

The `physicx` package

Wenjian Chern (Longaster*)

November 13, 2021, version v0.2

Abstract

`physicx`

1 Implementation

```
1 <*package>
2 <@@=physicx>
3 \cs_generate_variant:Nn \keys_set:nn { nx , on , ox }
4 \cs_generate_variant:Nn \use:n { nnnn { nnno }
5 \cs_generate_variant:Nn \seq_set_split:Nnn { Non, NVV, c, cnV, cVV }
6 \cs_generate_variant:Nn \tl_replace_all:Nnn { Non, Nox }
7 \cs_new:Npn \PHYSICXIGNORE
8 { \exp_stop_f: \exp_not:N \PHYSICXIGNORE }
9 \bool_new:N \g__physicx_mathtools_bool
10 \bool_new:N \g__physicx_physics_bool
11 \bool_new:N \g__physicx_compat_bool
12 \bool_new:N \g__physicx_short_bool
13 \bool_new:N \g__physicx_reqty_bool
14 \prg_new_conditional:Npnn \physicx_compat: { T, F, TF }
15 {
16   \bool_if:NTF \g__physicx_compat_bool
17   { \prg_return_true: } { \prg_return_false: }
18 }
19 \prg_new_conditional:Npnn \physicx_short: { T, F, TF }
20 {
21   \bool_if:NTF \g__physicx_short_bool
22   { \prg_return_true: } { \prg_return_false: }
23 }
24 \prg_new_conditional:Npnn \physicx_mathtools: { T, F, TF }
25 {
26   \bool_if:NTF \g__physicx_mathtools_bool
27   { \prg_return_true: } { \prg_return_false: }
28 }
29 \prg_new_conditional:Npnn \physicx_option_or:nn #1#2 { T, F, TF }
30 {
31   \bool_lazy_or:nnTF
32   { \cs:w g__physicx_ #1 _bool \cs_end: }
33   { \cs:w g__physicx_ #2 _bool \cs_end: }
```

*Email: longaster@163.com

```

34     { \prg_return_true: }
35     { \prg_return_false: }
36 }
37
38 \bool_new:N \l__physicx_tmpa_bool
39 \int_new:N \l__physicx_tmpa_int
40 \int_new:N \l__physicx_tmpb_int
41 \msg_new:nnnn { physicx } { unknown-key }
42 { The~key~'#1'~is~unknown~and~is~being~ignored. }
43 {
44     The~module~#2~does~not~have~a~key~called~#1.\\
45     Check~that~you~have~spelled~the~key~name~correctly.
46 }
47 \msg_new:nnn { physicx } { diag-key }
48 { The~value~'#1'~of~diag~key~is~unknown~and~is~being~ignored. }

```

1.1 Utils functions

```

\physicx_parse_range:nnnN Parse range, such as -3,6-8,9,10-.
\physicx_parse_range_check:
    \physicx_parse_range_nocheck:
49 \int_new:N \l__physicx_begin_int
50 \int_new:N \l__physicx_end_int
51 \int_new:N \l__physicx_max_int
52 \int_new:N \l__physicx_min_int
53 \bool_new:N \l__physicx_invalid_range_bool
54 \cs_new_protected:Npn \physicx_parse_range_check:
55 {
56     \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_check:n
57     \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_check:
58 }
59 \cs_new_protected:Npn \physicx_parse_range_nocheck:
60 {
61     \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_nocheck:n
62     \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_nocheck:
63 }
64 \cs_new_protected:Npn \physicx_parse_range:nnnN #1#2#3#4
65 {
66     \seq_set_eq:NN #4 \c_empty_seq
67     \int_set:Nn \l__physicx_min_int {#1}
68     \int_set:Nn \l__physicx_max_int {#2}
69     \clist_map_inline:nn {#3}
70     {
71         \__physicx_parse_range_aux:n {##1}
72         \bool_if:NF \l__physicx_invalid_range_bool
73         { \seq_concat:NNN #4 #4 \l__physicx_tmpa_seq }
74     }
75 }
76 \cs_generate_variant:Nn \physicx_parse_range:nnnN { nnvN, nnxN }
77 \cs_new_protected:Npn \physicx_parse_range:nnN
78 { \physicx_parse_range:nnnN { 1 } }
79 \cs_generate_variant:Nn \physicx_parse_range:nnN { nvN, nxN }
80 \cs_new_protected:Npn \__physicx_parse_range_aux:n #1
81 {
82     \bool_set_false:N \l__physicx_invalid_range_bool
83     \seq_clear:N \l__physicx_tmpa_seq

```

```

84 \tl_if_in:nnTF {#1} { - }
85 {
86   \seq_set_split:Nnn \l__physicx_tmpb_seq { - } {#1}
87   \seq_pop_left:NN \l__physicx_tmpb_seq \l__physicx_tmpa_tl
88   \tl_if_empty:NTF \l__physicx_tmpa_tl
89   { \int_set_eq:NN \l__physicx_begin_int \l__physicx_min_int }
90   {
91     \int_set:Nn \l__physicx_begin_int { \l__physicx_tmpa_tl }
92     \int_compare:nNnT \l__physicx_begin_int < \l__physicx_min_int
93     {
94       \int_set_eq:NN \l__physicx_begin_int \l__physicx_min_int
95     }
96   }
97   \seq_pop_left:NN \l__physicx_tmpb_seq \l__physicx_tmpa_tl
98   \tl_if_empty:NTF \l__physicx_tmpa_tl
99   { \int_set_eq:NN \l__physicx_end_int \l__physicx_max_int }
100   {
101     \int_set:Nn \l__physicx_end_int { \l__physicx_tmpa_tl }
102     \int_compare:nNnT \l__physicx_end_int > \l__physicx_max_int
103     {
104       \int_set_eq:NN \l__physicx_end_int \l__physicx_max_int
105     }
106   }
107   \__physicx_parse_range_range:
108 }
109 { \__physicx_parse_range_single:n {#1} }
110 }
111 \cs_new:Npn \__physicx_parse_range_single_check:n #1
112 {
113   \bool_lazy_or:nnTF
114   { \int_compare_p:nNn {#1} > \l__physicx_max_int }
115   { \int_compare_p:nNn {#1} < \l__physicx_min_int }
116   { \bool_set_true:N \l__physicx_invalid_range_bool }
117   { \seq_put_right:Nn \l__physicx_tmpa_seq {#1} }
118 }
119 \cs_new:Npn \__physicx_parse_range_single_noccheck:n #1
120 { \seq_put_right:Nn \l__physicx_tmpa_seq {#1} }
121 \cs_new_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_check:n
122 \cs_new:Npn \__physicx_parse_range_range_check:
123 {
124   \bool_lazy_or:nnTF
125   { \int_compare_p:nNn \l__physicx_begin_int > \l__physicx_max_int }
126   { \int_compare_p:nNn \l__physicx_begin_int > \l__physicx_end_int }
127   { \bool_set_true:N \l__physicx_invalid_range_bool }
128   {
129     \int_step_inline:nnn
130     { \l__physicx_begin_int } { \l__physicx_end_int }
131     { \seq_put_right:Nn \l__physicx_tmpa_seq {##1} }
132   }
133 }
134 \cs_new:Npn \__physicx_parse_range_range_noccheck:
135 {
136   \int_compare:nNnTF \l__physicx_begin_int > \l__physicx_end_int
137   { \bool_set_true:N \l__physicx_invalid_range_bool }

```

```

138     {
139         \int_step_inline:nnn
140         { \l__physicx_begin_int } { \l__physicx_end_int }
141         { \seq_put_right:Nn \l__physicx_tmpa_seq {##1} }
142     }
143 }
144 \cs_new_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_check:

(End definition for \physicx_parse_range:nnnN, \physicx_parse_range_check:, and \physicx_parse_
range_nocheck:. These functions are documented on page ??.)

145 \cs_new:Npn \__physicx_if_keyval:nTF #1
146 { \tl_if_in:nnTF {#1} { = } }
147 \prg_new_conditional:Npnn \physicx_if_num:n #1 { T, F, TF }
148 {
149     \regex_match:nnTF { \A [[:digit:]]+ \Z } {#1}
150     { \prg_return_true: } { \prg_return_false: }
151 }
152 \prg_new_conditional:Npnn \physicx_if_num_sign:n #1 { T, F, TF }
153 {
154     \regex_match:nnTF { \A [\+|-]* [[:digit:]]+ \Z } {#1}
155     { \prg_return_true: } { \prg_return_false: }
156 }
157 \cs_new:Npn \physicx_search_also:nn #1#2
158 {
159     \clist_map_inline:nn {#1}
160     {
161         \exp_args:Nno \keys_if_exist:nnT {##1} { \l_keys_key_str }
162         {
163             \clist_map_break:n
164             { \keys_set:no {##1} { \l_keys_key_str = {#2} } }
165         }
166     }
167 }
168 \prg_new_conditional:Npnn \physicx_search_also:nn #1#2 { T, F, TF }
169 {
170     \bool_set_false:N \l__physicx_tmpa_bool
171     \clist_map_inline:nn {#1}
172     {
173         \exp_args:Nno \keys_if_exist:nnT {##1} { \l_keys_key_str }
174         {
175             \clist_map_break:n
176             {
177                 \bool_set_true:N \l__physicx_tmpa_bool
178                 \keys_set:no {##1} { \l_keys_key_str = {#2} }
179             }
180         }
181     }
182     \bool_if:NTF \l__physicx_tmpa_bool
183     { \prg_return_true: } { \prg_return_false: }
184 }
185 \cs_generate_variant:Nn \physicx_search_also:nn { no , oo }
186 \prg_generate_conditional_variant:Nnn \physicx_search_also:nn { no , oo } { T , F , TF }
187 \cs_new_protected:Npn \physicx_new_type:nnn #1#2#3
188 { \keys_define:nn { physicx/#1 } { type / #2 .meta:n = {#3} } }

```

```

189 \tl_const:Nn \c_physicsx_order_tl { \mathcal{o} }
190 \tl_const:Nn \c_physicsx_Order_tl { \mathcal{O} }
191 \cs_new:Npn \physicsx_use_amssymb_type:
192 {
193   \cs_set_eq:NN \physicsx_bf: \boldsymbol
194 }
195 \cs_new:Npn \physicsx_use_uni_bfit_type:
196 {
197   \cs_set_eq:NN \physicsx_bf: \symbfit
198 }
199 \cs_new:Npn \physicsx_use_uni_bf_type:
200 {
201   \cs_set_eq:NN \physicsx_bf: \symbf
202 }
203 \cs_new:Npn \physicsx_left: { \mathopen{}\mathclose\bgroup\left }
204 \cs_new:Npn \physicsx_right: { \aftergroup\egroup\right }
205 \cs_new:Npn \physicsx_left:N { \mathopen{}\mathclose\bgroup }
206 \cs_new:Npn \physicsx_right:N { \egroup }
207 \keys_define:nn { physicsx }
208 {
209   compat .bool_set:N = \g__physicsx_compat_bool ,
210   compat .default:n = true ,
211   short .bool_set:N = \g__physicsx_short_bool ,
212   short .default:n = true ,
213   physics .code:n = \RequirePackage{physics} ,
214   mathtools .code:n = \RequirePackage{mathtools} ,
215   unimath .code:n = \RequirePackage{unicode-math} ,
216   reqty .bool_set:N = \g__physicsx_reqty_bool ,
217   reqty .default:n = true ,
218   reqty .initial:n = true ,
219   noqty .meta:n = { reqty = false } ,
220 }
221 %
222 \ProcessKeysPackageOptions { physicsx }
223 %
224 \@ifpackageloaded{physics}
225 { \bool_set_true:N \g__physicsx_compat_bool }
226 { }
227 \@ifpackageloaded{mathtools}
228 { \bool_set_true:N \g__physicsx_mathtools_bool }
229 { \bool_set_false:N \g__physicsx_mathtools_bool }
230 %
231 \physicsx_compat:T
232 {
233   \tl_set_eq:NN \ordersymbol \c_physicsx_order_tl
234   \tl_set_eq:NN \Ordersymbol \c_physicsx_Order_tl
235 }
236 %
237 \@ifpackageloaded {unicode-math}
238 { \physicsx_use_uni_bfit_type: }
239 { \physicsx_use_amssymb_type: }

```

`\physicsxset` physics setup command.

```

240 \NewDocumentCommand \physicsxset { s m }

```

```

241 {
242   \IfBooleanTF {#1}
243   { \keys_set:nn { physicx/#2 } }
244   { \keys_set:nn { physicx } {#2} }
245 }

```

(End definition for `\physicxset`. This function is documented on page ??.)

1.2 Quantity things

1.2.1 New quantity interfaces

```

246 \keys_define:nn { physicx }
247 { quantity .code:n = \keys_set:nn { physicx/quantity } {#1} }
248 \keys_define:nn { physicx/quantity }
249 {
250   pre .tl_set:N = \l__physicx_quantity_pre_tl ,
251   post .tl_set:N = \l__physicx_quantity_post_tl ,
252   left .tl_set:N = \l__physicx_quantity_left_tl ,
253   right .tl_set:N = \l__physicx_quantity_right_tl ,
254   left-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_left_size_tl #1 } ,
255   right-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_right_size_tl #1 } ,
256   size .meta:n = { left-size = {#1} , right-size = {#1} } ,
257   noauto .meta:n = { left-size = \c_empty_tl , right-size = \c_empty_tl } ,
258   noauto .value_required:n = false ,
259   args .code:n =
260     \tl_set:Nn \l__physicx_quantity_args_tl { [#1] } ,
261   args* .tl_set:N = \l__physicx_quantity_args_tl ,
262   code .tl_set:N = \l__physicx_quantity_code_tl ,
263   type .multichoice: ,
264
265   settype .code:n = \setquantitytype #1 ,
266
267   unknown .code:n =
268     \tl_set:Nx \l__physicx_tmpa_tl { \tl_head:N \l_keys_key_str }
269     \token_if_eq_charcode:NNTF \l__physicx_tmpa_tl \c_backslash_str
270     { \use:n } { \use_ii:nn }
271     {
272       \cs_if_exist:cTF { \tl_tail:N \l_keys_key_str }
273       {
274         \keys_set:nx { physicx/quantity }
275         { size = \exp_not:c { \tl_tail:N \l_keys_key_str } }
276         \use_none:n
277       }
278       { \use:n }
279     }
280     {
281       \physicx_search_also:nnF
282       {
283         physicx/quantity/type ,
284       }
285       {#1}
286       {
287         \msg_error:nnxx { physicx } { unknown-key }
288         \l_keys_path_str { physicx/quantity }

```

```

289     }
290   } ,
291 }
292 \NewDocumentCommand \setquantitytype { >{ \TrimSpaces } m }
293 { \physicx_new_type:nnn { quantity } {#1} }
294 \setquantitytype { b } { left={[] , right={[]}} , }
295 \setquantitytype { B } { left={\{ } , right={\}} , }
296 \setquantitytype { p } { left={({ } , right={})} , }
297 \setquantitytype { v } { left=\vert , right=\vert , }
298 \setquantitytype { V } { left=\Vert , right=\Vert , }
299 \setquantitytype { a } { left=\langle , right=\rangle , }
300 \setquantitytype { m } { left=\begin{matrix} , right=\end{matrix} , noauto }
301 \setquantitytype { bm } { left=\begin{bmatrix} , right=\end{bmatrix} , noauto }
302 \setquantitytype { Bm } { left=\begin{Bmatrix} , right=\end{Bmatrix} , noauto }
303 \setquantitytype { pm } { left=\begin{pmatrix} , right=\end{pmatrix} , noauto }
304 \setquantitytype { vm } { left=\begin{vmatrix} , right=\end{vmatrix} , noauto }
305 \setquantitytype { Vm } { left=\begin{Vmatrix} , right=\end{Vmatrix} , noauto }
306 \setquantitytype { sm } { left=\begin{smallmatrix} , right=\end{smallmatrix} , noauto }
307 \physicx_mathtools:T
308 {
309   \setquantitytype { m* } { left=\begin{matrix*} , right=\end{matrix*} , noauto }
310   \setquantitytype { bm* } { left=\begin{bmatrix*} , right=\end{bmatrix*} , noauto }
311   \setquantitytype { Bm* } { left=\begin{Bmatrix*} , right=\end{Bmatrix*} , noauto }
312   \setquantitytype { pm* } { left=\begin{pmatrix*} , right=\end{pmatrix*} , noauto }
313   \setquantitytype { vm* } { left=\begin{vmatrix*} , right=\end{vmatrix*} , noauto }
314   \setquantitytype { Vm* } { left=\begin{Vmatrix*} , right=\end{Vmatrix*} , noauto }
315   \setquantitytype { sm* } { left=\begin{smallmatrix*} , right=\end{smallmatrix*} , noauto }
316   \setquantitytype { sbm } { left=\begin{bsmallmatrix} , right=\end{bsmallmatrix} , noauto }
317   \setquantitytype { sBm } { left=\begin{Bsmallmatrix} , right=\end{Bsmallmatrix} , noauto }
318   \setquantitytype { spm } { left=\begin{psmallmatrix} , right=\end{psmallmatrix} , noauto }
319   \setquantitytype { svm } { left=\begin{vsmallmatrix} , right=\end{vsmallmatrix} , noauto }
320   \setquantitytype { sVm } { left=\begin{Vsmallmatrix} , right=\end{Vsmallmatrix} , noauto }
321   \setquantitytype { sbm* } { left=\begin{bsmallmatrix*} , right=\end{bsmallmatrix*} , noauto }
322   \setquantitytype { sBm* } { left=\begin{Bsmallmatrix*} , right=\end{Bsmallmatrix*} , noauto }
323   \setquantitytype { spm* } { left=\begin{psmallmatrix*} , right=\end{psmallmatrix*} , noauto }
324   \setquantitytype { svm* } { left=\begin{vsmallmatrix*} , right=\end{vsmallmatrix*} , noauto }
325   \setquantitytype { sVm* } { left=\begin{Vsmallmatrix*} , right=\end{Vsmallmatrix*} , noauto }
326 }
327 \keys_set:nn { physicx/quantity }
328 {
329   left-size = \left ,
330   right-size = \right ,
331   type = p ,
332 }
\physicx_xquantity:nn
\newxquantity
\NewXQuantity
333 \cs_new:Npn \physicx_xquantity:nn #1#2
334 {
335   \group_begin:
336   \keys_set:nn { physicx/quantity } {#1}
337   \tl_if_empty:nF {#2} { \tl_set:Nn \l__physicx_quantity_code_tl {#2} }
338   \__physicx_xquantity_aux:oooo
339   { \l__physicx_quantity_left_tl }
340   { \l__physicx_quantity_args_tl }

```

```

341     { \l__physicx_quantity_code_tl }
342     { \l__physicx_quantity_right_tl }
343   \group_end:
344 }
345 \cs_new:Npn \__physicx_xquantity_aux:nnnn #1#2#3#4
346 {
347   \l__physicx_quantity_pre_tl
348   \bool_lazy_or:nnTF
349     { \tl_if_empty_p:N \l__physicx_quantity_left_size_tl }
350     { \tl_if_empty_p:N \l__physicx_quantity_right_size_tl }
351     { #1 #2 #3 #4 }
352   {
353     \bool_lazy_or:nnTF
354       { \token_if_eq_meaning_p:NN \l__physicx_quantity_left_size_tl \left }
355       { \token_if_eq_meaning_p:NN \l__physicx_quantity_right_size_tl \right }
356       { \physicx_left: #1 #2 #3 \physicx_right: #4 }
357     {
358       \physicx_left:N \l__physicx_quantity_left_size_tl #1 #2
359       #3
360       \physicx_right:N \l__physicx_quantity_right_size_tl #4
361     }
362   }
363   \l__physicx_quantity_post_tl
364 }
365 \NewDocumentCommand \xquantity { } { \physicx_xquantity:nn }
366 \cs_generate_variant:Nn \__physicx_xquantity_aux:nnnn { oooo }
367 \NewDocumentCommand \newxquantity { m o o m m }
368 {
369   \IfNoValueTF {#2}
370   {
371     \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
372       { \newcommand ##1 }
373   }
374   {
375     \IfNoValueTF {#3}
376     {
377       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
378         { \newcommand ##1 [#2] }
379     }
380     {
381       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
382         { \newcommand ##1 [#2] [#3] }
383     }
384   }
385   \exp_args:Nc \__physicx_new_xquantity_aux:w
386     { \cs_to_str:N #1~star }
387     { \physicx_xquantity:nn { #4 , noauto } {#5} }
388   \exp_args:Nc \__physicx_new_xquantity_aux:w
389     { \cs_to_str:N #1~unstar }
390     { \physicx_xquantity:nn { #4 } {#5} }
391   \exp_args:NNx \newcommand #1
392   {
393     \exp_not:N \@ifstar
394     \exp_not:c { \cs_to_str:N #1~star }

```



```

395         \exp_not:c { \cs_to_str:N #1~unstar }
396     }
397 }
398 \NewDocumentCommand \NewXQuantity { m m m m }
399 {
400     \NewDocumentCommand #1 { s #2 }
401     {
402         \IfBooleanTF {##1}
403         { \physicsx_quantity:nn { #3 , noauto } {#4} }
404         { \physicsx_quantity:nn { #3 } {#4} }
405     }
406 }
407 \NewXQuantity \qxqty { 0{ } m } { #2 } {#3}
408 \NewXQuantity \txqty { 0{p} 0{ } m } { type={#2}, #3 } {#4}

```

(End definition for `\physicsx_quantity:nn`, `\newxquantity`, and `\NewXQuantity`. These functions are documented on page ??.)

1.2.2 Legacy quantity

```

\physicsx_declare_legacy_quantity:nnNn
  \@declarequantitycmd
409 \tl_new:N \physicsx_tmp
410 \tl_new:N \l__physicsx_cmd_noauto_body_tl
411 \bool_new:N \l__physicsx_cmd_noauto_body_bool
412 \tl_new:N \l__physicsx_cmd_auto_body_tl
413 \bool_new:N \l__physicsx_cmd_auto_body_bool
414 \tl_new:N \l__physicsx_cmd_arg_spec_tl
415 \int_new:N \l__physicsx_cmd_arg_int
416 \cs_new:Npn \__physicsx_declare_init:nnn #1#2#3
417 {
418     \tl_clear:N \l__physicsx_cmd_noauto_body_tl
419     \tl_clear:N \l__physicsx_cmd_auto_body_tl
420     \tl_clear:N \l__physicsx_cmd_arg_spec_tl
421     \int_set:Nn \l__physicsx_cmd_arg_int {#1}
422     \bool_set:Nn \l__physicsx_cmd_noauto_body_bool {#2}
423     \bool_set:Nn \l__physicsx_cmd_auto_body_bool {#3}
424 }
425 % noauto, auto, cmd, body
426 \cs_new:Npn \physicsx_declare_legacy_quantity:nnNn #1#2#3#4
427 {
428     \__physicsx_declare_init:nnn { 3 } {#1} {#2}
429     \__physicsx_declare_legacy_quantity_aux:nw #4
430     \q_recursion_tail \q_recursion_tail \q_recursion_stop
431     \__physicsx_declare_legacy_quantity_aux:NcVVV
432     #3 { \cs_to_str:N #3 ~ body }
433     \l__physicsx_cmd_arg_spec_tl
434     \l__physicsx_cmd_noauto_body_tl
435     \l__physicsx_cmd_auto_body_tl
436 }
437 % arg spec, pre, body to replace(start from #4), post
438 \cs_new:Npn \__physicsx_declare_legacy_quantity_aux:nnnn #1#2#3#4
439 {
440     \int_incr:N \l__physicsx_cmd_arg_int
441     \if_int_compare:w \l__physicsx_cmd_arg_int < 10 \exp_stop_f:

```

```

442 \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
443 \tl_set:Nx \l__physicx_tmp_tl
444 {
445   {
446     \exp_not:N \tl_if_novalue_p:n
447     {
448       \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:
449       \or: \or: \or:
450       \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
451       \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
452       \fi:
453     }
454   }
455 }
456 \if_bool:N \l__physicx_cmd_noauto_body_bool
457 \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
458 \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
459 {
460   {
461     % if is '.', use none
462     \str_if_eq:nnTF {#2} {.} {} {#2}
463     #3
464     \str_if_eq:nnTF {#4} {.} {} {#4}
465   }
466 }
467 \fi:
468 \if_bool:N \l__physicx_cmd_auto_body_bool
469 \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
470 \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
471 { { ##1 #2 #3 ##2 #4 } }
472 \fi:
473 \fi:
474 }
475 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
476 {
477   \quark_if_recursion_tail_stop:n {#1}
478   \quark_if_recursion_tail_stop:n {#2}
479   \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
480   \__physicx_declare_legacy_quantity_aux:nw
481 }
482 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
483 {
484   \__physicx_nauto_case:nnnn
485   { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
486   {
487     \cs_set_protected:Npn #1
488     {
489       \peek_charcode_ignore_spaces:NTF \let
490       { #2 } { #2 [ \physicx_left: ] \physicx_right: }
491     }
492     \DeclareDocumentCommand #2 { 0{##2} m s #3 }
493     {
494       \IfBooleanTF { ##3 }
495       { \bool_case_false:n {#4} }

```

```

496         { \bool_case_false:n {#5} }
497     }
498 }
499 {
500     \cs_set_protected:Npn #1
501     { #2 \c_empty_tl \c_empty_tl }
502     \DeclareDocumentCommand #2 { m m s #3 }
503     { \bool_case_false:n {#4} }
504 }
505 }
506 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
507 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
508 {
509     \bool_if:NTF \l__physicx_cmd_noauto_body_bool
510     {
511         \bool_if:NTF \l__physicx_cmd_auto_body_bool
512         {#1} {#2}
513     }
514     {
515         \bool_if:NTF \l__physicx_cmd_auto_body_bool
516         {#3} {#4}
517     }
518 }
519 \cs_set_protected:Npn \@declarequantitycmd
520 { \physicx_declare_legacy_quantity:nnNn }

```

(End definition for `\physicx_declare_legacy_quantity:nnNn` and `\@declarequantitycmd`. These functions are documented on page ??.)

<code>\quantity</code> <code>\evaluated</code> <code>\matrixquantity</code> <code>\smallmatrixquantity</code>	Redefine some macros in physics package. <pre> 521 \if_bool:N \g__physicx_reqty_bool 522 \physicx_declare_legacy_quantity:nnNn 523 \c_true_bool \c_true_bool \quantity 524 { 525 { !g } { { \{ } { #4 } { \} } } 526 { !o } { { [} { #5 } {] } } 527 { !d() } { { (} { #6 } {) } } 528 { !d } { { \vert } { #7 } { \vert } } 529 { !d<> } { { \langle } { #8 } { \rangle } } 530 { !d== } { { \Vert } { #9 } { \Vert } } 531 } 532 \physicx_declare_legacy_quantity:nnNn 533 \c_true_bool \c_true_bool \evaluated 534 { 535 { !g } { { . } { #4 \nobreak } { \vert } } 536 { !d[] } { { [} { #5 \nobreak } { \vert } } 537 { !d() } { { (} { #6 \nobreak } { \vert } } 538 } 539 \physicx_declare_legacy_quantity:nnNn 540 \c_true_bool \c_false_bool \matrixquantity 541 { 542 { !g } 543 { 544 { \IfBooleanT{#3}{\left\{ } } </pre>
--	--

```

545     { \begin{matrix} #4 \end{matrix} }
546     { \IfBooleanT{#3}{\right\}} }
547   }
548   { !o } { {\begin{bmatrix} } {#5} { \end{bmatrix} } }
549   { !d() }
550   {
551     { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
552     { \begin{matrix} #6 \end{matrix} }
553     { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
554   }
555   { !d|| } { { \begin{vmatrix} } {#7} { \end{vmatrix} } }
556   { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } { \right\rangle } }
557   { !d== } { { \begin{Vmatrix} } {#9} { \end{Vmatrix} } }
558 }
559 \physicsx_declare_legacy_quantity:nnNn
560 \c_true_bool \c_false_bool \smallmatrixquantity
561 {
562   { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } { \right\} } }
563   { !o } { { \left[ } { \begin{smallmatrix} #5 \end{smallmatrix} } { \right]} }
564   { !d() }
565   {
566     { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
567     { \begin{smallmatrix} #6 \end{smallmatrix} }
568     { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
569   }
570   { !d|| } { { \left\vert } { \begin{smallmatrix} #7 \end{smallmatrix} } { \right\vert} }
571   { !d<> } { { \left\langle } { \begin{smallmatrix} #8 \end{smallmatrix} } { \right\rangle } }
572   { !d== } { { \left\| } { \begin{smallmatrix} #9 \end{smallmatrix} } { \right\|} }
573 }
574 \fi:

```

(End definition for \quantity and others. These functions are documented on page ??.)

```

\physicsx_declare_legacy_paren:NnnnNnN
  \@declareparenccmd
575 %% cmd, arg spec, replace(start from #6), pre, left, right, post
576 \cs_new:Npn \physicsx_declare_legacy_paren:NnnnNnN #1#2#3#4#5#6#7
577 {
578   \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
579   {
580     \bool_case_true:nF
581     {
582       { \bool_if_p:n {##2} } { #4 \physicsx_left:N \bigl #5 #3 \physicsx_right:N \bigr
583       { \bool_if_p:n {##3} } { #4 \physicsx_left:N \Bigl #5 #3 \physicsx_right:N \Bigr
584       { \bool_if_p:n {##4} } { #4 \physicsx_left:N \biggl #5 #3 \physicsx_right:N \biggr
585       { \bool_if_p:n {##5} } { #4 \physicsx_left:N \Biggl #5 #3 \physicsx_right:N \Biggr
586     }
587     {
588       \IfBooleanTF {##1}
589       { #4 #5 #3 #6 #7 }
590       { #4 \physicsx_left: #5 #3 \physicsx_right: #6 #7 }
591     }
592   }
593 }
594 \cs_set_protected:Npn \@declareparenccmd
595 { \physicsx_declare_legacy_paren:NnnnNnN }

```

(End definition for `\physics_declare_legacy_paren:NnnnNNn` and `\@declareparencmd`. These functions are documented on page ??.)

```

\qty      Redefine some macros in physics package.
\mqty     596 \if_bool:N \g__physics_reqty_bool
\smqty    597 \physics_option_or:nnT { compat } { short }
\pqty     598 {
\bqty     599   \cs_set:Npn \qty { \quantity }
\vqty     600   \physics_declare_legacy_paren:NnnnNNn \pqty { m } {#6} { } { } { } { }
\Bqty     601   \physics_declare_legacy_paren:NnnnNNn \bqty { m } {#6} { } { } [ ] { }
\absolutevalue 602   \physics_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } { } \vert \vert { }
\eval     603   \physics_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } { } \{ \} { }
\abs      604 }
\norm     605 \physics_declare_legacy_paren:NnnnNNn \absolutevalue
\order    606 { m } {#6} { } { } \vert \vert { }
\oorder   607 \physics_option_or:nnT { compat } { short }
\commutator 608 {
\poissonbracket 609   \cs_set:Npn \eval { \evaluated }
\pb       610   \cs_set:Npn \abs { \absolutevalue }
\anticommutator 611 }
\acomm    612 \physics_declare_legacy_paren:NnnnNNn \norm
          613 { m } {#6} { } { } \lVert \rVert { }
          614 \physics_compat:TF
          615 {
          616   \physics_declare_legacy_paren:NnnnNNn \order
          617   { m } {#6} { } { \c_physics_Order_t1 } { } { }
          618 }
          619 {
          620   \physics_declare_legacy_paren:NnnnNNn \order
          621   { m } {#6} { } { \c_physics_order_t1 } { } { }
          622 }
          623 \physics_declare_legacy_paren:NnnnNNn \commutator
          624 { m m } { #6 , #7 } { } { } [ ] { }
          625 \physics_option_or:nnT { compat } { short }
          626 { \cs_set:Npn \comm { \commutator } }
          627 \physics_declare_legacy_paren:NnnnNNn \poissonbracket
          628 { m m } { #6 , #7 } { } { } \{ \} { }
          629 \physics_option_or:nnT { compat } { short }
          630 {
          631   \cs_set:Npn \pb { \poissonbracket }
          632   \cs_set:Npn \anticommutator { \poissonbracket }
          633   \cs_set:Npn \acomm { \poissonbracket }
          634 }
          635 \fi:
          636 \physics_declare_legacy_paren:NnnnNNn \OOrder
          637 { m } {#6} { } { \c_physics_Order_t1 } { } { }
          638 \physics_declare_legacy_paren:NnnnNNn \oorder
          639 { m } {#6} { } { \c_physics_order_t1 } { } { }

```

(End definition for `\qty` and others. These functions are documented on page ??.)

1.3 Matrix things

1.3.1 Matrix auxillary functions

```

640 \cs_new_nopar:Npn \__physicx_matrix_calc:nn #1#2
641 {
642   \int_set:Nn \l__physicx_matrix_rows_int
643     { \int_max:nn {#1} \l__physicx_matrix_rows_int }
644   \int_set:Nn \l__physicx_matrix_cols_int
645     { \int_max:nn {#2} \l__physicx_matrix_cols_int }
646 }
647 % use matrix element
648 \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
649 {
650   \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
651   \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
652   \else:
653     \exp_not:o { \physicxempty }
654   \fi:
655 }
656 % set matrix element, check or not
657 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
658 { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
659 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
660 {
661   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
662   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
663 }
664 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_cke:nnn #1#2#3
665 {
666   \tl_if_empty:nTF {#3}
667     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
668     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
669 }
670 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
671 {
672   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
673   {
674     \tl_if_empty:nTF {#3}
675       { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
676       { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
677   }
678 }
679 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
680   \__physicx_matrix_set_r_c_ckigep:nnn
681 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
682   \__physicx_matrix_set_r_c_nock:nnn
683 % align, cr, sep symbol
684 \str_const:Nn \physicx@align { , }
685 \str_const:Nn \physicx@cr { ; }
686 \str_const:Nn \physicx@sep { , }
687 \bool_new:N \l__physicx_matrix_infinite_bool
688 \bool_new:N \l__physicx_matrix_dotrow_bool
689 \bool_new:N \l__physicx_matrix_dotcol_bool
690 \tl_new:N \l__physicx_matrix_array_tl
691 \tl_new:N \l__physicx_matrix_body_tl
692 \int_new:N \l__physicx_matrix_rows_int
693 \int_new:N \l__physicx_matrix_cols_int

```

```

694 \tl_new:N \l__physicx_matrix_main_tl
695 \clist_new:N \l__physicx_matrix_diag_clist
696 \clist_new:N \l__physicx_matrix_item_clist
697 \bool_new:N \l__physicx_matrix_diag_bool
698 \seq_new:N \l__physicx_row_list_seq
699 \seq_new:N \l__physicx_col_list_seq
700 % expand input
701 \cs_new_eq:NN \__physicx_expand:w \exp_not:o
702 %% main, row iterate, col iterate
703 \cs_new_nopar:Npn \physicx@matrxiement #1#2#3 { #1 \sb { #2 #3 } }
704 \cs_new_nopar:Npn \__physicx_matrix_row_iterate:n #1 { #1 }
705 \tl_new:N \l__physicx_matrix_last_row_tl
706 \tl_new:N \l__physicx_matrix_last_col_tl
707 \cs_new_nopar:Npn \__physicx_matrix_col_iterate:n #1 { #1 }
708 \cs_new_nopar:Npn \__physicx_matrix_begin:w { }
709 \cs_new_nopar:Npn \__physicx_matrix_end:w { }
710 \cs_new_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn
711 \bool_new:N \l__physicx_matrix_expand_element_bool
712 % when element is empty use \physicxempty
713 \tl_new:N \physicxempty
714 % save 'element-except' key's value
715 \tl_new:N \physicxexcept
716 \tl_new:N \l__physicx_matrix_args_tl
717 \tl_new:N \l__physicx_matrix_after_begin_tl
718 \tl_new:N \l__physicx_matrix_after_end_tl
719 \bool_new:N \l__physicx_matrix_transpose_bool
720 \bool_new:N \l__physicx_matrix_enhanced_bool
721 \dim_new:N \l__physicx_matrix_sep_dim
722 \cs_new:Npn \__physicx_adi:nnn #1#2#3 { #1#2#3 }
723 \tl_new:N \l__physicx_matrix_beginning_tl
724 \tl_new:N \l__physicx_matrix_ending_tl

```

1.3.2 Matrix keys

```

725 \keys_define:nn { physicx }
726 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
727 \keys_define:nn { physicx/matrix }
728 {
729   array .tl_set:N = \l__physicx_matrix_array_tl ,
730   expand .choice: ,
731   expand / none .code:n =
732     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
733   expand / text-expand .code:n =
734     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
735   expand / f .code:n =
736     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
737   expand / romanual .meta:n = { expand = f } ,
738   expand / x .code:n =
739     \cs_set_eq:NN \__physicx_expand:w \use:n ,
740   expand / edef .meta:n = { expand = x } ,
741   rows .int_set:N = \l__physicx_matrix_rows_int ,
742   cols .int_set:N = \l__physicx_matrix_cols_int ,
743   auto-update .choice: ,
744   auto-update / true .code:n =
745     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,

```

```

746 auto-update / false .code:n =
747   \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
748   auto-update .default:n = true ,
749   main .tl_set:N = \l__physicx_matrix_main_tl ,
750   row-list .code:n =
751     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,
752   col-list .code:n =
753     \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
754   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
755   infinite .default:n = true ,
756   !infinite .code:n =
757     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
758   element-code .cs_set:Np = \physicx@matricelement #1#2##3 ,
759   element-code* .choice: ,
760   element-code* / except-empty .code:n =
761     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
762       \physicx@matricelement
763     \cs_set:Npn \physicx@matricelement ##1##2##3
764       {
765         \tl_if_empty:nTF {##1}
766           {##1}
767           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
768       } ,
769   element-code* / except-dots .code:n =
770     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
771       \physicx@matricelement
772     \cs_set:Npn \physicx@matricelement ##1##2##3
773       {
774         \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
775           {##1}
776           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
777       } ,
778   element-code* / except-tl .code:n =
779     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
780       \physicx@matricelement
781     \cs_set:Npn \physicx@matricelement ##1##2##3
782       {
783         \tl_if_in:onTF { \physicxexcept } {##1}
784           {##1}
785           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
786       } ,
787   element-code* / except-regex .code:n =
788     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
789       \physicx@matricelement
790     \cs_set:Npn \physicx@matricelement ##1##2##3
791       {
792         \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
793           {##1}
794           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
795       } ,
796   element-code* / only-regex .code:n =
797     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
798       \physicx@matricelement
799     \cs_set:Npn \physicx@matricelement ##1##2##3

```



```

800     {
801         \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
802         { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
803         {##1}
804     } ,
805 element-code* / unknown .code:n =
806     \cs_set:Npx \physicx@matricelement { \exp_not:c {#1} },
807 element-except .tl_set:N = \physicxexcept ,
808 element-except+ .code:n =
809     \tl_put_right:Nn \physicxexcept {#1} ,
810 expand-element .bool_set:N = \l__physicx_matrix_expand_element_bool ,
811 expand-element .default:n = true ,
812 empty .tl_set:N = \physicxempty ,
813 check .choice: ,
814 check / none .code:n =
815     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
816     \__physicx_matrix_set_r_c_nock:nnn ,
817 check / empty .code:n =
818     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
819     \__physicx_matrix_set_r_c_ckepp:nnn ,
820 check / ignore .code:n =
821     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
822     \__physicx_matrix_set_r_c_ckig:nnn ,
823 check / igep .code:n =
824     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
825     \__physicx_matrix_set_r_c_ckigep:nnn ,
826 check / all .code:n =
827     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
828     \__physicx_matrix_set_r_c_ckall:nnn ,
829 check .default:n = all ,
830 row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
831 col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
832 last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
833 last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
834 diag .clist_set:N = \l__physicx_matrix_diag_clist ,
835 diag+ .code:n =
836     \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
837 diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
838 diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
839 diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
840 item .clist_set:N = \l__physicx_matrix_item_clist ,
841 item+ .code:n =
842     \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
843 item-now .code:n = \physicx_matrix_item_parse:n {#1} ,
844 item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
845 item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
846 check-range .choice: ,
847 check-range / true .code:n = \physicx_parse_range_check: ,
848 check-range / false .code:n = \physicx_parse_range_noccheck: ,
849 check-range .default:n = true ,
850 begin .tl_set:N = \__physicx_matrix_begin:w ,
851 end .tl_set:N = \__physicx_matrix_end: ,
852 args .code:n =
853     \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,

```

```

854 args* .tl_set:N = \l__physicx_matrix_args_tl ,
855 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
856 after-begin+ .code:n =
857   { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
858 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
859 after-end+ .code:n =
860   { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
861 sepdim .dim_set:N = \l__physicx_matrix_sepdim ,
862 type .multichoice: ,
863 saveto .tl_set:N = \l__physicx_matrix_save_tl ,
864 saveto* .code:n =
865   \tl_set:N \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
866 transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
867 transpose .default:n = true ,
868 ' .meta:n = { transpose = true } ,
869 T .meta:n = { transpose = true } ,
870 MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
871 enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
872 enhanced .default:n = true ,
873 !enhanced .code:n =
874   \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
875 cr .tl_set:N = \physicx@cr ,
876 align .tl_set:N = \physicx@align ,
877 sep .tl_set:N = \physicx@sep ,
878 adi-order .choice: ,
879 adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
880 adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
881 adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
882 adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
883 adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
884 adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
885 beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
886 beginning+ .code:n =
887   \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
888 ending .tl_set:N = \l__physicx_matrix_ending_tl ,
889 ending+ .code:n =
890   \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
891
892 settype .code:n = \setmatrixtype #1 ,
893
894 unknown .code:n =
895   \physicx_search_also:nnF
896   {
897     physicx/matrix/type ,
898     physicx/matrix/expand ,
899     physicx/matrix/element-code* ,
900   }
901   {#1}
902   {
903     \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
904     {
905       \keys_set:nx { physicx/matrix }
906       { MaxMatrixCols = \l_keys_key_str }
907     }
908   }

```

```

908         {
909             \msg_error:nnxx { physicx } { unknown-key }
910             \l_keys_path_str { physicx/matrix }
911         }
912     } ,
913 }

```

\physicx_matrix_new_type:nnn

```

914 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
915 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
916 \cs_new:Npn \physicx_matrix_new_type:nn
917 { \physicx_new_type:nnn { matrix } }
918 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
919 {
920     \IfBooleanTF {#1}
921     { \physicx_matrix_new_type:nn {#2} }
922     { \physicx_matrix_new_type:nnn {#2} }
923 }

```

(End definition for \physicx_matrix_new_type:nnn, \physicx_matrix_new_type:nn, and \setmatrixtype.
These functions are documented on page ??.)

A few types.

```

924 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
925 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
926 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
927 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
928 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
929 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
930 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
931 \physicx_mathtools:T
932 {
933     \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
934     \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
935     \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
936     \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
937     \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
938     \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
939     \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
940     \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
941     \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
942     \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
943     \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
944     \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
945     \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
946     \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
947     \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
948     \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
949     \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
950 }

```

\setmatrixdata Set matrix data, one can use ‘...-data’ key to use it.

```

951 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
952 { \clist_set:cn { physicx@ #1 data@ #2 } }
953 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2

```

```

954 {
955   \clist_clear:c { l__physicx_matrix_ #1 _clist }
956   \__physicx_matrix_add_data:nn {#1} {#2}
957 }
958 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
959 {
960   \clist_map_inline:nn {#2}
961   {
962     \clist_concat:ccc
963     { l__physicx_matrix_ #1 _clist }
964     { l__physicx_matrix_ #1 _clist }
965     { physicx@ #1 data@ #2 }
966   }
967 }

```

(End definition for `\setmatrixdata`. This function is documented on page ??.)

Initial settings.

```

968 \keys_set:nn { physicx/matrix }
969 {
970   type = m ,
971   saveto = ? ,
972 }

```

`\qxmatri`

```

973 %% basicly, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
974 xmatmmn-in-the-physics-package, but changed some
975 % #1 = boolean, saveto matrix
976 % #2 = star, infinite
977 % #3 = options
978 % #4 = letter for the entries
979 % #5 = number of rows
980 % #6 = number of explicit rows, default = 3
981 % #7 = number of columns
982 % #8 = number of explicit columns, default = 3
983 \DeclareDocumentCommand \qxmatri { t= s 0{type=p} m m 0{3} m 0{3} }
984 {
985   \group_begin:
986   \IfBooleanTF { #2 }
987   { \bool_set_true:N \l__physicx_matrix_infinite_bool }
988   { \bool_set_false:N \l__physicx_matrix_infinite_bool }
989   \int_set:Nn \l__physicx_matrix_rows_int {#6}
990   \int_set:Nn \l__physicx_matrix_cols_int {#8}
991   \IfBooleanTF {#1}
992   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
993   { \keys_set:nn { physicx/matrix } {#3} }
994   \physicx_qxmatri:nnn {#4} {#5} {#7}
995   \__physicx_matrix_save_or_print:
996   \group_end:
997 }
998 \cs_new_protected:Nn \physicx_qxmatri:nnn
999 {
1000   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1001   {

```

```

1002     \_physicx_matrix_appto_body_e:nnn
1003 }
1004 {
1005     \cs_set_eq:NN \_physicx_qxmatrix_appto_body:nnn
1006     \_physicx_matrix_appto_body_ne:nnn
1007 }
1008 % clear the variable containing the body of the matrix
1009 \tl_clear:N \l__physicx_matrix_body_tl
1010 % set the tentative number of explicit rows
1011 \physicx_if_num:nTF { #2 }
1012 {
1013     % number of rows is an integer
1014     \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
1015     {
1016         % if #2 <= rows, we don't want a row of dots
1017         \bool_set_false:N \l__physicx_matrix_dotrow_bool
1018         \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1019     }
1020     % we want a row of dots
1021     \bool_set_true:N \l__physicx_matrix_dotrow_bool
1022 }
1023 {
1024     % number of rows is symbolic, we want a row of dots
1025     \bool_set_true:N \l__physicx_matrix_dotrow_bool
1026 }
1027 % set the tentative number of explicit columns
1028 \physicx_if_num:nTF { #3 }
1029 {
1030     % number of cols is an integer
1031     \int_compare:nTF { #3 <= \l__physicx_matrix_cols_int }
1032     {
1033         % if #3 <= cols, we don't want a column of dots
1034         \bool_set_false:N \l__physicx_matrix_dotcol_bool
1035         \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1036     }
1037     % we want a column of dots
1038     \bool_set_true:N \l__physicx_matrix_dotcol_bool
1039 }
1040 {
1041     % number of columns is symbolic, we want a column of dots
1042     \bool_set_true:N \l__physicx_matrix_dotcol_bool
1043 }
1044 % loop through the rows
1045 \int_step_inline:nn { \l__physicx_matrix_rows_int }
1046 {
1047     % add the first entry in the row
1048     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1049     \_physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
1050     % add the further entries in the explicit columns
1051     \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1052     {
1053         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
1054         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1055         \_physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
1056     }
1057     % if we have a column of dots, add \cdots and the last entry
1058     \bool_if:NT \l__physicx_matrix_dotcol_bool
1059     {

```

```

1056         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{##1 #3} }
1057         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1058         \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {#3}
1059     }
1060     % infinite matrix, add \cdots
1061     \bool_if:NT \l__physicx_matrix_infinite_bool
1062     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1063     \if_int_compare:w ##1 = \l__physicx_matrix_rows_int
1064     \scan_stop:
1065     \else:
1066         % finish up the row
1067         \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1068     \fi:
1069 }
1070 % finish up the rows
1071 \bool_if:NT \l__physicx_matrix_dotrow_bool
1072 {
1073     % finish up the row
1074     \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1075     % if we have a row of dots, fill it in
1076     \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1077     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1078     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1079     \bool_if:NT \l__physicx_matrix_dotcol_bool
1080     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
1081     \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1082     % fill the last row
1083     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
1084     \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
1085     \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1086     {
1087         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
1088         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1089         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
1090     }
1091     \bool_if:NT \l__physicx_matrix_dotcol_bool
1092     {
1093         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
1094         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1095         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
1096     }
1097     % if the matrix is infinite, add a further column with \cdots
1098     \bool_if:NT \l__physicx_matrix_infinite_bool
1099     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1100 }
1101 % if the matrix is infinite, add a final row
1102 \bool_if:NT \l__physicx_matrix_infinite_bool
1103 {
1104     % finish up the row
1105     \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1106     \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1107     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1108     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1109     \bool_if:NT \l__physicx_matrix_dotcol_bool

```

```

1110         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1111         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1112         % update cols
1113         \bool_if:NTF \l__physicx_matrix_dotcol_bool
1114         { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1115         { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1116     }
1117 }

```

(End definition for `\qmatrix`. This function is documented on page ??.)

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
1118 \cs_new:Npn \physicx_matrix_diag_parse:n #1
1119 {
1120     \keyval_parse:nnn
1121     \__physicx_matrix_diag_parse_aux:n
1122     \__physicx_matrix_diag_parse_aux:nn
1123     {#1}
1124 }
1125 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
1126 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
1127 {
1128     \str_case_e:nnF {#1}
1129     {
1130         { auto-update }
1131         {
1132             \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
1133             \__physicx_matrix_calc:nn
1134         }
1135         { noauto-update }
1136         {
1137             \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
1138         }
1139         { true }
1140         {
1141             \bool_set_true:N \l__physicx_matrix_diag_bool
1142             \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1143             \__physicx_diagonalmatrix_set_diag:
1144         }
1145         { false }
1146         {
1147             \bool_set_false:N \l__physicx_matrix_diag_bool
1148             \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1149             \__physicx_diagonalmatrix_no_diag:
1150         }
1151     }
1152     { \msg_error:nnn { physicx } { diag-key } {#1} }
1153 }
1154 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
1155 {
1156     \tl_set:Nn \l__physicx_tmpdiag_tl {#2}
1157     \tl_set:Nx \l__physicx_tmpdiag_tl
1158     { \__physicx_expand:w \l__physicx_tmpdiag_tl }
1159     \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl

```

```

1160 \tl_if_head_eq_charcode:nNTF {#1} '
1161 {
1162     \exp_args:Nf \__physicx_matrix_diag_parse_aux_anti:n
1163     { \tl_tail:n {#1} }
1164 }
1165 { \__physicx_matrix_diag_parse_aux_regu:n {#1} }
1166 }
1167 \cs_new:Npn \__physicx_diagonalmatrix_set_diag:
1168 {
1169     \int_zero:N \l__physicx_matrix_cols_int
1170     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1171     {
1172         \int_incr:N \l__physicx_matrix_cols_int
1173         \physicx_matrix_set_r_c:nnn {##1} {##1} {##2}
1174     }
1175     \int_set_eq:NN \l__physicx_matrix_rows_int
1176     \l__physicx_matrix_cols_int
1177 }
1178 \cs_new:Npn \__physicx_diagonalmatrix_no_diag:
1179 {
1180     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1181     { \physicx_matrix_set_r_c:nnn {##1} {##1} {##2} }
1182     \__physicx_matrix_diag_calc:nn
1183     { \seq_count:N \l__physicx_tmpdiag_seq }
1184     { \seq_count:N \l__physicx_tmpdiag_seq }
1185 }
1186 \cs_new_eq:NN \__physicx_diagonalmatrix_diag_main:
1187 \__physicx_diagonalmatrix_no_diag:
1188 \cs_new:Npn \__physicx_matrix_diag_parse_aux_regu:n #1
1189 {
1190     \if_int_compare:w #1 = 0 \exp_stop_f:
1191         \__physicx_diagonalmatrix_diag_main:
1192     \else:
1193         \if_int_compare:w #1 > 0 \exp_stop_f:
1194             \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1195             {
1196                 \physicx_matrix_set_r_c:nnn
1197                 {##1} { \int_eval:n { ##1 + #1 } } {##2}
1198             }
1199             \__physicx_matrix_diag_calc:nn
1200             { \seq_count:N \l__physicx_tmpdiag_seq }
1201             { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1202         \else:
1203             \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1204             {
1205                 \physicx_matrix_set_r_c:nnn
1206                 { \int_eval:n { ##1 - #1 } } {##1} {##2}
1207             }
1208             \__physicx_matrix_diag_calc:nn
1209             { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1210             { \seq_count:N \l__physicx_tmpdiag_seq }
1211         \fi:
1212     \fi:
1213 }

```



```

1214 \cs_new:Npn \__physicx_matrix_diag_parse_aux_anti:n #1
1215 {
1216   \if_int_compare:w #1 = 0 \exp_stop_f:
1217     \__physicx_matrix_diag_calc:nn
1218     { \seq_count:N \l__physicx_tmpdiag_seq }
1219     { \seq_count:N \l__physicx_tmpdiag_seq }
1220     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1221     {
1222       \physicx_matrix_set_r_c:nnn
1223       {##1}
1224       { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1225       {##2}
1226     }
1227   \else:
1228     \if_int_compare:w #1 > 0 \exp_stop_f:
1229       \__physicx_matrix_diag_calc:nn
1230       { \seq_count:N \l__physicx_tmpdiag_seq }
1231       { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1232       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1233       {
1234         \physicx_matrix_set_r_c:nnn
1235         {##1}
1236         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1237         {##2}
1238       }
1239     \else:
1240       \__physicx_matrix_diag_calc:nn
1241       { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1242       { \seq_count:N \l__physicx_tmpdiag_seq }
1243       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1244       {
1245         \physicx_matrix_set_r_c:nnn
1246         { \int_eval:n { ##1 - #1 } }
1247         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1248         {##2}
1249       }
1250     \fi:
1251   \fi:
1252 }
1253 \cs_new:Npn \__physicx_matrix_diag_calc:nn
1254 { \__physicx_matrix_autocalc:nn }

```

(End definition for \physicx_matrix_diag_parse:n. This function is documented on page ??.)

```

\physicx_matrix_item_parse:n Parse ‘item...’ keys.
\physicx_matrix_item_parse:o
1255 \cs_new:Npn \physicx_matrix_item_parse:n #1
1256 {
1257   \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1258   \keyval_parse:NNn
1259   \__physicx_matrix_item_parse_aux:n
1260   \__physicx_matrix_item_parse_aux:nn
1261   {#1}
1262 }
1263 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }

```

```

1264 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1265 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2
1266 {
1267   \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1268   \tl_set:Nx \l__physicx_tmpitem_tl
1269     { \__physicx_expand:w \l__physicx_tmpitem_tl }
1270   \physicx_parse_range:nxN \l__physicx_matrix_rows_int
1271     { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1272   \physicx_parse_range:nxN \l__physicx_matrix_cols_int
1273     { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1274   \exp_args:No \tl_if_eq:nnTF
1275     { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1276   {
1277     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1278       {
1279         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1280           {
1281             \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1282           }
1283       }
1284   }
1285   {
1286     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1287       {
1288         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1289           {
1290             \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }
1291             {
1292               \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1293                 {##1} {####1} { \l__physicx_tmpitem_tl }
1294             }
1295           }
1296       }
1297   }
1298 }

```

(End definition for \physicx_matrix_item_parse:n. This function is documented on page ??.)

\physicx_matrix_array_parse:n Parse ‘array...’ keys.

```

\physicx_matrix_array_parse:o 1299 \cs_new:Npn \physicx_matrix_array_parse:n #1
1300 {
1301   \tl_set:Nn \l__physicx_tmparr_tl {#1}
1302   \tl_set:Nx \l__physicx_tmparr_tl
1303     { \__physicx_expand:w \l__physicx_tmparr_tl }
1304   \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl
1305   \__physicx_matrix_autocalc:nn
1306     { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1307     { 0 }
1308   \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1309     {
1310       \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1311       \__physicx_matrix_autocalc:nn
1312         { 0 }
1313       { \seq_count:N \l__physicx_matrix_tmparr_c_sep }

```

```

1314     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1315     {
1316         \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1317     }
1318 }
1319 }
1320 \cs_generate_variant:Nn \physicx_matrix_array_parse:n { o }

```

(End definition for `\physicx_matrix_array_parse:n`. This function is documented on page ??.)

`\physicx_matrix_array_parse_main:` Process ‘main’ key.

```

1321 \cs_new:Npn \physicx_matrix_array_parse_main:
1322 {
1323     \int_step_inline:nn \l__physicx_matrix_rows_int
1324     {
1325         \int_step_inline:nn \l__physicx_matrix_cols_int
1326         {
1327             \exp_args:Nno \physicx_matrix_set_r_c:nnn
1328             {##1} {####1} \l__physicx_matrix_main_tl
1329         }
1330     }
1331 }

```

(End definition for `\physicx_matrix_array_parse_main:`. This function is documented on page ??.)

`__physicx_if_can_num:n` Test if can num, one can use `\int_eval:n`, `\fp_eval:n`, and `\inteval`, `\fpeval` in xfp package (if loaded).

```

1332 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1333 {
1334     \physicx_if_num:nTF {#1}
1335     { \prg_return_true: }
1336     {
1337         \bool_case_true:nTF
1338         {
1339             { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1340             { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1341             {
1342                 \bool_lazy_and_p:nn
1343                 { \cs_if_exist_p:N \inteval }
1344                 { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1345             } { }
1346             {
1347                 \bool_lazy_and_p:nn
1348                 { \cs_if_exist_p:N \fpeval }
1349                 { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1350             } { }
1351         }
1352         { \prg_return_true: }
1353         { \prg_return_false: }
1354     }
1355 }

```

(End definition for `__physicx_if_can_num:n`.)

```

\diagonalmatrix Define \diagonalmatrix.
1356 \DeclareDocumentCommand \diagonalmatrix { t= t+ 0{} m }
1357 {
1358   \group_begin:
1359   \IfBooleanTF {#1}
1360   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1361   { \keys_set:nn { physicx/matrix } { #3 } }
1362   \physicx_construct:nnn { }
1363   {
1364     \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist
1365     \tl_if_empty:nF {#4}
1366     {
1367       \__physicx_if_keyval:nTF {#4}
1368       { \physicx_matrix_diag_parse:n { true, #4 } }
1369       { \physicx_matrix_diag_parse:n { true, 0 = {#4} } }
1370     }
1371   }
1372   { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1373   \bool_lazy_or:nnTF
1374   { \bool_if_p:n {#2} }
1375   { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1376   {
1377     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1378     {
1379       \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1380       \__physicx_matrix_appto_body_e:off
1381     }
1382     {
1383       \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1384       \__physicx_matrix_appto_body_ne:off
1385     }
1386     \use_i_ii:nnn
1387   }
1388   { \use_i:nn }
1389   \__physicx_matrix_transpose:N
1390   \__physicx_diagonalmatrix_generate_enhanced_body:NNN
1391   \__physicx_diagonalmatrix_generate_body:NNN
1392   \__physicx_matrix_save_or_print:
1393   \group_end:
1394 }
1395 \cs_new:Npn \__physicx_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1396 {
1397   \__physicx_matrix_generate_body:NNNN #1#2#3
1398   \__physicx_diagonalmatrix_enhanced:nnn
1399 }
1400 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1401 {
1402   \int_step_inline:nn { #1 - 1 }
1403   {
1404     \int_step_inline:nn { #2 - 1 }
1405     {
1406       \tl_put_right:Nx \l__physicx_matrix_body_tl
1407       {
1408         \exp_after:wN

```

```

1409         \physicx_matrix_use_r_c:nn
1410         #3 {{##1}} {{####1}} &
1411     }
1412 }
1413 \tl_put_right:Nx \l__physicx_matrix_body_tl
1414 {
1415     \exp_after:wN
1416     \physicx_matrix_use_r_c:nn
1417     #3 {{##1}} {{ \int_use:N #2 }} \[\dim_use:N \l__physicx_matrix_sep_dim]
1418 }
1419 }
1420 \int_step_inline:nn { #2 - 1 }
1421 {
1422     \tl_put_right:Nx \l__physicx_matrix_body_tl
1423     {
1424         \exp_after:wN
1425         \physicx_matrix_use_r_c:nn
1426         #3 {{ \int_use:N #1 }} {{##1}} &
1427     }
1428 }
1429 \tl_put_right:Nx \l__physicx_matrix_body_tl
1430 {
1431     \exp_after:wN
1432     \physicx_matrix_use_r_c:nn
1433     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1434 }
1435 }

```

(End definition for \diagonalmatrix. This function is documented on page ??.)

__physicx_declare_init:

```

1436 \cs_new:Npn \__physicx_matrix_enhanced_init:
1437 {
1438     \seq_if_empty:NF \l__physicx_row_list_seq
1439     {
1440         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1441         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1442         { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1443     }
1444     \seq_if_empty:NF \l__physicx_col_list_seq
1445     {
1446         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1447         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1448         { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1449     }
1450 }

```

(End definition for __physicx_declare_init:.)

\commamatrix Define \commamatrix.

```

1451 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1452 {
1453     \group_begin:
1454     \keys_set:nn { physicx/matrix } {#3}
1455     \tl_if_empty:nF {#4}

```

```

1456     { \keys_set:nn { physicx/matrix } { array = {#4} } }
1457 \IfBooleanT {#1}
1458     { \keys_set:nn { physicx/matrix } { saveto = \physicx_tmp } }
1459 \tl_set:Nx \l__physicx_matrix_array_tl
1460     { \__physicx_expand:w \l__physicx_matrix_array_tl }
1461 \bool_lazy_or:nnTF
1462     { \bool_if_p:n {#2} }
1463     { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1464     { \__physicx_commamatrix_enhanced: }
1465     {
1466         \tl_replace_all:Nox \l__physicx_matrix_array_tl
1467         { \physicx@cr } { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1468         \tl_replace_all:Non \l__physicx_matrix_array_tl
1469         { \physicx@align } { & }
1470         \tl_set_eq:NN \l__physicx_matrix_body_tl
1471         \l__physicx_matrix_array_tl
1472     }
1473 \__physicx_matrix_save_or_print:
1474 \group_end:
1475 }
1476 \cs_new_nopar:Npn \__physicx_matrix_save_or_print:
1477 {
1478     \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1479     {
1480         \exp_after:wN \tl_gset_eq:NN
1481         \l__physicx_matrix_save_tl
1482         \l__physicx_matrix_body_tl
1483     }
1484     {
1485         \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1486         \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1487         \fi:
1488         \exp_after:wN \__physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matrix_body_tl
1489         \l__physicx_matrix_body_tl
1490         \__physicx_matrix_end: \l__physicx_matrix_after_end_tl
1491     }
1492 }
1493 \cs_new:Npn \__physicx_commamatrix_enhanced:
1494 {
1495     \tl_clear:N \l__physicx_matrix_body_tl
1496     \int_zero:N \l__physicx_tmpa_int
1497     \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1498     \l__physicx_matrix_array_tl
1499     \int_set:Nn \l__physicx_matrix_rows_int
1500     { \seq_count:N \l__physicx_tmp_seq }
1501     \__physicx_matrix_enhanced_init:
1502     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1503     {
1504         \seq_map_tokens:Nn \l__physicx_tmp_seq
1505         {
1506             \int_incr:N \l__physicx_tmpa_int
1507             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1508             \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_e:nnn
1509         }
1510     }
1511 }

```

```

1510     }
1511     {
1512         \seq_map_tokens:Nn \l__physicx_tmp_seq
1513         {
1514             \int_incr:N \l__physicx_tmpa_int
1515             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1516                 \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_ne:nnn
1517         }
1518     }
1519 }
1520 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1521 {
1522     \seq_set_split:Non \l__physicx_tmp_col_seq
1523     { \physicx@align } {#3}
1524     \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1525     \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1526     { #2 {##2} {#1} {##1} }
1527     \tl_put_right:Nx \l__physicx_matrix_body_tl
1528     {
1529         \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1530         \if_int_compare:w \l__physicx_matrix_rows_int = #1
1531             \scan_stop:
1532         \else:
1533             \[\dim_use:N \l__physicx_matrix_sep_dim]
1534         \fi:
1535     }
1536 }
1537 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1538 {
1539     \seq_put_right:Nx \l__physicx_tmp_coled_seq
1540     {
1541         \text_expand:n % \text_expand:n do the magic thing, but slower
1542         {
1543             \physicx@matricelement { #1 }
1544             { \__physicx_matrix_row_iterate:n {#2} }
1545             { \__physicx_matrix_col_iterate:n {#3} }
1546         }
1547     }
1548 }
1549 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1550 {
1551     \seq_put_right:No \l__physicx_tmp_coled_seq
1552     {
1553         \physicx@matricelement {#1}
1554         { \__physicx_matrix_row_iterate:n {#2} }
1555         { \__physicx_matrix_col_iterate:n {#3} }
1556     }
1557 }

```

(End definition for `\commamatrix`. This function is documented on page ??.)

`\generalmatrix` Define `\generalmatrix`.

```

1558 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1559 {

```

```

1560 \IfBooleanTF {#2}
1561 {
1562   \group_begin:
1563   \IfBooleanTF {#1}
1564     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicxtmp } }
1565     { \keys_set:nn { physicx/matrix } {#4} }
1566   \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1567   \physicx_construct:nnn
1568   {
1569     \tl_if_empty:NTF \l__physicx_matrix_main_tl
1570     {
1571       \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1572     }
1573     { \physicx_matrix_array_parse_main: }
1574   }
1575   { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1576   { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1577   \__physicx_generalmatrix:
1578   \__physicx_matrix_save_or_print:
1579   \group_end:
1580 }
1581 {
1582   \IfBooleanTF {#1}
1583     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1584     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1585   \qxmatrix = * [#4]
1586 }
1587 }
1588 \cs_new:Npn \__physicx_generalmatrix:
1589 {
1590   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1591   {
1592     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1593     \__physicx_matrix_appto_body_e:off
1594   }
1595   {
1596     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1597     \__physicx_matrix_appto_body_ne:off
1598   }
1599   \__physicx_matrix_transpose:N
1600   \__physicx_matrix_generate_body:NNNN
1601   \__physicx_generalmatrix_generate:nnn
1602 }

```

(End definition for \generalmatrix. This function is documented on page ??.)

__physicx_matrix_generate_body:NNNN

```

1603 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1604 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1605 {
1606   \__physicx_matrix_enhanced_init:
1607   \int_step_inline:nn { #1 - 1 }
1608   {
1609     \int_step_inline:nn { #2 - 1 }

```



```

1610     {
1611       \tl_set:Nx \l__physicx_tmp_tl
1612       {
1613         \exp_after:wN
1614         \physicx_matrix_use_r_c:nn
1615         #3 {{##1}} {{####1}}
1616       }
1617       #4 \l__physicx_tmp_tl {##1} {####1}
1618       \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1619     }
1620   \tl_set:Nx \l__physicx_tmp_tl
1621   {
1622     \exp_after:wN
1623     \physicx_matrix_use_r_c:nn
1624     #3 {{##1}} {{ \int_use:N #2 }}
1625   }
1626   #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1627   \tl_put_right:Nx \l__physicx_matrix_body_tl
1628   { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1629 }
1630 \int_step_inline:nn { #2 - 1 }
1631 {
1632   \tl_set:Nx \l__physicx_tmp_tl
1633   {
1634     \exp_after:wN
1635     \physicx_matrix_use_r_c:nn
1636     #3 {{ \int_use:N #1 }} {{##1}}
1637   }
1638   #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}
1639   \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1640 }
1641 \tl_set:Nx \l__physicx_tmp_tl
1642 {
1643   \exp_after:wN
1644   \physicx_matrix_use_r_c:nn
1645   #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1646 }
1647 #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1648 }

```

(End definition for __physicx_matrix_generate_body:NNNN.)

```

\__physicx_matrix_appto_body_e:nnn
\__physicx_matrix_appto_body_e:off
\__physicx_matrix_appto_body_e:xff
\__physicx_matrix_appto_body_ne:nnn
\__physicx_matrix_appto_body_ne:off
\__physicx_matrix_appto_body_ne:xff
1649 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1650 {
1651   \tl_put_right:Nx \l__physicx_matrix_body_tl
1652   {
1653     \text_expand:n
1654     {
1655       \physicx@matricelement {#1}
1656       { \__physicx_matrix_row_iterate:n {#2} }
1657       { \__physicx_matrix_col_iterate:n {#3} }
1658     }
1659   }

```

```

1660 }
1661 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1662 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1663 {
1664   \tl_put_right:No \l__physicx_matrix_body_tl
1665   {
1666     \physicx@matricelement {#1}
1667     { \__physicx_matrix_row_iterate:n {#2} }
1668     { \__physicx_matrix_col_iterate:n {#3} }
1669   }
1670 }
1671 \cs_generate_variant:Nn \__physicx_matrix_appto_body_ne:nnn { off, xff }

```

(End definition for __physicx_matrix_appto_body_e:nnn and __physicx_matrix_appto_body_ne:nnn.)

__physicx_matrix_transpose:N

```

1672 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1673 {
1674   \bool_if:NTF \l__physicx_matrix_transpose_bool
1675   {
1676     #1
1677     \l__physicx_matrix_cols_int
1678     \l__physicx_matrix_rows_int
1679     \use_ii_i:nn
1680   }
1681   {
1682     #1
1683     \l__physicx_matrix_rows_int
1684     \l__physicx_matrix_cols_int
1685     \use:nn
1686   }
1687 }

```

(End definition for __physicx_matrix_transpose:N.)

\physicx_construct:nnn Final construct. First is adi (array, diag, item), then ‘last-col’, ‘last-row’ and dots, then infinite, then ‘ending’ key.

```

1688 \cs_new:Npn \physicx_construct:nnn #1#2#3
1689 {
1690   \l__physicx_matrix_beginning_tl
1691   \__physicx_adi:nnn {#1} {#2} {#3}
1692   \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1693   {
1694     \int_incr:N \l__physicx_matrix_cols_int
1695     \__physicx_matrix_last_aux_c:
1696     \int_incr:N \l__physicx_matrix_cols_int
1697   }
1698   \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1699   {
1700     \int_incr:N \l__physicx_matrix_rows_int
1701     \__physicx_matrix_last_aux_r:
1702     \int_incr:N \l__physicx_matrix_rows_int
1703   }
1704   \bool_lazy_or:nnF

```

```

1705 { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1706 { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1707 {
1708   \physicx_matrix_set_r_c:nnn
1709   { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1710   { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1711   { \ddots }
1712 }
1713 \bool_if:NT \l__physicx_matrix_infinite_bool
1714 {
1715   \int_incr:N \l__physicx_matrix_rows_int
1716   \int_incr:N \l__physicx_matrix_cols_int
1717   \__physicx_matrix_last_aux_c:
1718   \__physicx_matrix_last_aux_r:
1719   \physicx_matrix_set_r_c:nnn
1720   { \int_use:N \l__physicx_matrix_rows_int }
1721   { \int_use:N \l__physicx_matrix_cols_int }
1722   { \ddots }
1723 }
1724 \l__physicx_matrix_ending_tl
1725 }
1726 \cs_new:Npn \__physicx_matrix_last_aux_c:
1727 {
1728   \int_step_inline:nn \l__physicx_matrix_rows_int
1729   {
1730     \physicx_matrix_set_r_c:nnn
1731     {##1} { \int_use:N \l__physicx_matrix_cols_int }
1732     { \cdots }
1733   }
1734 }
1735 \cs_new:Npn \__physicx_matrix_last_aux_r:
1736 {
1737   \int_step_inline:nn \l__physicx_matrix_cols_int
1738   {
1739     \physicx_matrix_set_r_c:nnn
1740     { \int_use:N \l__physicx_matrix_rows_int } {##1}
1741     { \vdots }
1742   }
1743 }

```

(End definition for `\physicx_construct:nnn`. This function is documented on page ??.)

1.3.3 Define new matrix command

```

\__physicx_new_matrix_cmd:NNN
  \newgeneralmatrix 1744 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
  \NewGeneralMatrix 1745 {
  \newdiagonalmatrix 1746   \NewDocumentCommand #2 { t+ m o o m m }
  \NewDiagonalMatrix 1747   {
  \newcommamatrix 1748     \IfBooleanTF {##1}
  \NewCommaMatrix 1749     {
1750       \IfNoValueTF {##3}
1751       { \newcommand ##2 { #1 + [##5] {##6} } }
1752       {

```

```

1753         \IfNoValueTF {##4}
1754         { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1755         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1756     }
1757 }
1758 {
1759     \IfNoValueTF {##3}
1760     { \newcommand ##2 { #1 [##5] {##6} } }
1761     {
1762         \IfNoValueTF {##4}
1763         { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1764         { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1765     }
1766 }
1767 }
1768 \NewDocumentCommand #3 { t+ m m m m }
1769 {
1770     \IfBooleanTF {##1}
1771     { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1772     { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1773 }
1774 }
1775 \_physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1776 \_physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1777 \NewDocumentCommand \newgeneralmatrix { t+ m o o m }
1778 {
1779     \IfBooleanTF {#1}
1780     {
1781         \IfNoValueTF {#3}
1782         { \newcommand #2 { \generalmatrix + {#5} } }
1783         {
1784             \IfNoValueTF {#4}
1785             { \newcommand #2 [3] { \generalmatrix + {#5} } }
1786             { \newcommand #2 [3] [4] { \generalmatrix + {#5} } }
1787         }
1788     }
1789     {
1790         \IfNoValueTF {#3}
1791         { \newcommand #2 { \generalmatrix {#5} } }
1792         {
1793             \IfNoValueTF {#4}
1794             { \newcommand #2 [3] { \generalmatrix {#5} } }
1795             { \newcommand #2 [3] [4] { \generalmatrix {#5} } }
1796         }
1797     }
1798 }
1799 \NewDocumentCommand \NewGeneralMatrix { t+ m m m m }
1800 {
1801     \IfBooleanTF {#1}
1802     { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1803     { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1804 }

```

(End definition for `_physicx_new_matrix_cmd:NNN` and others. These functions are documented on

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols

$\backslash +$	154
$\backslash -$	154
$\backslash \backslash$	44, 1067, 1074, 1081, 1105, 1417, 1467, 1533, 1628
$\backslash \{$	295, 525, 544, 562, 603, 628
$\backslash \}$	295, 525, 546, 562, 603, 628

A

$\backslash A$	149, 154
$\backslash \text{abs}$	<u>596</u>
$\backslash \text{absolutevalue}$	<u>596</u>
$\backslash \text{acomm}$	<u>596</u>
$\backslash \text{aftergroup}$	204
$\backslash \text{anticommutator}$	<u>596</u>

B

$\backslash \text{begin}$	300, 301, 302, 303, 304, 305, 306, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 545, 548, 552, 555, 556, 557, 562, 563, 567, 570, 571, 572, 924, 925, 926, 927, 928, 929, 930, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949
$\backslash \text{bgroup}$	203, 205
$\backslash \text{Big}$	578
$\backslash \text{big}$	578
$\backslash \text{Bigg}$	578
$\backslash \text{bigg}$	578
$\backslash \text{Biggl}$	585
$\backslash \text{biggl}$	584
$\backslash \text{Biggr}$	585
$\backslash \text{biggr}$	584
$\backslash \text{Bigl}$	583
$\backslash \text{bigl}$	582
$\backslash \text{Bigr}$	583
$\backslash \text{bigr}$	582
$\backslash \text{boldsymbol}$	193
bool commands:	
$\backslash \text{bool_case_false:n}$	495, 496, 503
$\backslash \text{bool_case_true:nTF}$	580, 1337

$\backslash \text{bool_if:NtF}$	16, 21, 26, 72, 182, 509, 511, 515, 999, 1054, 1061, 1071, 1079, 1091, 1098, 1102, 1109, 1113, 1377, 1502, 1590, 1674, 1713
$\backslash \text{bool_if_p:N}$	1375, 1463
$\backslash \text{bool_if_p:n}$	582, 583, 584, 585, 1374, 1462
$\backslash \text{bool_lazy_and_p:nn}$	1342, 1347
$\backslash \text{bool_lazy_or:nnTF}$	31, 113, 124, 348, 353, 1373, 1461, 1704
$\backslash \text{bool_new:N}$	9, 10, 11, 12, 13, 38, 53, 411, 413, 687, 688, 689, 697, 711, 719, 720
$\backslash \text{bool_set:Nn}$	422, 423, 1566
$\backslash \text{bool_set_false:N}$	82, 170, 229, 987, 1015, 1030, 1147
$\backslash \text{bool_set_inverse:N}$	757, 874
$\backslash \text{bool_set_true:N}$	116, 127, 137, 177, 225, 228, 986, 1019, 1023, 1034, 1038, 1141, 1440, 1446
$\backslash \text{c_false_bool}$	540, 560
$\backslash \text{c_true_bool}$	523, 533, 540, 560
$\backslash \text{Bqty}$	<u>596</u>
$\backslash \text{bqty}$	<u>596</u>

C

$\backslash \text{cdots}$	774, 1053, 1056, 1057, 1060, 1062, 1093, 1094, 1097, 1099, 1732
clist commands:	
$\backslash \text{clist_clear:N}$	955
$\backslash \text{clist_concat:NNN}$	962
$\backslash \text{clist_if_in:NnTF}$	1290
$\backslash \text{clist_map_break:n}$	163, 175
$\backslash \text{clist_map_inline:nn}$	69, 159, 171, 960
$\backslash \text{clist_new:N}$	695, 696
$\backslash \text{clist_put_right:Nn}$	836, 842, 1281
$\backslash \text{clist_set:Nn}$	952
$\backslash \text{clist_set_eq:NN}$	1257
$\backslash \text{c_empty_clist}$	1257
$\backslash \text{comm}$	626
$\backslash \text{commamatrix}$	29, 1451, 1776
$\backslash \text{commutator}$	<u>596</u>

cs commands:		else commands:	
<code>\cs:w</code>	32, 33, 865	<code>\else:</code>	652, 1065, 1192, 1202, 1227, 1239, 1532
<code>\cs_end:</code>	32, 33, 650, 865	<code>\end</code>	300, 301, 302, 303, 304, 305, 306, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 545, 548, 552, 555, 556, 557, 562, 563, 567, 570, 571, 572, 924, 925, 926, 927, 928, 929, 930, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949
<code>\cs_generate_variant:Nn</code>	3, 4, 5, 6, 76, 79, 185, 366, 506, 1125, 1263, 1320, 1661, 1671	<code>\eval</code>	596
<code>\cs_if_exist:NTF</code>	272	<code>\evaluated</code>	521, 609
<code>\cs_if_exist_p:N</code>	1343, 1348	exp commands:	
<code>\cs_new:Npn</code>	7, 111, 119, 122, 134, 145, 157, 191, 195, 199, 203, 204, 205, 206, 333, 345, 416, 426, 438, 475, 482, 507, 576, 722, 914, 916, 1118, 1126, 1154, 1167, 1178, 1188, 1214, 1253, 1255, 1264, 1265, 1299, 1321, 1395, 1400, 1436, 1493, 1520, 1537, 1549, 1588, 1604, 1649, 1662, 1672, 1688, 1726, 1735, 1744	<code>\exp_after:wN</code>	1408, 1415, 1424, 1431, 1478, 1480, 1488, 1613, 1622, 1634, 1643
<code>\cs_new_eq:NN</code>	121, 144, 681, 701, 710, 1186	<code>\exp_args:Nc</code>	385, 388
<code>\cs_new_nopar:Npn</code>	640, 648, 657, 659, 664, 670, 703, 704, 707, 708, 709, 1476	<code>\exp_args:Nf</code>	1162
<code>\cs_new_protected:Nn</code>	997	<code>\exp_args:Nnno</code>	1292, 1327
<code>\cs_new_protected:Npn</code>	54, 59, 64, 77, 80, 187	<code>\exp_args:Nno</code>	161, 173
<code>\cs_new_protected_nopar:Npn</code>	951, 953, 958	<code>\exp_args:NNx</code>	391
<code>\cs_set:Nn</code>	879, 880, 881, 882, 883, 884	<code>\exp_args:No</code>	792, 801, 903, 1274
<code>\cs_set:Npn</code>	371, 377, 381, 599, 609, 610, 626, 631, 632, 633, 763, 772, 781, 790, 799	<code>\exp_args:NW</code>	1507, 1515
<code>\cs_set:Npx</code>	806	<code>\exp_not:N</code>	8, 275, 393, 394, 395, 446, 806
<code>\cs_set_eq:NN</code>	56, 57, 61, 62, 193, 197, 201, 679, 732, 734, 736, 739, 745, 747, 761, 770, 779, 788, 797, 815, 818, 821, 824, 827, 1001, 1005, 1132, 1137, 1142, 1148, 1379, 1383, 1592, 1596	<code>\exp_not:n</code>	450, 451, 651, 653, 701, 732, 736
<code>\cs_set_nopar:Npn</code>	1441, 1447	<code>\exp_stop_f:</code>	8, 441, 448, 1190, 1193, 1216, 1228
<code>\cs_set_protected:Npn</code>	487, 500, 519, 594	F	
<code>\cs_to_str:N</code>	386, 389, 394, 395, 432	fi commands:	
D		<code>\fi:</code>	452, 467, 472, 473, 574, 635, 654, 1068, 1211, 1212, 1250, 1251, 1487, 1534
<code>\ddots</code>	774, 1080, 1111, 1711, 1722	fp commands:	
<code>\DeclareDocumentCommand</code>	492, 502, 578, 982, 1356, 1451, 1558	<code>\fp_eval:n</code>	27, 1340
<code>\diagonalmatrix</code>	28, 1356, 1775	<code>\fpeval</code>	27, 1348, 1349
dim commands:		G	
<code>\dim_new:N</code>	721	<code>\generalmatrix</code>	31, 1558, 1782, 1785, 1786, 1791, 1794, 1795, 1802, 1803
<code>\dim_use:N</code>	1067, 1074, 1081, 1105, 1417, 1467, 1533, 1628	group commands:	
E		<code>\group_begin:</code>	335, 984, 1358, 1453, 1562
<code>\egroup</code>	204, 206	<code>\group_end:</code>	343, 995, 1393, 1474, 1579
		I	
		if commands:	
		<code>\if_bool:N</code>	456, 468, 521, 596
		<code>\if_case:w</code>	448
		<code>\if_cs_exist:w</code>	650
		<code>\if_int_compare:w</code>	441, 1063, 1190, 1193, 1216, 1228, 1485, 1530

\physicx_declare_legacy_paren:NnnnNNn	\physicx_xquantity:nn	333
..... 575, 600, 601, 602, 603,	physicx internal commands:	
605, 612, 616, 620, 623, 627, 636, 638	__physicx_adi:nnn	722, 879, 880, 881, 882, 883, 884, 1691
\physicx_declare_legacy_quantity:nnNn	\l__physicx_begin_int	49,
..... 409, 522, 532, 539, 559	89, 91, 92, 94, 125, 126, 130, 136, 140	
\physicx_if_num:n	\l__physicx_cmd_arg_int	415, 421, 440, 441, 448
..... 147	\l__physicx_cmd_arg_spec_tl	414, 420, 433, 442
\physicx_if_num:nTF	\l__physicx_cmd_auto_body_bool	413, 423, 468, 511, 515
..... 903, 1011, 1026, 1334	\l__physicx_cmd_auto_body_tl	412, 419, 435, 469, 470
\physicx_if_num_sign:n	\l__physicx_cmd_noauto_body_bool	411, 422, 456, 509
..... 152	\l__physicx_cmd_noauto_body_tl	410, 418, 434, 457, 458
\physicx_left:	\l__physicx_col_list_seq	699, 753, 1444, 1448
203, 356, 490, 590	__physicx_commamatrix_enhanced:	1464, 1493
\physicx_left:N	__physicx_commamatrix_enhanced-	
..... 205, 358, 582, 583, 584, 585	aux:nNn	1507, 1515, 1520
\physicx_mathtools:	__physicx_commamatrix_enhanced-	
24	aux_e:nnn	1508, 1537
\physicx_mathtools:TF	__physicx_commamatrix_enhanced-	
..... 307, 931	aux_ne:nnn	1516, 1549
\physicx_matrix_array_parse:n	\g__physicx_compat_bool	11, 16, 209, 225
..... 1299, 1571	__physicx_declare_init:	1436
\physicx_matrix_array_parse-	__physicx_declare_init:nnn	416, 428
main:	__physicx_declare_legacy-	
1321, 1573	quantity_aux:nnn	438, 479
\physicx_matrix_diag_parse:n	__physicx_declare_legacy-	
..... 837, 1118, 1364, 1368, 1369, 1575	quantity_aux:NNnnn	431, 482, 506
\physicx_matrix_item_parse:n	__physicx_declare_legacy-	
..... 843, 1255, 1372, 1576	quantity_aux:nw	429, 475, 480
\physicx_matrix_new_type:nn	__physicx_diagonalmatrix_diag-	
..... 914	main:	1142, 1148, 1186, 1191
\physicx_matrix_new_type:nnn	__physicx_diagonalmatrix-	
..... 914	enhanced:nnn	1379, 1383, 1398
\physicx_matrix_set_r_c:nnn	__physicx_diagonalmatrix-	
681,	generate_body:NNN	1391, 1400
815, 818, 821, 824, 827, 1173, 1181,	__physicx_diagonalmatrix-	
1196, 1205, 1222, 1234, 1245, 1292,	generate_enhanced_body:NNN	1390, 1395
1316, 1327, 1708, 1719, 1730, 1739	__physicx_diagonalmatrix_no-	
\physicx_matrix_use_r_c:nn	diag:	1149, 1178, 1187
..... 648, 1409, 1416,	__physicx_diagonalmatrix_set-	
1425, 1432, 1614, 1623, 1635, 1644	diag:	1143, 1167
\physicx_new_type:nnn	\l__physicx_end_int	50,
..... 187, 293, 915, 917	99, 101, 102, 104, 126, 130, 136, 140	
\physicx_option_or:nn		
..... 29		
\physicx_option_or:nnTF		
..... 597, 607, 625, 629		
\c_physicx_Order_tl		
190, 234, 617, 637		
\c_physicx_order_tl		
189, 233, 621, 639		
\physicx_parse_range:nnN		
..... 77, 79, 1270, 1272		
\physicx_parse_range:nnnN		
..... 49		
\physicx_parse_range_check: .		
49, 847		
\physicx_parse_range_nocheck: 49, 848		
\physicx_qxmatrix:nnn		
..... 993, 997		
\physicx_right: ...		
204, 356, 490, 590		
\physicx_right:N		
..... 206, 360, 582, 583, 584, 585		
\physicx_search_also:nn		
..... 157, 168, 185, 186		
\physicx_search_also:nnTF		
..... 281, 895		
\physicx_short:		
19		
\physicx_use_amsymb_type: .		
191, 239		
\physicx_use_uni_bf_type:		
199		
\physicx_use_uni_bfit_type: 195, 238		


```

\__physicx_expand:w .... 701, 732,
734, 736, 739, 1158, 1269, 1303, 1460
\__physicx_generalmatrix: 1577, 1588
\__physicx_generalmatrix_-
generate:nnn .... 1592, 1596, 1601
\__physicx_if_can_num:n ..... 1332
\__physicx_if_keyval:nTF .. 145, 1367
\l__physicx_invalid_range_bool ..
..... 53, 72, 82, 116, 127, 137
\l__physicx_item_ignore_clist ...
..... 1257, 1281, 1290
\g__physicx_mathtools_bool .....
..... 9, 26, 228, 229
\__physicx_matrix_add_data:nn ...
..... 839, 845, 956, 958
\l__physicx_matrix_after_begin_-
tl ..... 717, 855, 857, 1488
\l__physicx_matrix_after_end_tl .
..... 718, 858, 860, 1490
\__physicx_matrix_appto_body_-
e:nnn ..... 1002, 1380, 1593, 1649
\__physicx_matrix_appto_body_-
ne:nnn ..... 1006, 1384, 1597, 1649
\l__physicx_matrix_args_tl .....
..... 716, 853, 854, 1488
\l__physicx_matrix_array_tl ....
..... 690, 729, 1459,
1460, 1466, 1468, 1471, 1498, 1571
\__physicx_matrix_autocalc:mn ...
..... 710, 745, 747, 1254, 1305, 1311
\__physicx_matrix_begin:w .....
..... 708, 850, 1488
\l__physicx_matrix_beginning_tl .
..... 723, 885, 887, 1690
\l__physicx_matrix_body_tl .....
..... 691, 1009, 1044,
1049, 1050, 1056, 1057, 1062, 1067,
1074, 1076, 1078, 1080, 1081, 1083,
1087, 1088, 1093, 1094, 1099, 1105,
1106, 1108, 1110, 1111, 1406, 1413,
1422, 1429, 1470, 1482, 1489, 1495,
1527, 1618, 1627, 1639, 1651, 1664
\__physicx_matrix_calc:nn .....
..... 640, 745, 1133
\__physicx_matrix_col_iterate:n .
707, 831, 1447, 1545, 1555, 1657, 1668
\l__physicx_matrix_cols_int ....
..... 644, 645,
693, 742, 989, 1028, 1031, 1047,
1077, 1085, 1107, 1114, 1115, 1169,
1172, 1176, 1224, 1236, 1247, 1272,
1325, 1485, 1486, 1677, 1684, 1694,
1696, 1710, 1716, 1721, 1731, 1737
\l__physicx_matrix_diag_bool ...
..... 697, 1141, 1147
\__physicx_matrix_diag_calc:nn ..
..... 1132, 1137, 1182,
1199, 1208, 1217, 1229, 1240, 1253
\l__physicx_matrix_diag_clist ...
..... 695, 834, 836, 1364, 1575
\__physicx_matrix_diag_parse_-
aux:n ..... 1121, 1126
\__physicx_matrix_diag_parse_-
aux:nn ..... 1122, 1154
\__physicx_matrix_diag_parse_-
aux_anti:n ..... 1162, 1214
\__physicx_matrix_diag_parse_-
aux_regu:n ..... 1165, 1188
\l__physicx_matrix_dotcol_bool ..
..... 689, 1030, 1034,
1038, 1054, 1079, 1091, 1109, 1113
\l__physicx_matrix_dotrow_bool ..
..... 688, 1015, 1019, 1023, 1071
\__physicx_matrix_element_-
aux:nnn ..... 761, 767,
770, 776, 779, 785, 788, 794, 797, 802
\__physicx_matrix_end: .... 851, 1490
\__physicx_matrix_end:w ..... 709
\l__physicx_matrix_ending_tl ...
..... 724, 888, 890, 1724
\l__physicx_matrix_enhanced_bool
..... 720, 871, 874, 1375, 1463
\__physicx_matrix_enhanced_init:
..... 1436, 1501, 1606
\l__physicx_matrix_expand_-
element_bool ..... 711,
810, 999, 1377, 1440, 1446, 1502, 1590
\__physicx_matrix_generate_-
body:NNNN ..... 1397, 1600, 1603
\l__physicx_matrix_infinite_bool
..... 687, 754, 757,
986, 987, 1061, 1098, 1102, 1566, 1713
\l__physicx_matrix_item_clist ...
..... 696, 840, 842, 1372, 1576
\__physicx_matrix_item_parse_-
aux:n ..... 1259, 1264
\__physicx_matrix_item_parse_-
aux:nn ..... 1260, 1265
\__physicx_matrix_last_aux_c: ...
..... 1695, 1717, 1726
\__physicx_matrix_last_aux_r: ...
..... 1701, 1718, 1735
\l__physicx_matrix_last_col_tl ..
..... 706, 833, 1692, 1706
\l__physicx_matrix_last_row_tl ..
..... 705, 832, 1698, 1705

```

\l__physicx_matrix_main_tl	__physicx_parse_range_single_-
. 694, 749, 1328, 1569	check:n 56, 111, 121
__physicx_matrix_row_iterate:n .	__physicx_parse_range_single_-
. 704, 830, 1441, 1544, 1554, 1656, 1667	nocheck:n 61, 119
\l__physicx_matrix_rows_int	\g__physicx_physics_bool 10
. 642, 643, 692, 741, 988,	\l__physicx_quantity_args_tl . . .
1013, 1016, 1041, 1063, 1175, 1270, 260, 261, 340
1323, 1499, 1530, 1678, 1683, 1700,	\l__physicx_quantity_code_tl . . .
1702, 1709, 1715, 1720, 1728, 1740 262, 337, 341
__physicx_matrix_save_or_print:	\l__physicx_quantity_left_size_-
. 994, 1392, 1473, 1476, 1578	tl 254, 349, 354, 358
\l__physicx_matrix_save_tl	\l__physicx_quantity_left_tl 252, 339
. 863, 865, 1478, 1481	\l__physicx_quantity_post_tl 251, 363
\l__physicx_matrix_sep_dim	\l__physicx_quantity_pre_tl 250, 347
. 721, 861, 1067, 1074,	\l__physicx_quantity_right_size_-
1081, 1105, 1417, 1467, 1533, 1628	tl 255, 350, 355, 360
__physicx_matrix_set_data:nn . . .	\l__physicx_quantity_right_tl . . .
. 838, 844, 953 253, 342
__physicx_matrix_set_r_c_-	__physicx_qxmatrix_appto_-
ckall:nnn 679, 828	body:nnn 1001, 1005,
__physicx_matrix_set_r_c_-	1045, 1051, 1058, 1084, 1089, 1095
ckep:nnn 664, 819	\g__physicx_reqty_bool
__physicx_matrix_set_r_c_- 13, 216, 521, 596
ckig:nnn 659, 822	\l__physicx_row_list_seq
__physicx_matrix_set_r_c_- 698, 751, 1438, 1442
ckigep:nnn 670, 680, 825	\g__physicx_short_bool . . . 12, 21, 211
__physicx_matrix_set_r_c_-	\l__physicx_tmp_col_seq . . 1522, 1525
nock:nnn 657, 682, 816	\l__physicx_tmp_coled_seq
\l__physicx_matrix_tmparr_c_sep 1524, 1529, 1539, 1551
. 1310, 1313, 1314	\l__physicx_tmp_colnum_seq
\l__physicx_matrix_tmparr_r_sep 1273, 1279, 1288
. 1304, 1306, 1308	\l__physicx_tmp_rownum_seq
__physicx_matrix_transpose:N 1271, 1277, 1286
. 1389, 1599, 1672	\l__physicx_tmp_seq
\l__physicx_matrix_transpose_- 1497, 1500, 1504, 1512
bool 719, 866, 1674	\l__physicx_tmp_tl
\l__physicx_max_int 443, 457, 469, 1611, 1617,
. 51, 68, 99, 102, 104, 114, 125	1620, 1626, 1632, 1638, 1641, 1647
\l__physicx_min_int	\l__physicx_tmpa_bool 38, 170, 177, 182
. 52, 67, 89, 92, 94, 115	\l__physicx_tmpa_int
__physicx_nauto_case:nnnn . 484, 507 39, 1496, 1506, 1508, 1514, 1516
__physicx_new_matrix_cmd:NNN . 1744	\l__physicx_tmpa_seq
__physicx_new_xquantity_aux:w 73, 83, 117, 120, 131, 141
. 371, 377, 381, 385, 388	\l__physicx_tmpa_tl
__physicx_parse_range_aux:n 71, 80 87, 88, 91, 97, 98, 101, 268, 269
__physicx_parse_range_range: . . .	\l__physicx_tmparr_tl
. 57, 62, 107, 144 1301, 1302, 1303, 1304
__physicx_parse_range_range_-	\l__physicx_tmpb_int 40
check: 57, 122, 144	\l__physicx_tmpb_seq 86, 87, 97
__physicx_parse_range_range_-	\l__physicx_tmpdiag_seq 1159, 1170,
nocheck: 62, 134	1180, 1183, 1184, 1194, 1200, 1201,
__physicx_parse_range_single:n .	1203, 1209, 1210, 1218, 1219, 1220,
. 56, 61, 109, 121	1230, 1231, 1232, 1241, 1242, 1243

\l__physicx_tmpdiag_tl	1156, 1157, 1158, 1159
\l__physicx_tmpitem_tl	1267, 1268, 1269, 1275, 1293
__physicx_xquantity_aux:nnnn	338, 345, 366
\physicxempty	653, 667, 675, 712, 713, 812
\physicxexcept	715, 783, 792, 801, 807, 809
\PHYSICXIGNORE	7, 8, 661, 672, 1275
\physicxset	240
\physicxtmp	409, 991, 1360, 1458, 1564
\poissonbracket	596
\pqty	596
prg commands:	
\prg_generate_conditional_-variant:Nnn	186
\prg_new_conditional:Npnn	14, 19, 24, 29, 147, 152, 168, 1332
\prg_replicate:nn	1077, 1107
\prg_return_false:	17, 22, 27, 35, 150, 155, 183, 1353
\prg_return_true:	17, 22, 27, 34, 150, 155, 183, 1335, 1352
\ProcessKeysPackageOptions	222
Q	
\qty	596
\quantity	521, 599
quark commands:	
\quark_if_recursion_tail_stop:n	477, 478
\q_recursion_stop	430
\q_recursion_tail	430
\qxmatrix	973, 1585
\qxqty	407
R	
\rangle	299, 529, 556, 571
regex commands:	
\regex_match:nnTF ..	149, 154, 792, 801
\RequirePackage	213, 214, 215
\rgroup	553, 568
\right	204, 330, 355, 546, 553, 556, 562, 563, 568, 570, 571, 572
\rVert	613
S	
\sb	703, 1044, 1049, 1056, 1083, 1087, 1093
scan commands:	
\scan_stop:	1064, 1531
seq commands:	
\c_empty_seq	66, 1524
\seq_clear:N	83
\seq_concat:NNN	73
\seq_count:N	1183, 1184, 1200, 1201, 1209, 1210, 1218, 1219, 1230, 1231, 1241, 1242, 1306, 1313, 1500
\seq_if_empty:NTF	1438, 1444
\seq_item:Nn	1442, 1448
\seq_map_indexed_inline:Nn	1170, 1180, 1194, 1203, 1220, 1232, 1243, 1308, 1314, 1525
\seq_map_inline:Nn	1277, 1279, 1286, 1288
\seq_map_tokens:Nn	1504, 1512
\seq_new:N	698, 699
\seq_pop_left:NN	87, 97
\seq_put_right:Nn	117, 120, 131, 141, 1539, 1551
\seq_set_eq:NN	66, 1524
\seq_set_split:Nnn	5, 86, 751, 753, 1159, 1304, 1310, 1497, 1522
\seq_use:Nn	1529
\setmatrixdata	951
\setmatrixtype	892, 914, 924, 925, 926, 927, 928, 929, 930, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949
\setquantitytype	265, 292, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325
\smallmatrixquantity	521
\smqty	596
str commands:	
\c_backslash_str	269
\str_case_e:nnTF	1128
\str_const:Nn	684, 685, 686
\str_if_eq:nnTF	462, 464
\symbf	201
\symbfite	197
T	
TeX and L ^A T _E X 2 _ε commands:	
\@declareparencmd	575
\@declarequantitycmd	409
\@ifpackageloaded	224, 227, 237
\@ifstar	393
\c@MaxMatrixCols	870, 1485, 1486
\physicx@align	684, 876, 1310, 1469, 1523
\physicx@cr ..	685, 875, 1304, 1467, 1497
\physicx@matricelement	703, 758, 762, 763, 771, 772, 780, 781, 789, 790, 798, 799, 806, 1543, 1553, 1655, 1666
\physicx@sep ..	686, 751, 753, 877, 1159

tex commands:	865, 1156, 1157, 1267, 1268, 1301, 1302, 1459, 1611, 1620, 1632, 1641
\tex_advance:D	1114, 1115
text commands:	\tl_set_eq:NN 233, 234, 254, 255, 1470
\text_expand:n	734, 1541, 1653
tl commands:	\tl_tail:N 272, 275
\c_empty_tl	257, 501
\tl_clear:N	418, 419, 420, 1009, 1495
\tl_const:Nn	189, 190
\tl_gset_eq:NN	1480
\tl_head:N	268
\tl_if_empty:NNTF	88, 98, 1569, 1692, 1698
\tl_if_empty:nTF	337, 666, 674, 765, 1365, 1455
\tl_if_empty_p:N	349, 350, 1705, 1706
\tl_if_eq:nnTF	661, 672, 1274
\tl_if_head_eq_charcode:nNTF	1160
\tl_if_head_eq_meaning_p:nN	1339, 1340, 1344, 1349
\tl_if_in:nnTF	84, 146, 774, 783
\tl_if_noval_p:n	446
\tl_new:N	409, 410, 412, 414, 690, 691, 694, 705, 706, 713, 715, 716, 717, 718, 723, 724
\tl_put_right:Nn	442, 457, 458, 469, 470, 809, 857, 860, 887, 890, 1044, 1049, 1050, 1056, 1057, 1062, 1067, 1074, 1076, 1078, 1080, 1081, 1083, 1087, 1088, 1093, 1094, 1099, 1105, 1106, 1108, 1110, 1111, 1406, 1413, 1422, 1429, 1527, 1618, 1627, 1639, 1651, 1664
\tl_replace_all:Nnn	6, 1466, 1468
\tl_set:Nn	260, 268, 337, 443, 658, 662, 667, 668, 675, 676, 853,
	865, 1156, 1157, 1267, 1268, 1301, 1302, 1459, 1611, 1620, 1632, 1641
	\tl_set_eq:NN 233, 234, 254, 255, 1470
	\tl_tail:N 272, 275
	\tl_tail:n 1163
token commands:	\token_if_cs:NNTF 1478
	\token_if_eq_charcode:NNTF 269
	\token_if_eq_meaning_p:NN 354, 355
	\TrimSpaces 292, 918
	\txqty 408
	U
use commands:	\use:n 270, 278, 739
	\use:nn 1603, 1685
	\use:nnnn 4
	\use_i:nn 485, 1271, 1388, 1584
	\use_i:nnn 1584
	\use_i_ii:nnn 1386, 1583
	\use_ii:nn 270, 485, 1273
	\use_ii_i:nn 1603, 1679
	\use_none:n 276
	\use_none:nn 710, 747, 1137
	V
\vdots	774, 1076, 1078, 1080, 1106, 1108, 1110, 1741
\Vert	298, 530, 572
\vert	297, 528, 535, 536, 537, 570, 602, 606
\vqty	596
	X
\xquantity	365
	Z
\Z	149, 154