

The `physicx` package

Wenjian Chern (Longaster*)

November 14, 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 \tl_new:N \l__physicx_quantity_args_tl
247 \tl_new:N \l__physicx_quantity_code_tl
248 \tl_new:N \l__physicx_quantity_left_size_tl
249 \tl_new:N \l__physicx_quantity_left_tl
250 \tl_new:N \l__physicx_quantity_post_tl
251 \tl_new:N \l__physicx_quantity_pre_tl
252 \tl_new:N \l__physicx_quantity_right_size_tl
253 \tl_new:N \l__physicx_quantity_right_tl
254 \keys_define:nn { physicx }
255   { quantity .code:n = \keys_set:nn { physicx/quantity } {#1} }
256 \keys_define:nn { physicx/quantity }
257   {
258     pre .tl_set:N = \l__physicx_quantity_pre_tl ,
259     post .tl_set:N = \l__physicx_quantity_post_tl ,
260     left .tl_set:N = \l__physicx_quantity_left_tl ,
261     right .tl_set:N = \l__physicx_quantity_right_tl ,
262     left-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_left_size_tl #1 } ,
263     right-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_right_size_tl #1 } ,
264     size .meta:n = { left-size = {#1} , right-size = {#1} } ,
265     noauto .meta:n = { left-size = \c_empty_tl , right-size = \c_empty_tl } ,
266     noauto .value_required:n = false ,
267     args .code:n =
268       \tl_set:Nn \l__physicx_quantity_args_tl { [#1] } ,
269     args* .tl_set:N = \l__physicx_quantity_args_tl ,
270     code .tl_set:N = \l__physicx_quantity_code_tl ,
271     type .multichoice: ,
272
273     settype .code:n = \setquantitytype #1 ,
274
275     unknown .code:n =
276       \tl_set:Nx \l__physicx_tmpa_tl { \tl_head:N \l_keys_key_str }
277       \token_if_eq_meaning:NNTF \l__physicx_tmpa_tl \c_backslash_str
278         { \use:n } { \use_ii:nn }
279       {
280         \cs_if_exist:CTF { \tl_tail:N \l_keys_key_str }
281           {
282             \keys_set:nx { physicx/quantity }
283               { size = \exp_not:c { \tl_tail:N \l_keys_key_str } }
284             \use_none:n
285           }
286         { \use:n }
287       }
288   {

```

```

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

```

```

\physicx_xquantity:nn
\newxquantity
\NewXQuantity

```

```

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

```



```

395     { \physicx_xquantity:nn { #4 , noauto } {#5} }
396 \exp_args:Nc \__physicx_new_xquantity_aux:w
397   { \cs_to_str:N #1~unstar }
398   { \physicx_xquantity:nn { #4 } {#5} }
399 \exp_args:NNx \newcommand #1
400   {
401     \exp_not:N \@ifstar
402     \exp_not:c { \cs_to_str:N #1~star }
403     \exp_not:c { \cs_to_str:N #1~unstar }
404   }
405 }
406 \NewDocumentCommand \NewXQuantity { m m m m }
407 {
408   \NewDocumentCommand #1 { s #2 }
409   {
410     \IfBooleanTF {##1}
411     { \physicx_xquantity:nn { #3 , noauto } {#4} }
412     { \physicx_xquantity:nn { #3 } {#4} }
413   }
414 }
415 \NewXQuantity \qxqty { 0{} m } { #2 } {#3}
416 \NewXQuantity \txqty { 0{p} 0{} m } { type={#2}, #3 } {#4}

```

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

1.2.2 Legacy quantity

```

\physicx_declare_legacy_quantity:nnNn
\@declarequantitycmd
417 \tl_new:N \physicx_tmp
418 \tl_new:N \l__physicx_cmd_noauto_body_tl
419 \bool_new:N \l__physicx_cmd_noauto_body_bool
420 \tl_new:N \l__physicx_cmd_auto_body_tl
421 \bool_new:N \l__physicx_cmd_auto_body_bool
422 \tl_new:N \l__physicx_cmd_arg_spec_tl
423 \int_new:N \l__physicx_cmd_arg_int
424 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
425 {
426   \tl_clear:N \l__physicx_cmd_noauto_body_tl
427   \tl_clear:N \l__physicx_cmd_auto_body_tl
428   \tl_clear:N \l__physicx_cmd_arg_spec_tl
429   \int_set:Nn \l__physicx_cmd_arg_int {#1}
430   \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
431   \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
432 }
433 % noauto, auto, cmd, body
434 \cs_new:Npn \physicx_declare_legacy_quantity:nnNn #1#2#3#4
435 {
436   \__physicx_declare_init:nnn { 3 } {#1} {#2}
437   \__physicx_declare_legacy_quantity_aux:nw #4
438   \q_recursion_tail \q_recursion_tail \q_recursion_stop
439   \__physicx_declare_legacy_quantity_aux:NcVVV
440   #3 { \cs_to_str:N #3 ~ body }
441   \l__physicx_cmd_arg_spec_tl

```

```

442     \l__physicx_cmd_noauto_body_tl
443     \l__physicx_cmd_auto_body_tl
444 }
445 % arg spec, pre, body to replace(start from #4), post
446 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
447 {
448     \int_incr:N \l__physicx_cmd_arg_int
449     \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
450         \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
451         \tl_set:Nx \l__physicx_tmp_tl
452         {
453             {
454                 \exp_not:N \tl_if_novalue_p:n
455                 {
456                     \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:
457                     \or: \or: \or:
458                     \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
459                     \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
460                     \fi:
461                 }
462             }
463         }
464     \if_bool:N \l__physicx_cmd_noauto_body_bool
465         \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
466         \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
467         {
468             {
469                 % if is '.', use none
470                 \str_if_eq:nnTF {#2} {.} {} {#2}
471                 #3
472                 \str_if_eq:nnTF {#4} {.} {} {#4}
473             }
474         }
475     \fi:
476     \if_bool:N \l__physicx_cmd_auto_body_bool
477         \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
478         \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
479         { { ##1 #2 #3 ##2 #4 } }
480     \fi:
481 }
482 }
483 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
484 {
485     \quark_if_recursion_tail_stop:n {#1}
486     \quark_if_recursion_tail_stop:n {#2}
487     \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
488     \__physicx_declare_legacy_quantity_aux:nw
489 }
490 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
491 {
492     \__physicx_nauto_case:nnnn
493     { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
494     {
495         \cs_set_protected:Npn #1

```

```

496     {
497         \peek_charcode_ignore_spaces:NTF \let
498         { #2 } { #2 [ \physicx_left: ] \physicx_right: }
499     }
500     \DeclareDocumentCommand #2 { 0{##2} m s #3 }
501     {
502         \IfBooleanTF { ##3 }
503         { \bool_case_false:n {#4} }
504         { \bool_case_false:n {#5} }
505     }
506 }
507 {
508     \cs_set_protected:Npn #1
509     { #2 \c_empty_tl \c_empty_tl }
510     \DeclareDocumentCommand #2 { m m s #3 }
511     { \bool_case_false:n {#4} }
512 }
513 }
514 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
515 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
516 {
517     \bool_if:NTF \l__physicx_cmd_noauto_body_bool
518     {
519         \bool_if:NTF \l__physicx_cmd_auto_body_bool
520         {#1} {#2}
521     }
522     {
523         \bool_if:NTF \l__physicx_cmd_auto_body_bool
524         {#3} {#4}
525     }
526 }
527 \cs_set_protected:Npn \@declarequantitycmd
528 { \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>	<p>Redefine some macros in physics package.</p> <pre> 529 \if_bool:N \g__physicx_reqty_bool 530 \physicx_declare_legacy_quantity:nnNn 531 \c_true_bool \c_true_bool \quantity 532 { 533 { !g } { { \{ } { #4 } { \} } } 534 { !o } { { [} { #5 } {] } } 535 { !d() } { { (} { #6 } {) } } 536 { !d } { { \vert } { #7 } { \vert } } 537 { !d<> } { { \langle } { #8 } { \rangle } } 538 { !d== } { { \Vert } { #9 } { \Vert } } 539 } 540 \physicx_declare_legacy_quantity:nnNn 541 \c_true_bool \c_true_bool \evaluated 542 { 543 { !g } { { . } { #4 \nobreak } { \vert } } 544 { !d[] } { { [] } { #5 \nobreak } { \vert } } </pre>
--	---

```

545     { !d(| } { { ( } { #6 \nobreak } { \vert } }
546   }
547 \physicsx_declare_legacy_quantity:nnNn
548 \c_true_bool \c_false_bool \matrixquantity
549 {
550   { !g }
551   {
552     { \IfBooleanT{#3}{\left\{ } }
553     { \begin{matrix} #4 \end{matrix} }
554     { \IfBooleanT{#3}{\right\}} }
555   }
556   { !o } { { \begin{bmatrix} } {#5} { \end{bmatrix} } }
557   { !d() }
558   {
559     { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
560     { \begin{matrix} #6 \end{matrix} }
561     { \IfBooleanTF{#3}{\right\rgroup}{\right)}} }
562   }
563   { !d|| } { { \begin{vmatrix} } {#7} { \end{vmatrix} } }
564   { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } { \right\rangle } }
565   { !d== } { { \begin{Vmatrix} } {#9} { \end{Vmatrix} } }
566 }
567 \physicsx_declare_legacy_quantity:nnNn
568 \c_true_bool \c_false_bool \smallmatrixquantity
569 {
570   { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } { \right\} } }
571   { !o } { { \left[ } { \begin{smallmatrix} #5 \end{smallmatrix} } { \right]} } }
572   { !d() }
573   {
574     { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
575     { \begin{smallmatrix} #6 \end{smallmatrix} }
576     { \IfBooleanTF{#3}{\right\rgroup}{\right)}} }
577   }
578   { !d|| } { { \left\vert } { \begin{smallmatrix} #7 \end{smallmatrix} } { \right\vert} }
579   { !d<> } { { \left\langle } { \begin{smallmatrix} #8 \end{smallmatrix} } { \right\rangle} }
580   { !d== } { { \left\Vert } { \begin{smallmatrix} #9 \end{smallmatrix} } { \right\Vert} } }
581 }
582 \fi:

```

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

```

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

```

```

595         {
596             \IfBooleanTF {##1}
597             { #4          #5 #3          #6 #7 }
598             { #4 \physicsx_left: #5 #3 \physicsx_right: #6 #7 }
599         }
600     }
601 }
602 \cs_set_protected:Npn \@declareparenccmd
603 { \physicsx_declare_legacy_paren:NnnnNNn }

```

(End definition for \physicsx_declare_legacy_paren:NnnnNNn and \@declareparenccmd. These functions are documented on page ??.)

```

\qty Redefine some macros in physics package.
\mqty 604 \if_bool:N \g__physicsx_reqty_bool
\smqty 605 \physicsx_option_or:nnT { compat } { short }
\pqty 606 {
\bqty 607     \cs_set:Npn \qty { \quantity }
\vqty 608     \physicsx_declare_legacy_paren:NnnnNNn \pqty { m } {#6} { } { } { } { }
\Bqty 609     \physicsx_declare_legacy_paren:NnnnNNn \bqty { m } {#6} { } { } [ ] { }
\absolutevalue 610     \physicsx_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } { } \vert \vert { }
\eval 611     \physicsx_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } { } \{ \} { }
\abs 612 }
\norm 613 \physicsx_declare_legacy_paren:NnnnNNn \absolutevalue
\order 614 { m } {#6} { } { } \vert \vert { }
\oorder 615 \physicsx_option_or:nnT { compat } { short }
\commutator 616 {
\poissonbracket 617     \cs_set:Npn \eval { \evaluated }
\pb 618     \cs_set:Npn \abs { \absolutevalue }
\anticommutator 619 }
\acomm 620 \physicsx_declare_legacy_paren:NnnnNNn \norm
621 { m } {#6} { } { } \lVert \rVert { }
622 \physicsx_compat:TF
623 {
624     \physicsx_declare_legacy_paren:NnnnNNn \order
625     { m } {#6} { } { } \cphysicsx_Order_tl { } { }
626 }
627 {
628     \physicsx_declare_legacy_paren:NnnnNNn \order
629     { m } {#6} { } { } \cphysicsx_order_tl { } { }
630 }
631 \physicsx_declare_legacy_paren:NnnnNNn \commutator
632 { m m } { #6 , #7 } { } { } [ ] { }
633 \physicsx_option_or:nnT { compat } { short }
634 { \cs_set:Npn \comm { \commutator } }
635 \physicsx_declare_legacy_paren:NnnnNNn \poissonbracket
636 { m m } { #6 , #7 } { } { } \{ \} { }
637 \physicsx_option_or:nnT { compat } { short }
638 {
639     \cs_set:Npn \pb { \poissonbracket }
640     \cs_set:Npn \anticommutator { \poissonbracket }
641     \cs_set:Npn \acomm { \poissonbracket }
642 }
643 \fi:

```

```

644 \physicx_declare_legacy_paren:NnnnNNn \0Order
645 { m } {#6} { \c_physicx_Order_tl } ( ) { }
646 \physicx_declare_legacy_paren:NnnnNNn \oorder
647 { m } {#6} { \c_physicx_order_tl } ( ) { }

```

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

1.3 Matrix things

1.3.1 Matrix auxillary functions

```

648 \cs_new_nopar:Npn \__physicx_matrix_calc:nn #1#2
649 {
650   \int_set:Nn \l__physicx_matrix_rows_int
651     { \int_max:nn {#1} \l__physicx_matrix_rows_int }
652   \int_set:Nn \l__physicx_matrix_cols_int
653     { \int_max:nn {#2} \l__physicx_matrix_cols_int }
654 }
655 % use matrix element
656 \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
657 {
658   \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
659     \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
660   \else:
661     \exp_not:o { \physicxempty }
662   \fi:
663 }
664 % set matrix element, check or not
665 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
666 { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
667 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
668 {
669   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
670   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
671 }
672 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_cke:nnn #1#2#3
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 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
679 {
680   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
681   {
682     \tl_if_empty:nTF {#3}
683       { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
684       { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
685   }
686 }
687 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
688   \__physicx_matrix_set_r_c_ckigep:nnn
689 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
690   \__physicx_matrix_set_r_c_nock:nnn
691 % align, cr, sep symbol

```

```

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

```

1.3.2 Matrix keys

```

733 \keys_define:nn { physicx }
734 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
735 \keys_define:nn { physicx/matrix }
736 {
737   array .tl_set:N = \l__physicx_matrix_array_tl ,
738   expand .choice: ,
739   expand / none .code:n =
740     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
741   expand / text-expand .code:n =
742     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
743   expand / f .code:n =

```

```

744 \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
745 expand / romanual .meta:n = { expand = f } ,
746 expand / x .code:n =
747 \cs_set_eq:NN \__physicx_expand:w \use:n ,
748 expand / edef .meta:n = { expand = x } ,
749 rows .int_set:N = \l__physicx_matrix_rows_int ,
750 cols .int_set:N = \l__physicx_matrix_cols_int ,
751 auto-update .choice: ,
752 auto-update / true .code:n =
753 \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,
754 auto-update / false .code:n =
755 \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
756 auto-update .default:n = true ,
757 main .tl_set:N = \l__physicx_matrix_main_tl ,
758 row-list .code:n =
759 \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,
760 col-list .code:n =
761 \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
762 infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
763 infinite .default:n = true ,
764 !infinite .code:n =
765 \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
766 element-code .cs_set:Np = \physicx@matricelement #1#2#3 ,
767 element-code* .choice: ,
768 element-code* / except-empty .code:n =
769 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
770 \physicx@matricelement
771 \cs_set:Npn \physicx@matricelement ##1##2##3
772 {
773 \tl_if_empty:nTF {##1}
774 {##1}
775 { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
776 } ,
777 element-code* / except-dots .code:n =
778 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
779 \physicx@matricelement
780 \cs_set:Npn \physicx@matricelement ##1##2##3
781 {
782 \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
783 {##1}
784 { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
785 } ,
786 element-code* / except-tl .code:n =
787 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
788 \physicx@matricelement
789 \cs_set:Npn \physicx@matricelement ##1##2##3
790 {
791 \tl_if_in:onTF { \physicxexcept } {##1}
792 {##1}
793 { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
794 } ,
795 element-code* / except-regex .code:n =
796 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
797 \physicx@matricelement

```



```

798 \cs_set:Npn \physicx@matricelement ##1##2##3
799 {
800   \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
801   {##1}
802   { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
803 } ,
804 element-code* / only-regex .code:n =
805 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
806 \physicx@matricelement
807 \cs_set:Npn \physicx@matricelement ##1##2##3
808 {
809   \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
810   { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
811   {##1}
812 } ,
813 element-code* / unknown .code:n =
814 \cs_set:Npx \physicx@matricelement { \exp_not:c {#1} },
815 element-except .tl_set:N = \physicxexcept ,
816 element-except+ .code:n =
817 \tl_put_right:Nn \physicxexcept {#1} ,
818 expand-element .bool_set:N = \l__physicx_matrix_expand_element_bool ,
819 expand-element .default:n = true ,
820 empty .tl_set:N = \physicxempty ,
821 check .choice: ,
822 check / none .code:n =
823 \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
824 \__physicx_matrix_set_r_c_nock:nnn ,
825 check / empty .code:n =
826 \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
827 \__physicx_matrix_set_r_c_ckepp:nnn ,
828 check / ignore .code:n =
829 \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
830 \__physicx_matrix_set_r_c_ckig:nnn ,
831 check / igep .code:n =
832 \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
833 \__physicx_matrix_set_r_c_ckigep:nnn ,
834 check / all .code:n =
835 \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
836 \__physicx_matrix_set_r_c_ckall:nnn ,
837 check .default:n = all ,
838 row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
839 col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
840 last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
841 last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
842 diag .clist_set:N = \l__physicx_matrix_diag_clist ,
843 diag+ .code:n =
844 \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
845 diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
846 diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
847 diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
848 item .clist_set:N = \l__physicx_matrix_item_clist ,
849 item+ .code:n =
850 \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
851 item-now .code:n = \physicx_matrix_item_parse:n {#1} ,

```

```

852 item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
853 item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
854 check-range .choice: ,
855 check-range / true .code:n = \physicx_parse_range_check: ,
856 check-range / false .code:n = \physicx_parse_range_nocheck: ,
857 check-range .default:n = true ,
858 begin .tl_set:N = \__physicx_matrix_begin:w ,
859 end .tl_set:N = \__physicx_matrix_end: ,
860 args .code:n =
861   \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,
862 args* .tl_set:N = \l__physicx_matrix_args_tl ,
863 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
864 after-begin+ .code:n =
865   { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
866 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
867 after-end+ .code:n =
868   { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
869 sepdim .dim_set:N = \l__physicx_matrix_sepdim ,
870 type .multichoice: ,
871 saveto .tl_set:N = \l__physicx_matrix_save_tl ,
872 saveto* .code:n =
873   \tl_set:Nn \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
874 transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
875 transpose .default:n = true ,
876 ' .meta:n = { transpose = true } ,
877 T .meta:n = { transpose = true } ,
878 MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
879 enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
880 enhanced .default:n = true ,
881 !enhanced .code:n =
882   \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
883 cr .tl_set:N = \physicx@cr ,
884 align .tl_set:N = \physicx@align ,
885 sep .tl_set:N = \physicx@sep ,
886 adi-order .choice: ,
887 adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
888 adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
889 adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
890 adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
891 adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
892 adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
893 beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
894 beginning+ .code:n =
895   \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
896 ending .tl_set:N = \l__physicx_matrix_ending_tl ,
897 ending+ .code:n =
898   \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
899
900 settype .code:n = \setmatrixtype #1 ,
901
902 unknown .code:n =
903   \physicx_search_also:nnF
904   {
905     physicx/matrix/type ,

```

```

906         physicx/matrix/expand ,
907         physicx/matrix/element-code* ,
908     }
909     {#1}
910     {
911         \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
912         {
913             \keys_set:nx { physicx/matrix }
914             { MaxMatrixCols = \l_keys_key_str }
915         }
916         {
917             \msg_error:nxx { physicx } { unknown-key }
918             \l_keys_path_str { physicx/matrix }
919         }
920     } ,
921 }

```

`\physicx_matrix_new_type:nnn`

`\physicx_matrix_new_type:nn`

`\setmatrixtype`

```

922 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
923 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
924 \cs_new:Npn \physicx_matrix_new_type:nn
925 { \physicx_new_type:nnn { matrix } }
926 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
927 {
928     \IfBooleanTF {#1}
929     { \physicx_matrix_new_type:nn {#2} }
930     { \physicx_matrix_new_type:nnn {#2} }
931 }

```

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

A few types.

```

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

```

```

954 \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
955 \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
956 \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
957 \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
958 }

```

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

```

959 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
960 { \clist_set:cn { physicx@ #1 data@ #2 } }
961 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
962 {
963   \clist_clear:c { l__physicx_matrix_ #1 _clist }
964   \__physicx_matrix_add_data:nn {#1} {#2}
965 }
966 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
967 {
968   \clist_map_inline:nn {#2}
969   {
970     \clist_concat:ccc
971     { l__physicx_matrix_ #1 _clist }
972     { l__physicx_matrix_ #1 _clist }
973     { physicx@ #1 data@ #2 }
974   }
975 }

```

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

Initial settings.

```

976 \keys_set:nn { physicx/matrix }
977 {
978   type = m ,
979   saveto = ? ,
980 }

```

`\qxmatrix`

```

981 %% basicly, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-xmatnm-in-the-physics-package, but changed some
982 % #1 = boolean, saveto matrix
983 % #2 = star, infinite
984 % #3 = options
985 % #4 = letter for the entries
986 % #5 = number of rows
987 % #6 = number of explicit rows, default = 3
988 % #7 = number of columns
989 % #8 = number of explicit columns, default = 3
990 \DeclareDocumentCommand \qxmatrix { t= s O{type=p} m m O{3} m O{3} }
991 {
992   \group_begin:
993   \IfBooleanTF { #2 }
994   { \bool_set_true:N \l__physicx_matrix_infinite_bool }
995   { \bool_set_false:N \l__physicx_matrix_infinite_bool }
996   \int_set:Nn \l__physicx_matrix_rows_int {#6}
997   \int_set:Nn \l__physicx_matrix_cols_int {#8}
998   \IfBooleanTF {#1}
999   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }

```

```

1000     { \keys_set:nn { physicx/matrix } {#3} }
1001     \physicx_qxmatrix:nnn {#4} {#5} {#7}
1002     \__physicx_matrix_save_or_print:
1003     \group_end:
1004 }
1005 \cs_new_protected:Nn \physicx_qxmatrix:nnn
1006 {
1007     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1008     {
1009         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1010         \__physicx_matrix_appto_body_e:nnn
1011     }
1012     {
1013         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1014         \__physicx_matrix_appto_body_ne:nnn
1015     }
1016     % clear the variable containing the body of the matrix
1017     \tl_clear:N \l__physicx_matrix_body_tl
1018     % set the tentative number of explicit rows
1019     \physicx_if_num:nTF { #2 }
1020     {% number of rows is an integer
1021         \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
1022         {% if #2 <= rows, we don't want a row of dots
1023             \bool_set_false:N \l__physicx_matrix_dotrow_bool
1024             \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1025         }
1026         {% we want a row of dots
1027             \bool_set_true:N \l__physicx_matrix_dotrow_bool
1028         }
1029     }
1030     {% number of rows is symbolic, we want a row of dots
1031         \bool_set_true:N \l__physicx_matrix_dotrow_bool
1032     }
1033     % set the tentative number of explicit columns
1034     \physicx_if_num:nTF { #3 }
1035     {% number of cols is an integer
1036         \int_compare:nTF { #3 <= \l__physicx_matrix_cols_int }
1037         {% if #3 <= cols, we don't want a column of dots
1038             \bool_set_false:N \l__physicx_matrix_dotcol_bool
1039             \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1040         }
1041         {% we want a column of dots
1042             \bool_set_true:N \l__physicx_matrix_dotcol_bool
1043         }
1044     }
1045     {% number of columns is symbolic, we want a column of dots
1046         \bool_set_true:N \l__physicx_matrix_dotcol_bool
1047     }
1048     % loop through the rows
1049     \int_step_inline:nn { \l__physicx_matrix_rows_int }
1050     {
1051         % add the first entry in the row
1052         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1053         \__physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }

```

```

1054 % add the further entries in the explicit columns
1055 \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1056 {
1057     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
1058     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1059     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
1060 }
1061 % if we have a column of dots, add \cdots and the last entry
1062 \bool_if:NT \l__physicx_matrix_dotcol_bool
1063 {
1064     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{##1 #3} }
1065     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1066     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {#3}
1067 }
1068 % infinite matrix, add \cdots
1069 \bool_if:NT \l__physicx_matrix_infinite_bool
1070 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1071 \if_int_compare:w ##1 = \l__physicx_matrix_rows_int
1072 \scan_stop:
1073 \else:
1074     % finish up the row
1075     \tl_put_right:Nx \l__physicx_matrix_body_tl { \\[\dim_use:N \l__physicx_matrix_sep_d
1076 \fi:
1077 }
1078 % finish up the rows
1079 \bool_if:NT \l__physicx_matrix_dotrow_bool
1080 {
1081     % finish up the row
1082     \tl_put_right:Nx \l__physicx_matrix_body_tl { \\[\dim_use:N \l__physicx_matrix_sep_d
1083     % if we have a row of dots, fill it in
1084     \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1085     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1086     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1087     \bool_if:NT \l__physicx_matrix_dotcol_bool
1088     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
1089     \tl_put_right:Nx \l__physicx_matrix_body_tl { \\[\dim_use:N \l__physicx_matrix_sep_d
1090     % fill the last row
1091     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
1092     \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
1093     \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1094     {
1095         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
1096         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1097         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
1098     }
1099     \bool_if:NT \l__physicx_matrix_dotcol_bool
1100     {
1101         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
1102         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1103         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
1104     }
1105     % if the matrix is infinite, add a further column with \cdots
1106     \bool_if:NT \l__physicx_matrix_infinite_bool
1107     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }

```

```

1108     }
1109     % if the matrix is infinite, add a final row
1110     \bool_if:NT \l__physicx_matrix_infinite_bool
1111     {
1112         % finish up the row
1113         \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1114         \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1115         \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1116         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1117         \bool_if:NT \l__physicx_matrix_dotcol_bool
1118         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1119         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1120         % update cols
1121         \bool_if:NTF \l__physicx_matrix_dotcol_bool
1122         { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1123         { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1124     }
1125 }

```

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

\physicx_matrix_diag_parse:n Parse 'diag...' keys.

```

\physicx_matrix_diag_parse:o
1126 \cs_new:Npn \physicx_matrix_diag_parse:n #1
1127 {
1128     \keyval_parse:nnn
1129     \__physicx_matrix_diag_parse_aux:n
1130     \__physicx_matrix_diag_parse_aux:nn
1131     {#1}
1132 }
1133 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
1134 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
1135 {
1136     \str_case_e:nnF {#1}
1137     {
1138         { auto-update }
1139         {
1140             \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
1141             \__physicx_matrix_calc:nn
1142         }
1143         { noauto-update }
1144         {
1145             \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
1146         }
1147         { true }
1148         {
1149             \bool_set_true:N \l__physicx_matrix_diag_bool
1150             \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1151             \__physicx_diagonalmatrix_set_diag:
1152         }
1153         { false }
1154         {
1155             \bool_set_false:N \l__physicx_matrix_diag_bool
1156             \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1157             \__physicx_diagonalmatrix_no_diag:

```

```

1158     }
1159   }
1160   { \msg_error:nnn { physicx } { diag-key } {#1} }
1161 }
1162 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
1163 {
1164   \tl_set:Nn \l__physicx_tmpdiag_tl {#2}
1165   \tl_set:Nx \l__physicx_tmpdiag_tl
1166     { \__physicx_expand:w \l__physicx_tmpdiag_tl }
1167   \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl
1168   \tl_if_head_eq_charcode:nNTF {#1} '
1169   {
1170     \exp_args:Nf \__physicx_matrix_diag_parse_aux_anti:n
1171       { \tl_tail:n {#1} }
1172   }
1173   { \__physicx_matrix_diag_parse_aux_regu:n {#1} }
1174 }
1175 \cs_new:Npn \__physicx_diagonalmatrix_set_diag:
1176 {
1177   \int_zero:N \l__physicx_matrix_cols_int
1178   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1179   {
1180     \int_incr:N \l__physicx_matrix_cols_int
1181     \physicx_matrix_set_r_c:nnn {##1} {##1} {##2}
1182   }
1183   \int_set_eq:NN \l__physicx_matrix_rows_int
1184   \l__physicx_matrix_cols_int
1185 }
1186 \cs_new:Npn \__physicx_diagonalmatrix_no_diag:
1187 {
1188   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1189     { \physicx_matrix_set_r_c:nnn {##1} {##1} {##2} }
1190   \__physicx_matrix_diag_calc:nn
1191     { \seq_count:N \l__physicx_tmpdiag_seq }
1192     { \seq_count:N \l__physicx_tmpdiag_seq }
1193 }
1194 \cs_new_eq:NN \__physicx_diagonalmatrix_diag_main:
1195   \__physicx_diagonalmatrix_no_diag:
1196 \cs_new:Npn \__physicx_matrix_diag_parse_aux_regu:n #1
1197 {
1198   \if_int_compare:w #1 = 0 \exp_stop_f:
1199     \__physicx_diagonalmatrix_diag_main:
1200   \else:
1201     \if_int_compare:w #1 > 0 \exp_stop_f:
1202       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1203       {
1204         \physicx_matrix_set_r_c:nnn
1205           {##1} { \int_eval:n { ##1 + #1 } } {##2}
1206       }
1207       \__physicx_matrix_diag_calc:nn
1208         { \seq_count:N \l__physicx_tmpdiag_seq }
1209         { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1210     \else:
1211       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq

```



```

1212         {
1213             \physicx_matrix_set_r_c:nnn
1214             { \int_eval:n { ##1 - #1 } } {##1} {##2}
1215         }
1216         \__physicx_matrix_diag_calc:nn
1217         { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1218         { \seq_count:N \l__physicx_tmpdiag_seq }
1219     \fi:
1220 \fi:
1221 }
1222 \cs_new:Npn \__physicx_matrix_diag_parse_aux_anti:n #1
1223 {
1224     \if_int_compare:w #1 = 0 \exp_stop_f:
1225         \__physicx_matrix_diag_calc:nn
1226         { \seq_count:N \l__physicx_tmpdiag_seq }
1227         { \seq_count:N \l__physicx_tmpdiag_seq }
1228         \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1229         {
1230             \physicx_matrix_set_r_c:nnn
1231             {##1}
1232             { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1233             {##2}
1234         }
1235     \else:
1236         \if_int_compare:w #1 > 0 \exp_stop_f:
1237             \__physicx_matrix_diag_calc:nn
1238             { \seq_count:N \l__physicx_tmpdiag_seq }
1239             { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1240             \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1241             {
1242                 \physicx_matrix_set_r_c:nnn
1243                 {##1}
1244                 { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1245                 {##2}
1246             }
1247         \else:
1248             \__physicx_matrix_diag_calc:nn
1249             { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1250             { \seq_count:N \l__physicx_tmpdiag_seq }
1251             \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1252             {
1253                 \physicx_matrix_set_r_c:nnn
1254                 { \int_eval:n { ##1 - #1 } }
1255                 { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1256                 {##2}
1257             }
1258         \fi:
1259     \fi:
1260 }
1261 \cs_new:Npn \__physicx_matrix_diag_calc:nn
1262 { \__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
1263 \cs_new:Npn \physicx_matrix_item_parse:n #1
1264 {
1265     \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1266     \keyval_parse:NNn
1267         \__physicx_matrix_item_parse_aux:n
1268         \__physicx_matrix_item_parse_aux:nn
1269     {#1}
1270 }
1271 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }
1272 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1273 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2
1274 {
1275     \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1276     \tl_set:Nx \l__physicx_tmpitem_tl
1277         { \__physicx_expand:w \l__physicx_tmpitem_tl }
1278     \physicx_parse_range:nxN \l__physicx_matrix_rows_int
1279         { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1280     \physicx_parse_range:nxN \l__physicx_matrix_cols_int
1281         { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1282     \exp_args:No \tl_if_eq:nnTF
1283         { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1284     {
1285         \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1286         {
1287             \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1288             {
1289                 \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1290             }
1291         }
1292     }
1293     {
1294         \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1295         {
1296             \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1297             {
1298                 \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }
1299                 {
1300                     \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1301                         {##1} {####1} { \l__physicx_tmpitem_tl }
1302                 }
1303             }
1304         }
1305     }
1306 }

```

(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
1307 \cs_new:Npn \physicx_matrix_array_parse:n #1
1308 {
1309     \tl_set:Nn \l__physicx_tmparr_tl {#1}
1310     \tl_set:Nx \l__physicx_tmparr_tl
1311         { \__physicx_expand:w \l__physicx_tmparr_tl }

```

```

1312 \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl
1313 \__physicx_matrix_autocalc:nn
1314 { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1315 { 0 }
1316 \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1317 {
1318   \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1319   \__physicx_matrix_autocalc:nn
1320   { 0 }
1321   { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1322   \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1323   {
1324     \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1325   }
1326 }
1327 }
1328 \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.

```

1329 \cs_new:Npn \physicx_matrix_array_parse_main:
1330 {
1331   \int_step_inline:nn \l__physicx_matrix_rows_int
1332   {
1333     \int_step_inline:nn \l__physicx_matrix_cols_int
1334     {
1335       \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1336       {##1} {####1} \l__physicx_matrix_main_tl
1337     }
1338   }
1339 }

```

(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).

```

1340 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1341 {
1342   \physicx_if_num:nTF {#1}
1343   { \prg_return_true: }
1344   {
1345     \bool_case_true:nTF
1346     {
1347       { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1348       { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1349       {
1350         \bool_lazy_and_p:nn
1351         { \cs_if_exist_p:N \inteval }
1352         { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1353       } { }
1354       {
1355         \bool_lazy_and_p:nn
1356         { \cs_if_exist_p:N \fpeval }

```

```

1357         { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1358     } { }
1359 }
1360 { \prg_return_true: }
1361 { \prg_return_false: }
1362 }
1363 }

```

(End definition for `_physicx_if_can_num:n`.)

`\diagonalmatrix` Define `\diagonalmatrix`.

```

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

```

```

1407 }
1408 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1409 {
1410   \int_step_inline:nn { #1 - 1 }
1411   {
1412     \int_step_inline:nn { #2 - 1 }
1413     {
1414       \tl_put_right:Nx \l__physicx_matrix_body_tl
1415       {
1416         \exp_after:wN
1417         \physicx_matrix_use_r_c:nn
1418         #3 {{##1}} {{####1}} &
1419       }
1420     }
1421   \tl_put_right:Nx \l__physicx_matrix_body_tl
1422   {
1423     \exp_after:wN
1424     \physicx_matrix_use_r_c:nn
1425     #3 {{##1}} {{ \int_use:N #2 }} \[\dim_use:N \l__physicx_matrix_sep_dim]
1426   }
1427 }
1428 \int_step_inline:nn { #2 - 1 }
1429 {
1430   \tl_put_right:Nx \l__physicx_matrix_body_tl
1431   {
1432     \exp_after:wN
1433     \physicx_matrix_use_r_c:nn
1434     #3 {{ \int_use:N #1 }} {{##1}} &
1435   }
1436 }
1437 \tl_put_right:Nx \l__physicx_matrix_body_tl
1438 {
1439   \exp_after:wN
1440   \physicx_matrix_use_r_c:nn
1441   #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1442 }
1443 }

```

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

`__physicx_declare_init:`

```

1444 \cs_new:Npn \__physicx_matrix_enhanced_init:
1445 {
1446   \seq_if_empty:NF \l__physicx_row_list_seq
1447   {
1448     \bool_set_true:N \l__physicx_matrix_expand_element_bool
1449     \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1450     { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1451   }
1452   \seq_if_empty:NF \l__physicx_col_list_seq
1453   {
1454     \bool_set_true:N \l__physicx_matrix_expand_element_bool
1455     \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1456     { \seq_item:Nn \l__physicx_col_list_seq {##1} }

```

```

1457     }
1458 }

```

(End definition for _physicx_declare_init:.)

\commamatrix Define \commamatrix.

```

1459 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1460 {
1461   \group_begin:
1462   \keys_set:nn { physicx/matrix } {#3}
1463   \tl_if_empty:nF {#4}
1464   { \keys_set:nn { physicx/matrix } { array = {#4} } }
1465   \IfBooleanT {#1}
1466   { \keys_set:nn { physicx/matrix } { saveto = \physicx_tmp } }
1467   \tl_set:Nx \l__physicx_matrix_array_tl
1468   { \_physicx_expand:w \l__physicx_matrix_array_tl }
1469   \bool_lazy_or:nnTF
1470   { \bool_if_p:n {#2} }
1471   { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1472   { \_physicx_commamatrix_enhanced: }
1473   {
1474     \tl_replace_all:Nox \l__physicx_matrix_array_tl
1475     { \physicx@cr } { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1476     \tl_replace_all:Non \l__physicx_matrix_array_tl
1477     { \physicx@align } { & }
1478     \tl_set_eq:NN \l__physicx_matrix_body_tl
1479     \l__physicx_matrix_array_tl
1480   }
1481   \_physicx_matrix_save_or_print:
1482   \group_end:
1483 }
1484 \cs_new_nopar:Npn \_physicx_matrix_save_or_print:
1485 {
1486   \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1487   {
1488     \exp_after:wN \tl_gset_eq:NN
1489     \l__physicx_matrix_save_tl
1490     \l__physicx_matrix_body_tl
1491   }
1492   {
1493     \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1494     \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1495     \fi:
1496     \exp_after:wN \_physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matrix_body_tl
1497     \_physicx_matrix_end: \l__physicx_matrix_after_end_tl
1498   }
1499 }
1500 }
1501 \cs_new:Npn \_physicx_commamatrix_enhanced:
1502 {
1503   \tl_clear:N \l__physicx_matrix_body_tl
1504   \int_zero:N \l__physicx_tmpa_int
1505   \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1506   \l__physicx_matrix_array_tl

```

```

1507 \int_set:Nn \l__physicx_matrix_rows_int
1508 { \seq_count:N \l__physicx_tmp_seq }
1509 \__physicx_matrix_enhanced_init:
1510 \bool_if:NTF \l__physicx_matrix_expand_element_bool
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_e:nnn
1517   }
1518 }
1519 {
1520   \seq_map_tokens:Nn \l__physicx_tmp_seq
1521   {
1522     \int_incr:N \l__physicx_tmpa_int
1523     \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1524     \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_ne:nnn
1525   }
1526 }
1527 }
1528 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1529 {
1530   \seq_set_split:Non \l__physicx_tmp_col_seq
1531   { \physicx@align } {#3}
1532   \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1533   \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1534   { #2 {##2} {#1} {##1} }
1535   \tl_put_right:Nx \l__physicx_matrix_body_tl
1536   {
1537     \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1538     \if_int_compare:w \l__physicx_matrix_rows_int = #1
1539     \scan_stop:
1540     \else:
1541     \[\dim_use:N \l__physicx_matrix_sep_dim]
1542     \fi:
1543   }
1544 }
1545 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1546 {
1547   \seq_put_right:Nx \l__physicx_tmp_coled_seq
1548   {
1549     \text_expand:n % \text_expand:n do the magic thing, but slower
1550     {
1551       \physicx@matricelement { #1 }
1552       { \__physicx_matrix_row_iterate:n {#2} }
1553       { \__physicx_matrix_col_iterate:n {#3} }
1554     }
1555   }
1556 }
1557 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1558 {
1559   \seq_put_right:No \l__physicx_tmp_coled_seq
1560   {

```

```

1561     \physicx@matricelement {#1}
1562     { \__physicx_matrix_row_iterate:n {#2} }
1563     { \__physicx_matrix_col_iterate:n {#3} }
1564   }
1565 }

```

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

\generalmatrix Define \generalmatrix.

```

1566 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1567 {
1568   \IfBooleanTF {#2}
1569   {
1570     \group_begin:
1571     \IfBooleanTF {#1}
1572     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicxtmp } }
1573     { \keys_set:nn { physicx/matrix } {#4} }
1574     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1575     \physicx_construct:nnn
1576     {
1577       \tl_if_empty:NTF \l__physicx_matrix_main_tl
1578       {
1579         \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1580       }
1581       { \physicx_matrix_array_parse_main: }
1582     }
1583     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1584     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1585     \__physicx_generalmatrix:
1586     \__physicx_matrix_save_or_print:
1587     \group_end:
1588   }
1589   {
1590     \IfBooleanTF {#1}
1591     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1592     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1593     \qxmatrix = * [#4]
1594   }
1595 }
1596 \cs_new:Npn \__physicx_generalmatrix:
1597 {
1598   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1599   {
1600     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1601     \__physicx_matrix_appto_body_e:off
1602   }
1603   {
1604     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1605     \__physicx_matrix_appto_body_ne:off
1606   }
1607   \__physicx_matrix_transpose:N
1608   \__physicx_matrix_generate_body:NNNN
1609   \__physicx_generalmatrix_generate:nnn
1610 }

```


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

_physicx_matrix_generate_body:NNNN

```

1611 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1612 \cs_new:Npn \_physicx_matrix_generate_body:NNNN #1#2#3#4
1613 {
1614   \_physicx_matrix_enhanced_init:
1615   \int_step_inline:nn { #1 - 1 }
1616   {
1617     \int_step_inline:nn { #2 - 1 }
1618     {
1619       \tl_set:Nx \l__physicx_tmp_tl
1620       {
1621         \exp_after:wN
1622         \physicx_matrix_use_r_c:nn
1623         #3 {{##1}} {{####1}}
1624       }
1625       #4 \l__physicx_tmp_tl {##1} {####1}
1626       \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1627     }
1628     \tl_set:Nx \l__physicx_tmp_tl
1629     {
1630       \exp_after:wN
1631       \physicx_matrix_use_r_c:nn
1632       #3 {{##1}} {{ \int_use:N #2 }}
1633     }
1634     #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1635     \tl_put_right:Nx \l__physicx_matrix_body_tl
1636     { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1637   }
1638   \int_step_inline:nn { #2 - 1 }
1639   {
1640     \tl_set:Nx \l__physicx_tmp_tl
1641     {
1642       \exp_after:wN
1643       \physicx_matrix_use_r_c:nn
1644       #3 {{ \int_use:N #1 }} {{##1}}
1645     }
1646     #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}
1647     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1648   }
1649   \tl_set:Nx \l__physicx_tmp_tl
1650   {
1651     \exp_after:wN
1652     \physicx_matrix_use_r_c:nn
1653     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1654   }
1655   #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1656 }

```

(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

1657 \cs_new:Npn _physicx_matrix_appto_body_e:nnn #1#2#3

```

1658 {
1659   \tl_put_right:Nx \l__physicx_matrix_body_tl
1660   {
1661     \text_expand:n
1662     {
1663       \physicx@matricelement {#1}
1664       { \__physicx_matrix_row_iterate:n {#2} }
1665       { \__physicx_matrix_col_iterate:n {#3} }
1666     }
1667   }
1668 }
1669 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1670 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1671 {
1672   \tl_put_right:No \l__physicx_matrix_body_tl
1673   {
1674     \physicx@matricelement {#1}
1675     { \__physicx_matrix_row_iterate:n {#2} }
1676     { \__physicx_matrix_col_iterate:n {#3} }
1677   }
1678 }
1679 \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

```

1680 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1681 {
1682   \bool_if:NTF \l__physicx_matrix_transpose_bool
1683   {
1684     #1
1685     \l__physicx_matrix_cols_int
1686     \l__physicx_matrix_rows_int
1687     \use_ii_i:nn
1688   }
1689   {
1690     #1
1691     \l__physicx_matrix_rows_int
1692     \l__physicx_matrix_cols_int
1693     \use:nn
1694   }
1695 }

```

(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.

```

1696 \cs_new:Npn \physicx_construct:nnn #1#2#3
1697 {
1698   \l__physicx_matrix_beginning_tl
1699   \__physicx_adi:nnn {#1} {#2} {#3}
1700   \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1701   {
1702     \int_incr:N \l__physicx_matrix_cols_int

```

```

1703     \__physicx_matrix_last_aux_c:
1704     \int_incr:N \l__physicx_matrix_cols_int
1705   }
1706   \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1707   {
1708     \int_incr:N \l__physicx_matrix_rows_int
1709     \__physicx_matrix_last_aux_r:
1710     \int_incr:N \l__physicx_matrix_rows_int
1711   }
1712   \bool_lazy_or:nnF
1713   { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1714   { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1715   {
1716     \physicx_matrix_set_r_c:nnn
1717     { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1718     { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1719     { \ddots }
1720   }
1721   \bool_if:NT \l__physicx_matrix_infinite_bool
1722   {
1723     \int_incr:N \l__physicx_matrix_rows_int
1724     \int_incr:N \l__physicx_matrix_cols_int
1725     \__physicx_matrix_last_aux_c:
1726     \__physicx_matrix_last_aux_r:
1727     \physicx_matrix_set_r_c:nnn
1728     { \int_use:N \l__physicx_matrix_rows_int }
1729     { \int_use:N \l__physicx_matrix_cols_int }
1730     { \ddots }
1731   }
1732   \l__physicx_matrix_ending_tl
1733 }
1734 \cs_new:Npn \__physicx_matrix_last_aux_c:
1735 {
1736   \int_step_inline:nn \l__physicx_matrix_rows_int
1737   {
1738     \physicx_matrix_set_r_c:nnn
1739     {##1} { \int_use:N \l__physicx_matrix_cols_int }
1740     { \cdots }
1741   }
1742 }
1743 \cs_new:Npn \__physicx_matrix_last_aux_r:
1744 {
1745   \int_step_inline:nn \l__physicx_matrix_cols_int
1746   {
1747     \physicx_matrix_set_r_c:nnn
1748     { \int_use:N \l__physicx_matrix_rows_int } {##1}
1749     { \vdots }
1750   }
1751 }

```

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

1.3.3 Define new matrix command

```

1752 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
1753 {
1754   \NewDocumentCommand #2 { t+ m o o m m }
1755   {
1756     \IfBooleanTF {##1}
1757     {
1758       \IfNoValueTF {##3}
1759       { \newcommand ##2 { #1 + [##5] {##6} } }
1760       {
1761         \IfNoValueTF {##4}
1762         { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1763         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1764       }
1765     }
1766     {
1767       \IfNoValueTF {##3}
1768       { \newcommand ##2 { #1 [##5] {##6} } }
1769       {
1770         \IfNoValueTF {##4}
1771         { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1772         { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1773       }
1774     }
1775   }
1776   \NewDocumentCommand #3 { t+ m m m m }
1777   {
1778     \IfBooleanTF {##1}
1779     { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1780     { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1781   }
1782 }
1783 \__physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1784 \__physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1785 \NewDocumentCommand \newgeneralmatrix { t+ m o o m m }
1786 {
1787   \IfBooleanTF {#1}
1788   {
1789     \IfNoValueTF {#3}
1790     { \newcommand #2 { \generalmatrix + {#5} } }
1791     {
1792       \IfNoValueTF {#4}
1793       { \newcommand #2 [#3] { \generalmatrix + {#5} } }
1794       { \newcommand #2 [#3] [#4] { \generalmatrix + {#5} } }
1795     }
1796   }
1797   {
1798     \IfNoValueTF {#3}
1799     { \newcommand #2 { \generalmatrix {#5} } }
1800     {
1801       \IfNoValueTF {#4}
1802       { \newcommand #2 [#3] { \generalmatrix {#5} } }

```

```

1803         { \newcommand #2 [#3] [#4] { \generalmatrix {#5} } }
1804     }
1805 }
1806 }
1807 \NewDocumentCommand \NewGeneralMatrix { t+ m m m }
1808 {
1809     \IfBooleanTF {#1}
1810     { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1811     { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1812 }

```

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

```

1813 </package>

```

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	
<code>\+</code>	154
<code>\-</code>	154
<code>\%</code>	44, 1075, 1082, 1089, 1113, 1425, 1475, 1541, 1636
<code>\{</code>	303, 533, 552, 570, 611, 636
<code>\}</code>	303, 533, 554, 570, 611, 636
A	
<code>\A</code>	149, 154
<code>\abs</code>	<u>604</u>
<code>\absolutevalue</code>	<u>604</u>
<code>\acommm</code>	<u>604</u>
<code>\aftergroup</code>	204
<code>\anticommutator</code>	<u>604</u>
B	
<code>\begin</code>	308, 309, 310, 311, 312, 313, 314, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 553, 556, 560, 563, 564, 565, 570, 571, 575, 578, 579, 580, 932, 933, 934, 935, 936, 937, 938, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957
<code>\bgroup</code>	203, 205
<code>\Big</code>	586
<code>\big</code>	586
<code>\Bigg</code>	586
<code>\bigg</code>	586
<code>\Biggl</code>	593
<code>\biggl</code>	592
<code>\Biggr</code>	593
<code>\biggr</code>	592
<code>\Bigl</code>	591
<code>\bigl</code>	590
<code>\Bigr</code>	591
<code>\bigr</code>	590
<code>\boldsymbol</code>	193
bool commands:	
<code>\bool_case_false:n</code>	503, 504, 511
<code>\bool_case_true:nTF</code>	588, 1345
<code>\bool_if:N</code> ..	16, 21, 26, 72, 182, 517, 519, 523, 1007, 1062, 1069, 1079, 1087, 1099, 1106, 1110, 1117, 1121, 1385, 1510, 1598, 1682, 1721
<code>\bool_if_p:N</code>	1383, 1471
<code>\bool_if_p:n</code>	590, 591, 592, 593, 1382, 1470
<code>\bool_lazy_and_p:nn</code>	1350, 1355
<code>\bool_lazy_or:nnTF</code>	31, 113, 124, 356, 361, 1381, 1469, 1712
<code>\bool_new:N</code>	9, 10, 11, 12, 13, 38, 53, 419, 421, 695, 696, 697, 705, 719, 727, 728
<code>\bool_set:Nn</code>	430, 431, 1574
<code>\bool_set_false:N</code>	82, 170, 229, 995, 1023, 1038, 1155
<code>\bool_set_inverse:N</code>	765, 882
<code>\bool_set_true:N</code>	116, 127, 137, 177, 225, 228, 994, 1027, 1031, 1042, 1046, 1149, 1448, 1454

<code>\c_false_bool</code>	548, 568	<code>\cs_set_eq:NN</code>	
<code>\c_true_bool</code>	531, 541, 548, 568		56, 57, 61, 62, 193, 197, 201, 687,
<code>\Bqty</code>	604		740, 742, 744, 747, 753, 755, 769,
<code>\bqty</code>	604		778, 787, 796, 805, 823, 826, 829,
			832, 835, 1009, 1013, 1140, 1145,
			1150, 1156, 1387, 1391, 1600, 1604
C			
<code>\cdots</code>	782, 1061, 1064, 1065, 1068,	<code>\cs_set_nopar:Npn</code>	1449, 1455
	1070, 1101, 1102, 1105, 1107, 1740	<code>\cs_set_protected:Npn</code>	
clist commands:			495, 508, 527, 602
<code>\clist_clear:N</code>	963	<code>\cs_to_str:N</code> ..	394, 397, 402, 403, 440
<code>\clist_concat:NNN</code>	970	D	
<code>\clist_if_in:NnTF</code>	1298	<code>\ddots</code>	782, 1088, 1119, 1719, 1730
<code>\clist_map_break:n</code>	163, 175	<code>\DeclareDocumentCommand</code>	
<code>\clist_map_inline:nm</code> ..	69, 159, 171, 968		500, 510, 586, 990, 1364, 1459, 1566
<code>\clist_new:N</code>	703, 704	<code>\diagonalmatrix</code>	28, 1364, 1783
<code>\clist_put_right:Nn</code> ..	844, 850, 1289	dim commands:	
<code>\clist_set:Nn</code>	960	<code>\dim_new:N</code>	729
<code>\clist_set_eq:NN</code>	1265	<code>\dim_use:N</code>	1075, 1082,
<code>\c_empty_clist</code>	1265		1089, 1113, 1425, 1475, 1541, 1636
<code>\comm</code>	634	E	
<code>\commamatrix</code>	30, 1459, 1784	<code>\egroup</code>	204, 206
<code>\commutator</code>	604	else commands:	
cs commands:		<code>\else:</code>	
<code>\cs:w</code>	32, 33, 873		660, 1073, 1200, 1210, 1235, 1247, 1540
<code>\cs_end:</code>	32, 33, 658, 873	<code>\end</code>	308, 309, 310,
<code>\cs_generate_variant:Nn</code>			311, 312, 313, 314, 317, 318, 319,
	3, 4, 5, 6, 76, 79, 185,		320, 321, 322, 323, 324, 325, 326,
	374, 514, 1133, 1271, 1328, 1669, 1679		327, 328, 329, 330, 331, 332, 333,
<code>\cs_if_exist:NTF</code>	280		553, 556, 560, 563, 564, 565, 570,
<code>\cs_if_exist_p:N</code>	1351, 1356		571, 575, 578, 579, 580, 932, 933,
<code>\cs_new:Npn</code>	7, 111, 119, 122,		934, 935, 936, 937, 938, 941, 942,
	134, 145, 157, 191, 195, 199, 203,		943, 944, 945, 946, 947, 948, 949,
	204, 205, 206, 341, 353, 424, 434,		950, 951, 952, 953, 954, 955, 956, 957
	446, 483, 490, 515, 584, 730, 922,	<code>\eval</code>	604
	924, 1126, 1134, 1162, 1175, 1186,	<code>\evaluated</code>	529, 617
	1196, 1222, 1261, 1263, 1272, 1273,	exp commands:	
	1307, 1329, 1403, 1408, 1444, 1501,	<code>\exp_after:wN</code>	
	1528, 1545, 1557, 1596, 1612, 1657,		1416, 1423, 1432, 1439, 1486,
	1670, 1680, 1696, 1734, 1743, 1752		1488, 1496, 1621, 1630, 1642, 1651
<code>\cs_new_eq:NN</code>		<code>\exp_args:Nc</code>	393, 396
	121, 144, 689, 709, 718, 1194	<code>\exp_args:Nf</code>	1170
<code>\cs_new_nopar:Npn</code> ..	648, 656, 665, 667,	<code>\exp_args:Nno</code>	1300, 1335
	672, 678, 711, 712, 715, 716, 717, 1484	<code>\exp_args:Nno</code>	161, 173
<code>\cs_new_protected:Nn</code>	1005	<code>\exp_args:NNx</code>	399
<code>\cs_new_protected:Npn</code>		<code>\exp_args:No</code>	800, 809, 911, 1282
	54, 59, 64, 77, 80, 187	<code>\exp_args:NV</code>	1515, 1523
<code>\cs_new_protected_nopar:Npn</code>		<code>\exp_not:N</code> ..	8, 283, 401, 402, 403, 454, 814
	959, 961, 966	<code>\exp_not:n</code>	
<code>\cs_set:Nn</code> ..	887, 888, 889, 890, 891, 892		458, 459, 659, 661, 709, 740, 744
<code>\cs_set:Npn</code>		<code>\exp_stop_f:</code>	
	379, 385, 389, 607, 617, 618, 634,		8, 449, 456, 1198, 1201, 1224, 1236
	639, 640, 641, 771, 780, 789, 798, 807		
<code>\cs_set:Npx</code>	814		

F		<code>\interval</code> 27, 1351, 1352
fi commands:		
<code>\fi:</code>	460,	
	475, 480, 481, 582, 643, 662, 1076,	
	1219, 1220, 1258, 1259, 1495, 1542	
fp commands:		
<code>\fp_eval:n</code>	27, 1348	
<code>\fpeval</code>	27, 1356, 1357	
G		
<code>\generalmatrix</code> ..	32, 1566, 1790, 1793,	
	1794, 1799, 1802, 1803, 1810, 1811	
group commands:		
<code>\group_begin:</code>	343, 992, 1366, 1461, 1570	
<code>\group_end:</code>	351, 1003, 1401, 1482, 1587	
I		
if commands:		
<code>\if_bool:N</code>	464, 476, 529, 604	
<code>\if_case:w</code>	456	
<code>\if_cs_exist:w</code>	658	
<code>\if_int_compare:w</code>	449, 1071,	
	1198, 1201, 1224, 1236, 1493, 1538	
<code>\IfBooleanT</code>	552, 554, 1465	
<code>\IfBooleanTF</code>	242, 410,	
	502, 559, 561, 574, 576, 596, 928,	
	993, 998, 1367, 1568, 1571, 1590,	
	1591, 1592, 1756, 1778, 1787, 1809	
<code>\IfNoValueTF</code>	377, 383, 1758, 1761,	
	1767, 1770, 1789, 1792, 1798, 1801	
int commands:		
<code>\int_compare:nNnTF</code>	92, 102, 136	
<code>\int_compare:nTF</code>	1021, 1036	
<code>\int_compare_p:nNn</code> .	114, 115, 125, 126	
<code>\int_eval:n</code> ..	27, 1205, 1214, 1232,	
	1244, 1254, 1255, 1347, 1717, 1718	
<code>\int_incr:N</code> .	448, 1180, 1514, 1522,	
	1702, 1704, 1708, 1710, 1723, 1724	
<code>\int_max:nn</code>	651, 653	
<code>\int_new:N</code>		
	.. 39, 40, 49, 50, 51, 52, 423, 700, 701	
<code>\int_set:Nn</code> ..	67, 68, 91, 101, 429,	
	650, 652, 996, 997, 1024, 1039, 1507	
<code>\int_set_eq:NN</code>		
 89, 94, 99, 104, 1183, 1494	
<code>\int_step_inline:nn</code>		
	... 1049, 1331, 1333, 1410, 1412,	
	1428, 1615, 1617, 1638, 1736, 1745	
<code>\int_step_inline:nnn</code>		
 129, 139, 1055, 1093	
<code>\int_use:N</code>	1425,	
	1434, 1441, 1632, 1634, 1644, 1646,	
	1653, 1655, 1728, 1729, 1739, 1748	
<code>\int_zero:N</code>	1177, 1504	
K		
keys commands:		
<code>\keys_define:nn</code>		
 188, 207, 254, 256, 733, 735	
<code>\keys_if_exist:nnTF</code>	161, 173	
<code>\l_keys_key_str</code>	161,	
	164, 173, 178, 276, 280, 283, 911, 914	
<code>\l_keys_path_str</code>	296, 918	
<code>\keys_set:nn</code>	3,	
	164, 178, 243, 244, 255, 282, 335,	
	344, 734, 913, 976, 999, 1000, 1368,	
	1369, 1462, 1464, 1466, 1572, 1573	
keyval commands:		
<code>\keyval_parse:NNn</code>	1266	
<code>\keyval_parse:nnn</code>	1128	
L		
<code>\langle</code>	307, 537, 564, 579	
<code>\ldots</code>	782	
<code>\left</code>	203, 337, 362, 552,	
	559, 564, 570, 571, 574, 578, 579, 580	
<code>\let</code>	497	
<code>\lgroup</code>	559, 574	
<code>\lVert</code>	621	
M		
<code>\mathcal</code>	189, 190	
<code>\mathclose</code>	203, 205	
<code>\mathopen</code>	203, 205	
<code>\matrixquantity</code>	529	
<code>\mqty</code>	604	
msg commands:		
<code>\msg_error:nnn</code>	1160	
<code>\msg_error:nnnn</code>	295, 917	
<code>\msg_new:nnn</code>	47	
<code>\msg_new:nnnn</code>	41	
N		
<code>\NewCommaMatrix</code>	1752	
<code>\newcommamatrix</code>	1752	
<code>\newcommand</code>	380, 386, 390, 399,	
	1759, 1762, 1763, 1768, 1771, 1772,	
	1790, 1793, 1794, 1799, 1802, 1803	
<code>\NewDiagonalMatrix</code>	1752	
<code>\newdiagonalmatrix</code>	1752	
<code>\NewDocumentCommand</code>	240, 300,	
	373, 375, 406, 408, 926, 1754, 1776,	
	1779, 1780, 1785, 1807, 1810, 1811	
<code>\NewGeneralMatrix</code>	1752	
<code>\newgeneralmatrix</code>	1752	
<code>\NewXQuantity</code>	341	
<code>\newxquantity</code>	341	
<code>\nobreak</code>	543, 544, 545	

<code>\norm</code>	604	<code>\physicx_option_or:nn</code>	29
O			
<code>\OOrder</code>	644	<code>\physicx_option_or:nnTF</code>	
<code>\oorder</code>	604	605, 615, 633, 637
or commands:		<code>\c_physicx_Order_tl</code>	190, 234, 625, 645
<code>\or:</code>	457, 458, 459	<code>\c_physicx_order_tl</code>	189, 233, 629, 647
<code>\order</code>	604	<code>\physicx_parse_range:nnN</code>	
<code>\Ordersymbol</code>	234	77, 79, 1278, 1280
<code>\ordersymbol</code>	233	<code>\physicx_parse_range:nnnN</code>	49
P			
<code>\pb</code>	604	<code>\physicx_parse_range_check:</code> ..	49, 855
peek commands:		<code>\physicx_parse_range_nocheck:</code>	49, 856
<code>\peek_charcode_ignore_spaces:N</code>	497	<code>\physicx_qxmatrix:nnn</code> ...	1001, 1005
physicx commands:		<code>\physicx_right:</code> ...	204, 364, 498, 598
<code>\physicx_bf:</code>	193, 197, 201	<code>\physicx_right:N</code>	
<code>\physicx_compat:</code>	14	206, 368, 590, 591, 592, 593
<code>\physicx_compat:TF</code>	231, 622	<code>\physicx_search_also:nn</code>	
<code>\physicx_construct:nnn</code>	157, 168, 185, 186
.....	1370, 1575, 1696	<code>\physicx_search_also:nnTF</code> ..	289, 903
<code>\physicx_declare_legacy_paren:NnnnNn</code>		<code>\physicx_short:</code>	19
.....	583, 608, 609, 610, 611,	<code>\physicx_use_amssymb_type:</code> ..	191, 239
.....	613, 620, 624, 628, 631, 635, 644, 646	<code>\physicx_use_uni_bf_type:</code>	199
<code>\physicx_declare_legacy_quantity:nnNn</code>		<code>\physicx_use_uni_bfit_type:</code>	195, 238
.....	417, 530, 540, 547, 567	<code>\physicx_xquantity:nn</code>	341
<code>\physicx_if_num:n</code>	147	physicx internal commands:	
<code>\physicx_if_num:nTF</code>		<code>__physicx_adi:nnn</code>	
.....	911, 1019, 1034, 1342	730, 887, 888, 889, 890, 891, 892, 1699
<code>\physicx_if_num_sign:n</code>	152	<code>\l__physicx_begin_int</code>	49,
<code>\physicx_left:</code>	203, 364, 498, 598	89, 91, 92, 94, 125, 126, 130, 136, 140
<code>\physicx_left:N</code>		<code>\l__physicx_cmd_arg_int</code>	
.....	205, 366, 590, 591, 592, 593	423, 429, 448, 449, 456
<code>\physicx_mathtools:</code>	24	<code>\l__physicx_cmd_arg_spec_tl</code>	
<code>\physicx_mathtools:TF</code>	315, 939	422, 428, 441, 450
<code>\physicx_matrix_array_parse:n</code> ...		<code>\l__physicx_cmd_auto_body_bool</code> ..	
.....	1307, 1579	421, 431, 476, 519, 523
<code>\physicx_matrix_array_parse_-</code>		<code>\l__physicx_cmd_auto_body_tl</code> ...	
main:	1329, 1581	420, 427, 443, 477, 478
<code>\physicx_matrix_diag_parse:n</code> ...		<code>\l__physicx_cmd_noauto_body_bool</code>	
... 845, 1126, 1372, 1376, 1377, 1583		419, 430, 464, 517
<code>\physicx_matrix_item_parse:n</code> ...		<code>\l__physicx_cmd_noauto_body_tl</code> ..	
.....	851, 1263, 1380, 1584	418, 426, 442, 465, 466
<code>\physicx_matrix_new_type:nn</code> ...	922	<code>\l__physicx_col_list_seq</code>	
<code>\physicx_matrix_new_type:nnn</code> ..	922	707, 761, 1452, 1456
<code>\physicx_matrix_set_r_c:nnn</code> 689,		<code>__physicx_commamatrix_enhanced:</code>	
823, 826, 829, 832, 835, 1181, 1189,		1472, 1501
1204, 1213, 1230, 1242, 1253, 1300,		<code>__physicx_commamatrix_enhanced_-</code>	
1324, 1335, 1716, 1727, 1738, 1747		aux:nNn	1515, 1523, 1528
<code>\physicx_matrix_use_r_c:nn</code>		<code>__physicx_commamatrix_enhanced_-</code>	
.....	656, 1417, 1424,	aux_e:nnn	1516, 1545
1433, 1440, 1622, 1631, 1643, 1652		<code>__physicx_commamatrix_enhanced_-</code>	
<code>\physicx_new_type:nnn</code>		aux_ne:nnn	1524, 1557
.....	187, 301, 923, 925	<code>\g__physicx_compat_bool</code>	
		11, 16, 209, 225
		<code>__physicx_declare_init:</code>	1444
		<code>__physicx_declare_init:nnn</code> 424,	436


```

\__physicx_declare_legacy_-
  quantity_aux:nnnn ..... 446, 487
\__physicx_declare_legacy_-
  quantity_aux:NNnnn .. 439, 490, 514
\__physicx_declare_legacy_-
  quantity_aux:nw .... 437, 483, 488
\__physicx_diagonalmatrix_diag_-
  main: ..... 1150, 1156, 1194, 1199
\__physicx_diagonalmatrix_-
  enhanced:nnn .... 1387, 1391, 1406
\__physicx_diagonalmatrix_-
  generate_body:NNN .... 1399, 1408
\__physicx_diagonalmatrix_-
  generate_enhanced_body:NNN ..
    ..... 1398, 1403
\__physicx_diagonalmatrix_no_-
  diag: ..... 1157, 1186, 1195
\__physicx_diagonalmatrix_set_-
  diag: ..... 1151, 1175
\l_physicx_end_int ..... 50,
  99, 101, 102, 104, 126, 130, 136, 140
\__physicx_expand:w .... 709, 740,
  742, 744, 747, 1166, 1277, 1311, 1468
\__physicx_generalmatrix: 1585, 1596
\__physicx_generalmatrix_-
  generate:nnn .... 1600, 1604, 1609
\__physicx_if_can_num:n ..... 1340
\__physicx_if_keyval:nTF .. 145, 1375
\l_physicx_invalid_range_bool ..
  ..... 53, 72, 82, 116, 127, 137
\l_physicx_item_ignore_clist ...
  ..... 1265, 1289, 1298
\g_physicx_mathtools_bool .....
  ..... 9, 26, 228, 229
\__physicx_matrix_add_data:nn ...
  ..... 847, 853, 964, 966
\l_physicx_matrix_after_begin_-
  tl ..... 725, 863, 865, 1496
\l_physicx_matrix_after_end_tl .
  ..... 726, 866, 868, 1498
\__physicx_matrix_appto_body_-
  e:nnn ..... 1010, 1388, 1601, 1657
\__physicx_matrix_appto_body_-
  ne:nnn ..... 1014, 1392, 1605, 1657
\l_physicx_matrix_args_tl .....
  ..... 724, 861, 862, 1496
\l_physicx_matrix_array_tl ....
  ..... 698, 737, 1467,
  1468, 1474, 1476, 1479, 1506, 1579
\__physicx_matrix_autocalc:nn ...
  ..... 718, 753, 755, 1262, 1313, 1319
\__physicx_matrix_begin:w .....
  ..... 716, 858, 1496
\l__physicx_matrix_beginning_tl .
  ..... 731, 893, 895, 1698
\l_physicx_matrix_body_tl .....
  ..... 699, 1017, 1052,
  1057, 1058, 1064, 1065, 1070, 1075,
  1082, 1084, 1086, 1088, 1089, 1091,
  1095, 1096, 1101, 1102, 1107, 1113,
  1114, 1116, 1118, 1119, 1414, 1421,
  1430, 1437, 1478, 1490, 1497, 1503,
  1535, 1626, 1635, 1647, 1659, 1672
\__physicx_matrix_calc:nn .....
  ..... 648, 753, 1141
\__physicx_matrix_col_iterate:n .
  ..... 715, 839, 1455, 1553, 1563, 1665, 1676
\l_physicx_matrix_cols_int ....
  ..... 652, 653,
  701, 750, 997, 1036, 1039, 1055,
  1085, 1093, 1115, 1122, 1123, 1177,
  1180, 1184, 1232, 1244, 1255, 1280,
  1333, 1493, 1494, 1685, 1692, 1702,
  1704, 1718, 1724, 1729, 1739, 1745
\l_physicx_matrix_diag_bool ...
  ..... 705, 1149, 1155
\__physicx_matrix_diag_calc:nn ..
  ..... 1140, 1145, 1190,
  1207, 1216, 1225, 1237, 1248, 1261
\l_physicx_matrix_diag_clist ...
  ..... 703, 842, 844, 1372, 1583
\__physicx_matrix_diag_parse_-
  aux:n ..... 1129, 1134
\__physicx_matrix_diag_parse_-
  aux:nn ..... 1130, 1162
\__physicx_matrix_diag_parse_-
  aux_anti:n ..... 1170, 1222
\__physicx_matrix_diag_parse_-
  aux_regu:n ..... 1173, 1196
\l_physicx_matrix_dotcol_bool ..
  ..... 697, 1038, 1042,
  1046, 1062, 1087, 1099, 1117, 1121
\l_physicx_matrix_dotrow_bool ..
  ..... 696, 1023, 1027, 1031, 1079
\__physicx_matrix_element_-
  aux:nnn ..... 769, 775,
  778, 784, 787, 793, 796, 802, 805, 810
\__physicx_matrix_end: .... 859, 1498
\__physicx_matrix_end:w ..... 717
\l_physicx_matrix_ending_tl ...
  ..... 732, 896, 898, 1732
\l_physicx_matrix_enhanced_bool
  ..... 728, 879, 882, 1383, 1471
\__physicx_matrix_enhanced_init:
  ..... 1444, 1509, 1614
\l_physicx_matrix_expand_-
  element_bool ..... 719,

```

818, 1007, 1385, 1448, 1454, 1510, 1598
 __physicx_matrix_generate_-
 body:NNNN 1405, 1608, 1611
 \l__physicx_matrix_infinite_bool
 695, 762, 765,
 994, 995, 1069, 1106, 1110, 1574, 1721
 \l__physicx_matrix_item_clist ...
 704, 848, 850, 1380, 1584
 __physicx_matrix_item_parse_-
 aux:n 1267, 1272
 __physicx_matrix_item_parse_-
 aux:nn 1268, 1273
 __physicx_matrix_last_aux_c: ...
 1703, 1725, 1734
 __physicx_matrix_last_aux_r: ...
 1709, 1726, 1743
 \l__physicx_matrix_last_col_tl ..
 714, 841, 1700, 1714
 \l__physicx_matrix_last_row_tl ..
 713, 840, 1706, 1713
 \l__physicx_matrix_main_tl
 702, 757, 1336, 1577
 __physicx_matrix_row_iterate:n .
 712, 838, 1449, 1552, 1562, 1664, 1675
 \l__physicx_matrix_rows_int
 650, 651, 700, 749, 996,
 1021, 1024, 1049, 1071, 1183, 1278,
 1331, 1507, 1538, 1686, 1691, 1708,
 1710, 1717, 1723, 1728, 1736, 1748
 __physicx_matrix_save_or_print:
 1002, 1400, 1481, 1484, 1586
 \l__physicx_matrix_save_tl
 871, 873, 1486, 1489
 \l__physicx_matrix_sep_dim
 729, 869, 1075, 1082,
 1089, 1113, 1425, 1475, 1541, 1636
 __physicx_matrix_set_data:nn ...
 846, 852, 961
 __physicx_matrix_set_r_c_-
 ckall:nnn 687, 836
 __physicx_matrix_set_r_c_-
 ckep:nnn 672, 827
 __physicx_matrix_set_r_c_-
 ckig:nnn 667, 830
 __physicx_matrix_set_r_c_-
 ckigep:nnn 678, 688, 833
 __physicx_matrix_set_r_c_-
 nock:nnn 665, 690, 824
 \l__physicx_matrix_tmparr_c_sep .
 1318, 1321, 1322
 \l__physicx_matrix_tmparr_r_sep .
 1312, 1314, 1316
 __physicx_matrix_transpose:N ...
 1397, 1607, 1680
 \l__physicx_matrix_transpose_-
 bool 727, 874, 1682
 \l__physicx_max_int
 51, 68, 99, 102, 104, 114, 125
 \l__physicx_min_int
 52, 67, 89, 92, 94, 115
 __physicx_nauto_case:nnnn . 492, 515
 __physicx_new_matrix_cmd:NNN . 1752
 __physicx_new_xquantity_aux:w ..
 379, 385, 389, 393, 396
 __physicx_parse_range_aux:n 71, 80
 __physicx_parse_range_range: ...
 57, 62, 107, 144
 __physicx_parse_range_range_-
 check: 57, 122, 144
 __physicx_parse_range_range_-
 nocheck: 62, 134
 __physicx_parse_range_single:n .
 56, 61, 109, 121
 __physicx_parse_range_single_-
 check:n 56, 111, 121
 __physicx_parse_range_single_-
 nocheck:n 61, 119
 \g__physicx_physics_bool 10
 \l__physicx_quantity_args_tl ...
 246, 268, 269, 348
 \l__physicx_quantity_code_tl ...
 247, 270, 345, 349
 \l__physicx_quantity_left_size_-
 tl 248, 262, 357, 362, 366
 \l__physicx_quantity_left_tl ...
 249, 260, 347
 \l__physicx_quantity_post_tl ...
 250, 259, 371
 \l__physicx_quantity_pre_tl
 251, 258, 355
 \l__physicx_quantity_right_size_-
 tl 252, 263, 358, 363, 368
 \l__physicx_quantity_right_tl ...
 253, 261, 350
 __physicx_qxmatrix_appto_-
 body:nnn 1009, 1013,
 1053, 1059, 1066, 1092, 1097, 1103
 \g__physicx_reqty_bool
 13, 216, 529, 604
 \l__physicx_row_list_seq
 706, 759, 1446, 1450
 \g__physicx_short_bool ... 12, 21, 211
 \l__physicx_tmp_col_seq .. 1530, 1533
 \l__physicx_tmp_coled_seq
 1532, 1537, 1547, 1559
 \l__physicx_tmp_colnum_seq
 1281, 1287, 1296

\l__physicx_tmp_rownum_seq	1279, 1285, 1294
\l__physicx_tmp_seq	1505, 1508, 1512, 1520
\l__physicx_tmp_tl	451, 465, 477, 1619, 1625, 1628, 1634, 1640, 1646, 1649, 1655
\l__physicx_tmpa_bool	38, 170, 177, 182
\l__physicx_tmpa_int	39, 1504, 1514, 1516, 1522, 1524
\l__physicx_tmpa_seq	73, 83, 117, 120, 131, 141
\l__physicx_tmpa_tl	87, 88, 91, 97, 98, 101, 276, 277
\l__physicx_tmparr_tl	1309, 1310, 1311, 1312
\l__physicx_tmpb_int	40
\l__physicx_tmpb_seq	86, 87, 97
\l__physicx_tmpdiag_seq	1167, 1178, 1188, 1191, 1192, 1202, 1208, 1209, 1211, 1217, 1218, 1226, 1227, 1228, 1238, 1239, 1240, 1249, 1250, 1251
\l__physicx_tmpdiag_tl	1164, 1165, 1166, 1167
\l__physicx_tmpitem_tl	1275, 1276, 1277, 1283, 1301
__physicx_xquantity_aux:nnnn	346, 353, 374
\physicxempty	661, 675, 683, 720, 721, 820
\physicxexcept	723, 791, 800, 809, 815, 817
\PHYSICXIGNORE	7, 8, 669, 680, 1283
\physicxset	240
\physicxtmp	417, 999, 1368, 1466, 1572
\poissonbracket	604
\pqty	604
prg commands:	
\prg_generate_conditional_-variant:Nnn	186
\prg_new_conditional:Npnn	14, 19, 24, 29, 147, 152, 168, 1340
\prg_replicate:nn	1085, 1115
\prg_return_false:	17, 22, 27, 35, 150, 155, 183, 1361
\prg_return_true:	17, 22, 27, 34, 150, 155, 183, 1343, 1360
\ProcessKeysPackageOptions	222
Q	
\qty	604
\quantity	529, 607
quark commands:	
\quark_if_recursion_tail_stop:n	485, 486
\q_recursion_stop	438
\q_recursion_tail	438
\qmatrix	981, 1593
\qxqty	415
R	
\rangle	307, 537, 564, 579
regex commands:	
\regex_match:nnTF ..	149, 154, 800, 809
\RequirePackage	213, 214, 215
\rgroup	561, 576
\right	204, 338, 363, 554, 561, 564, 570, 571, 576, 578, 579, 580
\rVert	621
S	
\sb	711, 1052, 1057, 1064, 1091, 1095, 1101
scan commands:	
\scan_stop:	1072, 1539
seq commands:	
\c_empty_seq	66, 1532
\seq_clear:N	83
\seq_concat:NNN	73
\seq_count:N	1191, 1192, 1208, 1209, 1217, 1218, 1226, 1227, 1238, 1239, 1249, 1250, 1314, 1321, 1508
\seq_if_empty:NTF	1446, 1452
\seq_item:Nn	1450, 1456
\seq_map_indexed_inline:Nn	1178, 1188, 1202, 1211, 1228, 1240, 1251, 1316, 1322, 1533
\seq_map_inline:Nn	1285, 1287, 1294, 1296
\seq_map_tokens:Nn	1512, 1520
\seq_new:N	706, 707
\seq_pop_left:NN	87, 97
\seq_put_right:Nn	117, 120, 131, 141, 1547, 1559
\seq_set_eq:NN	66, 1532
\seq_set_split:Nnn	5, 86, 759, 761, 1167, 1312, 1318, 1505, 1530
\seq_use:Nn	1537
\setmatrixdata	959
\setmatrixtype	900, 922, 932, 933, 934, 935, 936, 937, 938, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957
\setquantitytype	273, 300, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333
\smallmatrixquantity	529
\smqty	604

str commands:	
\c_backslash_str	277
\str_case_e:nnTF	1136
\str_const:Nn	692, 693, 694
\str_if_eq:nnTF	470, 472
\symbf	201
\symbfit	197
T	
T _E X and L ^A T _E X 2 _ε commands:	
\@declarepreamblecmd	583
\@declarequantitycmd	417
\@ifpackageloaded	224, 227, 237
\@ifstar	401
\c@MaxMatrixCols	878, 1493, 1494
\physics@align	692, 884, 1318, 1477, 1531
\physics@cr	693, 883, 1312, 1475, 1505
\physics@matricelement	711, 766, 770, 771, 779, 780, 788, 789, 797, 798, 806, 807, 814, 1551, 1561, 1663, 1674
\physics@sep	694, 759, 761, 885, 1167
tex commands:	
\tex_advance:D	1122, 1123
text commands:	
\text_expand:n	742, 1549, 1661
tl commands:	
\c_empty_tl	265, 509
\tl_clear:N	426, 427, 428, 1017, 1503
\tl_const:Nn	189, 190
\tl_gset_eq:NN	1488
\tl_head:N	276
\tl_if_empty:NTF	88, 98, 1577, 1700, 1706
\tl_if_empty:nTF	345, 674, 682, 773, 1373, 1463
\tl_if_empty_p:N	357, 358, 1713, 1714
\tl_if_eq:nnTF	669, 680, 1282
\tl_if_head_eq_charcode:nNTF	1168
\tl_if_head_eq_meaning_p:nN	1347, 1348, 1352, 1357
\tl_if_in:nnTF	84, 146, 782, 791
\tl_if_novalue_p:n	454
\tl_new:N	246, 247, 248, 249, 250, 251, 252, 253, 417, 418, 420, 422, 698, 699, 702, 713, 714, 721, 723, 724, 725, 726, 731, 732
\tl_put_right:Nn	450, 465, 466, 477, 478, 817, 865, 868, 895, 898, 1052, 1057, 1058, 1064, 1065, 1070, 1075, 1082, 1084, 1086, 1088, 1089, 1091, 1095, 1096, 1101, 1102, 1107, 1113, 1114, 1116, 1118, 1119, 1414, 1421, 1430, 1437, 1535, 1626, 1635, 1647, 1659, 1672
\tl_replace_all:Nnn	6, 1474, 1476
\tl_set:Nn	268, 276, 345, 451, 666, 670, 675, 676, 683, 684, 861, 873, 1164, 1165, 1275, 1276, 1309, 1310, 1467, 1619, 1628, 1640, 1649
\tl_set_eq:NN	233, 234, 262, 263, 1478
\tl_tail:N	280, 283
\tl_tail:n	1171
token commands:	
\token_if_cs:NTF	1486
\token_if_eq_meaning:NNTF	277
\token_if_eq_meaning_p:NN	362, 363
\TrimSpaces	300, 926
\txqty	416
U	
use commands:	
\use:n	278, 286, 747
\use:nn	1611, 1693
\use:nnnn	4
\use_i:nn	493, 1279, 1396, 1592
\use_i:nnn	1592
\use_i_ii:nnn	1394, 1591
\use_ii:nn	278, 493, 1281
\use_ii_i:nn	1611, 1687
\use_none:n	284
\use_none:nn	718, 755, 1145
V	
\vdots	782, 1084, 1086, 1088, 1114, 1116, 1118, 1749
\Vert	306, 538, 580
\vert	305, 536, 543, 544, 545, 578, 610, 614
\vqty	604
X	
\xquantity	373
Z	
\Z	149, 154