

IBJpluris 2.7

Eine Implementierung des Ausbreitungsmodells PLURIS

(Janicke, U., Janicke, L., Atmospheric Environment 35, 2001, 877-890)

Ing.-Büro Janicke, Überlingen, November 2010

Vorbemerkung	1
1 Einleitung	1
2 Das Programm	2
2.1 Benutzeranleitung	3
2.2 Eingabeparameter	4
2.3 Meteorologische Profile	9
Anhang	10
A Dateistruktur von DMN-Dateien	10



Vorbemerkung

Die Urheberrechte des Computerprogramms IBJpluris liegen beim Ingenieurbüro Janicke, Überlingen. Das Programm einschließlich des Quelltextes ist unter der GNU PUBLIC LICENCE verfügbar, die mit dem Programm verteilt wird. Das bedeutet im wesentlichen, daß es frei kopiert und weitergegeben werden darf, sofern dies kostenlos und wieder unter der GNU PUBLIC LICENCE erfolgt. Falls das Programm modifiziert und weitergegeben wird, ist der Programmname zu ändern, insbesondere darf der neue Name nicht mit „IBJ“ beginnen.

Diese Dokumentation, das Programm und Beispielrechnungen sind im Internet unter <http://www.janicke.de> verfügbar. Dort stehen auch Anmerkungen, Korrekturen und Beiträge Dritter.

Diese Dokumentation bezieht sich auf die Programmversion 2.7. Wichtige Änderungen zur Vorgängerversion 2.5:

- Andere bzw. neue Bedeutung der Quellparameter qq, sq, lq.
- Erweitertes Ausgabeformat bei oo = 1.
- Zusätzliches Ausgabeformat oo = 5 (Ausgabe der berechneten Überhöhung und der Überhöhung nach den Formeln von Briggs).
- Bestimmung des Temperaturprofils, wenn nicht explizit oder über einen Gradienten vorgegeben, nach KTA 1508 anstelle der Ähnlichkeitsfunktionen (siehe Fußnote in Abschnitt 2.3).
- Aufrufoption -f: Beendigung der Rechnung (*final rise*) nach den Kriterien von VDI 3945 Blatt 3.
- Beendigung der Rechnung, wenn die Fahnenachse die Höhe z_{\max} über Grund überschreitet (Standardwert 1100 m, ein anderer Wert kann mit der Aufrufoption -z festgelegt werden).

1 Einleitung

Ein Fahnenüberhöhungsmodell beschreibt die Auswirkung der dynamischen Eigenschaften der Abluft (Impuls-, Temperatur- und Feuchtedifferenz zur Umgebungsluft) auf den mittleren Verlauf der Fahnenachse. Nach der Freisetzung wird Umgebungsluft in die Fahne eingemischt, so daß sich die über den Fahnenquerschnitt gemittelten Parameter wie Impuls, Temperatur und Feuchte sowie der Fahnenradius ändern. Dies führt im Vergleich zu einer passiven Freisetzung zu einem anderen Verlauf der Fahnenachse und zu einer Vergrößerung des Fahnenradius.

Das Einmischen wird durch eine sogenannte Einmischfunktion beschrieben, die in gängigen Modellen nur von Differenzgrößen in Bezug auf die Umgebungsluft abhängt. Hiermit wird das Einmischen aufgrund der fahneninduzierten Turbulenz beschrieben, die im Anfangsstadium der Fahne in der Regel deutlich größer ist als die atmosphärische Turbulenz. Wenn die Fahnenparameter sich denen der Umgebungsluft angenähert haben, wird die weitere Aufweitung der Fahne



vor allem durch die atmosphärische Turbulenz bestimmt, so wie es in einem Ausbreitungsmodell beschrieben wird.

Es gibt daher einen Übergangsbereich zwischen beiden Modellkonzepten, der dynamischen Beschreibung der Fahne als Störung der atmosphärischen Strömung einerseits und der Modellierung einer passiven Spurenstoffwolke im Rahmen einer Ausbreitungsrechnung andererseits. Die beiden Bereiche können durch das Konzept einer Endüberhöhung miteinander verbunden werden. Hierbei wird ein Kriterium für die Annahme festgelegt, daß die Fahne ihre Eigenständigkeit verloren hat und sich in ihren mechanischen und thermodynamischen Eigenschaften nicht mehr von der Umgebungsluft unterscheidet.

Das Konzept der Endüberhöhung erlaubt es, in einer Ausbreitungsrechnung die Wirkung der Fahndynamik durch die Aufprägung einer Überhöhung der mittleren Fahnenachse zu berücksichtigen. Auch die durch die fahneninduzierte Turbulenz bewirkte zusätzliche Aufweitung der Fahne kann berücksichtigt werden, hierauf wird in Standardanwendungen bislang jedoch verzichtet.

Mit dem Fahnenüberhöhungsmodell PLURIS können die für die Festlegung von Überhöhung und zusätzlicher Aufweitung in einem Ausbreitungsmodell (etwa einem Lagrangeschen Partikelmodell nach VDI 3945 Blatt 3) notwendigen Parameter ermittelt werden. Es kann auch direkt für bestimmte Immissionsabschätzungen und – im Fall von Kühlturmschwaden – für Abschätzungen der Sichtbarkeit kondensierter Wasserdampfschwaden eingesetzt werden.

In der Veröffentlichung *A three-dimensional plume rise model for dry and wet plumes* (Janicke, U., Janicke, L., Atmospheric Environment 35, 2001, 877-890) wird das Abgasfahnenmodell PLURIS beschrieben. Das Computerprogramm IBJpluris ist eine Modell-Implementierung von PLURIS, das auch einen meteorologischen Präprozessor zur Erzeugung der Vertikalprofile von Windgeschwindigkeit, Windrichtung und Temperatur enthält (Modul IBJprf).

Das Modell ist in verschiedenen Untersuchungen eingesetzt worden und wurde auch auf unabhängiger Basis nachimplementiert.¹

Abschnitt 2 beschreibt die Eingabeparameter für das Programm. Ergänzend ist in Anhang A die Struktur der verwendeten Ausgabedateien (DMN-Format) dargestellt.

2 Das Programm

Das Modell ist als Computer-Programm in ANSI-C realisiert. Es ist ein reines Text-Programm, läuft also unter UNIX in einem Terminalfenster, unter Windows in einem DOS-Fenster.

Das Programm ist Teil einer Reihe von Programmen zu Problemen der Stoffausbreitung in der Atmosphäre, die alle eine ähnliche Struktur besitzen, teilweise gleiche Module enthalten und das gleiche Datenformat verwenden. Sie unterliegen alle der GNU PUBLIC LICENCE, sind also kostenlos und werden im Quelltext weitergegeben. Die Urheberrechte liegen beim Ingenieurbüro Janicke, Dunum. Das Programm ist sorgfältig und nach bestem Fachwissen erstellt, es werden aber keinerlei Garantien für die Korrektheit der Resultate gegeben. Wer das Programm verwendet, hat sich selbst anhand der Beispiele und des Quelltextes davon zu überzeugen, daß

¹Presotto, L., Bellasio, R., Bianconi, R. (2005): *Assessment of the visibility impact of a plume emitted by a desulphuration plant*. Atmospheric Environment 39, 719-738.



es für seine Zwecke geeignet ist.

2.1 Benutzeranleitung

Das Programm arbeitet nicht interaktiv sondern verwendet eine Befehlsdatei, in der alle Aufgaben spezifiziert sind, die das Programm in einem Rechenlauf durchführen soll. Während der Rechnung schreibt das Programm ein Arbeitsprotokoll in die Protokolldatei und erstellt eventuell weitere Ausgabedateien. Alle diese Dateien haben vorgegebene Namen und befinden sich in einem einzigen Verzeichnis, dem Arbeitsverzeichnis. Zum Aufruf des Programms ist einzugeben:

`IBJpluris Arbeitsverzeichnis Option ...`

Mögliche Optionen sind:

- Dparam=value* : Dem Parameter *param* wird der Wert *value* zugewiesen. Eine in der Befehlsdatei vorhandene Zuweisung wird hiermit überschrieben (Beispiel: `-Dhq=125`).
- e* : Alle in der Befehlsdatei eingelesenen Textzeilen werden in der Protokolldatei protokolliert (*echo*).
- f* : Abbruch (Annahme der Endüberhöhung) nach den Kriterien aus VDI 3945 Blatt 3 (Steigung kleiner 1/200 oder Vertikalgeschwindigkeit kleiner als σ_w). Die vertikalen Geschwindigkeitsfluktuationen der Umgebungsluft werden dabei abgeschätzt als $\sigma_w = 1.3u_*$; die interne Berechnung von u_* (Standardwert 0) erfordert die Vorgabe von `lm`.
- h* : Alle Aufgaben und Eingabeparameter werden am Bildschirm aufgelistet und anschließend das Programm beendet.
- iCommand* : Die Datei mit dem Namen *Command* soll als Befehlsdatei verwendet werden (der Standardname ist `plrcmd.txt`).
- lLog* : Die Datei mit dem Namen *Log* soll als Protokolldatei verwendet werden (der Standardname ist `plrlog.txt`).
- n* : Das Programm wird nicht beendet, wenn die mittlere Fahnenunterkante den Boden berührt.
- p* : Das Programm erzeugt die meteorologischen Vertikalprofile neu. Die eventuell im Arbeitsverzeichnis vorhandene Profildatei `profiles.dnna` wird nicht verwendet.
- q* : Es werden keine Meldungen auf den Bildschirm geschrieben (*quiet*).
- vVerbose* : Mit *Verbose* wird angegeben, wie ausführlich das Programm Meldungen auf dem Bildschirm und in der Protokolldatei ausgibt. Der Standardwert ist 1 (einfache Ausgabe).
- zzmax* : Vorgabe der Maximalhöhe über Grund (m) für die Fahnenachse, die zur Beendigung der Rechnung führt (Standardwert 1100 m). Mit dem Wert 0 m wird dieses Abbruchkriterium ausgeschaltet.



Die Befehlsdatei (Standardname `plrcmd.txt`) ist eine reine Textdatei, kann also vom Benutzer mit einem Text-Editor erstellt werden. Die Protokolldatei (Standardname `plrlog.txt`) ist ebenfalls eine Textdatei, die vom Programm bei jedem Programmlauf angelegt wird. Die Rechenergebnisse werden in eine DMN-Datei (Endung `.dmna`) geschrieben, deren Namen bis auf die Endung identisch mit der Befehlsdatei ist.

Die Befehlsdatei besteht aus mehreren Abschnitten. Jeder Abschnitt definiert eine Aufgabe mit den dazugehörigen Parametern. Eine vollständige Auflistung aller möglicher Aufgaben und der zugeordneten Parameter wird in Abschnitt 2.2 gegeben.

Die erste Zeile eines Abschnittes beginnt mit dem Zeichen „*“, unmittelbar gefolgt von einem Namen, und definiert die Teilaufgabe, die ausgeführt werden soll, beispielsweise

`*Quelle`

Signifikant ist hierbei nur der erste Buchstabe, also das „Q“, wobei auch nicht zwischen Klein- und Großschreibung unterschieden wird. Genauso gut hätte man also auch schreiben können

`*q`

Es folgen in diesem Falle die Parameter, die die Quelle charakterisieren, also z.B.

<code>*Quelle</code>	<code>'</code>	Angaben zur Quelle
<code>hq 100</code>	<code>'</code>	Quellhöhe 100m
<code>dq 5</code>	<code>'</code>	Durchmesser 5m
<code>tq 80</code>	<code>'</code>	Austrittstemperatur 80 Grad Celsius
<code>uq 17</code>	<code>'</code>	Ausströmgeschwindigkeit 17m/s

Bei den Parameternamen sind nur die ersten beiden Zeichen signifikant und es wird ebenfalls nicht zwischen Klein- und Großschreibung unterschieden. Anschließend folgen, durch Leerzeichen und/oder Tabulatoren getrennt, der oder die Werte. Dabei kann es sich um ganze Zahlen, Gleitkommazahlen oder Zeichenketten handeln. Zeichenketten können in Hochkommata eingeschlossen sein. Auch die Zeile, welche die Definition der Aufgabe enthält, kann nach dem Aufgabenamen noch Parameterwerte enthalten.

Ein Apostroph leitet einen Kommentar ein, der beim Einlesen überschlagen wird. Beginnt eine Zeile mit einem Leerzeichen oder einem Minuszeichen, wird die ganze Zeile als Kommentar angesehen.

Die Namen der Parameter sind über alle Abschnitte hinweg eindeutig und dürfen nur in dem Abschnitt gesetzt werden, für den sie vereinbart sind. Sie behalten ihren Wert solange bei, bis er explizit geändert wird.

2.2 Eingabeparameter

Das Programm *IBJpluris* erkennt folgende Aufgaben:

`*Q`

Die Quellparameter werden festgelegt.

`*A`

Die Parameter zur Charakterisierung der Umgebung und Bestimmung der meteorologischen Profile werden festgelegt. Falls die Datei `profiles.dmn` im Arbeitsverzeichnis existiert



und nicht die Aufrufoption `-p` angegeben wird, werden die meteorologischen Profile aus dieser übernommen.

*C

Die Rechenparameter werden festgelegt.

*E

Das Programm wird beendet. Dieser Abschnitt muß angegeben werden.

Die einzelnen Eingabeparameter sind diesen Aufgaben zugeordnet, dürfen also nur in dem betreffenden Abschnitt gesetzt werden. Sie stellen entweder ganze Zahlen, Gleitkommazahlen oder Zeichenketten dar und werden zu Beginn der Rechnung mit einem Standardwert belegt. Sie behalten ihren Wert solange bei, bis er durch Eingabe eines neuen Wertes geändert wird. Ausnahmen hiervon sind explizit aufgeführt. Alle Längen sind in Metern anzugeben, alle Zeiten in Sekunden.

Die einem Parameter zugewiesenen Werte folgen dem Namen, von diesem und untereinander durch Leerzeichen und/oder Tabulatoren getrennt.

In der folgenden Tabelle steht für jeden Parameter in einer Zeile der *Name*, anschließend der *Datentyp* (*integer*, *float* oder *string*), dahinter in Klammern die *Anzahl* der Werte und schließlich die *Standardsetzung*. In den darauf folgenden Zeilen ist die Bedeutung und Handhabung dieses Parameters beschrieben.

*Q Quelle _____

a1 *float*(1) 90

Austrittswinkel gegen die Horizontale (Grad).

a2 *float*(1) 0

Austrittswinkel gegen die *x*-Achse gegen den Uhrzeigersinn (Grad).

cq *float*(1) 1

Anfangskonzentration in der Fahne (ME/m³).

dq *float*(1) 1

Durchmesser der Quelle (m).

fq *float*(1)

Quadrat der Froudezahl beim Quellaustritt. Ist sie vorgegeben, so wird der Parameter tq ignoriert und die Austrittstemperatur berechnet.

hq *float*(1) 10

Bauhöhe der Quelle (m).

id *string*(1)

Kurzinformation zur Rechnung.

iq *float*(1) 0.1

Turbulenzintensität der Fahne beim Quellaustritt.

lq *float*(1) 0

Spezifischer Flüssigwassergehalt beim Quellaustritt (Masse Flüssigwasser pro Masse feuchter Luft).



- qq *float*(1) -1
Wärmestrom in MW (nach TA Luft).
- rq *float*(1) 0
Relative Feuchte beim Quellaustritt (Bruchteil der spezifischen Feuchte bei Sättigung).
- sq *float*(1) 0
Spezifische Feuchte beim Quellaustritt (Masse Wasserdampf pro Masse feuchter Luft).
- tq *float*(1) 20
Austrittstemperatur (Grad Celsius).
- uq *float*(1) 10
Austrittsgeschwindigkeit (m/s).
- xq *float*(1) 0
X-Koordinate des Quellmittelpunktes (m).
- yq *float*(1) 0
Y-Koordinate des Quellmittelpunktes (m).
- *A Umgebung _____
- ca *float*(1) 0
Konzentration in der Umgebungsluft (ME/m³).
- da *float*(1) 270
Windrichtung gegen Nord im Uhrzeigersinn (Grad) in Meßhöhe. Dies ist die in der Meteorologie übliche Definition. Ein Wert von 270 Grad bedeutet Westwind parallel zur *x*-Achse. Die horizontale Austrittsrichtung der Fahne wird relativ zur *x*-Achse gegen den Uhrzeigersinn angegeben!
- ef *float*(1) 1
Faktor, mit dem die Einmischfunktion multipliziert wird.
- ha *float*(1) 10
Anemometerhöhe (m). Wird nur bei *lm* ungleich 0 ausgewertet, siehe Abschnitt 2.3.
- hm *float*(1) 800
Mischungsschichthöhe (m) für das meteorologische Grenzschichtmodell. Wird nur bei *lm* ungleich 0 ausgewertet, siehe Abschnitt 2.3.
- ia *float*(1) 0.10
Turbulenzintensität des Windfeldes in der Umgebung.
- lm *float*(1) 0
Monin-Obukhov-Länge (m) für das meteorologische Grenzschichtmodell (siehe Abschnitt 2.3).
- nz *float*(1) 100
Anzahl der vertikalen Stützpunkte zur Festlegung der Profile.
- pa *float*(1) 101300
Standarddruck am untersten Stützpunkt (Pa). Das Vertikalprofil wird automatisch berechnet.
- ra *float*(1) 0.77
Relative Feuchte (Bruchteil der spezifischen Feuchte bei Sättigung).



- sa *float*(1) 0
Spezifischer Flüssigwassergehalt (Masse Flüssigwasser pro Masse feuchter Luft).
- ta *float*(1) 10
Temperatur in Meßhöhe (C).
- th *float*(1) 2
Meßhöhe der Temperatur (m). Wird nur bei 1m gleich 0 ausgewertet, siehe Abschnitt 2.3.
- tx *float*(1) -0.0098
Temperaturgradient (C/m). Wird nur bei 1m gleich 0 ausgewertet, siehe Abschnitt 2.3.
- ua *float*(1) 3
Windgeschwindigkeit in Meßhöhe (m/s).
- uh *float*(1) 10
Meßhöhe der Windgeschwindigkeit (m). Wird nur bei 1m gleich 0 ausgewertet, siehe Abschnitt 2.3.
- us *float*(1) 0
Schubspannungsgeschwindigkeit (m/s). Wird nur bei 1m ungleich 0 ausgewertet, siehe Abschnitt 2.3. Ist der Wert 0, wird sie aus dem Windprofil berechnet.
- ux *float*(1) 10
Exponent für das Vertikalprofil der Windgeschwindigkeit. Wird nur bei 1m gleich 0 ausgewertet, siehe Abschnitt 2.3.
- ve *integer*(1) 2
Profilversion für das Modul IBJprf. Wird nur bei 1m ungleich 0 ausgewertet, siehe Abschnitt 2.3.
- vs *float*(1) 0
Sedimentationsgeschwindigkeit (m/s). Sedimentation wird als konstantes Absinken der Fahnenachse modelliert.
- z0 *float*(1) 1.0
Rauhigkeitslänge (m). Wird nur bei 1m ungleich 0 ausgewertet, siehe Abschnitt 2.3.
- za *float*(1) 10
Vertikale Schrittweite zur Erzeugung der Profile (m).
- *C Calculation _____
- oo *integer*(1) 2
Ausgabemodus. Er gibt an, welche Fahnenparameter in die Ergebnisdatei ausgeschreiben werden. Folgende Werte sind möglich:
- 1: (einfach) Ausgabe von
 - xx (x-Koordinate, m),
 - yy (y-Koordinate, m),
 - zz (z-Koordinate, m),
 - hh (Höhe über Bauhöhe der Quelle, m),
 - ss (Distanz entlang der Fahnenachse, m),
 - rr (Radius, m),
 - aa (sichtbarer Radius bei kondensiertem Wasserdampf, m),
 - uu (Geschwindigkeit, m/s),



jj (Geschwindigkeitsfluktuationen, m/s),
tt (Temperatur, C),
cc (Konzentration, ME/m³).

2: (ausführlich) Ausgabe von

xx (x -Koordinate, m),
yy (y -Koordinate, m),
zz (z -Koordinate, m),
hh (Höhe über Bauhöhe der Quelle, m),
ss (Distanz entlang der Fahnenachse, m),
ll (Distanz in der horizontalen Projektion, m),
ww (Horizontaler Quellabstand, m),
rr (Radius, m),
uu (Geschwindigkeit, m/s),
jj (Geschwindigkeitsfluktuationen, m/s),
tt (Temperatur, C),
cc (Konzentration, ME/m³),
dd (Dichte, kg/m³),
ee (Einmischfunktion, m²/s),
w1 (Winkel zur Horizontalen, rad),
w2 (Winkel gegen Nord, rad),
aa (sichtbarer Fahnenradius, m),
qi (spezifische Feuchte),
li (spezifischer Flüssigwassergehalt),
zi (spezifischer Gesamtwassergehalt),
sowie weitere Parameter.

3: (Fahnenverlauf) Ausgabe von

xx (x -Koordinate, m),
yy (y -Koordinate, m),
zz (z -Koordinate, m),
hh (Höhe über Bauhöhe der Quelle, m),
ss (Distanz entlang der Fahnenachse, m),
c ww (Horizontaler Quellabstand, m),
rr (Radius, m),
aa (sichtbarer Fahnenradius, m),
w1r (Fahnenunterkante, $ww + rr \sin(w1)$, m),
z1r (Fahnenunterkante, $zz - rr \cos(w1)$, m),
h1r (Fahnenunterkante, $hh - rr \cos(w1)$, m),
w2r (Fahnenoberkante, $ww - rr \sin(w1)$, m),
z2r (Fahnenoberkante, $zz + rr \cos(w1)$, m),
h2r (Fahnenoberkante, $hh + rr \cos(w1)$, m),
w1a (Sichtbare Fahnenunterkante, $ww + rr \sin(w1)$, m),
z1a (Sichtbare Fahnenunterkante, $zz - rr \cos(w1)$, m),
h1a (Sichtbare Fahnenunterkante, $hh - rr \cos(w1)$, m),
w2a (Sichtbare Fahnenoberkante, $ww - rr \sin(w1)$, m),
z2a (Sichtbare Fahnenoberkante, $zz + rr \cos(w1)$, m),



h2a (Sichtbare Fahnenoberkante, $hh + rr \cos(w1)$, m).

4: (skalierter Fahnenverlauf) Ausgabe wie bei 3, nur sind die Werte mit *sc* skaliert.

5: (Vergleich mit Briggs) Ausgabe von
 ll (Distanz in der horizontalen Projektion, m),
 hh (berechnete Überhöhung, m),
 bm (Überhöhung nach Briggs für eine Impulsfahne, m),
 bb (Überhöhung nach Briggs für eine thermische Fahne, m).

sc *float*(1) 1
 Skalierungslänge (m).

sd *float*(1) 0.01
 Mit *sc* skalierte Rechenschrittweite.

se *float*(1) 100
 Mit *sc* skalierte Distanz auf der Fahnenachse, bis zu der gerechnet wird.

sn *integer*(1) 0
 Anzahl der Ausgabeschritte.

2.3 Meteorologische Profile

PLURIS benötigt die Vertikalprofile der Windgeschwindigkeit, Windrichtung und Temperatur der Umgebungsluft. Zur Festlegung gibt es drei Möglichkeiten:

1. Parameter *lm* (Monin-Obukhov-Länge) hat den Wert 0 (Standard): Die Windrichtung ist konstant und hat den Wert *da*. Die Windgeschwindigkeit *u* und Temperatur *T* werden als

$$u(z) = u_a \left(\frac{z}{h_u} \right)^m \quad (1)$$

$$T(z) = T_a + \gamma(z - h_t) \quad (2)$$

mit *u_a* (*ua*), *h_u* (*uh*), *m* (*ux*), *T_a* (*ta*), *γ* (*tx*), *h_t* (*th*) angesetzt. Für *z* > 200 m ist *u* konstant.

2. Parameter *lm* ist ungleich 0: Die Profile werden mit dem Modul *IBJprf* bestimmt. Die Temperatur wird auf Grundlage der Temperaturgradienten nach KTA 1508 festgelegt,² die Windgeschwindigkeit und Windrichtung wird nach LASAT (*ve* 1), VDI 3783 Blatt 8 (*ve* 2, wie LASAT) oder AUSTAL2000 (*ve* 3, wie LASAT, aber mit Winddrehung) festgelegt. Hierbei werden die Parameter *z0*, *d0*, *lm*, *ha*, *ta*, *da* und *ua* ausgewertet, zusätzlich kann *hm* und *us* vorgegeben werden.

²Vor Version 2.7 wurde das Temperaturprofil bestimmt nach: JANICKE, L., JANICKE, U. (2000): Vorschlag eines meteorologischen Grenzschichtmodells für Lagrangesche Ausbreitungsmodelle. *Berichte zur Umweltphysik* Nummer 2, Auflage 1, Ingenieurbüro Janicke, ISSN 1439-8222. Bei extremer Schichtung und hoher Geschwindigkeit führen diese Ansätze jedoch auf unrealistisch große Temperaturgradienten. Daher wird seit Version 2.7 behelfsweise die auf Messungen basierende KTA-Klassierung eingesetzt. Die KTA-Stabilitätsklasse wird aus der vorgegebenen Klug/Manier-Klasse oder den vorgegebenen Werten von *lm* und *z0* nach TA Luft bestimmt.



3. Im Arbeitsverzeichnis existiert die Profildatei `profiles.dmn`: In diesem Fall werden die Profile von Windgeschwindigkeit, Windrichtung, Turbulenzintensität, Temperatur, relativer Feuchte und Umgebungskonzentration aus dieser Datei übernommen, es sei denn, beim Programmaufruf wird die Option `-p` benutzt. Der Datenteil ist eindimensional und die Parameterwerte müssen für die verschiedenen Höhen in oben genannter Reihenfolge als *float*-Zahlen aufgeführt werden. Beispiel:

```
form  "ha%5.1f" "ua%5.1f" "da%5.1f" "ia%5.3f" "ta%5.1f" "ra%5.3f" "ca%12.2e"
mode  "text"
sequ  "i"
dims  1
size  28
lowb  1
hghb  12
*
  0.0   0.0 270.0 1.000 12.0 0.770  0.0
  5.0   2.0 270.0 1.000 12.0 0.770  0.0
 10.0   5.0 270.0 1.000 12.0 0.770  0.0
 20.0   6.5 270.0 1.000 12.0 0.770  0.0
 30.0   8.0 270.0 1.000 12.0 0.770  0.0
 40.0   9.0 270.0 1.000 12.0 0.770  0.0
 60.0  11.0 270.0 1.000 12.0 0.770  0.0
 80.0  12.5 270.0 1.000 12.0 0.770  0.0
100.0  14.0 270.0 1.000 12.0 0.770  0.0
200.0  19.0 270.0 1.000 12.0 0.770  0.0
400.0  26.0 270.0 1.000 12.0 0.770  0.0
600.0  30.0 270.0 1.000 12.0 0.770  0.0
-
```

Die verwendeten Profile werden in den Dateikopf der Ergebnisdatei ausgeschrieben.

A Dateistruktur von DMN-Dateien

Alle Dateien, die Ein- oder Ausgabefelder (Tabellen) repräsentieren, sind nach dem gleichen Prinzip aufgebaut. Sie enthalten zuerst einen Kopf, in dem alle Angaben zur Struktur und Darstellung der Tabelle stehen. Es folgt der Rumpf mit der eigentlichen Tabelle. Dieser Rumpf kann unmittelbar an den Kopf angehängt sein, wenn die Tabelle formatiert ausgeschrieben wird. Bei unformatierter Darstellung bildet der Rumpf immer eine eigene Datei. In dieser Binärdatei stehen die Tabellenelemente so, wie sie intern dargestellt sind, unmittelbar hintereinander ohne jegliche Steuerzeichen. Ein Tabellenelement kann eine einzelne Zahl (Datenelement) oder ein Verbund von mehreren Zahlen (*record*) sein.

Der Kopf ist eine Textdatei mit der Namenserverweiterung `dmn`, die aus einzelnen Zeilen besteht. In jeder Zeile ist ein Parameter definiert. Der Name des Parameters steht zu Beginn der Zeile, gefolgt von einem oder mehreren Werten. Zulässige Trennungszeichen sind Leerzeichen, Tabulator und Semikolon, die einzeln oder kombiniert verwendet werden können. Die Zeile kann durch LF oder CR+LF abgeschlossen sein.

Neben den vom Programm benötigten Parametern kann der Kopf auch weitere Parameter enthalten. Parameter, die das Programm nicht kennt, werden ignoriert. Der Kopf endet mit einer



Zeile, die zu Anfang einen Stern, also *, enthält. Mit der nächsten Zeile beginnt der Rumpf, sofern er mit dem Kopf zusammen eine einzige Datei bildet. Die (formatierten) Tabellenelemente im Rumpf werden durch Leerzeichen, Semikolon, Tabulator, CR oder LF getrennt. Die Tabelle wird durch eine Zeile, die mit drei Sternen beginnt, beendet.

Folgende Parameter im Kopf der Datei werden vom Programm erkannt und interpretiert (die Namen müssen klein geschrieben sein):

data *string*(1)

Name der Datei, die die eigentliche Tabelle enthält. Ist **data** nicht spezifiziert oder hat es den Wert „*“, dann wird bei formatierter Ausgabe die Tabelle in die gleiche Datei geschrieben wie der Kopf. Bei unformatierter Ausgabe wird der Dateiname des Kopfes übernommen, er erhält aber statt „.dmna“ die Endung „.dmnb“. Falls in **data** eine Pfadangabe gemacht ist, gilt diese relativ zu dem Ordner, in dem der Kopf gespeichert ist.

dims *integer*(1)

Anzahl der Dimensionen (maximal 5).

fact *float*(1)

Faktor, mit dem bei formatierter Ausgabe alle Datenelemente vom Typ *float* oder *double* multipliziert werden, bevor sie im angegebenen Format ausgeschrieben werden. Bei der Eingabe formatierter Daten werden diese Datenelemente nach dem Einlesen durch **fact** dividiert. Der Faktor wirkt nur auf die Datenelemente, für die in der Formatangabe (siehe **form**) kein eigener Faktor angegeben ist.

form *string*(1)

Format, nach welchem bei formatierter Speicherung die Daten abgelegt sind. Bestehen die Tabellenelemente des gespeicherten Feldes aus mehreren Datenelementen, dann ist für jedes Datenelement eine Formatangabe erforderlich, und alle Einzelformate verkettet ergeben die Zeichenkette **form**.

Format = *Format*₁*Format*₂...

*Format*_i = *Name*%(**Factor*)*Length*.*PrecisionSpecifier*

Es bedeuten:

Name Name des Datenelementes (optional).

Factor Skalierungsfaktor (optional einschl. Klammern).

Length Länge des Datenfeldes.

Precision Anzahl der Nachkommastellen (bei *float*-Zahlen).

Specifier Umwandlungsangabe.

Der Skalierungsfaktor *Factor* wird genauso gehandhabt wie der Parameter **fact**. Die Längenangabe *Length* ist die Mindestlänge des Datenfeldes. Sie kann überschritten werden, wenn dies zur korrekten Darstellung der Zahl erforderlich ist. Zwischen den Zahlen steht immer mindestens ein Trennungszeichen.

Folgende Umwandlungsangaben sind möglich:



Spec.	Typ	Länge	Beschreibung
c	<i>character</i>	1	einzelne Buchstaben
d	<i>integer</i>	4	Dezimalzahl
x	<i>integer</i>	4	Hexadezimalzahl
f	<i>float</i>	4	Festkommazahl (ohne Exponent)
e	<i>float</i>	4	Gleitkommazahl (mit Exponent)
t	<i>integer</i>	4	Zeitangabe (ohne Datum)

Den Angaben f und e kann ein 1 vorangestellt sein (*double* mit Länge 8 Bytes), den Angaben d und x ein h (*short integer* mit der Länge 2 Bytes).

Zeitdarstellung bei Binärausgabe: Bei Zeitangabe ohne Datum bezeichnet die Zahl die vergangenen Sekunden. Ist der Angabe t ein 1 vorangestellt, wird die Zahl (*double* mit Länge 8 Bytes) als Zeitangabe mit Datum interpretiert: Die Vorkommastellen bezeichnen die Anzahl der Tage seit 1899-12-30.00:00:00 plus 10^6 , die Nachkommastellen den Anteil der vergangenen Sekunden an diesem Tag. Zeitdarstellung bei Textausgabe: Bei der Angabe t hat die Zeitangabe die Form dd.hh:mm:ss oder hh:mm:ss, bei der Angabe 1t die Form yyyy-mm-dd.hh:mm:ss.

Gleichartige Formatangaben können zusammengefaßt werden:
vx%5.2fvy%5.2fvz%5.2f ist äquivalent zu vx%[3]5.2f³

hghb *integer*(dims)

Höchster Indexwert für die verschiedenen Laufindizes.

lowb *integer*(dims)

Niedrigster Indexwert für die verschiedenen Laufindizes.

mode *string*(1)

Bei binary sind die Daten unformatiert gespeichert, sonst formatiert.

sequ *string*(1)

Angabe, in welcher Indexfolge die Daten gespeichert sind. Normalerweise läuft der am weitesten rechts stehende Index am schnellsten (C-Konvention). Dies entspricht bei einem 3-dim. Feld A_{ijk} der Angabe i+, j+, k+. FORTRAN speichert gemäß k+, j+, i+. Ein Minuszeichen statt des Pluszeichens bedeutet, daß der betreffende Index rückwärts läuft. Es können auch Teilbereiche ausgewählt werden:

j=10..1/1, i=5..25/1, k=1. Die Angabe /n bedeutet, daß der betreffende Index des Ausschnittes mit dem Wert n anfängt. Wird mit sequ ein Ausschnitt des Datenfeldes definiert, dann beziehen sich die Indexgrenzen lowb und hghb auf die ursprünglichen Indexdefinitionen.

size *integer*(1)

Länge der einzelnen Daten (*record size*) in Bytes. Bei formatierter Speicherung muß die aus der Formatangabe resultierende Summe der Längen der einzelnen Datenelemente gleich size sein.

Zeichenketten müssen, wenn sie Leerzeichen enthalten, in Doppelhochkomma eingeschlossen sein, ansonsten sind diese optional.

³Bei zusammengefaßten Formatangaben gilt der angegebene Name nur für das erste Element. Bei den folgenden Elementen wird der letzte Buchstabe in Name jeweils um eine Stelle im Alphabet weitergerückt, so daß schließlich der angegebene Ausdruck entsteht.

**Beispiel:**

Ein Feld von Gleitkommazahlen $A_{ijk} = 100i + 10j + k, i = 1..3, j = 2..4, k = 0..1$ wird in horizontalen Schichten gespeichert:

```
form "%4.1f"
mode "text"
sequ "k+,j-,i+"
fact 1.000e-001
dims 3
size 4
lowb 1 2 0
hghb 3 4 1
*
  14.0  24.0  34.0
  13.0  23.0  33.0
  12.0  22.0  32.0

  14.1  24.1  34.1
  13.1  23.1  33.1
  12.1  22.1  32.1
```
