

The `physicx` package

Wenjian Chern (Longaster*)

April 27, 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_end: \exp_not:N \PHYSICXIGNORE }
9 \bool_new:N \g__physicx_mathtools_bool
10 \bool_new:N \g__physicx_physics_bool
11 \bool_new:N \g__physicx_compat_bool
12 \bool_new:N \g__physicx_short_bool
13 \bool_new:N \g__physicx_reqty_bool
14 \prg_new_conditional:Npnn \physicx_compat: { T, F, TF }
15 {
16   \bool_if:NTF \g__physicx_compat_bool
17   { \prg_return_true: } { \prg_return_false: }
18 }
19 \prg_new_conditional:Npnn \physicx_short: { T, F, TF }
20 {
21   \bool_if:NTF \g__physicx_short_bool
22   { \prg_return_true: } { \prg_return_false: }
23 }
24 \prg_new_conditional:Npnn \physicx_mathtools: { T, F, TF }
25 {
26   \bool_if:NTF \g__physicx_mathtools_bool
27   { \prg_return_true: } { \prg_return_false: }
28 }
29 \prg_new_conditional:Npnn \physicx_option_or:nn #1#2 { T, F, TF }
30 {
31   \bool_lazy_or:nnTF
32   { \cs:w g__physicx_ #1 _bool \cs_end: }
33   { \cs:w g__physicx_ #2 _bool \cs_end: }
```

*Email: longaster@163.com

```

34     { \prg_return_true: }
35     { \prg_return_false: }
36 }
37 \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\divisionsymbol\div
274     \let\div\divergence
275   }
276 }

```

`\physicxset` `physicx` setup command.

```

277 \NewDocumentCommand \physicxset { s m }
278 {
279   \IfBooleanTF {#1}
280   { \keys_set:nn { physicx/#2 } }
281   { \keys_set:nn { physicx } {#2} }
282 }

```

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

1.2 Quantity things

1.2.1 New quantity interfaces

```

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

```

```

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

```

```

341 \setquantitytype { p } { left={ ( } , right={ ) } , }
342 \setquantitytype { v } { left=\vert , right=\vert , }
343 \setquantitytype { V } { left=\Vert , right=\Vert , }
344 \setquantitytype { a } { left=\langle , right=\rangle , }
345 \setquantitytype { m } { left=\begin{matrix} , right=\end{matrix} , noauto }
346 \setquantitytype { bm } { left=\begin{bmatrix} , right=\end{bmatrix} , noauto }
347 \setquantitytype { Bm } { left=\begin{Bmatrix} , right=\end{Bmatrix} , noauto }
348 \setquantitytype { pm } { left=\begin{pmatrix} , right=\end{pmatrix} , noauto }
349 \setquantitytype { vm } { left=\begin{vmatrix} , right=\end{vmatrix} , noauto }
350 \setquantitytype { Vm } { left=\begin{Vmatrix} , right=\end{Vmatrix} , noauto }
351 \setquantitytype { sm } { left=\begin{smallmatrix} , right=\end{smallmatrix} , noauto }
352 \physicx_mathtools:T
353 {
354   \setquantitytype { m* } { left=\begin{matrix}* , right=\end{matrix}* , noauto }
355   \setquantitytype { bm* } { left=\begin{bmatrix}* , right=\end{bmatrix}* , noauto }
356   \setquantitytype { Bm* } { left=\begin{Bmatrix}* , right=\end{Bmatrix}* , noauto }
357   \setquantitytype { pm* } { left=\begin{pmatrix}* , right=\end{pmatrix}* , noauto }
358   \setquantitytype { vm* } { left=\begin{vmatrix}* , right=\end{vmatrix}* , noauto }
359   \setquantitytype { Vm* } { left=\begin{Vmatrix}* , right=\end{Vmatrix}* , noauto }
360   \setquantitytype { sm* } { left=\begin{smallmatrix}* , right=\end{smallmatrix}* , noauto }
361   \setquantitytype { sbm } { left=\begin{bsmallmatrix} , right=\end{bsmallmatrix} , noauto }
362   \setquantitytype { sBm } { left=\begin{Bsmallmatrix} , right=\end{Bsmallmatrix} , noauto }
363   \setquantitytype { spm } { left=\begin{psmallmatrix} , right=\end{psmallmatrix} , noauto }
364   \setquantitytype { svm } { left=\begin{vsmallmatrix} , right=\end{vsmallmatrix} , noauto }
365   \setquantitytype { sVm } { left=\begin{Vsmallmatrix} , right=\end{Vsmallmatrix} , noauto }
366   \setquantitytype { sbm* } { left=\begin{bsmallmatrix}* , right=\end{bsmallmatrix}* , noauto }
367   \setquantitytype { sBm* } { left=\begin{Bsmallmatrix}* , right=\end{Bsmallmatrix}* , noauto }
368   \setquantitytype { spm* } { left=\begin{psmallmatrix}* , right=\end{psmallmatrix}* , noauto }
369   \setquantitytype { svm* } { left=\begin{vsmallmatrix}* , right=\end{vsmallmatrix}* , noauto }
370   \setquantitytype { sVm* } { left=\begin{Vsmallmatrix}* , right=\end{Vsmallmatrix}* , noauto }
371 }
372 \keys_set:nn { physicx/quantity }
373 {
374   left-size = \left ,
375   right-size = \right ,
376   type = p ,
377 }
\physicx_xquantity:nn
  \newxquantity
  \NewXQuantity
378 \cs_new:Npn \physicx_xquantity:nn #1#2
379 {
380   \group_begin:
381   \keys_set:nn { physicx/quantity } {#1}
382   \tl_if_empty:nF {#2} { \tl_set:Nn \l__physicx_quantity_code_tl {#2} }
383   \__physicx_xquantity_aux:oooo
384     { \l__physicx_quantity_left_tl }
385     { \l__physicx_quantity_args_tl }
386     { \l__physicx_quantity_code_tl }
387     { \l__physicx_quantity_right_tl }
388   \group_end:
389 }
390 \cs_new:Npn \__physicx_xquantity_aux:nnnn #1#2#3#4
391 {
392   \l__physicx_quantity_pre_tl

```



```

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

```

```

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

```

```

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

```

```

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

```

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

```

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

```

```

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

```

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

```

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

```

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

```

\qty Redefine some macros in physics package.
\mqty
641 \if_bool:N \g__physicsx_reqty_bool
\smqty 642 \physicsx_option_or:nnT { compat } { short }
\pqty
\bqty
\vqty
\Bqty
\absolutevalue
\eval
\abs
\norm
\order
\oorder

```

```

643 {
644   \cs_set:Npn \qty { \quantity }
645   \physics_declare_legacy_paren:NnnnNNn \pqty { m } {#6} { } { } { } { }
646   \physics_declare_legacy_paren:NnnnNNn \bqty { m } {#6} { } { } [ ] { }
647   \physics_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } { } \vert \vert { }
648   \physics_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } { } \{ \} { }
649 }
650 \physics_declare_legacy_paren:NnnnNNn \absolutevalue
651 { m } {#6} { } { } \vert \vert { }
652 \physics_option_or:nnT { compat } { short }
653 {
654   \cs_set:Npn \eval { \evaluated }
655   \cs_set:Npn \abs { \absolutevalue }
656 }
657 \physics_declare_legacy_paren:NnnnNNn \norm
658 { m } {#6} { } { } \lVert \rVert { }
659 \physics_compat:TF
660 {
661   \physics_declare_legacy_paren:NnnnNNn \order
662   { m } {#6} { } { } \c_physicx_Order_tl { } { }
663 }
664 {
665   \physics_declare_legacy_paren:NnnnNNn \order
666   { m } {#6} { } { } \c_physicx_order_tl { } { }
667 }
668 \physics_declare_legacy_paren:NnnnNNn \commutator
669 { m m } { #6 , #7 } { } { } [ ] { }
670 \physics_option_or:nnT { compat } { short }
671 { \cs_set:Npn \comm { \commutator } }
672 \physics_declare_legacy_paren:NnnnNNn \poissonbracket
673 { m m } { #6 , #7 } { } { } \{ \} { }
674 \physics_option_or:nnT { compat } { short }
675 {
676   \cs_set:Npn \pb { \poissonbracket }
677   \cs_set:Npn \anticommutator { \poissonbracket }
678   \cs_set:Npn \acomm { \poissonbracket }
679 }
680 \fi:
681 \physics_declare_legacy_paren:NnnnNNn \OOrder
682 { m } {#6} { } { } \c_physicx_Order_tl { } { }
683 \physics_declare_legacy_paren:NnnnNNn \oorder
684 { 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

```

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

```

```

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

```

```

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

```

1.3.2 Matrix keys

```

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

```



```

797 col-list .code:n =
798   \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
799   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
800   infinite .default:n = true ,
801   !infinite .code:n =
802     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
803   element-code .cs_set:Np = \physicx@matricielement #1#2#3 ,
804   element-code* .choice: ,
805   element-code* / except-empty .code:n =
806     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
807     \physicx@matricielement
808     \cs_set:Npn \physicx@matricielement ##1##2##3
809     {
810       \tl_if_empty:nTF {##1}
811       {##1}
812       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
813     } ,
814   element-code* / except-blank .code:n =
815     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
816     \physicx@matricielement
817     \cs_set:Npn \physicx@matricielement ##1##2##3
818     {
819       \tl_if_blank:nTF {##1}
820       {##1}
821       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
822     } ,
823   element-code* / except-dots .code:n =
824     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
825     \physicx@matricielement
826     \cs_set:Npn \physicx@matricielement ##1##2##3
827     {
828       \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
829       {##1}
830       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
831     } ,
832   element-code* / except-tl .code:n =
833     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
834     \physicx@matricielement
835     \cs_set:Npn \physicx@matricielement ##1##2##3
836     {
837       \tl_if_in:onTF { \physicxexcept } {##1}
838       {##1}
839       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
840     } ,
841   element-code* / except-regex .code:n =
842     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
843     \physicx@matricielement
844     \cs_set:Npn \physicx@matricielement ##1##2##3
845     {
846       \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
847       {##1}
848       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
849     } ,
850   element-code* / only-regex .code:n =

```

```

851 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
852 \physicx@matricelement
853 \cs_set:Npn \physicx@matricelement ##1##2##3
854 {
855   \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
856   { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
857   {##1}
858 } ,
859 element-code* / unknown .code:n =
860   \cs_set:Npx \physicx@matricelement { \exp_not:c {#1} },
861 element-exception .tl_set:N = \physicxexcept ,
862 element-exception+ .code:n =
863   \tl_put_right:Nn \physicxexcept {#1} ,
864 expand-element .bool_set:N = \l__physicx_matrix_expand_element_bool ,
865 expand-element .default:n = true ,
866 empty .tl_set:N = \physicxempty ,
867 check .choice: ,
868 check / none .code:n =
869   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
870   \__physicx_matrix_set_r_c_nock:nnn ,
871 check / empty .code:n =
872   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
873   \__physicx_matrix_set_r_c_cke:nnn ,
874 check / ignore .code:n =
875   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
876   \__physicx_matrix_set_r_c_ckig:nnn ,
877 check / igep .code:n =
878   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
879   \__physicx_matrix_set_r_c_ckigep:nnn ,
880 check / all .code:n =
881   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
882   \__physicx_matrix_set_r_c_ckall:nnn ,
883 check .default:n = all ,
884 row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
885 col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
886 last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
887 last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
888 diag .clist_set:N = \l__physicx_matrix_diag_clist ,
889 diag+ .code:n =
890   \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
891 diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
892 diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
893 diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
894 item .clist_set:N = \l__physicx_matrix_item_clist ,
895 item+ .code:n =
896   \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
897 item-now .code:n = \physicx_matrix_item_parse:n {#1} ,
898 item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
899 item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
900 check-range .choice: ,
901 check-range / true .code:n = \physicx_parse_range_check: ,
902 check-range / false .code:n = \physicx_parse_range_noccheck: ,
903 check-range .default:n = true ,
904 begin .tl_set:N = \__physicx_matrix_begin:w ,

```

```

905 end .tl_set:N = \__physicx_matrix_end: ,
906 args .code:n =
907   \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,
908 args* .tl_set:N = \l__physicx_matrix_args_tl ,
909 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
910 after-begin+ .code:n =
911   { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
912 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
913 after-end+ .code:n =
914   { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
915 sepdim .dim_set:N = \l__physicx_matrix_sepdim ,
916 type .multichoice: ,
917 saveto .tl_set:N = \l__physicx_matrix_save_tl ,
918 saveto* .code:n =
919   \tl_set:Nn \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
920 transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
921 transpose .default:n = true ,
922 ' .meta:n = { transpose = true } ,
923 T .meta:n = { transpose = true } ,
924 MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
925 enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
926 enhanced .default:n = true ,
927 !enhanced .code:n =
928   \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
929 cr .tl_set:N = \physicx@cr ,
930 align .tl_set:N = \physicx@align ,
931 sep .tl_set:N = \physicx@sep ,
932 adi-order .choice: ,
933 adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
934 adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
935 adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
936 adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
937 adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
938 adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
939 beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
940 beginning+ .code:n =
941   \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
942 ending .tl_set:N = \l__physicx_matrix_ending_tl ,
943 ending+ .code:n =
944   \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
945
946 settype .code:n = \setmatritxttype #1 ,
947
948 unknown .code:n =
949   \physicx_search_also:nnF
950   {
951     physicx/matrix/type ,
952     physicx/matrix/expand ,
953     physicx/matrix/element-code* ,
954   }
955   {#1}
956   {
957     \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
958     {

```

```

959         \keys_set:nx { physicx/matrix }
960         { MaxMatrixCols = \l_keys_key_str }
961     }
962     {
963         \msg_error:nxxx { physicx } { unknown-key }
964         \l_keys_path_str { physicx/matrix }
965     }
966 } ,
967 }

```

\physicx_matrix_new_type:nnn

```

968 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
969 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
970 \cs_new:Npn \physicx_matrix_new_type:nn
971 { \physicx_new_type:nnn { matrix } }
972 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
973 {
974     \IfBooleanTF {#1}
975     { \physicx_matrix_new_type:nn {#2} }
976     { \physicx_matrix_new_type:nnn {#2} }
977 }

```

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

A few types.

```

978 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
979 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
980 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
981 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
982 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
983 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
984 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
985 \physicx_mathtools:T
986 {
987     \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
988     \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
989     \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
990     \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
991     \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
992     \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
993     \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
994     \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
995     \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
996     \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
997     \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
998     \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
999     \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
1000    \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
1001    \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
1002    \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
1003    \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
1004 }

```

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

```

1005 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
1006 { \clist_set:cn { physicx@ #1 data@ #2 } }
1007 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
1008 {
1009   \clist_clear:c { l__physicx_matrix_ #1 _clist }
1010   \__physicx_matrix_add_data:nn {#1} {#2}
1011 }
1012 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
1013 {
1014   \clist_map_inline:nn {#2}
1015   {
1016     \clist_concat:ccc
1017     { l__physicx_matrix_ #1 _clist }
1018     { l__physicx_matrix_ #1 _clist }
1019     { physicx@ #1 data@ #2 }
1020   }
1021 }

```

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

Initial settings.

```

1022 \keys_set:nn { physicx/matrix }
1023 {
1024   type = m ,
1025   saveto = ? ,
1026 }

```

\qxmatri

```

1027 %% basicly, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
1028 xmatmmn-in-the-physics-package, but changed some
1029 % #1 = boolean, saveto matrix
1030 % #2 = star, infinite
1031 % #3 = options
1032 % #4 = letter for the entries
1033 % #5 = number of rows
1034 % #6 = number of explicit rows, default = 3
1035 % #7 = number of columns
1036 % #8 = number of explicit columns, default = 3
1037 \DeclareDocumentCommand \qxmatri { t= s 0{type=p} m m 0{3} m 0{3} }
1038 {
1039   \group_begin:
1040   \IfBooleanTF { #2 }
1041   { \bool_set_true:N \l__physicx_matrix_infinite_bool }
1042   { \bool_set_false:N \l__physicx_matrix_infinite_bool }
1043   \int_set:Nn \l__physicx_matrix_rows_int {#6}
1044   \int_set:Nn \l__physicx_matrix_cols_int {#8}
1045   \IfBooleanTF {#1}
1046   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1047   { \keys_set:nn { physicx/matrix } {#3} }
1048   \physicx_qxmatri:nnn {#4} {#5} {#7}
1049   \__physicx_matrix_save_or_print:
1050   \group_end:
1051 }
1052 \cs_new_protected:Nn \physicx_qxmatri:nnn
1053 {

```

```

1053 \bool_if:NTF \l__physicx_matrix_expand_element_bool
1054 {
1055     \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1056     \__physicx_matrix_appto_body_e:nnn
1057 }
1058 {
1059     \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1060     \__physicx_matrix_appto_body_ne:nnn
1061 }
1062 % clear the variable containing the body of the matrix
1063 \tl_clear:N \l__physicx_matrix_body_tl
1064 % set the tentative number of explicit rows
1065 \physicx_if_num:nTF { #2 }
1066 {
1067     % number of rows is an integer
1068     \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
1069     {
1070         % if #2 <= rows, we don't want a row of dots
1071         \bool_set_false:N \l__physicx_matrix_dotrow_bool
1072         \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1073     }
1074     % we want a row of dots
1075     \bool_set_true:N \l__physicx_matrix_dotrow_bool
1076 }
1077 {
1078     % number of rows is symbolic, we want a row of dots
1079     \bool_set_true:N \l__physicx_matrix_dotrow_bool
1080 }
1081 % set the tentative number of explicit columns
1082 \physicx_if_num:nTF { #3 }
1083 {
1084     % number of cols is an integer
1085     \int_compare:nTF { #3 <= \l__physicx_matrix_cols_int }
1086     {
1087         % if #3 <= cols, we don't want a column of dots
1088         \bool_set_false:N \l__physicx_matrix_dotcol_bool
1089         \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1090     }
1091     % we want a column of dots
1092     \bool_set_true:N \l__physicx_matrix_dotcol_bool
1093 }
1094 {
1095     % number of columns is symbolic, we want a column of dots
1096     \bool_set_true:N \l__physicx_matrix_dotcol_bool
1097 }
1098 % loop through the rows
1099 \int_step_inline:nn { \l__physicx_matrix_rows_int }
1100 {
1101     % add the first entry in the row
1102     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1103     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
1104     % add the further entries in the explicit columns
1105     \int_step_inline:nn { 2 } { \l__physicx_matrix_cols_int }
1106     {
1107         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
1108         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1109         \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
1110     }

```

```

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

```

```

1161 \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1162 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1163 \bool_if:NT \l__physicx_matrix_dotcol_bool
1164 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1165 \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1166 % update cols
1167 \bool_if:NTF \l__physicx_matrix_dotcol_bool
1168 { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1169 { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1170 }
1171 }

```

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

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
1172 \cs_new:Npn \physicx_matrix_diag_parse:n #1
1173 {
1174   \keyval_parse:nnn
1175   \__physicx_matrix_diag_parse_aux:n
1176   \__physicx_matrix_diag_parse_aux:nn
1177   {#1}
1178 }
1179 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
1180 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
1181 {
1182   \str_case:e:nnF {#1}
1183   {
1184     { auto-update }
1185     {
1186       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
1187       \__physicx_matrix_calc:nn
1188     }
1189     { noauto-update }
1190     {
1191       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
1192     }
1193     { true }
1194     {
1195       \bool_set_true:N \l__physicx_matrix_diag_bool
1196       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1197       \__physicx_diagonalmatrix_set_diag:
1198     }
1199     { false }
1200     {
1201       \bool_set_false:N \l__physicx_matrix_diag_bool
1202       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1203       \__physicx_diagonalmatrix_no_diag:
1204     }
1205   }
1206   { \msg_error:nnn { physicx } { diag-key } {#1} }
1207 }
1208 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
1209 {
1210   \tl_set:Nn \l__physicx_tmpdiag_tl {#2}

```



```

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

```

```

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

```

```

1315     {#1}
1316   }
1317   \cs_generate_variant:Nn \physics_matrix_item_parse:n { o }
1318   \cs_new:Npn \__physics_matrix_item_parse_aux:n #1 { }
1319   \cs_new:Npn \__physics_matrix_item_parse_aux:nn #1#2
1320   {
1321     \tl_set:Nn \l__physics_tmpitem_tl {#2}
1322     \tl_set:Nx \l__physics_tmpitem_tl
1323     { \__physics_expand:w \l__physics_tmpitem_tl }
1324     \physics_parse_range:nxN \l__physics_matrix_rows_int
1325     { \use_i:nn #1 } \l__physics_tmp_rownum_seq
1326     \physics_parse_range:nxN \l__physics_matrix_cols_int
1327     { \use_ii:nn #1 } \l__physics_tmp_colnum_seq
1328     \exp_args:No \tl_if_eq:nnTF
1329     { \l__physics_tmpitem_tl } { \PHYSICXIGNORE }
1330     {
1331       \seq_map_inline:Nn \l__physics_tmp_rownum_seq
1332       {
1333         \seq_map_inline:Nn \l__physics_tmp_colnum_seq
1334         {
1335           \clist_put_right:Nn \l__physics_item_ignore_clist { [##1][####1] }
1336         }
1337       }
1338     }
1339     {
1340       \seq_map_inline:Nn \l__physics_tmp_rownum_seq
1341       {
1342         \seq_map_inline:Nn \l__physics_tmp_colnum_seq
1343         {
1344           \clist_if_in:NnF \l__physics_item_ignore_clist { [##1][####1] }
1345           {
1346             \exp_args:Nnno \physics_matrix_set_r_c:nnn
1347             {##1} {####1} { \l__physics_tmpitem_tl }
1348           }
1349         }
1350       }
1351     }
1352   }

```

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

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

```

\physics_matrix_array_parse:o
1353   \cs_new:Npn \physics_matrix_array_parse:n #1
1354   {
1355     \tl_set:Nn \l__physics_tmparr_tl {#1}
1356     \tl_set:Nx \l__physics_tmparr_tl
1357     { \__physics_expand:w \l__physics_tmparr_tl }
1358     \seq_set_split:NVV \l__physics_matrix_tmparr_r_sep \physicsx@cr \l__physics_tmparr_tl
1359     \__physics_matrix_autocalc:nn
1360     { \seq_count:N \l__physics_matrix_tmparr_r_sep }
1361     { 0 }
1362     \seq_map_indexed_inline:Nn \l__physics_matrix_tmparr_r_sep
1363     {
1364       \seq_set_split:Non \l__physics_matrix_tmparr_c_sep { \physicsx@align } {##2}

```

```

1365     \__physicx_matrix_autocalc:nn
1366     { 0 }
1367     { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1368     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1369     {
1370         \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1371     }
1372 }
1373 }
1374 \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.

```

1375 \cs_new:Npn \physicx_matrix_array_parse_main:
1376 {
1377     \int_step_inline:nn \l__physicx_matrix_rows_int
1378     {
1379         \int_step_inline:nn \l__physicx_matrix_cols_int
1380         {
1381             \exp_args:Nno \physicx_matrix_set_r_c:nnn
1382             {##1} {####1} \l__physicx_matrix_main_tl
1383         }
1384     }
1385 }

```

(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).

```

1386 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1387 {
1388     \physicx_if_num:nTF {#1}
1389     { \prg_return_true: }
1390     {
1391         \bool_case_true:nTF
1392         {
1393             { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1394             { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1395             {
1396                 \bool_lazy_and_p:nn
1397                 { \cs_if_exist_p:N \inteval }
1398                 { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1399             } { }
1400             {
1401                 \bool_lazy_and_p:nn
1402                 { \cs_if_exist_p:N \fpeval }
1403                 { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1404             } { }
1405         }
1406         { \prg_return_true: }
1407         { \prg_return_false: }
1408     }
1409 }

```

(End definition for `_physicx_if_can_num:n`.)

`\diagonalmatrix` Define `\diagonalmatrix`.

```

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

```

```

1461         {
1462             \exp_after:wN
1463             \physicx_matrix_use_r_c:nn
1464             #3 {{##1}} {{####1}} &
1465         }
1466     }
1467     \tl_put_right:Nx \l__physicx_matrix_body_tl
1468     {
1469         \exp_after:wN
1470         \physicx_matrix_use_r_c:nn
1471         #3 {{##1}} {{ \int_use:N #2 }} \[\dim_use:N \l__physicx_matrix_sep_dim]
1472     }
1473 }
1474 \int_step_inline:nn { #2 - 1 }
1475 {
1476     \tl_put_right:Nx \l__physicx_matrix_body_tl
1477     {
1478         \exp_after:wN
1479         \physicx_matrix_use_r_c:nn
1480         #3 {{ \int_use:N #1 }} {{##1}} &
1481     }
1482 }
1483 \tl_put_right:Nx \l__physicx_matrix_body_tl
1484 {
1485     \exp_after:wN
1486     \physicx_matrix_use_r_c:nn
1487     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1488 }
1489 }

```

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

__physicx_declare_init:

```

1490 \cs_new:Npn \__physicx_matrix_enhanced_init:
1491 {
1492     \seq_if_empty:NF \l__physicx_row_list_seq
1493     {
1494         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1495         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1496             { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1497     }
1498     \seq_if_empty:NF \l__physicx_col_list_seq
1499     {
1500         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1501         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1502             { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1503     }
1504 }

```

(End definition for __physicx_declare_init:.)

\commamatrix Define \commamatrix.

```

1505 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1506 {
1507     \group_begin:

```

```

1508 \keys_set:nn { physicx/matrix } {#3}
1509 \tl_if_empty:nF {#4}
1510 { \keys_set:nn { physicx/matrix } { array = {#4} } }
1511 \IfBooleanT {#1}
1512 { \keys_set:nn { physicx/matrix } { saveto = \physicxtmp } }
1513 \tl_set:Nx \l__physicx_matrix_array_tl
1514 { \__physicx_expand:w \l__physicx_matrix_array_tl }
1515 \bool_lazy_or:nnTF
1516 { \bool_if_p:n {#2} }
1517 { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1518 { \__physicx_commmatrix_enhanced: }
1519 {
1520 \tl_replace_all:Nox \l__physicx_matrix_array_tl
1521 { \physicx@cr } { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1522 \tl_replace_all:Non \l__physicx_matrix_array_tl
1523 { \physicx@align } { & }
1524 \tl_set_eq:NN \l__physicx_matrix_body_tl
1525 \l__physicx_matrix_array_tl
1526 }
1527 \__physicx_matrix_save_or_print:
1528 \group_end:
1529 }
1530 \cs_new_nopar:Npn \__physicx_matrix_save_or_print:
1531 {
1532 \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1533 {
1534 \exp_after:wN \tl_gset_eq:NN
1535 \l__physicx_matrix_save_tl
1536 \l__physicx_matrix_body_tl
1537 }
1538 {
1539 \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1540 \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1541 \fi:
1542 \exp_after:wN \__physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matri
1543 \l__physicx_matrix_body_tl
1544 \__physicx_matrix_end: \l__physicx_matrix_after_end_tl
1545 }
1546 }
1547 \cs_new:Npn \__physicx_commmatrix_enhanced:
1548 {
1549 \tl_clear:N \l__physicx_matrix_body_tl
1550 \int_zero:N \l__physicx_tmpa_int
1551 \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1552 \l__physicx_matrix_array_tl
1553 \int_set:Nn \l__physicx_matrix_rows_int
1554 { \seq_count:N \l__physicx_tmp_seq }
1555 \__physicx_matrix_enhanced_init:
1556 \bool_if:NTF \l__physicx_matrix_expand_element_bool
1557 {
1558 \seq_map_tokens:Nn \l__physicx_tmp_seq
1559 {
1560 \int_incr:N \l__physicx_tmpa_int
1561 \exp_args:NV \__physicx_commmatrix_enhanced_aux:nNn

```

```

1562         \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_e:nnn
1563     }
1564 }
1565 {
1566     \seq_map_tokens:Nn \l__physicx_tmp_seq
1567     {
1568         \int_incr:N \l__physicx_tmpa_int
1569         \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1570         \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_ne:nnn
1571     }
1572 }
1573 }
1574 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1575 {
1576     \seq_set_split:Non \l__physicx_tmp_col_seq
1577     { \physicx@align } {#3}
1578     \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1579     \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1580     { #2 {##2} {#1} {##1} }
1581     \tl_put_right:Nx \l__physicx_matrix_body_tl
1582     {
1583         \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1584         \if_int_compare:w \l__physicx_matrix_rows_int = #1
1585             \scan_stop:
1586         \else:
1587             \[\dim_use:N \l__physicx_matrix_sep_dim]
1588         \fi:
1589     }
1590 }
1591 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1592 {
1593     \seq_put_right:Nx \l__physicx_tmp_coled_seq
1594     {
1595         \text_expand:n % \text_expand:n do the magic thing, but slower
1596         {
1597             \physicx@matricelement { #1 }
1598             { \__physicx_matrix_row_iterate:n {#2} }
1599             { \__physicx_matrix_col_iterate:n {#3} }
1600         }
1601     }
1602 }
1603 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1604 {
1605     \seq_put_right:No \l__physicx_tmp_coled_seq
1606     {
1607         \physicx@matricelement {#1}
1608         { \__physicx_matrix_row_iterate:n {#2} }
1609         { \__physicx_matrix_col_iterate:n {#3} }
1610     }
1611 }

```

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

\generalmatrix Define \generalmatrix.


```

1612 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1613 {
1614   \IfBooleanTF {#2}
1615   {
1616     \group_begin:
1617     \IfBooleanTF {#1}
1618     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicxtmp } }
1619     { \keys_set:nn { physicx/matrix } {#4} }
1620     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1621     \physicx_construct:nnn
1622     {
1623       \tl_if_empty:NTF \l__physicx_matrix_main_tl
1624       {
1625         \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1626       }
1627       { \physicx_matrix_array_parse_main: }
1628     }
1629     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1630     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1631     \__physicx_generalmatrix:
1632     \__physicx_matrix_save_or_print:
1633     \group_end:
1634   }
1635   {
1636     \IfBooleanTF {#1}
1637     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1638     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1639     \qxmatrix = * [#4]
1640   }
1641 }
1642 \cs_new:Npn \__physicx_generalmatrix:
1643 {
1644   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1645   {
1646     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1647     \__physicx_matrix_appto_body_e:off
1648   }
1649   {
1650     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1651     \__physicx_matrix_appto_body_ne:off
1652   }
1653   \__physicx_matrix_transpose:N
1654   \__physicx_matrix_generate_body:NNNN
1655   \__physicx_generalmatrix_generate:nnn
1656 }

```

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

__physicx_matrix_generate_body:NNNN

```

1657 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1658 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1659 {
1660   \__physicx_matrix_enhanced_init:
1661   \int_step_inline:nn { #1 - 1 }

```

```

1662 {
1663   \int_step_inline:nn { #2 - 1 }
1664   {
1665     \tl_set:Nx \l__physicx_tmp_tl
1666     {
1667       \exp_after:wN
1668       \physicx_matrix_use_r_c:nn
1669       #3 {{##1}} {{####1}}
1670     }
1671     #4 \l__physicx_tmp_tl {##1} {####1}
1672     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1673   }
1674   \tl_set:Nx \l__physicx_tmp_tl
1675   {
1676     \exp_after:wN
1677     \physicx_matrix_use_r_c:nn
1678     #3 {{##1}} {{ \int_use:N #2 }}
1679   }
1680   #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1681   \tl_put_right:Nx \l__physicx_matrix_body_tl
1682   { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1683 }
1684 \int_step_inline:nn { #2 - 1 }
1685 {
1686   \tl_set:Nx \l__physicx_tmp_tl
1687   {
1688     \exp_after:wN
1689     \physicx_matrix_use_r_c:nn
1690     #3 {{ \int_use:N #1 }} {{##1}}
1691   }
1692   #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}
1693   \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1694 }
1695 \tl_set:Nx \l__physicx_tmp_tl
1696 {
1697   \exp_after:wN
1698   \physicx_matrix_use_r_c:nn
1699   #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1700 }
1701 #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1702 }

```

(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
1703 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1704 {
1705   \tl_put_right:Nx \l__physicx_matrix_body_tl
1706   {
1707     \text_expand:n
1708     {
1709       \physicx@matricelement {#1}
1710       { \__physicx_matrix_row_iterate:n {#2} }
1711       { \__physicx_matrix_col_iterate:n {#3} }

```

```

1712     }
1713   }
1714 }
1715 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1716 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1717 {
1718   \tl_put_right:No \l__physicx_matrix_body_tl
1719   {
1720     \physicx@matricelement {#1}
1721     { \__physicx_matrix_row_iterate:n {#2} }
1722     { \__physicx_matrix_col_iterate:n {#3} }
1723   }
1724 }
1725 \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

```

1726 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1727 {
1728   \bool_if:NTF \l__physicx_matrix_transpose_bool
1729   {
1730     #1
1731     \l__physicx_matrix_cols_int
1732     \l__physicx_matrix_rows_int
1733     \use_ii_i:nn
1734   }
1735   {
1736     #1
1737     \l__physicx_matrix_rows_int
1738     \l__physicx_matrix_cols_int
1739     \use:nn
1740   }
1741 }

```

(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.

```

1742 \cs_new:Npn \physicx_construct:nnn #1#2#3
1743 {
1744   \l__physicx_matrix_beginning_tl
1745   \__physicx_adi:nnn {#1} {#2} {#3}
1746   \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1747   {
1748     \int_incr:N \l__physicx_matrix_cols_int
1749     \__physicx_matrix_last_aux_c:
1750     \int_incr:N \l__physicx_matrix_cols_int
1751   }
1752   \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1753   {
1754     \int_incr:N \l__physicx_matrix_rows_int
1755     \__physicx_matrix_last_aux_r:
1756     \int_incr:N \l__physicx_matrix_rows_int

```

```

1757     }
1758     \bool_lazy_or:nnF
1759     { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1760     { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1761     {
1762         \physicx_matrix_set_r_c:nnn
1763         { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1764         { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1765         { \ddots }
1766     }
1767     \bool_if:NT \l__physicx_matrix_infinite_bool
1768     {
1769         \int_incr:N \l__physicx_matrix_rows_int
1770         \int_incr:N \l__physicx_matrix_cols_int
1771         \__physicx_matrix_last_aux_c:
1772         \__physicx_matrix_last_aux_r:
1773         \physicx_matrix_set_r_c:nnn
1774         { \int_use:N \l__physicx_matrix_rows_int }
1775         { \int_use:N \l__physicx_matrix_cols_int }
1776         { \ddots }
1777     }
1778     \l__physicx_matrix_ending_tl
1779 }
1780 \cs_new:Npn \__physicx_matrix_last_aux_c:
1781 {
1782     \int_step_inline:nn \l__physicx_matrix_rows_int
1783     {
1784         \physicx_matrix_set_r_c:nnn
1785         {##1} { \int_use:N \l__physicx_matrix_cols_int }
1786         { \cdots }
1787     }
1788 }
1789 \cs_new:Npn \__physicx_matrix_last_aux_r:
1790 {
1791     \int_step_inline:nn \l__physicx_matrix_cols_int
1792     {
1793         \physicx_matrix_set_r_c:nnn
1794         { \int_use:N \l__physicx_matrix_rows_int } {##1}
1795         { \vdots }
1796     }
1797 }

```

(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 1798 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
\NewGeneralMatrix 1799 {
\newdiagonalmatrix 1800 \NewDocumentCommand #2 { t+ m o o m m }
\NewDiagonalMatrix 1801 {
\newcommamatrix 1802 \IfBooleanTF {##1}
\NewCommaMatrix 1803 {
1804     \IfNoValueTF {##3}

```

```

1805         { \newcommand ##2 { #1 + [##5] {##6} } }
1806     {
1807         \IfNoValueTF {##4}
1808         { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1809         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1810     }
1811 }
1812 {
1813     \IfNoValueTF {##3}
1814     { \newcommand ##2 { #1 [##5] {##6} } }
1815     {
1816         \IfNoValueTF {##4}
1817         { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1818         { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1819     }
1820 }
1821 }
1822 \NewDocumentCommand #3 { t+ m m m m }
1823 {
1824     \IfBooleanTF {##1}
1825     { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1826     { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1827 }
1828 }
1829 \__physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1830 \__physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1831 \NewDocumentCommand \newgeneralmatrix { t+ m o o m }
1832 {
1833     \IfBooleanTF {#1}
1834     {
1835         \IfNoValueTF {#3}
1836         { \newcommand #2 { \generalmatrix + {#5} } }
1837         {
1838             \IfNoValueTF {#4}
1839             { \newcommand #2 [#3] { \generalmatrix + {#5} } }
1840             { \newcommand #2 [#3] [#4] { \generalmatrix + {#5} } }
1841         }
1842     }
1843     {
1844         \IfNoValueTF {#3}
1845         { \newcommand #2 { \generalmatrix {#5} } }
1846         {
1847             \IfNoValueTF {#4}
1848             { \newcommand #2 [#3] { \generalmatrix {#5} } }
1849             { \newcommand #2 [#3] [#4] { \generalmatrix {#5} } }
1850         }
1851     }
1852 }
1853 \NewDocumentCommand \NewGeneralMatrix { t+ m m m }
1854 {
1855     \IfBooleanTF {#1}
1856     { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1857     { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1858 }

```

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

1859 `\package`

Index

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

Symbols		bool commands:	
<code>\+</code>	167	<code>\bool_case_false:n</code>	540, 541, 548
<code>\-</code>	167	<code>\bool_case_true:nTF</code>	625, 1391
<code>\</code>	57, 1121, 1128, 1135, 1159, 1471, 1521, 1587, 1682	<code>\bool_if:N</code> TF ..	16, 21, 26, 85, 195, 554, 556, 560, 1053, 1108, 1115, 1125, 1133, 1145, 1152, 1156, 1163, 1167, 1431, 1556, 1644, 1728, 1767
<code>\{</code>	340, 570, 589, 607, 648, 673	<code>\bool_if_p:N</code>	1429, 1517
<code>\}</code>	340, 570, 591, 607, 648, 673	<code>\bool_if_p:n</code>	627, 628, 629, 630, 1428, 1516
A		<code>\bool_lazy_and_p:nn</code>	1396, 1401
<code>\A</code>	162, 167	<code>\bool_lazy_or:nnTF</code>	31, 126, 137, 393, 398, 1427, 1515, 1758
<code>\abs</code>	641	<code>\bool_new:N</code>	9, 10, 11, 12, 13, 51, 66, 456, 458, 732, 733, 734, 742, 756, 764, 765
<code>\absolutevalue</code>	641	<code>\bool_set:Nn</code>	467, 468, 1620
<code>\acomm</code>	641	<code>\bool_set_false:N</code>	95, 183, 250, 1041, 1069, 1084, 1201
<code>\aftergroup</code>	217	<code>\bool_set_inverse:N</code>	802, 928
<code>\anticommutator</code>	641	<code>\bool_set_true:N</code>	129, 140, 150, 190, 246, 249, 1040, 1073, 1077, 1088, 1092, 1195, 1494, 1500
<code>\AtBeginDocument</code>	262	<code>\c_false_bool</code>	585, 605
B		<code>\c_true_bool</code>	568, 578, 585, 605
<code>\begin</code>	345, 346, 347, 348, 349, 350, 351, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 590, 593, 597, 600, 601, 602, 607, 608, 612, 615, 616, 617, 978, 979, 980, 981, 982, 983, 984, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003	<code>\Bqty</code>	641
<code>\bgroup</code>	216, 218	<code>\bqty</code>	641
<code>\Big</code>	623	C	
<code>\big</code>	623	<code>\cdots</code>	828, 1107, 1110, 1111, 1114, 1116, 1147, 1148, 1151, 1153, 1786
<code>\Bigg</code>	623	clist commands:	
<code>\bigg</code>	623	<code>\clist_clear:N</code>	1009
<code>\Biggl</code>	630	<code>\clist_concat:NNN</code>	1016
<code>\biggl</code>	629	<code>\clist_if_empty:nTF</code>	222
<code>\Biggr</code>	630	<code>\clist_if_in:NnTF</code>	1344
<code>\biggr</code>	629	<code>\clist_map_break:n</code>	176, 188
<code>\Bigl</code>	628	<code>\clist_map_inline:nn</code>	82, 172, 184, 1014
<code>\bigl</code>	627	<code>\clist_new:N</code>	50, 740, 741
<code>\Bigr</code>	628	<code>\clist_put_right:Nn</code> ..	890, 896, 1335
<code>\bigr</code>	627	<code>\clist_set:Nn</code>	1006
<code>\boldsymbol</code>	206	<code>\clist_set_eq:NN</code>	1311

G		K	
<code>\generalmatrix</code>	.. 32, 1612, 1836, 1839, 1840, 1845, 1848, 1849, 1856, 1857	keys commands:	
group commands:		<code>\keys_define:nn</code> 201, 225, 291, 293, 770, 772
<code>\group_begin:</code> 380, 1038, 1412, 1507, 1616	<code>\keys_if_exist:nnTF</code> 174, 186
<code>\group_end:</code>	388, 1049, 1447, 1528, 1633	<code>\l_keys_key_str</code> 174, 177, 186, 191, 313, 317, 320, 957, 960
H		<code>\l_keys_path_str</code> 333, 964
<code>\hat</code> 268	<code>\keys_set:nn</code> 3, 177, 191, 280, 281, 292, 319, 372, 381, 771, 959, 1022, 1045, 1046, 1414, 1415, 1508, 1510, 1512, 1618, 1619
hook commands:		keyval commands:	
<code>\hook_gput_code:nnn</code> 37	<code>\keyval_parse:NNn</code> 1312
I		<code>\keyval_parse:nnn</code> 1174
if commands:		L	
<code>\if_bool:N</code> 501, 513, 566, 641	<code>\langle</code> 344, 574, 601, 616
<code>\if_case:w</code> 493	<code>\ldots</code> 828
<code>\if_cs_exist:w</code> 695	<code>\left</code> 216, 374, 399, 589, 596, 601, 607, 608, 611, 615, 616, 617
<code>\if_int_compare:w</code> 486, 1117, 1244, 1247, 1270, 1282, 1539, 1584	<code>\let</code> 273, 274, 534
<code>\IfBooleanT</code> 589, 591, 1511	<code>\lgroup</code> 596, 611
<code>\IfBooleanTF</code>	.. 264, 266, 268, 279, 447, 539, 596, 598, 611, 613, 633, 974, 1039, 1044, 1413, 1614, 1617, 1636, 1637, 1638, 1802, 1824, 1833, 1855	<code>\lVert</code> 658
<code>\IfNoValueTF</code> 414, 420, 1804, 1807, 1813, 1816, 1835, 1838, 1844, 1847	M	
int commands:		<code>\mathbf</code> 264, 266, 268
<code>\int_compare:nNnTF</code> 105, 115, 149	<code>\mathcal</code> 202, 203
<code>\int_compare:nTF</code> 1067, 1082	<code>\mathclose</code> 216, 218
<code>\int_compare_p:nNn</code>	. 127, 128, 138, 139	<code>\mathopen</code> 216, 218
<code>\int_eval:n</code>	.. 28, 1251, 1260, 1278, 1290, 1300, 1301, 1393, 1763, 1764	<code>\matrixquantity</code> 566
<code>\int_incr:N</code>	. 485, 1226, 1560, 1568, 1748, 1750, 1754, 1756, 1769, 1770	<code>\mqty</code> 641
<code>\int_max:nn</code> 688, 690	msg commands:	
<code>\int_new:N</code> 52, 53, 62, 63, 64, 65, 460, 737, 738	<code>\msg_error:nnn</code> 1206
<code>\int_set:Nn</code>	.. 80, 81, 104, 114, 466, 687, 689, 1042, 1043, 1070, 1085, 1553	<code>\msg_error:nnnn</code> 332, 963
<code>\int_set_eq:NN</code> 102, 107, 112, 117, 1229, 1540	<code>\msg_new:nnn</code> 60
<code>\int_step_inline:nn</code> 1095, 1377, 1379, 1456, 1458, 1474, 1661, 1663, 1684, 1782, 1791	<code>\msg_new:nnnn</code> 54
<code>\int_step_inline:nnn</code> 142, 152, 1101, 1139	N	
<code>\int_use:N</code> 1471, 1480, 1487, 1678, 1680, 1690, 1692, 1699, 1701, 1774, 1775, 1785, 1794	<code>\nabla</code> 272
<code>\int_zero:N</code> 1223, 1550	<code>\NewCommaMatrix</code> 1798
<code>\interval</code> 28, 1397, 1398	<code>\newcommamatrix</code> 1798
		<code>\newcommand</code> 417, 423, 427, 436, 1805, 1808, 1809, 1814, 1817, 1818, 1836, 1839, 1840, 1845, 1848, 1849
		<code>\NewDiagonalMatrix</code> 1798
		<code>\newdiagonalmatrix</code> 1798
		<code>\NewDocumentCommand</code> 277, 337, 410, 412, 443, 445, 972, 1800, 1822, 1825, 1826, 1831, 1853, 1856, 1857
		<code>\NewGeneralMatrix</code> 1798
		<code>\newgeneralmatrix</code> 1798
		<code>\NewXQuantity</code> 378
		<code>\newxquantity</code> 378

<code>\nobreak</code>	580, 581, 582	<code>\physicx_new_type:nnn</code>	200, 338, 969, 971
<code>\norm</code>	641	<code>\physicx_option_or:nn</code>	29
O			
<code>\OOrder</code>	681	<code>\physicx_option_or:nnTF</code>	642, 652, 670, 674
<code>\oorder</code>	641	<code>\c_physicx_Order_tl</code>	203, 255, 662, 682
or commands:		<code>\c_physicx_order_tl</code>	202, 254, 666, 684
<code>\or:</code>	494, 495, 496	<code>\physicx_parse_range:nnN</code>	90, 92, 1324, 1326
<code>\order</code>	641	<code>\physicx_parse_range:nnnN</code>	62
<code>\Ordersymbol</code>	255	<code>\physicx_parse_range_check:</code> ..	62, 901
<code>\ordersymbol</code>	254	<code>\physicx_parse_range_nocheck:</code> ..	62, 902
P			
<code>\PassOptionsToPackage</code>	222	<code>\physicx_qxmatrix:nnn</code> ...	1047, 1051
<code>\pb</code>	641	<code>\physicx_right:</code> ...	217, 401, 535, 635
peek commands:		<code>\physicx_right:N</code>	219, 405, 627, 628, 629, 630
<code>\peek_charcode_ignore_spaces:N</code>	534	<code>\physicx_search_also:nn</code>	170, 181, 198, 199
physicx commands:		<code>\physicx_search_also:nnTF</code> ..	326, 949
<code>\physicx_bf:</code>	206, 210, 214, 264, 266, 268	<code>\physicx_short:</code>	19
<code>\physicx_compat:</code>	14	<code>\physicx_unimath:</code>	44
<code>\physicx_compat:TF</code>	252, 659	<code>\physicx_unimath:TF</code> ..	40, 41, 42, 261
<code>\physicx_construct:nnn</code>	1416, 1621, 1742	<code>\physicx_use_amsymb_type:</code> ..	204, 260
<code>\physicx_declare_legacy_paren:NnnnN</code>	620, 645, 646, 647, 648,	<code>\physicx_use_uni_bf_type:</code>	212
<code>\physicx_declare_legacy_quantity:nnN</code>	650, 657, 661, 665, 668, 672, 681, 683	<code>\physicx_use_uni_bfit_type:</code> ..	208, 259
<code>\physicx_if_num:n</code>	160	<code>\physicx_xquantity:nn</code>	378
<code>\physicx_if_num:nTF</code>	957, 1065, 1080, 1388	physicx internal commands:	
<code>\physicx_if_num_sign:n</code>	165	<code>__physicx_adi:nnn</code>	767, 933, 934, 935, 936, 937, 938, 1745
<code>\physicx_left:</code>	216, 401, 535, 635	<code>\l__physicx_begin_int</code> ..	62, 102,
<code>\physicx_left:N</code>	218, 403, 627, 628, 629, 630	<code>\l__physicx_cmd_arg_int</code>	104, 105, 107, 138, 139, 143, 149, 153
<code>\physicx_mathtools:</code>	24	<code>\l__physicx_cmd_arg_spec_tl</code>	460, 466, 485, 486, 493
<code>\physicx_mathtools:TF</code>	352, 985	<code>\l__physicx_cmd_arg_spec_tl</code>	459, 465, 478, 487
<code>\physicx_matrix_array_parse:n</code>	1353, 1625	<code>\l__physicx_cmd_auto_body_bool</code> ..	458, 468, 513, 556, 560
<code>\physicx_matrix_array_parse_-</code>		<code>\l__physicx_cmd_auto_body_tl</code> ..	457, 464, 480, 514, 515
<code>main:</code>	1375, 1627	<code>\l__physicx_cmd_noauto_body_bool</code>	456, 467, 501, 554
<code>\physicx_matrix_diag_parse:n</code> ..	891, 1172, 1418, 1422, 1423, 1629	<code>\l__physicx_cmd_noauto_body_tl</code> ..	455, 463, 479, 502, 503
<code>\physicx_matrix_item_parse:n</code> ..	897, 1309, 1426, 1630	<code>\l__physicx_col_list_seq</code>	744, 798, 1498, 1502
<code>\physicx_matrix_new_type:nn</code> ...	968	<code>__physicx_commamatrix_enhanced:</code>	1518, 1547
<code>\physicx_matrix_new_type:nnn</code> ..	968	<code>__physicx_commamatrix_enhanced_-</code>	1561, 1569, 1574
<code>\physicx_matrix_set_r_c:nnn</code> ..	726,	<code>aux:nNn</code>	1561, 1569, 1574
<code>869, 872, 875, 878, 881, 1227, 1235,</code>		<code>__physicx_commamatrix_enhanced_-</code>	1562, 1591
<code>1250, 1259, 1276, 1288, 1299, 1346,</code>		<code>aux_e:nnn</code>	1570, 1603
<code>1370, 1381, 1762, 1773, 1784, 1793</code>		<code>__physicx_commamatrix_enhanced_-</code>	
<code>\physicx_matrix_use_r_c:nn</code>	693, 1463, 1470,	<code>aux_ne:nnn</code>	
<code>1479, 1486, 1668, 1677, 1689, 1698</code>			

<code>\g__physicx_compat_bool</code>	<code>\l__physicx_matrix_array_tl</code>
..... 11, 16, 227, 246 735, 774, 1513,
<code>__physicx_declare_init:</code> 1490	1514, 1520, 1522, 1525, 1552, 1625
<code>__physicx_declare_init:nnn</code> 461, 473	<code>__physicx_matrix_autocalc:nn</code> ...
<code>__physicx_declare_legacy_-</code> 755, 790, 792, 1308, 1359, 1365
<code>quantity_aux:nnnn</code> 483, 524	<code>__physicx_matrix_begin:w</code>
<code>__physicx_declare_legacy_-</code> 753, 904, 1542
<code>quantity_aux:NNnnn</code> .. 476, 527, 551	<code>\l__physicx_matrix_beginning_tl</code> .
<code>__physicx_declare_legacy_-</code> 768, 939, 941, 1744
<code>quantity_aux:nw</code> 474, 520, 525	<code>\l__physicx_matrix_body_tl</code>
<code>__physicx_diagonalmatrix_diag_-</code> 736, 1063, 1098,
<code>main:</code> 1196, 1202, 1240, 1245	1103, 1104, 1110, 1111, 1116, 1121,
<code>__physicx_diagonalmatrix_-</code>	1128, 1130, 1132, 1134, 1135, 1137,
<code>enhanced:nnn</code> 1433, 1437, 1452	1141, 1142, 1147, 1148, 1153, 1159,
<code>__physicx_diagonalmatrix_-</code>	1160, 1162, 1164, 1165, 1460, 1467,
<code>generate_body:NNN</code> 1445, 1454	1476, 1483, 1524, 1536, 1543, 1549,
<code>__physicx_diagonalmatrix_-</code>	1581, 1672, 1681, 1693, 1705, 1718
<code>generate_enhanced_body:NNN</code> ..	<code>__physicx_matrix_calc:nn</code>
..... 1444, 1449 685, 790, 1187
<code>__physicx_diagonalmatrix_no_-</code>	<code>__physicx_matrix_col_iterate:n</code> .
<code>diag:</code> 1203, 1232, 1241	752, 885, 1501, 1599, 1609, 1711, 1722
<code>__physicx_diagonalmatrix_set_-</code>	<code>\l__physicx_matrix_cols_int</code>
<code>diag:</code> 1197, 1221 689, 690,
<code>\l__physicx_end_int</code> 63,	738, 787, 1043, 1082, 1085, 1101,
112, 114, 115, 117, 139, 143, 149, 153	1131, 1139, 1161, 1168, 1169, 1223,
<code>__physicx_expand:w</code> 746, 777,	1226, 1230, 1278, 1290, 1301, 1326,
779, 781, 784, 1212, 1323, 1357, 1514	1379, 1539, 1540, 1731, 1738, 1748,
<code>__physicx_generalmatrix:</code> 1631, 1642	1750, 1764, 1770, 1775, 1785, 1791
<code>__physicx_generalmatrix_-</code>	<code>\l__physicx_matrix_diag_bool</code> ...
<code>generate:nnn</code> 1646, 1650, 1655 742, 1195, 1201
<code>__physicx_if_can_num:n</code> 1386	<code>__physicx_matrix_diag_calc:nn</code> ..
<code>__physicx_if_keyval:nTF</code> .. 158, 1421 1186, 1191, 1236,
<code>\l__physicx_invalid_range_bool</code> ..	1253, 1262, 1271, 1283, 1294, 1307
..... 66, 85, 95, 129, 140, 150	<code>\l__physicx_matrix_diag_clist</code> ...
<code>\l__physicx_item_ignore_clist</code> 740, 888, 890, 1418, 1629
..... 1311, 1335, 1344	<code>__physicx_matrix_diag_parse_-</code>
<code>__physicx_loadpackage_options:nnn</code>	<code>aux:n</code> 1175, 1180
..... 220, 231, 233, 235	<code>__physicx_matrix_diag_parse_-</code>
<code>\g__physicx_mathtools_bool</code>	<code>aux:nn</code> 1176, 1208
..... 9, 26, 249, 250	<code>__physicx_matrix_diag_parse_-</code>
<code>__physicx_matrix_add_data:nn</code> ...	<code>aux_anti:n</code> 1216, 1268
..... 893, 899, 1010, 1012	<code>__physicx_matrix_diag_parse_-</code>
<code>\l__physicx_matrix_after_begin_-</code>	<code>aux_regu:n</code> 1219, 1242
<code>tl</code> 762, 909, 911, 1542	<code>\l__physicx_matrix_dotcol_bool</code> ..
<code>\l__physicx_matrix_after_end_tl</code> 734, 1084, 1088,
..... 763, 912, 914, 1544	1092, 1108, 1133, 1145, 1163, 1167
<code>__physicx_matrix_appto_body_-</code>	<code>\l__physicx_matrix_dotrow_bool</code> ..
<code>e:nnn</code> 1056, 1434, 1647, 1703 733, 1069, 1073, 1077, 1125
<code>__physicx_matrix_appto_body_-</code>	<code>__physicx_matrix_element_-</code>
<code>ne:nnn</code> 1060, 1438, 1651, 1703	<code>aux:nnn</code> 806, 812, 815, 821,
<code>\l__physicx_matrix_args_tl</code>	824, 830, 833, 839, 842, 848, 851, 856
..... 761, 907, 908, 1542	<code>__physicx_matrix_end:</code> 905, 1544
	<code>__physicx_matrix_end:w</code>
	754

\l__physicx_matrix_ending_tl ...	__physicx_matrix_set_r_c_-
..... 769, 942, 944, 1778	nock:nnn 702, 727, 870
\l__physicx_matrix_enhanced_bool	\l__physicx_matrix_tmparr_c_sep .
..... 765, 925, 928, 1429, 1517 1364, 1367, 1368
__physicx_matrix_enhanced_init:	\l__physicx_matrix_tmparr_r_sep .
..... 1490, 1555, 1660 1358, 1360, 1362
\l__physicx_matrix_expand_-	__physicx_matrix_transpose:N ...
element_bool 756, 1443, 1653, 1726
864, 1053, 1431, 1494, 1500, 1556, 1644	\l__physicx_matrix_transpose_-
__physicx_matrix_generate_-	bool 764, 920, 1728
body:NNNN 1451, 1654, 1657	\l__physicx_max_int 64, 81, 112, 115, 117, 127, 138
\l__physicx_matrix_infinite_bool	\l__physicx_min_int 65, 80, 102, 105, 107, 128
..... 732, 799, 802, 1040,	__physicx_nauto_case:nnnn . 529, 552
1041, 1115, 1152, 1156, 1620, 1767	__physicx_new_matrix_cmd:NNN . 1798
\l__physicx_matrix_item_clist ...	__physicx_new_xquantity_aux:w ..
..... 741, 894, 896, 1426, 1630 416, 422, 426, 430, 433
__physicx_matrix_item_parse_-	__physicx_parse_range_aux:n 84, 93
aux:n 1313, 1318	__physicx_parse_range_range: ...
__physicx_matrix_item_parse_- 70, 75, 120, 157
aux:nn 1314, 1319	__physicx_parse_range_range_-
__physicx_matrix_last_aux_c: ...	check: 70, 135, 157
..... 1749, 1771, 1780	__physicx_parse_range_range_-
__physicx_matrix_last_aux_r: ...	nocheck: 75, 147
..... 1755, 1772, 1789	__physicx_parse_range_single:n .
\l__physicx_matrix_last_col_tl 69, 74, 122, 134
..... 751, 887, 1746, 1760	__physicx_parse_range_single_-
\l__physicx_matrix_last_row_tl ..	check:n 69, 124, 134
..... 750, 886, 1752, 1759	__physicx_parse_range_single_-
\l__physicx_matrix_main_tl	nocheck:n 74, 132
..... 739, 794, 1382, 1623	\g__physicx_physics_bool 10
__physicx_matrix_row_iterate:n .	\l__physicx_quantity_args_tl ...
749, 884, 1495, 1598, 1608, 1710, 1721 283, 305, 306, 385
\l__physicx_matrix_rows_int	\l__physicx_quantity_code_tl ...
..... 687, 688, 737, 786, 1042, 284, 307, 382, 386
1067, 1070, 1095, 1117, 1229, 1324,	\l__physicx_quantity_left_size_-
1377, 1553, 1584, 1732, 1737, 1754,	tl 285, 299, 394, 399, 403
1756, 1763, 1769, 1774, 1782, 1794	\l__physicx_quantity_left_tl ...
__physicx_matrix_save_or_print: 286, 297, 384
..... 1048, 1446, 1527, 1530, 1632	\l__physicx_quantity_post_tl ...
\l__physicx_matrix_save_tl 917, 919, 1532, 1535 287, 296, 408
\l__physicx_matrix_sep_dim	\l__physicx_quantity_pre_tl
..... 766, 915, 1121, 1128, 288, 295, 392
1135, 1159, 1471, 1521, 1587, 1682	\l__physicx_quantity_right_size_-
__physicx_matrix_set_data:nn ...	tl 289, 300, 395, 400, 405
..... 892, 898, 1007	\l__physicx_quantity_right_tl ...
__physicx_matrix_set_r_c_- 290, 298, 387
ckall:nnn 724, 882	__physicx_qxmatrix_appto_-
__physicx_matrix_set_r_c_-	body:nnn 1055, 1059,
ckep:nnn 709, 873	1099, 1105, 1112, 1138, 1143, 1149
__physicx_matrix_set_r_c_-	\g__physicx_reqty_bool 13, 237, 566, 641
ckig:nnn 704, 876	
__physicx_matrix_set_r_c_-	
ckigep:nnn 715, 725, 879	

\l__physicx_row_list_seq	44
..... 743, 796, 1492, 1496	
\g__physicx_short_bool ...	12, 21, 229
\l__physicx_tmp_col_seq ..	1576, 1579
\l__physicx_tmp_coled_seq	
..... 1578, 1583, 1593, 1605	
\l__physicx_tmp_colnum_seq	
..... 1327, 1333, 1342	
\l__physicx_tmp_rownum_seq	
..... 1325, 1331, 1340	
\l__physicx_tmp_seq	
..... 1551, 1554, 1558, 1566	
\l__physicx_tmp_tl	
..... 488, 502, 514, 1665, 1671,	
1674, 1680, 1686, 1692, 1695, 1701	
\l__physicx_tmpa_bool 51, 183, 190, 195	
\l__physicx_tmpa_clist	50
\l__physicx_tmpa_int	
.... 52, 1550, 1560, 1562, 1568, 1570	
\l__physicx_tmpa_seq	
..... 86, 96, 130, 133, 144, 154	
\l__physicx_tmpa_tl	
100, 101, 104, 110, 111, 114, 313, 314	
\l__physicx_tmparr_tl	
..... 1355, 1356, 1357, 1358	
\l__physicx_tmpb_int	53
\l__physicx_tmpb_seq ...	99, 100, 110
\l__physicx_tmpdiag_seq 1213, 1224,	
1234, 1237, 1238, 1248, 1254, 1255,	
1257, 1263, 1264, 1272, 1273, 1274,	
1284, 1285, 1286, 1295, 1296, 1297	
\l__physicx_tmpdiag_tl	
..... 1210, 1211, 1212, 1213	
\l__physicx_tmpitem_tl	
..... 1321, 1322, 1323, 1329, 1347	
__physicx_xquantity_aux:nnnn ...	
..... 383, 390, 411	
\physicxempty .	698, 712, 720, 757, 758, 866
\physicxexcept	760, 837, 846, 855, 861, 863
\PHYSICXIGNORE	7, 8, 706, 717, 1329
\physicxset	277
\physicxtmp ...	454, 1045, 1414, 1512, 1618
\poissonbracket	641
\pqty	641
prg commands:	
\prg_generate_conditional_-	
variant:Nnn	199
\prg_new_conditional:Npnn	
... 14, 19, 24, 29, 160, 165, 181, 1386	
\prg_replicate:nn	1131, 1161
\prg_return_false:	17,
22, 27, 35, 47, 163, 168, 196, 1407	
\prg_return_true:	17,
22, 27, 34, 47, 163, 168, 196, 1389, 1406	
\prg_set_conditional:Npnn	44
\ProcessKeysPackageOptions	243
Q	
\qty	641
\quantity	566, 644
quark commands:	
\quark_if_recursion_tail_stop:n .	
..... 522, 523	
\q_recursion_stop	475
\q_recursion_tail	475
\qxmatrix	1027, 1639
\qxqty	452
R	
\rangle	344, 574, 601, 616
regex commands:	
\regex_match:nnTF ..	162, 167, 846, 855
\RequirePackage	223
\rgroup	598, 613
\rightright	217, 375, 400, 591,
598, 601, 607, 608, 613, 615, 616, 617	
\rVert	658
S	
\sb	748, 1098, 1103, 1110, 1137, 1141, 1147
scan commands:	
\scan_stop:	1118, 1585
seq commands:	
\c_empty_seq	79, 1578
\seq_clear:N	96
\seq_concat:NNN	86
\seq_count:N	1237, 1238, 1254,
1255, 1263, 1264, 1272, 1273, 1284,	
1285, 1295, 1296, 1360, 1367, 1554	
\seq_if_empty:NFT	1492, 1498
\seq_item:Nn	1496, 1502
\seq_map_indexed_inline:Nn	
..... 1224, 1234, 1248, 1257,	
1274, 1286, 1297, 1362, 1368, 1579	
\seq_map_inline:Nn	
..... 1331, 1333, 1340, 1342	
\seq_map_tokens:Nn	1558, 1566
\seq_new:N	743, 744
\seq_pop_left:NN	100, 110
\seq_put_right:Nn	
..... 130, 133, 144, 154, 1593, 1605	
\seq_set_eq:NN	79, 1578
\seq_set_split:Nnn	5, 99,
796, 798, 1213, 1358, 1364, 1551, 1576	
\seq_use:Nn	1583
\setmathfont	269
\setmatrixdata	1005
\setmatrixtype	946, 968, 978, 979, 980,
981, 982, 983, 984, 987, 988, 989,	

<code>\vqty</code>	<u>641</u>		Z	
<code>\vysmbkcircle</code>	270	<code>\Z</code>	162, 167
			X		
<code>\xquantity</code>	410			