

# The `physicx` package

Wenjian Chern (Longaster\*)

February 28, 2022, version v0.3

## 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](mailto:longaster@163.com)

```

34     { \prg_return_true: }
35     { \prg_return_false: }
36 }
37 \cs_if_exist_use:NF \hook_gput_code:nnn { \use_none:nnn }
38 { package/unicode-math/after } { ./package }
39 {
40     \cs_gset_eq:NN \physicx_unimath:TF \use_i:nn
41     \cs_gset_eq:NN \physicx_unimath:T \use:n
42     \cs_gset_eq:NN \physicx_unimath:F \use_none:n
43 }
44 \prg_set_conditional:Npnn \physicx_unimath: { T, F, TF }
45 {
46     \tl_if_exist:cTF { ver @ unicode-math . \@pkgextension }
47     { \prg_return_true: } { \prg_return_false: }
48 }
49
50 \clist_new:N \l__physicx_tmpa_clist
51 \bool_new:N \l__physicx_tmpa_bool
52 \int_new:N \l__physicx_tmpa_int
53 \int_new:N \l__physicx_tmpb_int
54 \msg_new:nnnn { physicx } { unknown-key }
55 { The~key~‘#1’~is~unknown~and~is~being~ignored. }
56 {
57     The~module~#2~does~not~have~a~key~called~#1.\\
58     Check~that~you~have~spelled~the~key~name~correctly.
59 }
60 \msg_new:nnn { physicx } { diag-key }
61 { 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:
62 \int_new:N \l__physicx_begin_int
63 \int_new:N \l__physicx_end_int
64 \int_new:N \l__physicx_max_int
65 \int_new:N \l__physicx_min_int
66 \bool_new:N \l__physicx_invalid_range_bool
67 \cs_new_protected:Npn \physicx_parse_range_check:
68 {
69     \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_check:n
70     \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_check:
71 }
72 \cs_new_protected:Npn \physicx_parse_range_nocheck:
73 {
74     \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_nocheck:n
75     \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_nocheck:
76 }
77 \cs_new_protected:Npn \physicx_parse_range:nnnN #1#2#3#4
78 {
79     \seq_set_eq:NN #4 \c_empty_seq
80     \int_set:Nn \l__physicx_min_int {#1}
81     \int_set:Nn \l__physicx_max_int {#2}
82     \clist_map_inline:nn {#3}
83     {

```

```

84     \__physicx_parse_range_aux:n {##1}
85     \bool_if:NF \l__physicx_invalid_range_bool
86     { \seq_concat:NNN #4 #4 \l__physicx_tmpa_seq }
87   }
88 }
89 \cs_generate_variant:Nn \physicx_parse_range:nnnN { nnvN, nnxN }
90 \cs_new_protected:Npn \physicx_parse_range:nnN
91   { \physicx_parse_range:nnnN { 1 } }
92 \cs_generate_variant:Nn \physicx_parse_range:nnN { nvN, nxN }
93 \cs_new_protected:Npn \__physicx_parse_range_aux:n #1
94   {
95     \bool_set_false:N \l__physicx_invalid_range_bool
96     \seq_clear:N \l__physicx_tmpa_seq
97     \tl_if_in:nnTF {#1} { - }
98     {
99       \seq_set_split:Nnn \l__physicx_tmpb_seq { - } {#1}
100      \seq_pop_left:NN \l__physicx_tmpb_seq \l__physicx_tmpa_tl
101      \tl_if_empty:NTF \l__physicx_tmpa_tl
102      { \int_set_eq:NN \l__physicx_begin_int \l__physicx_min_int }
103      {
104        \int_set:Nn \l__physicx_begin_int { \l__physicx_tmpa_tl }
105        \int_compare:nNnT \l__physicx_begin_int < \l__physicx_min_int
106        {
107          \int_set_eq:NN \l__physicx_begin_int \l__physicx_min_int
108        }
109      }
110      \seq_pop_left:NN \l__physicx_tmpb_seq \l__physicx_tmpa_tl
111      \tl_if_empty:NTF \l__physicx_tmpa_tl
112      { \int_set_eq:NN \l__physicx_end_int \l__physicx_max_int }
113      {
114        \int_set:Nn \l__physicx_end_int { \l__physicx_tmpa_tl }
115        \int_compare:nNnT \l__physicx_end_int > \l__physicx_max_int
116        {
117          \int_set_eq:NN \l__physicx_end_int \l__physicx_max_int
118        }
119      }
120      \__physicx_parse_range_range:
121    }
122    { \__physicx_parse_range_single:n {#1} }
123  }
124 \cs_new:Npn \__physicx_parse_range_single_check:n #1
125   {
126     \bool_lazy_or:nnTF
127     { \int_compare_p:nNn {#1} > \l__physicx_max_int }
128     { \int_compare_p:nNn {#1} < \l__physicx_min_int }
129     { \bool_set_true:N \l__physicx_invalid_range_bool }
130     { \seq_put_right:Nn \l__physicx_tmpa_seq {#1} }
131   }
132 \cs_new:Npn \__physicx_parse_range_single_nocheck:n #1
133   { \seq_put_right:Nn \l__physicx_tmpa_seq {#1} }
134 \cs_new_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_check:n
135 \cs_new:Npn \__physicx_parse_range_range_check:
136   {
137     \bool_lazy_or:nnTF

```

```

138     { \int_compare_p:nNn \l__physicx_begin_int > \l__physicx_max_int }
139     { \int_compare_p:nNn \l__physicx_begin_int > \l__physicx_end_int }
140     { \bool_set_true:N \l__physicx_invalid_range_bool }
141     {
142         \int_step_inline:nnn
143         { \l__physicx_begin_int } { \l__physicx_end_int }
144         { \seq_put_right:Nn \l__physicx_tmpa_seq {##1} }
145     }
146 }
147 \cs_new:Npn \__physicx_parse_range_range_nocheck:
148 {
149     \int_compare:nNnTF \l__physicx_begin_int > \l__physicx_end_int
150     { \bool_set_true:N \l__physicx_invalid_range_bool }
151     {
152         \int_step_inline:nnn
153         { \l__physicx_begin_int } { \l__physicx_end_int }
154         { \seq_put_right:Nn \l__physicx_tmpa_seq {##1} }
155     }
156 }
157 \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 ??.)

```

158 \cs_new:Npn \__physicx_if_keyval:nTF #1
159 { \tl_if_in:nnTF {#1} { = } }
160 \prg_new_conditional:Npnn \physicx_if_num:n #1 { T, F, TF }
161 {
162     \regex_match:nnTF { \A [[:digit:]]+ \Z } {#1}
163     { \prg_return_true: } { \prg_return_false: }
164 }
165 \prg_new_conditional:Npnn \physicx_if_num_sign:n #1 { T, F, TF }
166 {
167     \regex_match:nnTF { \A [\+|-]* [[:digit:]]+ \Z } {#1}
168     { \prg_return_true: } { \prg_return_false: }
169 }
170 \cs_new:Npn \physicx_search_also:nn #1#2
171 {
172     \clist_map_inline:nn {#1}
173     {
174         \exp_args:Nno \keys_if_exist:nnT {##1} { \l_keys_key_str }
175         {
176             \clist_map_break:n
177             { \keys_set:no {##1} { \l_keys_key_str = {#2} } }
178         }
179     }
180 }
181 \prg_new_conditional:Npnn \physicx_search_also:nn #1#2 { T, F, TF }
182 {
183     \bool_set_false:N \l__physicx_tmpa_bool
184     \clist_map_inline:nn {#1}
185     {
186         \exp_args:Nno \keys_if_exist:nnT {##1} { \l_keys_key_str }
187         {
188             \clist_map_break:n

```

```

189         {
190             \bool_set_true:N \l__physicx_tmpa_bool
191             \keys_set:no {##1} { \l_keys_key_str = {#2} }
192         }
193     }
194 }
195 \bool_if:NTF \l__physicx_tmpa_bool
196 { \prg_return_true: } { \prg_return_false: }
197 }
198 \cs_generate_variant:Nn \physicx_search_also:nn { no , oo }
199 \prg_generate_conditional_variant:Nnn \physicx_search_also:nn { no , oo } { T , F , TF }
200 \cs_new_protected:Npn \physicx_new_type:nnn #1#2#3
201 { \keys_define:nn { physicx/#1 } { type / #2 .meta:n = {#3} } }
202 \tl_const:Nn \c_physicx_order_tl { \mathcal{o} }
203 \tl_const:Nn \c_physicx_Order_tl { \mathcal{O} }
204 \cs_new:Npn \physicx_use_amssymb_type:
205 {
206     \cs_set_eq:NN \physicx_bf: \boldsymbol
207 }
208 \cs_new:Npn \physicx_use_uni_bfit_type:
209 {
210     \cs_set_eq:NN \physicx_bf: \symbfit
211 }
212 \cs_new:Npn \physicx_use_uni_bf_type:
213 {
214     \cs_set_eq:NN \physicx_bf: \symbf
215 }
216 \cs_new:Npn \physicx_left: { \mathopen{}\mathclose\bgroup\left }
217 \cs_new:Npn \physicx_right: { \aftergroup\egroup\right }
218 \cs_new:Npn \physicx_left:N { \mathopen{}\mathclose\bgroup }
219 \cs_new:Npn \physicx_right:N { \egroup }
220 \cs_new:Npn \__physicx_loadpackage_options:nnn #1#2#3
221 {
222     \clist_if_empty:nF {#1} { \PassOptionsToPackage {#1} {#3} }
223     \RequirePackage {#3}
224 }
225 \keys_define:nn { physicx }
226 {
227     compat .bool_set:N = \g__physicx_compat_bool ,
228     compat .default:n = true ,
229     short .bool_set:N = \g__physicx_short_bool ,
230     short .default:n = true ,
231     physics .code:n = \__physicx_loadpackage_options:nnn {#1} { } {physics} ,
232     physics .default:n = { } ,
233     mathtools .code:n = \__physicx_loadpackage_options:nnn {#1} { } {mathtools} ,
234     mathtools .default:n = { } ,
235     unimath .code:n = \__physicx_loadpackage_options:nnn {#1} { } { unicode-math } ,
236     unimath .default:n = { } ,
237     reqty .bool_set:N = \g__physicx_reqty_bool ,
238     reqty .default:n = true ,
239     reqty .initial:n = true ,
240     noqty .meta:n = { reqty = false } ,
241 }

```

```

242 %
243 \ProcessKeysPackageOptions { physicx }
244 %
245 \@ifpackageloaded{physics}
246 { \bool_set_true:N \g__physicx_compat_bool }
247 { }
248 \@ifpackageloaded{mathtools}
249 { \bool_set_true:N \g__physicx_mathtools_bool }
250 { \bool_set_false:N \g__physicx_mathtools_bool }
251 %
252 \physicx_compat:T
253 {
254   \tl_set_eq:NN \ordersymbol \c_physicx_order_tl
255   \tl_set_eq:NN \Ordersymbol \c_physicx_Order_tl
256 }
257 %
258 \@ifpackageloaded {unicode-math}
259 { \physicx_use_uni_bfit_type: }
260 { \physicx_use_amssymb_type: }
261 \physicx_unimath:T { %% TODO:
262   \AtBeginDocument{
263     \DeclareDocumentCommand\vectorbold{ s m }
264     { \IfBooleanTF{#1} { \physicx_bf:{#2} } { \mathbf{#2} } }
265     \DeclareDocumentCommand\vectorarrow{ s m }
266     { \IfBooleanTF{#1} { \vec{\physicx_bf:{#2}} } { \vec{\mathbf{#2}} } }
267     \DeclareDocumentCommand\vectorunit{ s m }
268     { \IfBooleanTF{#1} { \physicx_bf:{\hat{#2}} } { \hat{\mathbf{#2}} } }
269     \setmathfont [range={"2219}]{STIX~Two~Math}
270     \DeclareDocumentCommand \dotproduct { } { \vysmbllkcircle }
271     \DeclareDocumentCommand \crossproduct { } { \vectimes }
272     \DeclareDocumentCommand \vnabla { } { \symbf \nabla }
273     \let\div\divergence
274   }
275 }

```

`\physicxset` `physicx` setup command.

```

276 \NewDocumentCommand \physicxset { s m }
277 {
278   \IfBooleanTF {#1}
279   { \keys_set:nn { physicx/#2 } }
280   { \keys_set:nn { physicx } {#2} }
281 }

```

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

## 1.2 Quantity things

### 1.2.1 New quantity interfaces

```

282 \tl_new:N \l__physicx_quantity_args_tl
283 \tl_new:N \l__physicx_quantity_code_tl
284 \tl_new:N \l__physicx_quantity_left_size_tl
285 \tl_new:N \l__physicx_quantity_left_tl
286 \tl_new:N \l__physicx_quantity_post_tl

```

```

287 \tl_new:N \l__physicx_quantity_pre_tl
288 \tl_new:N \l__physicx_quantity_right_size_tl
289 \tl_new:N \l__physicx_quantity_right_tl
290 \keys_define:nn { physicx }
291 { quantity .code:n = \keys_set:nn { physicx/quantity } {#1} }
292 \keys_define:nn { physicx/quantity }
293 {
294   pre .tl_set:N = \l__physicx_quantity_pre_tl ,
295   post .tl_set:N = \l__physicx_quantity_post_tl ,
296   left .tl_set:N = \l__physicx_quantity_left_tl ,
297   right .tl_set:N = \l__physicx_quantity_right_tl ,
298   left-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_left_size_tl #1 } ,
299   right-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_right_size_tl #1 } ,
300   size .meta:n = { left-size = {#1} , right-size = {#1} } ,
301   noauto .meta:n = { left-size = \c_empty_tl , right-size = \c_empty_tl } ,
302   noauto .value_required:n = false ,
303   args .code:n =
304     \tl_set:Nn \l__physicx_quantity_args_tl { [#1] } ,
305   args* .tl_set:N = \l__physicx_quantity_args_tl ,
306   code .tl_set:N = \l__physicx_quantity_code_tl ,
307   type .multichoice: ,
308
309   settype .code:n = \setquantitytype #1 ,
310
311   unknown .code:n =
312     \tl_set:Nx \l__physicx_tmpa_tl { \tl_head:N \l_keys_key_str }
313     \token_if_eq_meaning:NNTF \l__physicx_tmpa_tl \c_backslash_str
314     { \use:n } { \use_ii:nn }
315     {
316       \cs_if_exist:cTF { \tl_tail:N \l_keys_key_str }
317       {
318         \keys_set:nx { physicx/quantity }
319         { size = \exp_not:c { \tl_tail:N \l_keys_key_str } }
320         \use_none:n
321       }
322       { \use:n }
323     }
324     {
325       \physicx_search_also:nnF
326       {
327         physicx/quantity/type ,
328       }
329       {#1}
330       {
331         \msg_error:nxxx { physicx } { unknown-key }
332         \l_keys_path_str { physicx/quantity }
333       }
334     } ,
335   }
336 \NewDocumentCommand \setquantitytype { >{ \TrimSpaces } m }
337 { \physicx_new_type:nnn { quantity } {#1} }
338 \setquantitytype { b } { left={[] , right={}} , }
339 \setquantitytype { B } { left={\{ , right={\}} , }
340 \setquantitytype { p } { left={ ( , right={ ) } , }

```

```

341 \setquantitytype { v } { left=\vert , right=\vert , }
342 \setquantitytype { V } { left=\Vert , right=\Vert , }
343 \setquantitytype { a } { left=\langle , right=\rangle , }
344 \setquantitytype { m } { left=\begin{matrix} , right=\end{matrix} , noauto }
345 \setquantitytype { bm } { left=\begin{bmatrix} , right=\end{bmatrix} , noauto }
346 \setquantitytype { Bm } { left=\begin{Bmatrix} , right=\end{Bmatrix} , noauto }
347 \setquantitytype { pm } { left=\begin{pmatrix} , right=\end{pmatrix} , noauto }
348 \setquantitytype { vm } { left=\begin{vmatrix} , right=\end{vmatrix} , noauto }
349 \setquantitytype { Vm } { left=\begin{Vmatrix} , right=\end{Vmatrix} , noauto }
350 \setquantitytype { sm } { left=\begin{smallmatrix} , right=\end{smallmatrix} , noauto }
351 \physicx_mathtools:T
352 {
353   \setquantitytype { m* } { left=\begin{matrix*} , right=\end{matrix*} , noauto }
354   \setquantitytype { bm* } { left=\begin{bmatrix*} , right=\end{bmatrix*} , noauto }
355   \setquantitytype { Bm* } { left=\begin{Bmatrix*} , right=\end{Bmatrix*} , noauto }
356   \setquantitytype { pm* } { left=\begin{pmatrix*} , right=\end{pmatrix*} , noauto }
357   \setquantitytype { vm* } { left=\begin{vmatrix*} , right=\end{vmatrix*} , noauto }
358   \setquantitytype { Vm* } { left=\begin{Vmatrix*} , right=\end{Vmatrix*} , noauto }
359   \setquantitytype { sm* } { left=\begin{smallmatrix*} , right=\end{smallmatrix*} , noauto }
360   \setquantitytype { sbm } { left=\begin{bsmallmatrix} , right=\end{bsmallmatrix} , noauto }
361   \setquantitytype { sBm } { left=\begin{Bsmallmatrix} , right=\end{Bsmallmatrix} , noauto }
362   \setquantitytype { spm } { left=\begin{psmallmatrix} , right=\end{psmallmatrix} , noauto }
363   \setquantitytype { svm } { left=\begin{vsmallmatrix} , right=\end{vsmallmatrix} , noauto }
364   \setquantitytype { sVm } { left=\begin{Vsmallmatrix} , right=\end{Vsmallmatrix} , noauto }
365   \setquantitytype { sbm* } { left=\begin{bsmallmatrix*} , right=\end{bsmallmatrix*} , noauto }
366   \setquantitytype { sBm* } { left=\begin{Bsmallmatrix*} , right=\end{Bsmallmatrix*} , noauto }
367   \setquantitytype { spm* } { left=\begin{psmallmatrix*} , right=\end{psmallmatrix*} , noauto }
368   \setquantitytype { svm* } { left=\begin{vsmallmatrix*} , right=\end{vsmallmatrix*} , noauto }
369   \setquantitytype { sVm* } { left=\begin{Vsmallmatrix*} , right=\end{Vsmallmatrix*} , noauto }
370 }
371 \keys_set:nn { physicx/quantity }
372 {
373   left-size = \left ,
374   right-size = \right ,
375   type = p ,
376 }

\physicx_xquantity:nn
  \newxquantity
  \NewXQuantity
377 \cs_new:Npn \physicx_xquantity:nn #1#2
378 {
379   \group_begin:
380   \keys_set:nn { physicx/quantity } {#1}
381   \tl_if_empty:nF {#2} { \tl_set:Nn \l__physicx_quantity_code_tl {#2} }
382   \__physicx_xquantity_aux:oooo
383   { \l__physicx_quantity_left_tl }
384   { \l__physicx_quantity_args_tl }
385   { \l__physicx_quantity_code_tl }
386   { \l__physicx_quantity_right_tl }
387   \group_end:
388 }
389 \cs_new:Npn \__physicx_xquantity_aux:nnnn #1#2#3#4
390 {
391   \l__physicx_quantity_pre_tl
392   \bool_lazy_or:nnTF

```



```

393 { \tl_if_empty_p:N \l__physicx_quantity_left_size_tl }
394 { \tl_if_empty_p:N \l__physicx_quantity_right_size_tl }
395 { #1 #2 #3 #4 }
396 {
397   \bool_lazy_or:nnTF
398     { \token_if_eq_meaning_p:NN \l__physicx_quantity_left_size_tl \left }
399     { \token_if_eq_meaning_p:NN \l__physicx_quantity_right_size_tl \right }
400     { \physicx_left: #1 #2 #3 \physicx_right: #4 }
401     {
402       \physicx_left:N \l__physicx_quantity_left_size_tl #1 #2
403       #3
404       \physicx_right:N \l__physicx_quantity_right_size_tl #4
405     }
406   }
407   \l__physicx_quantity_post_tl
408 }
409 \NewDocumentCommand \xquantity { } { \physicx_xquantity:nn }
410 \cs_generate_variant:Nn \__physicx_xquantity_aux:nnnn { oooo }
411 \NewDocumentCommand \newxquantity { m o o m m }
412 {
413   \IfNoValueTF {#2}
414   {
415     \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
416       { \newcommand ##1 }
417   }
418   {
419     \IfNoValueTF {#3}
420     {
421       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
422         { \newcommand ##1 [#2] }
423     }
424     {
425       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
426         { \newcommand ##1 [#2] [#3] }
427     }
428   }
429   \exp_args:Nc \__physicx_new_xquantity_aux:w
430   { \cs_to_str:N #1~star }
431   { \physicx_xquantity:nn { #4 , noauto } {#5} }
432   \exp_args:Nc \__physicx_new_xquantity_aux:w
433   { \cs_to_str:N #1~unstar }
434   { \physicx_xquantity:nn { #4 } {#5} }
435   \exp_args:NNx \newcommand #1
436   {
437     \exp_not:N \@ifstar
438     \exp_not:c { \cs_to_str:N #1~star }
439     \exp_not:c { \cs_to_str:N #1~unstar }
440   }
441 }
442 \NewDocumentCommand \NewXQuantity { m m m m }
443 {
444   \NewDocumentCommand #1 { s #2 }
445   {
446     \IfBooleanTF {##1}

```

```

447         { \physicx_xquantity:nn { #3 , noauto } {#4} }
448         { \physicx_xquantity:nn { #3 } {#4} }
449     }
450 }
451 \NewXQuantity \qxqty { 0{} m } { #2 } {#3}
452 \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
453 \tl_new:N \physicx_tmp
454 \tl_new:N \l__physicx_cmd_noauto_body_tl
455 \bool_new:N \l__physicx_cmd_noauto_body_bool
456 \tl_new:N \l__physicx_cmd_auto_body_tl
457 \bool_new:N \l__physicx_cmd_auto_body_bool
458 \tl_new:N \l__physicx_cmd_arg_spec_tl
459 \int_new:N \l__physicx_cmd_arg_int
460 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
461 {
462     \tl_clear:N \l__physicx_cmd_noauto_body_tl
463     \tl_clear:N \l__physicx_cmd_auto_body_tl
464     \tl_clear:N \l__physicx_cmd_arg_spec_tl
465     \int_set:Nn \l__physicx_cmd_arg_int {#1}
466     \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
467     \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
468 }
469 % noauto, auto, cmd, body
470 \cs_new:Npn \physicx_declare_legacy_quantity:nnNn #1#2#3#4
471 {
472     \__physicx_declare_init:nnn { 3 } {#1} {#2}
473     \__physicx_declare_legacy_quantity_aux:nw #4
474     \q_recursion_tail \q_recursion_tail \q_recursion_stop
475     \__physicx_declare_legacy_quantity_aux:NcVVV
476     #3 { \cs_to_str:N #3 ~ body }
477     \l__physicx_cmd_arg_spec_tl
478     \l__physicx_cmd_noauto_body_tl
479     \l__physicx_cmd_auto_body_tl
480 }
481 % arg spec, pre, body to replace(start from #4), post
482 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
483 {
484     \int_incr:N \l__physicx_cmd_arg_int
485     \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
486         \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
487         \tl_set:Nx \l__physicx_tmp_tl
488         {
489             {
490                 \exp_not:N \tl_if_novalue_p:n
491                 {
492                     \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:
493                     \or: \or: \or:

```

```

494         \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
495         \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
496     \fi:
497 }
498 }
499 }
500 \if_bool:N \l__physicx_cmd_noauto_body_bool
501     \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
502     \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
503     {
504         {
505             % if is '.', use none
506             \str_if_eq:nnTF {#2} {.} {} {#2}
507             #3
508             \str_if_eq:nnTF {#4} {.} {} {#4}
509         }
510     }
511 \fi:
512 \if_bool:N \l__physicx_cmd_auto_body_bool
513     \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
514     \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
515     { { ##1 #2 #3 ##2 #4 } }
516 \fi:
517 \fi:
518 }
519 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
520 {
521     \quark_if_recursion_tail_stop:n {#1}
522     \quark_if_recursion_tail_stop:n {#2}
523     \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
524     \__physicx_declare_legacy_quantity_aux:nw
525 }
526 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
527 {
528     \__physicx_nauto_case:nnnn
529     { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
530     {
531         \cs_set_protected:Npn #1
532         {
533             \peek_charcode_ignore_spaces:NTF \let
534             { #2 } { #2 [ \physicx_left: ] \physicx_right: }
535         }
536         \DeclareDocumentCommand #2 { 0{##2} m s #3 }
537         {
538             \IfBooleanTF { ##3 }
539             { \bool_case_false:n {#4} }
540             { \bool_case_false:n {#5} }
541         }
542     }
543 {
544     \cs_set_protected:Npn #1
545     { #2 \c_empty_tl \c_empty_tl }
546     \DeclareDocumentCommand #2 { m m s #3 }
547     { \bool_case_false:n {#4} }

```

```

548     }
549   }
550   \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
551   \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
552   {
553     \bool_if:NTF \l__physicx_cmd_noauto_body_bool
554     {
555       \bool_if:NTF \l__physicx_cmd_auto_body_bool
556       {#1} {#2}
557     }
558     {
559       \bool_if:NTF \l__physicx_cmd_auto_body_bool
560       {#3} {#4}
561     }
562   }
563   \set_protected:Npn \@declarequantitycmd
564   { \physicx_declare_legacy_quantity:nnNn }

```

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

$\backslash$ quantity $\backslash$ evaluated $\backslash$ matrixquantity $\backslash$ smallmatrixquantity	Redefine some macros in physics package. <pre> 565 \if_bool:N \g__physicx_reqty_bool 566 \physicx_declare_legacy_quantity:nnNn 567 \c_true_bool \c_true_bool \quantity 568 { 569   { !g } { { \{ } { #4 } { \} } } 570   { !o } { { [ } { #5 } { ] } } 571   { !d() } { { ( } { #6 } { ) } } 572   { !d  } { { \vert } { #7 } { \vert } } 573   { !d&lt;&gt; } { { \langle } { #8 } { \rangle } } 574   { !d== } { { \Vert } { #9 } { \Vert } } 575 } 576 \physicx_declare_legacy_quantity:nnNn 577 \c_true_bool \c_true_bool \evaluated 578 { 579   { !g } { { . } { #4 \nobreak } { \vert } } 580   { !d[ ] } { { [ ] } { #5 \nobreak } { \vert } } 581   { !d( ) } { { ( ) } { #6 \nobreak } { \vert } } 582 } 583 \physicx_declare_legacy_quantity:nnNn 584 \c_true_bool \c_false_bool \matrixquantity 585 { 586   { !g } 587   { 588     { \IfBooleanT{#3}{\left\{ } } 589     { \begin{matrix} #4 \end{matrix} } 590     { \IfBooleanT{#3}{\right\} } 591   } 592   { !o } { { \begin{bmatrix} } { #5 } { \end{bmatrix} } } 593   { !d() } 594   { 595     { \IfBooleanTF{#3}{\left\lgroup}{\left( } } 596     { \begin{matrix} #6 \end{matrix} } </pre>
--	---

```

597         { \IfBooleanTF{#3}{\right\rgroup}{\right)}} }
598     }
599     { !d|| } { { \begin{vmatrix} } {#7} { \end{vmatrix} } } }
600     { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } } { \right\rangle } }
601     { !d== } { { \begin{Vmatrix} } {#9} { \end{Vmatrix} } } }
602 }
603 \physicsx_declare_legacy_quantity:nnNn
604 \c_true_bool \c_false_bool \smallmatrixquantity
605 {
606     { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } } { \right\} } }
607     { !o } { { \left[ } { \begin{smallmatrix} #5 \end{smallmatrix} } } { \right]} } }
608     { !d() }
609     {
610         { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
611         { \begin{smallmatrix} #6 \end{smallmatrix} } }
612         { \IfBooleanTF{#3}{\right\rgroup}{\right)}} }
613     }
614     { !d|| } { { \left\vert } { \begin{smallmatrix} #7 \end{smallmatrix} } } { \right\vert } }
615     { !d<> } { { \left\langle } { \begin{smallmatrix} #8 \end{smallmatrix} } } { \right\rangle } }
616     { !d== } { { \left\Vert } { \begin{smallmatrix} #9 \end{smallmatrix} } } { \right\Vert } }
617 }
618 \fi:

```

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

```

\physicsx_declare_legacy_paren:NnnnNnNn
\@declareparenccmd
619 %% cmd, arg spec, replace(start from #6), pre, left, right, post
620 \cs_new:Npn \physicsx_declare_legacy_paren:NnnnNnNn #1#2#3#4#5#6#7
621 {
622     \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
623     {
624         \bool_case_true:nF
625         {
626             { \bool_if_p:n {##2} } { #4 \physicsx_left:N \bigl #5 #3 \physicsx_right:N \bigr
627             { \bool_if_p:n {##3} } { #4 \physicsx_left:N \Bigl #5 #3 \physicsx_right:N \Bigr
628             { \bool_if_p:n {##4} } { #4 \physicsx_left:N \biggl #5 #3 \physicsx_right:N \biggr
629             { \bool_if_p:n {##5} } { #4 \physicsx_left:N \Biggl #5 #3 \physicsx_right:N \Biggr
630         }
631         {
632             \IfBooleanTF {##1}
633             { #4 #5 #3 #6 #7 }
634             { #4 \physicsx_left: #5 #3 \physicsx_right: #6 #7 }
635         }
636     }
637 }
638 \cs_set_protected:Npn \@declareparenccmd
639 { \physicsx_declare_legacy_paren:NnnnNnNn }

```

(End definition for \physicsx\_declare\_legacy\_paren:NnnnNnNn and \@declareparenccmd. These functions are documented on page ??.)

```

\qty Redefine some macros in physics package.
\mqty \if_bool:N \g__physicsx_reqty_bool
\smqty \physicsx_option_or:nnT { compat } { short }
\pqty {
\bqty
\vqty
\Bqty
\absolutevalue
\eval
\abs
\norm
\order
\oorder
\commutator

```

```

643 \cs_set:Npn \qty { \quantity }
644 \physics_declare_legacy_paren:NnnnNNn \pqty { m } {#6} { } { } { } { }
645 \physics_declare_legacy_paren:NnnnNNn \bqty { m } {#6} { } { } [ ] { }
646 \physics_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } { } \vert \vert { }
647 \physics_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } { } \{ \} { }
648 }
649 \physics_declare_legacy_paren:NnnnNNn \absolutevalue
650 { m } {#6} { } { } \vert \vert { }
651 \physics_option_or:nnT { compat } { short }
652 {
653 \cs_set:Npn \eval { \evaluated }
654 \cs_set:Npn \abs { \absolutevalue }
655 }
656 \physics_declare_legacy_paren:NnnnNNn \norm
657 { m } {#6} { } { } \lVert \rVert { }
658 \physics_compat:TF
659 {
660 \physics_declare_legacy_paren:NnnnNNn \order
661 { m } {#6} { } { } \c_physicx_Order_tl { } { }
662 }
663 {
664 \physics_declare_legacy_paren:NnnnNNn \order
665 { m } {#6} { } { } \c_physicx_order_tl { } { }
666 }
667 \physics_declare_legacy_paren:NnnnNNn \commutator
668 { m m } { #6 , #7 } { } { } [ ] { }
669 \physics_option_or:nnT { compat } { short }
670 { \cs_set:Npn \comm { \commutator } }
671 \physics_declare_legacy_paren:NnnnNNn \poissonbracket
672 { m m } { #6 , #7 } { } { } \{ \} { }
673 \physics_option_or:nnT { compat } { short }
674 {
675 \cs_set:Npn \pb { \poissonbracket }
676 \cs_set:Npn \anticommutator { \poissonbracket }
677 \cs_set:Npn \acomm { \poissonbracket }
678 }
679 \fi:
680 \physics_declare_legacy_paren:NnnnNNn \OOrder
681 { m } {#6} { } { } \c_physicx_Order_tl { } { }
682 \physics_declare_legacy_paren:NnnnNNn \oorder
683 { 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

```

684 \cs_new_nopar:Npn \__physicx_matrix_calc:nn #1#2
685 {
686 \int_set:Nn \l__physicx_matrix_rows_int
687 { \int_max:nn {#1} \l__physicx_matrix_rows_int }
688 \int_set:Nn \l__physicx_matrix_cols_int
689 { \int_max:nn {#2} \l__physicx_matrix_cols_int }
690 }

```

```

691 % use matrix element
692 \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
693 {
694   \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
695   \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
696   \else:
697     \exp_not:o { \physicxempty }
698   \fi:
699 }
700 % set matrix element, check or not
701 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
702 { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
703 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
704 {
705   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
706   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
707 }
708 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_cke:nnn #1#2#3
709 {
710   \tl_if_empty:nTF {#3}
711   { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
712   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
713 }
714 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
715 {
716   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
717   {
718     \tl_if_empty:nTF {#3}
719     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
720     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
721   }
722 }
723 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
724 \__physicx_matrix_set_r_c_ckigep:nnn
725 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
726 \__physicx_matrix_set_r_c_nock:nnn
727 % align, cr, sep symbol
728 \str_const:Nn \physicx@align { , }
729 \str_const:Nn \physicx@cr { ; }
730 \str_const:Nn \physicx@sep { , }
731 \bool_new:N \l__physicx_matrix_infinite_bool
732 \bool_new:N \l__physicx_matrix_dotrow_bool
733 \bool_new:N \l__physicx_matrix_dotcol_bool
734 \tl_new:N \l__physicx_matrix_array_tl
735 \tl_new:N \l__physicx_matrix_body_tl
736 \int_new:N \l__physicx_matrix_rows_int
737 \int_new:N \l__physicx_matrix_cols_int
738 \tl_new:N \l__physicx_matrix_main_tl
739 \clist_new:N \l__physicx_matrix_diag_clist
740 \clist_new:N \l__physicx_matrix_item_clist
741 \bool_new:N \l__physicx_matrix_diag_bool
742 \seq_new:N \l__physicx_row_list_seq
743 \seq_new:N \l__physicx_col_list_seq
744 % expand input

```

```

745 \cs_new_eq:NN \__physicx_expand:w \exp_not:o
746 %% main, row iterate, col iterate
747 \cs_new_nopar:Npn \physicx@matricxelement #1#2#3 { #1 \sb { #2 #3 } }
748 \cs_new_nopar:Npn \__physicx_matrix_row_iterate:n #1 { #1 }
749 \tl_new:N \l__physicx_matrix_last_row_tl
750 \tl_new:N \l__physicx_matrix_last_col_tl
751 \cs_new_nopar:Npn \__physicx_matrix_col_iterate:n #1 { #1 }
752 \cs_new_nopar:Npn \__physicx_matrix_begin:w { }
753 \cs_new_nopar:Npn \__physicx_matrix_end:w { }
754 \cs_new_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn
755 \bool_new:N \l__physicx_matrix_expand_element_bool
756 % when element is empty use \physicxempty
757 \tl_new:N \physicxempty
758 % save 'element-except' key's value
759 \tl_new:N \physicxexcept
760 \tl_new:N \l__physicx_matrix_args_tl
761 \tl_new:N \l__physicx_matrix_after_begin_tl
762 \tl_new:N \l__physicx_matrix_after_end_tl
763 \bool_new:N \l__physicx_matrix_transpose_bool
764 \bool_new:N \l__physicx_matrix_enhanced_bool
765 \dim_new:N \l__physicx_matrix_sep_dim
766 \cs_new:Npn \__physicx_adi:nnn #1#2#3 { #1#2#3 }
767 \tl_new:N \l__physicx_matrix_beginning_tl
768 \tl_new:N \l__physicx_matrix_ending_tl

```

### 1.3.2 Matrix keys

```

769 \keys_define:nn { physicx }
770 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
771 \keys_define:nn { physicx/matrix }
772 {
773   array .tl_set:N = \l__physicx_matrix_array_tl ,
774   expand .choice: ,
775   expand / none .code:n =
776     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
777   expand / text-expand .code:n =
778     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
779   expand / f .code:n =
780     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
781   expand / romanual .meta:n = { expand = f } ,
782   expand / x .code:n =
783     \cs_set_eq:NN \__physicx_expand:w \use:n ,
784   expand / edef .meta:n = { expand = x } ,
785   rows .int_set:N = \l__physicx_matrix_rows_int ,
786   cols .int_set:N = \l__physicx_matrix_cols_int ,
787   auto-update .choice: ,
788   auto-update / true .code:n =
789     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,
790   auto-update / false .code:n =
791     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
792   auto-update .default:n = true ,
793   main .tl_set:N = \l__physicx_matrix_main_tl ,
794   row-list .code:n =
795     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,
796   col-list .code:n =

```



```

797 \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
798 infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
799 infinite .default:n = true ,
800 !infinite .code:n =
801 \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
802 element-code .cs_set:Np = \physicx@matricelement #1#2#3 ,
803 element-code* .choice: ,
804 element-code* / except-empty .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 \tl_if_empty:nTF {##1}
810 {##1}
811 { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
812 } ,
813 element-code* / except-dots .code:n =
814 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
815 \physicx@matricelement
816 \cs_set:Npn \physicx@matricelement ##1##2##3
817 {
818 \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
819 {##1}
820 { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
821 } ,
822 element-code* / except-tl .code:n =
823 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
824 \physicx@matricelement
825 \cs_set:Npn \physicx@matricelement ##1##2##3
826 {
827 \tl_if_in:onTF { \physicxexcept } {##1}
828 {##1}
829 { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
830 } ,
831 element-code* / except-regex .code:n =
832 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
833 \physicx@matricelement
834 \cs_set:Npn \physicx@matricelement ##1##2##3
835 {
836 \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
837 {##1}
838 { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
839 } ,
840 element-code* / only-regex .code:n =
841 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
842 \physicx@matricelement
843 \cs_set:Npn \physicx@matricelement ##1##2##3
844 {
845 \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
846 { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
847 {##1}
848 } ,
849 element-code* / unknown .code:n =
850 \cs_set:Npx \physicx@matricelement { \exp_not:c {#1} } ,

```

```

851 element-except .tl_set:N = \physicsexcept ,
852 element-except+ .code:n =
853   \tl_put_right:Nn \physicsexcept {#1} ,
854 expand-element .bool_set:N = \l__physicx_matrix_expand_element_bool ,
855 expand-element .default:n = true ,
856 empty .tl_set:N = \physicxempty ,
857 check .choice: ,
858 check / none .code:n =
859   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
860   \__physicx_matrix_set_r_c_nock:nnn ,
861 check / empty .code:n =
862   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
863   \__physicx_matrix_set_r_c_cke:nnn ,
864 check / ignore .code:n =
865   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
866   \__physicx_matrix_set_r_c_ckig:nnn ,
867 check / igep .code:n =
868   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
869   \__physicx_matrix_set_r_c_ckigep:nnn ,
870 check / all .code:n =
871   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
872   \__physicx_matrix_set_r_c_ckall:nnn ,
873 check .default:n = all ,
874 row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
875 col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
876 last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
877 last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
878 diag .clist_set:N = \l__physicx_matrix_diag_clist ,
879 diag+ .code:n =
880   \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
881 diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
882 diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
883 diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
884 item .clist_set:N = \l__physicx_matrix_item_clist ,
885 item+ .code:n =
886   \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
887 item-now .code:n = \physicx_matrix_item_parse:n {#1} ,
888 item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
889 item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
890 check-range .choice: ,
891 check-range / true .code:n = \physicx_parse_range_check: ,
892 check-range / false .code:n = \physicx_parse_range_noccheck: ,
893 check-range .default:n = true ,
894 begin .tl_set:N = \__physicx_matrix_begin:w ,
895 end .tl_set:N = \__physicx_matrix_end: ,
896 args .code:n =
897   \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,
898 args* .tl_set:N = \l__physicx_matrix_args_tl ,
899 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
900 after-begin+ .code:n =
901   { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
902 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
903 after-end+ .code:n =
904   { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,

```

```

905 sepdim .dim_set:N = \l__physicx_matrix_sep_dim ,
906 type .multichoice: ,
907 saveto .tl_set:N = \l__physicx_matrix_save_tl ,
908 saveto* .code:n =
909   \tl_set:N \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
910 transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
911 transpose .default:n = true ,
912 ' .meta:n = { transpose = true } ,
913 T .meta:n = { transpose = true } ,
914 MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
915 enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
916 enhanced .default:n = true ,
917 !enhanced .code:n =
918   \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
919 cr .tl_set:N = \physicx@cr ,
920 align .tl_set:N = \physicx@align ,
921 sep .tl_set:N = \physicx@sep ,
922 adi-order .choice: ,
923 adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
924 adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
925 adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
926 adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
927 adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
928 adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
929 beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
930 beginning+ .code:n =
931   \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
932 ending .tl_set:N = \l__physicx_matrix_ending_tl ,
933 ending+ .code:n =
934   \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
935
936 settype .code:n = \setmatrixtype #1 ,
937
938 unknown .code:n =
939   \physicx_search_also:nnF
940   {
941     physicx/matrix/type ,
942     physicx/matrix/expand ,
943     physicx/matrix/element-code* ,
944   }
945   {#1}
946   {
947     \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
948     {
949       \keys_set:nx { physicx/matrix }
950       { MaxMatrixCols = \l_keys_key_str }
951     }
952     {
953       \msg_error:nnxx { physicx } { unknown-key }
954       \l_keys_path_str { physicx/matrix }
955     }
956   } ,
957 }

```

```

\physicx_matrix_new_type:nnn
\physicx_matrix_new_type:nn
\setmatrixtype
958 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
959 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
960 \cs_new:Npn \physicx_matrix_new_type:nn
961 { \physicx_new_type:nnn { matrix } }
962 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
963 {
964   \IfBooleanTF {#1}
965     { \physicx_matrix_new_type:nn {#2} }
966     { \physicx_matrix_new_type:nnn {#2} }
967 }

```

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

A few types.

```

968 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
969 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
970 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
971 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
972 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
973 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
974 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
975 \physicx_mathtools:T
976 {
977   \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
978   \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
979   \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
980   \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
981   \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
982   \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
983   \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
984   \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
985   \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
986   \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
987   \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
988   \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
989   \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
990   \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
991   \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
992   \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
993   \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
994 }

```

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

```

995 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
996 { \clist_set:cn { physicx@ #1 data@ #2 } }
997 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
998 {
999   \clist_clear:c { l__physicx_matrix_ #1 _clist }
1000   \__physicx_matrix_add_data:nn {#1} {#2}
1001 }
1002 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
1003 {
1004   \clist_map_inline:nn {#2}

```

```

1005     {
1006       \clist_concat:ccc
1007       { l__physicx_matrix_ #1 _clist }
1008       { l__physicx_matrix_ #1 _clist }
1009       { physicx@ #1 data@ #2 }
1010     }
1011   }

```

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

Initial settings.

```

1012 \keys_set:nn { physicx/matrix }
1013 {
1014   type = m ,
1015   saveto = ? ,
1016 }

```

\qxmatrix

```

1017 %% basically, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
xmatmmn-in-the-physics-package, but changed some
1018 % #1 = boolean, saveto matrix
1019 % #2 = star, infinite
1020 % #3 = options
1021 % #4 = letter for the entries
1022 % #5 = number of rows
1023 % #6 = number of explicit rows, default = 3
1024 % #7 = number of columns
1025 % #8 = number of explicit columns, default = 3
1026 \DeclareDocumentCommand \qxmatrix { t= s 0{type=p} m m 0{3} m 0{3} }
1027 {
1028   \group_begin:
1029   \IfBooleanTF { #2 }
1030     { \bool_set_true:N \l__physicx_matrix_infinite_bool }
1031     { \bool_set_false:N \l__physicx_matrix_infinite_bool }
1032   \int_set:Nn \l__physicx_matrix_rows_int {#6}
1033   \int_set:Nn \l__physicx_matrix_cols_int {#8}
1034   \IfBooleanTF {#1}
1035     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1036     { \keys_set:nn { physicx/matrix } {#3} }
1037   \physicx_qxmatrix:nnn {#4} {#5} {#7}
1038   \__physicx_matrix_save_or_print:
1039   \group_end:
1040 }
1041 \cs_new_protected:Nn \physicx_qxmatrix:nnn
1042 {
1043   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1044     {
1045       \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1046       \__physicx_matrix_appto_body_e:nnn
1047     }
1048     {
1049       \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1050       \__physicx_matrix_appto_body_ne:nnn
1051     }
1052   % clear the variable containing the body of the matrix

```

```

1053 \tl_clear:N \l__physicx_matrix_body_tl
1054 % set the tentative number of explicit rows
1055 \physicx_if_num:nTF { #2 }
1056 { % number of rows is an integer
1057   \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
1058   { % if #2 <= rows, we don't want a row of dots
1059     \bool_set_false:N \l__physicx_matrix_dotrow_bool
1060     \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1061   }
1062   { % we want a row of dots
1063     \bool_set_true:N \l__physicx_matrix_dotrow_bool
1064   }
1065 }
1066 { % number of rows is symbolic, we want a row of dots
1067   \bool_set_true:N \l__physicx_matrix_dotrow_bool
1068 }
1069 % set the tentative number of explicit columns
1070 \physicx_if_num:nTF { #3 }
1071 { % number of cols is an integer
1072   \int_compare:nTF { #3 <= \l__physicx_matrix_cols_int }
1073   { % if #3 <= cols, we don't want a column of dots
1074     \bool_set_false:N \l__physicx_matrix_dotcol_bool
1075     \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1076   }
1077   { % we want a column of dots
1078     \bool_set_true:N \l__physicx_matrix_dotcol_bool
1079   }
1080 }
1081 { % number of columns is symbolic, we want a column of dots
1082   \bool_set_true:N \l__physicx_matrix_dotcol_bool
1083 }
1084 % loop through the rows
1085 \int_step_inline:nn { \l__physicx_matrix_rows_int }
1086 {
1087   % add the first entry in the row
1088   %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1089   \__physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
1090   % add the further entries in the explicit columns
1091   \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1092   {
1093     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
1094     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1095     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
1096   }
1097   % if we have a column of dots, add \cdots and the last entry
1098   \bool_if:NT \l__physicx_matrix_dotcol_bool
1099   {
1100     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{##1 #3} }
1101     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1102     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {#3}
1103   }
1104   % infinite matrix, add \cdots
1105   \bool_if:NT \l__physicx_matrix_infinite_bool
1106   { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }

```

```

1107         \if_int_compare:w ##1 = \l__physicx_matrix_rows_int
1108         \scan_stop:
1109     \else:
1110         % finish up the row
1111         \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1112     \fi:
1113 }
1114 % finish up the rows
1115 \bool_if:NT \l__physicx_matrix_dotrow_bool
1116 {
1117     % finish up the row
1118     \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1119     % if we have a row of dots, fill it in
1120     \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1121     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1122         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1123     \bool_if:NT \l__physicx_matrix_dotcol_bool
1124         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
1125     \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1126     % fill the last row
1127     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
1128     \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
1129     \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1130     {
1131         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
1132         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1133         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
1134     }
1135     \bool_if:NT \l__physicx_matrix_dotcol_bool
1136     {
1137         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
1138         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1139         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
1140     }
1141     % if the matrix is infinite, add a further column with \cdots
1142     \bool_if:NT \l__physicx_matrix_infinite_bool
1143         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1144 }
1145 % if the matrix is infinite, add a final row
1146 \bool_if:NT \l__physicx_matrix_infinite_bool
1147 {
1148     % finish up the row
1149     \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1150     \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1151     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1152         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1153     \bool_if:NT \l__physicx_matrix_dotcol_bool
1154         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1155     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1156     % update cols
1157     \bool_if:NTF \l__physicx_matrix_dotcol_bool
1158         { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1159         { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1160 }

```

1161 }

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

```
\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
1162 \cs_new:Npn \physicx_matrix_diag_parse:n #1
1163 {
1164   \keyval_parse:nnn
1165   \__physicx_matrix_diag_parse_aux:n
1166   \__physicx_matrix_diag_parse_aux:nn
1167   {#1}
1168 }
1169 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
1170 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
1171 {
1172   \str_case:e:nnF {#1}
1173   {
1174     { auto-update }
1175     {
1176       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
1177       \__physicx_matrix_calc:nn
1178     }
1179     { noauto-update }
1180     {
1181       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
1182     }
1183     { true }
1184     {
1185       \bool_set_true:N \l__physicx_matrix_diag_bool
1186       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1187       \__physicx_diagonalmatrix_set_diag:
1188     }
1189     { false }
1190     {
1191       \bool_set_false:N \l__physicx_matrix_diag_bool
1192       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1193       \__physicx_diagonalmatrix_no_diag:
1194     }
1195   }
1196   { \msg_error:nnn { physicx } { diag-key } {#1} }
1197 }
1198 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
1199 {
1200   \tl_set:Nn \l__physicx_tmpdiag_tl {#2}
1201   \tl_set:Nx \l__physicx_tmpdiag_tl
1202   { \__physicx_expand:w \l__physicx_tmpdiag_tl }
1203   \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl
1204   \tl_if_head_eq_charcode:nNTF {#1} '
1205   {
1206     \exp_args:Nf \__physicx_matrix_diag_parse_aux_anti:n
1207     { \tl_tail:n {#1} }
1208   }
1209   { \__physicx_matrix_diag_parse_aux_regu:n {#1} }
1210 }
```



```

1211 \cs_new:Npn \__physicx_diagonalmatrix_set_diag:
1212 {
1213   \int_zero:N \l__physicx_matrix_cols_int
1214   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1215   {
1216     \int_incr:N \l__physicx_matrix_cols_int
1217     \physicx_matrix_set_r_c:nnn {##1} {##1} {##2}
1218   }
1219   \int_set_eq:NN \l__physicx_matrix_rows_int
1220   \l__physicx_matrix_cols_int
1221 }
1222 \cs_new:Npn \__physicx_diagonalmatrix_no_diag:
1223 {
1224   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1225   { \physicx_matrix_set_r_c:nnn {##1} {##1} {##2} }
1226   \__physicx_matrix_diag_calc:nn
1227   { \seq_count:N \l__physicx_tmpdiag_seq }
1228   { \seq_count:N \l__physicx_tmpdiag_seq }
1229 }
1230 \cs_new_eq:NN \__physicx_diagonalmatrix_diag_main:
1231 \__physicx_diagonalmatrix_no_diag:
1232 \cs_new:Npn \__physicx_matrix_diag_parse_aux_regu:n #1
1233 {
1234   \if_int_compare:w #1 = 0 \exp_stop_f:
1235     \__physicx_diagonalmatrix_diag_main:
1236   \else:
1237     \if_int_compare:w #1 > 0 \exp_stop_f:
1238       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1239       {
1240         \physicx_matrix_set_r_c:nnn
1241         {##1} { \int_eval:n { ##1 + #1 } } {##2}
1242       }
1243       \__physicx_matrix_diag_calc:nn
1244       { \seq_count:N \l__physicx_tmpdiag_seq }
1245       { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1246     \else:
1247       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1248       {
1249         \physicx_matrix_set_r_c:nnn
1250         { \int_eval:n { ##1 - #1 } } {##1} {##2}
1251       }
1252       \__physicx_matrix_diag_calc:nn
1253       { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1254       { \seq_count:N \l__physicx_tmpdiag_seq }
1255     \fi:
1256   \fi:
1257 }
1258 \cs_new:Npn \__physicx_matrix_diag_parse_aux_anti:n #1
1259 {
1260   \if_int_compare:w #1 = 0 \exp_stop_f:
1261     \__physicx_matrix_diag_calc:nn
1262     { \seq_count:N \l__physicx_tmpdiag_seq }
1263     { \seq_count:N \l__physicx_tmpdiag_seq }
1264   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq

```

```

1265     {
1266         \physicx_matrix_set_r_c:nnn
1267         {##1}
1268         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1269         {##2}
1270     }
1271 \else:
1272 \if_int_compare:w #1 > 0 \exp_stop_f:
1273 \__physicx_matrix_diag_calc:nn
1274 { \seq_count:N \l__physicx_tmpdiag_seq }
1275 { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1276 \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1277 {
1278     \physicx_matrix_set_r_c:nnn
1279     {##1}
1280     { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1281     {##2}
1282 }
1283 \else:
1284 \__physicx_matrix_diag_calc:nn
1285 { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1286 { \seq_count:N \l__physicx_tmpdiag_seq }
1287 \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1288 {
1289     \physicx_matrix_set_r_c:nnn
1290     { \int_eval:n { ##1 - #1 } }
1291     { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1292     {##2}
1293 }
1294 \fi:
1295 \fi:
1296 }
1297 \cs_new:Npn \__physicx_matrix_diag_calc:nn
1298 { \__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
1299 \cs_new:Npn \physicx_matrix_item_parse:n #1
1300 {
1301     \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1302     \keyval_parse:NNn
1303     \__physicx_matrix_item_parse_aux:n
1304     \__physicx_matrix_item_parse_aux:nn
1305     {#1}
1306 }
1307 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }
1308 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1309 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2
1310 {
1311     \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1312     \tl_set:Nx \l__physicx_tmpitem_tl
1313     { \__physicx_expand:w \l__physicx_tmpitem_tl }
1314     \physicx_parse_range:nxN \l__physicx_matrix_rows_int

```

```

1315     { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1316 \physicx_parse_range:nxN \l__physicx_matrix_cols_int
1317     { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1318 \exp_args:No \tl_if_eq:nnTF
1319     { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1320     {
1321         \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1322         {
1323             \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1324             {
1325                 \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1326             }
1327         }
1328     }
1329     {
1330         \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1331         {
1332             \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1333             {
1334                 \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }
1335                 {
1336                     \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1337                     {##1} {####1} { \l__physicx_tmpitem_tl }
1338                 }
1339             }
1340         }
1341     }
1342 }

```

(End definition for \physicx\_matrix\_item\_parse:n. This function is documented on page ??.)

\physicx\_matrix\_array\_parse:n  
\physicx\_matrix\_array\_parse:o

Parse ‘array..’ keys.

```

1343 \cs_new:Npn \physicx_matrix_array_parse:n #1
1344 {
1345     \tl_set:Nn \l__physicx_tmparr_tl {#1}
1346     \tl_set:Nx \l__physicx_tmparr_tl
1347     { \__physicx_expand:w \l__physicx_tmparr_tl }
1348     \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl
1349     \__physicx_matrix_autocalc:nn
1350     { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1351     { 0 }
1352     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1353     {
1354         \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1355         \__physicx_matrix_autocalc:nn
1356         { 0 }
1357         { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1358         \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1359         {
1360             \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1361         }
1362     }
1363 }
1364 \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.

```

1365 \cs_new:Npn \physicx_matrix_array_parse_main:
1366 {
1367   \int_step_inline:nn \l__physicx_matrix_rows_int
1368   {
1369     \int_step_inline:nn \l__physicx_matrix_cols_int
1370     {
1371       \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1372       {##1} {####1} \l__physicx_matrix_main_tl
1373     }
1374   }
1375 }
```

(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).

```

1376 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1377 {
1378   \physicx_if_num:nTF {#1}
1379   { \prg_return_true: }
1380   {
1381     \bool_case_true:nTF
1382     {
1383       { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1384       { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1385       {
1386         \bool_lazy_and_p:nn
1387         { \cs_if_exist_p:N \inteval }
1388         { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1389       } { }
1390       {
1391         \bool_lazy_and_p:nn
1392         { \cs_if_exist_p:N \fpeval }
1393         { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1394       } { }
1395     }
1396     { \prg_return_true: }
1397     { \prg_return_false: }
1398   }
1399 }
```

(End definition for \\_\_physicx\_if\_can\_num:n.)

\diagonalmatrix Define \diagonalmatrix.

```

1400 \DeclareDocumentCommand \diagonalmatrix { t= t+ 0{} m }
1401 {
1402   \group_begin:
1403   \IfBooleanTF {#1}
1404   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicx_tmp } }
1405   { \keys_set:nn { physicx/matrix } { #3 } }
1406   \physicx_construct:nnn { }
```

```

1407 {
1408   \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist
1409   \tl_if_empty:nF {#4}
1410   {
1411     \__physicx_if_keyval:nTF {#4}
1412     { \physicx_matrix_diag_parse:n { true, #4 } }
1413     { \physicx_matrix_diag_parse:n { true, 0 = {#4} } }
1414   }
1415 }
1416 { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1417 \bool_lazy_or:nnTF
1418 { \bool_if_p:n {#2} }
1419 { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1420 {
1421   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1422   {
1423     \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1424     \__physicx_matrix_appto_body_e:off
1425   }
1426   {
1427     \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1428     \__physicx_matrix_appto_body_ne:off
1429   }
1430   \use_i_ii:nnn
1431 }
1432 { \use_i:nn }
1433 \__physicx_matrix_transpose:N
1434   \__physicx_diagonalmatrix_generate_enhanced_body:NNN
1435   \__physicx_diagonalmatrix_generate_body:NNN
1436 \__physicx_matrix_save_or_print:
1437 \group_end:
1438 }
1439 \cs_new:Npn \__physicx_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1440 {
1441   \__physicx_matrix_generate_body:NNNN #1#2#3
1442   \__physicx_diagonalmatrix_enhanced:nnn
1443 }
1444 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1445 {
1446   \int_step_inline:nn { #1 - 1 }
1447   {
1448     \int_step_inline:nn { #2 - 1 }
1449     {
1450       \tl_put_right:Nx \l__physicx_matrix_body_tl
1451       {
1452         \exp_after:wN
1453         \physicx_matrix_use_r_c:nn
1454         #3 {{##1}} {{####1}} &
1455       }
1456     }
1457   }
1458   \tl_put_right:Nx \l__physicx_matrix_body_tl
1459   {
1460     \exp_after:wN
1461     \physicx_matrix_use_r_c:nn

```

```

1461         #3 {{##1}} {{ \int_use:N #2 }} \[\dim_use:N \l__physicx_matrix_sep_dim]
1462     }
1463 }
1464 \int_step_inline:nn { #2 - 1 }
1465 {
1466     \tl_put_right:Nx \l__physicx_matrix_body_tl
1467     {
1468         \exp_after:wN
1469         \physicx_matrix_use_r_c:nn
1470         #3 {{ \int_use:N #1 }} {{##1}} &
1471     }
1472 }
1473 \tl_put_right:Nx \l__physicx_matrix_body_tl
1474 {
1475     \exp_after:wN
1476     \physicx_matrix_use_r_c:nn
1477     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1478 }
1479 }

```

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

\\_\_physicx\_declare\_init:

```

1480 \cs_new:Npn \__physicx_matrix_enhanced_init:
1481 {
1482     \seq_if_empty:NF \l__physicx_row_list_seq
1483     {
1484         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1485         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1486         { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1487     }
1488     \seq_if_empty:NF \l__physicx_col_list_seq
1489     {
1490         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1491         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1492         { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1493     }
1494 }

```

(End definition for \\_\_physicx\_declare\_init:.)

\commamatrix Define \commamatrix.

```

1495 \DeclareDocumentCommand \commamatrix { t= t+ 0 } m {
1496     {
1497         \group_begin:
1498         \keys_set:nn { physicx/matrix } {#3}
1499         \tl_if_empty:NF {#4}
1500         { \keys_set:nn { physicx/matrix } { array = {#4} } }
1501         \IfBooleanT {#1}
1502         { \keys_set:nn { physicx/matrix } { saveto = \physicx_tmp } }
1503         \tl_set:Nx \l__physicx_matrix_array_tl
1504         { \__physicx_expand:w \l__physicx_matrix_array_tl }
1505         \bool_lazy_or:nnTF
1506         { \bool_if_p:n {#2} }
1507         { \bool_if_p:N \l__physicx_matrix_enhanced_bool }

```

```

1508 { \l__physicx_commamatrix_enhanced: }
1509 {
1510   \tl_replace_all:Nox \l__physicx_matrix_array_tl
1511   { \physicx@cr } { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1512   \tl_replace_all:Non \l__physicx_matrix_array_tl
1513   { \physicx@align } { & }
1514   \tl_set_eq:NN \l__physicx_matrix_body_tl
1515   \l__physicx_matrix_array_tl
1516 }
1517 \l__physicx_matrix_save_or_print:
1518 \group_end:
1519 }
1520 \cs_new_nopar:Npn \l__physicx_matrix_save_or_print:
1521 {
1522   \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1523   {
1524     \exp_after:wN \tl_gset_eq:NN
1525     \l__physicx_matrix_save_tl
1526     \l__physicx_matrix_body_tl
1527   }
1528   {
1529     \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1530     \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1531     \fi:
1532     \exp_after:wN \l__physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matrix_body_tl
1533     \l__physicx_matrix_body_tl
1534     \l__physicx_matrix_end: \l__physicx_matrix_after_end_tl
1535   }
1536 }
1537 \cs_new:Npn \l__physicx_commamatrix_enhanced:
1538 {
1539   \tl_clear:N \l__physicx_matrix_body_tl
1540   \int_zero:N \l__physicx_tmpa_int
1541   \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1542   \l__physicx_matrix_array_tl
1543   \int_set:Nn \l__physicx_matrix_rows_int
1544   { \seq_count:N \l__physicx_tmp_seq }
1545   \l__physicx_matrix_enhanced_init:
1546   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1547   {
1548     \seq_map_tokens:Nn \l__physicx_tmp_seq
1549     {
1550       \int_incr:N \l__physicx_tmpa_int
1551       \exp_args:NV \l__physicx_commamatrix_enhanced_aux:nNn
1552       \l__physicx_tmpa_int \l__physicx_commamatrix_enhanced_aux_e:nnn
1553     }
1554   }
1555   {
1556     \seq_map_tokens:Nn \l__physicx_tmp_seq
1557     {
1558       \int_incr:N \l__physicx_tmpa_int
1559       \exp_args:NV \l__physicx_commamatrix_enhanced_aux:nNn
1560       \l__physicx_tmpa_int \l__physicx_commamatrix_enhanced_aux_ne:nnn
1561     }
1562   }

```

```

1562     }
1563   }
1564   \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1565   {
1566     \seq_set_split:Non \l__physicx_tmp_col_seq
1567     { \physicx@align } {#3}
1568     \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1569     \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1570     { #2 {##2} {#1} {##1} }
1571     \tl_put_right:Nx \l__physicx_matrix_body_tl
1572     {
1573       \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1574       \if_int_compare:w \l__physicx_matrix_rows_int = #1
1575       \scan_stop:
1576       \else:
1577         \[\dim_use:N \l__physicx_matrix_sep_dim]
1578       \fi:
1579     }
1580   }
1581   \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1582   {
1583     \seq_put_right:Nx \l__physicx_tmp_coled_seq
1584     {
1585       \text_expand:n % \text_expand:n do the magic thing, but slower
1586       {
1587         \physicx@matricelement { #1 }
1588         { \__physicx_matrix_row_iterate:n {#2} }
1589         { \__physicx_matrix_col_iterate:n {#3} }
1590       }
1591     }
1592   }
1593   \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1594   {
1595     \seq_put_right:No \l__physicx_tmp_coled_seq
1596     {
1597       \physicx@matricelement {#1}
1598       { \__physicx_matrix_row_iterate:n {#2} }
1599       { \__physicx_matrix_col_iterate:n {#3} }
1600     }
1601   }

```

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

**\generalmatrix** Define \generalmatrix.

```

1602 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1603 {
1604   \IfBooleanTF {#2}
1605   {
1606     \group_begin:
1607     \IfBooleanTF {#1}
1608     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicx_tmp } }
1609     { \keys_set:nn { physicx/matrix } {#4} }
1610     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1611     \physicx_construct:nnn

```



```

1612     {
1613         \tl_if_empty:NTF \l__physicx_matrix_main_tl
1614         {
1615             \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1616         }
1617         { \physicx_matrix_array_parse_main: }
1618     }
1619     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1620     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1621     \__physicx_generalmatrix:
1622     \__physicx_matrix_save_or_print:
1623     \group_end:
1624 }
1625 {
1626     \IfBooleanTF {#1}
1627     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1628     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1629     \qxmtrx = * [#4]
1630 }
1631 }
1632 \cs_new:Npn \__physicx_generalmatrix:
1633 {
1634     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1635     {
1636         \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1637         \__physicx_matrix_appto_body_e:off
1638     }
1639     {
1640         \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1641         \__physicx_matrix_appto_body_ne:off
1642     }
1643     \__physicx_matrix_transpose:N
1644     \__physicx_matrix_generate_body:NNNN
1645     \__physicx_generalmatrix_generate:nnn
1646 }

```

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

`\__physicx_matrix_generate_body:NNNN`

```

1647 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1648 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1649 {
1650     \__physicx_matrix_enhanced_init:
1651     \int_step_inline:nn { #1 - 1 }
1652     {
1653         \int_step_inline:nn { #2 - 1 }
1654         {
1655             \tl_set:Nx \l__physicx_tmp_tl
1656             {
1657                 \exp_after:wN
1658                 \physicx_matrix_use_r_c:nn
1659                 #3 {{##1}} {{####1}}
1660             }
1661             #4 \l__physicx_tmp_tl {##1} {####1}

```

```

1662         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1663     }
1664     \tl_set:Nx \l__physicx_tmp_tl
1665     {
1666         \exp_after:wN
1667         \physicx_matrix_use_r_c:nn
1668         #3 {{##1}} {{ \int_use:N #2 }}
1669     }
1670     #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1671     \tl_put_right:Nx \l__physicx_matrix_body_tl
1672     { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1673 }
1674 \int_step_inline:nn { #2 - 1 }
1675 {
1676     \tl_set:Nx \l__physicx_tmp_tl
1677     {
1678         \exp_after:wN
1679         \physicx_matrix_use_r_c:nn
1680         #3 {{ \int_use:N #1 }} {{##1}}
1681     }
1682     #4 \l__physicx_tmp_tl { \int_use:N #1 } {{##1}}
1683     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1684 }
1685 \tl_set:Nx \l__physicx_tmp_tl
1686 {
1687     \exp_after:wN
1688     \physicx_matrix_use_r_c:nn
1689     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1690 }
1691 #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1692 }

```

(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
1693 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1694 {
1695     \tl_put_right:Nx \l__physicx_matrix_body_tl
1696     {
1697         \text_expand:n
1698         {
1699             \physicx@matricelement {#1}
1700             { \__physicx_matrix_row_iterate:n {#2} }
1701             { \__physicx_matrix_col_iterate:n {#3} }
1702         }
1703     }
1704 }
1705 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1706 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1707 {
1708     \tl_put_right:No \l__physicx_matrix_body_tl
1709     {
1710         \physicx@matricelement {#1}
1711         { \__physicx_matrix_row_iterate:n {#2} }

```

```

1712         { \_physicx_matrix_col_iterate:n {#3} }
1713     }
1714 }
1715 \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

```

1716 \cs_new:Npn \_physicx_matrix_transpose:N #1 % generate body command
1717 {
1718     \bool_if:NTF \l__physicx_matrix_transpose_bool
1719     {
1720         #1
1721         \l__physicx_matrix_cols_int
1722         \l__physicx_matrix_rows_int
1723         \use_ii_i:nn
1724     }
1725     {
1726         #1
1727         \l__physicx_matrix_rows_int
1728         \l__physicx_matrix_cols_int
1729         \use:nn
1730     }
1731 }

```

(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.

```

1732 \cs_new:Npn \physicx_construct:nnn #1#2#3
1733 {
1734     \l__physicx_matrix_beginning_tl
1735     \_physicx_adi:nnn {#1} {#2} {#3}
1736     \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1737     {
1738         \int_incr:N \l__physicx_matrix_cols_int
1739         \_physicx_matrix_last_aux_c:
1740         \int_incr:N \l__physicx_matrix_cols_int
1741     }
1742     \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1743     {
1744         \int_incr:N \l__physicx_matrix_rows_int
1745         \_physicx_matrix_last_aux_r:
1746         \int_incr:N \l__physicx_matrix_rows_int
1747     }
1748     \bool_lazy_or:nnF
1749     { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1750     { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1751     {
1752         \physicx_matrix_set_r_c:nnn
1753         { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1754         { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1755         { \ddots }
1756     }

```

```

1757 \bool_if:NT \l__physicx_matrix_infinite_bool
1758 {
1759   \int_incr:N \l__physicx_matrix_rows_int
1760   \int_incr:N \l__physicx_matrix_cols_int
1761   \__physicx_matrix_last_aux_c:
1762   \__physicx_matrix_last_aux_r:
1763   \physicx_matrix_set_r_c:nnn
1764   { \int_use:N \l__physicx_matrix_rows_int }
1765   { \int_use:N \l__physicx_matrix_cols_int }
1766   { \ddots }
1767 }
1768 \l__physicx_matrix_ending_tl
1769 }
1770 \cs_new:Npn \__physicx_matrix_last_aux_c:
1771 {
1772   \int_step_inline:nn \l__physicx_matrix_rows_int
1773   {
1774     \physicx_matrix_set_r_c:nnn
1775     {##1} { \int_use:N \l__physicx_matrix_cols_int }
1776     { \cdots }
1777   }
1778 }
1779 \cs_new:Npn \__physicx_matrix_last_aux_r:
1780 {
1781   \int_step_inline:nn \l__physicx_matrix_cols_int
1782   {
1783     \physicx_matrix_set_r_c:nnn
1784     { \int_use:N \l__physicx_matrix_rows_int } {##1}
1785     { \vdots }
1786   }
1787 }

```

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

### 1.3.3 Define new matrix command

```

\__physicx_new_matrix_cmd:NNN
\newgeneralmatrix 1788 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
\NewGeneralMatrix 1789 {
\newdiagonalmatrix 1790   \NewDocumentCommand #2 { t+ m o o m m }
\NewDiagonalMatrix 1791   {
\newcommamatrix 1792     \IfBooleanTF {##1}
\NewCommaMatrix 1793     {
1794       \IfNoValueTF {##3}
1795       { \newcommand ##2 { #1 + [##5] {##6} } }
1796       {
1797         \IfNoValueTF {##4}
1798         { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1799         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1800       }
1801     }
1802   }
1803   \IfNoValueTF {##3}
1804   { \newcommand ##2 { #1 [##5] {##6} } }

```

```

1805         {
1806             \IfNoValueTF {##4}
1807             { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1808             { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1809         }
1810     }
1811 }
1812 \NewDocumentCommand #3 { t+ m m m m }
1813 {
1814     \IfBooleanTF {##1}
1815     { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1816     { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1817 }
1818 }
1819 \_physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1820 \_physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1821 \NewDocumentCommand \newgeneralmatrix { t+ m o o m }
1822 {
1823     \IfBooleanTF {#1}
1824     {
1825         \IfNoValueTF {#3}
1826         { \newcommand #2 { \generalmatrix + {#5} } }
1827         {
1828             \IfNoValueTF {#4}
1829             { \newcommand #2 [#3] { \generalmatrix + {#5} } }
1830             { \newcommand #2 [#3] [#4] { \generalmatrix + {#5} } }
1831         }
1832     }
1833     {
1834         \IfNoValueTF {#3}
1835         { \newcommand #2 { \generalmatrix {#5} } }
1836         {
1837             \IfNoValueTF {#4}
1838             { \newcommand #2 [#3] { \generalmatrix {#5} } }
1839             { \newcommand #2 [#3] [#4] { \generalmatrix {#5} } }
1840         }
1841     }
1842 }
1843 \NewDocumentCommand \NewGeneralMatrix { t+ m m m }
1844 {
1845     \IfBooleanTF {#1}
1846     { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1847     { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1848 }

```

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

```

1849 \endpackage

```

# Index

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

Symbols	
$\backslash +$ .....	167
$\backslash -$ .....	167
$\backslash \backslash$ .....	57, 1111, 1118, 1125, 1149, 1461, 1511, 1577, 1672
$\backslash \{$ .....	339, 569, 588, 606, 647, 672
$\backslash \}$ .....	339, 569, 590, 606, 647, 672
A	
$\backslash A$ .....	162, 167
$\backslash \text{abs}$ .....	<u>640</u>
$\backslash \text{absolutevalue}$ .....	<u>640</u>
$\backslash \text{acomm}$ .....	<u>640</u>
$\backslash \text{aftergroup}$ .....	217
$\backslash \text{anticommutator}$ .....	<u>640</u>
$\backslash \text{AtBeginDocument}$ .....	262
B	
$\backslash \text{begin}$ .....	344, 345, 346, 347, 348, 349, 350, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 589, 592, 596, 599, 600, 601, 606, 607, 611, 614, 615, 616, 968, 969, 970, 971, 972, 973, 974, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993
$\backslash \text{bgroup}$ .....	216, 218
$\backslash \text{Big}$ .....	622
$\backslash \text{big}$ .....	622
$\backslash \text{Bigg}$ .....	622
$\backslash \text{bigg}$ .....	622
$\backslash \text{Biggl}$ .....	629
$\backslash \text{biggl}$ .....	628
$\backslash \text{Biggr}$ .....	629
$\backslash \text{biggr}$ .....	628
$\backslash \text{Bigl}$ .....	627
$\backslash \text{bigl}$ .....	626
$\backslash \text{Bigr}$ .....	627
$\backslash \text{bigr}$ .....	626
$\backslash \text{boldsymbol}$ .....	206
bool commands:	
$\backslash \text{bool\_case\_false:n}$ ....	539, 540, 547
$\backslash \text{bool\_case\_true:nTF}$ .....	624, 1381
$\backslash \text{bool\_if:N}$ ..	16, 21, 26, 85, 195, 553, 555, 559, 1043, 1098, 1105, 1115, 1123, 1135, 1142, 1146, 1153, 1157, 1421, 1546, 1634, 1718, 1757
$\backslash \text{bool\_if\_p:N}$ .....	1419, 1507
$\backslash \text{bool\_if\_p:n}$ .....	626, 627, 628, 629, 1418, 1506
$\backslash \text{bool\_lazy\_and\_p:nn}$ .....	1386, 1391
$\backslash \text{bool\_lazy\_or:nnTF}$ .....	31, 126, 137, 392, 397, 1417, 1505, 1748
$\backslash \text{bool\_new:N}$ .....	9, 10, 11, 12, 13, 51, 66, 455, 457, 731, 732, 733, 741, 755, 763, 764
$\backslash \text{bool\_set:Nn}$ .....	466, 467, 1610
$\backslash \text{bool\_set\_false:N}$ .....	95, 183, 250, 1031, 1059, 1074, 1191
$\backslash \text{bool\_set\_inverse:N}$ .....	801, 918
$\backslash \text{bool\_set\_true:N}$ .....	129, 140, 150, 190, 246, 249, 1030, 1063, 1067, 1078, 1082, 1185, 1484, 1490
$\backslash \text{c\_false\_bool}$ .....	584, 604
$\backslash \text{c\_true\_bool}$ .....	567, 577, 584, 604
$\backslash \text{Bqty}$ .....	<u>640</u>
$\backslash \text{bqty}$ .....	<u>640</u>
C	
$\backslash \text{cdots}$ ....	818, 1097, 1100, 1101, 1104, 1106, 1137, 1138, 1141, 1143, 1776
clist commands:	
$\backslash \text{clist\_clear:N}$ .....	999
$\backslash \text{clist\_concat:NNN}$ .....	1006
$\backslash \text{clist\_if\_empty:nTF}$ .....	222
$\backslash \text{clist\_if\_in:NnTF}$ .....	1334
$\backslash \text{clist\_map\_break:n}$ .....	176, 188
$\backslash \text{clist\_map\_inline:nn}$ ..	82, 172, 184, 1004
$\backslash \text{clist\_new:N}$ .....	50, 739, 740
$\backslash \text{clist\_put\_right:Nn}$ ..	880, 886, 1325
$\backslash \text{clist\_set:Nn}$ .....	996
$\backslash \text{clist\_set\_eq:NN}$ .....	1301
$\backslash \text{c\_empty\_clist}$ .....	1301
$\backslash \text{comm}$ .....	670
$\backslash \text{commamatrix}$ .....	30, <u>1495</u> , 1820
$\backslash \text{commutator}$ .....	<u>640</u>
$\backslash \text{crossproduct}$ .....	271
cs commands:	
$\backslash \text{cs:w}$ .....	32, 33, 909
$\backslash \text{cs\_end:}$ .....	32, 33, 694, 909
$\backslash \text{cs\_generate\_variant:Nn}$ .....	3, 4, 5, 6, 89, 92, 198, 410, 550, 1169, 1307, 1364, 1705, 1715
$\backslash \text{cs\_gset\_eq:NN}$ .....	40, 41, 42
$\backslash \text{cs\_if\_exist:N}$ .....	316



<code>\if_int_compare:w</code> . . . . .	485, 1107, 1234, 1237, 1260, 1272, 1529, 1574
<code>\IfBooleanT</code> . . . . .	588, 590, 1501
<code>\IfBooleanTF</code> . . . . .	264, 266, 268, 278, 446, 538, 595, 597, 610, 612, 632, 964, 1029, 1034, 1403, 1604, 1607, 1626, 1627, 1628, 1792, 1814, 1823, 1845
<code>\IfNoValueTF</code> . . . . .	413, 419, 1794, 1797, 1803, 1806, 1825, 1828, 1834, 1837
int commands:	
<code>\int_compare:nNnTF</code> . . . . .	105, 115, 149
<code>\int_compare:nTF</code> . . . . .	1057, 1072
<code>\int_compare_p:nNn</code> . . . . .	127, 128, 138, 139
<code>\int_eval:n</code> . . . . .	28, 1241, 1250, 1268, 1280, 1290, 1291, 1383, 1753, 1754
<code>\int_incr:N</code> . . . . .	484, 1216, 1550, 1558, 1738, 1740, 1744, 1746, 1759, 1760
<code>\int_max:nn</code> . . . . .	687, 689
<code>\int_new:N</code> . . . . .	. . . . .
	. . . . . 52, 53, 62, 63, 64, 65, 459, 736, 737
<code>\int_set:Nn</code> . . . . .	80, 81, 104, 114, 465, 686, 688, 1032, 1033, 1060, 1075, 1543
<code>\int_set_eq:NN</code> . . . . .	. . . . .
	. . . . . 102, 107, 112, 117, 1219, 1530
<code>\int_step_inline:nn</code> . . . . .	. . . . .
	. . . . . 1085, 1367, 1369, 1446, 1448, 1464, 1651, 1653, 1674, 1772, 1781
<code>\int_step_inline:nnn</code> . . . . .	. . . . .
	. . . . . 142, 152, 1091, 1129
<code>\int_use:N</code> . . . . .	1461, 1470, 1477, 1668, 1670, 1680, 1682, 1689, 1691, 1764, 1765, 1775, 1784
<code>\int_zero:N</code> . . . . .	1213, 1540
<code>\interval</code> . . . . .	28, 1387, 1388
<b>K</b>	
keys commands:	
<code>\keys_define:nn</code> . . . . .	. . . . . 201, 225, 290, 292, 769, 771
<code>\keys_if_exist:nnTF</code> . . . . .	174, 186
<code>\l_keys_key_str</code> . . . . .	174, 177, 186, 191, 312, 316, 319, 947, 950
<code>\l_keys_path_str</code> . . . . .	332, 954
<code>\keys_set:nn</code> . . . . .	3, 177, 191, 279, 280, 291, 318, 371, 380, 770, 949, 1012, 1035, 1036, 1404, 1405, 1498, 1500, 1502, 1608, 1609
keyval commands:	
<code>\keyval_parse:NNn</code> . . . . .	1302
<code>\keyval_parse:nnn</code> . . . . .	1164
<b>L</b>	
<code>\langle</code> . . . . .	343, 573, 600, 615
<code>\ldots</code> . . . . .	818
<code>\left</code> . . . . .	216, 373, 398, 588, 595, 600, 606, 607, 610, 614, 615, 616
<code>\let</code> . . . . .	273, 533
<code>\lggroup</code> . . . . .	595, 610
<code>\lVert</code> . . . . .	657
<b>M</b>	
<code>\mathbf</code> . . . . .	264, 266, 268
<code>\mathcal</code> . . . . .	202, 203
<code>\mathclose</code> . . . . .	216, 218
<code>\mathopen</code> . . . . .	216, 218
<code>\matrixquantity</code> . . . . .	565
<code>\mqty</code> . . . . .	640
msg commands:	
<code>\msg_error:nnn</code> . . . . .	1196
<code>\msg_error:nnnn</code> . . . . .	331, 953
<code>\msg_new:nnn</code> . . . . .	60
<code>\msg_new:nnnn</code> . . . . .	54
<b>N</b>	
<code>\nabla</code> . . . . .	272
<code>\NewCommaMatrix</code> . . . . .	1788
<code>\newcommatrix</code> . . . . .	1788
<code>\newcommand</code> . . . . .	416, 422, 426, 435, 1795, 1798, 1799, 1804, 1807, 1808, 1826, 1829, 1830, 1835, 1838, 1839
<code>\NewDiagonalMatrix</code> . . . . .	1788
<code>\newdiagonalmatrix</code> . . . . .	1788
<code>\NewDocumentCommand</code> . . . . .	276, 336, 409, 411, 442, 444, 962, 1790, 1812, 1815, 1816, 1821, 1843, 1846, 1847
<code>\NewGeneralMatrix</code> . . . . .	1788
<code>\newgeneralmatrix</code> . . . . .	1788
<code>\NewXQuantity</code> . . . . .	377
<code>\newxquantity</code> . . . . .	377
<code>\nobreak</code> . . . . .	579, 580, 581
<code>\norm</code> . . . . .	640
<b>O</b>	
<code>\OOrder</code> . . . . .	680
<code>\oorder</code> . . . . .	640
or commands:	
<code>\or:</code> . . . . .	493, 494, 495
<code>\order</code> . . . . .	640
<code>\Ordersymbol</code> . . . . .	255
<code>\ordersymbol</code> . . . . .	254
<b>P</b>	
<code>\PassOptionsToPackage</code> . . . . .	222
<code>\pb</code> . . . . .	640
peek commands:	
<code>\peek_charcode_ignore_spaces:N</code>	533
physicx commands:	
<code>\physicx_bf:</code>	206, 210, 214, 264, 266, 268
<code>\physicx_compat:</code>	14



<code>\physicx_compat:TF</code> . . . . .	252, 658
<code>\physicx_construct:nnn</code> . . . . .	1406, 1611, 1732
<code>\physicx_declare_legacy_paren:NnnnNNn</code> . . . . .	619, 644, 645, 646, 647, 649, 656, 660, 664, 667, 671, 680, 682
<code>\physicx_declare_legacy_quantity:nnNn</code> . . . . .	453, 566, 576, 583, 603
<code>\physicx_if_num:n</code> . . . . .	160
<code>\physicx_if_num:nTF</code> . . . . .	947, 1055, 1070, 1378
<code>\physicx_if_num_sign:n</code> . . . . .	165
<code>\physicx_left: . . . . .</code>	216, 400, 534, 634
<code>\physicx_left:N</code> . . . . .	218, 402, 626, 627, 628, 629
<code>\physicx_mathtools: . . . . .</code>	24
<code>\physicx_mathtools:TF</code> . . . . .	351, 975
<code>\physicx_matrix_array_parse:n . . . . .</code>	1343, 1615
<code>\physicx_matrix_array_parse_-main: . . . . .</code>	1365, 1617
<code>\physicx_matrix_diag_parse:n . . . . .</code>	881, 1162, 1408, 1412, 1413, 1619
<code>\physicx_matrix_item_parse:n . . . . .</code>	887, 1299, 1416, 1620
<code>\physicx_matrix_new_type:nn . . . . .</code>	958
<code>\physicx_matrix_new_type:nnn . . . . .</code>	958
<code>\physicx_matrix_set_r_c:nnn</code> . . . . .	725, 859, 862, 865, 868, 871, 1217, 1225, 1240, 1249, 1266, 1278, 1289, 1336, 1360, 1371, 1752, 1763, 1774, 1783
<code>\physicx_matrix_use_r_c:nn . . . . .</code>	692, 1453, 1460, 1469, 1476, 1658, 1667, 1679, 1688
<code>\physicx_new_type:nnn . . . . .</code>	200, 337, 959, 961
<code>\physicx_option_or:nn . . . . .</code>	29
<code>\physicx_option_or:nnTF . . . . .</code>	641, 651, 669, 673
<code>\c_physicx_Order_tl</code> . . . . .	203, 255, 661, 681
<code>\c_physicx_order_tl</code> . . . . .	202, 254, 665, 683
<code>\physicx_parse_range:nnN . . . . .</code>	90, 92, 1314, 1316
<code>\physicx_parse_range:nnnN . . . . .</code>	62
<code>\physicx_parse_range_check: . . . . .</code>	62, 891
<code>\physicx_parse_range_nocheck: . . . . .</code>	62, 892
<code>\physicx_qxmatrix:nnn . . . . .</code>	1037, 1041
<code>\physicx_right: . . . . .</code>	217, 400, 534, 634
<code>\physicx_right:N . . . . .</code>	219, 404, 626, 627, 628, 629
<code>\physicx_search_also:nn . . . . .</code>	170, 181, 198, 199
<code>\physicx_search_also:nnTF . . . . .</code>	325, 939
<code>\physicx_short: . . . . .</code>	19
<code>\physicx_unimath: . . . . .</code>	44
<code>\physicx_unimath:TF . . . . .</code>	40, 41, 42, 261
<code>\physicx_use_amsymb_type: . . . . .</code>	204, 260
<code>\physicx_use_uni_bf_type: . . . . .</code>	212
<code>\physicx_use_uni_bfit_type: . . . . .</code>	208, 259
<code>\physicx_xquantity:nn . . . . .</code>	377
physicx internal commands:	
<code>\__physicx_adi:nnn . . . . .</code>	766, 923, 924, 925, 926, 927, 928, 1735
<code>\l__physicx_begin_int . . . . .</code>	62, 102, 104, 105, 107, 138, 139, 143, 149, 153
<code>\l__physicx_cmd_arg_int . . . . .</code>	459, 465, 484, 485, 492
<code>\l__physicx_cmd_arg_spec_tl . . . . .</code>	458, 464, 477, 486
<code>\l__physicx_cmd_auto_body_bool . . . . .</code>	457, 467, 512, 555, 559
<code>\l__physicx_cmd_auto_body_tl . . . . .</code>	456, 463, 479, 513, 514
<code>\l__physicx_cmd_noauto_body_bool . . . . .</code>	455, 466, 500, 553
<code>\l__physicx_cmd_noauto_body_tl . . . . .</code>	454, 462, 478, 501, 502
<code>\l__physicx_col_list_seq . . . . .</code>	743, 797, 1488, 1492
<code>\__physicx_commamatrix_enhanced: . . . . .</code>	1508, 1537
<code>\__physicx_commamatrix_enhanced_-aux:nNn . . . . .</code>	1551, 1559, 1564
<code>\__physicx_commamatrix_enhanced_-aux_e:nnn . . . . .</code>	1552, 1581
<code>\__physicx_commamatrix_enhanced_-aux_ne:nnn . . . . .</code>	1560, 1593
<code>\g__physicx_compat_bool . . . . .</code>	11, 16, 227, 246
<code>\__physicx_declare_init: . . . . .</code>	1480
<code>\__physicx_declare_init:nnn . . . . .</code>	460, 472
<code>\__physicx_declare_legacy_-quantity_aux:nnn . . . . .</code>	482, 523
<code>\__physicx_declare_legacy_-quantity_aux:NNnnn . . . . .</code>	475, 526, 550
<code>\__physicx_declare_legacy_-quantity_aux:nw . . . . .</code>	473, 519, 524
<code>\__physicx_diagonalmatrix_diag_-main: . . . . .</code>	1186, 1192, 1230, 1235
<code>\__physicx_diagonalmatrix_-enhanced:nnn . . . . .</code>	1423, 1427, 1442
<code>\__physicx_diagonalmatrix_-generate_body:NNN . . . . .</code>	1435, 1444
<code>\__physicx_diagonalmatrix_-generate_enhanced_body:NNN . . . . .</code>	1434, 1439
<code>\__physicx_diagonalmatrix_no_-diag: . . . . .</code>	1193, 1222, 1231

```

\__physicx_diagonalmatrix_set_-
  diag: ..... 1187, 1211
\l__physicx_end_int ..... 63,
  112, 114, 115, 117, 139, 143, 149, 153
\__physicx_expand:w ..... 745, 776,
  778, 780, 783, 1202, 1313, 1347, 1504
\__physicx_generalmatrix: 1621, 1632
\__physicx_generalmatrix_-
  generate:nnn ..... 1636, 1640, 1645
\__physicx_if_can_num:n ..... 1376
\__physicx_if_keyval:nTF .. 158, 1411
\l__physicx_invalid_range_bool ..
  ..... 66, 85, 95, 129, 140, 150
\l__physicx_item_ignore_clist ...
  ..... 1301, 1325, 1334
\__physicx_loadpackage_options:nnn
  ..... 220, 231, 233, 235
\g__physicx_mathtools_bool .....
  ..... 9, 26, 249, 250
\__physicx_matrix_add_data:nn ...
  ..... 883, 889, 1000, 1002
\l__physicx_matrix_after_begin_-
  tl ..... 761, 899, 901, 1532
\l__physicx_matrix_after_end_tl .
  ..... 762, 902, 904, 1534
\__physicx_matrix_appto_body_-
  e:nnn ..... 1046, 1424, 1637, 1693
\__physicx_matrix_appto_body_-
  ne:nnn ..... 1050, 1428, 1641, 1693
\l__physicx_matrix_args_tl .....
  ..... 760, 897, 898, 1532
\l__physicx_matrix_array_tl ....
  ..... 734, 773, 1503,
  1504, 1510, 1512, 1515, 1542, 1615
\__physicx_matrix_autocalc:nn ...
  ..... 754, 789, 791, 1298, 1349, 1355
\__physicx_matrix_begin:w .....
  ..... 752, 894, 1532
\l__physicx_matrix_beginning_tl .
  ..... 767, 929, 931, 1734
\l__physicx_matrix_body_tl .....
  ..... 735, 1053, 1088,
  1093, 1094, 1100, 1101, 1106, 1111,
  1118, 1120, 1122, 1124, 1125, 1127,
  1131, 1132, 1137, 1138, 1143, 1149,
  1150, 1152, 1154, 1155, 1450, 1457,
  1466, 1473, 1514, 1526, 1533, 1539,
  1571, 1662, 1671, 1683, 1695, 1708
\__physicx_matrix_calc:nn .....
  ..... 684, 789, 1177
\__physicx_matrix_col_iterate:n .
  751, 875, 1491, 1589, 1599, 1701, 1712
\l__physicx_matrix_cols_int ....
  ..... 688, 689,
  737, 786, 1033, 1072, 1075, 1091,
  1121, 1129, 1151, 1158, 1159, 1213,
  1216, 1220, 1268, 1280, 1291, 1316,
  1369, 1529, 1530, 1721, 1728, 1738,
  1740, 1754, 1760, 1765, 1775, 1781
\l__physicx_matrix_diag_bool ...
  ..... 741, 1185, 1191
\__physicx_matrix_diag_calc:nn ..
  ..... 1176, 1181, 1226,
  1243, 1252, 1261, 1273, 1284, 1297
\l__physicx_matrix_diag_clist ...
  ..... 739, 878, 880, 1408, 1619
\__physicx_matrix_diag_parse_-
  aux:n ..... 1165, 1170
\__physicx_matrix_diag_parse_-
  aux:nn ..... 1166, 1198
\__physicx_matrix_diag_parse_-
  aux_anti:n ..... 1206, 1258
\__physicx_matrix_diag_parse_-
  aux_regu:n ..... 1209, 1232
\l__physicx_matrix_dotcol_bool ..
  ..... 733, 1074, 1078,
  1082, 1098, 1123, 1135, 1153, 1157
\l__physicx_matrix_dotrow_bool ..
  ..... 732, 1059, 1063, 1067, 1115
\__physicx_matrix_element_-
  aux:nnn ..... 805, 811,
  814, 820, 823, 829, 832, 838, 841, 846
\__physicx_matrix_end: ..... 895, 1534
\__physicx_matrix_end:w ..... 753
\l__physicx_matrix_ending_tl ...
  ..... 768, 932, 934, 1768
\l__physicx_matrix_enhanced_bool
  ..... 764, 915, 918, 1419, 1507
\__physicx_matrix_enhanced_init:
  ..... 1480, 1545, 1650
\l__physicx_matrix_expand_-
  element_bool ..... 755,
  854, 1043, 1421, 1484, 1490, 1546, 1634
\__physicx_matrix_generate_-
  body:NNNN ..... 1441, 1644, 1647
\l__physicx_matrix_infinite_bool
  ..... 731, 798, 801, 1030,
  1031, 1105, 1142, 1146, 1610, 1757
\l__physicx_matrix_item_clist ...
  ..... 740, 884, 886, 1416, 1620
\__physicx_matrix_item_parse_-
  aux:n ..... 1303, 1308
\__physicx_matrix_item_parse_-
  aux:nn ..... 1304, 1309
\__physicx_matrix_last_aux_c: ...
  ..... 1739, 1761, 1770
\__physicx_matrix_last_aux_r: ...
  ..... 1745, 1762, 1779

```

\l__physicx_matrix_last_col_tl ..	\__physicx_parse_range_range_-
..... 750, 877, 1736, 1750	nocheck: ..... 75, 147
\l__physicx_matrix_last_row_tl ..	\__physicx_parse_range_single:n .
..... 749, 876, 1742, 1749	..... 69, 74, 122, 134
\l__physicx_matrix_main_tl .....	\__physicx_parse_range_single_-
..... 738, 793, 1372, 1613	check:n ..... 69, 124, 134
\__physicx_matrix_row_iterate:n .	\__physicx_parse_range_single_-
748, 874, 1485, 1588, 1598, 1700, 1711	nocheck:n ..... 74, 132
\l__physicx_matrix_rows_int .....	\g__physicx_physics_bool ..... 10
..... 686, 687, 736, 785, 1032,	\l__physicx_quantity_args_tl ...
1057, 1060, 1085, 1107, 1219, 1314,	..... 282, 304, 305, 384
1367, 1543, 1574, 1722, 1727, 1744,	\l__physicx_quantity_code_tl ...
1746, 1753, 1759, 1764, 1772, 1784	..... 283, 306, 381, 385
\__physicx_matrix_save_or_print:	\l__physicx_quantity_left_size_-
..... 1038, 1436, 1517, 1520, 1622	tl ..... 284, 298, 393, 398, 402
\l__physicx_matrix_save_tl .....	\l__physicx_quantity_left_tl ...
..... 907, 909, 1522, 1525	..... 285, 296, 383
\l__physicx_matrix_sep_dim .....	\l__physicx_quantity_post_tl ...
..... 765, 905, 1111, 1118,	..... 286, 295, 407
1125, 1149, 1461, 1511, 1577, 1672	\l__physicx_quantity_pre_tl ....
\__physicx_matrix_set_data:nn ...	..... 287, 294, 391
..... 882, 888, 997	\l__physicx_quantity_right_size_-
\__physicx_matrix_set_r_c_-	tl ..... 288, 299, 394, 399, 404
ckall:nnn ..... 723, 872	\l__physicx_quantity_right_tl ...
\__physicx_matrix_set_r_c_-	..... 289, 297, 386
ckep:nnn ..... 708, 863	\__physicx_qxmatrix_appto_-
\__physicx_matrix_set_r_c_-	body:nnn ..... 1045, 1049,
ckig:nnn ..... 703, 866	1089, 1095, 1102, 1128, 1133, 1139
\__physicx_matrix_set_r_c_-	\g__physicx_reqty_bool .....
ckigep:nnn ..... 714, 724, 869	..... 13, 237, 565, 640
\__physicx_matrix_set_r_c_-	\l__physicx_row_list_seq .....
nock:nnn ..... 701, 726, 860	..... 742, 795, 1482, 1486
\l__physicx_matrix_tmparr_c_sep .	\g__physicx_short_bool ... 12, 21, 229
..... 1354, 1357, 1358	\l__physicx_tmp_col_seq .. 1566, 1569
\l__physicx_matrix_tmparr_r_sep .	\l__physicx_tmp_coled_seq .....
..... 1348, 1350, 1352	..... 1568, 1573, 1583, 1595
\__physicx_matrix_transpose:N ...	\l__physicx_tmp_colnum_seq .....
..... 1433, 1643, 1716	..... 1317, 1323, 1332
\l__physicx_matrix_transpose_-	\l__physicx_tmp_rownum_seq .....
bool ..... 763, 910, 1718	..... 1315, 1321, 1330
\l__physicx_max_int .....	\l__physicx_tmp_seq .....
..... 64, 81, 112, 115, 117, 127, 138	..... 1541, 1544, 1548, 1556
\l__physicx_min_int .....	\l__physicx_tmp_tl .....
..... 65, 80, 102, 105, 107, 128	..... 487, 501, 513, 1655, 1661,
\__physicx_nauto_case:nnnn . 528, 551	1664, 1670, 1676, 1682, 1685, 1691
\__physicx_new_matrix_cmd:NNN . 1788	\l__physicx_tmpa_bool 51, 183, 190, 195
\__physicx_new_xquantity_aux:w ..	\l__physicx_tmpa_clist .....
..... 415, 421, 425, 429, 432	..... 50
\__physicx_parse_range_aux:n 84, 93	\l__physicx_tmpa_int .....
\__physicx_parse_range_range: ...	..... 52, 1540, 1550, 1552, 1558, 1560
..... 70, 75, 120, 157	\l__physicx_tmpa_seq .....
\__physicx_parse_range_range_-	..... 86, 96, 130, 133, 144, 154
check: ..... 70, 135, 157	\l__physicx_tmpa_tl .....
	100, 101, 104, 110, 111, 114, 312, 313

<code>\l__physicx_tmparr_tl</code> .....	1345, 1346, 1347, 1348
<code>\l__physicx_tmpb_int</code> .....	53
<code>\l__physicx_tmpb_seq</code> ...	99, 100, 110
<code>\l__physicx_tmpdiag_seq</code>	1203, 1214,
	1224, 1227, 1228, 1238, 1244, 1245,
	1247, 1253, 1254, 1262, 1263, 1264,
	1274, 1275, 1276, 1285, 1286, 1287
<code>\l__physicx_tmpdiag_tl</code> .....	1200, 1201, 1202, 1203
<code>\l__physicx_tmpitem_tl</code> .....	1311, 1312, 1313, 1319, 1337
<code>\__physicx_xquantity_aux:nnnn</code> ...	382, 389, 410
<code>\physicxempty</code> .	697, 711, 719, 756, 757, 856
<code>\physicxexcept</code> .	759, 827, 836, 845, 851, 853
<code>\PHYSICXIGNORE</code> .....	7, 8, 705, 716, 1319
<code>\physicxset</code> .....	276
<code>\physicxtmp</code> ...	453, 1035, 1404, 1502, 1608
<code>\poissonbracket</code> .....	640
<code>\pqty</code> .....	640
prg commands:	
<code>\prg_generate_conditional_</code>	
<code>variant:Nnn</code> .....	199
<code>\prg_new_conditional:Npnn</code> .....	
... 14, 19, 24, 29, 160, 165, 181, 1376	
<code>\prg_replicate:nn</code> .....	1121, 1151
<code>\prg_return_false:</code> .....	17,
22, 27, 35, 47, 163, 168, 196, 1397	
<code>\prg_return_true:</code> .....	17,
22, 27, 34, 47, 163, 168, 196, 1379, 1396	
<code>\prg_set_conditional:Npnn</code> .....	44
<code>\ProcessKeysPackageOptions</code> .....	243
<b>Q</b>	
<code>\qty</code> .....	640
<code>\quantity</code> .....	565, 643
quark commands:	
<code>\quark_if_recursion_tail_stop:n</code> .	
.....	521, 522
<code>\q_recursion_stop</code> .....	474
<code>\q_recursion_tail</code> .....	474
<code>\qxmatrix</code> .....	1017, 1629
<code>\qxqty</code> .....	451
<b>R</b>	
<code>\rangle</code> .....	343, 573, 600, 615
regex commands:	
<code>\regex_match:nnTF</code> ..	162, 167, 836, 845
<code>\RequirePackage</code> .....	223
<code>\rgroup</code> .....	597, 612
<code>\right</code> .....	217, 374, 399, 590,
597, 600, 606, 607, 612, 614, 615, 616	
<code>\rVert</code> .....	657
<b>S</b>	
<code>\sb</code> .	747, 1088, 1093, 1100, 1127, 1131, 1137
scan commands:	
<code>\scan_stop:</code> .....	1108, 1575
seq commands:	
<code>\c_empty_seq</code> .....	79, 1568
<code>\seq_clear:N</code> .....	96
<code>\seq_concat:NNN</code> .....	86
<code>\seq_count:N</code> ....	1227, 1228, 1244,
1245, 1253, 1254, 1262, 1263, 1274,	
1275, 1285, 1286, 1350, 1357, 1544	
<code>\seq_if_empty:NTF</code> .....	1482, 1488
<code>\seq_item:Nn</code> .....	1486, 1492
<code>\seq_map_indexed_inline:Nn</code> .....	
.....	1214, 1224, 1238, 1247,
1264, 1276, 1287, 1352, 1358, 1569	
<code>\seq_map_inline:Nn</code> .....	
.....	1321, 1323, 1330, 1332
<code>\seq_map_tokens:Nn</code> .....	1548, 1556
<code>\seq_new:N</code> .....	742, 743
<code>\seq_pop_left:NN</code> .....	100, 110
<code>\seq_put_right:Nn</code> .....	
.....	130, 133, 144, 154, 1583, 1595
<code>\seq_set_eq:NN</code> .....	79, 1568
<code>\seq_set_split:Nnn</code> .....	5, 99,
795, 797, 1203, 1348, 1354, 1541, 1566	
<code>\seq_use:Nn</code> .....	1573
<code>\setmathfont</code> .....	269
<code>\setmatrixdata</code> .....	995
<code>\setmatrixtype</code> ....	936, 958, 968, 969,
970, 971, 972, 973, 974, 977, 978,	
979, 980, 981, 982, 983, 984, 985,	
986, 987, 988, 989, 990, 991, 992, 993	
<code>\setquantitytype</code> ....	309, 336, 338,
339, 340, 341, 342, 343, 344, 345,	
346, 347, 348, 349, 350, 353, 354,	
355, 356, 357, 358, 359, 360, 361,	
362, 363, 364, 365, 366, 367, 368, 369	
<code>\smallmatrixquantity</code> .....	565
<code>\smqty</code> .....	640
str commands:	
<code>\c_backslash_str</code> .....	313
<code>\str_case_e:nnTF</code> .....	1172
<code>\str_const:Nn</code> .....	728, 729, 730
<code>\str_if_eq:nnTF</code> .....	506, 508
<code>\symbf</code> .....	214, 272
<code>\symbfitt</code> .....	210
<b>T</b>	
T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:	
<code>\@declareparencmd</code> .....	619
<code>\@declarequantitycmd</code> .....	453
<code>\@ifpackageloaded</code> ....	245, 248, 258
<code>\@ifstar</code> .....	437

