

The `physicx` package

Wenjian Chern (Longaster*)

May 8, 2022, version v0.3.2

Abstract

`physicx`

1 Implementation

```
1 <*package>
2 <@@=physicx>
3 \cs_generate_variant:Nn \keys_set:nn { nx , on , ox }
4 \cs_generate_variant:Nn \use:n { nnnn { nnno }
5 \cs_generate_variant:Nn \seq_set_split:Nnn { Non, NVV, c, cnV, cVV }
6 \cs_generate_variant:Nn \tl_replace_all:Nnn { Non, Nox }
7 \cs_new:Npn \PHYSICXIGNORE
8 { \exp_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     \cs_set_eq:NN \divisionsymbol \div
274     \cs_set_eq:NN \div \divergence
275   }
276   \physicx_compat:T {
277     \AtBeginDocument{
278       \let\real\Re \DeclareDocumentCommand\Re{g}{\IfNoValueTF{#1}{\operatorname{Re}}{\fbrace
279       \let\imaginary\Im \DeclareDocumentCommand\Im{g}{\IfNoValueTF{#1}{\operatorname{Im}}{\fbrace
280     }
281   }
282 }

```

`\physicxset` `physicx` setup command.

```

283 \NewDocumentCommand \physicxset { s m }
284 {
285   \IfBooleanTF {#1}
286   { \keys_set:nn { physicx/#2 } }
287   { \keys_set:nn { physicx } {#2} }
288 }

```

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

1.2 Quantity things

1.2.1 New quantity interfaces

```

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

```

```

340     }
341   } ,
342 }
343 \NewDocumentCommand \setquantitytype { >{ \TrimSpaces } m }
344 { \physicx_new_type:nnn { quantity } {#1} }
345 \setquantitytype { b } { left={[] , right={}} , }
346 \setquantitytype { B } { left={\{ , right={\}} , }
347 \setquantitytype { p } { left={({ , right={})} , }
348 \setquantitytype { v } { left=\vert , right=\vert , }
349 \setquantitytype { V } { left=\Vert , right=\Vert , }
350 \setquantitytype { a } { left=\langle , right=\rangle , }
351 \setquantitytype { m } { left=\begin{matrix} , right=\end{matrix} , noauto }
352 \setquantitytype { bm } { left=\begin{bmatrix} , right=\end{bmatrix} , noauto }
353 \setquantitytype { Bm } { left=\begin{Bmatrix} , right=\end{Bmatrix} , noauto }
354 \setquantitytype { pm } { left=\begin{pmatrix} , right=\end{pmatrix} , noauto }
355 \setquantitytype { vm } { left=\begin{vmatrix} , right=\end{vmatrix} , noauto }
356 \setquantitytype { Vm } { left=\begin{Vmatrix} , right=\end{Vmatrix} , noauto }
357 \setquantitytype { sm } { left=\begin{smallmatrix} , right=\end{smallmatrix} , noauto }
358 \physicx_mathtools:T
359 {
360   \setquantitytype { m* } { left=\begin{matrix*} , right=\end{matrix*} , noauto }
361   \setquantitytype { bm* } { left=\begin{bmatrix*} , right=\end{bmatrix*} , noauto }
362   \setquantitytype { Bm* } { left=\begin{Bmatrix*} , right=\end{Bmatrix*} , noauto }
363   \setquantitytype { pm* } { left=\begin{pmatrix*} , right=\end{pmatrix*} , noauto }
364   \setquantitytype { vm* } { left=\begin{vmatrix*} , right=\end{vmatrix*} , noauto }
365   \setquantitytype { Vm* } { left=\begin{Vmatrix*} , right=\end{Vmatrix*} , noauto }
366   \setquantitytype { sm* } { left=\begin{smallmatrix*} , right=\end{smallmatrix*} , noauto }
367   \setquantitytype { sbm } { left=\begin{bsmallmatrix} , right=\end{bsmallmatrix} , noauto }
368   \setquantitytype { sBm } { left=\begin{Bsmallmatrix} , right=\end{Bsmallmatrix} , noauto }
369   \setquantitytype { spm } { left=\begin{psmallmatrix} , right=\end{psmallmatrix} , noauto }
370   \setquantitytype { svm } { left=\begin{vsmallmatrix} , right=\end{vsmallmatrix} , noauto }
371   \setquantitytype { sVm } { left=\begin{Vsmallmatrix} , right=\end{Vsmallmatrix} , noauto }
372   \setquantitytype { sbm* } { left=\begin{bsmallmatrix*} , right=\end{bsmallmatrix*} , noauto }
373   \setquantitytype { sBm* } { left=\begin{Bsmallmatrix*} , right=\end{Bsmallmatrix*} , noauto }
374   \setquantitytype { spm* } { left=\begin{psmallmatrix*} , right=\end{psmallmatrix*} , noauto }
375   \setquantitytype { svm* } { left=\begin{vsmallmatrix*} , right=\end{vsmallmatrix*} , noauto }
376   \setquantitytype { sVm* } { left=\begin{Vsmallmatrix*} , right=\end{Vsmallmatrix*} , noauto }
377 }
378 \keys_set:nn { physicx/quantity }
379 {
380   left-size = \left ,
381   right-size = \right ,
382   type = p ,
383 }
\physicx_xquantity:nn
\newxquantity
\NewXQuantity
384 \cs_new:Npn \physicx_xquantity:nn #1#2
385 {
386   \group_begin:
387   \keys_set:nn { physicx/quantity } {#1}
388   \tl_if_empty:nF {#2} { \tl_set:Nn \l__physicx_quantity_code_tl {#2} }
389   \__physicx_xquantity_aux:oooo
390   { \l__physicx_quantity_left_tl }
391   { \l__physicx_quantity_args_tl }

```



```

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

```

```

446         \exp_not:c { \cs_to_str:N #1~unstar }
447     }
448 }
449 \NewDocumentCommand \NewXQuantity { m m m m }
450 {
451     \NewDocumentCommand #1 { s #2 }
452     {
453         \IfBooleanTF {##1}
454         { \physicsx_quantity:nn { #3 , noauto } {#4} }
455         { \physicsx_quantity:nn { #3 } {#4} }
456     }
457 }
458 \NewXQuantity \qxqty { 0{ } m } { #2 } {#3}
459 \NewXQuantity \txqty { 0{p} 0{ } m } { type={#2}, #3 } {#4}

```

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

1.2.2 Legacy quantity

```

\physicsx_declare_legacy_quantity:nnNn
  \@declarequantitycmd
460 \tl_new:N \physicsx_tmp
461 \tl_new:N \l__physicsx_cmd_noauto_body_tl
462 \bool_new:N \l__physicsx_cmd_noauto_body_bool
463 \tl_new:N \l__physicsx_cmd_auto_body_tl
464 \bool_new:N \l__physicsx_cmd_auto_body_bool
465 \tl_new:N \l__physicsx_cmd_arg_spec_tl
466 \int_new:N \l__physicsx_cmd_arg_int
467 \cs_new:Npn \__physicsx_declare_init:nnn #1#2#3
468 {
469     \tl_clear:N \l__physicsx_cmd_noauto_body_tl
470     \tl_clear:N \l__physicsx_cmd_auto_body_tl
471     \tl_clear:N \l__physicsx_cmd_arg_spec_tl
472     \int_set:Nn \l__physicsx_cmd_arg_int {#1}
473     \bool_set:Nn \l__physicsx_cmd_noauto_body_bool {#2}
474     \bool_set:Nn \l__physicsx_cmd_auto_body_bool {#3}
475 }
476 % noauto, auto, cmd, body
477 \cs_new:Npn \physicsx_declare_legacy_quantity:nnNn #1#2#3#4
478 {
479     \__physicsx_declare_init:nnn { 3 } {#1} {#2}
480     \__physicsx_declare_legacy_quantity_aux:nw #4
481     \q_recursion_tail \q_recursion_tail \q_recursion_stop
482     \__physicsx_declare_legacy_quantity_aux:NcVVV
483     #3 { \cs_to_str:N #3 ~ body }
484     \l__physicsx_cmd_arg_spec_tl
485     \l__physicsx_cmd_noauto_body_tl
486     \l__physicsx_cmd_auto_body_tl
487 }
488 % arg spec, pre, body to replace(start from #4), post
489 \cs_new:Npn \__physicsx_declare_legacy_quantity_aux:nnnn #1#2#3#4
490 {
491     \int_incr:N \l__physicsx_cmd_arg_int
492     \if_int_compare:w \l__physicsx_cmd_arg_int < 10 \exp_stop_f:

```

```

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

```

```

547         { \bool_case_false:n {#5} }
548     }
549 }
550 {
551     \cs_set_protected:Npn #1
552     { #2 \c_empty_tl \c_empty_tl }
553     \DeclareDocumentCommand #2 { m m s #3 }
554     { \bool_case_false:n {#4} }
555 }
556 }
557 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
558 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
559 {
560     \bool_if:NTF \l__physicx_cmd_noauto_body_bool
561     {
562         \bool_if:NTF \l__physicx_cmd_auto_body_bool
563         {#1} {#2}
564     }
565     {
566         \bool_if:NTF \l__physicx_cmd_auto_body_bool
567         {#3} {#4}
568     }
569 }
570 \cs_set_protected:Npn \@declarequantitycmd
571 { \physicx_declare_legacy_quantity:nnNn }

```

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

\backslash quantity \backslash evaluated \backslash matrixquantity \backslash smallmatrixquantity	Redefine some macros in physics package.	<pre> 572 \if_bool:N \g__physicx_reqty_bool 573 \physicx_declare_legacy_quantity:nnNn 574 \c_true_bool \c_true_bool \quantity 575 { 576 { !g } { { \{ } { #4 } { \} } } 577 { !o } { { [} { #5 } {] } } 578 { !d() } { { (} { #6 } {) } } 579 { !d } { { \vert } { #7 } { \vert } } 580 { !d<> } { { \langle } { #8 } { \rangle } } 581 { !d== } { { \Vert } { #9 } { \Vert } } 582 } 583 \physicx_declare_legacy_quantity:nnNn 584 \c_true_bool \c_true_bool \evaluated 585 { 586 { !g } { { . } { #4 \nobreak } { \vert } } 587 { !d[] } { { [} { #5 \nobreak } { \vert } } 588 { !d() } { { (} { #6 \nobreak } { \vert } } 589 } 590 \physicx_declare_legacy_quantity:nnNn 591 \c_true_bool \c_false_bool \matrixquantity 592 { 593 { !g } 594 { 595 { \IfBooleanT{#3}{\left\{ } } </pre>
--	--	---

```

596 { \begin{matrix} #4 \end{matrix} }
597 { \IfBooleanT{#3}{\right\}} }
598 }
599 { !o } { {\begin{bmatrix} } {#5} { \end{bmatrix} } }
600 { !d() }
601 {
602 { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
603 { \begin{matrix} #6 \end{matrix} }
604 { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
605 }
606 { !d|| } { { \begin{vmatrix} } {#7} { \end{vmatrix} } }
607 { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } { \right\rangle } }
608 { !d== } { { \begin{Vmatrix} } {#9} { \end{Vmatrix} } }
609 }
610 \physicsx_declare_legacy_quantity:nnNn
611 \c_true_bool \c_false_bool \smallmatrixquantity
612 {
613 { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } { \right\} } }
614 { !o } { { \left[ } { \begin{smallmatrix} #5 \end{smallmatrix} } { \right]} }
615 { !d() }
616 {
617 { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
618 { \begin{smallmatrix} #6 \end{smallmatrix} }
619 { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
620 }
621 { !d|| } { { \left\vert } { \begin{smallmatrix} #7 \end{smallmatrix} } { \right\vert} }
622 { !d<> } { { \left\langle } { \begin{smallmatrix} #8 \end{smallmatrix} } { \right\rangle } }
623 { !d== } { { \left\Vert } { \begin{smallmatrix} #9 \end{smallmatrix} } { \right\Vert} }
624 }
625 \fi:

```

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

```

\physicsx_declare_legacy_paren:NnnnNnN
  \@declareparenccmd
626 %% cmd, arg spec, replace(start from #6), pre, left, right, post
627 \cs_new:Npn \physicsx_declare_legacy_paren:NnnnNnN #1#2#3#4#5#6#7
628 {
629   \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
630   {
631     \bool_case_true:nF
632     {
633       { \bool_if_p:n {##2} } { #4 \physicsx_left:N \bigl #5 #3 \physicsx_right:N \bigr
634       { \bool_if_p:n {##3} } { #4 \physicsx_left:N \Bigl #5 #3 \physicsx_right:N \Bigr
635       { \bool_if_p:n {##4} } { #4 \physicsx_left:N \biggl #5 #3 \physicsx_right:N \biggr
636       { \bool_if_p:n {##5} } { #4 \physicsx_left:N \Biggl #5 #3 \physicsx_right:N \Biggr
637     }
638     {
639       \IfBooleanTF {##1}
640       { #4 #5 #3 #6 #7 }
641       { #4 \physicsx_left: #5 #3 \physicsx_right: #6 #7 }
642     }
643   }
644 }
645 \cs_set_protected:Npn \@declareparenccmd
646 { \physicsx_declare_legacy_paren:NnnnNnN }

```

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

```

\qty      Redefine some macros in physics package.
\mqty     647 \if_bool:N \g__physics_reqty_bool
\smqty    648 \physics_option_or:nnT { compat } { short }
\pqty     649 {
\bqty     650   \cs_set:Npn \qty { \quantity }
\vqty     651   \physics_declare_legacy_paren:NnnnNNn \pqty { m } {#6} { } { } { } { }
\Bqty     652   \physics_declare_legacy_paren:NnnnNNn \bqty { m } {#6} { } { } [ ] { }
\absolutevalue 653   \physics_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } { } \vert \vert { }
\eval     654   \physics_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } { } \{ \} { }
\abs      655 }
\norm     656 \physics_declare_legacy_paren:NnnnNNn \absolutevalue
\order    657 { m } {#6} { } { } \vert \vert { }
\oorder   658 \physics_option_or:nnT { compat } { short }
\commutator 659 {
\poissonbracket 660   \cs_set:Npn \eval { \evaluated }
\pb       661   \cs_set:Npn \abs { \absolutevalue }
\anticommutator 662 }
\acomm    663 \physics_declare_legacy_paren:NnnnNNn \norm
          664 { m } {#6} { } { } \lVert \rVert { }
          665 \physics_compat:TF
          666 {
          667   \physics_declare_legacy_paren:NnnnNNn \order
          668   { m } {#6} { } { \c_physics_Order_t1 } { } { }
          669 }
          670 {
          671   \physics_declare_legacy_paren:NnnnNNn \order
          672   { m } {#6} { } { \c_physics_order_t1 } { } { }
          673 }
          674 \physics_declare_legacy_paren:NnnnNNn \commutator
          675 { m m } { #6 , #7 } { } { } [ ] { }
          676 \physics_option_or:nnT { compat } { short }
          677 { \cs_set:Npn \comm { \commutator } }
          678 \physics_declare_legacy_paren:NnnnNNn \poissonbracket
          679 { m m } { #6 , #7 } { } { } \{ \} { }
          680 \physics_option_or:nnT { compat } { short }
          681 {
          682   \cs_set:Npn \pb { \poissonbracket }
          683   \cs_set:Npn \anticommutator { \poissonbracket }
          684   \cs_set:Npn \acomm { \poissonbracket }
          685 }
          686 \fi:
          687 \physics_declare_legacy_paren:NnnnNNn \OOrder
          688 { m } {#6} { } { \c_physics_Order_t1 } { } { }
          689 \physics_declare_legacy_paren:NnnnNNn \oorder
          690 { m } {#6} { } { \c_physics_order_t1 } { } { }

```

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

1.3 Matrix things

1.3.1 Matrix auxillary functions

```

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

```

```

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

```

1.3.2 Matrix keys

```

776 \keys_define:nn { physicx }
777 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
778 \keys_define:nn { physicx/matrix }
779 {
780   array .tl_set:N = \l__physicx_matrix_array_tl ,
781   expand .choice: ,
782   expand / none .code:n =
783     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
784   expand / text-expand .code:n =
785     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
786   expand / f .code:n =
787     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
788   expand / romanual .meta:n = { expand = f } ,
789   expand / x .code:n =
790     \cs_set_eq:NN \__physicx_expand:w \use:n ,
791   expand / edef .meta:n = { expand = x } ,
792   rows .int_set:N = \l__physicx_matrix_rows_int ,
793   cols .int_set:N = \l__physicx_matrix_cols_int ,
794   auto-update .choice: ,
795   auto-update / true .code:n =
796     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,

```



```

797 auto-update / false .code:n =
798   \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
799   auto-update .default:n = true ,
800   main .tl_set:N = \l__physicx_matrix_main_tl ,
801   row-list .code:n =
802     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,
803   col-list .code:n =
804     \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
805   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
806   infinite .default:n = true ,
807   !infinite .code:n =
808     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
809   element-code .cs_set:Np = \physicx@matricelement #1#2##3 ,
810   element-code* .choice: ,
811   element-code* / except-empty .code:n =
812     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
813     \physicx@matricelement
814     \cs_set:Npn \physicx@matricelement ##1##2##3
815     {
816       \tl_if_empty:nTF {##1}
817       {##1}
818       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
819     } ,
820   element-code* / except-blank .code:n =
821     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
822     \physicx@matricelement
823     \cs_set:Npn \physicx@matricelement ##1##2##3
824     {
825       \tl_if_blank:nTF {##1}
826       {##1}
827       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
828     } ,
829   element-code* / except-dots .code:n =
830     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
831     \physicx@matricelement
832     \cs_set:Npn \physicx@matricelement ##1##2##3
833     {
834       \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
835       {##1}
836       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
837     } ,
838   element-code* / except-tl .code:n =
839     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
840     \physicx@matricelement
841     \cs_set:Npn \physicx@matricelement ##1##2##3
842     {
843       \tl_if_in:onTF { \physicxexcept } {##1}
844       {##1}
845       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
846     } ,
847   element-code* / except-regex .code:n =
848     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
849     \physicx@matricelement
850     \cs_set:Npn \physicx@matricelement ##1##2##3

```

```

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

```

```

905 item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
906 check-range .choice: ,
907 check-range / true .code:n = \physicx_parse_range_check: ,
908 check-range / false .code:n = \physicx_parse_range_nocheck: ,
909 check-range .default:n = true ,
910 begin .tl_set:N = \__physicx_matrix_begin:w ,
911 end .tl_set:N = \__physicx_matrix_end: ,
912 args .code:n =
913   \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,
914 args* .tl_set:N = \l__physicx_matrix_args_tl ,
915 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
916 after-begin+ .code:n =
917   { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
918 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
919 after-end+ .code:n =
920   { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
921 sepdim .dim_set:N = \l__physicx_matrix_sep_dim ,
922 type .multichoice: ,
923 saveto .tl_set:N = \l__physicx_matrix_save_tl ,
924 saveto* .code:n =
925   \tl_set:Nn \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
926 transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
927 transpose .default:n = true ,
928 ' .meta:n = { transpose = true } ,
929 T .meta:n = { transpose = true } ,
930 MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
931 enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
932 enhanced .default:n = true ,
933 !enhanced .code:n =
934   \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
935 cr .tl_set:N = \physicx@cr ,
936 align .tl_set:N = \physicx@align ,
937 sep .tl_set:N = \physicx@sep ,
938 adi-order .choice: ,
939 adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
940 adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
941 adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
942 adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
943 adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
944 adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
945 beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
946 beginning+ .code:n =
947   \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
948 ending .tl_set:N = \l__physicx_matrix_ending_tl ,
949 ending+ .code:n =
950   \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
951
952 settype .code:n = \setmatrixtype #1 ,
953
954 unknown .code:n =
955   \physicx_search_also:nnF
956   {
957     physicx/matrix/type ,
958     physicx/matrix/expand ,

```

```

959     physicx/matrix/element-code* ,
960   }
961   {#1}
962   {
963     \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
964     {
965       \keys_set:nx { physicx/matrix }
966       { MaxMatrixCols = \l_keys_key_str }
967     }
968     {
969       \msg_error:nxxx { physicx } { unknown-key }
970       \l_keys_path_str { physicx/matrix }
971     }
972   } ,
973 }

```

\physicx_matrix_new_type:nnn

```

974 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
975 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
976 \cs_new:Npn \physicx_matrix_new_type:nn
977 { \physicx_new_type:nnn { matrix } }
978 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
979 {
980   \IfBooleanTF {#1}
981   { \physicx_matrix_new_type:nn {#2} }
982   { \physicx_matrix_new_type:nnn {#2} }
983 }

```

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

A few types.

```

984 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
985 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
986 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
987 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
988 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
989 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
990 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
991 \physicx_mathtools:T
992 {
993   \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
994   \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
995   \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
996   \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
997   \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
998   \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
999   \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
1000   \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
1001   \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
1002   \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
1003   \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
1004   \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
1005   \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
1006   \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}

```

```

1007 \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
1008 \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
1009 \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
1010 }

```

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

```

1011 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
1012 { \clist_set:cn { physicx@ #1 data@ #2 } }
1013 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
1014 {
1015   \clist_clear:c { l__physicx_matrix_ #1 _clist }
1016   \__physicx_matrix_add_data:nn {#1} {#2}
1017 }
1018 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
1019 {
1020   \clist_map_inline:nn {#2}
1021   {
1022     \clist_concat:ccc
1023     { l__physicx_matrix_ #1 _clist }
1024     { l__physicx_matrix_ #1 _clist }
1025     { physicx@ #1 data@ #2 }
1026   }
1027 }

```

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

Initial settings.

```

1028 \keys_set:nn { physicx/matrix }
1029 {
1030   type = m ,
1031   saveto = ? ,
1032 }

```

`\qxmatrix`

```

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

```

```

1053     \physicx_qxmatrix:nnn {#4} {#5} {#7}
1054     \__physicx_matrix_save_or_print:
1055     \group_end:
1056 }
1057 \cs_new_protected:Nn \physicx_qxmatrix:nnn
1058 {
1059     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1060     {
1061         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1062         \__physicx_matrix_appto_body_e:nnn
1063     }
1064     {
1065         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1066         \__physicx_matrix_appto_body_ne:nnn
1067     }
1068     % clear the variable containing the body of the matrix
1069     \tl_clear:N \l__physicx_matrix_body_tl
1070     % set the tentative number of explicit rows
1071     \physicx_if_num:nTF { #2 }
1072     {% number of rows is an integer
1073         \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
1074         {% if #2 <= rows, we don't want a row of dots
1075             \bool_set_false:N \l__physicx_matrix_dotrow_bool
1076             \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1077         }
1078         {% we want a row of dots
1079             \bool_set_true:N \l__physicx_matrix_dotrow_bool
1080         }
1081     }
1082     {% number of rows is symbolic, we want a row of dots
1083         \bool_set_true:N \l__physicx_matrix_dotrow_bool
1084     }
1085     % set the tentative number of explicit columns
1086     \physicx_if_num:nTF { #3 }
1087     {% number of cols is an integer
1088         \int_compare:nTF { #3 <= \l__physicx_matrix_cols_int }
1089         {% if #3 <= cols, we don't want a column of dots
1090             \bool_set_false:N \l__physicx_matrix_dotcol_bool
1091             \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1092         }
1093         {% we want a column of dots
1094             \bool_set_true:N \l__physicx_matrix_dotcol_bool
1095         }
1096     }
1097     {% number of columns is symbolic, we want a column of dots
1098         \bool_set_true:N \l__physicx_matrix_dotcol_bool
1099     }
1100     % loop through the rows
1101     \int_step_inline:nn { \l__physicx_matrix_rows_int }
1102     {
1103         % add the first entry in the row
1104         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1105         \__physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
1106         % add the further entries in the explicit columns

```

```

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

```

```

1161 % if the matrix is infinite, add a final row
1162 \bool_if:NT \l__physicx_matrix_infinite_bool
1163 {
1164   % finish up the row
1165   \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1166   \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1167   \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1168   { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1169   \bool_if:NT \l__physicx_matrix_dotcol_bool
1170   { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1171   \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1172   % update cols
1173   \bool_if:NTF \l__physicx_matrix_dotcol_bool
1174   { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1175   { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1176 }
1177 }

```

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

```

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

```



```

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

```

```

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

```

1315 \cs_new:Npn \physicx_matrix_item_parse:n #1
1316 {
1317   \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1318   \keyval_parse:NNn
1319     \__physicx_matrix_item_parse_aux:n
1320     \__physicx_matrix_item_parse_aux:nn
1321     {#1}
1322 }
1323 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }
1324 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1325 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2
1326 {
1327   \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1328   \tl_set:Nx \l__physicx_tmpitem_tl
1329     { \__physicx_expand:w \l__physicx_tmpitem_tl }
1330   \physicx_parse_range:nxN \l__physicx_matrix_rows_int
1331     { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1332   \physicx_parse_range:nxN \l__physicx_matrix_cols_int
1333     { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1334   \exp_args:No \tl_if_eq:nnTF
1335     { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1336   {
1337     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1338       {
1339         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1340           {
1341             \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1342           }
1343       }
1344   }
1345   {
1346     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1347       {
1348         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1349           {
1350             \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }
1351             {
1352               \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1353                 {##1} {####1} { \l__physicx_tmpitem_tl }
1354             }
1355           }
1356       }
1357   }
1358 }

```

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

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

```

\physicx_matrix_array_parse:o 1359 \cs_new:Npn \physicx_matrix_array_parse:n #1
1360 {
1361   \tl_set:Nn \l__physicx_tmparr_tl {#1}
1362   \tl_set:Nx \l__physicx_tmparr_tl
1363     { \__physicx_expand:w \l__physicx_tmparr_tl }
1364   \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl

```

```

1365 \__physicx_matrix_autocalc:nn
1366 { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1367 { 0 }
1368 \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1369 {
1370   \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1371   \__physicx_matrix_autocalc:nn
1372   { 0 }
1373   { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1374   \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1375   {
1376     \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1377   }
1378 }
1379 }
1380 \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.

```

1381 \cs_new:Npn \physicx_matrix_array_parse_main:
1382 {
1383   \int_step_inline:nn \l__physicx_matrix_rows_int
1384   {
1385     \int_step_inline:nn \l__physicx_matrix_cols_int
1386     {
1387       \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1388       {##1} {####1} \l__physicx_matrix_main_tl
1389     }
1390   }
1391 }

```

(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).

```

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

```

```

1410         } { }
1411     }
1412     { \prg_return_true: }
1413     { \prg_return_false: }
1414 }
1415 }

```

(End definition for `__physicx_if_can_num:n`.)

`\diagonalmatrix` Define `\diagonalmatrix`.

```

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

```

```

1460 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1461 {
1462   \int_step_inline:nn { #1 - 1 }
1463   {
1464     \int_step_inline:nn { #2 - 1 }
1465     {
1466       \tl_put_right:Nx \l__physicx_matrix_body_tl
1467       {
1468         \exp_after:wN
1469         \physicx_matrix_use_r_c:nn
1470         #3 {{##1}} {{####1}} &
1471       }
1472     }
1473     \tl_put_right:Nx \l__physicx_matrix_body_tl
1474     {
1475       \exp_after:wN
1476       \physicx_matrix_use_r_c:nn
1477       #3 {{##1}} {{ \int_use:N #2 }} \[\dim_use:N \l__physicx_matrix_sep_dim]
1478     }
1479   }
1480   \int_step_inline:nn { #2 - 1 }
1481   {
1482     \tl_put_right:Nx \l__physicx_matrix_body_tl
1483     {
1484       \exp_after:wN
1485       \physicx_matrix_use_r_c:nn
1486       #3 {{ \int_use:N #1 }} {{##1}} &
1487     }
1488   }
1489   \tl_put_right:Nx \l__physicx_matrix_body_tl
1490   {
1491     \exp_after:wN
1492     \physicx_matrix_use_r_c:nn
1493     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1494   }
1495 }

```

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

__physicx_declare_init:

```

1496 \cs_new:Npn \__physicx_matrix_enhanced_init:
1497 {
1498   \seq_if_empty:NF \l__physicx_row_list_seq
1499   {
1500     \bool_set_true:N \l__physicx_matrix_expand_element_bool
1501     \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1502     { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1503   }
1504   \seq_if_empty:NF \l__physicx_col_list_seq
1505   {
1506     \bool_set_true:N \l__physicx_matrix_expand_element_bool
1507     \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1508     { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1509   }
1510 }

```

(End definition for _physicx_declare_init:.)

\commamatrix Define \commamatrix.

```

1511 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1512 {
1513   \group_begin:
1514   \keys_set:nn { physicx/matrix } {#3}
1515   \tl_if_empty:nF {#4}
1516   { \keys_set:nn { physicx/matrix } { array = {#4} } }
1517   \IfBooleanT {#1}
1518   { \keys_set:nn { physicx/matrix } { saveto = \physicx_tmp } }
1519   \tl_set:Nx \l__physicx_matrix_array_tl
1520   { \_physicx_expand:w \l__physicx_matrix_array_tl }
1521   \bool_lazy_or:nnTF
1522   { \bool_if_p:n {#2} }
1523   { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1524   { \_physicx_commamatrix_enhanced: }
1525   {
1526     \tl_replace_all:Nox \l__physicx_matrix_array_tl
1527     { \physicx@cr } { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1528     \tl_replace_all:Non \l__physicx_matrix_array_tl
1529     { \physicx@align } { & }
1530     \tl_set_eq:NN \l__physicx_matrix_body_tl
1531     \l__physicx_matrix_array_tl
1532   }
1533   \_physicx_matrix_save_or_print:
1534   \group_end:
1535 }
1536 \cs_new_nopar:Npn \_physicx_matrix_save_or_print:
1537 {
1538   \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1539   {
1540     \exp_after:wN \tl_gset_eq:NN
1541     \l__physicx_matrix_save_tl
1542     \l__physicx_matrix_body_tl
1543   }
1544   {
1545     \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1546     \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1547     \fi:
1548     \exp_after:wN \_physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matrix_body_tl
1549     \l__physicx_matrix_body_tl
1550     \_physicx_matrix_end: \l__physicx_matrix_after_end_tl
1551   }
1552 }
1553 \cs_new:Npn \_physicx_commamatrix_enhanced:
1554 {
1555   \tl_clear:N \l__physicx_matrix_body_tl
1556   \int_zero:N \l__physicx_tmpa_int
1557   \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1558   \l__physicx_matrix_array_tl
1559   \int_set:Nn \l__physicx_matrix_rows_int
1560   { \seq_count:N \l__physicx_tmp_seq }
1561   \_physicx_matrix_enhanced_init:

```

```

1562 \bool_if:NTF \l__physicx_matrix_expand_element_bool
1563 {
1564   \seq_map_tokens:Nn \l__physicx_tmp_seq
1565   {
1566     \int_incr:N \l__physicx_tmpa_int
1567     \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1568     \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_e:nnn
1569   }
1570 }
1571 {
1572   \seq_map_tokens:Nn \l__physicx_tmp_seq
1573   {
1574     \int_incr:N \l__physicx_tmpa_int
1575     \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1576     \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_ne:nnn
1577   }
1578 }
1579 }
1580 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1581 {
1582   \seq_set_split:Non \l__physicx_tmp_col_seq
1583   { \physicx@align } {#3}
1584   \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1585   \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1586   { #2 {##2} {#1} {##1} }
1587   \tl_put_right:Nx \l__physicx_matrix_body_tl
1588   {
1589     \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1590     \if_int_compare:w \l__physicx_matrix_rows_int = #1
1591     \scan_stop:
1592     \else:
1593     \[\dim_use:N \l__physicx_matrix_sep_dim]
1594     \fi:
1595   }
1596 }
1597 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1598 {
1599   \seq_put_right:Nx \l__physicx_tmp_coled_seq
1600   {
1601     \text_expand:n % \text_expand:n do the magic thing, but slower
1602     {
1603       \physicx@matricelement { #1 }
1604       { \__physicx_matrix_row_iterate:n {#2} }
1605       { \__physicx_matrix_col_iterate:n {#3} }
1606     }
1607   }
1608 }
1609 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1610 {
1611   \seq_put_right:No \l__physicx_tmp_coled_seq
1612   {
1613     \physicx@matricelement {#1}
1614     { \__physicx_matrix_row_iterate:n {#2} }
1615     { \__physicx_matrix_col_iterate:n {#3} }

```



```

1616     }
1617 }

```

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

\generalmatrix Define \generalmatrix.

```

1618 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1619 {
1620   \IfBooleanTF {#2}
1621   {
1622     \group_begin:
1623     \IfBooleanTF {#1}
1624     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicxtmp } }
1625     { \keys_set:nn { physicx/matrix } {#4} }
1626     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1627     \physicx_construct:nnn
1628     {
1629       \tl_if_empty:NTF \l__physicx_matrix_main_tl
1630       {
1631         \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1632       }
1633       { \physicx_matrix_array_parse_main: }
1634     }
1635     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1636     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1637     \__physicx_generalmatrix:
1638     \__physicx_matrix_save_or_print:
1639     \group_end:
1640   }
1641   {
1642     \IfBooleanTF {#1}
1643     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1644     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1645     \qxmatrix = * [#4]
1646   }
1647 }
1648 \cs_new:Npn \__physicx_generalmatrix:
1649 {
1650   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1651   {
1652     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1653     \__physicx_matrix_appto_body_e:off
1654   }
1655   {
1656     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1657     \__physicx_matrix_appto_body_ne:off
1658   }
1659   \__physicx_matrix_transpose:N
1660   \__physicx_matrix_generate_body:NNNN
1661   \__physicx_generalmatrix_generate:nnn
1662 }

```

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

_physicx_matrix_generate_body:NNNN

```

1663 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1664 \cs_new:Npn \_physicx_matrix_generate_body:NNNN #1#2#3#4
1665 {
1666   \_physicx_matrix_enhanced_init:
1667   \int_step_inline:nn { #1 - 1 }
1668   {
1669     \int_step_inline:nn { #2 - 1 }
1670     {
1671       \tl_set:Nx \l__physicx_tmp_tl
1672       {
1673         \exp_after:wN
1674         \physicx_matrix_use_r_c:nn
1675         #3 {{##1}} {{####1}}
1676       }
1677       #4 \l__physicx_tmp_tl {##1} {####1}
1678       \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1679     }
1680     \tl_set:Nx \l__physicx_tmp_tl
1681     {
1682       \exp_after:wN
1683       \physicx_matrix_use_r_c:nn
1684       #3 {{##1}} {{ \int_use:N #2 }}
1685     }
1686     #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1687     \tl_put_right:Nx \l__physicx_matrix_body_tl
1688     { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1689   }
1690   \int_step_inline:nn { #2 - 1 }
1691   {
1692     \tl_set:Nx \l__physicx_tmp_tl
1693     {
1694       \exp_after:wN
1695       \physicx_matrix_use_r_c:nn
1696       #3 {{ \int_use:N #1 }} {{##1}}
1697     }
1698     #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}
1699     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1700   }
1701   \tl_set:Nx \l__physicx_tmp_tl
1702   {
1703     \exp_after:wN
1704     \physicx_matrix_use_r_c:nn
1705     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1706   }
1707   #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1708 }

```

(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

```

1709 \cs_new:Npn \_physicx_matrix_appto_body_e:nnn #1#2#3
1710 {
1711   \tl_put_right:Nx \l__physicx_matrix_body_tl

```

```

1712     {
1713         \text_expand:n
1714         {
1715             \physicx@matricelement {#1}
1716             { \__physicx_matrix_row_iterate:n {#2} }
1717             { \__physicx_matrix_col_iterate:n {#3} }
1718         }
1719     }
1720 }
1721 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1722 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1723 {
1724     \tl_put_right:No \l__physicx_matrix_body_tl
1725     {
1726         \physicx@matricelement {#1}
1727         { \__physicx_matrix_row_iterate:n {#2} }
1728         { \__physicx_matrix_col_iterate:n {#3} }
1729     }
1730 }
1731 \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

```

1732 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1733 {
1734     \bool_if:NTF \l__physicx_matrix_transpose_bool
1735     {
1736         #1
1737         \l__physicx_matrix_cols_int
1738         \l__physicx_matrix_rows_int
1739         \use_ii_i:nn
1740     }
1741     {
1742         #1
1743         \l__physicx_matrix_rows_int
1744         \l__physicx_matrix_cols_int
1745         \use:nn
1746     }
1747 }

```

(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.

```

1748 \cs_new:Npn \physicx_construct:nnn #1#2#3
1749 {
1750     \l__physicx_matrix_beginning_tl
1751     \__physicx_adi:nnn {#1} {#2} {#3}
1752     \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1753     {
1754         \int_incr:N \l__physicx_matrix_cols_int
1755         \__physicx_matrix_last_aux_c:
1756         \int_incr:N \l__physicx_matrix_cols_int

```

```

1757     }
1758     \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1759     {
1760         \int_incr:N \l__physicx_matrix_rows_int
1761         \__physicx_matrix_last_aux_r:
1762         \int_incr:N \l__physicx_matrix_rows_int
1763     }
1764     \bool_lazy_or:nnF
1765     { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1766     { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1767     {
1768         \physicx_matrix_set_r_c:nnn
1769         { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1770         { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1771         { \ddots }
1772     }
1773     \bool_if:NT \l__physicx_matrix_infinite_bool
1774     {
1775         \int_incr:N \l__physicx_matrix_rows_int
1776         \int_incr:N \l__physicx_matrix_cols_int
1777         \__physicx_matrix_last_aux_c:
1778         \__physicx_matrix_last_aux_r:
1779         \physicx_matrix_set_r_c:nnn
1780         { \int_use:N \l__physicx_matrix_rows_int }
1781         { \int_use:N \l__physicx_matrix_cols_int }
1782         { \ddots }
1783     }
1784     \l__physicx_matrix_ending_tl
1785 }
1786 \cs_new:Npn \__physicx_matrix_last_aux_c:
1787 {
1788     \int_step_inline:nn \l__physicx_matrix_rows_int
1789     {
1790         \physicx_matrix_set_r_c:nnn
1791         {##1} { \int_use:N \l__physicx_matrix_cols_int }
1792         { \cdots }
1793     }
1794 }
1795 \cs_new:Npn \__physicx_matrix_last_aux_r:
1796 {
1797     \int_step_inline:nn \l__physicx_matrix_cols_int
1798     {
1799         \physicx_matrix_set_r_c:nnn
1800         { \int_use:N \l__physicx_matrix_rows_int } {##1}
1801         { \vdots }
1802     }
1803 }

```

(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
\NewGeneralMatrix
\newdiagonalmatrix
\NewDiagonalMatrix
\newcommamatrix
\NewCommaMatrix
1804 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3

```

```

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

```


C

\cdots 834, 1113, 1116, 1117, 1120,
 1122, 1153, 1154, 1157, 1159, 1792
 clist commands:
 \backslash clist_clear:N 1015
 \backslash clist_concat:NNN 1022
 \backslash clist_if_empty:nTF 222
 \backslash clist_if_in:NnTF 1350
 \backslash clist_map_break:n 176, 188
 \backslash clist_map_inline:nn 82, 172, 184, 1020
 \backslash clist_new:N 50, 746, 747
 \backslash clist_put_right:Nn . . . 896, 902, 1341
 \backslash clist_set:Nn 1012
 \backslash clist_set_eq:NN 1317
 \backslash c_empty_clist 1317
 \backslash comm 677
 \backslash commamatrix 31, 1511, 1836
 \backslash commutator 647
 \backslash crossproduct 271
 cs commands:
 \backslash cs:w 32, 33, 925
 \backslash cs_end: 32, 33, 701, 925
 \backslash cs_generate_variant:Nn
 3, 4, 5, 6, 89, 92, 198,
 417, 557, 1185, 1323, 1380, 1721, 1731
 \backslash cs_gset_eq:NN 40, 41, 42
 \backslash cs_if_exist:NnTF 323
 \backslash cs_if_exist_p:N 1403, 1408
 \backslash cs_if_exist_use:NnTF 37
 \backslash cs_new:Npn . . . 7, 124, 132, 135, 147,
 158, 170, 204, 208, 212, 216, 217,
 218, 219, 220, 384, 396, 467, 477,
 489, 526, 533, 558, 627, 773, 974,
 976, 1178, 1186, 1214, 1227, 1238,
 1248, 1274, 1313, 1315, 1324, 1325,
 1359, 1381, 1455, 1460, 1496, 1553,
 1580, 1597, 1609, 1648, 1664, 1709,
 1722, 1732, 1748, 1786, 1795, 1804
 \backslash cs_new_eq:NN
 134, 157, 732, 752, 761, 1246
 \backslash cs_new_nopar:Npn 691, 699, 708, 710,
 715, 721, 754, 755, 758, 759, 760, 1536
 \backslash cs_new_protected:Nn 1057
 \backslash cs_new_protected:Npn
 67, 72, 77, 90, 93, 200
 \backslash cs_new_protected_nopar:Npn
 1011, 1013, 1018
 \backslash cs_set:Nn . . . 939, 940, 941, 942, 943, 944
 \backslash cs_set:Npn 422,
 428, 432, 650, 660, 661, 677, 682,
 683, 684, 814, 823, 832, 841, 850, 859
 \backslash cs_set:Npx 866
 \backslash cs_set_eq:NN 69, 70, 74, 75,
 206, 210, 214, 273, 274, 730, 783,

785, 787, 790, 796, 798, 812, 821,
 830, 839, 848, 857, 875, 878, 881,
 884, 887, 1061, 1065, 1192, 1197,
 1202, 1208, 1439, 1443, 1652, 1656
 \backslash cs_set_nopar:Npn 1501, 1507
 \backslash cs_set_protected:Npn
 538, 551, 570, 645
 \backslash cs_to_str:N . . . 437, 440, 445, 446, 483

D

\ddots 834, 1140, 1171, 1771, 1782
 \backslash DeclareDocumentCommand 263,
 265, 267, 270, 271, 272, 278, 279,
 543, 553, 629, 1042, 1416, 1511, 1618
 \backslash diagonalmatrix 29, 1416, 1835
 dim commands:
 \backslash dim_new:N 772
 \backslash dim_use:N 1127, 1134,
 1141, 1165, 1477, 1527, 1593, 1688
 \backslash div 273, 274
 \backslash divergence 274
 \backslash divisionsymbol 273
 \backslash dotproduct 270

E

\backslash egroup 217, 219
 else commands:
 \backslash else:
 703, 1125, 1252, 1262, 1287, 1299, 1592
 \backslash end 351, 352, 353, 354, 355, 356,
 357, 360, 361, 362, 363, 364, 365,
 366, 367, 368, 369, 370, 371, 372,
 373, 374, 375, 376, 596, 599, 603,
 606, 607, 608, 613, 614, 618, 621,
 622, 623, 984, 985, 986, 987, 988,
 989, 990, 993, 994, 995, 996, 997,
 998, 999, 1000, 1001, 1002, 1003,
 1004, 1005, 1006, 1007, 1008, 1009
 \backslash eval 647
 \backslash evaluated 572, 660
 exp commands:
 \backslash exp_after:wN
 1468, 1475, 1484, 1491, 1538,
 1540, 1548, 1673, 1682, 1694, 1703
 \backslash exp_args:Nc 436, 439
 \backslash exp_args:Nf 1222
 \backslash exp_args:Nnno 1352, 1387
 \backslash exp_args:Nno 174, 186
 \backslash exp_args:NNx 442
 \backslash exp_args:No 852, 861, 963, 1334
 \backslash exp_args:NV 1567, 1575
 \backslash exp_end: 8
 \backslash exp_not:N 8, 326, 444, 445, 446, 497, 866

<code>\exp_not:n</code>	501, 502, 702, 704, 752, 783, 787
<code>\exp_stop_f:</code>	492, 499, 1250, 1253, 1276, 1288
F	
<code>\fbraces</code>	278, 279
fi commands:	
<code>\fi:</code>	503, 518, 523, 524, 625, 686, 705, 1128, 1271, 1272, 1310, 1311, 1547, 1594
fp commands:	
<code>\fp_eval:n</code>	28, 1400
<code>\fpeval</code>	28, 1408, 1409
G	
<code>\generalmatrix</code> ..	33, 1618, 1842, 1845, 1846, 1851, 1854, 1855, 1862, 1863
group commands:	
<code>\group_begin:</code>	386, 1044, 1418, 1513, 1622
<code>\group_end:</code> ..	394, 1055, 1453, 1534, 1639
H	
<code>\hat</code>	268
hook commands:	
<code>\hook_gput_code:nnn</code>	37
I	
if commands:	
<code>\if_bool:N</code>	507, 519, 572, 647
<code>\if_case:w</code>	499
<code>\if_cs_exist:w</code>	701
<code>\if_int_compare:w</code>	492, 1123, 1250, 1253, 1276, 1288, 1545, 1590
<code>\IfBooleanT</code>	595, 597, 1517
<code>\IfBooleanTF</code> ..	264, 266, 268, 285, 453, 545, 602, 604, 617, 619, 639, 980, 1045, 1050, 1419, 1620, 1623, 1642, 1643, 1644, 1808, 1830, 1839, 1861
<code>\IfNoValueTF</code>	278, 279, 420, 426, 1810, 1813, 1819, 1822, 1841, 1844, 1850, 1853
<code>\Im</code>	279
<code>\imaginary</code>	279
int commands:	
<code>\int_compare:nNnTF</code>	105, 115, 149
<code>\int_compare:nTF</code>	1073, 1088
<code>\int_compare_p:nNn</code> ..	127, 128, 138, 139
<code>\int_eval:n</code> ..	28, 1257, 1266, 1284, 1296, 1306, 1307, 1399, 1769, 1770
<code>\int_incr:N</code> ..	491, 1232, 1566, 1574, 1754, 1756, 1760, 1762, 1775, 1776
<code>\int_max:nn</code>	694, 696
<code>\int_new:N</code>	52, 53, 62, 63, 64, 65, 466, 743, 744
<code>\int_set:Nn</code> ..	80, 81, 104, 114, 472, 693, 695, 1048, 1049, 1076, 1091, 1559
<code>\int_set_eq:NN</code>	102, 107, 112, 117, 1235, 1546
<code>\int_step_inline:nn</code>	1101, 1383, 1385, 1462, 1464, 1480, 1667, 1669, 1690, 1788, 1797
<code>\int_step_inline:nnn</code>	142, 152, 1107, 1145
<code>\int_use:N</code>	1477, 1486, 1493, 1684, 1686, 1696, 1698, 1705, 1707, 1780, 1781, 1791, 1800
<code>\int_zero:N</code>	1229, 1556
<code>\interval</code>	28, 1403, 1404
K	
keys commands:	
<code>\keys_define:nn</code>	201, 225, 297, 299, 776, 778
<code>\keys_if_exist:nnTF</code>	174, 186
<code>\l_keys_key_str</code>	174, 177, 186, 191, 319, 323, 326, 963, 966
<code>\l_keys_path_str</code>	339, 970
<code>\keys_set:nn</code>	3, 177, 191, 286, 287, 298, 325, 378, 387, 777, 965, 1028, 1051, 1052, 1420, 1421, 1514, 1516, 1518, 1624, 1625
keyval commands:	
<code>\keyval_parse:NNn</code>	1318
<code>\keyval_parse:nnn</code>	1180
L	
<code>\langle</code>	350, 580, 607, 622
<code>\lbrace</code>	278, 279
<code>\ldots</code>	834
<code>\left</code>	216, 380, 405, 595, 602, 607, 613, 614, 617, 621, 622, 623
<code>\let</code>	278, 279, 540
<code>\lggroup</code>	602, 617
<code>\lVert</code>	664
M	
<code>\mathbf</code>	264, 266, 268
<code>\mathcal</code>	202, 203
<code>\mathclose</code>	216, 218
<code>\mathopen</code>	216, 218
<code>\matrixquantity</code>	572
<code>\mqty</code>	647
msg commands:	
<code>\msg_error:nnn</code>	1212
<code>\msg_error:nnnn</code>	338, 969
<code>\msg_new:nnn</code>	60

\msg_new:nnnn	54	\physicx_mathtools:	24
		\physicx_mathtools:TF	358, 991
		\physicx_matrix_array_parse:n	1359, 1631
		\physicx_matrix_array_parse_-	1381, 1633
		\physicx_matrix_diag_parse:n	897, 1178, 1424, 1428, 1429, 1635
		\physicx_matrix_item_parse:n	903, 1315, 1432, 1636
		\physicx_matrix_new_type:nn	974
		\physicx_matrix_new_type:nnn	974
		\physicx_matrix_set_r_c:nnn	732, 875, 878, 881, 884, 887, 1233, 1241, 1256, 1265, 1282, 1294, 1305, 1352, 1376, 1387, 1768, 1779, 1790, 1799
		\physicx_matrix_use_r_c:nn	699, 1469, 1476, 1485, 1492, 1674, 1683, 1695, 1704
		\physicx_new_type:nnn	200, 344, 975, 977
		\physicx_option_or:nn	29
		\physicx_option_or:nnTF	648, 658, 676, 680
		\c_physicx_Order_tl	203, 255, 668, 688
		\c_physicx_order_tl	202, 254, 672, 690
		\physicx_parse_range:nnN	90, 92, 1330, 1332
		\physicx_parse_range:nnnN	62
		\physicx_parse_range_check:	62, 907
		\physicx_parse_range_nocheck:	62, 908
		\physicx_qxmatrix:nnn	1053, 1057
		\physicx_right:	217, 407, 541, 641
		\physicx_right:N	219, 411, 633, 634, 635, 636
		\physicx_search_also:nn	170, 181, 198, 199
		\physicx_search_also:nnTF	332, 955
		\physicx_short:	19
		\physicx_unimath:	44
		\physicx_unimath:TF	40, 41, 42, 261
		\physicx_use_amssymb_type:	204, 260
		\physicx_use_uni_bf_type:	212
		\physicx_use_uni_bfit_type:	208, 259
		\physicx_xquantity:nn	384
		physicx internal commands:	
		__physicx_adi:nnn	773, 939, 940, 941, 942, 943, 944, 1751
		\l__physicx_begin_int	62, 102, 104, 105, 107, 138, 139, 143, 149, 153
		\l__physicx_cmd_arg_int	466, 472, 491, 492, 499
		\l__physicx_cmd_arg_spec_tl	465, 471, 484, 493
N			
\nabla	272		
\NewCommaMatrix	1804		
\newcommamatrix	1804		
\newcommand	423, 429, 433, 442, 1811, 1814, 1815, 1820, 1823, 1824, 1842, 1845, 1846, 1851, 1854, 1855		
\NewDiagonalMatrix	1804		
\newdiagonalmatrix	1804		
\NewDocumentCommand	283, 343, 416, 418, 449, 451, 978, 1806, 1828, 1831, 1832, 1837, 1859, 1862, 1863		
\NewGeneralMatrix	1804		
\newgeneralmatrix	1804		
\NewXQuantity	384		
\newxquantity	384		
\nobreak	586, 587, 588		
\norm	647		
O			
\OOrder	687		
\oorder	647		
\operatorname	278, 279		
or commands:			
\or:	500, 501, 502		
\order	647		
\Ordersymbol	255		
\ordersymbol	254		
P			
\PassOptionsToPackage	222		
\pb	647		
peek commands:			
\peek_charcode_ignore_spaces:NTF	540		
physicx commands:			
\physicx_bf:	206, 210, 214, 264, 266, 268		
\physicx_compat:	14		
\physicx_compat:TF	252, 276, 665		
\physicx_construct:nnn	1422, 1627, 1748		
\physicx_declare_legacy_paren:NnnnNn	626, 651, 652, 653, 654, 656, 663, 667, 671, 674, 678, 687, 689		
\physicx_declare_legacy_quantity:nnNn	460, 573, 583, 590, 610		
\physicx_if_num:n	160		
\physicx_if_num:nTF	963, 1071, 1086, 1394		
\physicx_if_num_sign:n	165		
\physicx_left:	216, 407, 541, 641		
\physicx_left:N	218, 409, 633, 634, 635, 636		

```

\l__physicx_cmd_auto_body_bool ..
..... 464, 474, 519, 562, 566
\l__physicx_cmd_auto_body_tl ...
..... 463, 470, 486, 520, 521
\l__physicx_cmd_noauto_body_bool
..... 462, 473, 507, 560
\l__physicx_cmd_noauto_body_tl ..
..... 461, 469, 485, 508, 509
\l__physicx_col_list_seq .....
..... 750, 804, 1504, 1508
\__physicx_commamatrix_enhanced:
..... 1524, 1553
\__physicx_commamatrix_enhanced-
aux:nNn ..... 1567, 1575, 1580
\__physicx_commamatrix_enhanced-
aux_e:nnn ..... 1568, 1597
\__physicx_commamatrix_enhanced-
aux_ne:nnn ..... 1576, 1609
\g__physicx_compat_bool .....
..... 11, 16, 227, 246
\__physicx_declare_init: ..... 1496
\__physicx_declare_init:nnn 467, 479
\__physicx_declare_legacy_-
quantity_aux:nnnn ..... 489, 530
\__physicx_declare_legacy_-
quantity_aux:NNnnn .. 482, 533, 557
\__physicx_declare_legacy_-
quantity_aux:nw .... 480, 526, 531
\__physicx_diagonalmatrix_diag_-
main: ..... 1202, 1208, 1246, 1251
\__physicx_diagonalmatrix_-
enhanced:nnn .... 1439, 1443, 1458
\__physicx_diagonalmatrix_-
generate_body:NNN .... 1451, 1460
\__physicx_diagonalmatrix_-
generate_enhanced_body:NNN ..
..... 1450, 1455
\__physicx_diagonalmatrix_no_-
diag: ..... 1209, 1238, 1247
\__physicx_diagonalmatrix_set_-
diag: ..... 1203, 1227
\l__physicx_end_int ..... 63,
112, 114, 115, 117, 139, 143, 149, 153
\__physicx_expand:w .... 752, 783,
785, 787, 790, 1218, 1329, 1363, 1520
\__physicx_generalmatrix: 1637, 1648
\__physicx_generalmatrix_-
generate:nnn .... 1652, 1656, 1661
\__physicx_if_can_num:n ..... 1392
\__physicx_if_keyval:nTF .. 158, 1427
\l__physicx_invalid_range_bool ..
..... 66, 85, 95, 129, 140, 150
\l__physicx_item_ignore_clist ...
..... 1317, 1341, 1350

\__physicx_loadpackage_options:nnn
..... 220, 231, 233, 235
\g__physicx_mathtools_bool .....
..... 9, 26, 249, 250
\__physicx_matrix_add_data:nn ...
..... 899, 905, 1016, 1018
\l__physicx_matrix_after_begin_-
tl ..... 768, 915, 917, 1548
\l__physicx_matrix_after_end_tl .
..... 769, 918, 920, 1550
\__physicx_matrix_appto_body_-
e:nnn ..... 1062, 1440, 1653, 1709
\__physicx_matrix_appto_body_-
ne:nnn ..... 1066, 1444, 1657, 1709
\l__physicx_matrix_args_tl .....
..... 767, 913, 914, 1548
\l__physicx_matrix_array_tl ....
..... 741, 780, 1519,
1520, 1526, 1528, 1531, 1558, 1631
\__physicx_matrix_autocalc:nn ...
..... 761, 796, 798, 1314, 1365, 1371
\__physicx_matrix_begin:w .....
..... 759, 910, 1548
\l__physicx_matrix_beginning_tl .
..... 774, 945, 947, 1750
\l__physicx_matrix_body_tl .....
..... 742, 1069, 1104,
1109, 1110, 1116, 1117, 1122, 1127,
1134, 1136, 1138, 1140, 1141, 1143,
1147, 1148, 1153, 1154, 1159, 1165,
1166, 1168, 1170, 1171, 1466, 1473,
1482, 1489, 1530, 1542, 1549, 1555,
1587, 1678, 1687, 1699, 1711, 1724
\__physicx_matrix_calc:nn .....
..... 691, 796, 1193
\__physicx_matrix_col_iterate:n .
..... 758, 891, 1507, 1605, 1615, 1717, 1728
\l__physicx_matrix_cols_int ....
..... 695, 696,
744, 793, 1049, 1088, 1091, 1107,
1137, 1145, 1167, 1174, 1175, 1229,
1232, 1236, 1284, 1296, 1307, 1332,
1385, 1545, 1546, 1737, 1744, 1754,
1756, 1770, 1776, 1781, 1791, 1797
\l__physicx_matrix_diag_bool ...
..... 748, 1201, 1207
\__physicx_matrix_diag_calc:nn ..
..... 1192, 1197, 1242,
1259, 1268, 1277, 1289, 1300, 1313
\l__physicx_matrix_diag_clist ...
..... 746, 894, 896, 1424, 1635
\__physicx_matrix_diag_parse_-
aux:n ..... 1181, 1186

```

__physicx_matrix_diag_parse_-	__physicx_matrix_save_or_print:
aux:nn 1182, 1214 1054, 1452, 1533, 1536, 1638
__physicx_matrix_diag_parse_-	\l__physicx_matrix_save_tl
aux_anti:n 1222, 1274 923, 925, 1538, 1541
__physicx_matrix_diag_parse_-	\l__physicx_matrix_sep_dim
aux_regu:n 1225, 1248 772, 921, 1127, 1134,
\l__physicx_matrix_dotcol_bool ..	1141, 1165, 1477, 1527, 1593, 1688
..... 740, 1090, 1094,	__physicx_matrix_set_data:nn ...
1098, 1114, 1139, 1151, 1169, 1173 898, 904, 1013
\l__physicx_matrix_dotrow_bool ..	__physicx_matrix_set_r_c_-
..... 739, 1075, 1079, 1083, 1131	ckall:nnn 730, 888
__physicx_matrix_element_-	__physicx_matrix_set_r_c_-
aux:nnn 812, 818, 821, 827,	ckep:nnn 715, 879
830, 836, 839, 845, 848, 854, 857, 862	__physicx_matrix_set_r_c_-
__physicx_matrix_end: 911, 1550	ckig:nnn 710, 882
__physicx_matrix_end:w 760	__physicx_matrix_set_r_c_-
\l__physicx_matrix_ending_tl ...	ckigep:nnn 721, 731, 885
..... 775, 948, 950, 1784	__physicx_matrix_set_r_c_-
\l__physicx_matrix_enhanced_bool	nock:nnn 708, 733, 876
..... 771, 931, 934, 1435, 1523	\l__physicx_matrix_tmparr_c_sep ..
__physicx_matrix_enhanced_init: 1370, 1373, 1374
..... 1496, 1561, 1666	\l__physicx_matrix_tmparr_r_sep ..
\l__physicx_matrix_expand_- 1364, 1366, 1368
element_bool 762,	__physicx_matrix_transpose:N ...
870, 1059, 1437, 1500, 1506, 1562, 1650 1449, 1659, 1732
__physicx_matrix_generate_-	\l__physicx_matrix_transpose_-
body:NNNN 1457, 1660, 1663	bool 770, 926, 1734
\l__physicx_matrix_infinite_bool	\l__physicx_max_int
..... 738, 805, 808, 1046, 64, 81, 112, 115, 117, 127, 138
1047, 1121, 1158, 1162, 1626, 1773	\l__physicx_min_int
\l__physicx_matrix_item_clist 65, 80, 102, 105, 107, 128
..... 747, 900, 902, 1432, 1636	__physicx_nauto_case:nnnn . 535, 558
__physicx_matrix_item_parse_-	__physicx_new_matrix_cmd:NNN . 1804
aux:n 1319, 1324	__physicx_new_xquantity_aux:w ..
__physicx_matrix_item_parse_- 422, 428, 432, 436, 439
aux:nn 1320, 1325	__physicx_parse_range_aux:n . 84, 93
__physicx_matrix_last_aux_c: ...	__physicx_parse_range_range: ...
..... 1755, 1777, 1786 70, 75, 120, 157
__physicx_matrix_last_aux_r: ...	__physicx_parse_range_range_-
..... 1761, 1778, 1795	check: 70, 135, 157
\l__physicx_matrix_last_col_tl ..	__physicx_parse_range_range_-
..... 757, 893, 1752, 1766	nocheck: 75, 147
\l__physicx_matrix_last_row_tl ..	__physicx_parse_range_single:n .
..... 756, 892, 1758, 1765 69, 74, 122, 134
\l__physicx_matrix_main_tl	__physicx_parse_range_single_-
..... 745, 800, 1388, 1629	check:n 69, 124, 134
__physicx_matrix_row_iterate:n .	__physicx_parse_range_single_-
755, 890, 1501, 1604, 1614, 1716, 1727	nocheck:n 74, 132
\l__physicx_matrix_rows_int	\g__physicx_physics_bool 10
..... 693, 694, 743, 792, 1048,	\l__physicx_quantity_args_tl ...
1073, 1076, 1101, 1123, 1235, 1330, 289, 311, 312, 391
1383, 1559, 1590, 1738, 1743, 1760,	\l__physicx_quantity_code_tl ...
1762, 1769, 1775, 1780, 1788, 1800 290, 313, 388, 392

\l__physicx_quantity_left_size_- tl	291, 305, 400, 405, 409
\l__physicx_quantity_left_tl	292, 303, 390
\l__physicx_quantity_post_tl	293, 302, 414
\l__physicx_quantity_pre_tl	294, 301, 398
\l__physicx_quantity_right_size_- tl	295, 306, 401, 406, 411
\l__physicx_quantity_right_tl	296, 304, 393
__physicx_qxmatrix_appto_- body:nnn	1061, 1065, 1105, 1111, 1118, 1144, 1149, 1155
\g__physicx_reqty_bool	13, 237, 572, 647
\l__physicx_row_list_seq	749, 802, 1498, 1502
\g__physicx_short_bool . . .	12, 21, 229
\l__physicx_tmp_col_seq . .	1582, 1585
\l__physicx_tmp_coled_seq	1584, 1589, 1599, 1611
\l__physicx_tmp_colnum_seq	1333, 1339, 1348
\l__physicx_tmp_rownum_seq	1331, 1337, 1346
\l__physicx_tmp_seq	1557, 1560, 1564, 1572
\l__physicx_tmp_tl	494, 508, 520, 1671, 1677, 1680, 1686, 1692, 1698, 1701, 1707
\l__physicx_tmpa_bool	51, 183, 190, 195
\l__physicx_tmpa_clist	50
\l__physicx_tmpa_int	52, 1556, 1566, 1568, 1574, 1576
\l__physicx_tmpa_seq	86, 96, 130, 133, 144, 154
\l__physicx_tmpa_tl	100, 101, 104, 110, 111, 114, 319, 320
\l__physicx_tmparr_tl	1361, 1362, 1363, 1364
\l__physicx_tmpb_int	53
\l__physicx_tmpb_seq . . .	99, 100, 110
\l__physicx_tmpdiag_seq	1219, 1230, 1240, 1243, 1244, 1254, 1260, 1261, 1263, 1269, 1270, 1278, 1279, 1280, 1290, 1291, 1292, 1301, 1302, 1303
\l__physicx_tmpdiag_tl	1216, 1217, 1218, 1219
\l__physicx_tmpitem_tl	1327, 1328, 1329, 1335, 1353
__physicx_xquantity_aux:nnnn	389, 396, 417
\physicxempty .	704, 718, 726, 763, 764, 872
\physicxexcept	766, 843, 852, 861, 867, 869
\PHYSICXIGNORE	7, 8, 712, 723, 1335
\physicxset	283
\physicxtmp . . .	460, 1051, 1420, 1518, 1624
\poissonbracket	647
\pqty	647
prg commands:	
\prg_generate_conditional_- variant:Nnn	199
\prg_new_conditional:Npnn	14, 19, 24, 29, 160, 165, 181, 1392
\prg_replicate:nn	1137, 1167
\prg_return_false:	17, 22, 27, 35, 47, 163, 168, 196, 1413
\prg_return_true:	17, 22, 27, 34, 47, 163, 168, 196, 1395, 1412
\prg_set_conditional:Npnn	44
\ProcessKeysPackageOptions	243
Q	
\qty	647
\quantity	572, 650
quark commands:	
\quark_if_recursion_tail_stop:n	528, 529
\q_recursion_stop	481
\q_recursion_tail	481
\qxmatrix	1033, 1645
\qxqty	458
R	
\rangle	350, 580, 607, 622
\rbrace	278, 279
\Re	278
\real	278
regex commands:	
\regex_match:nnTF . .	162, 167, 852, 861
\RequirePackage	223
\rgroup	604, 619
\right	217, 381, 406, 597, 604, 607, 613, 614, 619, 621, 622, 623
\rVert	664
S	
\sb	754, 1104, 1109, 1116, 1143, 1147, 1153
scan commands:	
\scan_stop:	1124, 1591
seq commands:	
\c_empty_seq	79, 1584
\seq_clear:N	96
\seq_concat:NNN	86
\seq_count:N	1243, 1244, 1260, 1261, 1269, 1270, 1278, 1279, 1290, 1291, 1301, 1302, 1366, 1373, 1560

<code>\use:nnnn</code>	4	<code>\vectorarrow</code>	265
<code>\use_i:nn</code> ...	40, 536, 1331, 1448, 1644	<code>\vectorbold</code>	263
<code>\use_i:nnn</code>	1644	<code>\vectorunit</code>	267
<code>\use_i_ii:nnn</code>	1446, 1643	<code>\Vert</code>	349, 581, 623
<code>\use_ii:nn</code>	321, 536, 1333	<code>\vert</code> .	348, 579, 586, 587, 588, 621, 653, 657
<code>\use_ii_i:nn</code>	1663, 1739	<code>\vnabla</code>	272
<code>\use_none:n</code>	42, 327	<code>\vqty</code>	647
<code>\use_none:nn</code>	761, 798, 1197	<code>\vysmbkcircle</code>	270
<code>\use_none:nnn</code>	37		
V			
<code>\vdots</code>	834, 1136, 1138, 1140, 1166, 1168, 1170, 1801	X	
<code>\vec</code>	266	<code>\xquantity</code>	416
<code>\vectimes</code>	271	Z	
		<code>\Z</code>	162, 167