Manuals of some programs for SCB-MW distribution estimation

Rapport IWDE 02 - 03

Manuals of some Programs for SCB-MW Distribution Estimation

J. Molenaar

April 2002

Instituut
Wiskundige Dienstverlening
Eindhoven

Technische Universiteit Eindhoven

Estimation of SCB-MW distributions

Introduction

In report IWDE 01-04 entitled "Estimating SCB-MW distributions from partial data" is discussed how certain distributions that characterize polymer properties could be estimated from measured data. In the present report the manuals of the corresponding software programs are given.
In program SCBD a one-dimensional distribution is given as a number of data points on a regular grid. The distribution is assumed to consist out of an unknown number of Gaussian peaks. The positions, heights, and widths of these peaks are estimated in an efficient way.
In programs SCBD-SKEW1 and 2 it is allowed that the peaks are not symmetric but skew. In SKEW1 the so-called Weibull distribution is applied, and in SKEW2 an alternative correction on the Gaussian distribution. It turns out in practice that the method in SKEW2 has by far preference to be applied to real data.
In program SCB-MWD a two-dimensional distribution is analyzed. It is assumed that the distribution is the sum of two-dimensional Gaussian peaks.
The listings of the programs are included at the end of this report.

Manual of program SCBD

In SCBD given SCBD-Mw distributions are analyzed. The structure of the input file is as follows. Per distribution we have ngrid + 2 lines:

- modulus
- tiechain
- ngrid lines with 4 real numbers; the first one is the temperature, the fourth the SCB value; the two other data values are not used.

While reading the input file the number of distributions ('NTEL') on the input file is counted.
Per distribution the parameters characterizing the peaks are determined in subroutine 'maxima'. The distribution is assumed to be composed out of a number of Gaussian peaks. Each peak is characterized by 4 parameters: xmax, alfa, beta, area. The 12 parameters of the three heighest peaks are, together with the corresponding modulus and tiechain, written on one line of the output file. If the distribution contains less than 3 peaks, some of the parameters will be equal to zero.

Subroutine maxima

The SCB distribution is given as ngrid data points (x_i, f_i), $= 1, ..., \text{ngrid}$. The temperature grid points x_i are stored in the first column of array x, the SCB values are stored in the fourth column of array x.
In subroutine maxima first a rough estimation of the position of a peak is found. It detects three data points (x_0, f_0), (x_1, f_1), (x_2, f_2) for which it holds that $f_0 \leq f_1$ and $f_1 \geq f_2$. Then, a maximum is present between x_0 and x_2. It is assumed that this peak has *locally* the Gaussian or 'manual' shape:

$$f(x) = \alpha e^{-\beta(x-\overline{x})^2} \tag{1}$$

So, the peak is characterized by the parameters:

- $\overline{x}$: peak position
- α: peak height
- β: peak width.

These parameters are estimated from the equations:

$$\begin{aligned} f_0 &= \alpha e^{-\beta(x_0-\overline{x})^2} \\ f_1 &= \alpha e^{-\beta(x_1-\overline{x})^2} \\ f_2 &= \alpha e^{-\beta(x_3-\overline{x})^2} \end{aligned} \tag{2}$$

Dividing and taking logarithms we obtain

$$\ln\left(\frac{f_0}{f_1}\right) = -\beta((x_0-\overline{x})^2-(x_1-\overline{x})^2)$$
$$\ln\left(\frac{f_0}{f_2}\right) = -\beta((x_0-\overline{x})^2-(x_2-\overline{x})^2) \tag{3}$$

This eliminates the parameter α. The parameter β is eliminated by introducing the factor Q:

$$Q = \frac{\ln(f_0/f_1)}{\ln(f_0/f_2)} = \frac{(x_0-\overline{x})^2-(x_1-\overline{x})^2}{(x_0-\overline{x})^2-(x_2-\overline{x})^2} \tag{4}$$

From this equation we can find $\overline{x}$. Although the equation seems to be quadratic in $\overline{x}$, it turns out that the quadratic terms drop out and eventually we arrive at:

$$\overline{x} = \tfrac{1}{2}\frac{x_0^2-x_1^2-Q(x_0^2-x_2^2)}{x_0-x_1-Q(x_0-x_2)} \tag{5}$$

The estimate for β then follows from (3a):

$$\beta = -\frac{\ln(f_0/f_1)}{(x_0-\overline{x})^2-(x_1-\overline{x})^2} \tag{6}$$

The estimate for α can be calculated from (2a):

$$\alpha = f_0 e^{\beta(x_0-\overline{x})^2} \tag{7}$$

Although the parameter β characterizes the peak width completely, it turns out that the area under the peak is also an important parameter to be used in regression models. This area is given by

$$\text{area} = \int_{-\infty}^{+\infty} \alpha e^{-\beta(x-\overline{x})^2} dx = \alpha\sqrt{\frac{\pi}{\beta}} \tag{8}$$

The estimation procedure for peak parameters is applied to nmax peaks, since it is assumed that the SCB distribution can be decomposed as

$$\text{SCB}(x) = \sum_{i=1}^{\text{nmax}} \alpha(i) e^{-\beta(i)(x-\overline{x}(i))^2} \tag{9}$$

If the first peak is estimated, it is subtracted from the data and the procedure is repeated. The peak positions $\overline{x}(i)$ are stored in array 'xmax', the parameters $\alpha(i)$ in array 'alfa', and the parameters $\beta(i)$ in array 'beta'. The three areas are stored in array 'area'. These arrays are written onto the output file, preceded by the corresponding values of modulus and tiechain.

In the data provided until now, the number nmax never exceeds 3, but this is not at all an essential limitation for the present method.

Manual of programs SCBD-SKEW1 and SCBD-SKEW2

These programs are an extended versions of program SCBD and allow for peaks that are not symmetrically shaped. The non-symmetric profile is represented by a modified version of the symmetric Gaussian profile as used in SCBD. However, the modification is not unique. We present here two alternatives leading to the programs SCBD-SKEW1 and SCBD-SKEW2. In practice it turns out that SCBD-SKEW2 is the most appropriate version, since the representation used in SCBD-SKEW1 does not always fit the measured profiles well, so that the fitting procedure does often not converge.

1. SCBD-SKEW1

For the non-symmetric profile an appropriate version of the so-called Weibull (or Tung) distribution is chosen. In this approach it is assumed that a peak has locally a shape described by

$$f(x) = \alpha\, x^{\gamma}\, e^{-\beta(x-\overline{x})^2},\ x > 0\ . \tag{1}$$

Note the difference with the Gaussian distribution used in SCBD. The extra factor x^{γ} in (1) allows the peak to be non-symmetric around $\overline{x}$. The deviation from a symmetrically shaped peak is measured by the parameter γ. If $\gamma = 0$ the peak is symmetric. If $\gamma > 0$ the right-hand side of the peak is wider then the left-hand side, whereas for $\gamma < 0$ it is the other way around.
Since (1) is applicable only if $x > 0$, we first scale the range of the x-values (i.e. the temperatures) such that they vary over the interval $[0, 1]$.

Since $f(x)$ in (1) has four parameters to be estimated $(\alpha, \gamma, \beta, \overline{x})$, we need four data points in the vicinity of the peak, say (x_0, f_0), (x_1, f_1), (x_2, f_2), and (x_4, f_4).
The parameters are estimated from the equations

$$\begin{aligned} f_0 &= \alpha\, x_0^{\gamma}\, e^{-\beta(x_0-\overline{x})^2} \\ f_1 &= \alpha\, x_1^{\gamma}\, e^{-\beta(x_1-\overline{x})^2} \\ f_2 &= \alpha\, x_2^{\gamma}\, e^{-\beta(x_2-\overline{x})^2} \\ f_3 &= \alpha\, x_3^{\gamma}\, e^{-\beta(x_3-\overline{x})^2} \end{aligned} \tag{2}$$

From (2) we find for $i = 0, 1, 2, 3$:

$$\ln f_i = \ln\alpha + \gamma \ln x_i - \beta(x_i - \overline{x})^2 \tag{3}$$

Subtracting these equations we obtain

$$\ln\left(\frac{f_0}{f_1}\right) = \gamma \ln\left(\frac{x_0}{x_1}\right) - \beta((x_0 - \overline{x})^2 - (x_1 - \overline{x})^2) \tag{4a}$$

and

$$\ln\left(\frac{f_2}{f_3}\right) = \gamma \ln\left(\frac{x_2}{x_3}\right) - \beta((x_2 - \overline{x})^2 - (x_3 - \overline{x})^2) \tag{4b}$$

From (4) and (5) it becomes clear that the quantity P_1 defined by

$$P_1 \equiv \ln\left(\frac{x_2}{x_3}\right) \cdot \ln\left(\frac{f_0}{f_1}\right) - \ln\left(\frac{x_0}{x_1}\right) \cdot \ln\left(\frac{f_2}{f_3}\right) , \tag{5}$$

plays an essential role, since

$$P_1 = -\beta\Big[\ln\left(\frac{x_2}{x_3}\right)(x_0^2 - x_1^2 - 2\overline{x}(x_0 - x_1))$$

$$- \ln\left(\frac{x_0}{x_1}\right)(x_2^2 - x_3^3 - 2\overline{x}(x_2 - x_3))\Big] . \tag{6}$$

For conciseness' sake we rewrite this as

$$P_1 = -\beta[a(0,1,2,3) + b(0,1,2,3)\overline{x}] \tag{7}$$

with the coefficients a and b defined by

$$a(0,1,2,3) \equiv \ln\left(\frac{x_2}{x_3}\right)(x_0^2 - x_1^2) - \ln\left(\frac{x_0}{x_1}\right)(x_2^2 - x_3^2) \tag{8}$$

$$b(0,1,2,3) \equiv -2[\ln\left(\frac{x_2}{x_3}\right)(x_0 - x_1) - \ln\left(\frac{x_0}{x_1}\right)(x_2 - x_3)] \tag{9}$$

If we apply the procedure in (4)-(7) and use instead of f_0, f_1, f_2, f_3 the permuted values f_0, f_2, f_1, f_3, we arrive at

$$P_2 = -\beta[a(0,2,1,3) + b(0,2,1,3)\overline{x}] , \tag{10}$$

where the definitions (6) - (9) are used with permuted entries. From (7) and (10) we have

$$\frac{P_1}{P_2} = \frac{a(0,1,2,3) + b(0,1,2,3)\overline{x}}{a(0,2,1,3) + b(0,2,1,3)\overline{x}} \tag{11}$$

This immediately yields the estimate for $\overline{x}$:

$$\overline{x} = \frac{P_2 \cdot a(0,1,2,3) - P_1 \cdot a(0,2,1,3)}{P_1 \cdot b(0,2,1,3) - P_2 \cdot b(0,1,2,3)} \tag{12}$$

The estimate for β follows from (7):

$$\beta = \frac{-P_1}{a(0,1,2,3) + b(0,1,2,3)\overline{x}} \tag{13}$$

The parameter γ follows from (4):

$$\gamma = \frac{\ln\left(\frac{f_0}{f_1}\right) + \beta((x_0 - \overline{x})^2 - (x_1 - \overline{x})^2)}{\ln\left(\frac{x_0}{x_1}\right)} \tag{14}$$

Finally, the height α can be deduced from (2):

$$\alpha = f_0 \, x_0^{-\gamma} \, e^{\beta(x_0 - \overline{x})^2} \tag{15}$$

In program SCBD the area under the peak is calculated. For the Weibull distribution (1) this property is hard to calculate from an explicit expression. However, since (1) has already an extra parameter compared to the Gaussian distribution, it is to be expected that the area is not a relevant regression parameter here, so it is omitted.

2. SCBD-SKEW2

In this alternative approach we modify the symmetric Gaussian profile by multiplying it with a straight line:

$$f(x) = \alpha(1 + s(x - \overline{x}))e^{-\beta(x-\overline{x})^2} \tag{16}$$

The slope s is directly related to the skewness of the peak. For $s = 0$, which corresponds to a horizontal line, the Gaussian profile is obtained. If $s > 0$, the peak is skew to the right-hand side, i.e. the area under the peak at the right-hand side of the peak position $\overline{x}$ is bigger than the one left of the peak position. For $s < 0$ it is the other way around. Note that in (16) α represents the peak height, β the peak width, and $\overline{x}$ peak position. To estimate α, β, $\overline{x}$, and s from data in the same sprint as in SCBD-SKEW1 (see above), is a hard task. Therefore, we make use of the fact that peak height and peak width are nearly independent from the skewness of the peak. The strategy is to estimate α, β, and $\overline{x}$ in the same way as done in SCBD. For the estimation of these parameters we refer to the manual of SCBD. To estimate the skewness parameter s, we take two datapoints (x_0, f_0) and (x_1, f_1) at both sides of the peak, so $x_0 < \overline{x}$ and $x_1 > \overline{x}$. This yields the equations

$$f_0 = f(x_0) = \alpha(1 + s(x_0 - \overline{x}))e^{-\beta(x_0-\overline{x})^2}$$
$$f_1 = f(x_1) = \alpha(1 + s(x_1 - \overline{x}))e^{-\beta(x_1-\overline{x})^2} \tag{17}$$

It is useful to introduce the notation (for $i = 1, 2$)

$$Q_i = e^{-\beta(x_i-\overline{x})^2} . \tag{18}$$

The estimate for s is explicitly given by

$$s = \frac{Q_0 f_1 - Q_1 f_0}{Q_1 f_0(x_1 - \overline{x}) - Q_0 f_1(x_0 - \overline{x})} . \tag{19}$$

The are under the peak A can be calculated easily. After the substitution $y = x - \overline{x}$ we have

$$A = \int_{-\infty}^{+\infty} f(x)dx = \alpha \int_{-\infty}^{+\infty} (1 + sy)e^{-\beta y^2} dy .$$

From this we see the extra term with s is odd in y and thus not contribute to A. Just as for the Gaussian profile (cf. SCBD) we obtain

$$A = \alpha\sqrt{\frac{\pi}{\beta}} . \tag{20}$$

Manual of program SCB-MWD

In SCB-MWD the full SCB-MW distributions are analyzed. The distribution is 2-dimensional with an SCB (i.e. temperature) and an Mw (i.e. molecular weight) axis. The axes are discretized using ngrid points, so that the full grid consists of ngrd*ngrid points. The values of the SCB-MW distribution are specified in the input file as follows:

- modulus
- tiechain
- ngrid blocks with per block:
  - ngrid lines with 3 real numbers; the first one is the temperature (constant per block), the second one the Mw value and the third one the value of the distribution in that particular point.

The peaks in the distribution are detected by applying the methods of program SCBD two times. Each block in the input file contains the distribution values of a cross-section parallel to the Mw-axis, thus for constant temperature. For such a cross-section the peaks and their characteristics (position, height, width, area) are estimated and stored. Next, the same procedure is applied to cross-sections parallel to the SCB (i.e. the temperature) axis. Each peak is estimated by an expression of the Gaussian form:

$$F(M,T) = \alpha\ e^{-\beta_M(M-\overline{M})^2-\beta_T(T-\overline{T})^2}$$

with M the mass coordinate and T the temperature. The parameters characterize the peak as follows:

- α: peak height
- β_M: peak width along a cross-section parallel to the T-axis
- β_T: peak width along a cross-section parallel to the M-axis
- $(\overline{M},\overline{T})$: coordinates of the maximum.

The area of a cross-section parallel to the T-axis is given by

$$\text{area } 1 = \alpha\sqrt{\frac{\pi}{\beta_T}}$$

and similarly the area of a cross-section parallel to the M-axis by

$$\text{area } 2 = \alpha\sqrt{\frac{\pi}{\beta_M}}$$

The full SCB-MW distribution is approximated by the sum of Gaussian peaks:

$$SCB\text{–}MW(M,T) = \sum_{i=1}^{\text{nmax}} \alpha(i) e^{-\beta_M(i)(M-\overline{M}(i))^2 - \beta_T(i)(T-\overline{T}(i))^2}$$

The output per peak is: $\overline{M}$, $\overline{T}$, α, β_M, β_T, area 1, area 2. Only the parameters of the three heighest peaks is written on the output file. Extension to more peaks is straightforward.

```
      program  SCBD

c     last update: January 2002
c     author: J.Molenaar, Eindhoven University of Technology

      parameter(ngrid=32)

      real x(ngrid,20)
      real modulus,tiechain
      integer i,n,ntel
      real xmax(5),alfa(5),beta(5),area(5)

c     Open data and output files

      call lees

c     Read the data per block of (ngrid+2) lines

      ntel = 0
      do 10 i = 1, 100000

        ntel = ntel + 1
        read(6,*,end=20) modulus
        read(6,*) tiechain
        write(*,*) 'ntel : ', ntel

      do 60 n = 1, ngrid
        read(6,*) x(n,1),x(n,2),x(n,3),x(n,4)
 60   continue

c     Estimate the maxima and their characteric parameters:

      call maxima(x,xmax,alfa,beta,area)

c     Write modulus, tiechain, and the parameters of the peaks onto
c     one line of the output file.
c     (4 parameters per peak; maximal 3 peaks)

      write(7,70) modulus, tiechain,
     &           xmax(1),alfa(1),beta(1),area(1),
     &           xmax(2),alfa(2),beta(2),area(2),
     &           xmax(3),alfa(3),beta(3),area(3)

  70  format(I8,f10.6,12F8.4)


 10   continue

 20   continue
      write(*,*) 'The total number of SCB distributions is: ', ntel

      stop
      end
c-----------------------------------------------------------------------
      subroutine maxima(x,xmax,alfa,beta,area)

      parameter(ngrid=32)

      real x(ngrid,20)
      integer i,j,nmax,imax(5)
      real x0,x1,x2,f0,f1,f2,Q,xm
```

```
      real xmax(5),alfa(5),beta(5),area(5)
      real hulp,eps,pi

      pi = 3.14159265
      eps = 0.000001

c     Very small or vanishing distibution values may give rise to
c     to numerical problems. That's why they are set at the minimum
c     value eps.

      do 5 j = 1, ngrid
        if (x(j,4).lt.(eps)) x(j,4) = eps
  5   continue

c     First, rough estimates of the peak positions are determined.

      nmax = 0
      do 10 j = 2, ngrid-1
          if ((x(j-1,4).lt.x(j,4)).and.(x(j,4).gt.x(j+1,4))) then
          nmax = nmax + 1
          imax(nmax) = j
        endif
  10  continue

c     The number of peaks is nmax located near gridpoints imax(i).

c     Per peak 4 parameters are estimated:

      do 20 i = 1, nmax

c     If a peak is found, its contribution to the distribution
c     is subtracted. Then it may happen that the the distribution
c     values become small. To avoid numerical errors all values
c     are given the minimum value eps.

      do 25 j = 1, ngrid
        if (x(j,4).lt.(eps)) x(j,4) = eps
  25  continue

c     The peak parameters are estimated from the data
c     points (x0,fo), (x1,f1), and (x2,f2).

      x0 = x(imax(i)-1,1)
      f0 = x(imax(i)-1,4)
      x1 = x(imax(i),1)
      f1 = x(imax(i),4)
      x2 = x(imax(i)+1,1)
      f2 = x(imax(i)+1,4)
      if (f0.le.(eps)) f0 = eps
      if (f1.le.(eps)) f1 = eps
      if (f2.le.(eps)) f2 = eps

c     Evaluation of formula (4) in the manual.

      hulp = dlog(f0/f2)
      if (abs(hulp).lt.(eps)) hulp = eps
      Q = dlog(f0/f1)/hulp

c     Evaluation of formula (5) in the manual.

      xmax(i) = x0*x0 - x1*x1 - Q*(x0*x0 - x2*x2)
```

```
      hulp = x0 - x1 - Q*(x0 - x2)
      if (abs(hulp).lt.(eps)) hulp = eps
      xmax(i) = xmax(i)/hulp/2
      xm = xmax(i)

      if (xm.lt.x(imax(i),1)) imax(i) = imax(i) - 1

c     Evaluation of formula (6) in the manual.

      hulp = ((x0-xm)*(x0-xm)-(x1-xm)*(x1-xm))
      if (abs(hulp).lt.(eps)) hulp = eps
      beta(i) = - dlog(f0/f1)/hulp
      if (abs(beta(i)).lt.(eps)) beta(i) = eps

c     Evaluation of formula (7) in the manual.

      alfa(i) = f0 * dexp(beta(i)*(x0-xm)*(x0-xm))

c     Evaluation of formula (8) in the manual.

      area(i) = alfa(i)*dsqrt(pi/beta(i))


c     The estimated Gaussian peak is subtracted from the data.

      do 40 j = 1, ngrid
        hulp = alfa(i)*dexp(-beta(i)*(x(j,1)-xmax(i))**2)
        x(j,4) = x(j,4) - hulp
 40   continue

 20   continue
      return
      end
c-----------------------------------------------------------------------
      subroutine lees

      character*15 ifn,ofn
      logical xin,xout

      write(*,*) ' Type the name of the input file:    '
      READ(*,*,end = 31) IFN
 31   continue
      inquire(file=IFN,exist=XIN)
      if (XIN) then
        write(*,*) ' The file  "',IFN,'"      indeed exists.'
        open(unit=6,file=IFN,status='OLD')
      ELSE
        write(*,*) ' The file  "',IFN,'"      does not exist. '

      endif


      write(*,*) ' Type the name of the output file:  '

      read(*,*,end = 32) OFN
 32   continue
      inquire(file=OFN,exist=XOUT)
      if (XOUT) then
        write(*,*) ' The file    "',OFN,'"      already exists.'
        open(unit=7,file=OFN,status='OLD')
      ELSE
```

```
   write(*,*) ' The file    "',OFN,'"     does not exist and is
$now created'
   open(unit=7,file=OFN,status='NEW')
 endif

 return
 end
```

```
      program  SCBD_SKEW1

c     last update: January 2002
c     author: J.Molenaar, Eindhoven University of Technology

      parameter(ngrid=32)

      real x(ngrid,20)
      real modulus,tiechain
      integer i,n,ntel
      real xmax(5),alfa(5),beta(5),skewness(5)

c     Open data and output files

      call lees

c     Read the data per block of (ngrid+2) lines

      ntel = 0
      do 10 i = 1, 100000

        ntel = ntel + 1
        read(6,*,end=20) modulus
        read(6,*) tiechain
        write(*,*) 'ntel : ', ntel
c        write(7,*) 'ntel : ', ntel

      do 60 n = 1, ngrid
        read(6,*) x(n,1),x(n,2),x(n,3),x(n,4)
 60   continue

c     Estimate the maxima and their characteric parameters:

      call maxima(x,xmax,alfa,beta,skewness)

c     Write modulus, tiechain, and the parameters of the peaks onto
c     one line of the output file.
c     (4 parameters per peak; maximal 3 peaks)

      write(7,70) modulus, tiechain,
     &           xmax(1),alfa(1),beta(1),skewness(1),
     &           xmax(2),alfa(2),beta(2),skewness(2),
     &           xmax(3),alfa(3),beta(3),skewness(3)

  70  format(I8,f10.6,12F12.4)


 10   continue

 20   continue
      write(*,*) 'The total number of SCB distributions is: ', ntel

      stop
      end
c-----------------------------------------------------------------------
      subroutine maxima(x,xmax,alfa,beta,skewness)

      parameter(ngrid=32)

      real x(ngrid,20)
      integer i,j,nmax,imax(5)
```

```
      real x0,x1,x2,x3,f0,f1,f2,f3
      real xmax(5),alfa(5),beta(5),skewness(5)
      real hulp,eps
      real p1,p2,a0123,b0123,a0213,b0213
      real xm,alf,bet,skew,xstart,xlength

      eps = 0.000001

c     Very small or vanishing distibution values may give rise to
c     to numerical problems. That's why they are set at the minimum
c     value eps.

      do 5 j = 1, ngrid
        if (x(j,4).lt.(eps)) x(j,4) = eps
  5   continue

c     The Weibull distribution (see (1) in the manual) works best if the
c     interval is scaled to [0,1]. This scaling is done here.
c     To avoid numerical errors the interval is taken as [eps,1].

      xstart = x(1,1)
      xlength = x(ngrid,1) - xstart

      do 7 j = 1, ngrid
        x(j,1) = (x(j,1) - xstart) / xlength
  7   continue
      x(1,1) = x(1,1) + eps

c     First, rough estimates of the peak positions are determined.

      nmax = 0
      do 10 j = 2, ngrid-2
          if ((x(j-1,4).lt.x(j,4)).and.(x(j,4).gt.x(j+1,4))) then
          nmax = nmax + 1
          imax(nmax) = j
        endif
  10  continue

c     The number of peaks is nmax; located near gridpoints imax(i).

c     Per peak 4 parameters are estimated:

      do 20 i = 1, nmax

c     If a peak is found, its contribution to the distribution
c     is subtracted in the '40' loop.
c     Then it may happen that the the distribution
c     values become small. To avoid numerical errors all values
c     are given the minimum value eps.

      do 25 j = 1, ngrid
        if (x(j,4).lt.(eps)) x(j,4) = eps
  25  continue

c     The peak parameters are estimated from the data
c     points (x0,f0), (x1,f1), (x2,f2), (x3,f3)

      x0 = x(imax(i)-1,1)
      f0 = x(imax(i)-1,4)
      x1 = x(imax(i),1)
      f1 = x(imax(i),4)
```

```
      x2 = x(imax(i)+1,1)
      f2 = x(imax(i)+1,4)
      x3 = x(imax(i)+2,1)
      f3 = x(imax(i)+2,4)

c      write(7,*) 'x0,x1,x2,x3 : ', x0,x1,x2,x3
c      write(7,*) 'f0,f1,f2,f3 : ', f0,f1,f2,f3

c     Evaluation of formula (4) in the manual.

      p1 = dlog(x2/x3)*dlog(f0/f1) - dlog(x0/x1)*dlog(f2/f3)
      p2 = dlog(x1/x3)*dlog(f0/f2) - dlog(x0/x2)*dlog(f1/f3)

c      write(7,*) 'p1,p2 : ', p1,p2

c     Evaluation of formulae (8) en (9) in the manual.

      a0123 = dlog(x2/x3)*(x0**2-x1**2) - dlog(x0/x1)*(x2**2-x3**2)
      b0123 = -2*dlog(x2/x3)*(x0-x1) + 2*dlog(x0/x1)*(x2-x3)
      a0213 = dlog(x1/x3)*(x0**2-x2**2) - dlog(x0/x2)*(x1**2-x3**2)
      b0213 = -2*dlog(x1/x3)*(x0-x2) + 2*dlog(x0/x2)*(x1-x3)

c      write(7,*) 'a,b : ', a0123,b0123,a0213,b0213

c     Evaluation of formula (12) in the manual.


      hulp = p1*b0213 - p2*b0123
      if (abs(hulp).lt.(eps)) hulp = eps
      xm = (p2*a0123 - p1*a0213)/hulp

c      write(7,*) 'xm : ', xm

c     Evaluation of formula (13) in the manual.

      hulp = a0123 + b0123*xm
      if (abs(hulp).lt.(eps)) hulp = eps
      bet = - p1/hulp
      if (abs(bet).lt.(eps)) bet = eps

c      write(7,*) 'beta : ', bet

c     Evaluation of formula (14) in the manual.

      hulp = dlog(f0/f1) + bet*((x0-xm)**2 - (x1-xm)**2)
      skew = hulp / dlog(x0/x1)
      if (skew.le.0.d0) write(7,*) ' Skew < 0 : Estimation failed.'
      if (skew.le.0.d0) write(*,*) ' Skew < 0 : Estimation failed.'

c      write(7,*) 'skew :', skew

c     Evaluation of formula (15) in the manual.

      alf = f0 * (x0**(-skew)) * dexp(bet*(x0-xm)*(x0-xm))

c      write(7,*) 'alfa :', alf

c     The estimated Gaussian peak is subtracted from the data.

      do 40 j = 1, ngrid
        hulp = alf * (x(j,1)**(skew)) *
```

```
     &  dexp(-bet*(x(j,1)-xm)**2)
        x(j,4) = x(j,4) - hulp
  40  continue

c     The estimates of the parameters are based on the temperature
c     interval [0,1]. The estimates for the original interval are:

      xmax(i) = xstart + xm*xlength
      alfa(i) = alf
      beta(i) = bet / xlength / xlength
      skewness(i) = skew

  20  continue
      return
      end
c-----------------------------------------------------------------------
      subroutine lees

      character*15 ifn,ofn
      logical xin,xout

      write(*,*) ' Type the name of the input file:   '
      read(*,*,end = 31) IFN
  31  continue
      inquire(file=IFN,exist=XIN)
      if (XIN) then
        write(*,*) ' The file  "',IFN,'"     indeed exists.'
        open(unit=6,file=IFN,status='OLD')
      else
        write(*,*) ' The file  "',IFN,'"     does not exist. '

      endif

      write(*,*) ' '
      write(*,*) ' Type the name of the output file:  '

      read(*,*,end = 32) OFN
  32  continue
      inquire(file=OFN,exist=XOUT)
      if (XOUT) then
        write(*,*) ' The file    "',OFN,'"     already exists.'
        open(unit=7,file=OFN,status='OLD')
      else
        write(*,*) ' The file    "',OFN,'"     does not exist and is
     $now created'
        open(unit=7,file=OFN,status='NEW')
      endif

      return
      end
```

```
      program  SCBD_SKEW2

c     last update: April 2002
c     author: J.Molenaar, Eindhoven University of Technology

      parameter(ngrid=32)

      real x(ngrid,20)
      real modulus,tiechain
      integer i,n,ntel
      real xmax(5),alfa(5),beta(5),skew(5),area(5)

c     Open data and output files

      call lees

c     Read the data per block of (ngrid+2) lines

      ntel = 0
      do 10 i = 1, 100000

        ntel = ntel + 1
        read(6,*,end=20) modulus
        read(6,*) tiechain
        write(*,*) 'ntel : ', ntel

      do 60 n = 1, ngrid
        read(6,*) x(n,1),x(n,2),x(n,3),x(n,4)
 60   continue

c     Estimate the maxima and their characteristic parameters:

      call maxima(x,xmax,alfa,beta,skew,area)

c     Write modulus, tiechain, and the parameters of the peaks onto
c     one line of the output file.
c     (5 parameters per peak; maximal 3 peaks)

      write(7,70) modulus, tiechain,
     &           xmax(1),alfa(1),beta(1),skew(1),area(1),
     &           xmax(2),alfa(2),beta(2),skew(2),area(2),
     &           xmax(3),alfa(3),beta(3),skew(3),area(3)

  70  format(I8,f10.6,15F9.4)


 10   continue

 20   continue
      write(*,*) 'The total number of SCB distributions is: ', ntel

      stop
      end
c-----------------------------------------------------------------------
      subroutine maxima(x,xmax,alfa,beta,skew,area)

      parameter(ngrid=32)

      real x(ngrid,20)
      integer i,j,nmax,imax(5)
      real x0,x1,x2,f0,f1,f2,Q,xm
```

```
      real xmax(5),alfa(5),beta(5),skew(5),area(5)
      real hulp,eps,pi,Q0,Q1

      pi = 3.14159265
      eps = 0.000001

c     Very small or vanishing distibution values may give rise to
c     to numerical problems. That's why they are set at the minimum
c     value eps.

      do 5 j = 1, ngrid
        if (x(j,4).lt.(eps)) x(j,4) = eps
  5   continue

c     First, rough estimates of the peak positions are determined.

      nmax = 0
      do 10 j = 2, ngrid-1
          if ((x(j-1,4).lt.x(j,4)).and.(x(j,4).gt.x(j+1,4))) then
          nmax = nmax + 1
          imax(nmax) = j
        endif
  10  continue

c     The number of peaks is nmax located near gridpoints imax(i).

c     Per peak 4 parameters are estimated:

      do 20 i = 1, nmax

c     If a peak is found, its contribution to the distribution
c     is subtracted. Then it may happen that the the distribution
c     values become small. To avoid numerical errors all values
c     are given the minimum value eps.

      do 25 j = 1, ngrid
        if (x(j,4).lt.(eps)) x(j,4) = eps
  25  continue

c     The peak parameters are estimated from the data
c     points (x0,fo), (x1,f1), and (x2,f2).

      x0 = x(imax(i)-1,1)
      f0 = x(imax(i)-1,4)
      x1 = x(imax(i),1)
      f1 = x(imax(i),4)
      x2 = x(imax(i)+1,1)
      f2 = x(imax(i)+1,4)
      if (f0.le.(eps)) f0 = eps
      if (f1.le.(eps)) f1 = eps
      if (f2.le.(eps)) f2 = eps

c     Evaluation of formula (4) in the manual of SCBD.

      hulp = dlog(f0/f2)
      if (abs(hulp).lt.(eps)) hulp = eps
      Q = dlog(f0/f1)/hulp

c     Evaluation of formula (5) in the manual of SCBD.

      xmax(i) = x0*x0 - x1*x1 - Q*(x0*x0 - x2*x2)
```

```
      hulp = x0 - x1 - Q*(x0 - x2)
      if (abs(hulp).lt.(eps)) hulp = eps
      xmax(i) = xmax(i)/hulp/2
      xm = xmax(i)

      if (xm.lt.x(imax(i),1)) imax(i) = imax(i) - 1

c     Evaluation of formula (6) in the manual of SCBD.

      hulp = ((x0-xm)*(x0-xm)-(x1-xm)*(x1-xm))
      if (abs(hulp).lt.(eps)) hulp = eps
      beta(i) = - dlog(f0/f1)/hulp
      if (abs(beta(i)).lt.(eps)) beta(i) = eps

c     Evaluation of formula (7) in the manual of SCBD.

      alfa(i) = f0 * dexp(beta(i)*(x0-xm)*(x0-xm))

c     Evaluation of formula (8) in the manual of SCBD.

      area(i) = alfa(i)*dsqrt(pi/beta(i))

c     Evaluation of formula (19) in the manual of SCBD-SKEW2.


      Q0 = dexp(beta(i)*(x0-xm)*(x0-xm))
      Q1 = dexp(beta(i)*(x2-xm)*(x2-xm))
      hulp = Q1*f0*(x1-xm) - Q0*f1*(x0-xm)
      skew(i) = (Q0*f1 - Q1*f0)/hulp


c     The estimated Gaussian peak is subtracted from the data.

      do 40 j = 1, ngrid
        hulp = alfa(i)*dexp(-beta(i)*(x(j,1)-xmax(i))**2)
        hulp = hulp * (1+skew(i)*(x(j,1)-xmax(i))**2)
        x(j,4) = x(j,4) - hulp
 40   continue

 20   continue
      return
      end
c-----------------------------------------------------------------------
      subroutine lees

      character*15 ifn,ofn
      logical xin,xout

      write(*,*) ' Type the name of the input file:    '
      READ(*,*,end = 31) IFN
 31   continue
      inquire(file=IFN,exist=XIN)
      if (XIN) then
        write(*,*) ' The file  "',IFN,'"      indeed exists.'
        open(unit=6,file=IFN,status='OLD')
      ELSE
        write(*,*) ' The file  "',IFN,'"      does not exist. '

      endif
```

```
      write(*,*) ' Type the name of the output file:  '

      read(*,*,end = 32) OFN
32    continue
      inquire(file=OFN,exist=XOUT)
      if (XOUT) then
        write(*,*) ' The file     "',OFN,'"      already exists.'
        open(unit=7,file=OFN,status='OLD')
      ELSE
        write(*,*) ' The file     "',OFN,'"      does not exist and is
     $now created'
        open(unit=7,file=OFN,status='NEW')
      endif

      return
      end
```

```
      program  SCB_MWD

c     last update: January 2002
c     author: J.Molenaar, Eindhoven University of Technology

      parameter(ngrid=32)

      real x(ngrid,2),y(ngrid,2),xy(ngrid,5),uit(5,4)
      real modulus,tiechain,temp
      integer i,nx,ny,ntel,nmax
      real xmax(5),xalfa(5),xbeta(5),xarea(5)
      real ymax(5),yalfa(5),ybeta(5),yarea(5)
      real tempstap

c     Open the input and output files:

      call lees

c     Read the data: first two lines for modulus and tiechain.
c     Then ngrid blocks of ngrid lines for the SCB-MW data.

      ntel = 0
      do 10 i = 1, 100000
        ntel = ntel + 1
        read(6,*,end=20) modulus
        read(6,*) tiechain
        write(*,*) 'ntel : ', ntel

      do 55 nx = 1, ngrid

      do 60 ny = 1, ngrid
        read(6,*) temp,y(ny,1),y(ny,2)
 60   continue

c     Estimate per constant-temperature cross-section the maxima.

      call maxima(y,ymax,yalfa,ybeta,yarea,nmax)

      xy(nx,1)= ymax(1)
      xy(nx,2)= yalfa(1)
      xy(nx,3)= ybeta(1)
      xy(nx,4)= yarea(1)
      xy(nx,5) = temp

      x(nx,1) = temp
      x(nx,2) = yalfa(1)

  55  continue

c     Estimate per constant-mass cross-section the maxima.

      call maxima(x,xmax,xalfa,xbeta,xarea,nmax)


      do 75 nx = 1,5
      do 77 ny = 1,4
        uit(nx,ny) = 0.0
  77  continue
  75  continue

c     Couple the maxima found in the cross-sections.
```

```
      tempstap = (xy(2,5)-xy(1,5))/2
      do 80 nx = 1,nmax
      do 85 ny = 1,ngrid
        if (dabs(xmax(nx) - xy(ny,5)).le.tempstap) then
        uit(nx,1) = xy(ny,1)
        uit(nx,2) = xy(ny,2)
        uit(nx,3) = xy(ny,3)
        uit(nx,4) = xy(ny,4)
        endif
   85 continue
   80 continue

c     The peak parameters are written onto one line of the output file
c     in the order of decreasing peak height.
c     Per peak 7 parameters are written (at most three maxima).

      if ((xalfa(3).le.xalfa(1)).and.(xalfa(3).le.xalfa(2))) then
      write(7,70) modulus, tiechain,
     &      xmax(1),xalfa(1),xbeta(1),xarea(1),
     &      uit(1,1),uit(1,3),uit(1,4),
     &      xmax(2),xalfa(2),xbeta(2),xarea(2),
     &      uit(2,1),uit(2,3),uit(2,4)
      goto 86
      endif

      if ((xalfa(2).le.xalfa(1)).and.(xalfa(2).le.xalfa(3))) then
      write(7,70) modulus, tiechain,
     &      xmax(1),xalfa(1),xbeta(1),xarea(1),
     &      uit(1,1),uit(1,3),uit(1,4),
     &      xmax(3),xalfa(3),xbeta(3),xarea(3),
     &      uit(3,1),uit(3,3),uit(3,4)
      goto 86
      endif

      if ((xalfa(1).le.xalfa(2)).and.(xalfa(1).le.xalfa(3))) then
      write(7,70) modulus, tiechain,
     &      xmax(2),xalfa(2),xbeta(2),xarea(2),
     &      uit(2,1),uit(2,3),uit(2,4),
     &      xmax(3),xalfa(3),xbeta(3),xarea(3),
     &      uit(3,1),uit(3,3),uit(3,4)
      goto 86
      endif


   70 format(I8,f10.6,14F8.4)
   86 continue

   10 continue

   20 continue
      write(*,*) ' The total number of distributions is ',ntel

      stop
      end
c-----------------------------------------------------------------------
      subroutine maxima(x,xmax,alfa,beta,area,nmax)

      parameter(ngrid=32)

      real x(ngrid,2)
```

```
      integer i,j,nmax,imax(5)
      real x0,x1,x2,f0,f1,f2,Q,xm
      real xmax(5),alfa(5),beta(5),area(5)
      real hulp

      do 5 j = 1, ngrid
        if (x(j,2).lt.(0.000001)) x(j,2) = 0.000001
  5   continue

      do 7 j = 1, 5
        xmax(j) = 0.0
        alfa(j) = 0.0
        beta(j) = 0.0
        area(j) = 0.0
  7    continue


      nmax = 0
      do 10 j = 2, 31
          if ((x(j-1,2).lt.x(j,2)).and.(x(j,2).gt.x(j+1,2))) then
          nmax = nmax + 1
          imax(nmax) = j
        endif
  10  continue

c      write(*,*)'discrete posities maxima: ',(imax(j),j=1,nmax)

      if (nmax.eq.0) then
        alfa(1) = 0.0
        beta(1) = 0.0
        area(1) = 0.0
        xmax(1) = 16
        return
      endif

      do 20 i = 1, nmax

      x0 = x(imax(i)-1,1)
      f0 = x(imax(i)-1,2)
      x1 = x(imax(i),1)
      f1 = x(imax(i),2)
      x2 = x(imax(i)+1,1)
      f2 = x(imax(i)+1,2)
      if (f0.le.(0.000001)) f0 = 0.000001
      if (f1.le.(0.000001)) f1 = 0.000001
      if (f2.le.(0.000001)) f2 = 0.000001

c        write(*,*) 'x0 etc:',x0,f0,x1,f1,x2,f2

      hulp = dlog(f0/f2)
      if (abs(hulp).lt.(0.000001)) hulp = 0.000001
      Q = dlog(f0/f1)/hulp
      xmax(i) = x0*x0 - x1*x1 - Q*(x0*x0 - x2*x2)
      hulp = x0 - x1 - Q*(x0 - x2)
      if (abs(hulp).lt.(0.000001)) hulp = 0.000001
      xmax(i) = xmax(i)/hulp/2
      xm = xmax(i)
      if (xm.lt.x(imax(i),1)) imax(i) = imax(i) - 1
      hulp = ((x0-xm)*(x0-xm)-(x1-xm)*(x1-xm))
      if (abs(hulp).lt.(0.000001)) hulp = 0.000001
      beta(i) = - dlog(f0/f1)/hulp
```

```
      alfa(i) = dexp(dlog(f0) + beta(i)*(x0-xm)*(x0-xm))
      if (abs(beta(i)).lt.(0.000001)) beta(i) = 0.000001
      area(i) = alfa(i)*dsqrt(3.14159265/beta(i))



      do 40 j = 1, ngrid
        hulp = alfa(i)*dexp(-beta(i)*(x(j,1)-xmax(i))**2)
        x(j,2) = x(j,2) - hulp
   40 continue

   20 continue
      return
      end
c-----------------------------------------------------------------------
      subroutine lees

      character*15 ifn,ofn
      logical xin,xout

      write(*,*) ' Type the name of the input file:    '
      READ(*,*,end = 31) IFN
   31 continue
      inquire(file=IFN,exist=XIN)
      if (XIN) then
        write(*,*) ' The file  "',IFN,'"     indeed exists.'
        open(unit=6,file=IFN,status='OLD')
      ELSE
        write(*,*) ' The file  "',IFN,'"     does not exist. '

      endif


      write(*,*) ' Type the name of the output file:  '

      read(*,*,end = 32) OFN
   32 continue
      inquire(file=OFN,exist=XOUT)
      if (XOUT) then
        write(*,*) ' The file    "',OFN,'"     already exists.'
        open(unit=7,file=OFN,status='OLD')
      ELSE
        write(*,*) ' The file    "',OFN,'"     does not exist and is
     $now created'
        open(unit=7,file=OFN,status='NEW')
      endif

      return
      end
```