



# tubsgbox – Box-Definitionen für *tubsglateg*\*

Enrico Jörns  
e dot joerns at tu minus bs dot de

30. November 2020

## Zusammenfassung

Diese Datei stellt die Umgebung `gaussbox` für Darstellungen im Gaußraster und die Umgebungen `modulebox` für Darstellungen im Modulraster zur Verfügung.

## Inhaltsverzeichnis

<b>1</b>	<b>Benutzung</b>	<b>1</b>
1.1	Gaußraster . . . . .	1
1.2	Modulsystem . . . . .	1
<b>2</b>	<b>Implementierung</b>	<b>2</b>
2.1	Paketoptionen . . . . .	2
2.2	Paket-Optionen . . . . .	3
2.3	Gemeinsame Box-Optionen . . . . .	4
2.4	Gaussbox . . . . .	5
2.4.1	Optionen . . . . .	5
2.4.2	Box . . . . .	7
2.5	Modul-System . . . . .	10
2.5.1	Optionen . . . . .	11
2.5.2	Modulzeile ( <code>modrow</code> ) . . . . .	16
2.5.3	Modulspalte ( <code>modcol</code> ) . . . . .	17
2.5.4	Modulunterzeile ( <code>modsubrow</code> ) . . . . .	18
2.5.5	Box . . . . .	19

---

\*This document corresponds to `tubsgbox` ?, dated ?.

# 1 Benutzung

## 1.1 Gaußraster

Das Gaußraster wird zur Inhalts-Darstellung auf Postern und Titelseiten verwendet.

`gaussbox` Box im Gauß-Raster  
Syntax: `\gaussbox[<options>]{<>}`

## 1.2 Modulsystem

Das Modulsystem wird ausschließlich für wissenschaftliche Plakate verwendet.

# 2 Implementierung

```
1 <*>package
```

Es werden die folgenden Pakete geladen: `ifthen` und `xkeyval` werden allgemein zum Definieren und Auswerten von Optionen etc. benötigt. `etoolbox` und `forloop` werden für die Auswertung von kommasetrennten Listen benutzt. Aus `tubstypearea` werden alle benötigten Längen entnommen, um die Boxen mit `texpos` korrekt positionieren zu können.

```
2 \RequirePackage{ifthen}
3 \RequirePackage{xkeyval}
4 \RequirePackage{etoolbox}
5 \RequirePackage{forloop}
6 \RequirePackage{tubsdebug}
7 \RequirePackage{tubstypearea}
8 \RequirePackage[absolute,overlay]{textpos}
```

## 2.1 Paketoptionen

Alle hier beschriebenen Optionen sind sowohl für die `gaussbox` als auch für die `modulebox` verwendbar.

`tubsbox@bottomsender` Merker für Option 'sender'.  

```
9 \newboolean{tubsbox@bottomsender}\setboolean{tubsbox@bottomsender}{false}
```

`\tubsbox@framebox` Speichert ggf. Kommando mit dem ein Rahmen um die jeweilige Box gezeichnet werden soll.  

```
10 \def\tubsbox@framebox{\relax}
```

`\tubsbox@colorbox` Speichert ggf. `colorbox`-Kommando mit dem die Hintergrundfarbe der jeweiligen Box gezeichnet werden soll.

```
11 \newcommand{\tubsbox@colorbox}{\relax}
```

`\if@tubs@oddpag` `{\langle odd cmd \rangle}{\langle even cmd \rangle}`

Switch um Kommandos zu definieren, die für gerade und ungerade Seiten unterschiedlich arbeiten. Ist nur aktiv, wenn Option `twoside` benutzt wird.

```
12 \newcommand{\if@tubs@oddpag}[2]{%
13   \ifthenelse{\boolean{tubsstyle@twoside}}{%
14     \Ifthispageodd{#1}{#2}%
15   }{#1}%
16 }
```

`\modulesetup` `[\langle option \rangle]`

Setzt keys für die `modulebox`. Die keys werden dabei an die Familien `tubsbox.sty`, `tubsbox` und `modulebox` delegiert. TODO: Genaue Verwendung noch unklar.

```
17 \newcommand*{\modulesetup}[1][{}]{%
18   \setkeys*{tubsbox.sty}{#1}%
19   \setrmkeys*{tubsbox}%
20   \setrmkeys{modulebox}%
21 }
```

`\gausspagesetup` `[\langle option \rangle]`

Setzt Paketoptionen der familie `tubsbox.sty`

```
22 \newcommand*{\gausspagesetup}[1][{}]{%
23   \setkeys*{tubsbox.sty}{#1}%
24 }
```

## 2.2 Paket-Optionen

Dummy-Option, die aktuell nur aus Designschwäche existiert.

```
25 \define@key{tubsbox.sty}{bgcolor}[{}]{}
```

‘`sender`’ Mögliche Werte: `bottom` oder `top`.

Schaltet zwischen den beiden Layoutvarianten mit Absenderbereich oben und unten auf der Seite um, damit die Boxen korrekt platziert werden.

Standardmäßig ist der Absenderbereich oben.

```
26 \define@choicekey{tubsbox.sty}{sender}[\var\nr]{top,bottom}{%
27   \ifcase\nr\relax
28     \setboolean{tubsbox@bottomsender}{false}
```

```

29 \or
30 \setboolean{tubsbox@bottomsender}{true}
31 \fi
32 }

```

‘nosep’ Kein Abstand zwischen Boxen (erzeugt \if@gb@nosep)

```

33 \define@boolkey{tubsbox.sty}[@gb@]{nosep}[true]{}

```

Paket-Optionen auswerten, Initialisierung mit Werten der Paket-Optionen.

```

34 \ProcessOptionsX*\relax%

```

## 2.3 Gemeinsame Box-Optionen

Optionen der Familie tubsbox sind sowohl für das Gaussraster (gaussbox) als auch für das Modulsystem (modulebox) verfügbar

‘bgcolor’ Standard-Hintergrundfarbe der Boxen festlegen.  
Mögliche Werte sind none, sowie die jeweilige gewünschte Farbe.  
Standardmäßig haben die Boxen keine Hintergrundfarbe.TODO?

```

35 \define@key{tubsbox}{bgcolor}{%
36 \ifthenelse{equal{#1}{none}}{%
37 \renewcommand\tubsbox@colorbox{\relax}
38 }{%
39 \def\bgcolor{#1}
40 \renewcommand\tubsbox@colorbox{\colorbox{\bgcolor}}
41 }
42 }

```

‘fgcolor’ Legt Vordergrund- bzw. Schriftfarbe der Box fest

```

43 \define@key{tubsbox}{fgcolor}{
44 \ifthenelse{equal{#1}{none}}{%
45 }{%
46 \color{#1}%
47 }
48 }

```

Hierin werden vfill-Kommandos gespeichert, um die vertikale Positionierung des Box-Inhaltes über Optionen steuern zu können.

```

49 \newcommand\tb@topfill{\relax}
50 \newcommand\tb@bottomfill{\relax}

```

‘t’ Inhalt in Boxen oben ausrichten.

```
51 \define@key{tubsbox}{t}[true]{%
52   \renewcommand\tb@topfill{\relax}
53   \renewcommand\tb@bottomfill{\relax}
54 }
```

‘c’ Inhalt in Boxen vertikal zentrieren.

```
55 \define@key{tubsbox}{c}[true]{%
56   \renewcommand\tb@topfill{\vfil}
57   \renewcommand\tb@bottomfill{\vfil}
58 }
```

‘b’ Inhalt in Boxen unten ausrichten.

```
59 \define@key{tubsbox}{b}[true]{%
60   \renewcommand\tb@topfill{\vfill}
61   \renewcommand\tb@bottomfill{\relax}
62 }
```

‘frame’ Zeichnet wenn gewünscht einen Rahmen um die jeweiligen Boxen.  
Mögliche Werte: none - Kein Rahmen, fbox - einfacher Rahmen.  
Standardmäßig haben die Boxen keinen Rahmen.

```
63 \define@choicekey{tubsbox}{frame}[\val\nr]{none,fbox}{%
64   \ifcase\nr\relax
65     \def\tubsbox@framebox{\relax}
66   \or
67     \def\tubsbox@framebox{\fbox}
68   \fi
69 }
```

## 2.4 Gaussbox

‘nosep’ Kein Abstand zwischen Boxen

```
70 % \define@boolkey{gaussbox}{@gb@}{nosep}[true]{}
```

Erlaubt zusätzlichen Abstand zum oberen Boxrand. Dies wird unter anderem für den 3fachen Abstand in der ersten Box verwendet.

```
71 \newcommand*\tubs@gb@addtoppadding{0mm}
```

## 2.4.1 Optionen

‘logosep’ Fügt zusätzlichen Top-Innenabstand hinzu. Wenn kein Wert angegeben, wird doppelte Rahmenstärke hinzugefügt. (Dies ist die Standardeinstellung für die erste tubsbox auf einer Seite mit Siegelbandlogo.)

```
72 \define@boolkey{gaussbox}{logosep}[true]{%
73   \def\tubs@gb@toppadding@outer{3\tubsborderwidth}
74 }
```

‘outerpadding’ Setzt äußeres Padding der gaussbox

```
75 \define@choicekey{gaussbox}{outerpadding}[\val\nr]{%
76   default,none,vnone,hnone}{%
77   \ifcase\nr\relax% default
78   \or% kein Außen-Padding
79     \def\tubs@gb@toppadding@outer{0mm}%
80     \def\tubs@gb@bottompadding@outer{0mm}%
81     \def\tubs@gb@leftpadding@outer{0mm}%
82     \def\tubs@gb@rightpadding@outer{0mm}%
83   \or% keine Abstände zum Blattrand
84     \def\tubs@gb@toppadding@outer{0mm}%
85     \def\tubs@gb@bottompadding@outer{0mm}%
86   \or% Innenabstände auf columnsep
87     \def\tubs@gb@leftpadding@outer{0mm}%
88     \def\tubs@gb@rightpadding@outer{0mm}%
89   \fi
90 }
```

‘innerpadding’ Setzt inneres Padding der gaussbox

```
91 \define@choicekey{gaussbox}{innerpadding}[\val\nr]{%
92   default,columnsep,none,vnone,hnone}{%
93   \ifcase\nr\relax% default
94   \or% Innenabstände auf columnsep
95     \def\tubs@gb@toppadding@inner{0.5\tubscolumnsep}%
96     \def\tubs@gb@bottompadding@inner{0.5\tubscolumnsep}%
97   \or% Alle Innenabstände auf 0
98     \def\tubs@gb@toppadding@inner{0mm}%
99     \def\tubs@gb@bottompadding@inner{0mm}%
100    \def\tubs@gb@leftpadding@inner{0mm}%
101    \def\tubs@gb@rightpadding@inner{0mm}%
102   \or% Alle vertikalen Innenabstände auf 0
103     \def\tubs@gb@toppadding@inner{0mm}%
104     \def\tubs@gb@bottompadding@inner{0mm}%
105   \or% Alle horizontalen Innenabstände auf 0
106     \def\tubs@gb@leftpadding@inner{0mm}%
107     \def\tubs@gb@rightpadding@inner{0mm}%
108   \fi
109 }
```

‘padding’ Setzt Padding der gaussbox

```
110 \define@choicekey{gaussbox}{padding}[\val\nr]{%
111     default,minimal,none}{%
112     \ifcase\nr\relax% default
113     \or% Abstände normal minimal
114     \setkeys{gaussbox}{outerpadding=none}%
115     \setkeys{gaussbox}{innerpadding=columnsep}%
116     \or% Alle Abstände auf 0
117     \setkeys{gaussbox}{outerpadding=none}%
118     \setkeys{gaussbox}{innerpadding=none}%
119     \fi
120 }
```

## 2.4.2 Box

\tb@gb@setorig Berechnet den Ursprung des von gaussbox benutzten Koordinatensystems (neu). Der Ursprung wird dabei horiz. auf den Beginn des Textbereichs gesetzt, um mit \TPHorizModule das Spaltenlayout realisieren zu können.

```
121 \providecommand\tb@gb@setorig{%
122     \ifthenelse{\boolean{tubsbox@bottomsender}}{%
123         \if@tubs@oddpag{%
124             \textblockorigin{\tubsborderwidth+\tubspage@bcor}{%
125                 \tubsborderwidth}%
126             }{%
127                 \textblockorigin{\tubsborderwidth}{%
128                     \tubsborderwidth}%
129             }%
130         }{%
131             \if@tubs@oddpag{%
132                 \textblockorigin{\tubsborderwidth+\tubspage@bcor}{%
133                     \tubssenderheight}%
134             }{%
135                 \textblockorigin{\tubsborderwidth}{%
136                     \tubssenderheight}%
137             }
138         }
139 }
```

Lege Rastermaße fest. \TPHorizModule entspricht der Breite des Kommunikationsbereichs, abzüglich der Breite des Rahmens. TODO: needs fix for bcor??

```
140 \setlength{\TPHorizModule}{%
141     (\paperwidth-2\tubsborderwidth-\tubspage@bcor)*\ratio{1mm}{\tubspage@xsegments mm}}
142 \setlength{\TPVertModule}{%
143     (\tubscommunicationheight)*\ratio{1mm}{\value{tubspage@gausssum} mm}}
```

`\tubs@gb@toppadding` Innenabstände der Box, d.h. Abstand zwischen Boxrand und Textbereich  
`\tubs@gb@toppadding`  
`\tubs@gb@toppadding` 144 `\newlength{\tubs@gb@toppadding}`  
`\tubs@gb@toppadding` 145 `\newlength{\tubs@gb@bottompadding}`  
`\tubs@gb@toppadding` 146 `\newlength{\tubs@gb@leftpadding}`  
147 `\newlength{\tubs@gb@rightpadding}`

`\tubs@gb@storebox` Savebox zum Speichern der Box-Inhalte.  
148 `\newsavebox{\tubs@gb@storebox}`

`tubs@gb@lastelement` Definition einiger benötigter Counter.  
`tubs@gb@calcypos`  
`tubs@gb@calcheight` 149 `\newcounter{tubs@gb@lastelement}`  
150 `\newcounter{tubs@gb@calcypos}`  
151 `\newcounter{tubs@gb@calcheight}`

`gaussbox` [*options*]{*xpos*}{*ypos*}{*width*}{*height*}

**options** Siehe Abschnitt 2.1  
**xpos** Horizontaler Startpunkt der Box, gemessen im Spalten-Raster, Standard-Wertebereich: [1-6]  
**ypos** Vertikaler Startpunkt der Box, gemessen im Gauß-Raster Standard-Wertebereich: [1-6] (Querformat), [1-8] (Hochformat)  
**width** Breite der Box in Spalten, Wertebereich: [1-6]  
**height** Höhre der Box in Gaußraster-Elementen, Wertebereich: [1-6] / [1-8]

```

152 \newenvironment{gaussbox}[5][{}]{%
153 % Innere Padding-Werte (durch Optionen anpassbar)
154 \def\tubs@gb@toppadding@inner{\tubsborderwidth}%
155 \def\tubs@gb@bottompadding@inner{\tubsborderwidth}%
156 \def\tubs@gb@leftpadding@inner{0.5\tubscolumnsep}%
157 \def\tubs@gb@rightpadding@inner{0.5\tubscolumnsep}%
158 % Äußere Padding-Werte (durch Optionen anpassbar)
159 \def\tubs@gb@toppadding@outer{\tubsborderwidth}%
160 \def\tubs@gb@bottompadding@outer{\tubsborderwidth}%
161 \def\tubs@gb@leftpadding@outer{\tubsborderwidth}%
162 \def\tubs@gb@rightpadding@outer{\tubsborderwidth}%
163 %
164 \setkeys*{tubsbox}{#1}%
165 \setrmkeys{gaussbox}%
166 \tb@gb@setorig

Berechnung der linken und rechten Ränder

167 % Berechne Positionen
168 \ifnum#2=1% Segment beginnt bei erster column?
169 \setlength\tubs@gb@leftpadding{\tubs@gb@leftpadding@outer}%
170 \else
171 \setlength\tubs@gb@leftpadding{\tubs@gb@leftpadding@inner}%

```



```

172 \fi
173 %
174 \setcounter{tubs@gb@lastelement}{#2-1}%
175 \edef\tubs@gb@xpos{\thetubs@gb@lastelement}%TODO: etwas unschön...
176 \addtocounter{tubs@gb@lastelement}{#4}%
177 \ifnum\value{tubs@gb@lastelement}=\tubspage@xsegments% Segment endet bei letzter column?
178   \setlength{tubs@gb@rightpadding}{tubs@gb@rightpadding@outer}%
179 \else
180   \setlength{tubs@gb@rightpadding}{tubs@gb@rightpadding@inner}%
181 \fi

```

### Berechnung der oberen und unteren Ränder

```

182 \ifnum#3=1%
183   \setlength{tubs@gb@toppadding}{tubs@gb@toppadding@outer}% Erstes Segment?
184 %   \addtolength{tubs@gb@toppadding}{tubs@gb@addtoppadding}
185 \else
186   \setlength{tubs@gb@toppadding}{tubs@gb@toppadding@inner}%
187 \fi
188 %
189 \setcounter{tubs@gb@lastelement}{#3+#5-1}%
190 % Prüfe, ob Werte korrekt
191 \ifnum\value{tubs@gb@lastelement}>\tubspage@ysegments%
192   \PackageError{tubstypearea}{Invalid segment number}{}%
193 \fi
194 %
195 \ifnum\value{tubs@gb@lastelement}=\tubspage@ysegments% Letztes Segment?
196   \setlength{tubs@gb@bottompadding}{tubs@gb@bottompadding@outer}%
197 \else
198   \setlength{tubs@gb@bottompadding}{tubs@gb@bottompadding@inner}%
199 \fi
200 %
201 % Makro |\@inv@arg| wird benutzt, um Argument 'inverted' zu übergeben
202 \def\@inv@arg{\relax}%
203 \ifthenelse{\boolean{tubsbox@bottomsender}}{%
204   \def\@inv@arg{inverted}%
205 }{}%
206 \calc@gauss@elementpos[\@inv@arg]{tubs@gb@calcypos}{#3}%
207 \def\tubs@gb@ypos{\thetubs@gb@calcypos}%
208 %
209 \calc@gauss@elementpos[\@inv@arg]{tubs@gb@calcheight}{#3+#5}%
210 \addtocounter{tubs@gb@calcheight}{-\thetubs@gb@calcypos}%
211 %
212 \def\tubs@gb@width{#4}%
213 \def\tubs@gb@height{\thetubs@gb@calcheight}%
214 %TODO: Option to set margins of boxes to 0
215 \begin{lrbox}{tubs@gb@storebox}%
216 % Store parskip and parindent to restore them in minipage
217 \newdimen\tubs@restore@parskip
218 \newdimen\tubs@restore@parindent

```

```

219 \tubs@restore@parskip\parskip
220 \tubs@restore@parindent\parindent
221 \begin{minipage}%
222     [t]%
223     [\tubs@gb@height\TPVertModule-\tubs@gb@toppadding-\tubs@gb@bottompadding]%
224     {\tubs@gb@width\TPHorizModule-\tubs@gb@leftpadding-\tubs@gb@rightpadding}% boxwidth min
225 % restore par-lengths, avoid initial parskip
226 \parskip\tubs@restore@parskip%
227 \parindent\tubs@restore@parindent%
228 \vspace*{-\parskip}%
229 % Standard-Version von |\includegraphics| mit imagefit-Version ersetzen, wenn keine Argumente
230 \let\oldincludegraphics\includegraphics
231 \renewcommand\includegraphics[2][\relax]{%
232     \ifx##1\relax
233         \tubs@sb@calc@autoscale{##2}{\tubs@gb@width\TPHorizModule-\tubs@gb@leftpadding-\tubs@gb@r
234         \expandafter\oldincludegraphics\expandafter[\@img@scale@param]{##2}%
235     \else
236         \oldincludegraphics[##1]{##2}%
237     \fi
238 }%
239 \tb@topfill
240 \noindent\ignorespaces
241 }{%
242 \ignorespacesafterend
243 \tb@bottomfill
244 \par%
245 \end{minipage}%
246 \end{lrbox}%
247 \setlength{\fboxsep}{0cm}% needed here??
248 \begin{textblock}{\tubs@gb@width}(\tubs@gb@xpos,\tubs@gb@ypos)%
249 \vspace*{-\parskip}%
250 \vspace*{\tubs@gb@toppadding}%
251 \newdimen\temp@
252 \newdimen\temp@b
253 \setlength\temp@{\tubs@gb@height\TPVertModule}%
254 \setlength\temp@b{\tubs@gb@width\TPHorizModule}%
255 \noindent
256 \fboxsep0mm%
257 \tubsbox@framebox{\tubsbox@colorbox{% inner paddings are set *only* here!
258     \hbox to \temp@b{%
259         \hspace*{\tubs@gb@leftpadding}%
260         \usebox{\tubs@gb@storebox}%
261         \hspace*{\tubs@gb@rightpadding}%
262     }%
263 }}%
264 \end{textblock}%
265 }

```

## 2.5 Modul-System

`\tb@sb@setorig` Berechnet den Ursprung des von modulebox benutzten Koordinatensystems (neu).

```

266 \providecommand\tb@sb@setorig{%
267   \ifthenelse{\boolean{tubsbox@bottomsender}}{%
268     \if@tubs@oddpag{%
269       \textblockorigin{\tubsborderwidth+\tubspage@bcor}{%
270         \tubsborderwidth}%
271     }{%
272       \textblockorigin{\tubsborderwidth}{%
273         \tubsborderwidth}%
274     }%
275   }{%
276     \if@tubs@oddpag{%
277       \textblockorigin{\tubsborderwidth+\tubspage@bcor}{%
278         0.1429\paperheight}%
279     }{%
280       \textblockorigin{\tubsborderwidth}{%
281         0.1429\paperheight}% TODO: Kompatibilität mit Querformat
282     }
283   }
284 }

285 \newsavebox{\tubs@sb@storebox}

286 \newsavebox{\tubs@sb@bgimagebox}

```

`tubs@sb@elementcount` Counter zählt Anzahl Zeilen für Box-Layout.

```

287 \newcounter{tubs@sb@elementcount}

```

`tubs@sb@xcount` Counter zählt Anzahl Xen

```

288 \newcounter{tubs@sb@xcount}

```

`\tubs@sb@xfreespace` Länge speichert aktuellen freien Platz für Xe

```

289 \newlength{\tubs@sb@xfreespace}

```

`\tubs@sb@xlength` Länge eines X-Elements

```

290 \newlength{\tubs@sb@xlength}

```

`tubs@cnt` Counter für forloop-Schleife

```

291 \newcounter{tubs@cnt}

```

Boolean für Hintergrundbild

```

292 \newboolean{modulebox@showbgimage}
293 \setboolean{modulebox@showbgimage}{false}
294 \newcommand{\@bgimage@to@insert}{dummy}

```

## 2.5.1 Optionen

‘bgimage’ Setze Hintergrundbild.

```

295 \define@key{modulebox}{bgimage}[none]{%
296   \ifthenelse{\equal{#1}{none}}{%
297     \setboolean{modulebox@showbgimage}{false}%
298   }{%
299     \setkeys{tubsbox}{bgcolor=none}%TODO?
300     \renewcommand{\@bgimage@to@insert}{#1}%
301     \setboolean{modulebox@showbgimage}{true}%
302   }
303 }
```

‘imagefit’ Einstellung der Bildeinpassung in den Darstellungsbereich.

scaled	Bild wird horiz. und vert. passend skaliert
autoclip	Bild wird entweder horiz. skaliert und vert. beschnitten oder vert. skaliert und horiz. beschnitten. Jeweils so, dass der Darstellungsbereich komplett ausgefüllt wird
clipx	Bild wird vert. skaliert und horiz. beschnitten
clipy	Bild wird horiz. skaliert und vert. beschnitten

NOTE: crop-Options are only kept for compatibility reasons

TODO: rename internal cmds

TODO: add vclip, hclip

```

304 \define@choicekey{tubsbox}{imagefit}[\val\nr]{%
305   default,scaled,autoclip,cropped,hclip,cropx,vclip,cropy,fitheight,fitwidth}{%
306   \ifcase\nr\relax% default
307     \PackageInfo{tubsstyle}{%
308       Option 'imagefit' not set. Using standard value 'cropped'.
309     }
310     \renewcommand{\tubs@sb@imagefit}{cropped}
311   \or% scaled
312     \renewcommand{\tubs@sb@imagefit}{scaled}
313   \or% autoclip
314     \renewcommand{\tubs@sb@imagefit}{cropped}
315   \or% cropped
316     \renewcommand{\tubs@sb@imagefit}{cropped}
317   \or% clipx
318     \renewcommand{\tubs@sb@imagefit}{cropx}
319   \or% cropx
320     \renewcommand{\tubs@sb@imagefit}{cropx}
321   \or% clipy
322     \renewcommand{\tubs@sb@imagefit}{cropy}
323   \or% cropy
324     \renewcommand{\tubs@sb@imagefit}{cropy}
325   \or% fitheight
```

```

326 \renewcommand{\tubs@sb@imagefit}{cropx}
327 \or% fitwidth
328 \renewcommand{\tubs@sb@imagefit}{cropy}
329 \fi
330 }

```

```

331 \newcommand\tubs@sb@imagefit{cropped}
332 \newcommand\@img@scale@param{}%

```

\tubs@sb@calc@autoscale Setzt \@img@scale@param als Parameter für \includegraphics

{\imagefile}{\viewwidth}{\viewheight} TODO: generalize with other packages?

```

333 \newcommand\tubs@sb@calc@autoscale[3]{%
334 % Temporäre Längen für die Berechnung
335 \newdimen\@sb@image@xorig}
336 \newdimen\@sb@image@xcalc}
337 \newdimen\@sb@image@yorig}
338 \newdimen\@sb@image@ycalc}
339 % store dimension arguments (width, height)
340 \newdimen\@view@width%
341 \newdimen\@view@height%
342 \setlength\@view@width{#2}%
343 \setlength\@view@height{#3}%
344 \ifthenelse{\equal{\tubs@sb@imagefit}{scaled}}{%
345 \renewcommand\@img@scale@param{clip,%
346 height=\@view@height,
347 width=\@view@width}
348 }{\ifthenelse{\equal{\tubs@sb@imagefit}{cropped}}{%
349 % Ermittelt, ob das Bild an den Seiten oder oben und unten beschnitten
350 % werden muss, um in den Darstellungsbereich zu passen
351 % Dazu wird die Höhe des auf korrekte Breite skalierten Bildes
352 % mit der Höhe des Darstellungsbereichs verglichen und entsprechend
353 % eine crop-Option gesetzt.
354 \settoheight{\@sb@image@ycalc}{%
355 \includegraphics[clip,width=\@view@width]{#1}}
356 \ifthenelse{\lengthtest{\@sb@image@ycalc>\@view@height}}{%
357 \renewcommand{\tubs@sb@imagefit}{cropy}
358 }{%
359 \renewcommand{\tubs@sb@imagefit}{cropx}
360 }
361 }}
362 \ifthenelse{\equal{\tubs@sb@imagefit}{cropy}}{%
363 % Berechne abzuschneidende Ränder (oben+unten)
364 % Dazu wird die Differenz zwischen Darstellungsbereich und Höhe des
365 % korrekt auf die Breite skalierten Bildes berechnet und mit dem
366 % ermittelten Skalierungsfaktor multipliziert, sowie durch 2 geteilt.
367 % Das Ergebnis wird dann einmal am oberen und einmal am unteren Teil
368 % des (Original-)Bildes mit Hilfe der 'trim'-Option abgeschnitten.
369 \settoheight{\@sb@image@yorig}{%

```

```

370     \includegraphics[clip]{#1}}%
371     \settoheight{\@sb@image@ycalc}{%
372     \includegraphics[clip,width=\@view@width]{#1}}%
373     \setlength{\@sb@image@ycalc}{(\@sb@image@ycalc-\@view@height)*\ratio{\@sb@image@yorig}{\@
374     \setlength{\@sb@image@ycalc}{0.5\@sb@image@ycalc}%
375     \renewcommand\@img@scale@param{clip,%
376     width=\@view@width,
377     trim=0pt {\@sb@image@ycalc} 0pt {\@sb@image@ycalc}}%
378     {\ifthenelse{\equal{\tubs@sb@imagefit}{cropx}}}%
379     \settowidth{\@sb@image@xorig}{%
380     \includegraphics[clip]{#1}}%
381     \settowidth{\@sb@image@xcalc}{%
382     \includegraphics[clip,height=\@view@height]{#1}}%
383     \setlength{\@sb@image@xcalc}{(\@sb@image@xcalc-(\@view@width))*\ratio{\@sb@image@xorig}{\@
384     \setlength{\@sb@image@xcalc}{0.5\@sb@image@xcalc}%
385     \renewcommand\@img@scale@param{clip,%
386     height=\@view@height,
387     trim={\@sb@image@xcalc} 0pt {\@sb@image@xcalc} 0pt}%
388     {\ifthenelse{\equal{\tubs@sb@imagefit}{keepsizes}}}%
389     \renewcommand\@img@scale@param{}%
390     }}}}
391 }%
392 }

```

`\tubs@sb@setelements`  $\{ \langle list \rangle \}$

Erwartet als Parameter eine komma-getrennte Liste, deren Elemente entweder Längen oder der Buchstabe X ist. Elemente mit Buchstaben X teilen den restlichen zur Verfügung stehenden Platz gleichmäßig untereinander auf.

```

393 \newcommand\tubs@sb@setelements[2]{%
394   \setcounter{tubs@sb@elementcount}{0}
395   \setcounter{tubs@sb@xcount}{0}
396   % Iteriere über Liste
397   \let\do\tubs@sb@parsenext
398   \docsvlist{#1}
399   % X-Länge setzen
400   \setlength\tubs@sb@xlength{\tubs@sb@xfreespace/\thetubs@sb@xcount}
401   % Iteriere über alle Zeilen und ersetze X-Zeilen durch die entsprechende Länge
402   \stepcounter{tubs@sb@elementcount}% only for forloop!
403   \forloop[1]{tubs@cnt}{1}{\value{tubs@cnt}<\thetubs@sb@elementcount}{%
404     \expandafter\def\expandafter\tubs@sb@current@row\expandafter{%
405       \csname tubs@sb@tmplength@\thetubs@cnt\endcsname%
406     }
407     % Schreibe nun gespeicherte Längen und X-Längen in endgültiges Makro
408     \ifthenelse{\equal{\tubs@sb@current@row}{X}}{%
409       \writetubslog{replace X with \the\tubs@sb@xlength}%
410       \expandafter\edef\csname #2\thetubs@cnt\endcsname{%
411         \the\tubs@sb@xlength%
412       }%

```

```

413     }{%
414         \expandafter\edef\csname #2\thetubs@cnt\endcsname{%
415             \tubs@sb@current@row%
416         }%
417     }
418 }
419 \addtocounter{tubs@sb@elementcount}{-1}% only for forloop!
420 }

```

`\tubs@sb@setrows`  $\{ \langle element \rangle \}$

Prüft, ob das übergebene Element ein X ist, ansonsten wird davon ausgegangen, dass es sich um eine Länge handelt und es wird versucht diese zu speichern. TODO: Optionen 'XX', 'XXX', um Verhältnisse wiederzugeben

$X : XX : XXX = 1 : 2 : 3$

```

421 \newcommand*{\tubs@sb@parsenext}[1]{%
422     \stepcounter{tubs@sb@elementcount}
423     \ifthenelse{\equal{#1}{X}}{% X
424         \stepcounter{tubs@sb@xcount}%
425         \expandafter\def\csname tubs@sb@tmplength@\thetubs@sb@elementcount\endcsname{X}
426     }{% Laenge
427         % Ziehe Länge von xfreespace ab und speichere sie.
428         \addtolength\tubs@sb@xfreespace{-#1}
429         \expandafter\def\csname
430             tubs@sb@tmplength@\thetubs@sb@elementcount\endcsname{#1}%
431     }
432     % Ziehe zusätzlich Stegbreite ab.
433     \addtolength{\tubs@sb@xfreespace}{-0.5\tubsborderwidth}
434 }

```

`modcontainer`  $[ \langle options \rangle ] \{ \langle rows \rangle \}$

Extra-Umgebung für wissenschaftliche Poster mit Modulsystem. Als Argument ist die gewünschte Anzahl an Modul-Zeilen anzugeben. Dies geschieht mit einer durch Kommas getrennten Liste. Jedes Element muss dabei entweder eine Längenangabe oder der Buchstabe 'X' sein. Jedes Element steht für eine Modul-Zeile und legt gleichzeitig deren Höhe fest. Der Buchstabe 'X' sorgt dabei dafür, dass der restliche zur Verfügung stehende Platz gleichmäßig auf X-Elemente aufgeteilt wird. Die Verwendung von mindestens einem 'X' pro Layout ist sinnvoll, da sonst der Darstellungsraum mühsam von Hand auf die richtige Größe gebracht werden muss.

Jede angelegte Modul-Zeile muss dann mit der Umgebung `modrow` mit Inhalt gefüllt werden.

Wertet Argument-Liste aus und speichert ermittelte Längen in Makros mit Präfix `tubs@sb@rowlength`.

```

435 \newenvironment{modcontainer}[2][ ]{%
436     \writetubslog{modcontainer: #1, #2}%

```

```

437 % \modulesetup[bgcolor=tuWhite,#1]%
438 \setkeys{tubsbox.sty}{#1}%
439 \setkeys{tubsbox}{t,bgcolor=tuWhite,fgcolor=tuBlack}{}% TODO: check!
440 % reset modrow counter
441 \setcounter{tubs@sb@current@row}{0}%
442 \setlength{tubs@sb@xfreespace}{\textheight+1.5\tubsborderwidth}%
443 \writetubslog{page: xfree: \the\tubs@sb@xfreespace}%
444 \tubs@sb@setelements{#2}{tubs@sb@rowlength}%
445 }{%
446 }

```

tubs@sb@current@col    Zähler für die jeweils aktuell gesetzte Modul-Zeile/-Spalte.

tubs@sb@current@row

tubs@sb@current@subrow

```

447 \newcounter{tubs@sb@current@col} % aktuelle Spalte
448 \newcounter{tubs@sb@current@row} % aktuelle Zeile
449 \newcounter{tubs@sb@current@subrow} % aktuelle Subspalte

450 \newboolean{tubs@sb@borderless}\setboolean{tubs@sb@borderless}{false}

```

## 2.5.2 Modulzeile (modrow)

@modrow    Implementierung von modrow.

```

451 \newenvironment{@modrow}[1][X]{%
452   % setze keys
453   \setkeys*{tubsbox}{#1}%
454   \setrmkeys*{modulebox}%
455   \edef\@remaining@keys{\XKV@rm}%
456   % Ersetze mit X, falls keine Länge angegeben
457   \ifthenelse{\equal{\@remaining@keys}{} }{%
458     \edef\@remaining@keys{X}}{}%
459   % reset modcol counter
460   \setcounter{tubs@sb@current@col}{0}%
461   % Setze Länge auf Textbreite + linker und rechter Rand + Stegkorrektur
462   \setlength{tubs@sb@xfreespace}{\textwidth+1.5\tubsborderwidth}%
463   % Parse Komma-Liste und speichere Längen mit Präfix 'tubs@sb@collength@'
464   \expandafter\tubs@sb@setelements\expandafter{\@remaining@keys}{tubs@sb@collength@}%
465   \stepcounter{tubs@sb@current@row}%
466   % Wenn nur eine Spalte, dann direkt modulebox setzen
467   \ifthenelse{\expandafter\equal\expandafter{\@remaining@keys}{X}}{%
468     \stepcounter{tubs@sb@current@col}%
469     \expandafter\edef\expandafter\@argI\expandafter{%
470       \csname tubs@sb@rowlength@\thetubs@sb@current@row\endcsname}%
471     \expandafter\edef\expandafter\@argII\expandafter{%
472       \csname tubs@sb@collength@\thetubs@sb@current@col\endcsname}%
473     \def\tubs@sb@box@cmd{\@modulebox{\@argI}{\@argII}}%
474     \def\tubs@sb@box@endcmd{\end@modulebox}%
475   }{%
476     \def\tubs@sb@box@cmd{\relax}%

```



```

477 \def\tubs@sb@box@endcmd{\relax}%
478 }
479 \writetubslog{modrow: xfree: \the\tubs@sb@xfreespace}%
480 \tubs@sb@box@cmd%
481 }{%
482 \tubs@sb@box@endcmd%
483 }

```

`modrow` [*<cols>*]

`modrow*` Umgebung mit der Modulzeilen mit Inhalt gefüllt oder zusätzliche Spaltenbereiche angelegt werden können. Das optionale Argument *rows* hat dabei dieselbe Funktion wie in `modcontainer` beschrieben, allerdings definiert sie nun *Spaltenbreiten*.

Wenn nur eine einzelne (X)-Spalte definiert wurde, kann der Inhalt direkt in die Umgebung geschrieben werden, ansonsten müssen die einzelnen Spalteninhalte mittels der Umgebung `modcol` definiert werden.

Wertet Argument-Liste aus und speichert ermittelte Längen in Makros mit Präfix `tubs@sb@collength@`. Wenn nur eine einzelne (X)-Spalte definiert wurde, bzw. das optionale Argument leer gelassen, so wird direkt eine `modulebox` erstellt.

```

484 \newcommand{\@@modrow}{%
485 \ifnextchar[{%
486 }{%
487 }%
488 \@modrow%
489 }
490 %
491 \newenvironment{modrow}{%
492 \setboolean{tubs@sb@borderless}{false}%
493 \@@modrow%
494 }{%
495 \end@modrow%
496 }

497 \newenvironment{modrow*}{%
498 \setboolean{tubs@sb@borderless}{true}%
499 \@@modrow%
500 }{%
501 \end@modrow%
502 }

```

### 2.5.3 Modulspalte (`modcol`)

`@modcol` Implementierung von `modcol`.

```

503 \newenvironment{@modcol}[1][ ]{%
504 % setze keys

```

```

505 % \modulebox@setdefault@opts
506 \setkeys*{tubsbox}{#1}
507 \setrmkeys*{modulebox}
508 \edef\@remaining@keys{\XKV@rm}%
509 % Ersetze mit X, falls keine Länge angegeben
510 \ifthenelse{\equal{\@remaining@keys}{}}{%
511 \edef\@remaining@keys{X}}{%
512 % reset modsubrow counter
513 \setcounter{tubs@sb@current@subrow}{0}%
514 % Setze Länge auf Zeilenhöhe + Trennstrich
515 \expandafter\setlength\expandafter\tubs@sb@xfreespace\expandafter{\csname tubs@sb@rowlength@
516 % Parse Komma-Liste und speichere Längen mit Präfix 'tubs@sb@subrowlength@'
517 \expandafter\tubs@sb@setelements\expandafter{\@remaining@keys}{tubs@sb@subrowlength@}%
518 \stepcounter{tubs@sb@current@col}%
519 % Wenn nur eine Spalte, dann direkt modulebox setzen
520 \ifthenelse{\expandafter\equal\expandafter{\@remaining@keys}{X}}{%
521 \stepcounter{tubs@sb@current@subrow}%
522 \expandafter\edef\expandafter\@argI\expandafter{%
523 \csname tubs@sb@subrowlength@\thetubs@sb@current@subrow\endcsname}%
524 \expandafter\edef\expandafter\@argII\expandafter{%
525 \csname tubs@sb@collength@\thetubs@sb@current@col\endcsname}%
526 \def\tubs@sb@box@cmd{\@modulebox{\@argI}{\@argII}}%
527 \def\tubs@sb@box@endcmd{\end@modulebox}%
528 }{%
529 \def\tubs@sb@box@cmd{\relax}%
530 \def\tubs@sb@box@endcmd{\relax}%
531 }
532 \tubs@sb@box@cmd%
533 }{%
534 \tubs@sb@box@endcmd%
535 }

```

`modcol` [*subrows,settings*]

`modcol*` Umgebung mit der Modulspalten mit Inhalt gefüllt oder zusätzliche Unterzeilen angelegt werden können.

Die Sternchen-Version sorgt dafür, dass der innere Rand weggelassen wird, wie dies zum Beispiel für das Einfügen von Grafiken sinnvoll ist.

```

536 \newenvironment{modcol}{%
537 \setboolean{tubs@sb@borderless}{false}%
538 \@modcol%
539 }{%
540 \end@modcol%
541 }

542 \newenvironment{modcol*}{%
543 \setboolean{tubs@sb@borderless}{true}%
544 \@modcol%

```

```

545 }{%
546   \end@modcol%
547 }

```

#### 2.5.4 Modulunterzeile (modsubrow)

modsubrow Umgebung mit der Modul-Unterzeilen mit Inhalt gefüllt werden können.  
modsubrow\*

```

548 \newenvironment{modsubrow}{%
549   \setboolean{tubs@sb@borderless}{false}%
550   \@modsubrow%
551 }{%
552   \end@modsubrow%
553 }

```

```

554 \newenvironment{modsubrow*}{%
555   \setboolean{tubs@sb@borderless}{true}%
556   \@modsubrow%
557 }{%
558   \end@modsubrow%
559 }

```

@modsubrow Implementierung von modsubrow.

```

560 \newenvironment{@modsubrow}[1][ ]{%
561   % setze keys
562   \setkeys*{tubsbox}{#1}%
563   \setrmkeys{modulebox}%
564   \stepcounter{tubs@sb@current@subrow}%
565   \expandafter\edef\expandafter\@oarg\expandafter{%
566     \csname tubs@sb@rowlength@\thetubs@sb@current@row\endcsname}%
567   \expandafter\edef\expandafter\@argI\expandafter{%
568     \csname tubs@sb@subrowlength@\thetubs@sb@current@subrow\endcsname}%
569   \expandafter\edef\expandafter\@argII\expandafter{%
570     \csname tubs@sb@collength@\thetubs@sb@current@col\endcsname}%
571   \@modulebox[][\@oarg]{\@argI}{\@argII}%
572 }{%
573   \end@modulebox%
574 }

```

#### 2.5.5 Box

Benutzer-Längen für Höhe und Breite des aktuell verwendeten Moduls

```

575 \newlength{\modulewidth}
576 \newlength{\moduleheight}
577 \newdimen\@box@tmp% Generelle temp-Var.

```

`\tubs@sb@box@xpos` Von @modulebox verwendete Längen.  
`\tubs@sb@box@ypos`  
`\tubs@sb@box@rowheight` 578 `\newlength{\tubs@sb@box@xpos}`% x-Position  
`\tubs@sb@box@splitrow` 579 `\newlength{\tubs@sb@box@ypos}`% y-Position  
580 `\newlength{\tubs@sb@box@rowheight}`% Höhe einer Zeile, sofern als opt. Arg. gegeben  
581 `\newlength{\tubs@sb@box@splitrow}`% Speichert benutze Länge von Splitrow für späteren Reset  
582 `\newlength{\tubs@sb@padding}`% Abstand Text zu Boxrand

@modulebox [`\rowheight`][`\height`]{`\width`}

Box für Inhalt auf wissenschaftlichen Postern (Modulsystem). Dies ist unabhängig vom Gaußraster und hat schmalere Ränder als die gaussbox. Das optionale Argument `rowheight` muss gesetzt werden, wenn in einer Zeile mehrere Boxen untergebracht werden sollen.

```

583 \newenvironment{@@modulebox}[3][\relax]{%
584 % Argumente sichern
585 \ifx#1\relax%
586 \def\@current@rowheight{#2}%
587 \else%
588 \def\@current@rowheight{#1}%
589 \fi%
590 \def\@current@height{(#2)}%
591 \def\@current@width{(#3)}% Breite der weißen Box, ohne Ränder!
592 \ifthenelse{\boolean{tubs@sb@borderless}}{%
593 \setlength\tubs@sb@padding{0mm}%
594 }{%
595 \setlength\tubs@sb@padding{2\tubspage@module@padding}%
596 }%
597 \newdimen\@temp@width
598 \newdimen\@temp@height
599 \setlength\@temp@height{#2}%
600 \@temp@width#3%
601 \tb@sb@setorig%
602 \newdimen\tubs@restore@parskip%
603 \newdimen\tubs@restore@parindent%
604 \tubs@restore@parskip\parskip%
605 \tubs@restore@parindent\parindent%
606 % Speichere Inhalt in minipage-Box.
607 \begin{lrbox}{\tubs@sb@storebox}%
608 \begin{minipage}[t][\@current@height-\tubs@sb@padding]{\@current@width-\tubs@sb@padding}%
609 \setlength{\moduleheight}{\@current@height}%
610 \setlength{\modulewidth}{\@current@width}%
611 \parskip\tubs@restore@parskip%
612 \parindent\tubs@restore@parindent%
613 \tb@topfill
614 \noindent\ignorespaces
615 }{%
616 \ignorespacesafterend
617 \tb@bottomfill

```

```

618 \end{minipage}%
619 \end{lrbox}%
620 \ifthenelse{\boolean{modulebox@showbgimage}}{%
621 \tubs@sb@calc@autoscale{\@bgimage@to@insert}%
622 {\@current@width}%
623 {\@current@height}%
624 \begin{lrbox}{\tubs@sb@bgimagebox}%
625 \expandafter\includegraphics\expandafter[\@img@scale@param]{%
626 \@bgimage@to@insert}%
627 \end{lrbox}%
628 }{}%
629 % Setze Box mit Hintergrundbild/-farbe
630 \begin{textblock*}{\@current@width}(\tubs@sb@box@xpos,\tubs@sb@box@ypos)%
631 \fboxsep0mm%
632 \tubsbox@framebox{\tubsbox@colorbox{%
633 \vbox to \@temp@height{\usebox{\tubs@sb@bgimagebox}}}%
634 }}%
635 \end{textblock*}%
636 % Definiere und platziere Textblock mit gespeichertem Inhalt.
637 \begin{textblock*}{\@current@width}(\tubs@sb@box@xpos,\tubs@sb@box@ypos)%
638 \fboxsep0mm%
639 \vbox to \@temp@height{%
640 \vspace*{0.5\tubs@sb@padding}%
641 \hbox to \@temp@width{%
642 \hspace*{0.5\tubs@sb@padding}%
643 \usebox{\tubs@sb@storebox}%
644 \hspace*{0.5\tubs@sb@padding}%
645 }%
646 \vspace*{0.5\tubs@sb@padding}%
647 }%
648 \end{textblock*}%
649 \writetubslog{modulebox @(\the\tubs@sb@box@xpos,\the\tubs@sb@box@ypos)}%
650 \if@tubs@dbg% DEBUGGING output!!
651 \begin{textblock*}{\@current@width}(\tubs@sb@box@xpos,\tubs@sb@box@ypos)%
652 \colorbox{tuWhite}{\the\tubs@sb@box@xpos,\the\tubs@sb@box@ypos}%
653 \end{textblock*}%
654 \fi%
655 % Prüfe, ob Zeile gesplittet werden soll (opt. Arg. != \relax)
656 \ifthenelse{\equal{\@current@rowheight}{\relax}}{%
657 \addtolength\tubs@sb@box@xpos{\@current@width+0.5\tubsborderwidth}%
658 }{}%
659 \setlength\tubs@sb@box@rowheight{\@current@rowheight}
660 \addtolength\tubs@sb@box@ypos{\@current@height+0.5\tubsborderwidth}%
661 \addtolength\tubs@sb@box@splitrow{\@current@height+0.5\tubsborderwidth}%
662 % Wenn Zeilenhöhe ausgefüllt, normal weiter machen...
663 \ifthenelse{\lengthtest{\tubs@sb@box@splitrow > \tubs@sb@box@rowheight}}{%
664 \addtolength\tubs@sb@box@xpos{\@current@width+0.5\tubsborderwidth}%
665 \addtolength\tubs@sb@box@ypos{-\tubs@sb@box@splitrow}%
666 \setlength\tubs@sb@box@splitrow{0mm}%
667 }{}

```

```

668 }
669 % Wenn Zeile gefüllt ist, springen in nächste Zeile
670 \setlength\@box@tmp{\textwidth+\tubsborderwidth}% Texthöhe + Rahmenhöhe
671 \writetubslog{width: is \the\tubs@sb@box@xpos > \the\@box@tmp ?}%
672 \ifthenelse{\lengthtest{\tubs@sb@box@xpos > \@box@tmp}}{%
673   \addtolength\tubs@sb@box@ypos{\@current@rowheight+0.5\tubsborderwidth}%
674   \setlength\tubs@sb@box@xpos{0mm}%
675   % Wenn darüber hinaus Seite gefüllt ist, fange neue an!
676   \setlength\@box@tmp{\textheight+\tubsborderwidth}% Texthöhe + Rahmenhöhe
677   \writetubslog{height: is \the\tubs@sb@box@ypos > \the\@box@tmp ?}%
678   \ifthenelse{\lengthtest{\tubs@sb@box@ypos > \@box@tmp}}{%
679     \setlength\tubs@sb@box@ypos{0mm}%
680   }{}
681 }{}
682 % speichere global!
683 \global\tubs@sb@box@xpos=\tubs@sb@box@xpos%
684 \global\tubs@sb@box@ypos=\tubs@sb@box@ypos%
685 \global\tubs@sb@box@splitrow=\tubs@sb@box@splitrow%
686 }
687 % Umgebungsende
688 \let\end@modulebox\end@@modulebox
689 %
690 \newcommand\modulebox@setdefault@opts{%
691   \setkeys{tubsbox}{bgcolor=tuWhite,fgcolor=tuRed}%TODO
692 }

```

#### Behandlung von erstem optionalen Argument (keyvals)

```

693 \def\@modulebox@oparam[#1]{%
694   \setkeys*{modulebox}{#1}%
695   \setkeys*{gaussbox}{#1}%
696   \@@modulebox%
697 }

```

#### Test nach erstem optionalen Argument (keyvals)

```

698 \def\@modulebox{%
699   \@ifnextchar[%
700     \@modulebox@oparam%
701   ]{%
702     \@@modulebox%
703   }%
704 }

```

```

705 </package>

```