BMBF – geförderte Forschung
Förderkennzeichen: 02 WA 0016

Schlußbericht

zum
Forschungsvorhaben

In situ Filterregeneration bei der TNT-Elimination aus Grundwasser: Anwendung des Verfahrens auf einen Faseraktivkohle-Adsorber

Projektleitung:

Prof. Dr.-Ing. W. Hegemann
Technische Universität Berlin
Institut für Technischen Umweltschutz
Fachgebiet Siedlungswasserwirtschaft

Dr.-Ing. M. Hoff
Prof. Dr.-Ing. W. Hegemann

Berlin, April 2003

Inhalt

1 Einleitung

Das Forschungsvorhaben *„In-situ Filterregeneration bei der TNT-Elimination aus Grundwasser: Anwendung des Verfahrens auf einen Faseraktivkohle-Adsorber“* ist aus dem ebenfalls BMBF-geförderten Vorhaben *„Entwicklung eines Kombinationsverfahrens zur physikalischen, chemischen und biologischen Reinigung von mit Nitrotoluolen verunreinigten Grundwässern“* (Laufzeit: 1995-1998; Förderkennzeichen: 02WA9445/5) hervorgegangen. Die Laufzeit dieses Vorhabens betrug 12 Monate (01.08.2001 - 31.08.2002)

Der Zwischenbericht vom 31.05.2002 enthält bereits die wesentlichen laborexperimentellen Arbeiten einschließlich Material und Methoden zu diesem Vorhaben. Er ist als Anhang Bestandteil dieses Berichts. Die Ergebnisse der Untersuchungen zum Einsatz von Faseraktivkohle (ACF) als Adsorbens in chemisch regenerierbaren Adsorbern werden hier in der abschließenden Diskussion zusammengefaßt

Der zweite Schwerpunkt des vorliegenden Schlußberichts ist die Weiterentwicklung des Adsorber-Berechnungsverfahrens bzw. dessen Umsetzung in eine Software. Dargestellt wird die Anpassung an faserförmiges Adsorbens und die Gewinnung der ACF-spezifischen Größen zur Berechnung des Adsorptionsvorgangs. Die durchgeführten Durchbruchskurven (DBK) -Messungen wurden mit der neuen Software modelliert und sind ebenfalls dargestellt. Im Anhang findet sich die vollständige Dokumentation des Source-Codes der Software zum Berechnungsverfahren. Die Ausführungen zum Berechnungsverfahren verstehen sich als Weiterführung der bisherigen Arbeiten [1 + 3] zu diesem Thema, die hier nicht wiederholt werden sollen.

2 Anpassung des Berechnungsverfahrens an faserförmiges Adsorbens

Das im Vorgängerprojekt für granuliertes Adsorbens entwickelte Adsorberberechnungsverfahren [1, 2] wurde zur Berechnung von faserförmigen Adsorbens erweitert. Die Geometrie des Adsorbens muß bei der Berechnung des Stoffübergangs berücksichtigt werden und für die Adsorbensdichte (Scheindichte) muß eine Bestimmungsgleichung abgeleitet werden, da die direkte Bestimmung nach der Quecksilbermethode aufgrund der feinen Gewebestruktur nur ungenaue Werte liefert:

Die Faseroberfläche *A* berechnet sich analog der eines Zylinders aus Radius *R* und Länge *L*

$$A = 2pR(R + L) \tag{1}$$

für $L >> R$ (lange Fasern) gilt

$$A = 2pRL \tag{2}$$

Mit der allgemeinen Dichtegleichung (Masse *m*, Volumen *V*)

$$r = \frac{m}{V} \tag{3}$$

und dem Faservolumen (entspricht dem Zylindervolumen)

$$V = 2pR^2 L \tag{4}$$

erhält man [(4) in (3), nach *L* aufgelöst]

$$L = \frac{m}{pR^2 r} \tag{5}$$

und somit für die gesamte Faseroberfläche [(5) in (2)]

$$\boxed{A = \frac{2m}{Rr}} \tag{6}$$

Gleichung (6) zur Berechnung der Faseroberfläche unterscheidet sich von der Gleichung zur Berechnung von kugelförmigen Adsorbens nur um den Faktor 3/2 und muß im Berechnungsverfahren bei der Beschreibung des Stoffübergangs *dy* berücksichtigt werden.

$$dy = \frac{b \cdot \Delta c \cdot dt}{m_{Adsorbens}} \cdot A_{Adsorbens} \tag{7}$$

Das Programm zur Adsorberberechnung muß also lediglich um eine Fallunterscheidung (Faser oder Granulat) erweitert werden, um eine der beiden folgenden Gleichungen (8) oder (9) einzusetzen (mit Stoffübergangskoeffizient b, Konzentrationsgradient am Phasenübergang Δc und differentiellem Zeitschritt dt):

$$dy_{Korn} = b_{Korn} \cdot \Delta c \cdot dt \cdot \frac{3}{R_{Korn} \cdot r_{Korn}} \qquad dy_{Faser} = b_{Faser} \cdot \Delta c \cdot dt \cdot \frac{2}{R_{Faser} \cdot r_{Faser}} \tag{8+9}$$

Die Dichte der Einzelfaser kann durch mikroskopisches Auszählen der Einzelfasern eines Faserbündels bekannter Länge und Gewichts ermittelt werden:

$$n_{F/B} = \frac{L_{Faser,gesamt}}{L_{Faserbündel}} \tag{10}$$

(10) aufgelöst nach $L_{Faser,gesamt}$ ergibt mit (5):

$$r = \frac{m}{pR^2 n_{F/B} L_{Faserbündel}} \tag{11}$$

Tabelle 1: Kennzahlen und Stoffgrößen von granulierter und faserförmiger Aktivkohle im Vergleich (ACF: CS 1501, GAC: F-400; Adsorberabmessungen und Zulauf in beiden Filterversuchen gleich)

Parameter	Dimension	ACF	GAC	ACF / GAC
gemessen				
Adsorbensform	[-]	Gewebe	Granulat	
Adsorbensradius	m	4,500e-06	3,610e-04	1%
Adsorbensbettdichte	kg/m³	1,717e+02	4,780e+02	36%
Adsorbensdichte (Hg-Dichte)	kg/m³	9,820e+02	8,100e+02	121%
Konstante der Freundlich-Isotherme	L/g	6,620e+00	2,320e+00	285%
Exponent der Freundlich-Isotherme	[-]	1,911e-01	9,500e-02	201%
Zulaufkonzentration	mg/L	8,81		
Adsorbensbettlänge	mm	19,00		
Volumenstrom	L/h	0,133		
Adsorberradius	m	2,000e-03		
berechnet				
Leerrohrgeschwindigkeit	m/h	10,6		
hydrodynamische Verweilzeit	s	5,3	2,6	204%
Adsorbensmasse	kg	4,100e-05	1,141e-04	36%
Hold-Up des Adsorbensbettes (flüssig)	m³	1,970e-07	9,786e-08	201%
Zwischenraumgeschwindigkeit	m/s	3,568e-03	7,184e-03	50%
Stoffübergangkoeffizient	m/s	2,961E-04	3,205E-05	924%
Reynoldszahl	[-]	3,211e-02	5,187e+00	1%
Sherwoodzahl	[-]	4,284E+00	3,720E+01	12%

Für das hier eingesetzte ACF-Gewebe der Type CS 1501 liegt als Literaturangabe für den Durchmesser der Einzelfaser nur ein Richtwert (10 µm) vor. Genauere Werte wurden daher mikroskopisch ermittelt (Meßwerte: 8,5 bis 9,5 µm). Für den Mittelwert $\boxed{R = 4{,}5mm}$ und der Zählung $\boxed{n_{F/B} = 625}$ ergibt sich für die Dichte der ACF der Wert $\boxed{r = 9{,}82\ \mathrm{E}\text{-}02\ \mathrm{kg/m^3}}$.

Mit diesen Werten können nun der Stoffübergangskoeffizient und alle weiteren zur Adsorberberechnung notwendigen Kennzahlen berechnet werden (was vom Programm automatisch durchgeführt wird). Tabelle 1 zeigt den Vergleich mit granulierter Aktivkohle: Trotz der kleinen Reynoldszahl ist der Stoffübergangskoeffizient fast zehnmal so groß wie der des Aktivkohlegranulats (GAC). Die etwas geringere Dichte von ACF führt zwar über die verminderte Zwischenraumgeschwindigkeit zu einer kleineren Reynoldszahl, aber erhöht gleichzeitig die Verweilzeit, was eine effizientere Nutzung des Adsorbens bedeutet.

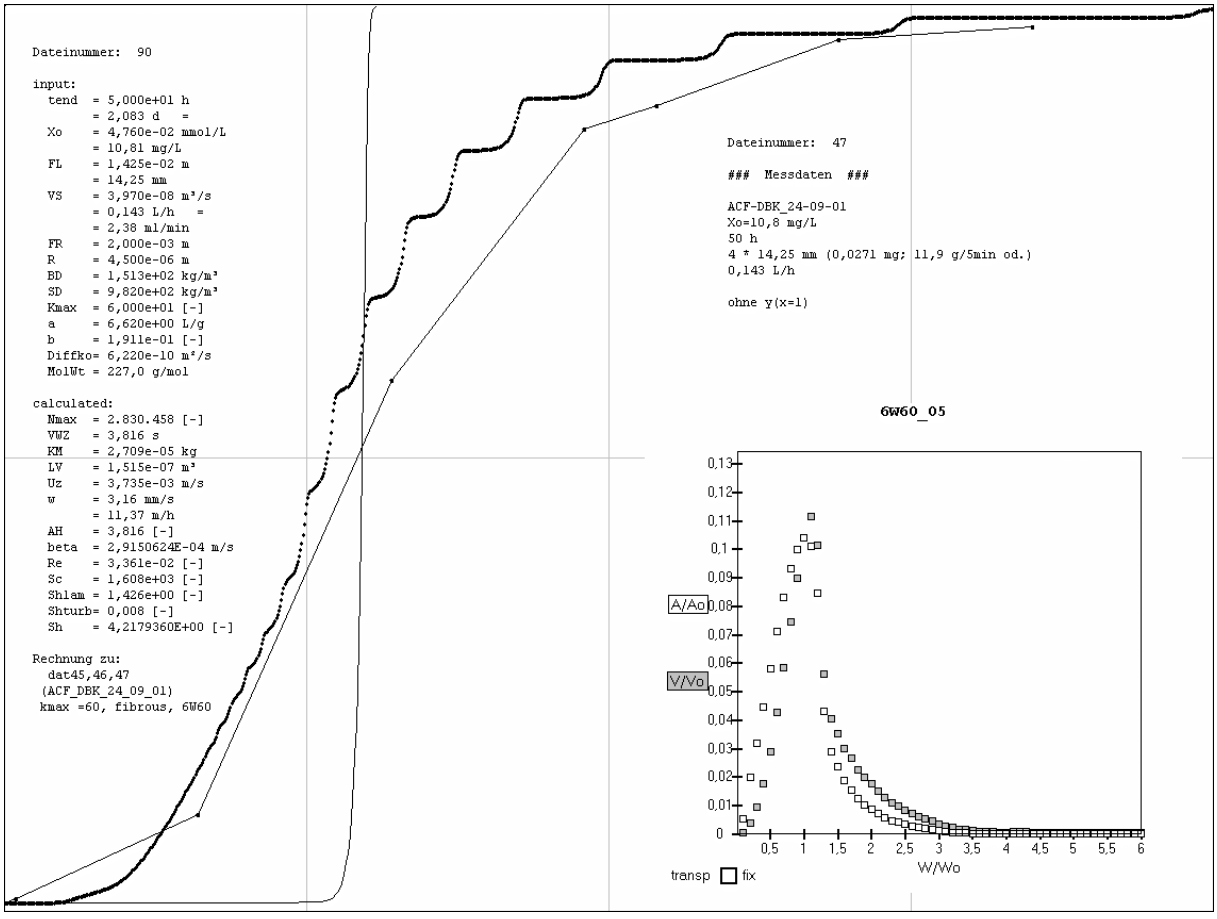


Bild 1: Meßwerte (Punkt-Linie) und Rechnung (fein: einfache Rechnung; fett: Berücksichtigung des abgebildeten Geschwindigkeitsprofils im Adsorber) der Durchbruchskurve von TNT_{aq} an einem 14 mm –Laboradsorber

Bild 1 und Bild 2 geben die Modellierung gemessener Durchbruchskurven wieder. Der gewellte Verlauf (Kurven sind nicht geglättet) der Durchbruchskurven mit Berücksichtigung der Geschwindigkeitsverteilung hat seine Ursache in der Mittelung der steil verlaufenden Durchbruchskurvenschar und kann durch eine Erhöhung der Anzahl Geschwindigkeitsbereiche ausgeglichen werden.

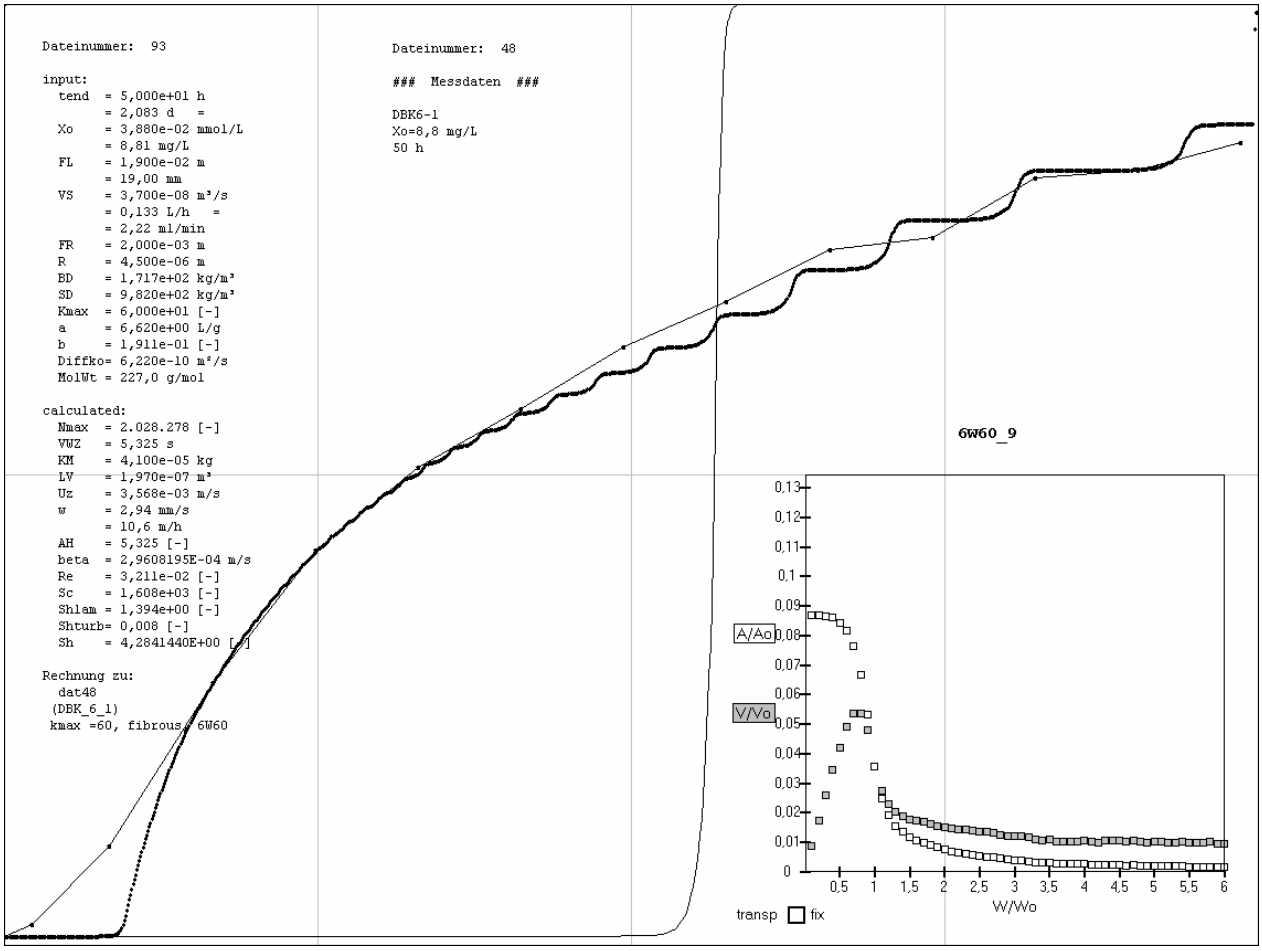


Bild 2: Berechnung des für die Versuche zur Wiederholten Regeneration eingesetzten Adsorbers (Adsorberbettlänge: 19 mm).

3 Abschließende Diskussion zum Einsatz von Faseraktivkohle (ACF) als Adsorbens in chemisch regenerierbaren Adsorbern

ACF ist im Vergleich zu anderen Aktivkohlen ein technisch herausragendes Adsorbens, da es hohe Beladungskapazitäten und eine sehr schnelle Adsorptionskinetik bei gleichzeitig guter Durchströmbarkeit aufweist. Beim Einsatz von ACF müssen diese Eigenschaften gegenüber ihrem sehr hohem Preis im Vergleich zu anderen Aktivkohlen abgewogen werden. Dies ist nicht zwangsläufig eine reine Entscheidung zwischen Performance und Kosten, denn die kinetischen und kapazitiven Eigenschaften von ACF können auch zu erheblichen Einsparungen führen, da Adsorber und Fördereinrichtungen deutlich kleiner zu dimensionieren sind und die adsorbierten Stoffe in einer höheren Konzentration auf dem Adsorbens vorliegen.

Bei Einsatz in chemisch regenerierbaren Adsorbern sind diese Kriterien von besonderer Bedeutung. Die Adsorber sollten hier nicht auf lange Standzeiten sondern auf häufige Regenerationszyklen ausgelegt werden, um die nachfolgenden Verfahrensschritte zur Ab- und Prozeßwasserbehandung ohne umfangreiche Zwischenspeichereinrichtungen möglichst kontinuierlich zu beaufschlagen, damit sie entsprechend klein ausgelegt werden können. Ermöglicht wird dies durch die hohen Beladungskapazitäten und die steilen Beladungsverläufe (kurze Stoffübergangszonen) als Folge der guten Kinetik.

Die kurzen Diffusionsstrecken, die auch die Ursache der schnellen Adsorptionskinetik ist, sollte auch bei der Regeneration der ACF deutliche Vorteile bringen, da die Diffusion aus dem Adsorbens in das umgebene Fluid bei granulierter Aktivkohle ein geschwindigkeitslimitierender Schritt ist. Unklar war bisher, ob die Desorption analog der GAC-Regeneration gelingt und ob die Adsorptionseigenschaften nach erfolgter Desorption wieder hergestellt werden können.

Die vorliegenden Untersuchungen konnten alle Annahmen bestätigen, die Grundlage der Vorhabens-Initiierung waren. Neben der gemessenen Kinetik und den Isothermen spiegelt vor allem das Durchbruchsverhalten bei Säulenversuchen mit praktisch vernachlässigbaren Schlupfkonzentrationen die adsorptiven Eigenschaften des ACF wider. Dasselbe gilt für die Durchströmbarkeit von ACF-Betten, deren Druckverlust kaum meßbar war, sowie für die damit verbundene Kompressionsstabilität und der Abwesenheit von Adsorbensabrieb. Diese Eigenschaften blieben auch nach wiederholter Anwendung der untersuchten Regenerationsmethoden unverändert.

Die Regeneration der ACF-Adsorber führt zu ähnlichen Abwässern, wie sie von der Behandlung von GAC-Adsorbern bekannt ist. Diese konnten somit mit den etablierten Prozeß- und Abwasseraufbereitungskomponenten unveränderter Konfiguration weiterbehandelt werden.

Die eigentliche Regeneration verlief nicht im selben Maße beschleunigt wie die Vorgänge der Adsorption, was die ursprüngliche Überlegung stützt, daß die beschleunigende Wirkung der kurzen Diffusionsstrecken durch die aus den Isothermen ableitbare höhere Bindungsstärke zwischen Adsorbens und Adsorptiv teilweise kompensiert wird. Die erwünscht hohe Konzentration der Regenerationsabwässer liegt somit als Folge der einzusetzenden Volumina an Regenerationslauge und Spülwässer nur wenig über den Werten der GAC-Regeneration.

Problematisch stellte sich zunächst die Wiederherstellung der adsorptiven Eigenschaften der ACF dar, was später auf die Konditionierung zurückgeführt werden konnte. Als nicht übertragbar erwiesen sich die Bedingungen, unter denen GAC optimal konditioniert wird. Obwohl alternative Konditionierungsvorschriften entwickelt wurden, mit denen um ein Vielfaches bessere Ergebnisse erzielt werden, ist die Wiederherstellung der adsorptiven Eigenschaften von ACF nach der Laugenbehandlung neben der mäßigen Desorptiongeschwindigkeit die Komponente des Verfahrens, an der die herausragend positiven Eigenschaften des ACF gemessen werden müssen. Es ist bisher kein Weg gefunden worden, nach einer Regeneration die Adsorptionseigenschaften unbehandelter ACF wieder vollständig

herzustellen. Nach der ersten Regeneration ist vielmehr eine erhebliche Kapazitäts-Verringerung der ACF zu kalkulieren, die in weiteren Adsorptions-Regenerationszyklen allerdings stagniert.

Ob ein wirtschaftlicher Einsatz von ACF-Adsorbern für eine gegebene Aufgabe in der Wasserreinigung möglich ist, sollte daher anhand der Adsorptionsleistung des mindestens einmalig regenerierten ACF bestimmt werden. Für die weitere Auslegung des Verfahrens sind die Erfahrungen aus der Prozeß- und Abwasseraufbereitung von GAC-Adsorbern direkt übertragbar. Die entwickelte Software ist dazu das geeignete Hilfsmittel.

Literatur

1. **Hoff, M.; Hegemann, W. (1999):** Entwicklung eines Kombinationsverfahrens zur physikalischen, chemischen und biologischen Reinigung von mit Nitrotoluolen verunreinigten Grundwässern, Schlußbericht zum Forschungsvorhaben, Förderkennzeichen: 02WA9445/5, BMBF-Förderaktivität Gewässerschutz und Gewässerreinigung, Leistungsplan F21013 Gewässersanierung, Grundwasserschutz

2. **Hoff, M.; Hegemann, W. (1999):** In-situ Filterregeneration bei der TNT-Elimination aus Grundwasser: Anwendung des Verfahrens auf einen Faseraktivkohle-Adsorber, Antrag auf Förderung des Vorhabens, eingereicht beim Bundesministerium für Bildung und Forschung Projektträger Wassertechnologie und Schlammbehandlung (PtWT) Forschungszentrum Karlsruhe GmbH, Technik und Umwelt PtWT-Außenstelle Dresden

3. **Hoff, M. (2002):** Entwicklung einer in situ Adsorbensregeneration und mathematische Beschreibung des Adsorptionsvorgangs bei der Wasserreinigung auf Rüstungsaltlasten, Dissertation D 83, TU Berlin. Erschienen als Band Nr. 17 in der Schriftenreihe Berichte zur Siedlungswasserwirtschaft des Fachgebietes Siedlungswasserwirtschaft der Technischen Universität Berlin, ISBN 3-936812-17-9

4. **LeCloirec, P.; Brasquet, C.; Subrenat, E. (1997):** Adsorption Onto Fibrous Activated Carbon - Applications to Water-Treatment, Energy & Fuels 11/2: 331-336

Berlin, den 14.04.2003

Dr.-Ing. M. Hoff

Prof. Dr.-Ing. W. Hegemann

Anhang I _ DBK-Programm Dokumentation (Source-Code)

A) Kurzbeschreibung der DBK-Programm Komponenten (Interface)

B) Screenshot (Runtime)

C) Grafische Übersicht der Programmobjekte (Bezeichner)

D) Source-Code C++ Dynamic Link Library (dbk32.dll)

E) Source-Code VB-Programmteil (Objects, Form-Code, Module-Code)

A) Kurzbeschreibung der DBK-Programm Komponenten (Interface)

1) Grafischer DBK-Ausgabebereich

- Meßdaten
- Berechnung nach Filmdiffusionsmodell
- Berechnung mit VS-Profil
- grafische Optionen: Skalierung; Grafen-Kennzeichnung durch Strichstärke, -Farbe und Datenlabel; permanent/erasable-Modus; Bildexport (Excel, Word)

2) Daten Ein- und Ausgabe

- Anzeige aller Eingabe- und berechneter Parameter
- DBK-Datenverwaltung
- Dimensionsumrechung
- Datenexport (Excel, Word)
- Kommentierungsfunktion

3) Volumenstrom (VS)-Profil

- Profildatenverwaltung
- manuelle Vorgabe der Profilbreite (Anzahl + Größe der Geschwindigkeitsbereiche)
- manuelle Zuordnung der Adsorberteilquerflächen zu den Geschwindigkeitsbereichen
- iterative Korrektur der manuellen Profilvorgabe (Summe der Flächenanteile = 1 und Summe der Volumenstromanteile = 1) und Berechnung des Volumenstromprofils
- Berechnung der DBK-Schar gemäß Geschwindigkeitsprofil, Gewichtung mit VS-Profil, Mittelung und Ausgabe der VS-optimierten DBK
- Bildexport (Excel, Word) des Profils
- Glättungsfunktion

B) Screenshot (Runtime)

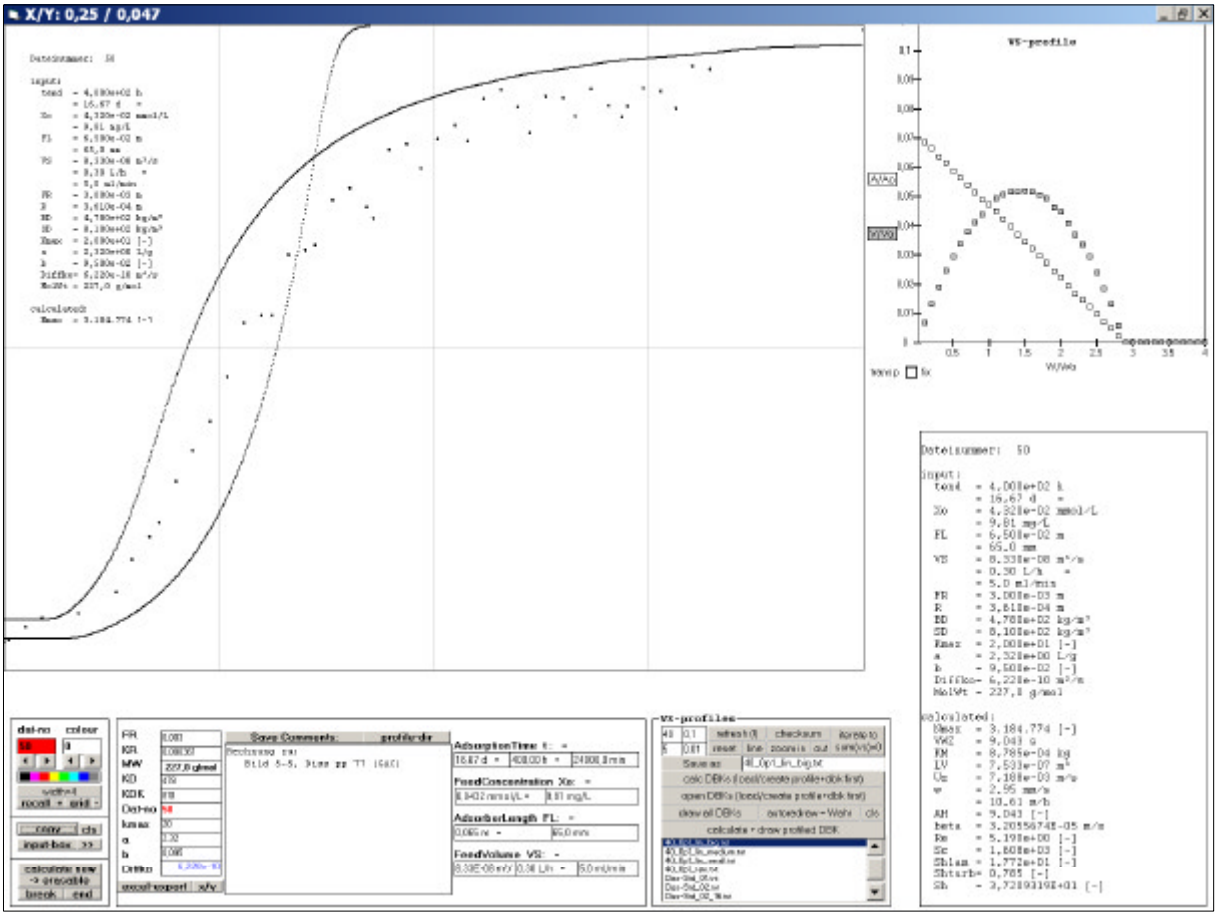
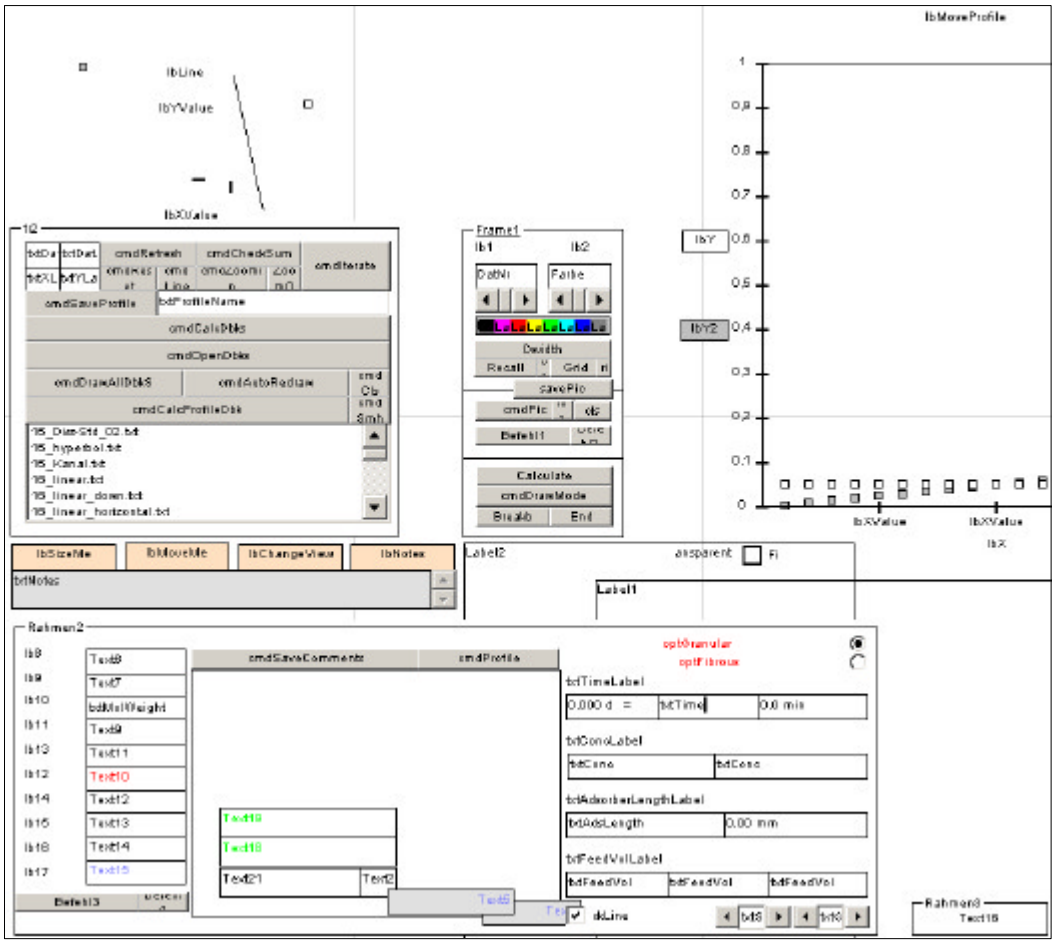
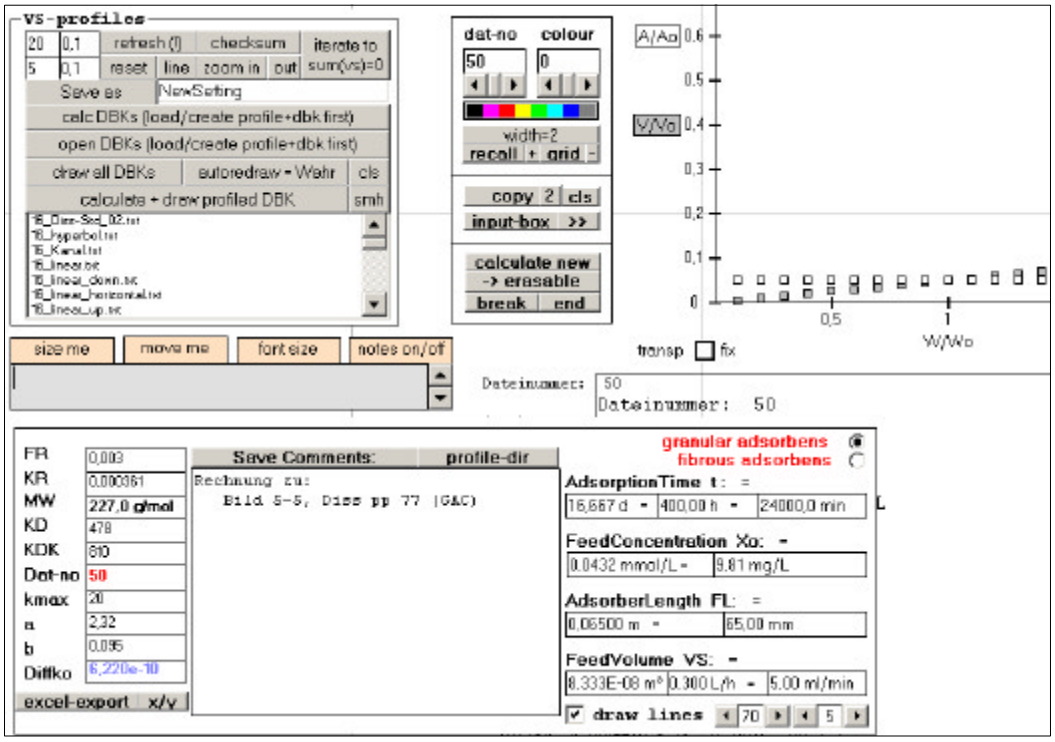


Bild 3: Screenshot der Software zur Adsorberberechnung

C) Grafische Übersicht der Programmobjekte (Bezeichner)



(A)



(B)

Bild 4: Übersicht vb-Objekte (A: Bezeichner, B: Runtime)

D) Source-Code C++ Dynamic Link Library (dbk32.dll)

Source-Code (C++ File dbk32.cpp) :

```
#include <windows.h>
#include <math.h>
#include <stdio.h>
void _stdcall  Calcdbk(double y[],double far *x,double aa,double cc,double
cv,double cf,double xn,long kmax)
{
double xr;
long k;
      *x=xn;
      { for (k=0; k<kmax; k++)
            {
            xr=pow((y[k]/aa),cc);
            y[k]=y[k]+cv*(*x-xr);
            *x=*x-cf*(*x-xr);
            }
      }
}
BOOL WINAPI DllMain (HINSTANCE hDLL, DWORD dwReason, LPVOID lpReserved)
{
return TRUE; }
```

Deklaration der C-Dll in VB:

```
Declare Sub Calcdbk Lib "dbk32.dll" (Y As Double, X As Double, ByVal aa As
Double,
      ByVal cc As Double, ByVal cv As Double, ByVal cf As Double, ByVal xn
      As  _ Double, ByVal kmax As Long)
```

Aufruf als Funktion in VB:

```
Calcdbk Y(0), X, aa, cc, cv, cf, xn, kmax
```

Ersetzter Code in VB:

```
x = xn                          'Zulaufkonzentration zur 1. Filterebene
For k = 1 To kmax               'Start innere Programmschleife (b)
    xr = (y(k) / aa) ^ cc       'Kornrandkonzentration in Filterebene k
    y(k) = y(k) + cv * (x - xr) 'neue Beladung der k-ten Filterebene
    x = x - cf * (x - xr)       'Zulaufkonzentration zur k+1-ten
Filterebene
Next k                          'nächste Filterebene (Sprung nach b)
```

E) Source-Code VB-Programmteil (Objects, Form-Code, Module-Code)

‘###### Form-Code #####

```
Dim s, zoomx, bb, Diffko, Break, tend, FL, vs,
beta, FR, AH, KD, R, KDK, Dateinummer,
Dateiname$
Dim aa As Double, cc As Double, xn As Double,
kmax As Long, tmp
Dim MeHeight, MeWidth
Dim SelectMemoryItem, SelectMemoryIndex
Dim OverWriteWarning As Boolean,
ProfileCalcRunning As Boolean
Dim RedrawP(), PCount, TempP(1), StdDir
Dim AreaArray(), VsArray(), DataCount,
XLabelFreq, YLabelDist, DataUnit,
VsProfilesDir 'vs-frame only
Dim MultiDbkArray()

Private Sub cmdAutoRedraw_Click()
Me.AutoRedraw = Not Me.AutoRedraw
cmdAutoRedraw.Caption = "autoredraw = " &
Me.AutoRedraw
End Sub

Private Sub Calculate_Click()
Dim cx, cy, KM, LV, VWZ, Uz, Re, Sc, Shlam,
Shturb, Sh, dKM, dLV, C1, C2, dt, nmax, k, n,
yk, xk
Dim Y() As Double
Dim X As Double
Dim cv As Double, cf As Double
Dateiname$ = "dat" + LTrim$(Str$(Text10)) +
".dat"
If FileExist(Dateiname$) Then
   If OverWriteWarning = True Then
  If MsgBox(Dateiname$ & " exists - replace?",
vbOKCancel, "confirm overwrite") = vbCancel
Then Exit Sub
   End If
   Kill Dateiname$
End If
cmdSaveComments_Click
AutoRedraw = False
cx = MeWidth
cy = MeHeight
Line (0.98 * cx, 0.98 * cy)-(0.99 * cx, 0.99 *
cy), QBColor(15), BF
Line (0.97 * cx, 0.97 * cy)-(0.98 * cx, 0.98 *
cy), QBColor(12), BF

'Program liefert fuer gegeb Zeitpkt
Filterablaufkonz
'Beispiel-System TNT auf F-400
tend = Val(txtTime(1))      'h
Beobachtungsdauer
xn = Val(Replace(txtConc(0), ",", "."))
'mmol/L Zulaufkonzentration
FL = Val(Replace(txtAdsLength(0), ",", "."))
'm      Filterlänge
vs = RemoveChar(txtFeedVol(0), "m³/s= ")
'm³/s   Volumenstrom
'beta = Text5    'm/s    Stoffübergangskoeffizient
FR = Text6   'm      Filterradius
R = Text7    'm      Adsorbenskornradius
'AH = Text8      '-      Adsorbensbettporosität
KD = Text9   'kg/m³  Adsorbensbettdichte
Dateinummer = Text10
KDK = Text11      'kg/m³  scheinbare (Hg-)
Dichte
kmax = Text12    '-      Filterschichtung
aa = Text13      'L/g    Freundlich-
bb = Text14      '-      Konstanten
cc = 1 / bb
Diffko = Text15   'm²/s   Diffusionskoeffizient
Break = 0

Konstantenberechnung:
AH = 1 - KD / KDK      '-
Adsorbensbettporosität
KM = 3.141592654 * FR ^ 2 * FL * KD   'ges.
Adsorbensmasse
LV = AH * KM / KD      'Flüssigholdup des
Filterbetts
VWZ = LV / vs     'Verweilzeit (der Lösung im
Filterbett)
Uz = vs / (AH * FR ^ 2 * 3.141592654)
'Zwischenraumströmungsgeschwindigkeit
Re = R * 2 * Uz / 0.000001
Sc = 0.000001 / Diffko
Shlam = 0.664 * Sc ^ (1 / 3) * Re ^ (1 / 2)
Shturb = 0.037 * Re ^ 0.8 * Sc / (1 + 2.44 * (Sc
^ (2 / 3) - 1) * Re ^ (-0.1))
Sh = (1 + 1.5 * (1 - AH)) * (2 + (Shlam ^ 2 +
Shturb ^ 2) ^ (1 / 2))
beta = Sh * Diffko / (2 * R)
Text5 = beta      'm/s    Stoffübergangskoeffizient
Text8 = Format(AH, "0.00000")    '-
Adsorbensbettporosität

differenzielleGrößen:
dKM = KM / kmax   'Adsorbensmasse pro
Filterebene
dLV = LV / kmax   'Volumen eines
differenziellen Zulaufelements
dt = VWZ / kmax   'Dauer eines Zeitschritts
nmax = Fix(tend * 3600 / dt)     'Zerlegung des
Zulaufs für die Beobachtungsdauer (Ganzzahl)

Zusammenfassungen:
C1 = 3 * beta * dt / (R * KDK)
If optFibrous.Value = True Then C1 = 2 * beta *
dt / (R * KDK)
C2 = dKM / dLV
cv = C1 / (1 + C1 * C2)
cf = cv * C2

Probennahme:
   Dateiname$ = "dat" +
LTrim$(Str$(Dateinummer)) + ".dat"
   Open Dateiname$ For Output As #1
   Write #1, tend, xn, FL, vs, beta, FR, AH, KD,
R, kmax, aa, bb, Diffko, KDK, Dateinummer

DbkBerechnung:
PCount = -1
ReDim InfoArray(0)
Erase InfoArray
ReDim Y(0 To kmax - 1)
Dim TempString As String
For n = 1 To nmax
   Calcdbk Y(0), X, aa, cc, cv, cf, xn, kmax
   'Dim xr
   'x = xn     'Zulaufkonzentration zur 1.
Filterebene
   '      For k = 1 To kmax   'Start innere
Programmschleife (b)
   '    xr = (y(k) / aa) ^ cc
'Kornrandkonzentration in Filterebene k
   '    y(k) = y(k) + cv * (x - xr)    'neue
Beladung der k-ten Filterebene
   '    x = x - cf * (x - xr)     'Zulaufkonzentration
zur k+1-ten Filterebene
   '    Next k    'nächste Filterebene (Sprung
nach b)
   'DoEvents
   'If Break = 1 Then GoTo Ende
   If X / xn > 0.999 Then GoTo Ende
    If Fix(n / Fix(nmax / 1000)) - n / Fix(nmax /
1000) = 0 Then
  AddInfo Chr(13) & Chr(10) & Replace(CStr(n
/ nmax), ".", ",")
  For k = 0 To kmax - 1
     AddInfo Chr(9) & Replace(CStr(Y(k)), ".",
",")
  Next k
  Write #1, n / nmax, X / xn
  yk = (1 - X / xn) * MeHeight * 0.99
  xk = n / nmax * MeWidth
  PCount = PCount + 1
  ReDim Preserve RedrawP(PCount)
  TempP(0) = xk
  TempP(1) = yk
  RedrawP(PCount) = TempP
  PSet (xk, yk)
    End If
Next n      'nächstes Fluidelement (Sprung nach
a)

Ende:
   Close #1
   Line (0.97 * cx, 0.97 * cy)-(0.98 * cx, 0.98 *
cy), QBColor(15), BF
   Line (0.98 * cx, 0.98 * cy)-(0.99 * cx, 0.99 *
cy), QBColor(10), BF
   Clipboard.Clear
   Clipboard.SetText Join(InfoArray, "")
   If ProfileCalcRunning = True Then
  MsgBox "y(FL, t) stored in clipboard", , "data
prepared for excel"
  AutoRedraw = True
   End If
   For PCount = 0 To UBound(RedrawP)
  PSet (RedrawP(PCount)(0),
RedrawP(PCount)(1))
   Next PCount
End Sub

Private Sub Befehl1_Click()
Rahmen2.Visible = Not Rahmen2.Visible
txtTime(1) = tend
txtConc(0) = xn
txtAdsLength(0) = Format(FL, "General
Number")
txtFeedVol(0) = Format(vs, "0.0000000E+00")
Text5 = Format(beta, "0.0000000E+00")
Text6 = FR
Text7 = Format(R, "General Number")
Text8 = Format(AH, "0.0000000")
Text9 = KD
Text10 = Dateinummer
Text11 = KDK
Text12 = kmax
Text13 = aa
Text14 = Format(bb, "General Number")
Text15 = Format(Diffko, "0.000E+00")
End Sub

Private Sub Befehl2_Click()
tmp = Label2.Count
Load Label2(tmp)
Label2(tmp).Caption = Label1.Caption
Label2(tmp).Visible = True
Label2(tmp).ForeColor = QBColor(Farbe)
Label2(tmp).BorderStyle = 0
End Sub

Private Sub Befehl3_Click()
Text20.Visible = True
Text21.Visible = True
s = s + 1
SendToExcel txtTime(1), 1
SendToExcel txtConc(0), 2
SendToExcel txtAdsLength(0), 3
SendToExcel txtFeedVol(0), 4
SendToExcel Text5, 5
SendToExcel Text6, 6
SendToExcel Text7, 7
SendToExcel Text8, 8
SendToExcel Text9, 9
SendToExcel Text10, 10
SendToExcel Text11, 11
SendToExcel Text12, 12
SendToExcel Text13, 13
SendToExcel Text14, 14
SendToExcel Text15, 15
End Sub

Sub SendToExcel(TObject As Object,
LineNumber)
   TObject.LinkTopic = "Excel|Tabelle1"    '
Verknüpfungsthema setzen.
   TObject.LinkMode = 2     '
Verknüpfungsmodus setzen.
   TObject.LinkItem = "Z" & LineNumber & "S"
& s
   TObject.LinkPoke
End Sub

Private Sub Befehl4_Click()
Dim exrate, Bild, kurz$, z, i, rn, rx
exrate = Text20
Text18.Visible = True
Text19.Visible = True
   Bild = DatNr
   kurz$ = "dat" + LTrim$(Str$(Bild)) + ".dat"
   z = 0
   Open kurz$ For Input As #1
   Input #1, tend, xn, FL, vs, beta, FR, AH, KD,
R, kmax, aa, bb, Diffko, KDK, Dateinummer
   For i = 1 To 1015
    Input #1, rn, rx      'Einlesen der relativen
Zeit und Ablaufkonz
   '  yk = (1 - rx) *MeHeight * .99   'Berechng d
Koordin, Abstand z unteren Bildrahmen
   '  xk = rn * MeWidth
     If i = 1 Then GoTo ErsterWert
     If Int(i / exrate) - i / exrate = 0 Then
ErsterWert:
  z = z + 1
  s = s + 1
  Text18 = rn
  Text18.LinkTopic = "Excel|Tabelle1"  '
Verknüpfungsthema setzen.
  Text18.LinkMode = 2    ' Verknüpfungsmodus
setzen.
  Text18.LinkItem = "Z" & z & "S" & s
  Text18.LinkPoke
  s = s + 1
  Text19 = rx
  Text19.LinkTopic = "Excel|Tabelle1"  '
Verknüpfungsthema setzen.
  Text19.LinkMode = 2    ' Verknüpfungsmodus
setzen.
  Text19.LinkItem = "Z" & z & "S" & s
  Text19.LinkPoke
  s = s - 2
     End If
    If EOF(1) Then Exit For
   Next i
  Close #1
Text18.Visible = False
Text19.Visible = False
Text20.Visible = False
Text21.Visible = False
s = s + 2
End Sub

Private Sub Breakb_Click()
Break = 1
End Sub

Private Sub cls_Click()
Dim i
cls
For i = 1 To Label2.UBound
   Unload Label2(i)
Next i
Label1.Caption = ""
Grid_Click
End Sub

Private Sub cmdCalcDbks_MouseMove(Button
As Integer, Shift As Integer, X As Single, Y As
Single)
Static RedBack As Boolean
RedBack = Not RedBack
If RedBack Then DatNr.BackColor =
&H80000005 Else DatNr.BackColor = &HFF&
End Sub

Private Sub cmdCls_Click()
Me.cls
End Sub

Private Sub cmdDrawMode_Click()
If Me.DrawMode = vbCopyPen Then
   Me.DrawMode = vbNotXorPen
   cmdDrawMode.Caption = "-> permanent"
   Else
   Me.DrawMode = vbCopyPen
   cmdDrawMode.Caption = "-> erasable"
End If
End Sub

Private Sub cmdOpenDbks_Click()
lb = Chr(13) & Chr(10)
profiledbkdir = StdDir & "\" & DatNr.Text &
"_Profile"
If Not FolderExist(profiledbkdir) Then MsgBox
profiledbkdir & "does not exist (-> calculate
DBKs!)": Exit Sub
ChDrive Left(profiledbkdir, 1)
ChDir profiledbkdir
ReDim MultiDbkArray(0 To DataCount + 1,
1015) '0: x-data, DataCount+1: calculation
For i = 1 To DataCount
   Open "dat" & 10 & i & ".dat" For Input As #1
  tmp = Input(LOF(1), 1)
   Close #1
   tmp = Split(tmp, lb)
' nicht gut: split wird 1015-mal ausgeführt
   For n = 1 To 1015      'first line: constants
  If Not tmp(n) = "" Then
      MultiDbkArray(i, n) = Replace(Split(tmp(n),
",")(1), ".", ",")  'format of tmp: x,y (x are all the
same)
     Else
     Exit For
  End If
   Next n
   For n = n To 1015
  MultiDbkArray(i, n) = 1
   Next n
   If i = 1 Then      'i=1: smallest W/Wo-Value --
> most x-values
  For n = 1 To 1015      'first line: constants
     If Not tmp(n) = "" Then
     MultiDbkArray(0, n) = Replace(Split(tmp(n),
",")(0), ".", ",")
     Else
     Exit For
     End If
  Next n
   End If
Next i
ChDrive Left(StdDir, 1)
ChDir StdDir
Me.Caption = "loaded"
End Sub

Private Sub cmdPic_Click()
```

```
tmp = (Me.Height - Me.ScaleHeight) / 420 'Dell-Rechner: 420*(Me.Height - Me.ScaleHeight)=1
PicClip1.Picture = GetScreenBitmap(True)
Me.ScaleMode = 3
PicClip1.ClipHeight = MeHeight / TwipsPerPixel + 1
PicClip1.ClipWidth = MeWidth / TwipsPerPixel + 1
PicClip1.ClipX = 0
PicClip1.ClipY = 22 / tmp
Clipboard.SetData PicClip1.Clip
Me.ScaleMode = 1
'SavePicture Image1(i).Picture, _
'Image1(i).Picture = PicClip1.Clip
'infile = SThumbPath & ThumbFileSEL.Text1 & "_" & i & ".bmp"
'outfile = SThumbPath & ThumbFileSEL.Text1 & "_" & i & ".jpg"
'Shell IrfanDir & " " & infile & " /convert=" & outfile
'Kill infile
'SavePicture GetScreenBitmap(True), infile
End Sub

Private Sub cmdProfile_Click()
If CurDir = StdDir Then
   profiledbkdir = StdDir & "\" & DatNr.Text & "_Profile"
   ChDrive Left(profiledbkdir, 1)
   ChDir profiledbkdir
   Else
   ChDrive Left(StdDir, 1)
   ChDir StdDir
End If
Me.Caption = CurDir
End Sub

Private Sub cmdSaveComments_Click()
Dim Dateiname$
Dateiname$ = "cm" & LTrim$(Str$(Text10)) & ".txt"
If Text10 = "0" Then
   Dateiname$ = "cm" + Trim$(Str$(DatNr)) + ".txt"
End If
If FileExist(Dateiname$) Then
   If OverWriteWarning = True Then
  If MsgBox(Dateiname$ & " exists - replace?", vbOKCancel, "confirm overwrite") = vbCancel Then Exit Sub
   End If
   Kill Dateiname$
End If
Open Dateiname$ For Output As #1
   Write #1, txtComments
Close #1
End Sub

Private Sub cmdDrawAllDbkS_Click()
'########## draw all DBKs ###
For i = 1 To DataCount
   For n = 1 To UBound(MultiDbkArray, 2)
  yk = (1 - MultiDbkArray(i, n)) * MeHeight * 0.99
  xk = MultiDbkArray(0, n) * MeWidth
  PSet (xk, yk), QBColor(Farbe)
   Next n
Next i
End Sub

Private Sub cmdCalcProfileDbk_Click()
'########## calc ###
For n = 1 To UBound(MultiDbkArray, 2)
  MultiDbkArray(DataCount + 1, n) = 0
   For i = 1 To DataCount
  'MultiDbkArray(DataCount + 1, n) = MultiDbkArray(DataCount + 1, n) + MultiDbkArray(i, n) / DataCount ' * VsArray(i)
  MultiDbkArray(DataCount + 1, n) = MultiDbkArray(DataCount + 1, n) + MultiDbkArray(i, n) * VsArray(i)
   Next i
Next n
'########## draw calculated DBK ###
For n = 1 To UBound(MultiDbkArray, 2)
   yk = (1 - MultiDbkArray(DataCount + 1, n)) * MeHeight * 0.99
   xk = MultiDbkArray(0, n) * MeWidth
   PSet (xk, yk), QBColor(Farbe)
Next n
End Sub

Private Sub cmdSmh_Click()
Dim SmhArray()
ReDim SmhArray(1 To UBound(MultiDbkArray, 2))
For i = 1 To 1000
   For n = 1 To UBound(MultiDbkArray, 2)
  SmhArray(n) = MultiDbkArray(DataCount + 1, n)
   Next n
   For n = 2 To UBound(MultiDbkArray, 2) - 1
  MultiDbkArray(DataCount + 1, n) = (SmhArray(n) + (SmhArray(n + 1) + SmhArray(n - 1)) / 2) / 2
   Next n
Next i
For n = 1 To UBound(MultiDbkArray, 2)
   yk = (1 - MultiDbkArray(DataCount + 1, n)) * MeHeight * 0.99
   xk = MultiDbkArray(0, n) * MeWidth
   PSet (xk, yk), QBColor(Farbe)
Next n
End Sub

Private Sub Command1_Click()
tmp = (Me.Height - Me.ScaleHeight) / 420 'Dell-Rechner: 420*(Me.Height - Me.ScaleHeight)=1
PicClip1.Picture = GetScreenBitmap(True)
Me.ScaleMode = 3
Dim ClipObject As Object
Set ClipObject = lbMoveProfile
PicClip1.ClipHeight = ClipObject.Height ' / TwipsPerPixel + 1
PicClip1.ClipWidth = ClipObject.Width ' / TwipsPerPixel + 1
PicClip1.ClipY = 22 / tmp + ClipObject.Top
PicClip1.ClipX = ClipObject.Left
If PicClip1.ClipWidth + PicClip1.ClipX > Screen.Width / TwipsPerPixel Then
PicClip1.ClipWidth = Screen.Width / TwipsPerPixel - PicClip1.ClipX
Clipboard.SetData PicClip1.Clip
Me.ScaleMode = 1
End Sub

Private Sub DatNr_Change()
HBildlauf1 = DatNr
Dim Dateiname$
Dateiname$ = "cm" & LTrim$(Str$(DatNr)) & ".txt"
If FileExist(Dateiname$) Then
   Open Dateiname$ For Input As #1
  Input #1, tmp
   Close #1
   txtComments.Text = tmp
   Else
   txtComments.Text = ""
End If
End Sub

Private Sub Dwidth_Click()
If Fenster.DrawWidth = 4 Then
   Fenster.DrawWidth = 1
Else
   Fenster.DrawWidth = Fenster.DrawWidth + 1End If
Dwidth.Caption = "width=" & Fenster.DrawWidth
End Sub

Private Sub End_Click()
End
End Sub

Private Sub Form_DragDrop(Source As Control, X As Single, Y As Single)
Source.Left = X
Source.Top = Y
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
Dim lb As String
lb = Chr(13) & Chr(10)
If KeyCode = vbKeyF1 Then MsgBox _
"- F3 to calculate constants (beta, ...)" & Chr(13) & Chr(10) & _
"- F4 to view code" & Chr(13) & Chr(10) & _
"- changes to MolWeight will not be saved (use comments-text)" & Chr(13) & Chr(10) & _
"" & Chr(13) & Chr(10) & _
"" & Chr(13) & Chr(10) & _
"written by  m. hoff (2002)" _
, , "calculation of adsorber-breakthrough-curves"
If KeyCode = vbKeyF3 Then CalculateConstants
If KeyCode = vbKeyReturn Then CalculateConstants
If KeyCode = vbKeyEscape Then
   If Me.WindowState = 2 Then
   Me.WindowState = 0
   Else
  End
   End If
End If
If KeyCode = vbKeyF2 Then
   On Error Resume Next
   For Each object In Fenster
  object.Caption = object.Name
   Next
End If
End Sub

Private Sub Form_Load()
StdDir = "c:\basdat"
If Not FolderExist(StdDir) Then MkDir StdDir
ChDrive Left(StdDir, 1)
ChDir StdDir
ProfileCalcRunning = True
OverWriteWarning = True
   txtTime(0).Tag = 0
   txtTime(1).Tag = 1
   txtTime(2).Tag = 0
   txtTime(1) = 1
txtConc(0).Tag = 1
txtConc(1).Tag = 0
txtConc(0) = 1
   txtFeedVol(0).Tag = 1
   txtFeedVol(1).Tag = 0
   txtFeedVol(2).Tag = 0
   txtFeedVol(0) = 1
txtAdsLength(0).Tag = 1
txtAdsLength(1).Tag = 0
txtAdsLength(0) = 1
Frame1Move = False
DatNr = 50
Farbe = 0
Break = 0
zoomx = 1
Fenster.Caption = App.EXEName
SetScale txtScale, 1
Me.WindowState = 2
Me.Height = Screen.Height * txtScale / 100
Me.Width = Screen.Width * txtScale / 100
Dwidth_Click
SetUpVsGraph
cmdAutoRedraw.Caption = "autoredraw = " & Me.AutoRedraw
If FileExist(StdDir & "\Notes.txt") Then
   Open StdDir & "\Notes.txt" For Input As #1
     tmp = Input(LOF(1), 1)
   Close #1
   txtNotes = tmp
End If
txtNotes.Width = 7500
txtNotes.Height = 10000
End Sub

Sub SetScale(ScaleFactor, ScaleOffset)
MeHeight = Screen.Height * (ScaleFactor + ScaleOffset) / 100
MeWidth = Screen.Width * (ScaleFactor + ScaleOffset) / 100
Grid_ClickEnd Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 And Shift = 0 Then
   If Abs(Line1.X1 - X) + Abs(Line1.Y1 - Y) < Abs(Line1.X2 - X) + Abs(Line1.Y2 - Y) Then
  Line1.X1 = X
  Line1.Y1 = Y
  Else
  Line1.X2 = X
  Line1.Y2 = Y
   End If
   lbLine.Left = X: lbLine.Top = Y
   lbLine.Caption = Format(GetXPos(sh1, lbLine), "0.000") & "/" & Format(GetYPos(sh1, lbLine), "0.000")
   Exit Sub
End If
End Sub

Private Sub Form_Resize()
Dim BottomDist, HSpace
BottomDist = 100
Frame1.Top = Me.ScaleHeight - Frame1.Height - BottomDist
Rahmen2.Top = Me.ScaleHeight - Rahmen2.Height - BottomDist
fr2.Top = Me.ScaleHeight - fr2.Height - BottomDist
Label1.Top = Me.ScaleHeight - Label1.Height - BottomDist
Rahmen3.Top = MeHeight - Rahmen3.Height - 50
Rahmen3.Left = MeWidth - Rahmen3.Width - 50
HSpace = 100
Frame1.Left = HSpace
Rahmen2.Left = Frame1.Left + Frame1.Width + HSpace
fr2.Left = Rahmen2.Left + Rahmen2.Width + HSpace
Label1.Left = Me.ScaleWidth - Label1.Width - 120
End Sub

Private Sub Form_Unload(Cancel As Integer)
If FileExist(StdDir & "\Notes.txt") Then Kill StdDir & "\Notes.txt"
Open StdDir & "\Notes.txt" For Output As #1
   Print #1, txtNotes
Close #1
End
End Sub

Private Sub Frame1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
Frame1.Tag = X & "#" & Y & "#" & "go"
End Sub
Private Sub Frame1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
If InStr(1, Frame1.Tag, "go") <> 0 Then
   Frame1.Left = X + Frame1.Left - Split(Frame1.Tag, "#")(0)
   Frame1.Top = Y + Frame1.Top - Split(Frame1.Tag, "#")(1)
End If
End Sub
Private Sub Frame1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
Frame1.Tag = Replace(Frame1.Tag, "go", "stop")
End Sub

Private Sub Grid_Click()
tmp = Me.DrawWidth
Me.DrawWidth = 1
Dim cx, cy
cx = MeWidth
cy = MeHeight
Line (0, 0)-(cx / 2, cy), QBColor(7), B
Line (0, 0)-(cx, cy / 2), QBColor(7), B
Line (cx / 4, 0)-(cx * 3 / 4, cy), QBColor(7), B
Line (0, 0)-(cx, cy), QBColor(0), B
Me.DrawWidth = tmp
End Sub

Private Sub HBildlauf1_Change()
DatNr = HBildlauf1
txtTime(1) = ""
txtConc(0) = ""
txtAdsLength(0) = ""
txtFeedVol(0) = ""
Text5 = ""
Text6 = ""
Text7 = ""
Text8 = ""
Text9 = ""
Text10 = ""
Text11 = ""
Text12 = ""
Text13 = ""
Text14 = ""
Text15 = ""
Me.Caption = App.EXEName
AutoRedraw = True
Dateiname$ = "dat" + LTrim$(Str$(DatNr)) + ".dat"
If FileExist(Dateiname$) Then
   Open Dateiname$ For Input As #1
  Input #1, tend, xn, FL, vs, beta, FR, AH, KD, R, kmax, aa, bb, Diffko, KDK, Dateinummer
   Close #1
   txtTime(1) = tend
   txtConc(0) = xn
   txtAdsLength(0) = Format(FL, "General Number")
   txtFeedVol(0) = Format(vs, "0.0000000E+00")
   Text5 = Format(beta, "0.0000000e+00")
   Text6 = FR
   Text7 = Format(R, "General Number")
   Text8 = Format(AH, "0.0000000")
   Text9 = KD
   Text10 = Dateinummer
   Text11 = KDK
   Text12 = kmax
   Text13 = aa
   Text14 = Format(bb, "General Number")
   Text15 = Format(Diffko, "0.000e+00")
   CalculateConstants
   Else
   Label1.Caption = ""
   Me.Caption = "[ no file]"
End If
End Sub

Private Sub HBildlauf2_Change()
Farbe = HBildlauf2
Farbe.ForeColor = QBColor(Farbe)
End Sub

Private Sub HScroll1_Change()
txtGroup = HScroll1.Value
End Sub

Private Sub HScroll2_Change()
txtScale = HScroll2.Value
SetScale txtScale, 1
End Sub

Private Sub Label1_DblClick()
Dim lb
lb = Chr(13) & Chr(10)
Clipboard.Clear
Clipboard.SetText Label1.Caption
End Sub

Private Sub Label1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If Button = 2 Then
   If Label1.Height = 7548 Then
  Label1.Height = 11000
   Else
  Label1.Height = 7548
   End If
```

```
Exit Sub
End If
Label1.Tag = X & "#" & Y & "#" & "go"
End Sub
Private Sub Label1_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
If InStr(1, Label1.Tag, "go") <> 0 Then
   Label1.Left = X + Label1.Left -
Split(Label1.Tag, "#")(0)
   Label1.Top = Y + Label1.Top -
Split(Label1.Tag, "#")(1)
End If
End Sub
Private Sub Label1_MouseUp(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
Label1.Tag = Replace(Label1.Tag, "go", "stop")
End Sub

Private Sub Label3_Click(Index As Integer)
If Index = 0 Then Farbe = 12
If Index = 1 Then Farbe = 9
If Index = 2 Then Farbe = 10
If Index = 3 Then Farbe = 13
If Index = 4 Then Farbe = 8
If Index = 5 Then Farbe = 11
If Index = 6 Then Farbe = 14
If Index = 7 Then Farbe = 0
HBildlauf2 = Farbe
Farbe.ForeColor = QBColor(Farbe)
End Sub

Private Sub lbChangeView_MouseDown(Button
As Integer, Shift As Integer, X As Single, Y As
Single)
If Shift <> 0 Or Button = 2 Then
txtNotes.FontSize = txtNotes.FontSize + 1 Else
txtNotes.FontSize = txtNotes.FontSize - 1
End Sub

Private Sub lbMoveProfile_MouseDown(Button
As Integer, Shift As Integer, X As Single, Y As
Single)
If Button = 2 Then lbMoveProfile.MousePointer
= 2
lbMoveProfile.Tag = X & "#" & Y & "#" & "go"
End Sub
Private Sub lbMoveProfile_MouseMove(Button
As Integer, Shift As Integer, X As Single, Y As
Single)
If Button = 2 And Shift = 0 Then
   xrel = X + lbMoveProfile.Left
   yrel = Y + lbMoveProfile.Top
   If Abs(Line1.X1 - xrel) + Abs(Line1.Y1 - yrel)
< Abs(Line1.X2 - xrel) + Abs(Line1.Y2 - yrel)
Then
   Line1.X1 = xrel
   Line1.Y1 = yrel
   Else
   Line1.X2 = xrel
   Line1.Y2 = yrel
    End If
   lbLine.Left = xrel: lbLine.Top = yrel
   lbLine.Caption = Format(GetXPos(sh1,
lbLine), "0.000") & "/" & Format(GetYPos(sh1,
lbLine), "0.000")
    Exit Sub
Else
   If InStr(1, lbMoveProfile.Tag, "go") <> 0 And
ckFix.Value = 0 Then
  lbMoveProfile.Left = X + lbMoveProfile.Left -
Split(lbMoveProfile.Tag, "#")(0)
  lbMoveProfile.Top = Y + lbMoveProfile.Top -
Split(lbMoveProfile.Tag, "#")(1)
   MoveWithMe lbMoveProfile, X, Y, sh1
   MoveWithMe lbMoveProfile, X, Y, lbLine
   MoveWithMe lbMoveProfile, X, Y, lbX
   MoveWithMe lbMoveProfile, X, Y, lbY2
   MoveWithMe lbMoveProfile, X, Y, lbY
   MoveWithMe lbMoveProfile, X, Y,
lbTransparent
   MoveWithMe lbMoveProfile, X, Y, ckFix
   MoveWithMe lbMoveProfile, X, Y, Shape1
   MoveArrayWithMe lbMoveProfile, X, Y, sh2
'##
   MoveArrayWithMe lbMoveProfile, X, Y, sh3
'##
   MoveArrayWithMe lbMoveProfile, X, Y,
lbXValue '##
   MoveArrayWithMe lbMoveProfile, X, Y,
lbYValue '##
   MoveArrayWithMe lbMoveProfile, X, Y, img1
'##
   MoveArrayWithMe lbMoveProfile, X, Y, img2
'##
   Line1.X1 = X + Line1.X1 -
Split(lbMoveProfile.Tag, "#")(0)
   Line1.Y1 = Y + Line1.Y1 -
Split(lbMoveProfile.Tag, "#")(1)
   Line1.X2 = X + Line1.X2 -
Split(lbMoveProfile.Tag, "#")(0)
   Line1.Y2 = Y + Line1.Y2 -
Split(lbMoveProfile.Tag, "#")(1)
    End If
End If
End Sub
Private Sub lbMoveProfile_MouseUp(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
If Button = 2 Then lbMoveProfile.MousePointer
= 0
lbMoveProfile.Tag =
Replace(lbMoveProfile.Tag, "go", "stop")
End Sub
Sub MoveWithMe(ActiveObject As Object,
ActiveX, ActiveY, PassiveObject As Object)
   PassiveObject.Left = ActiveX +
PassiveObject.Left - Split(ActiveObject.Tag,
"#")(0)
   PassiveObject.Top = ActiveY +
PassiveObject.Top - Split(ActiveObject.Tag,
"#")(1)
End Sub
Sub MoveArrayWithMe(ActiveObject As Object,
ActiveX, ActiveY, PassiveObject As Object)
For i = 0 To PassiveObject.Count - 1
   PassiveObject(i).Left = ActiveX +
PassiveObject(i).Left - Split(ActiveObject.Tag,
"#")(0)
   PassiveObject(i).Top = ActiveY +
PassiveObject(i).Top - Split(ActiveObject.Tag,
"#")(1)
Next i
End Sub

Private Sub lbProfileBack_Click()
lbProfileBack.ZOrder 1
End Sub

Private Sub lbNotes_Click()
txtNotes.Visible = Not txtNotes.Visible
lbMoveMe.Visible = txtNotes.Visible
lbChangeView.Visible = txtNotes.Visible
lbSizeMe.Visible = txtNotes.Visible
If txtNotes.Visible Then
   txtNotes.SetFocus
   txtNotes.SelStart = Len(txtNotes)
   Else
   If FileExist(StdDir & "\Notes.txt") Then Kill
StdDir & "\Notes.txt"
   Open StdDir & "\Notes.txt" For Output As #1
  Print #1, txtNotes
   Close #1
End If
End Sub

Private Sub lbSizeMe_MouseDown(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
   lbSizeMe.Tag = X & "#" & Y & "#" & "go"
End Sub
Private Sub lbSizeMe_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
On Error Resume Next
If InStr(1, lbSizeMe.Tag, "go") <> 0 Then
   lbSizeMe.Left = X + lbSizeMe.Left -
Split(lbSizeMe.Tag, "#")(0)
   lbSizeMe.Top = Y + lbSizeMe.Top -
Split(lbSizeMe.Tag, "#")(1)
   MoveWithMe lbSizeMe, X, Y, txtNotes
   MoveWithMe lbSizeMe, X, Y, lbChangeView
   MoveWithMe lbSizeMe, X, Y, lbNotes
   MoveWithMe lbSizeMe, X, Y, lbMoveMe
   txtNotes.Height = txtNotes.Height - (Y -
Split(lbSizeMe.Tag, "#")(1))
   txtNotes.Width = txtNotes.Width - (X -
Split(lbSizeMe.Tag, "#")(0))
End If
End Sub
Private Sub lbSizeMe_MouseUp(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
   lbSizeMe.Tag = Replace(lbSizeMe.Tag, "go",
"stop")
End Sub

Private Sub lbTransparent_Click()
   lbMoveProfile.BackStyle =
Abs(lbMoveProfile.BackStyle - 1)
End Sub

Private Sub Rahmen3_Click()
  Rahmen3.Left = ScaleWidth - Rahmen3.Width
End Sub

Private Sub Rahmen3_DblClick()
  Rahmen3.Left = 0
End Sub

Private Sub Recall_Click()
Dim i As Integer, rn, rx, yk, xk, Msg As String,
Bild, kurz$
Dim PrevX, PrevY
PrevX = 0 * MeWidth
PrevY = (1 - 0) * MeHeight * 0.99       'Berechng
d Koordin, Abstand z unteren Bildrahmen
On Error GoTo ErrorHandler   '
Fehlerbehandlungroutine einrichten.
ErrorHandler:     ' Zeilenmarke für
Fehlerbehandlungsroutine.
   Select Case Err
  Case 53: Msg = "ERROR 53: Diese Datei
existiert nicht."
  Case 55: Msg = "ERROR 55: Diese Datei ist
bereits geöffnet."
  Case Else: GoTo ErrorEnd
   End Select
   MsgBox Msg    ' Fehlermeldung anzeigen.
ErrorEnd:
   Resume Next   ' Prozedur fortsetzen.
   AutoRedraw = True
   Bild = DatNr
   kurz$ = "dat" + LTrim$(Str$(Bild)) + ".dat"
   Open kurz$ For Input As #1
  Input #1, tend, xn, FL, vs, beta, FR, AH, KD,
R, kmax, aa, bb, Diffko, KDK, Dateinummer
  For i = 1 To 1015
     Input #1, rn, rx      'Einlesen der relativen
Zeit und Ablaufkonz
     yk = (1 - rx) * MeHeight * 0.99
'Berechng d Koordin, Abstand z unteren
Bildrahmen
      xk = rn * zoomx * MeWidth
      PSet (xk, yk), QBColor(Farbe)
      If ckLine.Value = 1 Then
     temp = Me.DrawWidth
     Me.DrawWidth = 1
     Line (PrevX, PrevY)-(xk, yk),
QBColor(Farbe)
     PrevX = xk
     PrevY = yk
     Me.DrawWidth = temp
      End If
      If EOF(1) Then Exit For
  Next i
   Close #1
   txtTime(1) = tend
   txtConc(0) = xn
   txtAdsLength(0) = Format(FL, "General
Number")
   txtFeedVol(0) = Format(vs,
"0.0000000E+00")
   Text5 = Format(beta, "0.0000000E+00")
   Text6 = FR
   Text7 = Format(R, "General Number")
   Text8 = Format(AH, "0.0000000")
   Text9 = KD
   Text10 = Dateinummer
   Text11 = KDK
   Text12 = kmax
   Text13 = aa
   Text14 = Format(bb, "General Number")
   Text15 = Format(Diffko, "0.000E+00")
End Sub

Private Sub savePic_Click()
SavePicture Image, "db-Pict.bmp"
End Sub

Private Sub SetDwidth_Click()
Fenster.DrawWidth = Dwidth
End Sub

Private Sub shrink_Click()
tempcol = Farbe
Farbe = 15
Recall_Click
zoomx = zoomx - 0.1
If zoomx < 0.1 Then zoomx = 0.1
Farbe = tempcol
Recall_Click
Rahmen3.Visible = True
Text16 = "zoom = " & zoomx
Grid_Click
End Sub

Private Sub Text16_DblClick()
Rahmen3.Visible = Not Rahmen3.Visible
End Sub

Private Sub Text6_Change()
On Error Resume Next
Text6.ToolTipText = " R(Adsorber) = " &
Text6.Text * 1000 & " mm "
End Sub

Private Sub Text7_Change()
On Error Resume Next
Text7.ToolTipText = " R(Particle) = " &
Text7.Text * 1000 & " mm "
End Sub

Private Sub txtMolWeight_Change()
txtConc(0).Tag = 1: txtConc(1).Tag = 0
txtConc_Change (0)
End Sub

Private Sub lbMoveMe_MouseDown(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
lbMoveMe.Tag = X & "#" & Y & "#" & "go"
End Sub
Private Sub lbMoveMe_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
If InStr(1, lbMoveMe.Tag, "go") <> 0 Then
   lbMoveMe.Left = X + lbMoveMe.Left -
Split(lbMoveMe.Tag, "#")(0)
   lbMoveMe.Top = Y + lbMoveMe.Top -
Split(lbMoveMe.Tag, "#")(1)
   MoveWithMe lbMoveMe, X, Y, txtNotes
   MoveWithMe lbMoveMe, X, Y,
lbChangeView
   MoveWithMe lbMoveMe, X, Y, lbNotes
   MoveWithMe lbMoveMe, X, Y, lbSizeMe
End If
End Sub
Private Sub lbMoveMe_MouseUp(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
lbMoveMe.Tag = Replace(lbMoveMe.Tag, "go",
"stop")
End Sub

Private Sub txtScale_Change()
On Error Resume Next
HScroll2.Value = txtScale
SetScale txtScale, 1
End Sub

Private Sub zoom_Click()
temp = Farbe
Farbe = 15
Recall_Click
zoomx = zoomx + 0.1
Farbe = temp
Recall_Click
Rahmen3.Visible = True
Text16 = "zoom = " & zoomx
Grid_Click
End Sub

Private Sub txtTime_GotFocus(Index As
Integer)
   Select Case Index
  Case 0: txtTime(Index).Tag = 1: txtTime(Index
+ 1).Tag = 0: txtTime(Index + 2).Tag = 0
  Case 1: txtTime(Index).Tag = 1: txtTime(Index
- 1).Tag = 0: txtTime(Index + 1).Tag = 0
  Case 2: txtTime(Index).Tag = 1: txtTime(Index
- 1).Tag = 0: txtTime(Index - 2).Tag = 0
   End Select
End Sub

Private Sub txtTime_Change(Index As Integer)
   Select Case Index
  Case 0
     txtTime(Index) = Format(txtTime(Index),
"0.000 d   =")
     If txtTime(Index).Tag = 0 Then Exit Sub
     txtTime(Index + 1) = 24 *
Val(Replace(txtTime(Index), ",", "."))
     txtTime(Index + 2) = 24 * 60 *
Val(Replace(txtTime(Index), ",", "."))
  Case 1
     txtTime(Index) = Format(txtTime(Index),
"0.00 h   =")
     If txtTime(Index).Tag = 0 Then Exit Sub
     txtTime(Index - 1) = 1 / 24 *
Val(Replace(txtTime(Index), ",", "."))
     txtTime(Index + 1) = 60 *
Val(Replace(txtTime(Index), ",", "."))
  Case 2
     txtTime(Index) = Format(txtTime(Index),
"0.0 min")
     If txtTime(Index).Tag = 0 Then Exit Sub
     txtTime(Index - 2) = 1 / 24 / 60 *
Val(Replace(txtTime(Index), ",", "."))
     txtTime(Index - 1) = 1 / 60 *
Val(Replace(txtTime(Index), ",", "."))
   End Select
End Sub

Private Sub txtConc_GotFocus(Index As
Integer)
   Select Case Index
  Case 0: txtConc(Index).Tag = 1:
txtConc(Index + 1).Tag = 0
  Case 1: txtConc(Index).Tag = 1:
txtConc(Index - 1).Tag = 0
'  Case 2: txtConc(Index).Tag = 1:
txtConc(Index - 1).Tag = 0: txtConc(Index -
2).Tag = 0
   End Select
End Sub

Private Sub txtConc_Change(Index As Integer)
tmp = Val(Replace(txtMolWeight, ",", "."))
   Select Case Index
  Case 0
     txtConc(Index) = Format(txtConc(Index),
"0.0000 mmol/L =")
     If txtConc(Index).Tag = 0 Then Exit Sub
     txtConc(Index + 1) = tmp *
Val(Replace(txtConc(Index), ",", "."))
'      txtConc(Index + 2) = 24 * 60 *
Val(Replace(txtConc(Index), ",", "."))
  Case 1
     txtConc(Index) = Format(txtConc(Index),
"0.00 mg/L")
     If txtConc(Index).Tag = 0 Then Exit Sub
```

```
      txtConc(Index - 1) = 1 / tmp *
Val(Replace(txtConc(Index), ",", "."))
'     txtConc(Index + 1) = 60 *
Val(Replace(txtConc(Index), ",", "."))
'   Case 2
'     txtConc(Index) = Format(txtConc(Index),
"0.0 min")
'     If txtConc(Index).Tag = 0 Then Exit Sub
'     txtConc(Index - 2) = 1 / 24 / 60 *
Val(Replace(txtConc(Index), ",", "."))
'     txtConc(Index - 1) = 1 / 60 *
Val(Replace(txtConc(Index), ",", "."))
    End Select
End Sub
Private Sub txtFeedVol_GotFocus(Index As
Integer)
    Select Case Index
   Case 0: txtFeedVol(Index).Tag = 1:
txtFeedVol(Index + 1).Tag = 0:
txtFeedVol(Index + 2).Tag = 0
   Case 1: txtFeedVol(Index).Tag = 1:
txtFeedVol(Index - 1).Tag = 0:
txtFeedVol(Index1).Tag = 0   Case 2:
txtFeedVol(Index).Tag = 1: txtFeedVol(Index -
1).Tag = 0: txtFeedVol(Index - 2).Tag = 0
    End Select
End Sub

Private Sub txtFeedVol_Change(Index As
Integer)
On Error Resume Next
    Select Case Index
   Case 0
      txtFeedVol(Index) =
Format(txtFeedVol(Index), "0.000E+00 m³/s
=")
      If txtFeedVol(Index).Tag = 0 Then Exit Sub
      tmp =
CDbl(Replace(Replace(Replace(Replace(txtFeed
Vol(Index), ",", "."), "m³/s", ""), "=", ""), " ", ""))
      txtFeedVol(Index + 1) = CDbl(3600) *
CDbl(1000) * tmp
      txtFeedVol(Index + 2) = 1000 / 60 * 3600 *
1000 * tmp
   Case 1
      txtFeedVol(Index) =
Format(txtFeedVol(Index), "0.000 L/h   =")
      If txtFeedVol(Index).Tag = 0 Then Exit Sub
      txtFeedVol(Index - 1) = 1 / 3600 / 1000 *
Val(Replace(txtFeedVol(Index), ",", "."))
      txtFeedVol(Index + 1) = 1000 / 60 *
Val(Replace(txtFeedVol(Index), ",", "."))
   Case 2
      txtFeedVol(Index) =
Format(txtFeedVol(Index), "0.00 ml/min")
      If txtFeedVol(Index).Tag = 0 Then Exit Sub
      txtFeedVol(Index - 2) = 1 / 1000 * 60 /
3600 / 1000 * Val(Replace(txtFeedVol(Index),
",", "."))
      txtFeedVol(Index - 1) = 1 / 1000 * 60 *
Val(Replace(txtFeedVol(Index), ",", "."))
    End Select
End Sub

Private Sub txtAdsLength_GotFocus(Index As
Integer)
    Select Case Index
   Case 0: txtAdsLength(Index).Tag = 1:
txtAdsLength(Index + 1).Tag = 0 ':
txtAdsLength(Index + 2).Tag = 0
   Case 1: txtAdsLength(Index).Tag = 1:
txtAdsLength(Index - 1).Tag = 0 ':
txtAdsLength(Index + 1).Tag = 0
'   Case 2: txtAdsLength(Index).Tag = 1:
txtAdsLength(Index - 1).Tag = 0:
txtAdsLength(Index - 2).Tag = 0
    End Select
End Sub

Private Sub txtAdsLength_Change(Index As
Integer)
    Select Case Index
   Case 0
      txtAdsLength(Index) =
Format(txtAdsLength(Index), "0.00000 m   =")
      If txtAdsLength(Index).Tag = 0 Then Exit
Sub
      txtAdsLength(Index + 1) = 1000 *
Val(Replace(txtAdsLength(Index), ",", "."))
      'txtAdsLength(Index + 2) = 24 * 60 *
Val(Replace(txtAdsLength(Index), ",", "."))
   Case 1
      txtAdsLength(Index) =
Format(txtAdsLength(Index), "0.00 mm")
      If txtAdsLength(Index).Tag = 0 Then Exit
Sub
      txtAdsLength(Index - 1) = 1 / 1000 *
Val(Replace(txtAdsLength(Index), ",", "."))
      'txtAdsLength(Index + 1) = 60 *
Val(Replace(txtAdsLength(Index), ",", "."))
'   Case 2
'     txtAdsLength(Index) =
Format(txtAdsLength(Index), "0.0 min")
'     If txtAdsLength(Index).Tag = 0 Then Exit
Sub
'     txtAdsLength(Index - 2) = 1 / 24 / 60 *
Val(Replace(txtAdsLength(Index), ",", "."))
'     txtAdsLength(Index - 1) = 1 / 60 *
Val(Replace(txtAdsLength(Index), ",", "."))
    End Select
End Sub

Sub CalculateConstants()
On Error GoTo ErrorHandler
Dim lb
lb = Chr(13) & Chr(10)
Dim KM, LV, VWZ, Uz, Re, Sc, Shlam, Shturb,
Sh, dKM, dLV, C1, C2, dt, nmax, Dateiname$,
k, n, yk, xk
Dim cv As Double, cf As Double
Break = 0

tend = Val(txtTime(1))       'h
Beobachtungsdauer
xn = Val(Replace(txtConc(0), ",", "."))
'mmol/L Zulaufkonzentration
FL = Val(Replace(txtAdsLength(0), ",", "."))
'm      Filterlänge
vs = RemoveChar(txtFeedVol(0), "m³/s= ")
'm³/s   Volumenstrom
FR = Text6    'm      Filterradius
R = Text7     'm      Adsorbenskornradius
KD = Text9    'kg/m³  Adsorbensbettdichte
Dateinummer = Text10
KDK = Text11       'kg/m³  scheinbare (Hg-)
Dichte
kmax = Text12      '-      Filterschichtung
aa = Text13   'L/g    Freundlich-
bb = Text14   '-      Konstanten
cc = 1 / bb
Diffko = Text15    'm²/s   Diffusionskoeffizient

'Konstantenberechnung:
AH = 1 - KD / KDK       '-
Adsorbensbettporosität
KM = 3.141592654 * FR ^ 2 * FL * KD    'ges.
Adsorbensmasse
LV = AH * KM / KD       'Flüssigholdup des
Filterbetts
VWZ = LV / vs       'Verweilzeit (der Lösung im
Filterbett)
Uz = vs / (AH * FR ^ 2 * 3.141592654)
'Zwischenraumströmungsgeschwindigkeit
Re = R * 2 * Uz / 0.000001   'Reynoldszahl [-]
Sc = 0.000001 / Diffko       'Schmidtzahl [-]
Shlam = 0.664 * Sc ^ (1 / 3) * Re ^ (1 / 2)
'Sherwoodkennzahlen (laminar, turbulent, ges.) [-
]
Shturb = 0.037 * Re ^ 0.8 * Sc / (1 + 2.44 * (Sc
^ (2 / 3) - 1) * Re ^ (-0.1))
Sh = (1 + 1.5 * (1 - AH)) * (2 + (Shlam ^ 2 +
Shturb ^ 2) ^ (1 / 2))
beta = Sh * Diffko / (2 * R)      'm/s
Stoffübergangskoeffizient
Text5 = Format(beta, "0.0000000E+00")
Text8 = Format(AH, "0.00000")

'differenzielle Größen:
dKM = KM / kmax    'Adsorbensmasse pro
Filterebene
dLV = LV / kmax    'Volumen eines
differenziellen Zulaufelements
dt = VWZ / kmax    'Dauer eines Zeitschritts
nmax = Fix(tend * 3600 / dt)      'Zerlegung des
Zulaufs für die Beobachtungsdauer    '(Ganzzahl)

'Zusammenfassungen:
C1 = 3 * beta * dt / (R * KDK)
C2 = dKM / dLV
cv = C1 / (1 + C1 * C2)
cf = cv * C2

tmp = RemoveChar(txtFeedVol(0), "m³/s= ") /
(3.14159265358979 * Text6 ^ 2)
Label1.Caption = lb & _
"Dateinummer:  " & Dateinummer
Label1.Caption = Label1.Caption & lb & lb & _
"input:" & lb & _
"   tend  = " & Format(tend, "0.000e+00") & " h"
& lb & _
"     = " & txtTime(0) & lb & _
"   Xo    = " & Format(xn, "0.000e+00") & "
mmol/L" & lb & _
"     = " & txtConc(1) & lb & _
"   FL    = " & Format(FL, "0.000e+00") & " m"
& lb & _
"     = " & txtAdsLength(1) & lb & _
"   VS    = " & Format(vs, "0.000e+00") & "
m³/s" & lb & _
"     = " & txtFeedVol(1) & lb & _
"     = " & txtFeedVol(2) & lb & _
"   FR    = " & Format(FR, "0.000e+00") & " m"
& lb & _
"   R     = " & Format(R, "0.000e+00") & " m" &
lb & _
"   BD    = " & Format(KD, "0.000e+00") & "
kg/m³" & lb & _
"   SD    = " & Format(KDK, "0.000e+00") & "
kg/m³" & lb & _
"   Kmax  = " & Format(kmax, "0.000e+00") & "
[-]" & lb & _
"   a     = " & Format(aa, "0.000e+00") & " L/g"
& lb & _
"   b     = " & Format(bb, "0.000e+00") & " [-]"
& lb & _
"   Diffko= " & Format(Diffko, "0.000e+00") & "
m²/s" & lb & _
"   MolWt = " & txtMolWeight

Label1.Caption = Label1.Caption & lb & lb & _
"calculated:" & lb & _
"   Nmax  = " & Format(nmax, "0,000,000") & "
[-]" & lb & _
"   VWZ   = " & Format(VWZ, "0.000") & " s"
& lb & _
"   KM    = " & Format(KM, "0.000e+00") & "
kg" & lb & _
"   LV    = " & Format(LV, "0.000e+00") & "
m³" & lb & _
"   Uz    = " & Format(Uz, "0.000e+00") & "
m/s" & lb & _
"   w     = " & Round(tmp * 1000, 2) & " mm/s"
& lb & _
"     = " & Round(tmp * 3600, 2) & " m/h" & lb &
_
"   AH    = " & Format(VWZ, "0.000") & " [-]"
& lb & _
"   beta  = " & Format(beta, "0.0000000E+00")
& " m/s" & lb & _
"   Re    = " & Format(Re, "0.000e+00") & " [-]"
& lb & _
"   Sc    = " & Format(Sc, "0.000e+00") & " [-]"
& lb & _
"   Shlam = " & Format(Shlam, "0.000e+00") & "
[-]" & lb & _
"   Shturb= " & Format(Shturb, "0.000") & " [-]"
& lb & _
"   Sh    = " & Format(Sh, "0.0000000E+00") &
" [-]"
GoTo ErrEnd
ErrorHandler:
tmp = "error"
ErrEnd:
If tmp = "error" Then
    Me.Caption = " [no calculated data]"
    Label1.Caption = lb & "Dateinummer:  " &
Dateinummer & lb & lb & txtComments
    Else
    Me.Caption = App.EXEName
    tmp = ""
    Label1.Caption = Label1.Caption & lb & lb &
txtComments
End If
End Sub

Private Sub Label2_MouseDown(Index As
Integer, Button As Integer, Shift As Integer, X
As Single, Y As Single)
If Button = 2 Then
    If Label2(Index).Height = 4464 Then
   Label2(Index).Height = 10000
   Else
   Label2(Index).Height = 4464
    End If
Exit Sub
End If
Label2(Index).Tag = X & "#" & Y & "#" & "go"
End Sub
Private Sub Label2_MouseMove(Index As
Integer, Button As Integer, Shift As Integer, X
As Single, Y As Single)
If InStr(1, Label2(Index).Tag, "go") <> 0 Then
    Label2(Index).Left = X + Label2(Index).Left -
Split(Label2(Index).Tag, "#")(0)
    Label2(Index).Top = Y + Label2(Index).Top -
Split(Label2(Index).Tag, "#")(1)
End If
End Sub
Private Sub Label2_MouseUp(Index As Integer,
Button As Integer, Shift As Integer, X As Single,
Y As Single)
Label2(Index).Tag = Replace(Label2(Index).Tag,
"go", "stop")
End Sub

Private Sub Label2_DblClick(Index As Integer)
Dim lb
lb = Chr(13) & Chr(10)
Clipboard.Clear
Clipboard.SetText Label2(Index).Caption
End Sub

Function TwipsPerPixel()
'## PictureClip arbeitet mit Pixeln (ScaleMode =
3).
'## Problem: Me.ScaleHeight aber nicht
Me.Height liefern in ScaleMode=3 Angaben in
Pixeln
Dim tmp01, ScaleModeMemory
ScaleModeMemory = Me.ScaleMode
Me.ScaleMode = 1 '(= default)
tmp01 = Me.ScaleHeight
Me.ScaleMode = 3
TwipsPerPixel = tmp01 / Me.ScaleHeight '=
Twips per Pixels
Me.ScaleMode = 1 '(= default)
'FormPixelTop = (Me.Height - Me.ScaleHeight) /
tmp01  'Form-top ab Kopfzeile-Unterkante in
Pixels (Dell: 35)
Me.ScaleMode = ScaleModeMemory
End Function

Private Sub cmdCalcDbks_Click()
Dim TempFeedVol As String, TempDatNo
ProfileCalcRunning = False
profiledbkdir = StdDir & "\" & DatNr.Text &
"_Profile"
If Not FolderExist(profiledbkdir) Then MkDir
profiledbkdir
ChDrive Left(profiledbkdir, 1)
ChDir profiledbkdir
TempFeedVol = txtFeedVol(0)
TempDatNo = Text10
For i = 1 To DataCount
    txtFeedVol(0) = RemoveChar(TempFeedVol,
"m³/s= ") * img1(i).ToolTipText
    Text10 = "10" & i
    Me.Caption = " DBK " & i & " / " &
DataCount & "  (" & txtFeedVol(0) & " m³/s)"
    Calculate_Click
Next i
txtFeedVol(0) = TempFeedVol
Text10 = TempDatNo
ChDrive Left(StdDir, 1)
ChDir StdDir
ProfileCalcRunning = True
AutoRedraw = True
End Sub

Private Sub cmdSaveProfile_Click()
Dim SaveString
lb = Chr(13) & Chr(10)
SaveString = txtDataCount & lb & txtDataUnit
& lb & txtXLabelFreq & lb & txtYLabelDist
SaveString = SaveString & lb & Join(AreaArray,
lb)
Open VsProfilesDir & "\" &
Replace(txtProfileName, ".txt", "") & ".txt" For
Output As #1
    Print #1, SaveString
Close #1
File1.Refresh
End Sub

Private Sub File1_Click()
txtProfileName = File1
End Sub
Private Sub File1_DblClick()
lb = Chr(13) & Chr(10)
txtProfileName = File1
Open VsProfilesDir & "\" & txtProfileName For
Input As #1
    tmp = Input(LOF(1), 1)
Close
tmp = Split(tmp, lb)
txtDataCount = tmp(0)
txtDataUnit = tmp(1)
txtXLabelFreq = tmp(2)
txtYLabelDist = tmp(3)
SetUpVsGraph
For i = 4 To UBound(tmp)
    If tmp(i) <> "" Then AreaArray(i - 3) = tmp(i)
Next i
CalcArrayS
lbMoveProfile.Caption = SplitFile(File1).FName
End Sub

Sub SetUpVsGraph()
VsProfilesDir = StdDir & "\VsProfiles"
If Not FolderExist(VsProfilesDir) Then MkDir
VsProfilesDir
File1.Path = VsProfilesDir
DataCount = txtDataCount ' 20
DataUnit = txtDataUnit ' 0.13
XLabelFreq = txtXLabelFreq ' 2
YLabelDist = txtYLabelDist ' 0.1
img1(0).Picture = LoadPicture(App.Path &
"\square_sw_small.bmp")
img2(0).Picture = LoadPicture(App.Path &
"\square_grey-fill_small.bmp")
'img1(0).Picture = LoadPicture(App.Path &
"\circle_sw_small.bmp")
'img1(0).Picture = LoadPicture(App.Path &
"\caro_sw_small.bmp")
'img1(0).Picture = LoadPicture(App.Path &
"\cross_sw_small.bmp")
'img1(0).Picture = LoadPicture(App.Path &
"\cross_sw_small_bold.bmp")
ReDim AreaArray(1 To DataCount)
ReDim VsArray(1 To DataCount)
lbMoveProfile.Top = 200
lbMoveProfile.Left = MeWidth - 160
sh1.Left = lbMoveProfile.Left + 1000 +
img1(0).Width / 2
sh1.Top = lbMoveProfile.Top + 500 +
img1(0).Height / 2
sh1.Width = lbMoveProfile.Width - 1500
sh1.Height = lbMoveProfile.Height - 1200
Shape1.Left = sh1.Left
Shape1.Top = sh1.Top
Shape1.Width = sh1.Width
Shape1.Height = sh1.Height
```

```
sh1.Visible = False
lbTransparent.Left = lbMoveProfile.Left + 200
lbTransparent.Top = lbMoveProfile.Top +
lbMoveProfile.Height - 300
ckFix.Left = lbTransparent.Left + 100 +
lbTransparent.Width
ckFix.Top = lbTransparent.Top
For i = 1 To img1.Count - 1
   Unload img1(i)
Next i
For i = 1 To DataCount
   Load img1(i)
   img1(i).Visible = True
   img1(i).ToolTipText = i * DataUnit
Next i
For i = 1 To img2.Count - 1
   Unload img2(i)
Next i
For i = 1 To DataCount
   Load img2(i)
   img2(i).Visible = True
Next i
PosObject sh1, lbY, -0.12, 0.6
PosObject sh1, lbY2, -0.12, 0.4
PosObject sh1, lbX, 0.5, -0.09
SetXLabels
SetYLabels
cmdReset_Click
lbMoveProfile.ZOrder 1
End Sub

Sub SetXLabels()
For i = 1 To lbXValue.Count - 1
   Unload lbXValue(i)
   Unload sh2(i)
Next i
For i = 1 To DataCount
   If i / XLabelFreq = Int(i / XLabelFreq) Then
  Load sh2(sh2.Count)
  sh2(sh2.Count - 1).Visible = True
  PosObject sh1, sh2(sh2.Count - 1), i /
DataCount, 0
  Load lbXValue(lbXValue.Count)
  lbXValue(lbXValue.Count - 1).Visible = True
  PosObject sh1, lbXValue(lbXValue.Count - 1),
i / DataCount, -0.04
  lbXValue(lbXValue.Count - 1).Caption = i *
DataUnit
   End If
Next i
lbMoveProfile.ZOrder 1
End Sub

Sub SetYLabels()
For i = 1 To lbYValue.Count - 1
   Unload lbYValue(i)
   Unload sh3(i)
Next i
For i = 0 To 1 / YLabelDist
   Load sh3(sh3.Count)
   sh3(sh3.Count - 1).Visible = True
   PosObject sh1, sh3(sh3.Count - 1), 0, i *
YLabelDist
   Load lbYValue(lbYValue.Count)
   lbYValue(lbYValue.Count - 1).Visible = True
   PosObject sh1, lbYValue(lbYValue.Count -
1), -0.04, i * YLabelDist
   lbYValue(lbYValue.Count - 1).Caption = i *
YLabelDist
   If sh3(sh3.Count - 1).Top < Shape1.Top - 30
Then Unload lbYValue(lbYValue.Count - 1):
Unload sh3(sh3.Count - 1): Exit For
Next i
lbMoveProfile.ZOrder 1
End Sub

Private Sub cmdLine_Click()
If lbLine.Caption = "line-xy" Then
lbLine.Caption = "" Else lbLine.Caption = "line-
xy"
Line1.Visible = Not Line1.Visible
lbLine.Visible = Line1.Visible
End Sub

Private Sub cmdRefresh_Click()
SetUpVsGraph
End Sub

Private Sub cmdZoomIn_Click()
newheight = 1.2 * sh1.Height
sh1.Top = sh1.Top - newheight + sh1.Height
sh1.Height = newheight
SetYLabels
CalcArrayS
End Sub

Private Sub cmdZoomOut_Click()
newheight = sh1.Height / 1.2
sh1.Top = sh1.Top - newheight + sh1.Height
sh1.Height = newheight
SetYLabels
CalcArrayS
End Sub

Private Sub cmdCheckSum_Click()
Dim AreaSum
AreaSum = 0
For i = 1 To DataCount
   AreaSum = AreaSum + AreaArray(i)
Next i
Dim VsSum
VsSum = 0
For i = 1 To DataCount
   VsSum = VsSum + VsArray(i)
Next i
Me.Caption = "AreaSum=" & Format(AreaSum,
"0.000") & "   VsSum=" & Format(VsSum,
"0.000")
End Sub

Private Sub cmdReset_Click()
For i = 1 To DataCount
   AreaArray(i) = 0
   If img1(i).ToolTipText = 1 Then AreaArray(i)
= 0
Next i
CalcArrayS
End Sub

Sub GraphToAreaArray()
For i = 1 To DataCount
   AreaArray(i) = GetYPos(sh1, img1(i))
Next i
CalcArrayS
End Sub

Sub CalcArrayS()
CalcAreaArray
CalcVs
DisplayArrayS
End Sub

Sub CalcVs()
For i = 1 To DataCount
    VsArray(i) = AreaArray(i) *
img1(i).ToolTipText
Next i
End Sub

Private Sub cmdIterate_Click()
IterateVsAndArea
DisplayArrayS
End Sub

Sub IterateVsAndArea()
Do
   debugcount = debugcount + 1
   If debugcount = 5000 Then MsgBox "couldn't
iterate": Exit Sub
   '-------check and correct 'sum(vs)=1' ----------
---
   Dim VsSum, VsDelta, PointZero
   VsSum = 0
   For i = 1 To DataCount
  VsSum = VsSum + VsArray(i)
   Next i
   VsDelta = VsSum - 1
   If Abs(VsDelta) <= 0.0001 Then Exit Do
   For i = 1 To DataCount
  If img1(i).ToolTipText = 1 Then PointZero = i:
Exit For
  If img1(i).ToolTipText > 1 Then PointZero = i
'falls x-skalierung so, dass kein W/Wo=1 existiert
   Next i
   If VsDelta < 0 Then
  For i = 1 To PointZero - 1
     AreaArray(i) = AreaArray(i) * 0.99
  Next i
  For i = PointZero + 1 To DataCount
     AreaArray(i) = AreaArray(i) / 0.99
  Next i
  Else
  For i = 1 To PointZero - 1
     AreaArray(i) = AreaArray(i) / 0.99
  Next i
  For i = PointZero + 1 To DataCount
     AreaArray(i) = AreaArray(i) * 0.99
  Next i
   End If
   CalcAreaArray
   CalcVs
Loop
cmdCheckSum_Click
End Sub

Sub CalcAreaArray()
Dim AreaSum, AreaDelta, DataCount2,
dAreaDelta, ZeroArray()
AreaSum = 0
For i = 1 To DataCount
   AreaSum = AreaSum + AreaArray(i)
Next i
AreaDelta = AreaSum - 1
DataCount2 = DataCount
ReDim ZeroArray(1 To DataCount)
Do
   dAreaDelta = AreaDelta / DataCount2
   For i = 1 To DataCount   If ZeroArray(i) <> 1
Then
      AreaArray(i) = AreaArray(i) - dAreaDelta
      AreaDelta = AreaDelta - dAreaDelta
      If AreaArray(i) < 0 Then
     DataCount2 = DataCount2 - 1
     AreaDelta = AreaDelta - AreaArray(i)
     AreaArray(i) = 0
      End If
      If AreaArray(i) = 0 Then ZeroArray(i) = 1
   End If
   Next i
   If AreaDelta <= 1E-20 Then Exit Do
Loop
End Sub

Sub DisplayArrayS()
For i = 1 To DataCount
   PosObject sh1, img2(i), i / DataCount,
VsArray(i)
Next i
For i = 1 To DataCount
   PosObject sh1, img1(i), i / DataCount,
AreaArray(i)
Next i
End Sub

Function GetXPos(XYSystem As Object,
PointObject As Object)
GetXPos = (PointObject.Left +
PointObject.Width / 2 - XYSystem.Left) /
XYSystem.Width
End Function
Function GetYPos(XYSystem As Object,
PointObject As Object)
GetYPos = 1 - (PointObject.Top +
PointObject.Height / 2 - XYSystem.Top) /
XYSystem.Height
End Function
Sub PosObject(XYSystem As Object,
PointObject As Object, RelativeX, RelativeY)
YOffSet = 0
PointObject.Left = XYSystem.Left + RelativeX *
XYSystem.Width - PointObject.Width / 2
PointObject.Top = XYSystem.Top + (1 -
RelativeY) * (XYSystem.Height - YOffSet) -
PointObject.Height / 2
End Sub

Private Sub Rahmen2_MouseDown(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
Rahmen2.Tag = X & "#" & Y & "#" & "go"
End Sub
Private Sub Rahmen2_MouseMove(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
If InStr(1, Rahmen2.Tag, "go") <> 0 Then
   Rahmen2.Left = X + Rahmen2.Left -
Split(Rahmen2.Tag, "#")(0)
   Rahmen2.Top = Y + Rahmen2.Top -
Split(Rahmen2.Tag, "#")(1)
End If
End Sub
Private Sub Rahmen2_MouseUp(Button As
Integer, Shift As Integer, X As Single, Y As
Single)
Rahmen2.Tag = Replace(Rahmen2.Tag, "go",
"stop")
End Sub

Private Sub fr2_MouseDown(Button As Integer,
Shift As Integer, X As Single, Y As Single)
fr2.Tag = X & "#" & Y & "#" & "go"
End Sub
Private Sub fr2_MouseMove(Button As Integer,
Shift As Integer, X As Single, Y As Single)
If InStr(1, fr2.Tag, "go") <> 0 Then
   fr2.Left = X + fr2.Left - Split(fr2.Tag, "#")(0)
   fr2.Top = Y + fr2.Top - Split(fr2.Tag, "#")(1)
End If
End Sub
Private Sub fr2_MouseUp(Button As Integer,
Shift As Integer, X As Single, Y As Single)
fr2.Tag = Replace(fr2.Tag, "go", "stop")
End Sub

Private Sub img1_MouseDown(Index As
Integer, Button As Integer, Shift As Integer, X
As Single, Y As Single)
img1(Index).Tag = X & "#" & Y & "#" & "go"
End Sub
Private Sub img1_MouseMove(Index As Integer,
Button As Integer, Shift As Integer, X As Single,
Y As Single)
If InStr(1, img1(Index).Tag, "go") <> 0 Then
On Error Resume Next
   DeltaY = Y - Split(img1(Index).Tag, "#")(1)
   img1(Index).Top = img1(Index).Top + DeltaY
   img1(Index - 1).Top = img1(Index - 1).Top +
txtGroup * DeltaY / 20
   img1(Index + 1).Top = img1(Index + 1).Top +
txtGroup * DeltaY / 20
   img1(Index - 2).Top = img1(Index - 2).Top +
txtGroup * DeltaY / 40
   img1(Index + 2).Top = img1(Index + 2).Top +
txtGroup * DeltaY / 40
   img1(Index - 3).Top = img1(Index - 3).Top +
txtGroup * DeltaY / 80
   img1(Index + 3).Top = img1(Index + 3).Top +
txtGroup * DeltaY / 80
End If
   Me.Caption = "X/Y: " &
Round(GetXPos(sh1, img1(Index)), 3) & " / " &
Round(GetYPos(sh1, img1(Index)), 3)
End Sub
Private Sub img1_MouseUp(Index As Integer,
Button As Integer, Shift As Integer, X As Single,
Y As Single)
img1(Index).Tag = Replace(img1(Index).Tag,
"go", "stop")
GraphToAreaArrayEnd Sub
```

‘###### Module-Code

```
Attribute VB_Name = "Module1"
Private Declare Sub keybd_event Lib "user32"
(ByVal bVk As Byte, _
    ByVal bScan As Byte, ByVal dwFlags As
Long, ByVal dwExtraInfo As Long)
Private Const KEYEVENTF_KEYUP = &H2

Public Declare Function SetWindowPos& Lib
"user32" (ByVal hWnd&, ByVal
WndInsertAfter&, ByVal X&, ByVal Y&, ByVal
cx&, ByVal cy&, ByVal wFlags&)
Public Const SWP_SHOWWINDOW = &H40
Public Const HWND_TOPMOST = -1
Public Const HWND_NOTTOPMOST = -2

Declare Sub Calcdbk Lib "dbk32.dll" (Y As
Double, X As Double, ByVal aa As Double,
ByVal cc As Double, _
    ByVal cv As Double, ByVal cf As Double,
ByVal xn As Double, ByVal kmax As Long)

Public InfoArray() As String

Public Type TFile
    Path As String
    PathAndFName As String
    FName As String
    FNameAndExtension As String
    Extension As String
    DotExtension As String
    Dir1 As String
    Dir2 As String
    Dir3 As String
    Dir4 As String
    Dir5 As String
End Type

Function GetScreenBitmap(Optional
ActiveWindow As Boolean) As Picture
    ' save the current picture in the clipboard, if
any
    Dim pic As StdPicture
    Set pic = Clipboard.GetData(vbCFBitmap)
    If ActiveWindow Then
   ' Press the Alt key
   keybd_event vbKeyMenu, 0, 0, 0
    End If
    ' Press the Print Screen key
    keybd_event vbKeySnapshot, 0, 0, 0
    DoEvents
    ' Release the Print Screen key
    keybd_event vbKeySnapshot, 0,
KEYEVENTF_KEYUP, 0
    If ActiveWindow Then
   ' Release the Alt key
   keybd_event vbKeyMenu, 0,
KEYEVENTF_KEYUP, 0
    End If
    DoEvents
    ' return the bitmap now in the clipboard
    Set GetScreenBitmap =
Clipboard.GetData(vbCFBitmap)
    ' restore the original contents of the clipboard
    Clipboard.SetData pic, vbCFBitmap
End Function

Function FileExist(Dateiname) As Boolean
If Dateiname = "" Then
    FileExist = False
    Exit Function
End If
On Error GoTo Fehler:
FileExist = Dir(Dateiname) <> ""
Exit Function
Fehler:
    FileExist = False
    Resume Next
End Function

Function FolderExist(Dateiname) As Boolean
If Dateiname = "" Then
    FolderExist = False
    Exit Function
End If
On Error GoTo Fehler:
FolderExist = Dir(Dateiname, 16) <> ""
Exit Function
Fehler:
    FolderExist = False
    Resume Next
End Function

Sub AddInfo(InfoText As String)
'check whether it is the first ReDim for a [then
a(0)]
IsAlreadyReDim = False
On Error GoTo FirstReDim
dummy = LBound(InfoArray)
IsAlreadyReDim = True
FirstReDim:
If IsAlreadyReDim Then
    ReDim Preserve
InfoArray(UBound(InfoArray) + 1)
    Else
    ReDim InfoArray(0)
End If
'Add
InfoArray(UBound(InfoArray)) = InfoText
End Sub

Function RemoveChar(RawString As String,
CharString As String, Optional ReplaceChar As
String = "") As String
Dim i As Integer
RemoveChar = RawString
For i = 1 To Len(CharString)
    RemoveChar = Replace(RemoveChar,
Mid(CharString, i, 1), "")
Next i
End Function

Function NextFile(Dateiname) As String
If FileExist(Dateiname) Then
    'check if file is a numerated one:
    If (InStr(SplitFile(Dateiname).FName, " #")
<> 0 And _
   IsNumeric(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, " #") +
2)) _
    And InStr(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, " #") +
2), ".") = 0 _
    And InStr(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, " #") +
2), ",") = 0 _
    And InStr(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, " #") +
2), " ") = 0 _
    ) Then 'it is. so try the next number:
    NextFile = SplitFile(Dateiname).Path & "\" & _
    Mid(SplitFile(Dateiname).FName, 1,
InStrRev(SplitFile(Dateiname).FName, " #") - 1)
& " #" & _
    (CInt(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, " #") +
2)) + 1) _
    & SplitFile(Dateiname).DotExtension
     Else 'it is not. so try the first number ( #2):
    NextFile = SplitFile(Dateiname).Path & "\" & _
    SplitFile(Dateiname).FName & " #2" &
SplitFile(Dateiname).DotExtension
     End If
     Do
   If FileExist(NextFile) Then 'count up till
number is free:
       NextFile = SplitFile(NextFile).Path & "\" &
_
       Mid(SplitFile(NextFile).FName, 1,
InStrRev(SplitFile(NextFile).FName, " #") - 1) &
" #" & _
       (CInt(Mid(SplitFile(NextFile).FName,
InStrRev(SplitFile(NextFile).FName, " #") + 2))
+ 1) _
      & SplitFile(NextFile).DotExtension
   Else
      Exit Do
   End If
    Loop
Else
    NextFile = Dateiname
End If
End Function

Function NextFile2(Dateiname) As String
If FileExist(Dateiname) Then
    'check if file is a numerated one:
    If (InStr(SplitFile(Dateiname).FName, "_v")
<> 0 And _
   IsNumeric(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, "_v") +
2)) _
    And InStr(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, "_v") +
2), ".") = 0 _
    And InStr(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, "_v") +
2), ",") = 0 _
    And InStr(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, "_v") +
2), " ") = 0 _
    ) Then 'it is. so try the next number:
    NextFile2 = SplitFile(Dateiname).Path & "\" &
_
    Mid(SplitFile(Dateiname).FName, 1,
InStrRev(SplitFile(Dateiname).FName, "_v") - 1)
& "_v" & _
    (CInt(Mid(SplitFile(Dateiname).FName,
InStrRev(SplitFile(Dateiname).FName, "_v") +
2)) + 1) _
    & SplitFile(Dateiname).DotExtension
     Else 'it is not. so try the first number (_v2):
    NextFile2 = SplitFile(Dateiname).Path & "\" &
_
    SplitFile(Dateiname).FName & "_v2" &
SplitFile(Dateiname).DotExtension
     End If
     Do
    If FileExist(NextFile2) Then 'count up till
number is free:
       NextFile2 = SplitFile(NextFile2).Path & "\"
& _
       Mid(SplitFile(NextFile2).FName, 1,
InStrRev(SplitFile(NextFile2).FName, "_v") - 1)
& "_v" & _
       (CInt(Mid(SplitFile(NextFile2).FName,
InStrRev(SplitFile(NextFile2).FName, "_v") +
2)) + 1) _
      & SplitFile(NextFile2).DotExtension
    Else
       Exit Do
    End If
     Loop
Else
    NextFile2 = Dateiname
End If
End Function


Function SplitFile(FullPath) As TFile
'Print "Input = " & "c:\Temp\abc.xyz"
'Print "Path = " &
SplitFile("c:\Temp\abc.xyz").Path    'c:\Temp
'Print "PathAndFName = " &
SplitFile("c:\Temp\abc.xyz").PathAndFName
'c:\Temp\abc
'Print "FName = " &
SplitFile("c:\Temp\abc.xyz").FName       'abc
'Print "FNameAndExtension = " &
SplitFile("c:\Temp\abc.xyz").FNameAndExtensio
n   'abc.xyz
'Print "Extension = " &
SplitFile("c:\Temp\abc.xyz").Extension    'xyz
    Dim FileName$, PathName$, FileExt$
    If InStr(FullPath, "\") Then
   PathName = Mid(FullPath, 1,
InStrRev(FullPath, "\") - 1)
    FileName = Mid(FullPath, InStrRev(FullPath,
"\") + 1)
    '#####new#####
    If InStr(PathName, "\") Then
       Remain1 = Mid(PathName, 1,
InStrRev(PathName, "\") - 1)
       Directory1 = Mid(PathName,
InStrRev(PathName, "\") + 1)
       If InStr(Remain1, "\") Then
       Remain2 = Mid(Remain1, 1,
InStrRev(Remain1, "\") - 1)
       Directory2 = Mid(Remain1,
InStrRev(Remain1, "\") + 1)
       If InStr(Remain2, "\") Then
      Remain3 = Mid(Remain2, 1,
InStrRev(Remain2, "\") - 1)
      Directory3 = Mid(Remain2,
InStrRev(Remain2, "\") + 1)
       If InStr(Remain3, "\") Then
      Remain4 = Mid(Remain3, 1,
InStrRev(Remain3, "\") - 1)
      Directory4 = Mid(Remain3,
InStrRev(Remain3, "\") + 1)
      If InStr(Remain4, "\") Then
     Remain5 = Mid(Remain4, 1,
InStrRev(Remain4, "\") - 1)
     Directory5 = Mid(Remain4,
InStrRev(Remain4, "\") + 1)
       End If
       End If
        End If
         End If
    End If
    '#####new#####
     Else
    PathName = ""
    FileName = FullPath
     End If
     If InStr(FileName, ".") Then
    FileExt = Mid(FileName, InStrRev(FileName,
".") + 1)
     FileName = Mid(FileName, 1,
InStrRev(FileName, ".") - 1)
     Else
    FileExt = ""
     End If
     SplitFile.Path = PathName
     SplitFile.PathAndFName = PathName & "\" &
FileName
     SplitFile.FName = FileName
     SplitFile.FNameAndExtension = FileName &
"." & FileExt
     SplitFile.Extension = FileExt
     SplitFile.DotExtension = "." & FileExt
     SplitFile.Dir1 = Directory1
     SplitFile.Dir2 = Directory2
     SplitFile.Dir3 = Directory3
     SplitFile.Dir4 = Directory4
     SplitFile.Dir5 = Directory5
End Function
```

‘### Objects ###

```
VERSION 5.00
Object = "{27395F88-0C0C-101B-A3C9-08002B2F49FB}#1.1#0"; "PICCLP32.OCX"
Begin VB.Form Fenster
   Appearance      = 0 'Flat
   BackColor       = &H00FFFFFF&
   Caption         = "TNT-Fr-6"
   ClientHeight    = 11304
   ClientLeft      = 2628
   ClientTop       = 348
   ClientWidth     = 13344
   ForeColor       = &H00000000&
   KeyPreview      = -1 'True
   LinkTopic       = "Form2"
   MousePointer    = 1 'Arrow
   PaletteMode     = 1 'UseZOrder
   ScaleHeight     = 11304
   ScaleWidth      = 13344
   WindowState     = 2 'Maximized
   Begin VB.TextBox txtNotes
      Appearance      = 0 'Flat
      BackColor       = &H00E0E0E0&
      BeginProperty Font
         Name            = "Arial"
         Size            = 10.2
         Charset         = 0
         Weight          = 400
         Underline       = 0 'False
         Italic          = 0 'False
         Strikethrough   = 0 'False
      EndProperty
      ForeColor       = &H00000000&
      Height          = 2796
      Left            = 444
      MultiLine       = -1 'True
      ScrollBars      = 2 'Vertical
      TabIndex        = 119
      Top             = 444
      Visible         = 0 'False
      Width           = 2736
   End
   Begin VB.CheckBox ckFix
      Appearance      = 0 'Flat
      BackColor       = &H00FFFFFF&
      Caption         = "fix"
      ForeColor       = &H00000000&
      Height          = 216
      Left            = 8988
      TabIndex        = 109
      Top             = 276
      Value           = 1 'Checked
      Width           = 420
   End
   Begin VB.Frame fr2
      Appearance      = 0 'Flat
      BackColor       = &H00FFFFFF&
      Caption         = "VS-profiles"
      BeginProperty Font
         Name            = "Courier New"
         Size            = 9
         Charset         = 0
         Weight          = 700
         Underline       = 0 'False
         Italic          = 0 'False
         Strikethrough   = 0 'False
      EndProperty
      ForeColor       = &H80000008&
      Height          = 3084
      Left            = 6456
      MousePointer    = 15 'Size All
      TabIndex        = 79
      Top             = 4872
      Width           = 3792
      Begin VB.CommandButton cmdSmh
         Caption         = "smh"
         Height          = 276
         Left            = 3300
         MousePointer    = 1 'Arrow
         TabIndex        = 112
         Top             = 1716
         Width           = 408
      End
      Begin VB.CommandButton cmdCalcProfileDbk
         Caption         = "calculate + draw profiled DBK"
         Height          = 276
         Left            = 144
         MousePointer    = 1 'Arrow
         TabIndex        = 97
         Top             = 1716
         Width           = 3168
      End
      Begin VB.CommandButton cmdCls
         Caption         = "cls"
         Height          = 276
         Left            = 3300
         MousePointer    = 1 'Arrow
         TabIndex        = 100
         Top             = 1452
         Width           = 408
      End
      Begin VB.CommandButton cmdAutoRedraw
         Caption         = "autoredraw=false"
         Height          = 276
         Left            = 1680
         MousePointer    = 1 'Arrow
         TabIndex        = 99
         Top             = 1452
         Width           = 1632
      End
      Begin VB.CommandButton cmdDrawAllDbkS
         Caption         = "draw all DBKs"
         Height          = 276
         Left            = 144
         MousePointer    = 1 'Arrow
         TabIndex        = 96
         Top             = 1452
         Width           = 1548
      End
      Begin VB.CommandButton cmdOpenDbks
         Caption         = "open DBKs (load/create profile+dbk first)"
         Height          = 288
         Left            = 144
         MousePointer    = 1 'Arrow
         TabIndex        = 80
         Top             = 1176
         Width           = 3564
      End
      Begin VB.CommandButton cmdCalcDbks
         Caption         = "calc DBKs (load/create profile+dbk first)"
         Height          = 264
         Left            = 144
         MousePointer    = 1 'Arrow
         TabIndex        = 81
         Top             = 924
         Width           = 3564
      End
      Begin VB.CommandButton cmdSaveProfile
         Caption         = "Save as"
         Height          = 264
         Left            = 144
         MousePointer    = 1 'Arrow
         TabIndex        = 84
         Top             = 684
         Width           = 1296
      End
      Begin VB.CommandButton cmdIterate
         Caption         = "iterate to sum(vs)=0"
         Height          = 492
         Left            = 2832
         MousePointer    = 1 'Arrow
         TabIndex        = 95
         Top             = 204
         Width           = 876
      End
      Begin VB.CommandButton cmdZoomOut
         Caption         = "out"
         Height          = 252
         Left            = 2508
         MousePointer    = 1 'Arrow
         TabIndex        = 92
         Top             = 444
         Width           = 336
      End
      Begin VB.CommandButton cmdZoomIn
         Caption         = "zoom in"
         Height          = 252
         Left            = 1812
         MousePointer    = 1 'Arrow
         TabIndex        = 91
         Top             = 444
         Width           = 708
      End
      Begin VB.CommandButton cmdLine
         Caption         = "line"
         Height          = 252
         Left            = 1428
         MousePointer    = 1 'Arrow
         TabIndex        = 90
         Top             = 444
         Width           = 396
      End
      Begin VB.CommandButton cmdReset
         Caption         = "reset"
         Height          = 252
         Left            = 876
         MousePointer    = 1 'Arrow
         TabIndex        = 89
         Top             = 444
         Width           = 564
      End
      Begin VB.CommandButton cmdCheckSum
         Caption         = "checksum"
         Height          = 252
         Left            = 1812
         MousePointer    = 1 'Arrow
         TabIndex        = 94
         Top             = 204
         Width           = 1032
      End
      Begin VB.CommandButton cmdRefresh
         Caption         = "refresh (!)"
         Height          = 252
         Left            = 876
         MousePointer    = 1 'Arrow
         TabIndex        = 93
         ToolTipText     = " data will be lost ! "
         Top             = 204
         Width           = 948
      End
      Begin VB.TextBox txtDataCount
         Appearance      = 0 'Flat
         Height          = 252
         Left            = 144
         MousePointer    = 3 'I-Beam
         TabIndex        = 88
         Text            = "20"
         ToolTipText     = "DataCount"
         Top             = 216
         Width           = 348
      End
      Begin VB.TextBox txtDataUnit
         Appearance      = 0 'Flat
         Height          = 252
         Left            = 480
         MousePointer    = 3 'I-Beam
         TabIndex        = 87
         Text            = "0,1"
         ToolTipText     = "DataUnit"
         Top             = 216
         Width           = 396
      End
      Begin VB.TextBox txtXLabelFreq
         Appearance      = 0 'Flat
         Height          = 240
         Left            = 144
         MousePointer    = 3 'I-Beam
         TabIndex        = 86
         Text            = "5"
         ToolTipText     = "XLabelFreq"
         Top             = 456
         Width           = 348
      End
      Begin VB.TextBox txtYLabelDist
         Appearance      = 0 'Flat
         Height          = 240
         Left            = 468
         MousePointer    = 3 'I-Beam
         TabIndex        = 85
         Text            = "0,1"
         ToolTipText     = "YLabelDist"
         Top             = 456
         Width           = 408
      End
      Begin VB.TextBox txtProfileName
         Height          = 252
         Left            = 1416
         MousePointer    = 3 'I-Beam
         TabIndex        = 83
         Text            = "NewSetting"
         Top             = 672
         Width           = 2280
      End
      Begin VB.FileListBox File1
         BeginProperty Font
            Name            = "Small Fonts"
            Size            = 6
            Charset         = 0
            Weight          = 400
            Underline       = 0 'False
            Italic          = 0 'False
            Strikethrough   = 0 'False
         EndProperty
         Height          = 1080
         Left            = 144
         MousePointer    = 1 'Arrow
         TabIndex        = 82
         Top             = 1944
         Width           = 3564
      End
      Begin PicClip.PictureClip PicClip1
         Left            = 456
         Top             = 5940
         _ExtentX        = 2688
         _ExtentY        = 3069
         _Version        = 393216
      End
      Begin VB.Frame Rahmen3
         Appearance      = 0 'Flat
         BackColor       = &H00FFFFFF&
         BeginProperty Font
            Name            = "MS Sans Serif"
            Size            = 7.8
            Charset         = 0
            Weight          = 700
            Underline       = 0 'False
            Italic          = 0 'False
            Strikethrough   = 0 'False
         EndProperty
         Height          = 216
         Left            = 6828
         Max             = 98
         Min             = 8
         MousePointer    = 1 'Arrow
         TabIndex        = 116
         Top             = 2796
         Value           = 70
         Width           = 732
         Begin VB.TextBox txtGroup
            Alignment       = 2 'Center
            Height          = 252
            Left            = 7812
            MousePointer    = 1 'Arrow
            TabIndex        = 113
            Text            = "5"
            ToolTipText     = " group VS-profile points (0: no grouping) "
            Top             = 2784
            Width           = 276
         End
         Begin VB.CheckBox ckLine
            BackColor       = &H00FFFFFF&
            Caption         = "draw lines"
            BeginProperty Font
               Name            = "Courier New"
               Size            = 9
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            Height          = 204
            Left            = 5376
            MousePointer    = 1 'Arrow
            TabIndex        = 111
            Top             = 2796
            Value           = 1 'Checked
            Width           = 1632
         End
         Begin VB.TextBox Text5
            Alignment       = 1 'Right Justify
            Appearance      = 0 'Flat
            BackColor       = &H00E0E0E0&
            BeginProperty Font
               Name            = "Small Fonts"
               Size            = 6.6
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            ForeColor       = &H00FF8080&
            Height          = 250
            Left            = 3636
            Locked          = -1 'True
            MousePointer    = 3 'I-Beam
            TabIndex        = 12
            Text            = "2,2222222E+22"
            Top             = 2628
            Visible         = 0 'False
            Width           = 1230
         End
         Begin VB.TextBox Text8
            Alignment       = 1 'Right Justify
            Appearance      = 0 'Flat
            BackColor       = &H00E0E0E0&
            BeginProperty Font
               Name            = "Small Fonts"
               Size            = 6.6
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            ForeColor       = &H00FF8080&
            Height          = 250
            Left            = 4560
            Locked          = -1 'True
            MousePointer    = 3 'I-Beam
            TabIndex        = 15
            Top             = 2736
            Visible         = 0 'False
            Width           = 975
         End
         Begin VB.CommandButton cmdProfile
            Caption         = "profile-dir"
            BeginProperty Font
               Name            = "MS Sans Serif"
               Size            = 7.8
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            Height          = 285
            Left            = 2004
            MousePointer    = 1 'Arrow
            TabIndex        = 47
            Text            = "export-rate:"
            Top             = 2424
            Visible         = 0 'False
            Width           = 1368
         End
         Begin VB.TextBox Text20
            Appearance      = 0 'Flat
            BeginProperty Font
               Name            = "MS Sans Serif"
               Size            = 7.8
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            Height          = 285
            Left            = 3360
            MousePointer    = 3 'I-Beam
            TabIndex        = 46
            Text            = "5"
            Top             = 2424
            Visible         = 0 'False
            Width           = 375
         End
         Begin VB.TextBox Text19
            Appearance      = 0 'Flat
            BeginProperty Font
               Name            = "MS Sans Serif"
               Size            = 7.8
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            ForeColor       = &H0000FF00&
            Height          = 300
            Left            = 2004
            MousePointer    = 3 'I-Beam
            TabIndex        = 45
            Text            = "Text19"
            Top             = 1836
            Visible         = 0 'False
            Width           = 1728
         End
         Begin VB.TextBox Text18
            Appearance      = 0 'Flat
            BeginProperty Font
               Name            = "MS Sans Serif"
               Size            = 7.8
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            ForeColor       = &H0000FF00&
            Height          = 285
            Left            = 2004
            MousePointer    = 3 'I-Beam
            TabIndex        = 44
            Text            = "Text18"
            Top             = 2124
            Visible         = 0 'False
            Width           = 1728
         End
         Begin VB.TextBox txtMolWeight
            Appearance      = 0 'Flat
            BackColor       = &H00FFFFFF&
            BeginProperty Font
               Name            = "Small Fonts"
               Size            = 6.6
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            ForeColor       = &H00000000&
            Height          = 230
            Left            = 708
            MousePointer    = 3 'I-Beam
            TabIndex        = 65
            Text            = "227,0 g/mol"
            Top             = 756
            Width           = 972
         End
         Begin VB.TextBox Text21
            Appearance      = 0 'Flat
            BeginProperty Font
               Name            = "MS Sans Serif"
               Size            = 7.8
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            Height          = 285
            Left            = 2004
            MousePointer    = 1 'Arrow
            TabIndex        = 47
            Text            = "export-rate:"
            Top             = 2424
            Visible         = 0 'False
            Width           = 1368
         End
         Begin VB.CommandButton cmdSaveComments
            Caption         = "Save Comments:"
            BeginProperty Font
               Name            = "MS Sans Serif"
               Size            = 7.8
               Charset         = 0
               Weight          = 700
               Underline       = 0 'False
               Italic          = 0 'False
               Strikethrough   = 0 'False
            EndProperty
            Height          = 200
            Left            = 1728
            MousePointer    = 1 'Arrow
            TabIndex        = 64
            Top             = 2256
            Width           = 2376
         End
         Begin VB.TextBox txtTime
            Appearance      = 0 'Flat
            BackColor       = &H00FFFFFF&
            Height          = 252
            Index           = 0
            Left            = 5376
            MousePointer    = 3 'I-Beam
            TabIndex        = 59
            Text            = "0"
            Top             = 732
            Width           = 900
         End
         Begin VB.TextBox txtTime
            Appearance      = 0 'Flat
            BackColor       = &H00FFFFFF&
            Height          = 252
            Index           = 1
            Left            = 6264
            MousePointer    = 3 'I-Beam
            TabIndex        = 58
            Text            = "0"
            Top             = 732
            Width           = 1000
         End
         Begin VB.TextBox txtTime
            Appearance      = 0 'Flat
```

```
   BackColor = &H00FFFFFF&
   Height = 252
   Index = 2
   Left = 7248
   MousePointer = 3 'I-Beam
   TabIndex = 57
   Text = "0"
   Top = 732
   Width = 1068
End
Begin VB.TextBox
txtTimeLabel
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   BorderStyle = 0 'None
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 204
   Left = 5376
   Locked = -1 'True
   MousePointer = 1 'Arrow
   TabIndex = 56
   Text = "AdsorptionTime t : ="
   Top = 528
   Width = 2376
End
Begin VB.TextBox
txtConcLabel
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   BorderStyle = 0 'None
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 204
   Left = 5376
   Locked = -1 'True
   MousePointer = 1 'Arrow
   TabIndex = 55
   Text = "FeedConcentration
Xo: ="
   Top = 1104
   Width = 2376
End
Begin VB.TextBox txtConc
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   Height = 252
   Index = 0
   Left = 5388
   MousePointer = 3 'I-Beam
   TabIndex = 54
   Text = "0"
   Top = 1308
   Width = 1440
End
Begin VB.TextBox txtConc
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   Height = 252
   Index = 1
   Left = 6816
   MousePointer = 3 'I-Beam
   TabIndex = 53
   Text = "0"
   Top = 1308
   Width = 1500
End
Begin VB.TextBox
txtFeedVol
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   Height = 252
   Index = 0
   Left = 5376
   MousePointer = 3 'I-Beam
   TabIndex = 52
   Text = "0"
   Top = 2460
   Width = 1000
End
Begin VB.TextBox
txtAdsorberLengthLabel
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   BorderStyle = 0 'None
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 204
   Left = 5376
   Locked = -1 'True
   MousePointer = 1 'Arrow
   TabIndex = 55
   Text = "AdsorberLength L :
="
   Top = 1680
   Width = 2376
End
Begin VB.TextBox
txtFeedVol
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   Height = 252
   Index = 1
   Left = 6360
   MousePointer = 3 'I-Beam
   TabIndex = 50
   Text = "0"
   Top = 2460
   Width = 1000
End
Begin VB.TextBox
txtFeedVol
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   Height = 252
   Index = 2
   Left = 7344
   MousePointer = 3 'I-Beam
   TabIndex = 49
   Text = "0"
   Top = 2460
   Width = 972
End
Begin VB.TextBox Text6
   Appearance = 0 'Flat
   BeginProperty Font
      Name = "Small Fonts"
      Size = 6.6
      Charset = 0
      Weight = 400
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 204
   Left = 708
   MousePointer = 3 'I-Beam
   TabIndex = 48
   Top = 2664
   Width = 1215
End
Begin VB.TextBox Text14
   Appearance = 0 'Flat
   BeginProperty Font
      Name = "Small Fonts"
      Size = 6.6
      Charset = 0
      Weight = 400
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 250
   Left = 708
   MousePointer = 3 'I-Beam
   TabIndex = 33
   Top = 2112
   Width = 975
End
Begin VB.TextBox Text13
   Appearance = 0 'Flat
   BeginProperty Font
      Name = "Small Fonts"
      Size = 6.6
      Charset = 0
      Weight = 400
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 250
   Left = 708
   MousePointer = 3 'I-Beam
   TabIndex = 32
   Top = 1884
   Width = 975
End
Begin VB.TextBox Text12
   Appearance = 0 'Flat
   BeginProperty Font
      Name = "Small Fonts"
      Size = 6.6
      Charset = 0
      Weight = 400
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 250
   Left = 708
   MousePointer = 3 'I-Beam
   TabIndex = 27
   Top = 1656
   Width = 975
End
Begin VB.TextBox Text15
   Appearance = 0 'Flat
   BeginProperty Font
      Name = "Small Fonts"
      Size = 6.6
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 250
   Left = 708
   MousePointer = 3 'I-Beam
   TabIndex = 34
   Top = 2340
   Width = 975
End
Begin VB.TextBox Text7
   Appearance = 0 'Flat
   BeginProperty Font
      Name = "Small Fonts"
      Size = 6.6
      Charset = 0
      Weight = 400
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 250
   Left = 708
   MousePointer = 3 'I-Beam
   TabIndex = 14
   Top = 516
   Width = 975
End
Begin VB.TextBox Text9
   Appearance = 0 'Flat
   BeginProperty Font
      Name = "Small Fonts"
      Size = 6.6
      Charset = 0
      Weight = 400
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 250
   Left = 708
   MousePointer = 3 'I-Beam
   TabIndex = 16
   Top = 972
   Width = 975
End
Begin VB.TextBox Text11
   TabIndex = 28
   Top = 1668
   Width = 732
End
Begin VB.Label lb13
   Appearance = 0 'Flat
   BackColor = &H80000005&
   Caption = "KDK"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 252
   Left = 96
   MousePointer = 1 'Arrow
   TabIndex = 26
   Top = 1188
   Width = 732
End
Begin VB.Label lb12
   Appearance = 0 'Flat
   BackColor = &H80000005&
   Caption = "Dat-no"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 252
   Left = 96
   MousePointer = 1 'Arrow
   TabIndex = 25
   Top = 1428
   Width = 732
End
Begin VB.Label lb11
   Appearance = 0 'Flat
   BackColor = &H80000005&
   Caption = "KD"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 252
   Left = 96
   MousePointer = 1 'Arrow
   TabIndex = 24
   Top = 948
   Width = 732
End
Begin VB.Label lb10
   Appearance = 0 'Flat
   BackColor = &H80000005&
   Caption = "MW"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 252
   Left = 96
   MousePointer = 1 'Arrow
   TabIndex = 23
   Top = 708
   Width = 732
End
Begin VB.Label lb9
   Appearance = 0 'Flat
   BackColor = &H80000005&
   Caption = "KR"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 252
   Left = 96
   MousePointer = 1 'Arrow
   TabIndex = 22
   Top = 492
   Width = 732
End
Begin VB.Label lb8
   Appearance = 0 'Flat
   BackColor = &H80000005&
   Caption = "FR"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 252
   Left = 96
   MousePointer = 1 'Arrow
   TabIndex = 21
   Top = 252
   Width = 732
End
Begin VB.Frame Frame1
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = -1 'True
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H00000000&
   Height = 3084
   Left = 144
   MousePointer = 15 'Size All
   TabIndex = 0
   Top = 8088
   Width = 1575
Begin VB.CommandButton
Command1
   Caption = "2"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 204
   Left = 864
   MousePointer = 1 'Arrow
   TabIndex = 110
   ToolTipText = "diagram to
clipboard"
   Top = 1740
   Width = 216
End
Begin VB.CommandButton
cmdDrawMode
   Caption = "-> erasable"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 216
   Left = 132
   MousePointer = 1 'Arrow
   TabIndex = 98
   Top = 2568
   Width = 1332
End
Begin VB.CommandButton
End
   Appearance = 0 'Flat
   Caption = "end"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 852
   MousePointer = 1 'Arrow
   TabIndex = 10
   Top = 2784
   Width = 612
End
Begin VB.CommandButton
cls
   Appearance = 0 'Flat
   Caption = "cls"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 1080
   MousePointer = 1 'Arrow
   TabIndex = 6
   Top = 1752
   Width = 375
End
Begin VB.CommandButton
shrink
   Appearance = 0 'Flat
   Caption = "-"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 1308
   MousePointer = 1 'Arrow
   TabIndex = 39
   Top = 1332
   Width = 144
End
Begin VB.CommandButton
Grid
   Appearance = 0 'Flat
   Caption = "grid"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 852
   MousePointer = 1 'Arrow
   TabIndex = 31
   Top = 1332
   Width = 516
End
Begin VB.CommandButton
savePic
   Appearance = 0 'Flat
   Caption = "save pict"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 492
   MousePointer = 1 'Arrow
   TabIndex = 30
   Top = 1536
   Visible = 0 'False
   Width = 975
End
Begin VB.CommandButton
Breakb
   Appearance = 0 'Flat
   Caption = "break"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 132
   MousePointer = 1 'Arrow
   TabIndex = 29
   Top = 2784
   Width = 735
End
Begin VB.CommandButton
Befehl2
   Appearance = 0 'Flat
   Caption = ">>"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
```

```
   Height = 195
   Left = 1044
   MousePointer = 1 'Arrow
   TabIndex = 19
   Top = 1992
   Width = 408
End
Begin VB.CommandButton
Calculate
   Appearance = 0 'Flat
   Caption = "calculate new"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 132
   MousePointer = 1 'Arrow
   TabIndex = 9
   Top = 2388
   Width = 1335
End
Begin VB.HScrollBar
HBildlauf2
   Height = 240
   Left = 840
   Max = 14
   MousePointer = 1 'Arrow
   TabIndex = 4
   Top = 648
   Width = 615
End
Begin VB.HScrollBar
HBildlauf1
   Height = 240
   Left = 120
   Max = 1000
   Min = 1
   MousePointer = 1 'Arrow
   TabIndex = 3
   Top = 648
   Value = 371
   Width = 615
End
Begin VB.TextBox DatNr
   Appearance = 0 'Flat
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H00000000&
   Height = 285
   Left = 120
   MousePointer = 1 'Arrow
   TabIndex = 2
   Top = 408
   Width = 615
End
Begin VB.TextBox Farbe
   Appearance = 0 'Flat
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 285
   Left = 840
   MousePointer = 1 'Arrow
   TabIndex = 1
   Top = 408
   Width = 615
End
Begin VB.CommandButton
Befehl1
   Appearance = 0 'Flat
   Caption = "input-box"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 120
   MousePointer = 1 'Arrow
   TabIndex = 20
   Top = 1992
   Width = 948
End
Begin VB.CommandButton
zoom
   Appearance = 0 'Flat
   Caption = "+"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 720
   MousePointer = 1 'Arrow
   TabIndex = 38
   Top = 1332
   Width = 156
End
Begin VB.CommandButton
Recall
   Appearance = 0 'Flat
   Caption = "recall"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Left = 108
   MousePointer = 1 'Arrow
   TabIndex = 5
   Top = 1332
   Width = 624
End
Begin VB.CommandButton
Dwidth
   Appearance = 0 'Flat
   Caption = "width=1"
   Height = 195
   Left = 108
   MousePointer = 1 'Arrow
   TabIndex = 48
   Top = 1152
   Width = 1344
End
Begin VB.CommandButton
cmdPic
   Caption = "copy"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 204
   Left = 120
   MousePointer = 1 'Arrow
   TabIndex = 77
   ToolTipText = "diagram to
clipboard"
   Top = 1740
   Width = 972
End
Begin VB.Label Label3
   BackColor = &H000000FF&
   BeginProperty Font
      Name = "Courier New"
      Size = 9
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 195
   Index = 2
   Left = 780
   MousePointer = 1 'Arrow
   TabIndex = 73
   Top = 948
   Width = 156
End
Begin VB.Label Label3
   BackColor = &H00FF00FF&
   BeginProperty Font
      Name = "Courier New"
      Size = 9
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 132
   Index = 3
   Left = 312
   MousePointer = 1 'Arrow
   TabIndex = 72
   Top = 948
   Width = 156
End
Begin VB.Label Label3
   BackColor = &H00808080&
   BeginProperty Font
      Name = "Courier New"
      Size = 9
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 132
   Index = 4
   Left = 1248
   MousePointer = 1 'Arrow
   TabIndex = 71
   Top = 948
   Width = 156
End
Begin VB.Label Label3
   BackColor = &H0000FFFF&
   BeginProperty Font
      Name = "Courier New"
      Size = 9
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 132
   Index = 5
   Left = 936
   MousePointer = 1 'Arrow
   TabIndex = 70
   Top = 948
   Width = 156
End
Begin VB.Label Label3
   BackColor = &H0000FF00&
   BeginProperty Font
      Name = "Courier New"
      Size = 9
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 132
   Index = 6
   Left = 624
   MousePointer = 1 'Arrow
   TabIndex = 69
   Top = 948
   Width = 156
End
Begin VB.Label Label3
   BackColor = &H00000000&
   BeginProperty Font
      Name = "Courier New"
      Size = 9
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   Height = 132
   Index = 7
   Left = 156
   MousePointer = 1 'Arrow
   TabIndex = 68
   Top = 948
   Width = 156
End
Begin VB.Label lb1
   Appearance = 0 'Flat
   BackColor = &H80000005&
   Caption = "dat-no"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 252
   Left = 840
   MousePointer = 1 'Arrow
   TabIndex = 8
   Top = 168
   Width = 612
End
Begin VB.Label lb2
   Alignment = 2 'Center
   Appearance = 0 'Flat
   BackColor = &H80000005&
   Caption = "colour"
   BeginProperty Font
      Name = "MS Sans Serif"
      Size = 7.8
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 276
   Left = 2652
   TabIndex = 120
   Top = 192
   Visible = 0 'False
   Width = 1020
End
Begin VB.Image Image1
   Height = 2496
   Left = 6744
   Top = 72
   Width = 2688
End
Begin VB.Shape Shape1
   Height = 2292
   Left = 7500
   Top = 504
   Width = 3516
End
Begin VB.Label lbTransparent
   Alignment = 1 'Right Justify
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   BackStyle = 0 'Transparent
   Caption = "transp"
   ForeColor = &H80000008&
   Height = 228
   Left = 8376
   MousePointer = 1 'Arrow
   TabIndex = 108
   Top = 276
   Width = 552
End
Begin VB.Label lbLine
   Alignment = 2 'Center
   Appearance = 0 'Flat
   BackColor = &H80000005&
   BackStyle = 0 'Transparent
   Caption = "line-xy"
   ForeColor = &H80000008&
   Height = 240
   Left = 8424
   TabIndex = 106
   Top = 744
   Visible = 0 'False
   Width = 960
End
Begin VB.Line Line1
   Visible = 0 'False
   X1 = 9372
   X2 = 9660
   Y1 = 852
   Y2 = 2172
End
Begin VB.Label lbYValue
   Alignment = 2 'Center
   Appearance = 0 'Flat
   BackColor = &H80000005&
   BackStyle = 0 'Transparent
   Caption = "yyy"
   ForeColor = &H80000008&
   Height = 216
   Index = 0
   Left = 8652
   TabIndex = 105
   Top = 1092
   Visible = 0 'False
   Width = 528
End
Begin VB.Shape sh3
   Height = 24
   Index = 0
   Left = 8976
   Top = 1860
   Visible = 0 'False
   Width = 120
End
Begin VB.Label lbXValue
   Alignment = 2 'Center
   Appearance = 0 'Flat
   BackColor = &H80000005&
   BackStyle = 0 'Transparent
   Caption = "xxx"
   ForeColor = &H80000008&
   Height = 216
   Index = 0
   Left = 8700
   TabIndex = 104 Top = 2136
   Visible = 0 'False
   Width = 528
End
Begin VB.Image img1
   Height = 1752
   Index = 0
   Left = 10056
   Top = 624
   Visible = 0 'False
   Width = 552
End
Begin VB.Shape sh2
   Height = 120
   Index = 0
   Left = 9336
   Top = 1884
   Visible = 0 'False
   Width = 24
End
Begin VB.Image img2
   Height = 588
   Index = 0
   Left = 7872
   Top = 732
   Visible = 0 'False
   Width = 384
End
Begin VB.Label lbY2
   Alignment = 2 'Center
   Appearance = 0 'Flat
   BackColor = &H00C0C0C0&
   BorderStyle = 1 'Fixed Single
   Caption = "V/Vo"
   ForeColor = &H80000008&
   Height = 216
   Left = 7980
   TabIndex = 103
   Top = 1848
   Width = 468
End
Begin VB.Label lbX
   Alignment = 2 'Center
   Appearance = 0 'Flat
   BackColor = &H80000005&
   BackStyle = 0 'Transparent
   Caption = "W/Wo"
   ForeColor = &H80000008&
   Height = 216
   Left = 8004
   TabIndex = 102
   Top = 1512
   Width = 528
End
Begin VB.Label lbY
   Alignment = 2 'Center
   Appearance = 0 'Flat
   BackColor = &H80000005&
   BorderStyle = 1 'Fixed Single
   Caption = "A/Ao"
   ForeColor = &H80000008&
   Height = 216
   Left = 7980
   TabIndex = 101
   Top = 2172
   Width = 456
End
Begin VB.Shape sh1
   Height = 2136
   Left = 7716
   Top = 576
   Width = 3108
End
Begin VB.Label Label2
   Appearance = 0 'Flat
   BackColor = &H80000005&
   BackStyle = 0 'Transparent
   BorderStyle = 1 'Fixed Single
   Caption = "Label1"
   BeginProperty Font
      Name = "Courier New"
      Size = 7.2
      Charset = 0
      Weight = 400
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 4464
   Index = 0
   Left = 11748
   MousePointer = 15 'Size All
   TabIndex = 67
   ToolTipText = "dblClick: data
-> clipboard, right-button:
more data"
   Top = 4104
   Visible = 0 'False
   Width = 3804
End
Begin VB.Label Label1
   Appearance = 0 'Flat
   BackColor = &H80000005&
   BackStyle = 0 'Transparent
   BorderStyle = 1 'Fixed Single
   Caption = "Label1"
   BeginProperty Font
      Name = "Courier"
      Size = 9.6
      Charset = 0
      Weight = 400
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 7548
   Left = 10752
   MousePointer = 15 'Size All
   TabIndex = 66
   ToolTipText = "DoubleClick
to get data in clipboard"
   Top = 2952
   Width = 4548
End
Begin VB.Label
lbMoveProfile
   Alignment = 2 'Center
   Appearance = 0 'Flat
   BackColor = &H00FFFFFF&
   BackStyle = 0 'Transparent
   Caption = "VS-profile"
   BeginProperty Font
      Name = "Courier New"
      Size = 9
      Charset = 0
      Weight = 700
      Underline = 0 'False
      Italic = 0 'False
      Strikethrough = 0 'False
   EndProperty
   ForeColor = &H80000008&
   Height = 5520
   Left = 7260
   MousePointer = 15 'Size All
   TabIndex = 107
   Top = 84
   Width = 6060
End
End
Attribute VB_Name =
"Fenster"
Attribute
VB_GlobalNameSpace = False
Attribute VB_Creatable =
False
Attribute VB_PredeclaredId =
True
Attribute VB_Exposed = False
```

Anhang II _ Zwischenbericht 2002

BMBF – geförderte Forschung
Förderkennzeichen: 02 WA 0016

Zwischenbericht

zum
Forschungsvorhaben

In-situ Filterregeneration bei der TNT-Elimination aus Grundwasser: Anwendung des Verfahrens auf einen Faseraktivkohle-Adsorber

Projektleitung:

Prof. Dr.-Ing. W. Hegemann
Technische Universität Berlin
Institut für Technischen Umweltschutz
Fachgebiet Siedlungswasserwirtschaft

Dipl.-Ing. M. Hoff
Prof. Dr.-Ing. W. Hegemann

Berlin, Mai 2002

Inhalt

4 Einleitung

Das Forschungsvorhaben *„In-situ Filterregeneration bei der TNT-Elimination aus Grundwasser: Anwendung des Verfahrens auf einen Faseraktivkohle-Adsorber“* ist aus dem ebenfalls BMBF-geförderten Vorhaben *„Entwicklung eines Kombinationsverfahrens zur physikalischen, chemischen und biologischen Reinigung von mit Nitrotoluolen verunreinigten Grundwässern“* (Laufzeit: 1995-1998; Förderkennzeichen: 02WA9445/5) hervorgegangen. Die Laufzeit dieses Vorhabens beträgt 12 Monate. Der vorliegende Zwischenbericht faßt die Arbeiten seit August 2001 zusammen.

5 Zusammenfassung der Arbeiten des Vorgängerprojekts

Im Vorgängerprojekt wurde die chemische Regeneration granulierter Aktivkohlen (GAC) untersucht, die auf Rüstungsaltlasten zur Reinigung TNT-belasteter Grund- und Sickerwässer eingesetzt werden.

Es wurde ermittelt, wo, in welchen Mengen und in welcher Zusammensetzung diese toxischen Wässer anfallen und anhand wichtiger Beispiele zusammengefaßt, wie diese Wässer durch Filtration über Aktivkohlegranulat gereinigt werden.

Während die thermische Reaktivierung in zentralen Großanlagen zur Zeit das Standardverfahren für die Wiederaufbereitung dieser Aktivkohlen ist, wurde die chemische Regeneration in diesem Zusammenhang bisher nicht eingesetzt. Sie hat gegenüber der thermischen Reaktivierung eine Reihe potentieller Vorzüge, da sie in der Filtersäule und somit dezentral, vor Ort, ohne Transporte, Be- und Entladearbeiten durchgeführt werden kann und nicht die Rauchgas- oder CO_2-Emissionen verursacht, die bei der thermischen Reaktivierung durch Abbrand von Schadstoffen und eines Teils der Aktivkohle entstehen.

Es wurde daher untersucht, wie die chemische Regeneration von Aktivkohlen zur Bearbeitung dieses Umweltproblems eingesetzt werden kann. Es wurde dazu ein Verfahren entwickelt, das 1-molare Natronlauge bei Temperaturen von 85 - 105 °C im Aktivkohlebett einsetzt, um die adsorbierten Schadstoffe zu desorbieren. Diese Bedingungen führen zu einer Umsetzung (Hydrolyse) dieser Stoffe zu polaren, schwächer adsorbierbaren Verbindungen und zu einer Polarisierung der auf der Aktivkohleoberfläche lokalisierten Oberflächenoxide und somit zu einer Schwächung der Adsorptionskräfte.

Die desorbierten und hydrolysierten Stoffe können dann mit Wasserspülungen unter Beibehaltung der Hydrolyse-Temperaturen aus dem Filterbett ausgetragen werden. Eine sich anschließende Konditionierung mit 1-molarer Salzsäure depolarisiert die Aktivkohleoberfläche und stellt die adsorptiven Eigenschaften wieder her. An einer

kleintechnischen Anlage wurden für wiederholte Adsorptionen/Regenerationen nach diesem Verfahren Regenerationseffizienzen von 95 % der Beladungskapazität pro Regenerationszyklus gefunden.

Die aus dem Adsorbensbett entfernten Nitroaromaten fallen in Form großer Mengen tief rot-braun gefärbter, alkalischer Wässer an. Durch die alkalische Hydrolyse gehen zwar jegliche explosiven Eigenschaften der zuvor adsorbierten Sprengstoffe verloren, aber der TOC, der Mangel an biologischer Abbaubarkeit und die Toxizität werden nicht vermindert. Das Verfahren beinhaltet daher zwei weitere Komponenten, mit denen die Abwassermengen verringert, Regenerationslauge wiederverwendet und die Desorbate biologischem Abbau zugänglich gemacht werden können:

Mit einer Festphasenextraktion über Aluminiumoxid werden die Desorbat-beladenen Spülwässer abgereichert und können so wiederverwendet werden. Eine physikalische Nachbehandlung der beladenen Natronlauge, sowie der hochkonzentrierten Extrakte der Festphasenextraktion, die zu einem weitergehenden Abbau der Hydrolysate führt, wird eingesetzt, um die Lauge für weitere Regenerationen einsetzen zu können und um eine biologische Entsorgung der Schadstoffe zu ermöglichen. Mineralisierungsvorgänge sind sowohl für thermische Nachbehandlungen, als auch für Oxidationsverfahren nachgewiesen. Es wurde gezeigt, daß sowohl kombinierte Oxidationsverfahren mit H_2O_2 und Ozon, wie auch eine einfache UV-Behandlung, die bei 254 nm im Durchflußreaktor zur Entfärbung der Prozeßwässer führt, den Wiedereinsatz der Regenerationslauge ermöglichen.

Es wurden verschiedene Einsatzmöglichkeiten des Verfahrens diskutiert und aufgezeigt, daß zur Konfiguration des Verfahrens eine möglichst genaue Abschätzung der einzelnen Prozeßschritte zur sinnvollen Auslegung der Verfahrens-Komponenten notwendig ist, denn die Einrichtungen zur Abwasseraufbereitung sollten nur so groß ausgelegt werden, wie nötig, um die diskontinuierlich anfallenden Prozeßwässer in den Zeiträumen zwischen den Filterregenerationen vollständig umzusetzen. Für den zeitaufwändigsten Verfahrensschritt, der Filterbeladung, wurden daher Berechnungsgleichungen hergeleitet.

Die Berechnung des Adsorptionsvorganges zur Beschreibung des Durchbruchsverhaltens des Aktivkohlefilters führte zur Modellierung eines einfachen, neuen Berechnungsverfahrens, das gute Übereinstimmung mit Literatur- und Meßdaten zeigt und geeignet ist, das Verfahren einer Anwendung besser anzupassen. Darüber hinaus konnte mit dem Modell der Einfluß der Prozeßparameter wie Korngröße, Filtergeschwindigkeit und Adsorberabmessungen ohne weiteren Meßaufwand anschaulich simuliert werden und es wurden neue Fragestellungen deutlich, die auf die Bedeutung von Strömungsinhomogenitäten im Filter und die Art der Isothermenbestimmung bei Berechnungen dieser Art hinweisen. So wurden die Gültigkeitsgrenzen des Modells aufgezeigt und anhand einer Modellrechnung mit qualitativer Berücksichtigung einer Geschwindigkeitsverteilung des Fluids im Filter diskutiert, wie solche Modelle weiterentwickelt werden könnten.

Sowohl die Modellrechnungen, wie auch die Erfahrungen an der Laboranlage haben gezeigt, daß weitere, wesentliche Steigerungen der Effektivität des Verfahrens vor allem durch Verringerung der Diffusionsstrecken in der Aktivkohle, also durch geringere Korngrößen, zu erzielen sind. Das neue Adsorbens, Faseraktivkohle (ACF), bietet diese Eigenschaft, ohne, wie die Alternative Aktivkohlepulver, die Durchströmbarkeit des Filters wesentlich zu verschlechtern.

Anhand von Messungen zu Adorptionskinetik und -gleichgewicht wurde der Einsatz der faserförmigen Aktivkohle in dem Verfahren diskutiert. Dabei wurde deutlich, daß sich durch die automatisierbare in-situ Regeneration vor Ort kleine, sehr effizient arbeitende Filteranlagen mit kurzen Regenerationsintervallen verwirklichen lassen sollten.

6 Material und Methoden

An dieser Stelle sei ergänzend auf das Kapitel Material und Methoden im Schlußbericht des Vorgängerprojektes hingewiesen.

Faseraktivkohlen (ACF, activated carbon fabrics)

Als Adsorbens für die hier diskutierten Untersuchungen wurde das ACF-Gewebe CS 1501 von Actitex, Levallois/Frankreich eingesetzt. Diese Faseraktivkohle ist ein einfaches Gewebe aus Faserbündeln, die aus einigen hundert parallel angeordneten Karbon-Einzelfasern mit einem mittlerem Durchmesser von 10 µm bestehen. Bild 3-1 zeigt ein ACF-Gewebestück mit 2 cm Kantenlänge.

Bild 3-1: ACF-Gewebe CS 1501 (Gewebestück mit 2 cm Kantenlänge)

Applikation von ACF in Adsorbern

Während sich GAC-Adsorber prinzipiell beliebig skalieren lassen, ergeben sich für Bauform bzw. Modulgröße von ACF-Adsorbern Vorgaben durch die ACF-Makrostruktur. In der technischen Anwendung können z. B. Wickelmodule eingesetzt werden, deren Breite sich aus den Bahnbreite des ACF-Gewebes ergeben. Bei Festbettversuchen im Labormaßstab müssen zur Minimierung der Versuchsdauer

möglichst kleine Adsorber eingesetzt werden. Wickelmodule für diese Dimensionen müßten aus einzelnen Fasern hergestellt werden, hätten somit wenig geometrische Ähnlichkeit mit technischen Modulen und die Prüfung auf gleichmäßige Anströmung der ACF ließe sich nicht mit vertretbarem Aufwand verwirklichen. Es wurden daher (wie bei Versuchen mit granulierter Aktivkohle) kreiszylindrische Adsorbersäulen eingesetzt, die mit einzelnen Fasersträngen befüllt werden.

Der Mindestdurchmesser für GAC-Laboradsorber ist durch die "Faustregel" vorgegeben, den Filterdurchmesser mindestens 10-mal größer als den Adsorbenskorndurchmesser zu dimensionieren, da andernfalls Effekte wie Randgängigkeit und Kanalströmung die Messungen verfälschen. ACF hat zwar einen extrem geringen Adsorbensdurchmesser, aber aufgrund seiner Makrostruktur ist die Ausbildung eines homogenen Adsorbensbettes nicht bei jeden Adsorberabmessungen gegeben. In Vorversuchen mit Adsorberdurchmessern von 4 und 6 mm sowie Bettlängen von 14 und 20 mm wurden drei Applikationsmöglichkeiten verglichen: Verfilzte Fasern führten zwar zu einer homogenen Adsorbenspackung, aber die zerstörte Elastizität der Fasern führte nach wiederholten Regenerationen zu einer starken Kompression des Adsorbensbettes. Schwer handhabbar erwies sich ein Schüttgut aus geschnittenen Fasern. Am besten geeignet für Adsorptions- und Regenerationsversuche zeigte sich der Einsatz einzelner, langer Faserbündel.

Im Gegensatz zu GAC- sind ACF-Betten kompressibel, so daß durch die Bettabmessungen nicht gleichzeitig die Adsorbensmenge festgelegt ist. In dieser Arbeit wurden die Adsorberbetten so gepackt, daß die Bettkontraktion durch den Strömungsdruck während der Filtration zwischen 5 und 10 % beträgt: Die 4 x 20 mm Adsorbersäulen wurden mit 41 mg ACF (2 Faserbündelstränge von jeweils 525 mm) bepackt, woraus sich eine Bettdichte von163 kg/m³ ergibt. (Zum Vergleich: Mit dem Schüttgut aus auf eine Länge von 3-4 mm geschnittenen Fasern ergäbe sich eine Bettdichte von 188 kg/m³, für die GAC F-400 478 kg/m³.)

Messung von Durchbruchskurven (DBKs)

Die Durchbruchskurven wurden nach dem in Bild 3-2 schematisch dargestellten Versuchsaufbau durchgeführt. Die TNT_{aq}-Lösung wird aus einer 20 L-Vorlage mit einer pulsationsfreien, volumenkonstanten HPLC-Pumpe über einen keramischen Vorfilter, die Adsorbersäule und einen modifizierten HPLC-Autosampler gefördert. Der Volumenstrom betrug in den dargestellten Durchbruchskurven 2,22 ml/min (+/- 4 %), woraus sich eine Leerrohrgeschwindigkeit von 10,6 m/h ergibt. Es wurde eine TNT-Konzentration von 8,8 mg/L (+/- 2%) eingesetzt.

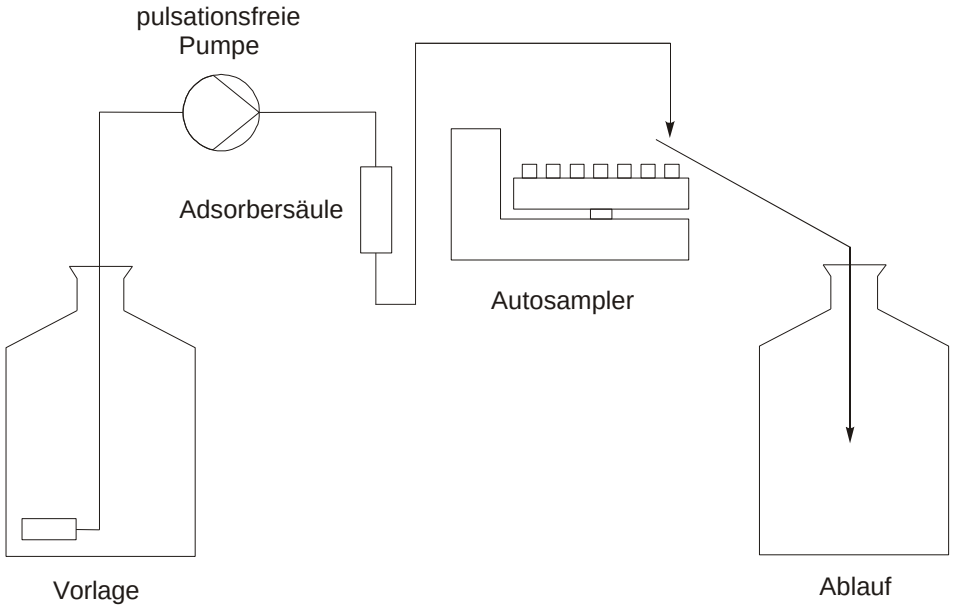


Bild 3-2: Schematischer Versuchsaufbau zur Messung von Durchbruchskurven

Bild 3-3 zeigt die Konstruktion der 4 x 20 mm Adsorbersäule, die mit 41 mg ACF gefüllt wurde (siehe oben). Das Adsorbensbett wird durch zwei 5 µm Edelstahlsiebe gehalten. Mediumsberührte Teile der Adsorbersäule, sowie des gesamten Versuchsaufbaus bestehen ausschließlich aus Teflon, Edelstahl oder Glas. Die luftdicht verschlossenen Probengefäße sind der Raumtemperatur im Autosampler maximal 24 h ausgesetzt und werden bis zur Analyse bei 4 °C aufbewahrt.

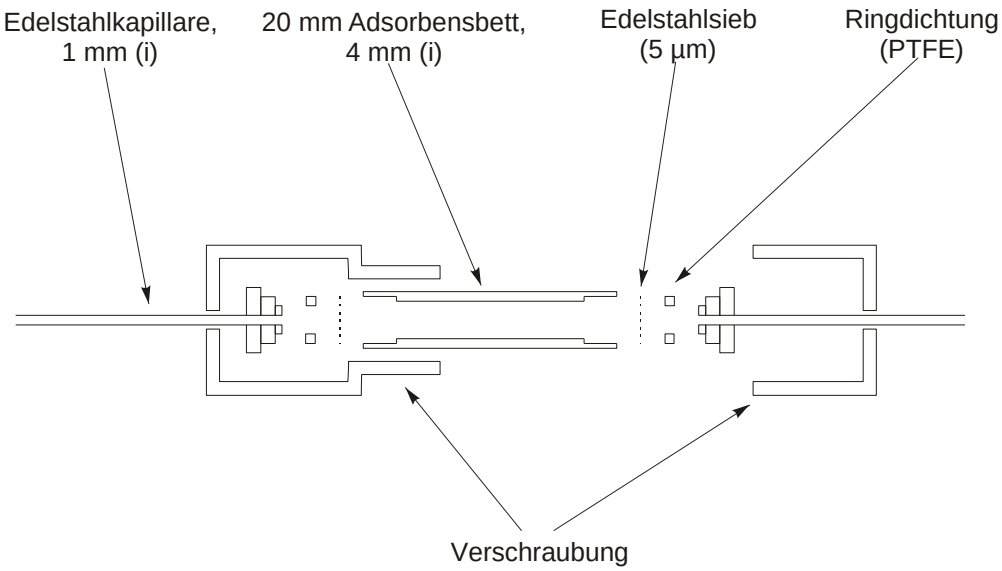


Bild 3-3: Konstruktion der ACF-Adsorbersäule

Adsorbensregeneration

Zur Regeneration des ACF in den Versuchen zum wiederholten Be- und Entladen des Adsorbers wurden die ACF-Stränge der Adsorbersäule entnommen und in 50 ml Erlenmeyerkolben mit Magnetrührfiche und Rückflußkühler weiterbehandelt.

Standard-Regenerationsprotokoll

Das in dieser Arbeit als Standard-Regenerationsprotokoll angesprochene Versuchsprotokoll ist an die im Vorgängerprojekt entwickelte Behandlungsroutine zur chemischen Regeneration granulierter Aktivkohle angelehnt (Tabelle 1). Das eingesetzte Fluidvolumen beträgt jeweils 5 ml.

Tabelle 1: Standard-Regenerationsprotokoll

Schritt	Dauer	Medium	Temp.
1	2 h	1 n NaOH	siedend
2	2 h	1 n NaOH	siedend
3	1 h	Bidest	siedend
4	1 h	Bidest	siedend
5	1 h	Bidest	siedend
6	1/2 h	1 n HCl	Raumtemp.
7	1/4 h	Bidest	Raumtemp.

Dieses Protokoll wird auch hier zur Regeneration der Faseraktivkohle verwendet und ist Ausgangspunkt für die Anpassung des Verfahrens.

TNT-Analytik

Die TNT-Gehalte wurden chromatographisch bestimmt. Eingesetzt wurde ein HPLC der Fa. Dionex-Softron, Germering mit Gradientenpumpe (Type M480), Diodenarraydetektor (Typ UVD 340S), Autosampler, Steuer- und Auswerte-Soft/Hardware, sowie ein Degasser der Fa. Optilab.

Alle Analysen wurden mit Methanol/Wasser-Eluenten und einer 4 x 250 mm LiChrospher -gefüllten Eco-Trennsäule (100 Å RP18, 5 µm) mit 10 mm Vorsäule gleicher Füllung durchgeführt. Die TNT-Konzentrationen der Adsorptionsversuche wurden mit einem isokratischen (15 % Wasser) 5-Minuten Programm bestimmt. Die Proben wurde jeweils 3-fach vermessen (drei Injektionen). Trübstoffe wurden mit 0,2 µm Teflonmembran- Einwegspritzenfiltern entfernt.

TNT-Lösungen

TNT wurde in technischer Reinheit (TNT_{tech}) als rezirkuliertes Material militärischer Qualität aus dem Chemiewerk Schönebeck über die BAM, Berlin bezogen. Konzentrierte TNT_{aq}-Stammlösung wurde bei 90 °C im Rundkolben durch quantitatives Lösen von 1 g TNT_{tech} in 10 L Bidest hergestellt. Die in den Versuchen eingesetzten TNT-Lösungen wurden durch Verdünnung dieser Stammlösung mit Bidest angesetzt.

7 Anwendung des Regenerationsverfahrens auf ACF-Adsorber

7.1 Regeneration nach Standard-Protokoll

Bild 4-1 zeigt die Verläufe der Durchbruchskurven (DBK) während der Adsorberbeladung für vier in Folge durchgeführte Adsorptions-/Regenerationszyklen. Der S-förmige Verlauf von DBK6-1 (Beladung unbehandelter ACF) zeigt die Eignung der für das Versuchssystem gewählten Strömungsgeschwindigkeit, Zulaufkonzentration und Adsorberabmessungen, denn die Stoffübergangszone in dem kurzen 20 mm-Adsorber kann sich gut ausbilden.

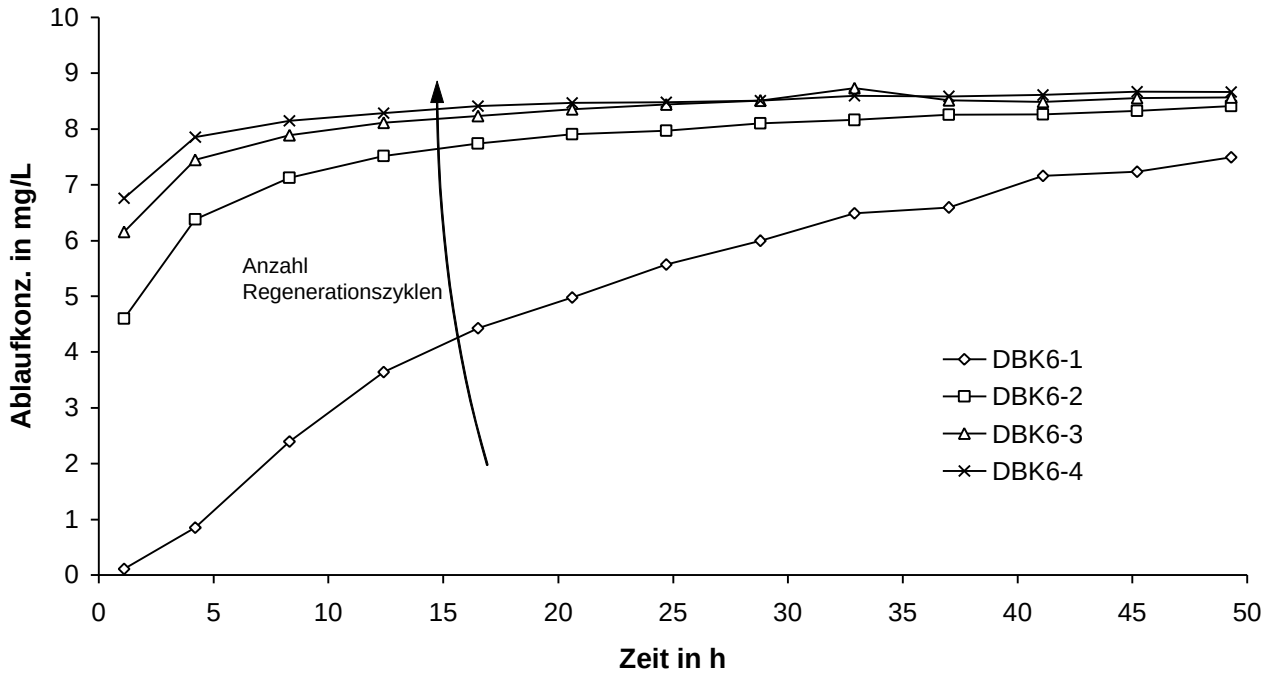


Bild 4-1: Durchbruchskurvenverläufe nach wiederholten Standardregenerationen

DBK6-1 weist praktisch keine Schlupfkonzentration (Ablaufkonzentration zur Zeit t_0) auf und bestätigt damit die Überlegenheit der ACF-Kinetik gegenüber der von granulierter Aktivkohle (GAC). Nach Modellrechnungen (nicht gezeigt) läge die Anfangsablaufkonzentration bei Einsatz von GAC in diesem System bereits bei über

3 mg/L. Bild 4-2 gibt die Beladungsverläufe zu diesen Versuchen wieder. Schnelle Kinetik und hohe Beladungskapazität führen demnach bei der gewählten Zulaufgeschwindigkeit und –konzentration innerhalb von 48 h zu Beladungen der ersten 2 cm eines Adsorbers unbehandelter ACF von über 50 gew.-%.

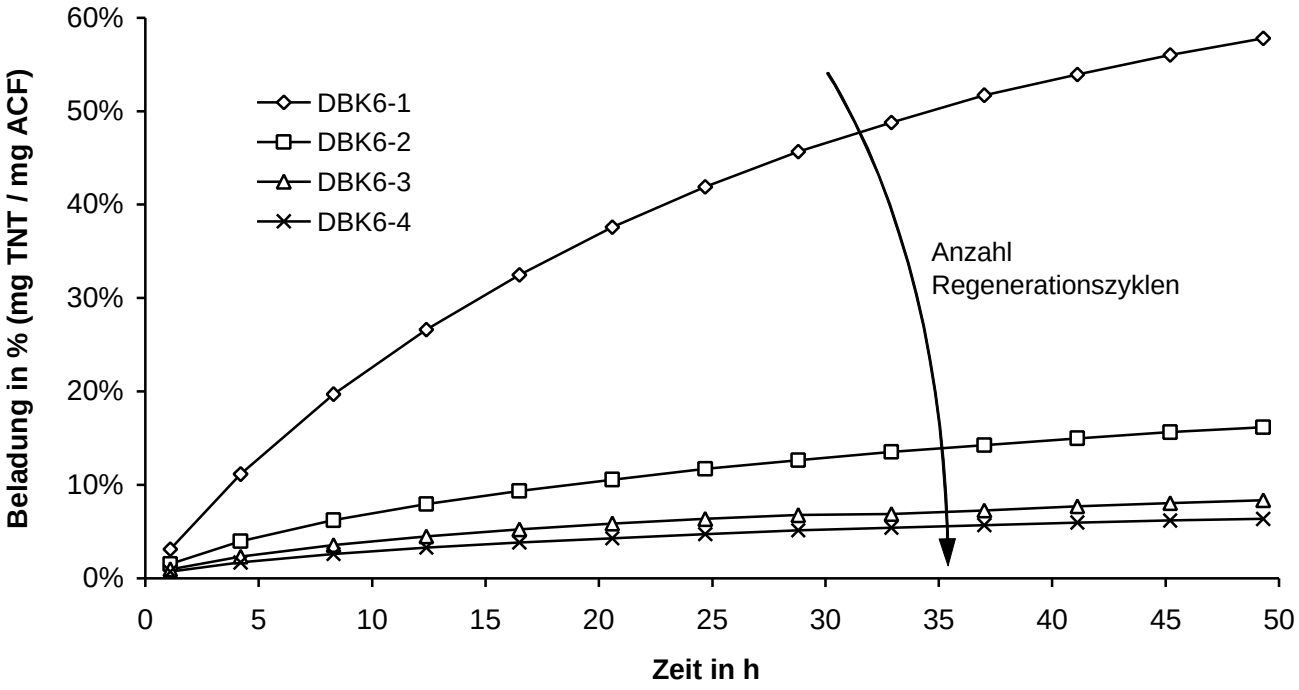


Bild 4-2: Beladungsverläufe nach wiederholten Standardregenerationen

Die Regeneration des beladenen ACF-Bettes nach dem Standard-Regenerationsprotokoll der GAC-Adsorber führt zunächst zu sehr großen Adsorptionskapazitätsverlusten (DBK6-2 in Bild 4-1 und 4-2), während diese bei weiteren Adsorptions-/Regenerationszyklen deutlich geringer ausfallen (DBK6-3 und -4).

7.2 Modifikation des Regenerationsverfahrens

Offensichtlich reagiert Faseraktivkohle auf die bisher (im Vorgängerprojekt) auf granulierte Aktivkohle angewendete Regenerationsmethode völlig anders als GAC. Auf der Suche nach den Ursachen hierfür wurde als erster Schritt das Regenerationsprotokoll modifiziert und auf den beladenen Adsorber der in Bild 4-1 und Bild 4-2 dargestellten Versuchsreihe angewendet (DBK6-5 in Bild 4-3): Alle Schritte der Regeneration wurden deutlich verlängert. Vor allem der letzte Spülschritt wurde auf 3 Tage ausgedehnt, um zur Desorption der Lauge eine vollständige Gleichgewichtseinstellung von wäßriger und Adsorbens-Phase sicherzustellen.

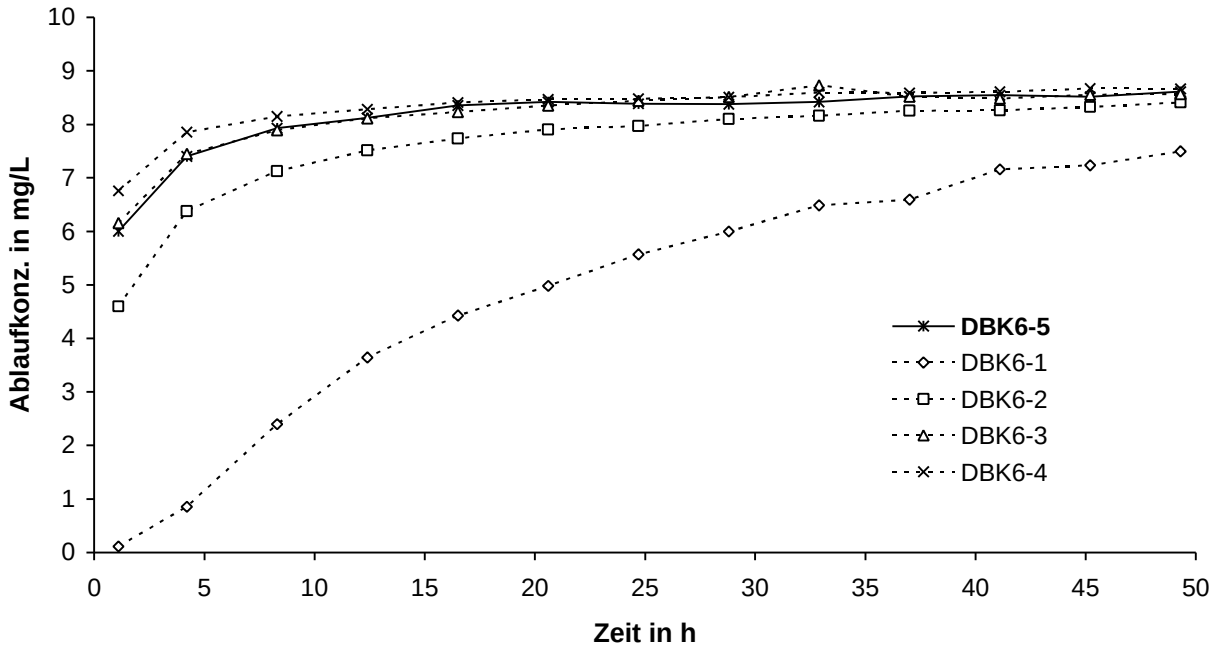


Bild 4-3: Einfluß der Regenerationsdauer. DBK6-5: 6 h NaOH, 16 h Wasserspülung, 3 d Equilibrierung in Wasser. Zum Vergleich die 4 vorhergegangenen Adsorptions-/Regenerationszyklen nach Std.-Protokoll

Festzustellen ist, daß durch dieses Vorgehen das Regenerationsergebnis wesentlich verbessert werden kann, denn statt der für eine Standard-Regeneration erwartete Ablaufkonzentration oberhalb von DBK6-4 liegen die Ablaufkonzentrationen noch unterhalb von DBK6-3. Gegenüber den Adsorptionseigenschaften vor dieser Regeneration sind also keine weiteren Adsorptionskapazitätsverluste aufgetreten,

sondern die der letzten beiden Adsorptions-/Regenerationszyklen rückgängig gemacht.

Nicht geklärt wird hierdurch, ob die beobachteten Adsorptionskapazitätsverluste eher auf eine unvollständige Desorption bzw. eine Readsorption hydrolysierten Adsorptivs oder auf eine bleibende Veränderung der adsorptiven Eigenschaften des ACF durch die Regenerationsprozedur selbst zurückzuführen ist.

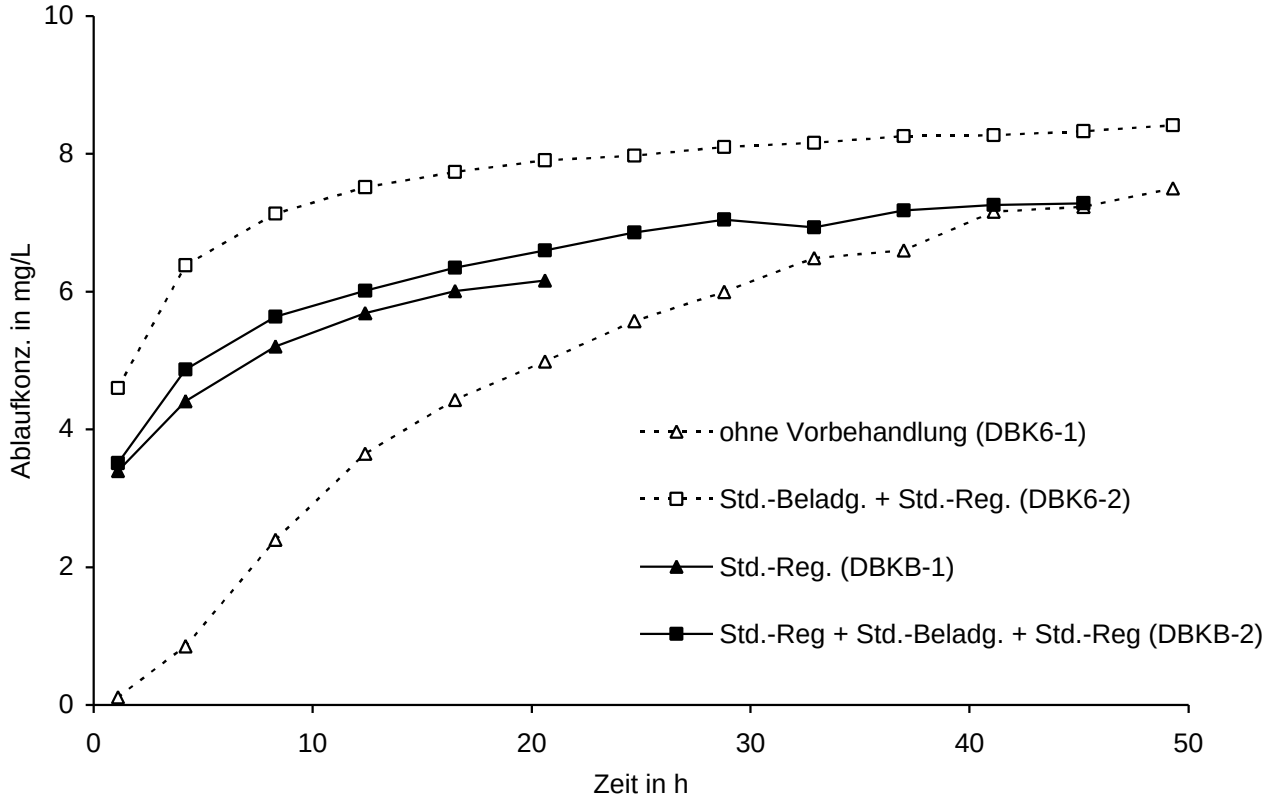


Bild 4-4: Blindversuch mit dem Std.-Reg.-Protokoll. DBKB-1: DBK-Verlauf nach Regeneration unbeladener ACF mit Std.-Protokoll; DBKB-2: DBK-Verlauf nach Regeneration der beladenen ACF von DBKB-1; zum Vergleich: DBK-Verlauf vor (DBK6-1) und nach (DBK6-2) einer Regeneration mit Std.-Protokoll.

Es wurde daher ein Blindversuch (bei dem unbeladene ACF nach dem Standard-Regenerationsprotokoll behandelt wird) durchgeführt, der eher letztere Annahme bestätigt (DBKB-1 in Bild 4-4): Allein die Regenerationsprozedur verursacht bereits einen großen Adsorptionskapazitätsverlust. Dieses so behandelte ACF-Bett konnte dann fast ohne weitere Adsorptionskapazitätsverluste regeneriert werden. DBKB-2

zeigt die Wiederbeladung. Die Anfangsablaufkonzentrationen von DBKB-1 und DBKB-2 sind fast identisch. Daß dennoch nicht die Regenerationsprozedur allein die Adsorptionskapazitätsverluste verursacht, zeigt der Vergleich von DBKB-2 und DBK6-2: Beide Kurven stellen eine DBK nach einer einmaligen Regeneration dar, aber DBK6-2 verläuft deutlich ungünstiger, was nur mit der Beladung zu erklären ist, die seiner Regeneration vorangegangenen war.

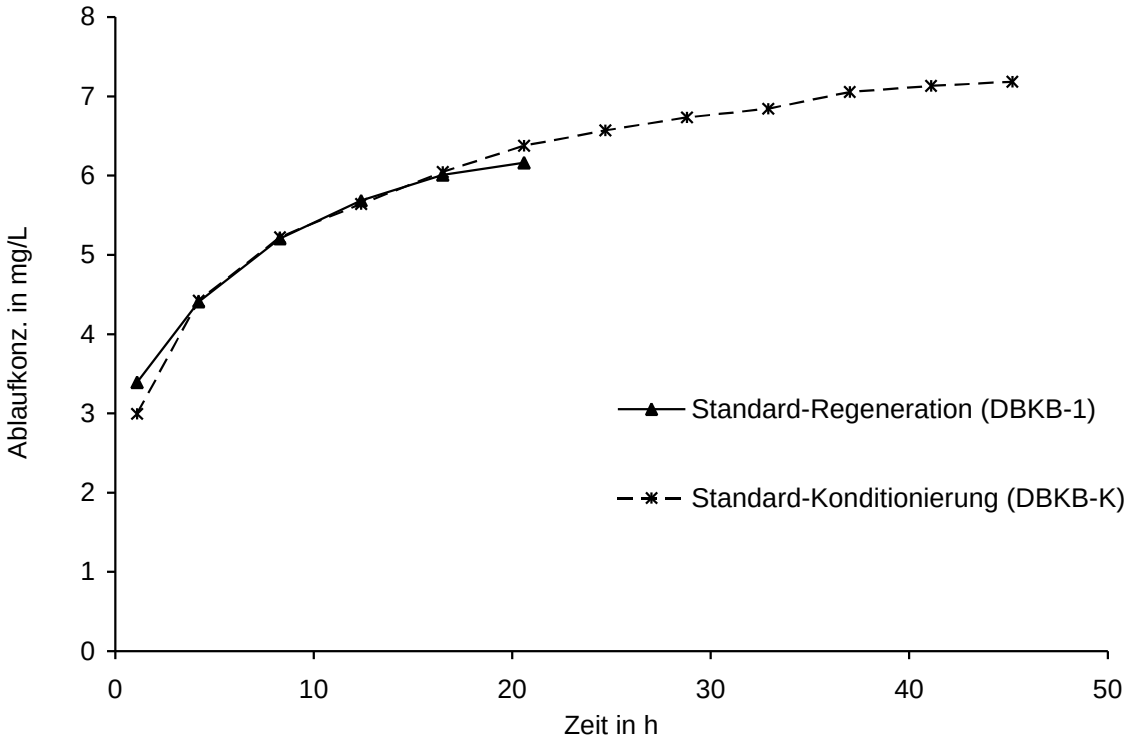


Bild 4-5: Durchbruchskurve nach Blindversuch zur Konditionierung gemäß Std.-Reg.-Protokoll im Vergleich zur DBK nach Blindversuch mit (vollständiger) Standard-Regeneration.

Bild 4-5 gibt einen unerwarteten Hinweis darauf, wo das Problem der ACF-Regeneration nach dem Standard-Protokoll liegt. Dargestellt sind zwei ACF-DBKs, die zuvor als Blindversuche nach dem Standard-Protokoll regeneriert (DBKB-1) bzw. nach dem Standard-Protokoll nur konditioniert (DBKB-K) wurden: Entgegen den Erfahrungen mit granulierter Aktivkohle, die grundsätzlich verbesserte Adsorptionseigenschaften nach einer Säurebehandlung aufweist, sind die Regenerationsprozedur-bedingten Adsorptionskapazitätsverluste hier offensichtlich vollständig auf die Säurekonditionierung zurückzuführen. Der vorhergehende Schritt

der Laugenbehandlung, der sowohl bei GAC, als auch bei ACF eine massive Schwächung der Adsorptionskräfte und die erwünschte Desorption bewirkt, zeigt sich beim ACF als vollständig reversibel und ohne bleibenden Einfluß auf das Adsorbens.

Die GAC-Regenerationsmethode muß also hinsichtlich der Konditionierung der ACF angepaßt werden.

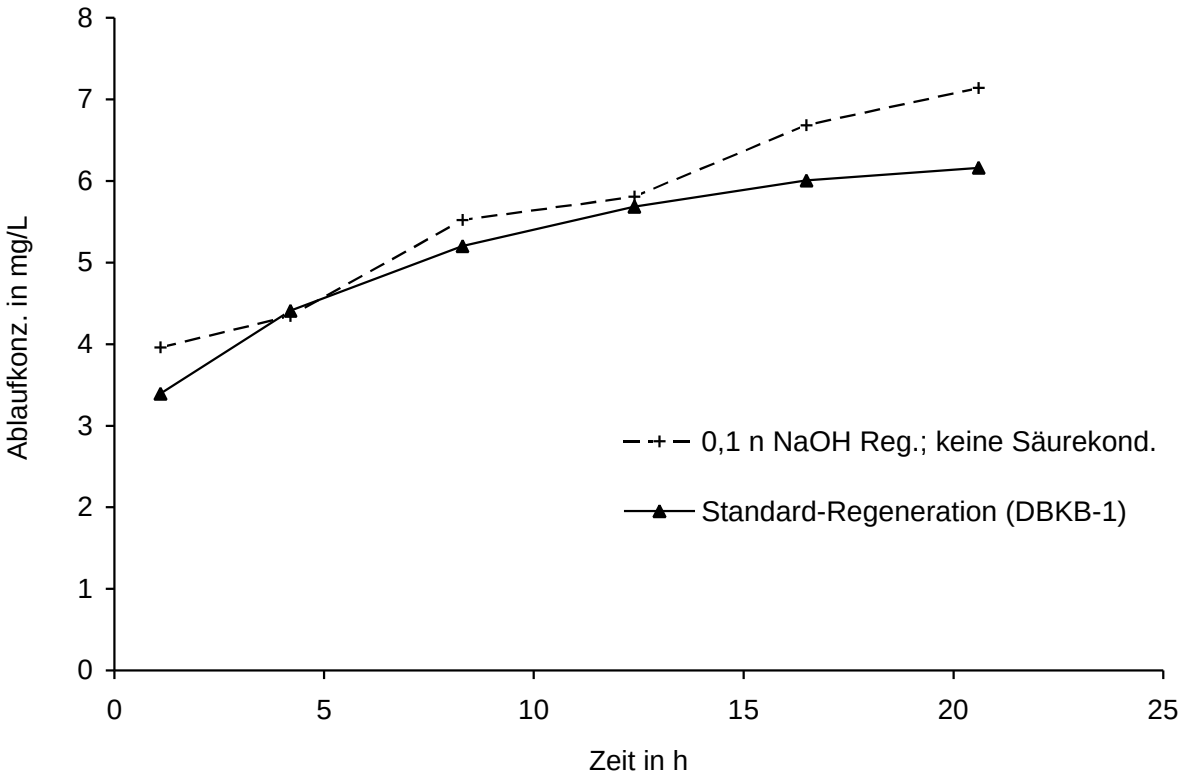


Bild 4-6: Durchbruchskurve nach Vorbehandlung unbeladener ACF mit einer Regeneration ohne Säurekonditionierung im Vergleich zur DBK nach Std.-Regeneration.

Es wurde zunächst ausgeschlossen, daß auf die Konditionierung ganz verzichtet werden könnte, indem eine wesentlich geringere NaOH-Konzentration eingesetzt würde und die Wasserspülungen solange wiederholt würden, bis der Filterablauf pH-neutral wäre (Bild 4-6). Im nächsten Schritt wurde die Konditionierung selber modifiziert. Geklärt werden sollte, ob einerseits eine Konzentrationsverringerung zu einer Verbesserung führen kann und ob es andererseits sinnvoll ist, die Salzsäure durch ein anderes Konditionierungsagens zu substituieren. Die Auswahl alternativer

Säuren beruht auf der Arbeitshypothese, daß die für die Adsorption wichtigen Oberflächenoxide (siehe Schlußbericht zum vorangegangenen Projekt) der ACF anders als bei GAC durch die leicht reduzierende Eigenschaften des HCl nachhaltig geschädigt werden könnten. Die Untersuchungen zur Konditionierung wurden daher neben Schwefelsäure und Salzsäure mit stark oxidierender Salpetersäure und einem Gemisch aus Salzsäure und dem Oxidationsmittel Wasserstoffperoxid durchgeführt.

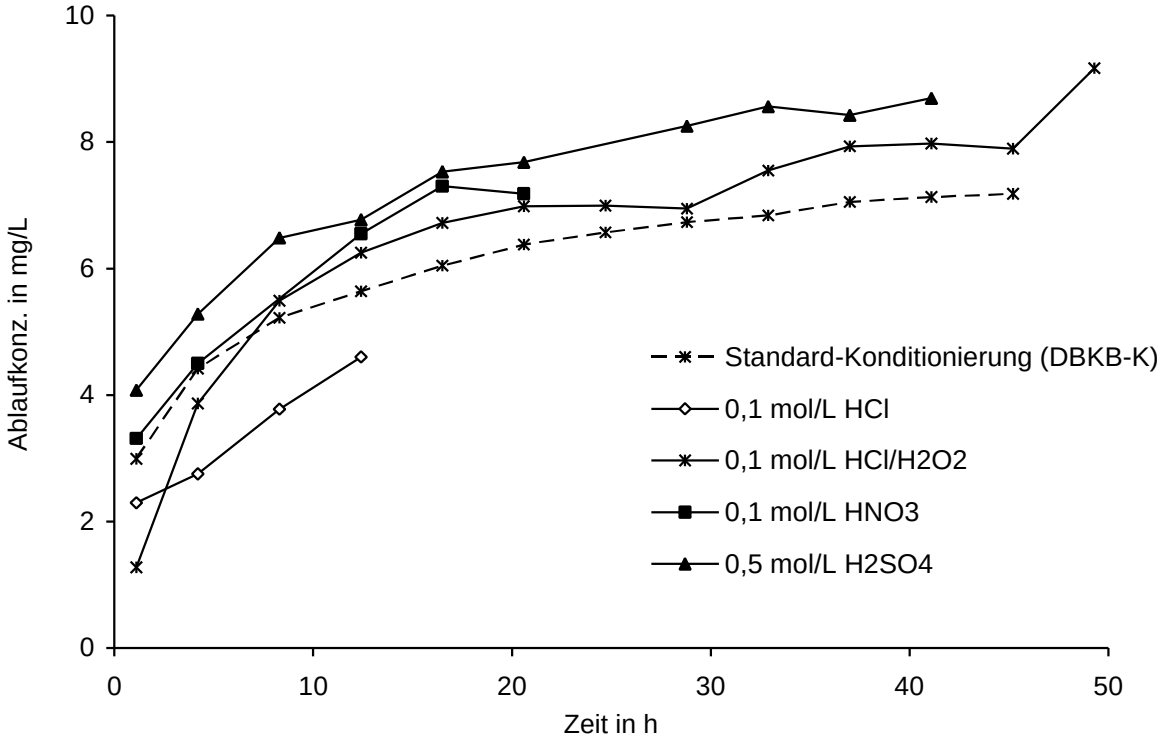


Bild 4-7: Alternativen zur Konditionierung mit 1-molarer Salzäure
(Durchbruchskurven nach Vorbehandlung unbeladener ACF)

Bild 4-7 zeigt die DBKs von entsprechend behandelten (zuvor unbeladenen) ACF-Betten. Abgesehen von der Standard-Konditionierung wurde bei allen Versuchen die gleiche Hydronium-Ion Konzentration (bei Annahme einer vollst. Dissoziation) vorgegeben. Tatsächlich führt die Kombination HCl/H_2O_2 zu den niedrigsten Anfangsablaufkonzentrationen. Diese sind jedoch vermutlich nicht auf eine verbesserte Rekonstitution der Oberflächenoxide, sondern auf Oxidationsreaktionen des Adsorptivs mit Restkonzentrationen des H_2O_2 zurückzuführen, denn die Ablaufkonzentrationen steigen schnell an. Zwar sind die Adsorptionskapazitätsverluste nach einer Konditionierung mit der vergleichsweise

gering oxidativ-wirkenden Schwefelsäure am höchsten, aber das Oxidationsmittel HNO_3 bietet gegenüber der Standard-Konditionierung ebenfalls keine Verbesserung.

Die wirkungsvollste Modifikation der Regenerationsmethode besteht also in der ebenso einfachen wie günstigen Lösung, die HCl-Konzentration der Säurekonditionierung herabzusetzen.

Zur Zeit wird untersucht, wie weit sich auch die Laugenkonzentration verringern läßt. Es ist geplant, das modifizierte Regenerationsprotokoll analog der in Kapitel 4.1 dargestellten Versuchsreihe anzuwenden, um so die Regenerationseffizienz des Verfahrens abschließend zu bewerten.

8 Arbeiten zum Berechnungsmodell und Prozeßmodellierung

Es wurden bereits umfangreiche Arbeiten zur Weiterentwicklung des im Vorgängerprojekt dargestellten Berechnungsverfahrens zum Beladungsvorgang des Adsorbers und zur Prozeßmodellierung vorgenommen:

1) Der verzögerte Austrag von stark alkalischem Desorbat aus dem Adsorbensbett wurde auf die Deprotonierung der Aktivkohleoberflächenoxide zurückgeführt und mathematisch modelliert. Zur Auslegung der nachfolgenden Verfahrenskomponenten Festphasenextraktion und UV-Behandlung kann so Zulaufvolumen und -konzentration genauer bestimmt werden.

2) Das Programm zur Durchbruchskurvenberechnung wurde verbessert und arbeitet jetzt bis zu 10-mal schneller. Dazu wurde ein Teil des Programms in vollständig Maschinensprache-kompilierbare Programmiersprache umgesetzt (Auslagerung eines Teils des Visual-Basic Programms in eine sog. C++ Dll). Außerdem können jetzt Beladungs- und Konzentrationsprofile im Adsorber berechnet und dargestellt werden.

3) Auf Anregung von Herrn Prof. Worch, Institut für Wasserchemie, TU-Dresden wurde zur Validierung des im Schlußbericht des Vorgängerprojekts dargestellten Modells eine wichtige Randbedingung experimentell untersucht.

Zu (1) und (2) werden zur Zeit noch Messungen durchgeführt. Zugunsten einer geschlossenen Darstellung wird daher hier auf eine eingehende Diskussion verzichtet und auf den Schlußbericht verwiesen. Die Untersuchungen zu (3) sind abgeschlossen und werden im Folgenden kurz abgebildet:

Im Modell wird aufgrund von Systemvergleichen angenommen, daß im untersuchten System keine Korndiffusionslimitierung vorliegt. Im Vorgängerprojekt wurde daher für die Abweichung in der zweiten Hälfte des DBK-Verlaufs von berechneten und

gemessenen Daten eine andere Erklärung entwickelt (stationäre Geschwindigkeitsverteilung der Adsorberströmung), da diese typische Abweichung in der Regel mit dem Vorliegen einer Korndiffusionslimitierung begründet wird. Um die Gültigkeit dieser Erklärung zu überprüfen, wurde ein Versuch durchgeführt, der auf experimentellem Weg den Nachweis von Existenz bzw. Abwesenheit der Korndiffusion ermöglicht. Der Versuch wurde an dem System durchgeführt, anhand dessen das Modell ursprünglich entwickelt wurde (granuliertes Adsorbens). Da der Korndiffusionseinfluß mit abnehmenden Adsorbensdurchmesser geringer wird, würde der Nachweis der Abwesenheit der Korndiffusionslimitierung bei GAC auch für ACF gelten. Umgekehrt müßte der Versuch allerdings für ACF wiederholt werden, wenn die Messung für GAC das Vorliegen von Korndiffusionslimitierung anzeigt.

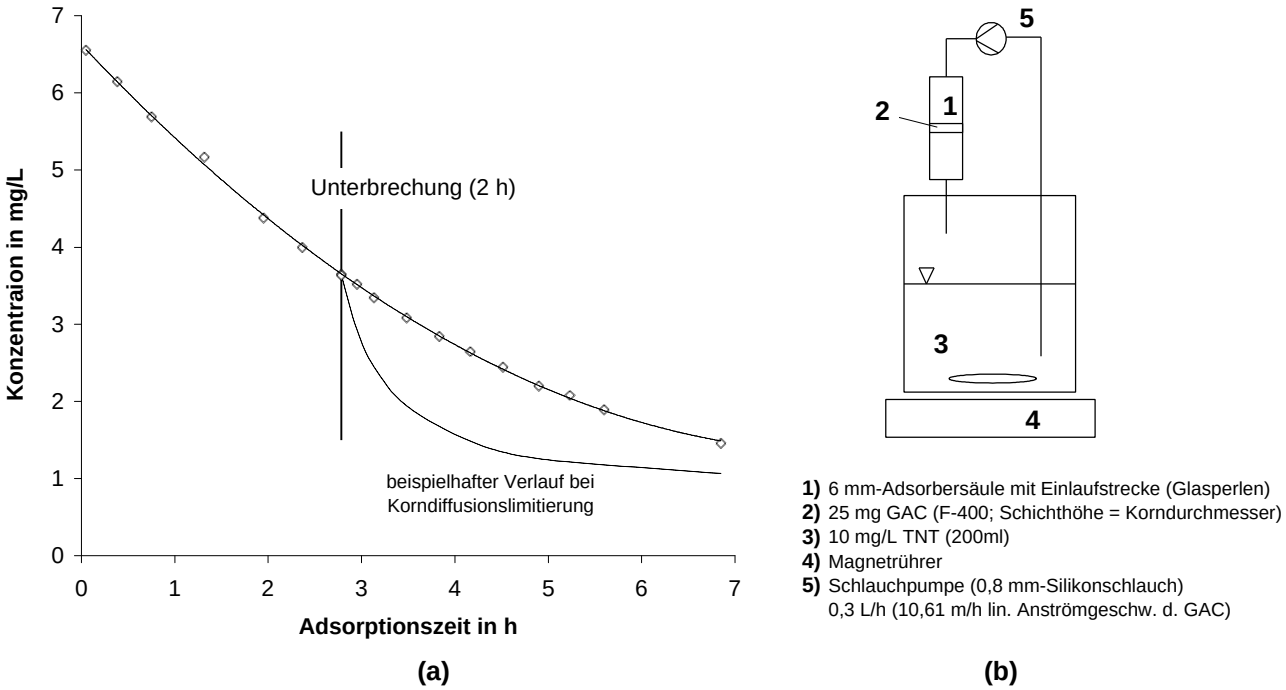


Bild 5-1: Unterbrechungsversuch im Differentialfestbett-Umlaufadsorber. Meßergebnisse (a) und schematische Versuchsanordnung (b)

Mit einem sogenannten Unterbrechungsversuch wird im Batchverfahren die Konzentrationsabnahme einer (TNT-) Lösung gegen die Zeit durch Abreicherung mit einem Adsorbens (hier Aktivkohle) gemessen. Während der Unterbrechung des Versuchs, bei der Adsorbens und Lösung voneinander getrennt sind, verläuft die Korndiffusion, also der Konzentrationsausgleich im Adsorbenskorn, weiter. Lag vor

der Unterbrechung eine Korndiffusionslimitierung vor, das heißt, war der Konzentrationsausgleich im Korn im Vergleich zum äußeren Stoffübergang langsam, erfolgt nach der Unterbrechung ein steiler Konzentrationsabfall, bis sich wieder ein Konzentrationsgradient im Korn aufbaut.

Um die selben Strömungsbedingungen wie im Filterbett zu gewährleisten, wurde der Versuch nicht im Rührkessel ausgeführt, sondern in einem Differentialfestbett-Umlaufadsorber (Bild 5-1b). Bild 5-1a zeigt den Konzentrationsverlauf während der Adsorption: Der Kurvenverlauf bleibt von der (zweistündigen, auf der Adsorptions-Zeitachse nicht dargestellten) Unterbrechung unbeeinflußt. Die Ausgleichsvorgänge im Korn sind schnell im Verhältnis zum Stoffübergang und haben keinen signifikanten Einfluß auf diesen. Das System ist Filmdiffusions-limitiert, die Korndiffusion ist, wie aufgrund von Systemvergleichen angenommen, vernachlässigbar.

Berlin, den 31.05.2002

Dipl.-Ing. M. Hoff

Prof. Dr.-Ing. W. Hegemann