-cell parameters:"
PRINT : PRINT
INPUT "  a/au = "; a
INPUT "  b/au = "; b
INPUT "  c/au = "; c
PRINT

INPUT "  α/° = "; alpha
INPUT "  β/° = "; betta
INPUT "  γ/° = "; gamma

                                       ' Interception of errors                '
IF alpha >= betta + gamma THEN GOTO UnitCellParameters
IF betta >= gamma + alpha THEN GOTO UnitCellParameters
IF gamma >= alpha + betta THEN GOTO UnitCellParameters
IF alpha + betta + gamma >= 360 THEN GOTO UnitCellParameters

                                       ' Conversion: degrees to radians        '
alpha = alpha * Pi / 180
betta = betta * Pi / 180
gamma = gamma * Pi / 180

GOSUB Correct                          ' Check input data for errors           '
IF Answer$ = "N" THEN GOTO UnitCellParameters

                                       ' Products of unit-cell parameters      '
aabb = a * a * b * b: bbcc = b * b * c * c: ccaa = c * c * a * a
abbc = a * b * b * c: bcca = b * c * c * a: caab = c * a * a * b

Indices:                               ' Low and high values of h, k, l        '

CLS
PRINT : PRINT
PRINT "    Enter low and high limits for Müller indices (-4 to 4). Be-"
PRINT "    ware that using the entire range will result in an overlong"
PRINT "    calculation (more than 330,000 pairs of planes).  It may be"
PRINT "    a good idea to start  with a narrow range of indices and to"
PRINT "    repeat the calculation with an extended range if it appears"
PRINT "    necessary."
PRINT
PRINT "    Entering identical values for high and low values of Müller"
```

```
  PRINT "      indices  for the first plane  will restrict the calculation"
  PRINT "      to the angles built with that specific plane."
  PRINT : PRINT : PRINT
  GOSUB ContinueOrQuit

h1:

  CLS
  PRINT "  Enter low limit for h1-index  "
  GOSUB IndexLimit
  H1min = Limit
  PRINT "  Enter high limit for h1-index "
  GOSUB IndexLimit
  H1max = Limit
  IF H1min > H1max THEN GOTO h1

k1:

  CLS
  PRINT "  Enter low limit for k1-index  "
  GOSUB IndexLimit
  K1min = Limit
  PRINT "  Enter high limit for k1-index "
  GOSUB IndexLimit
  K1max = Limit
  IF K1min > K1max THEN GOTO k1

l1:

  CLS
  PRINT "  Enter low limit for l1-index  "
  GOSUB IndexLimit
  L1min = Limit
  PRINT "  Enter high limit for l1-index "
  GOSUB IndexLimit
  L1max = Limit
  IF L1min > L1max THEN GOTO l1

h2:

  CLS
  PRINT "  Enter low limit for h2-index  "
  GOSUB IndexLimit
  H2min = Limit
  PRINT "  Enter high limit for h2-index "
  GOSUB IndexLimit
  H2max = Limit
  IF H2min > H2max THEN GOTO h2

k2:

  CLS
```

```
PRINT "  Enter low limit for k2-index  "
GOSUB IndexLimit
K2min = Limit
PRINT "  Enter high limit for k2-index "
GOSUB IndexLimit
K2max = Limit
IF K2min > K2max THEN GOTO k2

l2:

CLS
PRINT "  Enter low limit for l2-index  "
GOSUB IndexLimit
L2min = Limit
PRINT "  Enter high limit for l2-index "
GOSUB IndexLimit
L2max = Limit
IF L2min > L2max THEN GOTO l2

ReviewIndices:                          ' Print limits to check input          '

CLS
PRINT
PRINT "  H1min = "; H1min, "H1max = "; H1max
PRINT "  K1min = "; K1min, "K1max = "; K1max
PRINT "  L1min = "; L1min, "L1max = "; L1max
PRINT
PRINT "  H2min = "; H2min, "H2max = "; H2max
PRINT "  K2min = "; K2min, "K2max = "; K2max
PRINT "  L2min = "; L2min, "L2max = "; L2max

GOSUB Correct                           ' Check input data for errors          '
IF Answer$ = "N" THEN GOTO Indices


ScreenOrDisk:                           ' Print angles to screen or disk       '

CLOSE #1: CLOSE #2                      ' Ensure all channels are closed       '

PRINT : PRINT
PRINT "  Output results to disk ?   (Y/N)"
PRINT : PRINT

Flag$ = ""
WHILE Flag$ <> "Y" AND Flag$ <> "N"
  Flag$ = UCASE$(INKEY$)
WEND

IF Flag$ = "N" THEN                     ' Output angles to screen              '
  Fl = 1: OPEN "SCRN:" FOR OUTPUT AS #1
```

```
ELSE                    ' Output angles to disk                    '
 Fl = 2: OPEN "ANGLES.CMC" FOR OUTPUT AS #2
 PRINT #Fl, " Counter  h1 k1 l1      h2 k2 l2          Angle/°"
 PRINT #Fl,
END IF

TrigFunctions:                      ' Sin and Cos functions of α,β,γ,          '
                                    ' used in calculation of angles            '

sa = SIN(alpha): sb = SIN(betta): sg = SIN(gamma)
ca = COS(alpha): cb = COS(betta): cg = COS(gamma)
sasa = sa * sa: sbsb = sb * sb: sgsg = sg * sg
cabg = ca * cb - cg: cbga = cb * cg - ca: cgab = cg * ca - cb

MainProgram:                        ' Angles between all possible              '
                                    ' pairs of planes                          '

Counter = 0
FOR h1 = H1min TO H1max
FOR k1 = K1min TO K1max
FOR l1 = L1min TO L1max
 FOR h2 = H2min TO H2max
 FOR k2 = K2min TO K2max
 FOR l2 = L2min TO L2max

  Sc1 = ABS(h1 ^ 3) + ABS(k1 ^ 3) + ABS(l1 ^ 3)
  Sc2 = ABS(h2 ^ 3) + ABS(k2 ^ 3) + ABS(l2 ^ 3)

                                    ' The sum of cubes is unique,              '
                                    ' unlike the sum of squares.               '
                                    ' Skip the following planes:               '

  IF Sc1 = 0 OR Sc2 = 0 THEN GOTO Jump          ' (000)-no such plane        '
  IF Sc1 = 8 OR Sc2 = 8 THEN GOTO Jump          ' (200), (020), (002)        '
  IF Sc1 = 16 OR Sc2 = 16 THEN GOTO Jump        ' (220), (202), (022)        '
  IF Sc1 = 24 OR Sc2 = 24 THEN GOTO Jump        ' (222)                      '
  IF Sc1 = 27 OR Sc2 = 27 THEN GOTO Jump        ' (300), (030), (003)        '
  IF Sc1 = 54 OR Sc2 = 54 THEN GOTO Jump        ' (330), (303), (033)        '
  IF Sc1 = 64 OR Sc2 = 64 THEN GOTO Jump        ' (400), (040), (004)        '
  IF Sc1 = 72 OR Sc2 = 72 THEN GOTO Jump        ' (420), (204), (042)        '
                                                ' (402), (024), (240)        '
  IF Sc1 = 80 OR Sc2 = 80 THEN GOTO Jump        ' (422), (242), (224)        '
  IF Sc1 = 81 OR Sc2 = 81 THEN GOTO Jump        ' (333)                      '
  IF Sc1 = 128 OR Sc2 = 128 THEN GOTO Jump      ' (440), (404), (044)        '
  IF Sc1 = 136 OR Sc2 = 136 THEN GOTO Jump      ' (442), (424), (244)        '
  IF Sc1 = 192 OR Sc2 = 192 THEN GOTO Jump      ' (444)                      '

                                    ' Same (1) or parallel (2) plane           '
  IF h1 = h2 AND k1 = k2 AND l1 = l2 THEN GOTO Jump      '  (1)              '
  IF h1 = -h2 AND k1 = -k2 AND l1 = -l2 THEN GOTO Jump   '  (2)              '
```

```
  GOSUB Angle                          ' cos(fi) is calculated, where fi      '
                                       ' is the angle between the planes      '

ResultsOut:                            ' Results to screen or disk            '

  Counter = Counter + 1
                                       ' If output is sent to screen,         '
                                       ' print "pages" with 20 rows           '
  IF Fl = 1 THEN
   IF INT((Counter - 1) / 20) = (Counter - 1) / 20 THEN
    GOSUB ContinueOrQuit
    CLS
    PRINT #Fl, " Counter   h1 k1 l1      h2 k2 l2              Angle/°"
   END IF
  END IF

                                       ' Disk output - continuous print       '
  Format$ = " ######    ## ## ##      ## ## ##              ###.##"
  PRINT #Fl, USING Format$; Counter; h1; k1; l1; h2; k2; l2; fi

Jump:                                  ' Jump here after every calculated     '
                                       ' angle or skipped plane               '
  NEXT l2
  NEXT k2
  NEXT h2
 NEXT l1
 NEXT k1
 NEXT h1

RestartOrEnd:                          ' Restart or terminate                 '

 IF Fl = 1 THEN GOTO ScreenOrDisk

 IF Fl = 2 THEN END                    ' Avoid loss of saved disk data        '


Angle:          ' Subroutine for angle calculation '

 Aa = aabb * l1 * l2 * sgsg + bbcc * h1 * h2 * sasa + ccaa * k1 * k2 * sbsb
 Aa = Aa + abbc * (h1 * l2 + h2 * l1) * cgab
 Aa = Aa + bcca * (k1 * h2 + k2 * h1) * cabg
 Aa = Aa + caab * (k1 * l2 + k2 * l1) * cbga

 Bb = aabb * l1 * l1 * sgsg + bbcc * h1 * h1 * sasa + ccaa * k1 * k1 * sbsb
 Bb = Bb + 2 * abbc * h1 * l1 * cgab
 Bb = Bb + 2 * bcca * k1 * h1 * cabg
 Bb = Bb + 2 * caab * l1 * k1 * cbga

 Cc = aabb * l2 * l2 * sgsg + bbcc * h2 * h2 * sasa + ccaa * k2 * k2 * sbsb
```

```
  Cc = Cc + 2 * abbc * h2 * l2 * cgab
  Cc = Cc + 2 * bcca * k2 * h2 * cabg
  Cc = Cc + 2 * caab * l2 * k2 * cbga

  cosfi = Aa / SQR(Bb * Cc)
  IF ABS(cosfi) = 1 THEN
   fi = 0
  ELSE
   fi = 180 / Pi * (Pi / 2 + ATN(cosfi / SQR(1 - cosfi * cosfi)))
  END IF

RETURN

Correct:                                ' In case of error, reenter data        '

  PRINT : PRINT
  PRINT "  Are these values correct?      (Y/N)"

  Answer$ = ""
  WHILE Answer$ <> "Y" AND Answer$ <> "N"
    Answer$ = UCASE$(INKEY$)
  WEND

RETURN

ContinueOrQuit:                         ' Space - to continue execution;        '
                                        ' Q - to quit; R - for a new run        '

  LOCATE 23
  PRINT "      Press space to continue, 'Q' to quit or 'R' for a new run"
  Sp$ = ""

  WHILE (Sp$ <> " " AND Sp$ <> "Q" AND Sp$ <> "R")
    Sp$ = UCASE$(INKEY$)
  WEND

  IF Sp$ = "Q" THEN END
  IF Sp$ = "R" THEN RUN

RETURN

IndexLimit:                             ' Set limits for h, k, l                '

  Limit = 5
  WHILE ABS(Limit) > 4 OR INT(Limit) <> Limit
    LOCATE CSRLIN - 1, 32
    INPUT Limit
  WEND

RETURN
```

Резиме

CRYMCALC – КОМПЈУТЕРСКА ПРОГРАМА ЗА ПРЕСМЕТКИ ВО ВРСКА СО МОРФОЛОГИЈАТА НА КРИСТАЛИТЕ

Владимир М. Петрушевски[1], **Владимир Ѓ. Ивановски**[1] и **Костадин Г. Тренчевски**[2]

[1] *Институт за хемија, ПМФ, Универзитет „Св. Кирил и Методиј", Архимедова 5, Скопје, Република Македонија*
[2] *Институт за математика, ПМФ, Универзитет „Св. Кирил и Методиј", Архимедова 5, Скопје, Република Македонија*

Клучни зборови: индексирање на површините на кристали; Милерови индекси; кристална рамнина; пресметка на агол

Точно индексирање на површините од монокристали е важно во многу подрачја на спектроскопијата, кристалографијата и физиката и хемијата на цврста состојба воопшто. Прецизното мерење на аглите, заедно со познавањето на параметрите на елементарната ќелија, овозможува сигурна идентификација на плоските. Изведена е равенка за пресметување на вредноста на аголот помеѓу две рамнини во кристалот, зададени со нивните Милерови идекси, за најопшт случај (косоаголен координатен систем). За пресметување на теориската вредност на аголот помеѓу различни двојки рамнини е напишана компјутерска програма (*CRYMCALC*) во програмскиот јазик Qbasic. Пресметаната вредност што е најблиска до измерената открива за која двојка рамнини станува збор.