

SORPTIONSISOTERMER

Program- og brugerdokumentation for programmerne
DATAIND, SORPF, DESORPF og UDTEGN fra disketten SORPTION

Kurt Kielsgaard Hansen



THE TECHNICAL UNIVERSITY OF DENMARK
DEPARTMENT OF CIVIL ENGINEERING
BUILDING MATERIALS LABORATORY

SORPTIONSISOTERMER

**Program- og brugerdokumentation for programmerne
DATAIND, SORPF, DESORPF og UDTEGN fra disketten SORPTION**

Kurt Kielsgaard Hansen

December 1986

RESUME

Denne rapport supplerer Teknisk Rapport 162/86, "Sorption Isotherms. A Catalogue", reference /1/, idet alle de i /1/ viste sorptionsisotermer er udtegnet med de her beskrevne programmer.

Med DATAIND læses et materiales sorptionsværdier (ϕ, u) ind på en fil. Indlæsningen kan ske via PC'ens tastatur eller ved digitizing. Digitizing er anvendeligt, når sorptionsværdierne kun findes som datapunkter i figurerne i litteraturen.

Curve fitting programmerne DESORPF og SORPF til henholdsvis de- og adsorptionsværdier anvendes derefter til at bestemme den bedste kurve gennem punkterne. Curve fitting resultatet indsættes manuelt via tastatur i materialefilen.

Med UDTEGN optegnes materialets katalogside i A3-format på plotter. Katalogudtegningen viser både dataværdier m.m. og en figur med datapunkter og tilnærmede kurver.

SUMMARY

This report is a supplement to Teknisk Rapport 162/86, "Sorption Isotherms. A Catalogue", Reference /1/, as all the sorption isotherms shown in /1/ are made using the programs described in the present report. The programs are developed for an IBM Personal Computer connected to an HP Graphics Plotter 7475A, and are stored on a diskette marked SORPTION.

The sorption values (ϕ, u) can be read into a file on the diskette by means of the DATAIND program, which has two possibilities for input:

- using the HP Graphics Plotter as a digitizer,
- via the keyboard of the PC.

Digitizing can be used when the sorption values are shown only as data points in figures in the literature. Examples of the two possibilities for input are given in Chapters 2.2 and 2.3, respectively.

The curve fitting programs DESORPF and SORPF for desorption and adsorption values, respectively, may then be used to fit the best curve through the points using the equation

$$u = c_1 \cdot \exp((-1/c_2) \cdot \ln(1 - \ln(\phi)/c_3))$$

Here c_1 is the u -value for curve intersect with $\phi = 100\%$ RH, while c_2 and c_3 form the curve (Chapter 3). The DESORPF program gives proposals for a start guess for c_1 , c_2 , and c_3 , which may be used by pressing \leftarrow . The SORPF program is mainly for adsorption values, but also difficult desorption values may be run. The program is a reduced form of DESORPF as only two variables are fitted (c_2 and c_3), while c_1 is iterated for minimum sum of square deviations. It is important that the connected printer is on line running the DESORPF and SORPF programs as the results c_1 , c_2 , and c_3 are printed here. Examples running the two programs are given in Chapters 3.2 and 3.3, respectively. The values of c_1 , c_2 , and c_3 for each sorption type must be inserted manually via the keyboard in their respective places in the material file.

Using the UDTEGN program, the catalogue page of the material is plotted in A3-format on the plotter. The plotted page shows both data values and a figure with data points and fitted curves.

FORORD

Ved Laboratoriet for Bygningsmaterialer indgår emnet fugt-binding og fugtvandring i bygningsmaterialer i det langsigtede forskningsprogram. Den her foreliggende rapport redigerer for opbygningen af den database, som de kommende fugtvandringsberegninger skal støtte sig til.

Programmerne er redigeret af lic.techn. Kurt Kielsgaard Hansen. Programmerne DATAIND og UDETEGN er skrevet i BASIC med hjælp fra civ.ing. Lene Rathkjen. Programmerne SORPF og DESORPF er skrevet i TURBO PASCAL med hjælp fra civ.ing. Lars Mouritsen.

Programmerne er udviklet på IBM Personal Computer med tilkoblet HP Graphics Plotter 7475A, og de findes lagret på 5 1/4" diskette mrk. SORPTION. På samme diskette findes materialefiler for de mere end 100 materialer, der indgår i kataloget. LBM's IBM PC er suppleret med 8087 matematikprocesser, der gør udførelsen af PASCAL-programmerne betydeligt hurtigere.

Arbejdet indgår i STVF-projektet "Fugt i byggematerialer" (j.nr. 16-3722.B-153), som er udført ved LBM i 1985-86.

Anders Nielsen
Projektleder

December 1986

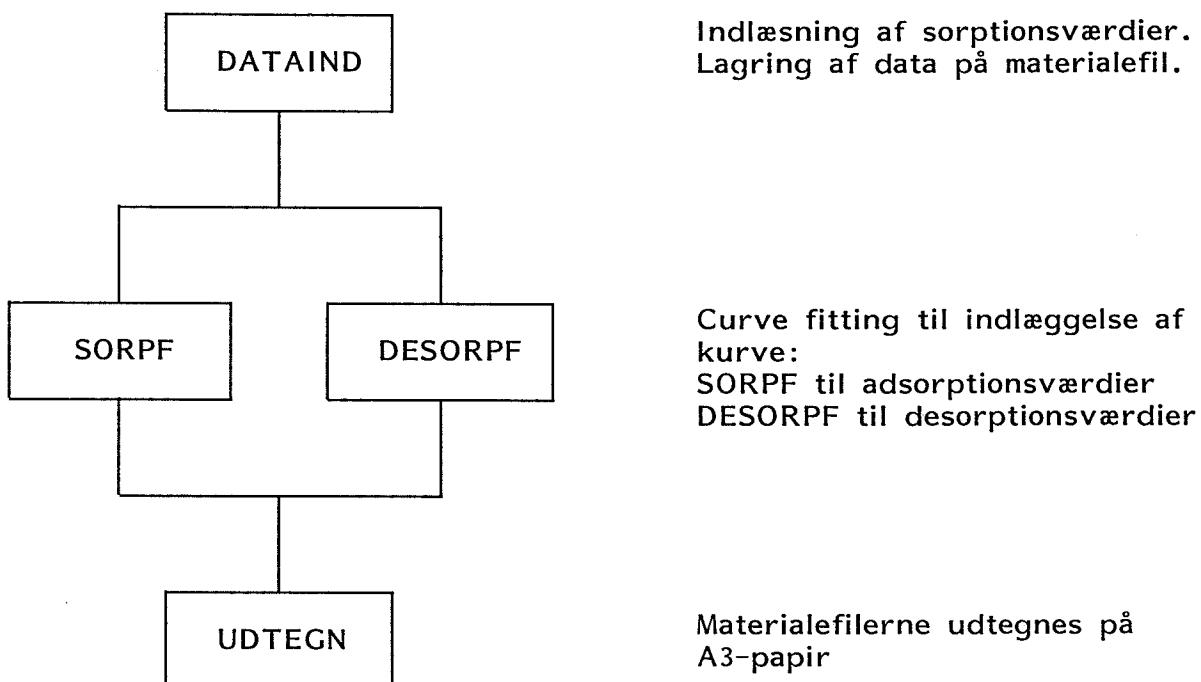
INDHOLD

Side

1.	Database for sorptionsisotermer. Sorptionsisotermkatalog	1
2.	Indlæsning af sorptionsværdier	2
2.1	Indlæseprogrammet DATAIND	2
2.2	Eksempel med digitizing	3
2.3	Eksempel med indlæsning via tastatur	3
3.	Curve fitting til indlæggelse af bedste kurve gennem punkterne ...	8
3.1	Curve fitting med ikke-lineære udtryk	8
3.2	Eksempel med DESORPF til desorptionsværdier	9
3.3	Eksempel med SORPF til adsorptionsværdier	9
3.4	Resultatets indsættelse i materialefil	9
4.	Udtegning af katalogside v.h.a. UDTEGN	14
5.	Afslutning	14
6.	Referencer	14
Appendix A.	Udskrift af BASIC-programmet DATAIND	16
Appendix B.	Udskrift af PASCAL-programmet DESORPF	19
Appendix C.	Udskrift af PASCAL-programmet SORPF	26
Appendix D.	Udskrift af BASIC-programmet UDTEGN	33
Appendix E.	Udskrift af de på SORPTION værende filer	40

1. Database for sorptionsisotermer. Sorptionsisotermkatalog

En oversigt over programmerne til indlæsning af sorptionsværdier (ϕ, u) og udtegning af værdier og kurver er vist nedenfor. ϕ er relativ fugtighed (RF), og u er vand-tørstof-forhold (fugtbrøk, vandindhold i vægtprocent).



Med DATAIND indlæses sorptionsværdierne for et materiale på en fil. Derefter læses de indlæste sorptionsværdier med curve fitting programmerne SORPF og DESORPF, der indlægger den bedste kurve gennem punkterne af formen $u = c_1 \cdot \exp((-1/c_2) \cdot \ln(1-\ln(\phi)/c_3))$, idet c_1 , c_2 og c_3 bestemmes. c_1 er u -værdien for kurvens skæring med $\phi = 100\%$ RF, mens c_2 og c_3 bestemmer kurveformen. Konstanterne c_1 , c_2 og c_3 indsættes manuelt på deres respektive pladser i materialefilen, og UDTEGN kan udtegne det pågældende materiales side i kataloget (A3-side). c_1 , c_2 og c_3 anvendes som en fælles betegnelse for konstanterne i ovennævnte udtryk; i det følgende anvendes også d_1 , d_2 og d_3 for desorptionskonstanterne og a_1 , a_2 og a_3 for adsorptionskonstanterne.

Dette gøres for de i litteraturen værende materialer, og et sorptionsisotermkatalog fremkommer. Dataværdier m.m. findes på diskette for eventuel senere viderebearbejdning, og katalogudtegningen viser både talværdier m.m. og en figur med datapunkter og tilnærmede kurver.

Programkomplekset er delt op på denne måde, fordi især curve fittingen kan drille. Dårlige startgæt for konstanterne kan føre til ustabilitet og dermed ingen bestemmelse af konstanterne; det er derfor gjort nemt at køre programmet igen med formodet bedre startgæt.

Arbejdet med curve fittingen har vist, at slutresultatet afhænger lidt af de anvendte startgæt; dog ikke mere end, at de optegnede regressionskurver visuelt synes identiske. Arbejdet har også vist, at ikke alle måleresultater kan tilnærmes med det anvendte udtryk; det gælder især, hvor u-værdierne stiger kraftigt for høje ϕ .

2. Indlæsning af sorptionsværdier

Sorptionsværdierne for hvert materiale indlæses på en fil (et program, der i dette tilfælde kun indeholder data), der overføres til en diskette for lagring og for senere curve fitting og udtegning. Dette afsnit indeholder kun beskrivelse af indlæseprogrammet; programmerne til curve fitting og udtegning omtales i senere afsnit. Generelt gælder det for alle programmerne, at de er forsøgt skrevet på dialogisk form, så programmet spørger om inddata, hvorefter programmøren svarer ved indtastning efterfulgt af tryk på \leftarrow . Formen på inddata simplificeres på denne måde samtidig med, at alle inddata kommer med i den rigtige rækkefølge.

2.1 Indlæseprogrammet DATAIND

IBM Personal Computer (i det følgende kun PC) tændes, PC-DOS (operativsystem) og BASIC indlæses fra IBM-diskette. Disketten SORPTION med indlæseprogram, programmer til curve fitting og udtegning af sorptionsværdier indsættes i venstre disketteenhed. Der skrives LOAD" DATAIND, og med et tryk på \leftarrow indlæses programmet.

Efter indtastning af RUN og \leftarrow er programmet nu klart til indlæsning af sorptionsværdier. Indlæseprogrammet, der er i BASIC, er konstrueret, så inddata kan indtastes via PC'ens tastatur eller via digitizing.

Indtastning via tastatur anvendes primært, når sorptionsværdierne foreligger i resultattabel.

Ved digitizing fæstes figurerne med sorptionsværdierne i HP plotterens papirholder, der således fungerer som digitizerbord. Efter skalering føres digitizerøjet til første sorptionsværdipunkt (ϕ, u) på figuren. Ved

tryk på ENTER på HP plotteren sendes punktets koordinater til PC'en. Digitizerøjet føres ved tryk på knapperne på HP plotteren efter tur til de øvrige punkter på figuren, hvis koordinater efter tur med tryk på ENTER føres til PC'en.

Fordelen ved digitizing er, at alle figurens punkter på en nem måde føres korrekt over i PC'en, hvis skaleringen er korrekt. Ved konventionel indlæsning via tastatur kræves først udmåling af koordinater for sorptionsværdipunkterne på figuren. Herefter skal koordinaterne (ϕ, u) til hvert punkt indtastes. Fejlmulighederne ved indlæsning via tastatur er generelt større end ved indlæsning via digitizing.*

I det følgende gengives dialogen mellem PC og programmør for et eksempel med digitizing og et eksempel med indtastning. Maksimalt 20 punkter må indlæses for hver sorptionstype.

2.2 Eksempel med digitizing

Indlæseprogrammet er lavet til konfigurationen vist på figur 1. Den består af IBM PC og HP 7475A Graphics Plotter som digitizerbord. Kablet mellem PC og HP 7475A er et RS-232-C kabel, der på Laboratoriet for Bygningsmaterialer er sat sammen af to kabler: SLIDE-kabel og fladt mangefarvet multikabel.

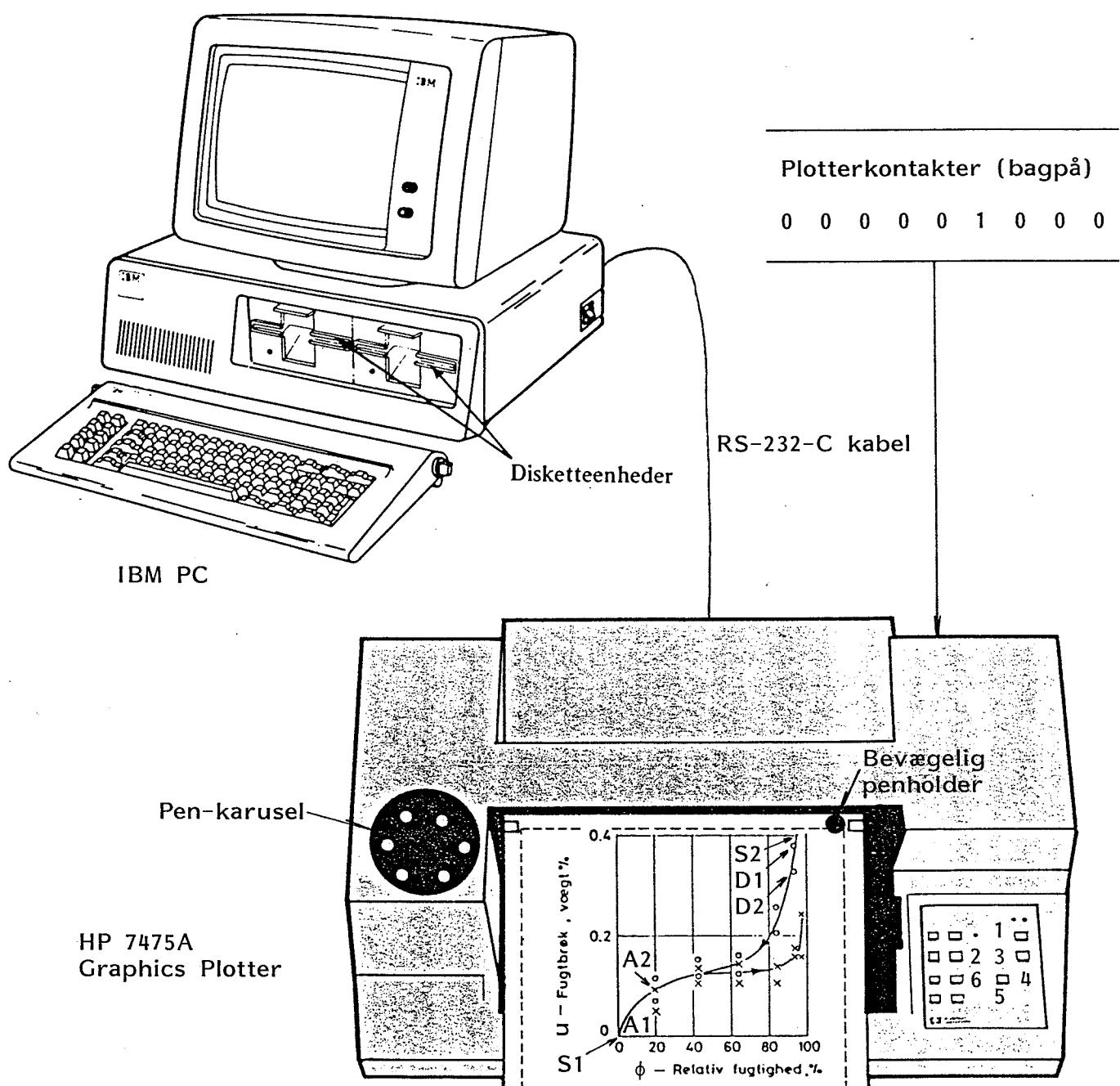
Når digitizerøjet er indsat i den bevægelige penholder (ikke i karusellen!), og indlæseprogrammet er indlæst på PC'en, kan programmet køre ved RUN $\leftarrow \rightarrow$. I figur 2 vises linie for linie, hvad der fremkommer på PC'ens skærm, og hvad programmøren skal indtaste.

De indlæste data er nu lagret under filnavnet EXPCLA20.910 på disketten. De kan kaldes frem på skærmen ved hjælp af TYPE EXPCLA20.910 $\leftarrow \rightarrow$. Et print på papir med PRINT EXPCLA20.910 $\leftarrow \rightarrow$ ser ud som vist på figur 3.

2.3 Eksempel med indlæsning via tastatur

Indlæsningen kan ske via PC'ens tastatur alene. I figur 4 vises linie for linie, hvad der fremkommer på PC'ens skærm, og hvad programmøren skal indtaste.

* Hvis der i litteraturreferencen er angivet flere målinger af u for samme ϕ , er u -værdierne korrekt indlæst, men en lille smule forskudt omkring den korrekte ϕ -værdi. Dette skyldes, at vi i starten arbejdede med en polynomial tilnærmelse til punkterne, der ikke tillod flere u -værdier for samme ϕ . I praksis har ovennævnte forskydning ingen betydning.



Knapper på HP 7475A frontpanel:

- 1: Digitizerøje (PEN) op fra eller ned på papir
- 2: ENTER
- 3-6: Digitizerøjet flyttes med disse knapper

Figur 1. Konfiguration til indlæsning af sorptionsværdier (ϕ, u) fra figur via digitizing.

The arrangement for digitizing the sorption values (ϕ, u) into a file from a figure.

PC-skærm

RUN
INDLÆS MATERIALENAVN (MAX 20 KARAKTERER)
? EXPANDED CLAY
INDLÆS DENSITET, TEMPERATUR (HVIS EJ MÅLT, INDLÆS 0)
? 910,20
ØNSKES SORPTIONSDATA INDLÆST FRA TASTATUR (1)
ELLER VHA DIGITIZING (2) ? 2
INDLÆSNING AF TO PUNKTER TIL SKALERING AF SYSTEMET
DIGITIZE PUNKT (PHI,U)=(0,0)
INDLÆS KENDT PHI, U ? 100,0.6
DIGITIZE PUNKTET
ANTAL KURVER ? 2
KURVE NR. 1
SORPTIONSISOTERMTYPE (DESORPTION (1), ADSORPTION (2),
SCANNING (3), UKENDT (4)) ? 1
ANTAL MÅLEPUNKTER ? 12
DIGITIZE PUNKT NR 1
DIGITIZE PUNKT NR 2
DIGITIZE PUNKT NR 3
DIGITIZE PUNKT NR 4
DIGITIZE PUNKT NR 5
DIGITIZE PUNKT NR 6
DIGITIZE PUNKT NR 7
DIGITIZE PUNKT NR 8
DIGITIZE PUNKT NR 9
DIGITIZE PUNKT NR 10
DIGITIZE PUNKT NR 11
DIGITIZE PUNKT NR 12
KURVE NR. 2
SORPTIONSISOTERMTYPE (DESORPTION (1), ADSORPTION (2),
SCANNING (3), UKENDT (4)) ? 2
ANTAL MÅLEPUNKTER ? 12
DIGITIZE PUNKT NR 1
DIGITIZE PUNKT NR 2
DIGITIZE PUNKT NR 3
DIGITIZE PUNKT NR 4
DIGITIZE PUNKT NR 5
DIGITIZE PUNKT NR 6
DIGITIZE PUNKT NR 7
DIGITIZE PUNKT NR 8
DIGITIZE PUNKT NR 9
DIGITIZE PUNKT NR 10
DIGITIZE PUNKT NR 11
DIGITIZE PUNKT NR 12
YDERLIGERE MATERIALE EGENSKABER (3 LINIER A MAX 50 KARAKTERER)
Open porosity 62%.

KILDEANGIVELSE (3 LINIER A MAX 50 KARAKTERER)
Ahlgren, Lennart: Moisture fixation in porous
building materials. Div. of Build. Techn., Lund
Inst. of Techn. Report 36, Lund, Sweden, 1972.
DATO DD,MM,YY, INITIALER ? 15,04,86,KKH
HVILKET FILNAVN ØNSKES DATA GEMT UNDER ?
? EXPCLA20.910

Programmør

RUN ↵
Materialenavn ↵
Densitet, temperatur ↵
2 ↵
Digitizerøje på S1. ENTER
100,0.6 ↵ (52)
Digitizerøje på S2. ENTER
2 ↵
1* ↵
12 ↵
Digitizerøje på D1. ENTER
Digitizerøje på D2. ENTER
o.s.v.
*) Desorptionsværdier
indlæses med
faldende φ.
2** ↵
12 ↵
Digitizerøje på A1. ENTER
Digitizerøje på A2. ENTER
o.s.v.
**) Adsorptionsværdier
indlæses med stigende φ.
Evt. bemærkninger ↵

1. linie ↵
2. linie ↵
3. linie ↵
dato og initialer ↵
Filnavn ↵

Figur 2. Eksempel med indlæsning af sorptionsværdier (ϕ, u) fra figur via digitizing. DATAIND.

Example of digitizing the sorption values (ϕ, u) from a figure. DATAIND.

EXPANDED CLAY

910	20										
2											
1	12										
98.13	98.50	94.71	95.42	84.51	85.24	65.18	65.56	43.85	43.51	20.27	20.55
0.53	0.43	0.38	0.33	0.26	0.21	0.16	0.12	0.15	0.12	0.11	0.07
	d1		d2		d3						
2	12										
20.55	20.07	43.74	44.08	65.41	64.72	84.88	84.56	95.33	94.60	97.92	98.66
0.05	0.10	0.11	0.14	0.11	0.14	0.11	0.14	0.16	0.19	0.16	0.24
	a1		a2		a3						
Open porosity 62%.											

Ahlgren, Lennart: Moisture fixation in porous
building materials. Div. of Build. Techn., Lund
Inst. of Techn. Report 36, Lund, Sweden, 1972.
3 10 85 KKH

Note: d1, d2 og d3 bestemmes med DESORPF og indsættes manuelt via tastatur.
a1, a2 og a3 bestemmes med SORPF og indsættes manuelt via tastatur.

Figur 3. Et print af materialefilen EXPCLA20.910 umiddelbart efter
indlæsning.

A print of the material file EXPCLA20.910 just after input. d1, d2, and d3
may be determined by DESORPF and inserted manually via the keyboard. a1,
a2, and a3 may be determined by SORPF and inserted manually via the keyboard.

PC-skærm

- 7 -

Programmør

RUN
 INDLÆS MATERIALENAVN (MAX 20 KARAKTERER)
? BIRCH
 INDLÆS DENSITET, TEMPERATUR (HVIS EJ MALT, INDLÆS 0)
? 600,20
ØNSKES SORPTIONSDATA INDLÆST FRA TASTATUR (1)
ELLER VHA DIGITIZING (2) ? 1
ANTAL KURVER ? 2
KURVE NR 1
SORPTIONSISOTERMTYPE (DESORPTION (1), ADSORPTION (2),
SCANNING (3), UKENDT (4)) ? 1
ANTAL MÅLEPUNKTER ? 6
MÅLEPUNKT NR 1 (PHI,U)=
? 94.5,29.3
MÅLEPUNKT NR 2 (PHI,U)=
? 89.9,25.4
MÅLEPUNKT NR 3 (PHI,U)=
? 79.3,20.3
MÅLEPUNKT NR 4 (PHI,U)=
? 64.3,15.5
MÅLEPUNKT NR 5 (PHI,U)=
? 44.7,10.9
MÅLEPUNKT NR 6 (PHI,U)=
? 19.8,5.8
KURVE NR 2
SORPTIONSISOTERMTYPE (DESORPTION (1), ADSORPTION (2),
SCANNING (3), UKENDT (4)) ? 2
ANTAL MÅLEPUNKTER ? 6
MÅLEPUNKT NR 1 (PHI,U)=
? 20.1,4.9
MÅLEPUNKT NR 2 (PHI,U)=
? 43.2,8.2
MÅLEPUNKT NR 3 (PHI,U)=
? 65.1,11.6
MÅLEPUNKT NR 4 (PHI,U)=
? 84.6,17.5
MÅLEPUNKT NR 5 (PHI,U)=
? 94.8,25.2
MÅLEPUNKT NR 6 (PHI,U)=
? 98.0,31.6
YDERLIGERE MATERIALE EGENSKABER (3 LINIER A MAX 50 KARAKTERER)

Evt. bemærkninger ↵

KILDEANGIVELSE (3 LINIER A MAX 50 KARAKTERER)
Ahlgren, Lennart: Moisture fixation in porous
building materials. Div. of Build. Techn., Lund
Inst. of Techn. Report 36, Lund, Sweden, 1972.
DATO DD,MM,YY, INITIALER ? 15,04,86,KKH
HVILKET FILNAVN ØNSKES DATA GEMT UNDER ?
? birch020.600

1. linie ↵
2. linie ↵
3. linie ↵
Dato og initialer ↵

Filnavn ↵

Figur 4. Eksempel med indlæsning af sorptionsværdier (ϕ, u) via PC'ens tastatur. Datasættene foreligger på separat liste som forsøgsprotokol eller efter manuel opmåling på en figur.

Example of putting the sorption values (ϕ, u) into a file via the keyboard of the PC. DATAIND.

3. Curve fitting til indlæggelse af bedste kurve gennem punkterne

Til curve fitting af desorptionsværdierne anvendes DESORPF, der med gode startgæt direkte kan bestemme konstanterne c_1 , c_2 og c_3 i udtrykket $u = c_1 \cdot \exp((-1/c_2) \cdot \ln(1 - \ln(\phi)/c_3))$. c_1 er u -værdien for kurvens skæring med $\phi = 100\%$ RF, mens c_2 og c_3 bestemmer kurveformen. Programmet er opbygget med forslag til startgæt for c_1 , c_2 og c_3 , der kan anvendes ved tryk på \leftarrow . Programmets løsningsmetode er skitseret i næste afsnit.

Til curve fitting af adsorptionsværdier, vanskelige desorptionsværdier og eventuelle "ukendte" sorptionsværdier (litteraturreferencen angiver ikke, om det er de- eller adsorptionsværdier) anvendes SORPF. Programmet er en reduceret version af DESORPF, idet kun to variable fittes (c_2 og c_3), mens c_1 itereres for at opnå minimum sum af afvigelsernes kvadrat (dev). Iterationen udføres med en variant af "Golden section" metoden [2]. Metoden skaber ud fra startgættet for c_1 et interval omkring c_1 , der gradvis indsnævres afhængig af dev. Startgættene for c_2 og c_3 kan skønnes ud fra allerede optegnede katalogsider.

Det er vigtigt, at den tilkoblede printer er On Line under udførelsen af DESORPF og SORPF, da resultatet for c_1 , c_2 og c_3 udskrives her.

Når c_1 , c_2 og c_3 er fundet, indsættes de manuelt via tastatur for hver sorptionstype på deres respektive pladser i materialefilen.

Efter næste afsnit gengives i figur 6 og i figur 7 dialogen mellem PC og programmør for et eksempel med DESORPF og et eksempel med SORPF. DESORPF og SORPF er PASCAL-programmer.

3.1 Curve fitting med ikke-lineære udtryk

Princip og algoritme til curve fitting med ikke-lineære udtryk er omtalt i Numerisk Institut, DTH, hæfte 20 "CURVE-FITTING". Her betragtes en reel funktion $y = f(x, c_1, c_2, \dots, c_m)$, som er differentiabel med hensyn til samtlige variable, men ikke nødvendigvis lineær med hensyn til c' erne.

Problem: Der er givet n talpar (x_i, y_i) . Find konstanter c_1, \dots, c_m , så f approksimerer de n talpar, således at

$$\text{dev} = \sum_{i=1}^n (y_i - f(x_i, c_1, \dots, c_m))^2$$

er minimaliseret.

Algoritmen til bestemmelse af c'erne er vist i figur 5.

I første linie af algoritmen skal de aflede med hensyn til c'erne anvendes:

$$f(x, c_1, c_2, c_3) = c_1 \cdot \exp((-1/c_2) \cdot \ln(1 - \ln(x)/c_3))$$

$$\frac{\partial f}{\partial c_1}(x, c_1, c_2, c_3) = \exp((-1/c_2) \cdot \ln(1 - \ln(x)/c_3))$$

$$\frac{\partial f}{\partial c_2}(x, c_1, c_2, c_3) = c_1 \cdot \exp((-1/c_2) \cdot \ln(1 - \ln(x)/c_3)) \cdot \ln(1 - \ln(x)/c_3) / c_2^2$$

$$\frac{\partial f}{\partial c_3}(x, c_1, c_2, c_3) = -c_1/c_2 \cdot \exp((-1/c_2 - 1) \cdot \ln(1 - \ln(x)/c_3)) \cdot \ln(x)/c_3^2$$

Med et startgæt for c'erne kan opstilles et ligningssystem, der ved løsning giver korrektioner e_1, e_2, e_3 til c_1, c_2 og c_3 . Hvis korrektionerne er for store, foretages en ny beregning med de nye c'er som startværdi.

3.2 Eksempel med DESORPF til desorptionsværdier

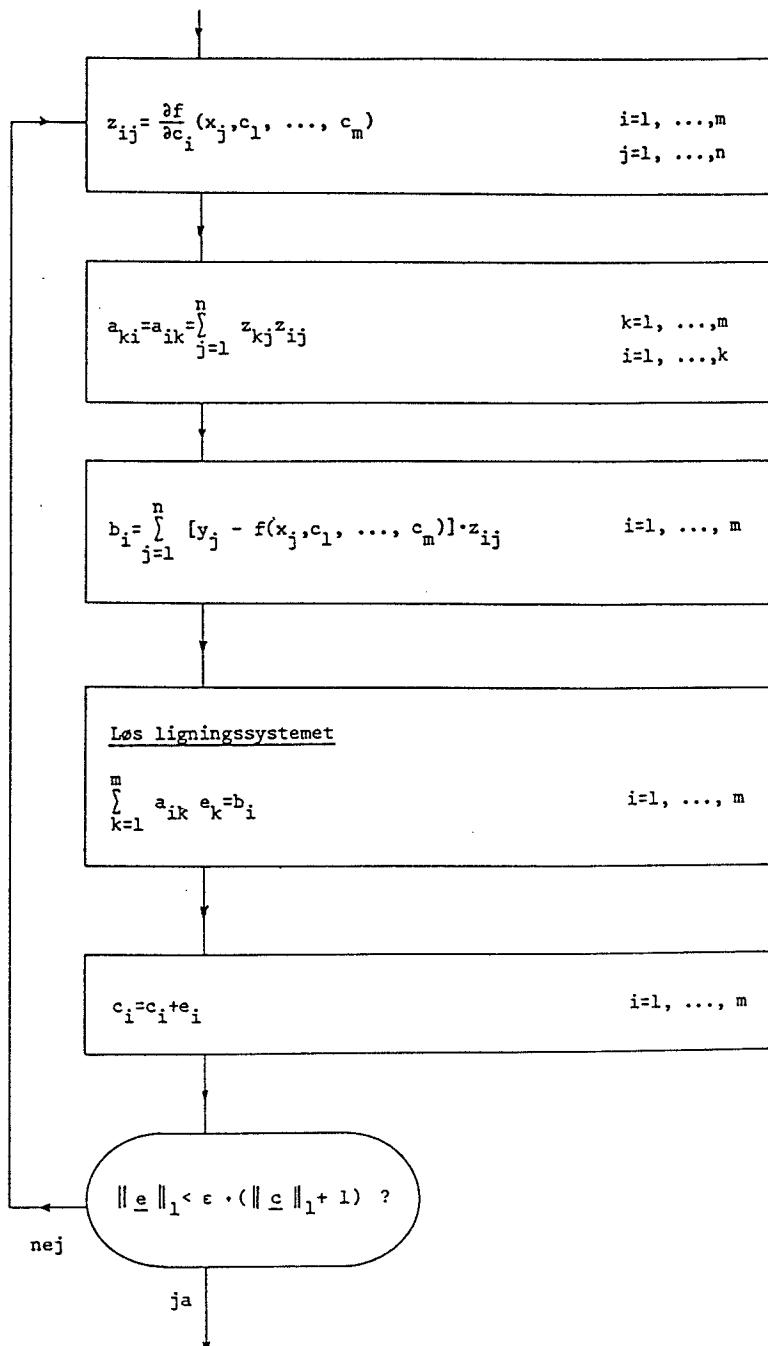
I figur 6 vises linie for linie, hvad der fremkommer på PC'ens skærm, og hvad programmøren skal indtaste.

3.3 Eksempel med SORPF til adsorptionsværdier

I figur 7 vises linie for linie, hvad der fremkommer på PC'ens skærm, og hvad programmøren skal indtaste.

3.4 Resultatets indsættelse i materialefil

Resultatet udskrevet på printer skal indsættes manuelt i materialefil ved hjælp af TURBO PASCAL Edit eller DOS Edlin som vist på figur 8. Resultatet for desorption indsættes i linie 7 og resultatet for adsorption indsættes i linie 11.



Figur 5. Algoritme til bestemmelse af c'erne. Fra reference /3/.

The algorithm for determining the c-values. From Reference /3/.

PC-skærm

Programmør

DESORPF

Program for fitting a nonlinear function to desorptiondata given in a file.

Enter name of file containing desorbtion data : \c\celcon20.510 \c\celcon20.510 ←

The data of \c\celcon20.510 (There are 7 dataset)

Desorption curve.

0.980800	20.200000
0.903300	13.940000
0.756200	7.420000
0.553100	4.900000
0.347000	3.460000
0.197800	2.660000
0.056500	1.000000

Estimation of startguesses for the constants d1,d2 and d3.

a) My guess for d1 based on the first 2 points is : 2.175E+001

b) My guess for d1 based on the first 4 points is : 1.851E+001

Your guess for d1 ? (<return> to use b)

My guesses for d2 and d3 is : 8.662E-001 1.653E-001 ←

Enter your guesses for d2 and d3 :

Iter: 2.088E+001 1.049E+000 1.231E-001

Iter: 2.203E+001 1.109E+000 1.192E-001

Iter: 2.260E+001 1.133E+000 1.183E-001

Iter: 2.288E+001 1.144E+000 1.180E-001

Iter: 2.303E+001 1.149E+000 1.178E-001

Iter: 2.310E+001 1.152E+000 1.177E-001

Iter: 2.313E+001 1.153E+000 1.176E-001

Iter: 2.315E+001 1.154E+000 1.176E-001

Iter: 2.316E+001 1.154E+000 1.176E-001

Iter: 2.317E+001 1.154E+000 1.175E-001

Iter: 2.317E+001 1.154E+000 1.175E-001

Again ? (1=yes/0=no)

På printer udskrives:

Fitting of desorptiondata from file \c\celcon20.510

The constants is :

d(1) : 2.31695E+001

d(2) : 1.15451E+000

d(3) : 1.17534E-001

Sum of squares = 1.058E+000

Figur 6. Eksempel med DESORPF.

Example using DESORPF.

PC-skærm

Programmør

SORPF

Program for fitting a nonlinear function to sorption data given in a file.

Enter name of file containing sorption data: \d_k\dilusand

\d_k\dilusand ←

There are 2 curves in the file.

Enter type of curve to fit:

1=desorption, 2=adsorption, 4=unknown type : 2

The data from \d_k\dilusand (The file contains 6 dataset.)

0.1074	0.29
0.2288	0.52
0.7500	1.08
0.9212	1.69
0.9719	2.26
0.9997	5.99

Enter startguesses for the constants c2,c3 : 2.7 4.2E-003

2.7 4.2E-003 ←

Enter a guess for c1 : 6

6 ←

Interval :	5.4545	9.0000	Sum.o.sq.	0.0909
Interval :	5.4545	7.5818	Sum.o.sq.	0.2178
Interval :	5.4545	6.6000	Sum.o.sq.	0.0515
Interval :	5.9127	6.6000	Sum.o.sq.	0.1294
Interval :	5.9127	6.3251	Sum.o.sq.	0.0523
Interval :	6.0777	6.3251	Sum.o.sq.	0.0723
Interval :	6.2261	6.3251	Sum.o.sq.	0.0519
Interval :	6.2261	6.3055	Sum.o.sq.	0.0510
Interval :	6.2261	6.2855	Sum.o.sq.	0.0509
Interval :	6.2499	6.2855	Sum.o.sq.	0.0511
Interval :	6.2579	6.2855	Sum.o.sq.	0.0508
Interval :	6.2579	6.2744	Sum.o.sq.	0.0509
Interval :	6.2645	6.2744	Sum.o.sq.	0.0509

Iteration ended.

6.26947E+000 2.60451E+000 2.37169E-003 s.o.sq. 5.087E-002

Again ? (1=yes/0=no)

På printer udskrives:

Fitting of adsorption data from file \d_k\dilusand

6.26947E+000 2.60451E+000 2.37169E-003

Sum of squares = 5.087E-002

Figur 7. Eksempel med SORPF.

Example using SORPF.

EXPANDED CLAY

910	20										
2											
1	12										
98.13	98.50	94.71	95.42	84.51	85.24	65.18	65.56	43.85	43.51	20.27	20.55
0.53	0.43	0.38	0.33	0.26	0.21	0.16	0.12	0.15	0.12	0.11	0.07
0.605		2.20			2.35E-2						
2	12										
20.55	20.07	43.74	44.08	65.41	64.72	84.88	84.56	95.33	94.60	97.92	98.66
0.05	0.10	0.11	0.14	0.11	0.14	0.11	0.14	0.16	0.19	0.16	0.24
0.909		5.75			3.18E-6						

Open porosity 62%.

Ahlgren, Lennart: Moisture fixation in porous
building materials. Div. of Build. Techn., Lund
Inst. of Techn. Report 36, Lund, Sweden, 1972.
3 10 85 KKH

Figur 8. Et print af materialefilen EXPCLA20.910 efter indsættelse
af curve fitting resultater.

A print of the material file EXPCLA20.910 after inserting the results
of the curve fitting.

4. Udtegning af katalogside ved hjælp af UDTEGN

Til udtegning af katalogside på A3-papir ved hjælp af BASIC programmet UDTEGN anvendes samme opstilling som vist på figur 1. Dog skal plotterkontakterne (bagpå) stilles 0 0 0 0 1 1 0 0 0 (til A3-papir). Der indsættes en tynd pen (0.3) i nr. 1 i penkarusellen og en tyk pen (0.7) i nr. 2 i penkarusellen. A3-papir indsættes. Med RUN ← startes programmet:

<u>PC-skærm</u>	<u>Programmør</u>
RUN	RUN ←
ANGIV NAVN PÅ ØNSKET DATAFIL ? \A_B\ASBCEM20.188	\A_B\ASBCEM20.188 ←
INDLÆS MÅKSIMALT ØNSKET FUGTINDHOLD ? 15	15 ←

Udtegningen starter på plotter. Efter nogle minutter er katalogsiden færdig. En færdig katalogside kan ses på følgende figur 9.

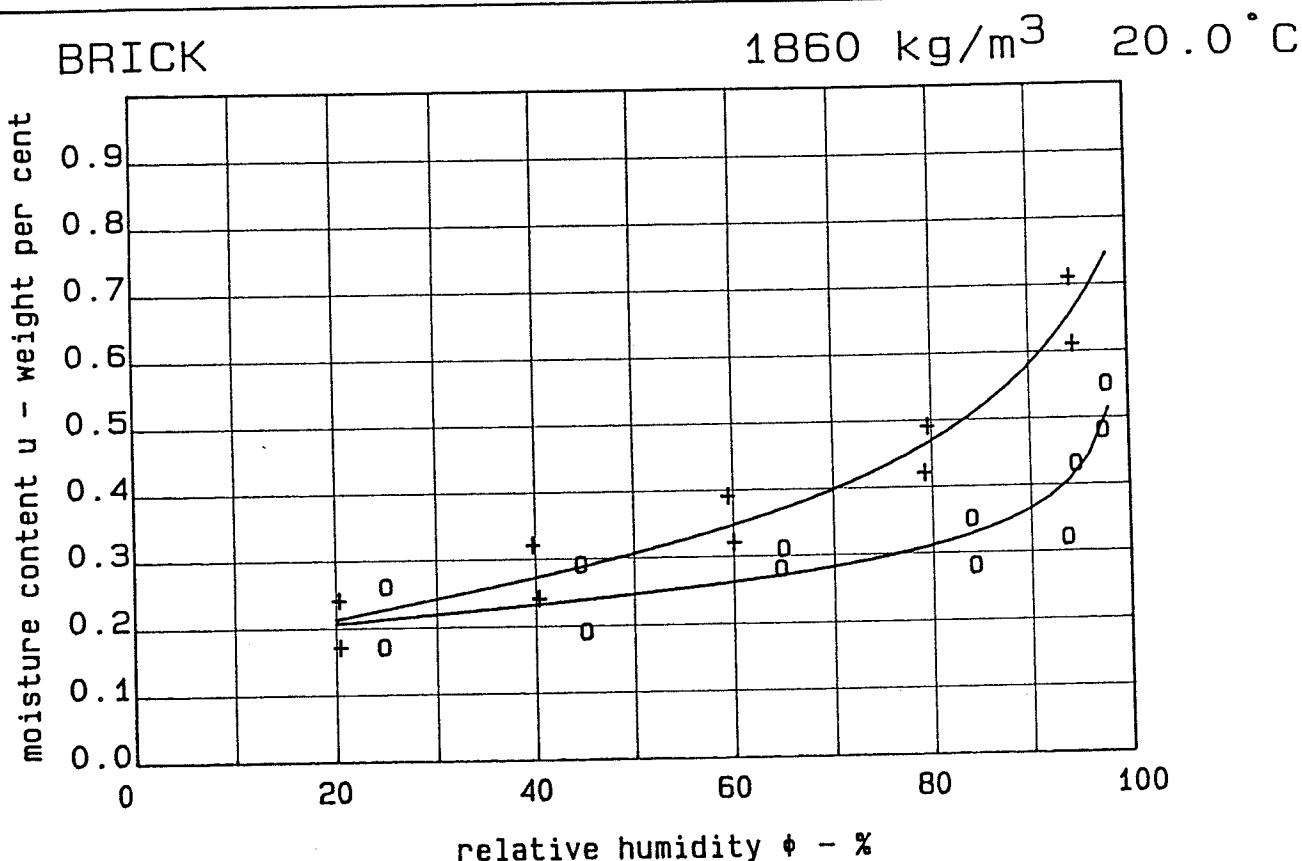
5. Afslutning

Der er opstillet en database for sorptionsisotermer med tilhørende udtegnet katalog. De udarbejdede programmer DATAIND til indlæsning på materialefiler, DESORPF og SORPF til curve fitting og UDTEGN til udtegning af katalogsider er omtalt og udkrevet. Eksempler på programmernes udførelse er vist.

6. Referencer

- /1/ Hansen, Kurt Kielsgaard: SORPTION ISOTHERMS, A Catalogue, Building Materials Laboratory, DTH, Teknisk Rapport 162/86, 1986.
- /2/ Kowalik, J., Osborne, M.R.: Methods for Unconstrained Optimization Problems. American Elsevier Publishing Company, Inc., New York, 1968.
- /3/ Madsen, Kaj: CURVE-FITTING. Numerisk Institut, DTH. Hæfte 20, 1973.

TECHNICAL UNIVERSITY OF DENMARK, Building Materials Laboratory
Sorption of water in building materials



+ measured desorption values

ϕ 94.2 94.5 79.6 79.3 59.5 60.1 39.8 40.3 20.3 20.4
 u 0.71 0.61 0.49 0.42 0.39 0.32 0.32 0.24 0.24 0.17

Approximation:

$$u = 8.11E-01 \times \exp((-1/2.09) \times \ln(1-\ln(\phi)) / 1.04E-01)$$

o measured adsorption values

ϕ 24.7 25.0 45.1 44.6 64.7 65.0 84.3 84.0 93.7 94.6 97.4 97.8
 u 0.17 0.26 0.19 0.29 0.28 0.31 0.28 0.35 0.32 0.43 0.48 0.55

Approximation:

$$u = 1.15E+00 \times \exp((-1/4.76) \times \ln(1-\ln(\phi)) / 4.45E-04)$$

No scanning values

Notes: OPEN POROSITY 31%

Litterature: Ahlgren, Lennart: Moisture fixation in porous building materials. Div. of Build. Techn., Lund Inst. of Techn. Report 36, Lund, Sweden, 1972.

Date: 3-10-85

Initials: KKH

File: \A_B\BRICK20.186

Figur 9. Et eksempel på en katalogside.

An example of a catalogue page.

APPENDIX A. Udskrift af BASIC programmet DATAIND

```
10 REM LENE RATHKJEN 1985
20 REM
30 REM PROGRAM TIL OPRETTELSE AF DATAFILER
40 REM INDEHOLDENDE SORPTIONSISOTERMVÆRDIER
50 REM FOR BYGNINGSMATERIALE
60 REM
70 DIM BEM$(3),LITT$(3)
80 PRINT "INDLÆS MATERIALENAVN (MAX 20 KARAKTERER)"
90 INPUT NAVN$
100 PRINT "INDLÆS DENSITET, TEMPERATUR (HVIS EJ MALT, INDLÆS 0)"
110 INPUT;DENS,TEMP
120 PRINT "
130 PRINT "ØNSKES SORPTIONSDATA INDLÆST FRA TASTATUR (1)"
140 INPUT "ELLER VHA DIGITIZING (2) ";SVAR
150 ON SVAR GOSUB 650,790
160 IF SVAR<>1 AND SVAR<>2 GOTO 130
170 DIM AAA(NKURV,5)
180 FOR I=1 TO NKURV
190 IF K(I)=3 GOTO 240
210 FOR J=0 TO 5
220 AAA(I,J)=A(J)
230 NEXT J
240 NEXT I
250 PRINT "YDERLIGERE MATERIALE EGENSKABER (3 LINIER A MAX 50 KARAKTERER)"
260 FOR I=1 TO 3
270 LINE INPUT BEM$(I)
280 NEXT I
290 PRINT "KILDEANGIVELSE (3 LINIER A MAX 50 KARAKTERER)"
300 FOR I=1 TO 3
310 LINE INPUT LITT$(I)
320 NEXT I
330 INPUT "DATO DD,MM,YY, INITIALER ";D,M,Y,INIT$
340 PRINT "HVILKET FILNAVN ØNSKES DATA GEMT UNDER ? "
350 INPUT FIL$
360 OPEN FIL$ FOR OUTPUT AS #1
370 PRINT#1,NAVN$
375 PRINT#1,DENS,TEMP
380 PRINT#1,NKURV
390 FOR I=1 TO NKURV
400 PRINT#1,K(I),NPKT(I)
410 FOR J=1 TO NPKT(I)-1
420 PRINT#1,USING "##.## ";PHI(I,J);
430 NEXT J
440 PRINT#1,USING "##.## ";PHI(I,NPKT(I))
450 FOR J=1 TO NPKT(I)-1
460 PRINT#1,USING "##.## ";U(I,J);
470 NEXT J
480 PRINT#1,USING "##.## ";U(I,NPKT(I))
500 IF K(I)=1 THEN PRINT#1," d1      d2      d3"
510 IF K(I)=2 THEN PRINT#1," a1      a2      a3"
520 IF K(I)=4 THEN PRINT#1," u1      u2      u3"
```

```
540 NEXT I
550 FOR I=1 TO 3
560 PRINT#1,BEM$(I)
570 NEXT I
580 FOR I=1 TO 3
590 PRINT#1,LITT$(I)
600 NEXT I
610 PRINT#1,USING "## ";D,M,Y;
620 PRINT#1,INIT$
630 CLOSE #1
640 END
650 REM INDLÆSNING FRA TASTATURET
660 INPUT "ANTAL KURVER ";NKURV
670 DIM NPKT(NKURV),PHI(NKURV,20),U(NKURV,20),K(NKURV)
680 FOR I=1 TO NKURV
690 PRINT "KURVE NR";I
700 PRINT "SORPTIONSISOTERMTYPE (DESORPTION (1), ADSORPTION (2),""
710 INPUT "SCANNING (3), UKENDT (4) ";K(I)
720 INPUT "ANTAL MÅLEPUNKTER ";NPKT(I)
730 FOR J=1 TO NPKT(I)
740 PRINT "MÅLEPUNKT NR";J;"(PHI,U)= "
750 INPUT PHI(I,J),U(I,J)
760 NEXT J
770 NEXT I
780 RETURN
790 REM DIGITIZING
800 OPEN "COM1:2400,S,7,1,CS65535,DS,CD" AS #2
810 PRINT "INDLÆSNING AF TO PUNKTER TIL SKALERING AF SYSTEMET"
820 PRINT "DIGITIZE PUNKT (PHI,U)=(0,0)"
830 GOSUB 1070
840 X0=X
850 Y0=Y
860 INPUT "INDLÆS KENDT PHI, U ";PHI1,U1
870 PRINT "DIGITIZE PUNKTET"
880 GOSUB 1070
890 X1=X
900 Y1=Y
910 INPUT "ANTAL KURVER ";NKURV
920 DIM NPKT(NKURV),PHI(NKURV,20),U(NKURV,20),K(NKURV)
930 FOR I=1 TO NKURV
940 PRINT "KURVE NR.";I
950 PRINT "SORPTIONSISOTERMTYPE (DESORPTION (1), ADSORPTION (2),""
960 INPUT "SCANNING (3), UKENDT (4) ";K(I)
970 INPUT "ANTAL MÅLEPUNKTER ";NPKT(I)
980 FOR J=1 TO NPKT(I)
990 PRINT "DIGITIZE PUNKT NR";J
1000 GOSUB 1070
1010 PHI(I,J)=(X-X0)/(X1-X0)*PHI1
1020 U(I,J)=(Y-Y0)/(Y1-Y0)*U1
1030 NEXT J
1040 NEXT I
```

```
1050 CLOSE#2
1060 RETURN
1070 REM  DIGITIZING AF ET PUNKT
1080 PRINT#2,"DP;"
1090 PRINT#2,"OS;"
1100 INPUT#2,STATUS
1110 STATUS=INT(STATUS/4)
1120 IF STATUS=INT(STATUS/2)*2 THEN GOTO 1090
1130 PRINT#2,"OD;"
1140 INPUT#2,X,Y,P
1150 RETURN
Ok
```

APPENDIX B. Udskrift af PASCAL programmet DESORPF

```
1  program DESORPF;

3  (* Program for fitting a nonlinear functions describing a desorption isotherm
4   to numeric data. Use this program for fitting unknown sorption data.
5   The calculated values for d1, d2 and d3 is inserted manually in the file
6   for the actual material.
7   The nonlinear fitting function is supplied as the procedure g(x,k).
8   The program reads data from a textfile, and performs a iteration to
9   find the constants for the function.

11  Included procedures are Gauss and Nolinfit.

13  Directed by K.Kielsgaard Hansen.
14  Programmed by L.Mouritsen jan 1986. *)
```

```
17  const
18    m=3;          (*Number of constants in function.*)
19    maxpoints=20; (*Maximum number of points to fit.*)
20    fak=0.5;      (*Step reduction factor.*)
21    q=1;          (*scale factor for c3*)

23  type
24    gaussarr     =array[1..m,1..m] of real;
25    gaussvec     =array[1..m] of real;
26    fitarr       =array[1..maxpoints,1..2] of real;
27    name         =string[24];
28    answertype   =0..1;

30  var
31    c             :array[1..m] of real;
32    arr,linarr   :fitarr;
33    tol,dev,alfa,beta
34                  :real;
35    npoints,ctype,i,nocurves
36                  :integer;
37    filename      :name;
38    infile        :text;
39    answer        :answertype;
40    ok            :boolean;

42  function g(x:real; k:integer):real;
43  (*Returns the value of the fitting function for k=0.
44   For k>0 returns the derivative from the k'th constant.*)

46  var
47    temp :real;
48  begin
49    case k of
50      0: temp:=c[1]*exp((-1/c[2])*ln(abs(1-ln(x)/(q*c[3]))));  (*the function*)
```

```
51      1: temp:=exp((-1/c[2])*ln(abs(1-ln(x)/(q*c[3]))));      (*the derivatives*)
52      2: temp:=c[1]*exp((-1/c[2])*ln(abs(1-ln(x)/(q*c[3]))))
53          *ln(abs(1-ln(x)/(q*c[3])))/sqr(c[2]);
54      3: temp:=-c[1]/c[2]*exp((-1/c[2]-1)*ln(abs(1-ln(x)/(q*c[3]))))
55          *ln(x)/sqr(q*c[3]);
56    end;
57    g:=temp
58  end;

60  procedure gauss(a:gaussarr;
61                  y:gaussvec;
62                  n:integer;
63                  var x:gaussvec);
64 (*Procedure for solving linear equations using gauss elimination,
65   backsubstitution and pivoting.
66       a * x = y
67       = - -
68   n is the number of equations.
69   ref Numerisk Institut haefte 23.
70   L.Mouritzen.....Jan 86           *)
71
72  var
73      i,r,s,rownr            :integer;
74      c,ann,aii,arr,collmax,temp :real;

75  begin
76      (*gauss elimination*)
77      for i:=1 to n-1 do
78      begin
79          (*pivoting*)
80          collmax:=abs(a[i,i]);
81          rownr:=i;
82          for r:=i+1 to n do      (*find largest element*)
83              if abs(a[r,i])>collmax then
84                  begin
85                      collmax:=abs(a[r,i]);
86                      rownr:=r;
87                  end;
88          if rownr>i then        (*swap rows if current row is smaller*)
89          begin
90              for s:=i to n do
91                  begin          (*row in matrix*)
92                      temp:=a[i,s]; a[i,s]:=a[rownr,s]; a[rownr,s]:=temp;
93                  end;             (*right side too*)
94                  temp:=y[i]; y[i]:=y[rownr]; y[rownr]:=temp;
95              end;
96          (*eliminate*)
97          aii:=a[i,i];
98          for r:=i+1 to n do
99          begin
100             c:=a[r,i]/aii;
```

```
102      for s:=i+1 to n do a[r,s]:=a[r,s]-a[i,s]*c;
103          y[r]:=y[r]-y[i]*c;
104      end;
105  end;
106 (*backsubstitution*)
107 ann:=a[n,n];
108 x[n]:=y[n]/ann;
109 for r:=n-1 downto 1 do
110 begin
111     arr:=a[r,r];
112     temp:=0;
113     for s:=r+1 to n do temp:=temp+a[r,s]*x[s];
114     x[r]:=(y[r]-temp)/arr;
115 end;
116 end; (*gauss*)

118 procedure nolinfit(val:fitarr;
119                      n:integer; eps:real; var stdev:real);
120 (* Program for fitting nonlinear functions to numeric data.
121   The derivatives of the fitting function must be supplied
122   in the function g(x,k).
123   Fits the function given by g( ) and the constants c[1]-c[m]
124   to the n points in val[ ], using iteration. eps is the wanted
125   precision after the iteration.
126   ref. Numerisk Institut haefte 20.
127   L.Mouritsen ... jan 1986. *)
128
131 var
132     aa           :gaussarr;
133     z            :array[1..m,1..maxpoints] of real;
134     b,e          :gaussvec;
135     suma,sumb,cinorm,einorm :real;
136     i,j,k,iteration,stop    :integer;
137
138 begin
139     iteration:=0;
140     stop:=0;
141     repeat
142         iteration:=iteration+1;
143         (*set up the equation system*)
144         for i:=1 to m do
145             for j:=1 to n do
146                 z[i,j]:=g(val[j,1],i);      (*values of derivatives*)
147         for k:=1 to m do
148             for i:=1 to k do
149             begin
150                 suma:=0; sumb:=0;
151                 for j:=1 to n do
152                     begin
```

```
153     suma:=suma+z[k,j]*z[i,j];--  
154     if i=1 then sumb:=sumb+(val[j,2]-g(val[j,1],0))*z[k,j];  
155     end;  
156     aa[k,i]:=suma;           (*aa is symmetric*)  
157     if k>>i then aa[i,k]:=suma;  
158     if i=1 then b[k]:=sumb;  
159     end;  
160     (*solve the system*)  
161     gauss(aa,b,m,e);  
162     cinorm:=0;   einorm:=0;  
163     for i:=1 to m do  
164     begin  
165       c[i]:=c[i]+fak*e[i];  
166       cinorm:=cinorm+abs(c[i]);  
167       einorm:=einorm+abs(e[i]);  
168     end;  
169     write('Iter: '); for i:=1 to m do write(c[i]:10,' '); writeln;  
170     (*test for near overflow*)  
171     if abs(c[1])+abs(c[2])+abs(c[3])>10000 then  
172     begin  
173       write('Overflow approaching, (1=stop / 0=continue) ');  
174       readln(stop);  
175     end;  
176     (*test for too many iterations*)  
177     if iteration mod 25=0 then  
178     begin  
179       write('No convergence after ',iteration:3,  
180             ' iterations, (1=stop / 0=continue) ');  
181       readln(stop);  
182     end;  
183     (*test for convergence*)  
184     until (einorm<eps*(cinorm+1)) or (stop=1);  
185     stdev:=0;  
186     for j:=1 to n do stdev:=stdev+sqr(val[j,2]-g(val[j,1],0));  
187     (*if aborted*)  
188     if stop=1 then for i:=1 to n do c[i]:=0;  
189     end;      (*nolinfit*)  
  
191     procedure linreg(val:fitarr;  
192                      n:integer; var a,b:real);  
193     (*Fits a line to the n points in val[] using linear regression.  
194      The result is a and b:  
195      y = a*x + b;  
196      L.Mouritzen feb 1986 *)  
  
198     var  
199       xi,yi,sx,sy,sq,sp          :real;  
200       i                           :integer;  
201     begin  
202       sx:=0; sy:=0; sq:=0; sp:=0;  
203       for i:= 1 to n do
```

```
204 begin
205   xi:=val[i,1];
206   yi:=val[i,2];
207   sx:=sx+xi;                      (*sum of x*)
208   sy:=sy+yi;                      (*sum of y*)
209   sq:=sq+xi*xi;                  (*sum of squares*)
210   sp:=sp+xi*yi;                  (*sum of products*)
211 end;
212 a:=(sp-sx*sy/n)/(sq-sx*sx/n);
213 b:=sy/n-a*sx/n;
214 end;      (*linreg*)

217 (*main program*)
218 begin
219   textmode;                      (*clear the screen*)
220   writeln;
221   writeln(                         DESORPF');
222   writeln(
223   'Program for fitting a nonlinear function to desorption data given in a file.');
224   repeat
225     repeat
226       writeln;
227       write('Enter name of file containing desorption data : ');
228       readln(filename);
229       assign(infile,filename);          (*open the file*)
230       {$I-}                          (*check off*)
231       reset(infile);                 (*point to start of file*)
232       {$I+}                          (*check on*)
233       ok:=(ioreadlt=0);              (*IO error?*)
234       if not ok then writeln('File not found !');
235     until ok;
236     readln(infile);                (*skip name*)
237     readln(infile);                (*skip density,temp.*)
238     readln(infile,nocurves);        (*number of curves in file*)
239     readln(infile,ctype,npoinnts);   (*type of curve, points*)
240     while (ctype<>1) and (ctype<>4) and (nocurves>0) do
241       begin                         (*desorption or unknown*)
242         readln(infile); readln(infile); readln(infile);
243         nocurves:=nocurves-1;
244         if nocurves>0 then readln(infile,ctype,npoinnts);
245       end;
246     if nocurves=0 then             (*no curves found*)
247       begin
248         close(infile);
249         writeln('No "desorption" or "unknown type" found.');
250       end
251     else
252       begin
253         for i:=1 to npoinnts do
```

```
255 begin
256   read(infile,arr[i,1]);           (*read x-values*)
257   arr[i,1]:=arr[i,1]/100;          (*scale down*)
258 end;
259 for i:=1 to npoints do read(infile,arr[i,2]);(*read y-values*)
260 close(infile);                  (*close the file*)
261 writeln;
262 writeln(' The data of ',filename,' (There are ',npoints,', dataset)');
263 if ctype=1 then writeln('Desorption curve.');
264 if ctype=4 then writeln('Type of curve unknown.');
265 for i:=1 to npoints do
266   writeln(arr[i,1]:12:6,arr[i,2]:12:6);
267 writeln;

269 (*startguesses*)
270 writeln('Estimation of startguesses for the constants d1,d2 and d3.');
271 writeln;
272 (*d1 is curve intersection with x=1, the line through the first 2 points
273   is extended.*)
274 if arr[1,1]<>arr[2,1] then
275   c[1]:=(1-arr[1,1])*(arr[1,2]-arr[2,2])/
276     (arr[1,1]-arr[2,1])+arr[1,2]
277 else
278   c[1]:=0;
279 writeln('a) My guess for d1 based on the first 2 points is : ',c[1]:10);
280 (*or linear regression through the first 4 points.*)
281 for i:=1 to 4 do
282 begin
283   linarr[i,1]:=arr[i,1];
284   linarr[i,2]:=arr[i,2];
285 end;
286 linreg(linarr,4,alfa,beta);
287 c[1]:=alfa+beta;
288 writeln('b) My guess for d1 based on the first 4 points is : ',c[1]:10);
289 write('Your guess for d1 ? (<return> to use b) ');
290 readln(c[1]);
291 (*If d2 is set to 1, the equation can be solved for d3.*)
292 c[2]:=1;
293 c[3]:=0;                         (*make average of d3's*)
294 for i:=1 to npoints do
295   c[3]:=c[3]+arr[i,2]*ln(arr[i,1])/(arr[i,2]-c[1]);
296 c[3]:=c[3]/npoints;
297 (*Hereafter d2 can be improved by solving the equation *)
298 c[2]:=0;                         (*averages of d2's*)
299 for i:=1 to npoints do
300   if arr[i,2]>0 then
301     c[2]:=c[2]-ln(abs(1-ln(arr[i,1])/c[3]))/ln(abs(arr[i,2]/c[1]));
302   c[2]:=c[2]/npoints;
303 writeln('My guesses for d2 and d3 is : ',c[2]:10,' ',c[3]:10);
304 write('Enter your guesses for d2 and d3 : ');
305 readln(c[2],c[3]);
```

```
307 {   write('Enter maximum tolerance on fitting : '); readln(tol); }
308 tol:=0.0001;

310 (*fit, output to printer*)
311 nolinfit(arr,npoints,tol,dev);
312 if c[1]+c[2]+c[3]<>0 then
313 begin
314   if ctype=1 then
315     writeln(lst,'Fitting of desorptiondata from file ',filename)
316   else
317     writeln(lst,'Fitting of unknown type of data from file ',filename);
318     writeln(lst,'The constants is ');
319     for i:=1 to m do writeln(lst,'d(',i:1,') : ',c[i]:12);
320     writeln(lst,'Sum of squares = ',dev:10);
321     writeln(lst); writeln(lst);
322   end
323   else
324   -- writeln('Iteration aborted.');
325   end;
326   answer:=1;
327   write('Again ? (1=yes/0=no) '); readln(answer);
328   until answer=0;
329 end.
```

APPENDIX C. Udskrift af PASCAL programmet SORPF

```
1  program SORPF;

3  (* Program for fitting a nonlinear functions describing a adsorption or
4   desorption isotherm to numeric data, a reduced version of DESORPF,
5   only two variables are fitted, use it for fitting adsorption data or
6   difficult desorption data. Calculates values for a2 and a3, while a1
7   is iterated to obtain minimum sum of squares of deviations. a1, a2 and
8   a3 is inserted manually in the file for the actual material.
9   The nonlinear fitting function is supplied as the procedure g(x,k).
10  The program reads data from a textfile, and performs a iteration to
11  find the constants for the function.

13  Included procedures are Gauss and Nolinfit.

15  Directed by K.Kielsgaard Hansen.
16  Programmed by L.Mouritsen jan 1986. *)
```

```
18  const
19    m=2;          (*Number of constants in function.*)
20    maxpoints=20; (*Maximum number of points to fit.*)
21    fak=0.5;      (*Step reduction factor.*)

23  type
24    gaussarr     =array[1..m,1..m] of real;
25    gaussvec     =array[1..m] of real;
26    fitarr       =array[1..maxpoints,1..2] of real;
27    name         =string[24];
28    answertype   =0..1;

30  var
31    c            :array[1..m] of real;
32    arr          :fitarr;
33    a,x1,x2,b,fx1,fx2,sta,stb,temp,
34    tol,dev,ck,s,q :real;
35    npoints,count,iterno,i,ctype,wtype,nocurves,stop
36                  :integer;
37    filename     :name;
38    infile       :text;
39    answer       :answertype;
40    ok,again     :boolean;

42  function g(x:real;k:integer):real;
43  (*Returns the value of the fitting function for k=0.
44   For k>0 returns the derivative from the k'th constant.*)

46  var
47    temp :real;

49  begin
50    case k of
```

```
51      0: temp:=ck*exp((-1/c[1])*ln(abs(1-ln(x)/(q*c[2])))); (*the function*)
52      1: temp:=ck*exp((-1/c[1])*ln(abs(1-ln(x)/(q*c[2]))))
53          *ln(abs(1-ln(x)/(q*c[2])))/sqr(c[1]);
54      2: temp:=-ck/c[1]*exp((-1/c[1]-1)*ln(abs(1-ln(x)/(q*c[2]))))
55          *ln(x)/sqr(q*c[2]);
56    end;
57    g:=temp
58  end;

59
60  procedure gauss(a:gaussarr;
61                  y:gaussvec;
62                  n:integer;
63                  var x:gaussvec);
64 (*Procedure for solving linear equations using gauss elimination,
65   backsubstitution and pivoting.
66   a * x = y
67   = - -
68   n is the number of equations.
69   ref Numerisk Institut haefte 23.
70   L. Mouritzen .....jan 86.      *)
71
72  var
73      i,r,s,rownr           :integer;
74      c,ann,aII,arr,collmax,temp :real;

75  begin
76      (*gauss elimination*)
77      for i:=1 to n-1 do
78      begin
79          begin
80              (*pivoting*)
81              collmax:=abs(a[i,i]);
82              rownr:=i;
83              for r:=i+1 to n do      (*find largest element*)
84                  if abs(a[r,i])>collmax then
85                  begin
86                      collmax:=abs(a[r,i]);
87                      rownr:=r;
88                  end;
89                  if rownr>i then      (*swap rows if current row is smaller*)
90                  begin
91                      for s:=i to n do
92                          begin          (*row in matrix*)
93                              temp:=a[i,s]; a[i,s]:=a[rownr,s]; a[rownr,s]:=temp;
94                          end;          (*right side too*)
95                          temp:=y[i]; y[i]:=y[rownr]; y[rownr]:=temp;
96                      end;
97                      (*eliminate*)
98                      aII:=a[i,i];
99                      for r:=i+1 to n do
100                         begin
101                             c:=a[r,i]/aII;
```

```
102      for s:=i+1 to n do a[r,s]:=a[r,s]-a[i,s]*c;
103      y[r]:=y[r]-y[i]*c;
104    end;
105  end;
106 (*backsubstitution*)
107 ann:=a[n,n];
108 x[n]:=y[n]/ann;
109 for r:=n-1 downto 1 do
110 begin
111   arr:=a[r,r];
112   temp:=0;
113   for s:=r+1 to n do temp:=temp+a[r,s]*x[s];
114   x[r]:=(y[r]-temp)/arr;
115 end;
116 end; (*gauss*)

118 procedure molinfit(val:fitarr;
119                      n:integer; eps:real; var stdev:real);

121 (* Program for fitting nonlinear functions to numeric data.
122   The derivatives of the fitting function must be supplied
123   in the function g(x,k).
124   Fits the function given by g( ) and the constants c[1]-c[m]
125   to the n points in val[ ], using iteration. eps is the wanted
126   precision after the iteration.
127   ref. Numerisk Institut haeftte 20.
128   L.Mouritsen ... jan 1986. *)
129
130 var
131   aa           :gaussarr;
132   z            :array[1..m,1..maxpoints] of real;
133   b,e          :gaussvec;
134   suma,sumb,cinorm,einorm :real;
135   i,j,k        :integer;

136 begin
137   stop:=0;
138   repeat      (*iteration loop*)
139     (*set up the equation system*)
140     for i:=1 to m do
141       for j:=1 to n do
142         z[i,j]:=g(val[j,1],i);  (*values of derivatives*)
143     for k:=1 to m do
144       for i:=1 to k do
145         begin
146           suma:=0; sumb:=0;
147           for j:=1 to n do
148             begin
149               suma:=suma+z[k,j]*z[i,j];
150               if i=1 then sumb:=sumb+(val[j,2]-g(val[j,1],0))*z[k,j];
151             end;
```

```
153     aa[k,i]:=suma;          (*aa is symmetric*)
154     if k>>i then aa[i,k]:=suma;
155     if i=1 then b[k]:=sumb;
156   end;
157   (*solve the system*)
158   gauss(aa,b,m,e);
159   c1norm:=0;  einorm:=0;
160   for i:=1 to m do
161   begin
162     c[i]:=c[i]+fak*e[i];
163     c1norm:=c1norm+abs(c[i]);
164     einorm:=einorm+abs(e[i]);
165   end;
166   for i:=1 to m do c[i]:=abs(c[i]);
167   write('Iter: ',ck:10,' ');
168   for i:=1 to m do write(c[i]:10,' '); write(#13);
169   (*test for near overflow*)
170   if abs(c[1])+abs(c[2])>10000 then
171   begin
172     write('Overflow approaching, (1=stop / 0=continue) ');
173     readln(stop);
174   end;
175   (*test for convergence*)
176   until (einorm<eps*(c1norm+1)) or (stop=1);
177   delline;                      (*clear iteration line*)
178   if stop>>1 then
179   begin
180     stdev:=0;
181     for j:=1 to n do
182       stdev:=stdev+sqr(val[j,2]-g(val[j,1],0));
183   end;
184 end;      (*nolinfits*)

187 (*main program*)
188 begin
189   textmode;
190   writeln;
191   writeln(
192 ,                               SORPF');
193   writeln(
194 'Program for fitting a nonlinear function to sorption data given in a file.');
195   writeln;
196   repeat
197     repeat
198       writeln;
199       write('Enter name of file containing sorption data: ');
200       readln(filename);
201       assign(infile,filename);           (*open the file*)
202       {$I-}                            (*check off*)
203       reset(infile);                  (*point to start of file*)
```

```
204      {$I+}                                (*check on*)
205      ok:=(iore result=0);
206      if not ok then writeln('File not found !');
207      until ok;
208      readln(infile);                      (*skip name*)
209      readln(infile);                      (*skip density,temp*)
210      readln(infile,nocurves);             (*number of curves in file*)
211      writeln;
212      writeln('There are ',nocurves:1,' curves in the file.');
213      repeat
214          writeln('Enter type of curve to fit:');
215          write('1=desorption, 2=adsorption, 4=unknown type : ');
216          readln(wtype);
217      until (wtype=1) or (wtype=2) or (wtype=4);
218      readln(infile,ctype,np points);        (*type of curve, points*)
219      while (ctype<>wtype) and (nocurves>0) do
220      begin
221          readln(infile); readln(infile); readln(infile);
222          nocurves:=nocurves-1;
223          if nocurves>0 then readln(infile,ctype,np points);
224      end;
225      if nocurves=0 then                   (*no curves found*)
226      begin
227          close(infile);
228          writeln('No number ',wtype:1,' curve found.');
229      end
230      else
231      begin
232          for i:=1 to np points do
233          begin
234              read(infile,arr[i,1]);           (*read x-values*)
235              arr[i,1]:=arr[i,1]/100;         (*scale down*)
236          end;
237          for i:=1 to np points do
238              read(infile,arr[i,2]);           (*read y-values*)
239          close(infile);
240          writeln(
241          'The data from ',filename,' (The file contains ',np points,', dataset.)');
242          for i:=1 to np points do writeln(arr[i,1]:12:4,arr[i,2]:12:2);
243          writeln;
244      {      write('Enter scale factor for a3 : '); readln(q);}
245      q:=1;
246      c[1]:=1; c[2]:=0.1;
247      write('Enter startguesses for the constants c2,c3 : ');
248      readln(c[1],c[2]);
249      write('Enter a guess for c1 : '); readln(Ck);
250      {      write('Enter maximum tolerance on fitting : '); readln(tol);}
251      tol:=0.0001;
252      repeat
253          (*fit*)
254          (*start values for fitting*)
```

```
255 (*find minimum for ck using a variant of golden section method.*)
256 (*interval 1.5 to each side of startguess for ck*)
257 a:=ck/1.5;    b:=ck*1.5;
258 sta:=a;        stb:=b;                      (*keep start interval*)
259 x1:=ck/1.1;   x2:=ck*1.1;
260 fx1:=0;       fx2:=0;
261 repeat
262     if fx1=0 then                         (*calc only if needed*)
263         begin
264             ck:=x1;
265             nolinfit(arr,npoints,tol,dev);
266             fx1:=dev;
267         end;
268         if fx2=0 then
269             begin
270                 ck:=x2;
271                 nolinfit(arr,npoints,tol,dev);
272                 fx2:=dev;
273             end;
274         if stop<>1 then -
275             begin
276                 if fx2>fx1 then                  (*select new interval*)
277                     begin
278                         b:=x2;
279                         x2:=x1;
280                         fx2:=fx1; fx1:=0;
281                         x1:=a+0.4*(b-a);
282                     end
283                 else
284                     begin
285                         a:=x1;
286                         x1:=x2;
287                         fx1:=fx2; fx2:=0;
288                         x2:=b-0.4*(b-a);
289                     end;
290                 if x1>x2 then                  (*swap so x2>x1*)
291                     begin
292                         temp:=x1; x1:=x2; x2:=temp;
293                         temp:=fx1; fx1:=fx2; fx2:=temp;
294                     end;
295                     writeln('Interval : ',a:9:4,b:9:4,' Sum.o.sq. ',dev:9:4);
296                 end;
297             until ((b-a)/(b+a)<0.001) or (stop=1);
298             (*insure that min. is not in endpoint.*)
299             again:=false;
300             if ((a=sta) or (b=stb)) and (stop=0) then
301                 begin
302                     ck:=(b+a)/2;                   (*new startguess*)
303                     again:=true;
304                     writeln(
305             'The result of the iteration is one of the endpoints of the startinterval.';
```

```
306      writeln('A new iteration is attempted with new startguess : ',ck:8:3);
307      stop:=0;
308      write('Abort this iteration ? (1 = stop / 0 = continue) ');
309      readln(stop);
310      end;
311      until (not again) or (stop=1);
312      if stop=1 then
313          writeln('Iteration aborted.')
314      else
315      begin
316          (*use mean value as result*)
317          ck:=(b+a)/2;
318          if wtype=1 then
319              writeln(lst,'Fitting of desorption data from file ',filename);
320          if wtype=2 then
321              writeln(lst,'Fitting of adsorption data from file ',filename);
322          if wtype=4 then
323              writeln(lst,'Fitting of unknown sorption data from file ',filename);
324          writeln('Iteration ended.');
325          writeln(ck:12,' ',c[1]:12,' ',c[2]:12,' s.o.sq. ',dev:10);
326          writeln(lst,ck:12,' ',c[1]:12,' ',c[2]:12);
327          writeln(lst,'Sum of squares = ',dev:10);
328          writeln(lst); writeln(lst);
329      end;
330  end;
331  answer:=1; -
332  write('Again ? (1=yes/0=no) '); readln(answer);
333  until answer=0;
334 end.
```

APPENDIX D. Udskrift af BASIC programmet UDTEGN

```
10 REM LENE RATHKJEN, 1985
20 REM
30 REM PROGRAM TIL UDTEGNING AF DATAFILER
40 REM INDEHOLDENDE SORPTIONSISOTERMVÆRDIER
50 REM FOR BYGNINGSMATERIALER
60 REM
70 DIM TEKST$(4),BEM$(3),LITT$(3),UA(100)
80 INPUT "ANGIV NAVN PÅ ØNSKET DATAFIL ";FIL$
85 INPUT "INDLÆS MÅKSIMALT ØNSKET FUGTINDHOLD ";UMAX
90 OPEN FIL$ FOR INPUT AS #1
100 INPUT#1, NAVN$, DENS, TEMP
110 INPUT#1, NKURV
120 DIM A(NKURV,3),U(NKURV,20),PHI(NKURV,20)
130 FOR I=1 TO NKURV
140 INPUT#1, K(I), NPKT(I)
150 FOR J=1 TO NPKT(I)
160 INPUT#1, PHI(I,J)
170 NEXT J
180 FOR J=1 TO NPKT(I)
190 INPUT#1, U(I,J)
200 NEXT J
210 IF K(I)=3 GOTO 250
220 FOR J=1 TO 3
230 INPUT#1, A(I,J)
240 NEXT J
250 NEXT I
280 FOR I=1 TO 3
290 LINE INPUT#1, BEM$(I)
300 NEXT I
310 FOR I=1 TO 3
320 LINE INPUT#1, LITT$(I)
330 NEXT I
340 INPUT#1, D,M,Y,INIT$
350 CLOSE #1
480 OPEN "COM1:2400,S,7,1,CS65535,DS,CD" AS #2
490 PRINT#2,"IN;R090;IP;IW;SI0.275,0.375;"
500 REM
510 REM UDSKRIVNING AF HOVED
520 TEKST1$="TECHNICAL UNIVERSITY OF DENMARK,"
530 TEKST2$="Buildings Materials Laboratory"
540 PRINT#2,"SP1;PU500,15650;LB"+TEKST1$+TEKST2$+CHR$(3);
550 TEKST3$="Sorption of water in building materials"
560 PRINT#2,"PU500,15300;LB"+TEKST3$+CHR$(3);
570 PRINT#2,"SP2;PU500,15150;PD11000,15150;"
590 PH$="1,2,99,2,0,0.5,1,0,2,-0.5,1,-2,0,-0.5,-1,-99,1,-2,99,0,8;"
1000 REM
1010 REM UDSKRIVNING AF NAVN, DENSITET OG TEMPERATUR
1020 PRINT#2,"CS37;SI0.39,0.5;PU7325,14700;LBkg/m" {C "+CHR$(3);
1030 PRINT#2,"CS;PU8261,14700;CP0,0.25;LB3"+CHR$(3);
1040 PRINT#2,"CF;PU1000,14700;LB"+NAVN$+CHR$(3);
1050 PRINT#2,"PU6155,14700;LB";
```

```
1060 PRINT#2,USING "#####";DENS;
1065 PRINT#2,CHR$(3);
1070 PRINT#2,"PU8964,14700;LB";
1080 PRINT#2,USING "##.#";TEMP;
1085 PRINT#2,CHR$(3);
1090 PRINT#2,"PU1500,14500;PD9000,14500,9000,9500,1500,9500,1500,14500;PU;"  
1500 REM
1510 REM    UDTEGNING AF RAMME, NET OG AKSETEKSTER
1530 IF UMAX>0 AND UMAX<=1 THEN UNET=1
1540 IF UMAX>1 AND UMAX<=10 THEN UNET=2
1550 IF UMAX>10 AND UMAX<=20 THEN UNET=3
1560 IF UMAX>20 THEN UNET=4
1570 ON UNET GOSUB 4510,5010,5510,6010
1590 YTEKST$="moisture content u - weight per cent"
1600 PRINT#2,"CP;SP2;SI0.225,0.375;PU750,9570;DIO,1;LB"+YTEKST$+CHR$(3);
1610 XTEKST$="relative humidity - %"
1620 PRINT#2,"DI;PU3890,8750;LB"+XTEKST$+CHR$(3);
1630 PRINT#2,"PU6320,8750;UC"+PH$+CHR$(3);
1640 FOR K=10 TO 90 STEP 10
1650 X=1500+K*75
1660 PRINT#2,"SP1;PU";X,9500;"PD";X,14500;
1670 NEXT K
1680 FOR K=0 TO 100 STEP 20
1690 X=1500+K*75
1700 PRINT#2,"SP2;PU";X,9500;"CP-2,-1;LB";K;CHR$(3);
1710 NEXT K
2000 REM
2010 REM    UDTEGNING AF MALEPUNKTER OG KURVER
2020 N=0
2030 FOR I=1 TO NKURV
2040 IF K(I)=3 THEN N=N+1
2050 ON K(I) GOSUB 6510,7010,8010,7510
2060 NEXT I
2250 REM
2255 DEC=1
2260 FOR I=1 TO NKURV
2270 IF (K(I)=1 AND U(I,1)<10) OR (K(I)=2 AND U(I,NPKT(I))<10) OR (K(I)=4 AND
U(I,NPKT(I))<10) THEN DEC=2
2280 NEXT I
2500 REM
2510 REM    UDSKRIVNING AF SORPTIONSDATA
2520 TEKST$(1)="+ measured desorption values"
2530 TEKST$(2)="o measured adsorption values"
2540 TEKST$(3)=" measured scanning values"
2550 TEKST$(4)="* measured sorption values"
2560 N=0
2570 FOR I=1 TO NKURV
2580 IF K(I)=3 THEN N=N+1
2590 ON K(I) GOSUB 8510,9010,10010,9500
2600 NEXT I
2610 IF N=0 THEN PRINT#2,"PU500,5200;LBNo scanning values";CHR$(3);
3000 REM
```

```
3010 PRINT#2,"PU500,3000;LBNotes: "+CHR$(3);
3020 PRINT#2,"PU1655,3000;LB"+BEM$(1)+CHR$(3);
3030 PRINT#2,"PU1655,2750;LB"+BEM$(2)+CHR$(3);
3040 PRINT#2,"PU1655,2500;LB"+BEM$(3)+CHR$(3);
3500 REM
3510 PRINT#2,"PU500,2000;LBLitterature: "+CHR$(3);
3520 PRINT#2,"PU2645,2000;LB"+LITT$(1)+CHR$(3);
3530 PRINT#2,"PU2645,1750;LB"+LITT$(2)+CHR$(3);
3540 PRINT#2,"PU2645,1500;LB"+LITT$(3)+CHR$(3);
4000 REM
4010 PRINT#2,"PU500,1000;LBDDate: - -"+CHR$(3);
4020 PRINT#2,"PU1175,1000;LB";
4030 PRINT#2,USING "###";D,M,Y;
4040 PRINT#2,CHR$(3);
4050 PRINT#2,"PU4000,1000;LBInitials: "+INIT$+CHR$(3);
4060 PRINT#2,"PU7500,1000;LBFile: "+FIL$+CHR$(3);
4070 PRINT#2,"SP0;";
4480 CLOSE#2
4490 END
4500 REM
4510 REM VANDRET NET PA 0.1%
4520 LYAX=5000/UMAX
4530 PRINT#2,"SP1;"
4540 FOR K=.1 TO UMAX STEP .1
4550 PRINT#2,"PU";1500,9500+K*LYAX;"PD"9000,9500+K*LYAX;
4560 NEXT K
4570 PRINT#2,"SP2;"
4580 FOR K=0 TO UMAX STEP .1
4590 PRINT#2,"S10.275,0.375;PU";1500,9500+K*LYAX;"CP-3,0;LB";
4600 PRINT#2,USING "#.#";K;
4610 PRINT#2,CHR$(3);
4620 NEXT K
4630 RETURN
5000 REM
5010 REM VANDRET NET PA 1.0%
5020 LYAX=5000/UMAX
5030 PRINT#2,"SP1;"
5040 FOR K=1 TO UMAX-1
5050 PRINT#2,"PU";1500,9500+K*LYAX;"PD";9000,9500+K*LYAX;
5060 NEXT K
5070 PRINT#2,"SP2;"
5080 FOR K=0 TO UMAX-1
5090 PRINT#2,"S10.275,0.375;PU";1500,9500+K*LYAX;"CP-3,0;LB";
5100 PRINT#2,USING "##";K;
5110 PRINT#2,CHR$(3);
5120 NEXT K
5130 RETURN
5500 REM
5510 REM VANDRET NET PA 2.0%
5520 LYAX=5000/UMAX
5530 PRINT#2,"SP1;"
```

```
5540 FOR K=2 TO UMAX-1 STEP 2
5550 PRINT#2,"PU",1500,9500+K*LYAX;"PD",9000,9500+K*LYAX,
5560 NEXT K
5570 PRINT#2,"SP2;"
5580 FOR K=0 TO UMAX-1 STEP 2
5590 PRINT#2,"SI0.275,0.375;PU";1500,9500+K*LYAX;"CP-3,0;LB";
5600 PRINT#2,USING "##";K;
5610 PRINT#2,CHR$(3);
5620 NEXT K
5630 RETURN
6000 REM
6010 REM VANDRET NET PA 5.0%
6020 LYAX=5000/UMAX
6030 PRINT#2,"SP1;"
6040 FOR K=5 TO UMAX-1 STEP 5
6050 PRINT#2,"PU",1500,9500+K*LYAX;"PD",9000,9500+K*LYAX,
6060 NEXT K
6070 PRINT#2,"SP2;"
6080 FOR K=0 TO UMAX-1 STEP 5
6090 PRINT#2,"SI0.275,0.375;PU";1500,9500+K*LYAX;"CP-3,0;LB";
6100 PRINT#2,USING "##";K;
6110 PRINT#2,CHR$(3);
6120 NEXT K
6130 RETURN
6500 REM
6510 REM UDTEGNING AF DESORPTIONSISOTERM
6520 PRINT#2,"IW1500,9500,9000,14500;SM+;SP2;PA"
6530 FOR J=1 TO NPKT(I)
6540 PRINT#2,1500+PHI(I,J)*75,9500+LYAX*U(I,J)
6550 NEXT J
6560 UA(20)=A(I,1)*EXP((-1/A(I,2))*LOG(1-LOG(.2)/A(I,3)))
6570 PRINT#2,"SM;PU";1500+20*75,9500+LYAX*UA(20);"PD"
6580 FOR K=22 TO 98 STEP 2
6590 UA(K)=A(I,1)*EXP((-1/A(I,2))*LOG(1-LOG(K/100)/A(I,3)))
6600 PRINT#2,1500+K*75,9500+LYAX*UA(K)
6610 NEXT K
6620 PRINT#2,"PU,IW,"
6630 RETURN
7000 REM
7010 REM UDTEGNING AF ADSORPTIONSISOTERM
7020 PRINT#2,"IW1500,9500,9000,14500;SMo;SP2;PA"
7030 FOR J=1 TO NPKT(I)
7040 PRINT#2,1500+PHI(I,J)*75,9500+LYAX*U(I,J)
7050 NEXT J
7060 UA(20)=A(I,1)*EXP((-1/A(I,2))*LOG(1-LOG(.2)/A(I,3)))
7070 PRINT#2,"SM;PU";1500+20*75,9500+LYAX*UA(20);"PD"
7080 FOR K=22 TO 98 STEP 2
7090 UA(K)=A(I,1)*EXP((-1/A(I,2))*LOG(1-LOG(K/100)/A(I,3)))
7100 PRINT#2,1500+K*75,9500+LYAX*UA(K)
7110 NEXT K
7120 PRINT#2,"PU,IW,"
```

```
7130 RETURN
7500 REM
7510 REM    UDTEGNING AF UKENDT SORPTIONSISOTERM
7520 PRINT#2,"IW1500,9500,9000,14500;SM*,SP2;PA"
7530 FOR J=1 TO NPKT(I)
7540 PRINT#2,1500+PHI(I,J)*75,9500+LYAX*U(I,J)
7550 NEXT J
7560 UA(20)=A(I,1)*EXP((-1/A(I,2))*LOG(1-LOG(.2)/A(I,3)))
7570 PRINT#2,"SM;PU";1500+20*75,9500+LYAX*UA(20);"PD"
7580 FOR K=22 TO 98 STEP -2
7590 UA(K)=A(I,1)*EXP((-1/A(I,2))*LOG(1-LOG(K/100)/A(I,3)))
7600 PRINT#2,1500+K*75,9500+LYAX*UA(K)
7610 NEXT K
7620 PRINT#2,"PU;IW;"
7630 RETURN
8000 REM
8010 REM    UDTEGNING AF SCANNINGISOTERM(ER)
8020 ON N GOSUB 8120,8130,8140,8150
8030 PRINT#2,"SP1;PA"
8040 FOR J=1 TO NPKT(I)
8050 PRINT#2,1500+PHI(I,J)*75,9500+LYAX*U(I,J)
8060 NEXT J
8070 PRINT#2,"SM;LT2,1;PU";1500+PHI(I,1)*75,9500+LYAX*U(I,1)
8080 FOR J=2 TO NPKT(I)
8090 PRINT#2,"PD";1500+PHI(I,J)*75,9500+LYAX*U(I,J)
8100 NEXT J
8110 PRINT#2,"PU;"
8120 PRINT#2,"SM1;":RETURN
8130 PRINT#2,"SM2;":RETURN
8140 PRINT#2,"SM3;":RETURN
8150 PRINT#2,"SM4;":RETURN
8160 RETURN
8500 REM
8510 REM    DESORPTIONSISOTERM
8520 PRINT#2,"SM;CP;SP1;PU500,8400,SI0.225,0.35;LB"TEKST$(1)+CHR$(3);
8530 PRINT#2,"PU500,8050;UC"+PH$+CHR$(3);
8540 FOR K=1 TO NPKT(I)
8550 PRINT#2,"PU";770+(K-1)*675,8050;"LB";
8560 PRINT#2,USING "##.##";PHI(I,K);
8570 PRINT#2,CHR$(3);
8580 NEXT K
8590 PRINT#2,"PU500,7800;LBu"+CHR$(3);
8600 FOR K=1 TO NPKT(I)
8610 PRINT#2,"PU";770+(K-1)*675,7800;"LB";
8620 IF DEC=1 THEN PRINT#2,USING "##.##";U(I,K);
8625 IF DEC=2 THEN PRINT#2,USING "#.##";U(I,K);
8630 PRINT#2,CHR$(3);
8640 NEXT K
8650 PRINT#2,"PU500,7550;LBApproximation: "+CHR$(3);
8660 PRINT#2,"PU500,7300;LBu= "+CHR$(3);
8680 PRINT#2,"CP;PU";905,7300;"LB";
```

```
8690 PRINT#2,USING "##.##^^^^";A(I,1)
8700 PRINT#2,CHR$(3);
8710 PRINT#2,"PU";2120,7300;"LB*exp((-1/"+CHR$(3));
8720 PRINT#2,"PU";3335,7300;"LB";
8725 PRINT#2,USING "#.##";A(I,2)
8730 PRINT#2,CHR$(3);
8740 PRINT#2,"PU";3875,7300;"LB)*ln(1-ln("+CHR$(3);
8745 PRINT#2,"UC"+PH$+CHR$(3);"LB)"/"+CHR$(3);
8750 PRINT#2,"PU";5630,7300;"LB";
8760 PRINT#2,USING "##.##^^^^";A(I,3)
8765 PRINT#2,CHR$(3);
8766 PRINT#2,"PU";6845,7300;"LB))"+CHR$(3);
8770 RETURN
9000 REM
9010 REM    ADSORPTIONSISOTERM
9020 PRINT#2,"CP;SP1;PU500,6800;SI0.225,0.35;LB"TEKST$(2)+CHR$(3);
9030 PRINT#2,"PU500,6450;UC"+PH$+CHR$(3);
9040 FOR K=1 TO NPKT(I)
9050 PRINT#2,"PU";770+(K-1)*675,6450;"LB";
9060 PRINT#2,USING "##.##";PHI(I,K);
9070 PRINT#2,CHR$(3);
9080 NEXT K
9090 PRINT#2,"PU500,6200;LBu"+CHR$(3);
9100 FOR K=1 TO NPKT(I)
9110 PRINT#2,"PU";770+(K-1)*675,6200;"LB";
9120 IF DEC=1 THEN PRINT#2,USING "##.##";U(I,K);
9125 IF DEC=2 THEN PRINT#2,USING "#.##";U(I,K);
9130 PRINT#2,CHR$(3);
9140 NEXT K
9150 PRINT#2,"PU500,5950;LBApproximation?"+CHR$(3);
9160 PRINT#2,"PU500,5700;LBu= "+CHR$(3);
9180 PRINT#2,"CP;PU";905,5700;"LB";
9190 PRINT#2,USING "##.##^^^^";A(I,1)
9200 PRINT#2,CHR$(3);
9210 PRINT#2,"PU";2120,5700;"LB*exp((-1/"+CHR$(3);
9220 PRINT#2,"PU";3335,5700;"LB";
9225 PRINT#2,USING "#.##";A(I,2)
9230 PRINT#2,CHR$(3);
9240 PRINT#2,"PU";3875,5700;"LB)*ln(1-ln("+CHR$(3);"UC"+PH$+CHR$(3);"LB)"/"+CHR$(3);
9250 PRINT#2,"PU";5630,5700;"LB";
9260 PRINT#2,USING "##.##^^^^";A(I,3)
9265 PRINT#2,CHR$(3);
9266 PRINT#2,"PU";6845,5700;"LB))"+CHR$(3);
9270 RETURN
9500 REM
9510 REM    UKENDT SORPTIONSISOTERM
9520 PRINT#2,"CP;SP1;PU500,8400;SI0.225,0.35;LB"TEKST$(4)+CHR$(3);
9530 PRINT#2,"PU500,8050;UC"+PH$+CHR$(3);
9540 FOR K=1 TO NPKT(I)
9550 PRINT#2,"PU";770+(K-1)*675,8050;"LB";
9560 PRINT#2,USING "##.##";PHI(I,K);
```

```
9570 PRINT#2,CHR$(3);
9580 NEXT K
9590 PRINT#2,"PU500,7800;LBu"+CHR$(3);
9600 FOR K=1 TO NPKT(I)
9610 PRINT#2,"PU";770+(K-1)*675,7800;"LB";
9620 IF DEC=1 THEN PRINT#2,USING "##.##";U(I,K);
9625 IF DEC=2 THEN PRINT#2,USING ".##";U(I,K);
9630 PRINT#2,CHR$(3);
9640 NEXT K
9650 PRINT#2,"PU500,7550;LBApproximation:"+CHR$(3);
9660 PRINT#2,"PU500,7300;LBu= "+CHR$(3);
9680 PRINT#2,"CP;PU";905,7300;"LB";
9690 PRINT#2,USING "##.##^^^^";A(I,1)
9700 PRINT#2,CHR$(3);
9710 PRINT#2,"PU";2120,7300;"LB*exp((-1/"+CHR$(3));
9720 PRINT#2,"PU";3335,7300;"LB";
9725 PRINT#2,USING ".##";A(I,2)
9730 PRINT#2,CHR$(3);
9740 PRINT#2,"PU";3875,7300;"LB)*ln(1-ln("+CHR$(3);"UC"+PH$+CHR$(3);"LB)/*"+CHR$(3);
9750 PRINT#2,"PU";5630,7300;"LB";
9760 PRINT#2,USING "##.##^^^^";A(I,3)
9765 PRINT#2,CHR$(3);
9766 PRINT#2,"PU";6845,7300;"LB))"+CHR$(3);
9770 RETURN
10000 REM
10010 REM UDSKRIVNING AF SCANNINGKURVER
10020 ON N GOSUB 10180,10190,10200,10210
10030 PRINT#2,"SP1;PU";X0,Y0;"SI0.225,0.35;LB";
10040 PRINT#2,USING "#";N;
10050 PRINT#2,TEKST$(3);CHR$(3);
10060 PRINT#2,"PU";X0,Y0-350;"UC"+PH$+CHR$(3);
10070 FOR K=1 TO NPKT(I)
10080 PRINT#2,"PU";X0+270+(K-1)*675,Y0-350;"LB";
10090 PRINT#2,USING "##.##";PHI(I,K);
10100 PRINT#2,CHR$(3);
10110 NEXT K
10120 PRINT#2,"PU";X0,Y0-600;"LBu"+CHR$(3);
10130 FOR K=1 TO NPKT(I)
10140 PRINT#2,"PU";X0+270+(K-1)*675,Y0-600;"LB";
10150 IF DEC=1 THEN PRINT#2,USING "##.##";U(I,K);
10155 IF DEC=2 THEN PRINT#2,USING ".##";U(I,K);
10160 PRINT#2,CHR$(3);
10170 NEXT K
10180 X0=500:Y0=5200:RETURN
10190 X0=6400:Y0=5200:RETURN
10200 X0=500:Y0=4100:RETURN
10210 X0=6400:Y0=4100:RETURN
10220 RETURN
Ok
```

APPENDIX E. Udskrift af de på SORPTION værende filer

Volume in drive B is SORPTION
Directory of B:\

A_B	<DIR>	3-25-86	2:49p
C	<DIR>	3-25-86	2:49p
D_K	<DIR>	3-25-86	2:49p
L	<DIR>	3-25-86	2:49p
M_U	<DIR>	3-25-86	2:49p
V_Z	<DIR>	3-25-86	2:49p
UDTEGN	BAS	9314	4-11-86 2:38p
DATAIND	BAS	2740	4-14-86 10:01a
SORPF	PAS	11008	3-25-86 2:11p
DESORPF	PAS	10953	3-20-86 3:20p
10 File(s) 191488 bytes free			

Note: Materialefilerne findes i de viste sub-directories.
Eksempel: En materialefil fås ved \A_B\BRICK20.186.