

# The `physicx` package

Wenjian Chern (Longaster\*)

July 30, 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](mailto:longaster@163.com)

```

34     { \prg_return_true: }
35     { \prg_return_false: }
36 }
37 \cs_if_exist_use:NF \hook_gput_code:nnn { \use_none:nnn }
38 { package/unicode-math/after } { ./package }
39 {
40     \cs_gset_eq:NN \physicx_unimath:TF \use_i:nn
41     \cs_gset_eq:NN \physicx_unimath:T \use:n
42     \cs_gset_eq:NN \physicx_unimath:F \use_none:n
43 }
44 \prg_set_conditional:Npnn \physicx_unimath: { T, F, TF }
45 {
46     \tl_if_exist:cTF { ver @ unicode-math . \@pkgextension }
47     { \prg_return_true: } { \prg_return_false: }
48 }
49
50 \clist_new:N \l__physicx_tmpa_clist
51 \bool_new:N \l__physicx_tmpa_bool
52 \int_new:N \l__physicx_tmpa_int
53 \int_new:N \l__physicx_tmpb_int
54 \msg_new:nnnn { physicx } { unknown-key }
55 { The~key~‘#1’~is~unknown~and~is~being~ignored. }
56 {
57     The~module~#2~does~not~have~a~key~called~#1.\\
58     Check~that~you~have~spelled~the~key~name~correctly.
59 }
60 \msg_new:nnn { physicx } { diag-key }
61 { The~value~‘#1’~of~diag~key~is~unknown~and~is~being~ignored. }

```

## 1.1 Utils functions

```

\physicx_parse_range:nnnN Parse range, such as -3,6-8,9,10-.
\physicx_parse_range_check:
\physicx_parse_range_nocheck:
62 \int_new:N \l__physicx_begin_range_int
63 \int_new:N \l__physicx_end_range_int
64 \int_new:N \l__physicx_max_range_int
65 \int_new:N \l__physicx_min_range_int
66 \bool_new:N \l__physicx_invalid_range_bool % range
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_range_int {#1}
81     \int_set:Nn \l__physicx_max_range_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, nneN }
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, neN }
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     \__physicx_parse_range_action:nnn
98       {#1}
99     { \__physicx_parse_range_single:n {#1} }
100    {
101      \tl_if_empty:NTF \l__physicx_tmpa_tl
102        { \int_set_eq:NN \l__physicx_begin_range_int \l__physicx_min_range_int }
103        { \int_set:Nn \l__physicx_begin_range_int { \l__physicx_tmpa_tl } }
104      \tl_if_empty:NTF \l__physicx_tmpb_tl
105        { \int_set_eq:NN \l__physicx_end_range_int \l__physicx_max_range_int }
106        { \int_set:Nn \l__physicx_end_range_int { \l__physicx_tmpb_tl } }
107      \__physicx_parse_range_range:
108    }
109  }
110 \cs_new:Npn \physicx_set_parse_range_delimiter:n #1
111   {
112     \tl_if_empty:NTF {#1}
113     {
114       \cs_set:Npn \__physicx_parse_range_action:nnn ##1
115         { \__physicx_parse_range_aux:w ##1 \__physicx_do_nothing: \q_nil \q_physicx_special }
116       \cs_set:Npn \__physicx_parse_range_aux:w ##1##2 ##3 \q_physicx_special
117         {
118           \tl_set:Nx \l__physicx_tmpa_tl { \tl_trim_spaces:n {##1} }
119           \tl_set:Nx \l__physicx_tmpb_tl { \tl_trim_spaces:n {##2} }
120           \quark_if_nil:NTF {##3}
121         }
122     }
123     {
124       \cs_set:Npn \__physicx_parse_range_action:nnn ##1
125         { \__physicx_parse_range_aux:w ##1 #1 #1 \q_physicx_special }
126       \cs_set:Npn \__physicx_parse_range_aux:w ##1 #1 ##2 #1 ##3 \q_physicx_special
127         {
128           \tl_set:Nx \l__physicx_tmpa_tl { \tl_trim_spaces:n {##1} }
129           \tl_set:Nx \l__physicx_tmpb_tl { \tl_trim_spaces:n {##2} }
130           \tl_if_blank:NTF {##3}
131         }
132     }
133   }
134 \physicx_set_parse_range_delimiter:n { - }
135 \cs_new:Npn \__physicx_parse_range_single_check:n #1
136   {
137     \bool_lazy_or:nnTF

```

```

138     { \int_compare_p:nNn {#1} > \l__physicx_max_range_int }
139     { \int_compare_p:nNn {#1} < \l__physicx_min_range_int }
140     { \bool_set_true:N \l__physicx_invalid_range_bool }
141     { \seq_put_right:Nn \l__physicx_tmpa_seq {#1} }
142   }
143   \cs_new:Npn \__physicx_parse_range_single_nocheck:n #1
144     { \seq_put_right:Nn \l__physicx_tmpa_seq {#1} }
145   \cs_new_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_check:n
146   \cs_new:Npn \__physicx_parse_range_range_check:
147     {
148       \int_compare:nNnT \l__physicx_begin_range_int < \l__physicx_min_range_int
149       { \int_set_eq:NN \l__physicx_begin_range_int \l__physicx_min_range_int }
150       \int_compare:nNnT \l__physicx_end_range_int > \l__physicx_max_range_int
151       { \int_set_eq:NN \l__physicx_end_range_int \l__physicx_max_range_int }
152       \bool_lazy_or:nnTF
153       { \int_compare_p:nNn \l__physicx_begin_range_int > \l__physicx_max_range_int }
154       { \int_compare_p:nNn \l__physicx_begin_range_int > \l__physicx_end_range_int }
155       { \bool_set_true:N \l__physicx_invalid_range_bool }
156       {
157         \int_step_inline:nnn
158         { \l__physicx_begin_range_int } { \l__physicx_end_range_int }
159         { \seq_put_right:Nn \l__physicx_tmpa_seq {##1} }
160       }
161     }
162   \cs_new:Npn \__physicx_parse_range_range_nocheck:
163     {
164       \int_compare:nNnTF \l__physicx_begin_range_int > \l__physicx_end_range_int
165       { \bool_set_true:N \l__physicx_invalid_range_bool }
166       {
167         \int_step_inline:nnn
168         { \l__physicx_begin_range_int } { \l__physicx_end_range_int }
169         { \seq_put_right:Nn \l__physicx_tmpa_seq {##1} }
170       }
171     }
172   \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 ??.)

```

173   \cs_new:Npn \__physicx_if_keyval:nTF #1
174     { \tl_if_in:nnTF {#1} { = } }
175   \prg_new_conditional:Npnn \physicx_if_num:n #1 { T, F, TF }
176     {
177       \regex_match:nnTF { \A [[:digit:]]+ \Z } {#1}
178       { \prg_return_true: } { \prg_return_false: }
179     }
180   \prg_new_conditional:Npnn \physicx_if_num_sign:n #1 { T, F, TF }
181     {
182       \regex_match:nnTF { \A [\+|-]* [[:digit:]]+ \Z } {#1}
183       { \prg_return_true: } { \prg_return_false: }
184     }
185   \cs_new:Npn \physicx_search_also:nn #1#2
186     {
187       \clist_map_inline:nn {#1}
188       {

```

```

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

```

```

242     compat .bool_set:N = \g__physicx_compat_bool ,
243     compat .default:n = true ,
244     short .bool_set:N = \g__physicx_short_bool ,
245     short .default:n = true ,
246     physics .code:n = \__physicx_loadpackage_options:nnn {#1} { } {physics} ,
247     physics .default:n = { } ,
248     mathtools .code:n = \__physicx_loadpackage_options:nnn {#1} { } {mathtools} ,
249     mathtools .default:n = { } ,
250     unimath .code:n = \__physicx_loadpackage_options:nnn {#1} { } { unicode-math } ,
251     unimath .default:n = { } ,
252     reqty .bool_set:N = \g__physicx_reqty_bool ,
253     reqty .default:n = true ,
254     reqty .initial:n = true ,
255     noqty .meta:n = { reqty = false } ,
256     fixdif .bool_set:N = \g__physicx_fixdif_bool ,
257   }
258   %
259   \ProcessKeysPackageOptions { physicx }
260   %
261   \@ifpackageloaded{physics}
262   { \bool_set_true:N \g__physicx_compat_bool }
263   { }
264   \@ifpackageloaded{mathtools}
265   { \bool_set_true:N \g__physicx_mathtools_bool }
266   { \bool_set_false:N \g__physicx_mathtools_bool }
267   %
268   \physicx_compat:T
269   {
270     \tl_set_eq:NN \ordersymbol \c_physicx_order_tl
271     \tl_set_eq:NN \Ordersymbol \c_physicx_Order_tl
272   }
273   %
274   \@ifpackageloaded {unicode-math}
275   { \physicx_use_uni_bfit_type: }
276   { \physicx_use_amssymb_type: }
277   \physicx_unimath:T { %% TODO:
278     \cs_set:Npn \__physicx_vnabla: { \sympbf \nabla }
279     \AtBeginDocument{
280       \DeclareDocumentCommand\vectorbold{ s m }
281       { \IfBooleanTF{#1} { \physicx_bf:{#2} } { \mathbf{#2} } }
282       \DeclareDocumentCommand\vectorarrow{ s m }
283       { \IfBooleanTF{#1} { \vec{\physicx_bf:{#2}} } { \vec{\mathbf{#2}} } }
284       \DeclareDocumentCommand\vectorunit{ s m }
285       { \IfBooleanTF{#1} { \physicx_bf:{\hat{#2}} } { \hat{\mathbf{#2}} } }
286       \setmathfont [range={"2219}]{STIX~Two~Math}
287       \DeclareDocumentCommand \dotproduct { } { \vysmblkcircle }
288       \DeclareDocumentCommand \crossproduct { } { \vectimes }
289       \DeclareDocumentCommand \vnabla { } { \__physicx_vnabla: }
290       \cs_set_eq:NN \divisionsymbol \div
291       \cs_set_eq:NN \div \divergence
292     }
293     \physicx_compat:T {
294       \AtBeginDocument{
295         \bool_if:NT \g__physicx_fixdif_bool { \cs_set_eq:NN \diffd \@dif }

```

```

296 \let\real\Re \DeclareDocumentCommand\Re{g}{\IfNoValueTF{#1}{\operatorname{Re}}{\fbrace
297 \let\imaginary\Im \DeclareDocumentCommand\Im{g}{\IfNoValueTF{#1}{\operatorname{Im}}{\f
298 }
299 }
300 }
301 %
302 \bool_if:NT \g__physicx_fixdif_bool
303 {
304 \hook_gput_code:nnn { package/fixdif/before } { physicx }
305 { \cs_set_eq:NN \__physicx_nabla: \nabla }
306 \hook_gput_code:nnn { package/fixdif/after } { physicx }
307 { \tl_map_function:nN { \letdif \newdif \renewdif } \__physicx_fixdif_list:N }
308 \AtBeginDocument { \renewdif \__physicx_vnabla: { \sympbf \__physicx_nabla: } }
309 }
310 \cs_new_protected:Npn \__physicx_fixdif_list:N #1
311 {
312 \cs_if_free:cT { \cs_to_str:N #1 list }
313 {
314 \cs_new_protected:cpn { \cs_to_str:N #1 list }
315 {
316 \keyval_parse:nnn { \__physicx_fixdif:Nn #1 } { \__physicx_fixdif:Nnn #1 }
317 }
318 }
319 }
320 \cs_new_protected:Npn \__physicx_fixdif:Nnn #1#2#3
321 {
322 \tl_if_head_eq_meaning:nNTF {#2} *
323 { \exp_args:NNc #1 * { \tl_tail:n {#2} } {#3} }
324 { \exp_args:Nc #1 {#2} {#3} }
325 }
326 \cs_new_protected:Npn \__physicx_fixdif:Nn #1#2
327 {
328 \exp_args:NNnx \__physicx_fixdif:Nnn #1 {#2}
329 { \tl_if_head_eq_meaning:nNTF {#2} * { \tl_tail:n {#2} } {#2} }
330 }

```

`\physicxset` `physicx` setup command.

```

331 \NewDocumentCommand \physicxset { s m }
332 {
333 \IfBooleanTF {#1}
334 { \keys_set:nn { physicx/#2 } }
335 { \keys_set:nn { physicx } {#2} }
336 }

```

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

## 1.2 Quantity things

### 1.2.1 New quantity interfaces

```

337 \tl_new:N \l__physicx_quantity_args_tl
338 \tl_new:N \l__physicx_quantity_code_tl
339 \tl_new:N \l__physicx_quantity_left_size_tl
340 \tl_new:N \l__physicx_quantity_left_tl

```

```

341 \tl_new:N \l__physicx_quantity_post_tl
342 \tl_new:N \l__physicx_quantity_pre_tl
343 \tl_new:N \l__physicx_quantity_right_size_tl
344 \tl_new:N \l__physicx_quantity_right_tl
345 \keys_define:nn { physicx }
346 { quantity .code:n = \keys_set:nn { physicx/quantity } {#1} }
347 \keys_define:nn { physicx/quantity }
348 {
349   pre .tl_set:N = \l__physicx_quantity_pre_tl ,
350   post .tl_set:N = \l__physicx_quantity_post_tl ,
351   left .tl_set:N = \l__physicx_quantity_left_tl ,
352   right .tl_set:N = \l__physicx_quantity_right_tl ,
353   left-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_left_size_tl #1 } ,
354   right-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_right_size_tl #1 } ,
355   size .meta:n = { left-size = {#1} , right-size = {#1} } ,
356   noauto .meta:n = { left-size = \c_empty_tl , right-size = \c_empty_tl } ,
357   noauto .value_required:n = false ,
358   args .code:n =
359     \tl_set:Nn \l__physicx_quantity_args_tl { [#1] } ,
360   args* .tl_set:N = \l__physicx_quantity_args_tl ,
361   code .tl_set:N = \l__physicx_quantity_code_tl ,
362   type .multichoice: ,
363
364   settype .code:n = \setquantitytype #1 ,
365
366   unknown .code:n =
367     \tl_set:Nx \l__physicx_tmpa_tl { \tl_head:N \l_keys_key_str }
368     \token_if_eq_meaning:NNTF \l__physicx_tmpa_tl \c_backslash_str
369     { \use:n } { \use_ii:nn }
370   {
371     \cs_if_exist:CTF { \tl_tail:N \l_keys_key_str }
372     {
373       \keys_set:nx { physicx/quantity }
374       { size = \exp_not:c { \tl_tail:N \l_keys_key_str } }
375       \use_none:n
376     }
377     { \use:n }
378   }
379   {
380     \physicx_search_also:nnF
381     {
382       physicx/quantity/type ,
383     }
384     {#1}
385     {
386       \msg_error:nnxx { physicx } { unknown-key }
387       \l_keys_path_str { physicx/quantity }
388     }
389   } ,
390 }
391 \NewDocumentCommand \setquantitytype { >{ \TrimSpaces } m }
392 { \physicx_new_type:nnn { quantity } {#1} }
393 \setquantitytype { b } { left={[] , right={[] } , }
394 \setquantitytype { B } { left={\{ , right={\} } , }

```



```

395 \setquantitytype { p } { left={ ( } , right={ ) } , }
396 \setquantitytype { v } { left=\vert , right=\vert , }
397 \setquantitytype { V } { left=\Vert , right=\Vert , }
398 \setquantitytype { a } { left=\langle , right=\rangle , }
399 \setquantitytype { m } { left=\begin{matrix} , right=\end{matrix} , noauto }
400 \setquantitytype { bm } { left=\begin{bmatrix} , right=\end{bmatrix} , noauto }
401 \setquantitytype { Bm } { left=\begin{Bmatrix} , right=\end{Bmatrix} , noauto }
402 \setquantitytype { pm } { left=\begin{pmatrix} , right=\end{pmatrix} , noauto }
403 \setquantitytype { vm } { left=\begin{vmatrix} , right=\end{vmatrix} , noauto }
404 \setquantitytype { Vm } { left=\begin{Vmatrix} , right=\end{Vmatrix} , noauto }
405 \setquantitytype { sm } { left=\begin{smallmatrix} , right=\end{smallmatrix} , noauto }
406 \physics_mathtools:T
407 {
408   \setquantitytype { m* } { left=\begin{matrix}* , right=\end{matrix}* , noauto }
409   \setquantitytype { bm* } { left=\begin{bmatrix}* , right=\end{bmatrix}* , noauto }
410   \setquantitytype { Bm* } { left=\begin{Bmatrix}* , right=\end{Bmatrix}* , noauto }
411   \setquantitytype { pm* } { left=\begin{pmatrix}* , right=\end{pmatrix}* , noauto }
412   \setquantitytype { vm* } { left=\begin{vmatrix}* , right=\end{vmatrix}* , noauto }
413   \setquantitytype { Vm* } { left=\begin{Vmatrix}* , right=\end{Vmatrix}* , noauto }
414   \setquantitytype { sm* } { left=\begin{smallmatrix}* , right=\end{smallmatrix}* , noauto }
415   \setquantitytype { sbm } { left=\begin{bsmallmatrix} , right=\end{bsmallmatrix} , noauto }
416   \setquantitytype { sBm } { left=\begin{Bsmallmatrix} , right=\end{Bsmallmatrix} , noauto }
417   \setquantitytype { spm } { left=\begin{psmallmatrix} , right=\end{psmallmatrix} , noauto }
418   \setquantitytype { svm } { left=\begin{vsmallmatrix} , right=\end{vsmallmatrix} , noauto }
419   \setquantitytype { sVm } { left=\begin{Vsmallmatrix} , right=\end{Vsmallmatrix} , noauto }
420   \setquantitytype { sbm* } { left=\begin{bsmallmatrix}* , right=\end{bsmallmatrix}* , noauto }
421   \setquantitytype { sBm* } { left=\begin{Bsmallmatrix}* , right=\end{Bsmallmatrix}* , noauto }
422   \setquantitytype { spm* } { left=\begin{psmallmatrix}* , right=\end{psmallmatrix}* , noauto }
423   \setquantitytype { svm* } { left=\begin{vsmallmatrix}* , right=\end{vsmallmatrix}* , noauto }
424   \setquantitytype { sVm* } { left=\begin{Vsmallmatrix}* , right=\end{Vsmallmatrix}* , noauto }
425 }
426 \keys_set:nn { physics/quantity }
427 {
428   left-size = \left ,
429   right-size = \right ,
430   type = p ,
431 }

\physics_xquantity:nn
  \newxquantity
  \NewXQuantity
392 \cs_new:Npn \physics_xquantity:nn #1#2
393 {
394   \group_begin:
395   \keys_set:nn { physics/quantity } {#1}
396   \tl_if_empty:nF {#2} { \tl_set:Nn \l__physics_quantity_code_tl {#2} }
397   \__physics_xquantity_aux:oooo
398   { \l__physics_quantity_left_tl }
399   { \l__physics_quantity_args_tl }
400   { \l__physics_quantity_code_tl }
401   { \l__physics_quantity_right_tl }
402   \group_end:
403 }
404 \cs_new:Npn \__physics_xquantity_aux:nnnn #1#2#3#4
405 {
406   \l__physics_quantity_pre_tl

```

```

447 \bool_lazy_or:nnTF
448 { \tl_if_empty_p:N \l__physicx_quantity_left_size_tl }
449 { \tl_if_empty_p:N \l__physicx_quantity_right_size_tl }
450 { #1 #2 #3 #4 }
451 {
452   \bool_lazy_or:nnTF
453   { \token_if_eq_meaning_p:NN \l__physicx_quantity_left_size_tl \left }
454   { \token_if_eq_meaning_p:NN \l__physicx_quantity_right_size_tl \right }
455   { \physicx_left: #1 #2 #3 \physicx_right: #4 }
456   {
457     \physicx_left:N \l__physicx_quantity_left_size_tl #1 #2
458     #3
459     \physicx_right:N \l__physicx_quantity_right_size_tl #4
460   }
461 }
462 \l__physicx_quantity_post_tl
463 }
464 \NewDocumentCommand \xquantity { } { \physicx_xquantity:nn }
465 \cs_generate_variant:Nn \__physicx_xquantity_aux:nnnn { oooo }
466 \NewDocumentCommand \newxquantity { m o o m m }
467 {
468   \IfNoValueTF {#2}
469   {
470     \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
471     { \newcommand ##1 }
472   }
473   {
474     \IfNoValueTF {#3}
475     {
476       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
477       { \newcommand ##1 [#2] }
478     }
479     {
480       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
481       { \newcommand ##1 [#2] [#3] }
482     }
483   }
484   \exp_args:Nc \__physicx_new_xquantity_aux:w
485   { \cs_to_str:N #1~star }
486   { \physicx_xquantity:nn { #4 , noauto } {#5} }
487   \exp_args:Nc \__physicx_new_xquantity_aux:w
488   { \cs_to_str:N #1~unstar }
489   { \physicx_xquantity:nn { #4 } {#5} }
490   \exp_args:NNx \newcommand #1
491   {
492     \exp_not:N \@ifstar
493     \exp_not:c { \cs_to_str:N #1~star }
494     \exp_not:c { \cs_to_str:N #1~unstar }
495   }
496 }
497 \NewDocumentCommand \NewXQuantity { m m m m }
498 {
499   \NewDocumentCommand #1 { s #2 }
500   {

```

```

501         \IfBooleanTF {##1}
502         { \physicx_xquantity:nn { #3 , noauto } {#4} }
503         { \physicx_xquantity:nn { #3 } {#4} }
504     }
505 }
506 \NewXQuantity \qxqty { 0{ } m } { #2 } {#3}
507 \NewXQuantity \txqty { 0{p} 0{ } m } { type={#2}, #3 } {#4}

```

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

## 1.2.2 Legacy quantity

```

\physicx_declare_legacy_quantity:nnNn
\@declarequantitycmd
508 \tl_new:N \physicx_tmp
509 \tl_new:N \l__physicx_cmd_noauto_body_tl
510 \bool_new:N \l__physicx_cmd_noauto_body_bool
511 \tl_new:N \l__physicx_cmd_auto_body_tl
512 \bool_new:N \l__physicx_cmd_auto_body_bool
513 \tl_new:N \l__physicx_cmd_arg_spec_tl
514 \int_new:N \l__physicx_cmd_arg_int
515 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
516 {
517     \tl_clear:N \l__physicx_cmd_noauto_body_tl
518     \tl_clear:N \l__physicx_cmd_auto_body_tl
519     \tl_clear:N \l__physicx_cmd_arg_spec_tl
520     \int_set:Nn \l__physicx_cmd_arg_int {#1}
521     \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
522     \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
523 }
524 % noauto, auto, cmd, body
525 \cs_new:Npn \physicx_declare_legacy_quantity:nnNn #1#2#3#4
526 {
527     \__physicx_declare_init:nnn { 3 } {#1} {#2}
528     \__physicx_declare_legacy_quantity_aux:nw #4
529     \q_recursion_tail \q_recursion_tail \q_recursion_stop
530     \__physicx_declare_legacy_quantity_aux:NcVWV
531     #3 { \cs_to_str:N #3 ~ body }
532     \l__physicx_cmd_arg_spec_tl
533     \l__physicx_cmd_noauto_body_tl
534     \l__physicx_cmd_auto_body_tl
535 }
536 % arg spec, pre, body to replace(start from #4), post
537 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
538 {
539     \int_incr:N \l__physicx_cmd_arg_int
540     \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
541         \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
542         \tl_set:Nx \l__physicx_tmp_tl
543         {
544             {
545                 \exp_not:N \tl_if_novalue_p:n
546                 {
547                     \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:

```

```

548         \or: \or: \or:
549         \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
550         \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
551     \fi:
552 }
553 }
554 }
555 \if_bool:N \l__physicx_cmd_noauto_body_bool
556     \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
557     \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
558     {
559         {
560             % if is '.', use none
561             \str_if_eq:nnTF {#2} {.} {} {#2}
562             #3
563             \str_if_eq:nnTF {#4} {.} {} {#4}
564         }
565     }
566 \fi:
567 \if_bool:N \l__physicx_cmd_auto_body_bool
568     \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
569     \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
570     { { ##1 #2 #3 ##2 #4 } }
571 \fi:
572 \fi:
573 }
574 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
575 {
576     \quark_if_recursion_tail_stop:n {#1}
577     \quark_if_recursion_tail_stop:n {#2}
578     \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
579     \__physicx_declare_legacy_quantity_aux:nw
580 }
581 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
582 {
583     \__physicx_nauto_case:nnnn
584     { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
585     {
586         \cs_set_protected:Npn #1
587         {
588             \peek_charcode_ignore_spaces:NTF \let
589             { #2 } { #2 [ \physicx_left: ] \physicx_right: }
590         }
591         \DeclareDocumentCommand #2 { 0{##2} m s #3 }
592         {
593             \IfBooleanTF { ##3 }
594             { \bool_case_false:n {#4} }
595             { \bool_case_false:n {#5} }
596         }
597     }
598 {
599     \cs_set_protected:Npn #1
600     { #2 \c_empty_tl \c_empty_tl }
601     \DeclareDocumentCommand #2 { m m s #3 }

```

```

602         { \bool_case_false:n {#4} }
603     }
604 }
605 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
606 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
607 {
608     \bool_if:NTF \l__physicx_cmd_noauto_body_bool
609     {
610         \bool_if:NTF \l__physicx_cmd_auto_body_bool
611         {#1} {#2}
612     }
613     {
614         \bool_if:NTF \l__physicx_cmd_auto_body_bool
615         {#3} {#4}
616     }
617 }
618 \cs_set_protected:Npn \@declarequantitycmd
619 { \physicx_declare_legacy_quantity:nnNn }

```

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

```

\quantity      Redefine some macros in physics package.
\evaluated
\matrixquantity
\smallmatrixquantity
620 \if_bool:N \g__physicx_reqty_bool
621 \physicx_declare_legacy_quantity:nnNn
622 \c_true_bool \c_true_bool \quantity
623 {
624     { !g } { { \{ } { #4 } { \} } }
625     { !o } { { [ } { #5 } { ] } }
626     { !d() } { { ( } { #6 } { ) } }
627     { !d|| } { { \vert } { #7 } { \vert } }
628     { !d<> } { { \langle } { #8 } { \rangle } }
629     { !d== } { { \Vert } { #9 } { \Vert } }
630 }
631 \physicx_declare_legacy_quantity:nnNn
632 \c_true_bool \c_true_bool \evaluated
633 {
634     { !g } { { . } { #4 \nobreak } { \vert } }
635     { !d[ ] } { { [ } { #5 \nobreak } { \vert } }
636     { !d( ) } { { ( } { #6 \nobreak } { \vert } }
637 }
638 \physicx_declare_legacy_quantity:nnNn
639 \c_true_bool \c_false_bool \matrixquantity
640 {
641     { !g }
642     {
643         { \IfBooleanT{#3}{\left\{ } }
644         { \begin{matrix} #4 \end{matrix} }
645         { \IfBooleanT{#3}{\right\} }
646     }
647     { !o } { { \begin{bmatrix} } {#5} { \end{bmatrix} } }
648     { !d() }
649     {
650         { \IfBooleanTF{#3}{\left\lgroup}{\left( } }

```

```

651 { \begin{matrix} #6 \end{matrix} }
652 { \IfBooleanTF{#3}{\right\rgroup}{\right}} }
653 }
654 { !d|| } { { \begin{vmatrix} } {#7} { \end{vmatrix} } } }
655 { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } } { \right\rangle } }
656 { !d== } { { \begin{Vmatrix} } {#9} { \end{Vmatrix} } } }
657 }
658 \physicsx_declare_legacy_quantity:nnNn
659 \c_true_bool \c_false_bool \smallmatrixquantity
660 {
661 { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } } { \right\} } }
662 { !o } { { \left[ } { \begin{smallmatrix} #5 \end{smallmatrix} } } { \right]} } }
663 { !d() }
664 {
665 { \IfBooleanTF{#3}{\left\lgroup}{\left( }
666 { \begin{smallmatrix} #6 \end{smallmatrix} }
667 { \IfBooleanTF{#3}{\right\rgroup}{\right)}} }
668 }
669 { !d|| } { { \left\vert } { \begin{smallmatrix} #7 \end{smallmatrix} } } { \right\vert } }
670 { !d<> } { { \left\lvert } { \begin{smallmatrix} #8 \end{smallmatrix} } } { \right\rvert } }
671 { !d== } { { \left\Vert } { \begin{smallmatrix} #9 \end{smallmatrix} } } { \right\Vert } }
672 }
673 \fi:

```

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

```

\physicsx_declare_legacy_paren:NnnnNnN
\@declareparenccmd
674 %% cmd, arg spec, replace(start from #6), pre, left, right, post
675 \cs_new:Npn \physicsx_declare_legacy_paren:NnnnNnN #1#2#3#4#5#6#7
676 {
677 \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
678 {
679 \bool_case_true:nF
680 {
681 { \bool_if_p:n {##2} } { #4 \physicsx_left:N \bigl #5 #3 \physicsx_right:N \bigr
682 { \bool_if_p:n {##3} } { #4 \physicsx_left:N \Bigl #5 #3 \physicsx_right:N \Bigr
683 { \bool_if_p:n {##4} } { #4 \physicsx_left:N \biggl #5 #3 \physicsx_right:N \biggr
684 { \bool_if_p:n {##5} } { #4 \physicsx_left:N \Biggl #5 #3 \physicsx_right:N \Biggr
685 }
686 {
687 \IfBooleanTF {##1}
688 { #4 #5 #3 #6 #7 }
689 { #4 \physicsx_left: #5 #3 \physicsx_right: #6 #7 }
690 }
691 }
692 }
693 \cs_set_protected:Npn \@declareparenccmd
694 { \physicsx_declare_legacy_paren:NnnnNnN }

```

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

```

\qty Redefine some macros in physics package.
\mqty
695 \if_bool:N \g__physicsx_reqty_bool
\smqty 696 \physicsx_option_or:nnT { compat } { short }
\pqty
\bqty
\vqty
\Bqty
\absolutevalue
\eval
\abs
\norm
\order
\oorder

```

```

697 {
698   \cs_set:Npn \qty { \quantity }
699   \physics_declare_legacy_paren:NnnnNNn \pqty { m } {#6} { } { } { } { }
700   \physics_declare_legacy_paren:NnnnNNn \bqty { m } {#6} { } { } [ ] { }
701   \physics_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } { } \vert \vert { }
702   \physics_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } { } \{ \} { }
703 }
704 \physics_declare_legacy_paren:NnnnNNn \absolutevalue
705 { m } {#6} { } { } \vert \vert { }
706 \physics_option_or:nnT { compat } { short }
707 {
708   \cs_set:Npn \eval { \evaluated }
709   \cs_set:Npn \abs { \absolutevalue }
710 }
711 \physics_declare_legacy_paren:NnnnNNn \norm
712 { m } {#6} { } { } \lVert \rVert { }
713 \physics_compat:TF
714 {
715   \physics_declare_legacy_paren:NnnnNNn \order
716   { m } {#6} { } { \c_physicsx_Order_tl } { } { }
717 }
718 {
719   \physics_declare_legacy_paren:NnnnNNn \order
720   { m } {#6} { } { \c_physicsx_order_tl } { } { }
721 }
722 \physics_declare_legacy_paren:NnnnNNn \commutator
723 { m m } { #6 , #7 } { } { } [ ] { }
724 \physics_option_or:nnT { compat } { short }
725 { \cs_set:Npn \comm { \commutator } }
726 \physics_declare_legacy_paren:NnnnNNn \poissonbracket
727 { m m } { #6 , #7 } { } { } \{ \} { }
728 \physics_option_or:nnT { compat } { short }
729 {
730   \cs_set:Npn \pb { \poissonbracket }
731   \cs_set:Npn \anticommutator { \poissonbracket }
732   \cs_set:Npn \acomm { \poissonbracket }
733 }
734 \fi:
735 \physics_declare_legacy_paren:NnnnNNn \OOrder
736 { m } {#6} { } { \c_physicsx_Order_tl } { } { }
737 \physics_declare_legacy_paren:NnnnNNn \oorder
738 { m } {#6} { } { \c_physicsx_order_tl } { } { }

```

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

## 1.3 Matrix things

### 1.3.1 Matrix auxillary functions

```

739 \cs_new_nopar:Npn \__physicsx_matrix_calc:nn #1#2
740 {
741   \int_set:Nn \l__physicsx_matrix_rows_int
742   { \int_max:nn {#1} \l__physicsx_matrix_rows_int }
743   \int_set:Nn \l__physicsx_matrix_cols_int
744   { \int_max:nn {#2} \l__physicsx_matrix_cols_int }

```

```

745 }
746 % use matrix element
747 \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
748 {
749   \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
750     \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
751   \else:
752     \exp_not:o { \physicxempty }
753   \fi:
754 }
755 % set matrix element, check or not
756 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
757 { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
758 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
759 {
760   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
761   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
762 }
763 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_cke:nnn #1#2#3
764 {
765   \tl_if_empty:nTF {#3}
766   { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
767   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
768 }
769 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
770 {
771   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
772   {
773     \tl_if_empty:nTF {#3}
774     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
775     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
776   }
777 }
778 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
779 \__physicx_matrix_set_r_c_ckigep:nnn
780 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
781 \__physicx_matrix_set_r_c_nock:nnn
782 % align, cr, sep symbol
783 \str_const:Nn \physicx@align { , }
784 \str_const:Nn \physicx@cr { ; }
785 \str_const:Nn \physicx@sep { , }
786 \bool_new:N \l__physicx_matrix_infinite_bool
787 \bool_new:N \l__physicx_matrix_dotrow_bool
788 \bool_new:N \l__physicx_matrix_dotcol_bool
789 \tl_new:N \l__physicx_matrix_array_tl
790 \tl_new:N \l__physicx_matrix_body_tl
791 \int_new:N \l__physicx_matrix_rows_int
792 \int_new:N \l__physicx_matrix_cols_int
793 \tl_new:N \l__physicx_matrix_main_tl
794 \clist_new:N \l__physicx_matrix_diag_clist
795 \clist_new:N \l__physicx_matrix_item_clist
796 \bool_new:N \l__physicx_matrix_diag_bool
797 \seq_new:N \l__physicx_row_list_seq
798 \seq_new:N \l__physicx_col_list_seq

```



```

799 % expand input
800 \cs_new_eq:NN \__physicx_expand:w \exp_not:o
801 %% main, row iterate, col iterate
802 \cs_new_nopar:Npn \physicx@matricelement #1#2#3 { #1 \sb { #2 #3 } }
803 \cs_new_nopar:Npn \__physicx_matrix_row_iterate:n #1 { #1 }
804 \tl_new:N \l__physicx_matrix_last_row_tl
805 \tl_new:N \l__physicx_matrix_last_col_tl
806 \cs_new_nopar:Npn \__physicx_matrix_col_iterate:n #1 { #1 }
807 \cs_new_nopar:Npn \__physicx_matrix_begin:w { }
808 \cs_new_nopar:Npn \__physicx_matrix_end:w { }
809 \cs_new_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn
810 \bool_new:N \l__physicx_matrix_expand_element_bool
811 % when element is empty use \physicxempty
812 \tl_new:N \physicxempty
813 % save 'element-except' key's value
814 \tl_new:N \physicxexcept
815 \tl_new:N \l__physicx_matrix_args_tl
816 \tl_new:N \l__physicx_matrix_after_begin_tl
817 \tl_new:N \l__physicx_matrix_after_end_tl
818 \bool_new:N \l__physicx_matrix_transpose_bool
819 \bool_new:N \l__physicx_matrix_enhanced_bool
820 \dim_new:N \l__physicx_matrix_sep_dim
821 \cs_new:Npn \__physicx_adi:nnn #1#2#3 { #1#2#3 }
822 \tl_new:N \l__physicx_matrix_beginning_tl
823 \tl_new:N \l__physicx_matrix_ending_tl

```

### 1.3.2 Matrix keys

```

824 \keys_define:nn { physicx }
825 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
826 \keys_define:nn { physicx/matrix }
827 {
828   array .tl_set:N = \l__physicx_matrix_array_tl ,
829   expand .choice: ,
830   expand / none .code:n =
831     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
832   expand / text-expand .code:n =
833     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
834   expand / f .code:n =
835     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
836   expand / romanual .meta:n = { expand = f } ,
837   expand / x .code:n =
838     \cs_set_eq:NN \__physicx_expand:w \use:n ,
839   expand / edef .meta:n = { expand = x } ,
840   rows .int_set:N = \l__physicx_matrix_rows_int ,
841   cols .int_set:N = \l__physicx_matrix_cols_int ,
842   auto-update .choice: ,
843   auto-update / true .code:n =
844     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,
845   auto-update / false .code:n =
846     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
847   auto-update .default:n = true ,
848   main .tl_set:N = \l__physicx_matrix_main_tl ,
849   row-list .code:n =
850     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,

```

```

851 col-list .code:n =
852   \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
853   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
854   infinite .default:n = true ,
855   !infinite .code:n =
856     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
857   element-code .cs_set:Np = \physicx@matricielement #1#2#3 ,
858   element-code* .choice: ,
859   element-code* / except-empty .code:n =
860     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
861     \physicx@matricielement
862     \cs_set:Npn \physicx@matricielement ##1##2##3
863     {
864       \tl_if_empty:nTF {##1}
865       {##1}
866       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
867     } ,
868   element-code* / except-blank .code:n =
869     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
870     \physicx@matricielement
871     \cs_set:Npn \physicx@matricielement ##1##2##3
872     {
873       \tl_if_blank:nTF {##1}
874       {##1}
875       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
876     } ,
877   element-code* / except-dots .code:n =
878     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
879     \physicx@matricielement
880     \cs_set:Npn \physicx@matricielement ##1##2##3
881     {
882       \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
883       {##1}
884       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
885     } ,
886   element-code* / except-tl .code:n =
887     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
888     \physicx@matricielement
889     \cs_set:Npn \physicx@matricielement ##1##2##3
890     {
891       \tl_if_in:onTF { \physicxexcept } {##1}
892       {##1}
893       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
894     } ,
895   element-code* / except-regex .code:n =
896     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
897     \physicx@matricielement
898     \cs_set:Npn \physicx@matricielement ##1##2##3
899     {
900       \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
901       {##1}
902       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
903     } ,
904   element-code* / only-regex .code:n =

```

```

905 \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
906 \physicx@matricelement
907 \cs_set:Npn \physicx@matricelement ##1##2##3
908 {
909   \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
910   { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
911   {##1}
912 } ,
913 element-code* / unknown .code:n =
914   \cs_set:Npx \physicx@matricelement { \exp_not:c {#1} },
915 element-exception .tl_set:N = \physicxexcept ,
916 element-exception+ .code:n =
917   \tl_put_right:Nn \physicxexcept {#1} ,
918 expand-element .bool_set:N = \l__physicx_matrix_expand_element_bool ,
919 expand-element .default:n = true ,
920 empty .tl_set:N = \physicxempty ,
921 check .choice: ,
922 check / none .code:n =
923   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
924   \__physicx_matrix_set_r_c_nock:nnn ,
925 check / empty .code:n =
926   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
927   \__physicx_matrix_set_r_c_cke:nnn ,
928 check / ignore .code:n =
929   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
930   \__physicx_matrix_set_r_c_ckig:nnn ,
931 check / igep .code:n =
932   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
933   \__physicx_matrix_set_r_c_ckigep:nnn ,
934 check / all .code:n =
935   \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
936   \__physicx_matrix_set_r_c_ckall:nnn ,
937 check .default:n = all ,
938 row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
939 col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
940 last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
941 last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
942 diag .clist_set:N = \l__physicx_matrix_diag_clist ,
943 diag+ .code:n =
944   \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
945 diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
946 diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
947 diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
948 item .clist_set:N = \l__physicx_matrix_item_clist ,
949 item+ .code:n =
950   \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
951 item-now .code:n = \physicx_matrix_item_parse:n {#1} ,
952 item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
953 item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
954 check-range .choice: ,
955 check-range / true .code:n = \physicx_parse_range_check: ,
956 check-range / false .code:n = \physicx_parse_range_noccheck: ,
957 check-range .default:n = true ,
958 begin .tl_set:N = \__physicx_matrix_begin:w ,

```

```

959 end .tl_set:N = \__physicx_matrix_end: ,
960 args .code:n =
961 \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,
962 args* .tl_set:N = \l__physicx_matrix_args_tl ,
963 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
964 after-begin+ .code:n =
965 { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
966 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
967 after-end+ .code:n =
968 { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
969 sepdim .dim_set:N = \l__physicx_matrix_sepdim ,
970 type .multichoice: ,
971 saveto .tl_set:N = \l__physicx_matrix_save_tl ,
972 saveto* .code:n =
973 \tl_set:Nn \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
974 transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
975 transpose .default:n = true ,
976 ' .meta:n = { transpose = true } ,
977 T .meta:n = { transpose = true } ,
978 MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
979 enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
980 enhanced .default:n = true ,
981 !enhanced .code:n =
982 \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
983 cr .tl_set:N = \physicx@cr ,
984 align .tl_set:N = \physicx@align ,
985 sep .tl_set:N = \physicx@sep ,
986 adi-order .choice: ,
987 adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
988 adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
989 adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
990 adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
991 adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
992 adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
993 beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
994 beginning+ .code:n =
995 \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
996 ending .tl_set:N = \l__physicx_matrix_ending_tl ,
997 ending+ .code:n =
998 \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
999
1000 settype .code:n = \setmatrxtype #1 ,
1001
1002 unknown .code:n =
1003 \physicx_search_also:nnF
1004 {
1005     physicx/matrix/type ,
1006     physicx/matrix/expand ,
1007     physicx/matrix/element-code* ,
1008 }
1009 {#1}
1010 {
1011     \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
1012     {

```

```

1013         \keys_set:nx { physicx/matrix }
1014         { MaxMatrixCols = \l_keys_key_str }
1015     }
1016     {
1017         \msg_error:nxxx { physicx } { unknown-key }
1018         \l_keys_path_str { physicx/matrix }
1019     }
1020 } ,
1021 }

```

`\physicx_matrix_new_type:nnn`  
`\physicx_matrix_new_type:nn`  
`\setmatrixtype`

```

1022 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
1023 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
1024 \cs_new:Npn \physicx_matrix_new_type:nn
1025 { \physicx_new_type:nnn { matrix } }
1026 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
1027 {
1028     \IfBooleanTF {#1}
1029     { \physicx_matrix_new_type:nn {#2} }
1030     { \physicx_matrix_new_type:nnn {#2} }
1031 }

```

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

A few types.

```

1032 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
1033 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
1034 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
1035 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
1036 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
1037 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
1038 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
1039 \physicx_mathtools:T
1040 {
1041     \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
1042     \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
1043     \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
1044     \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
1045     \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
1046     \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
1047     \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
1048     \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
1049     \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
1050     \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
1051     \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
1052     \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
1053     \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
1054     \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
1055     \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
1056     \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
1057     \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
1058 }

```

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

```

1059 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
1060 { \clist_set:cn { physicx@ #1 data@ #2 } }
1061 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
1062 {
1063   \clist_clear:c { l__physicx_matrix_ #1 _clist }
1064   \__physicx_matrix_add_data:nn {#1} {#2}
1065 }
1066 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
1067 {
1068   \clist_map_inline:nn {#2}
1069   {
1070     \clist_concat:ccc
1071     { l__physicx_matrix_ #1 _clist }
1072     { l__physicx_matrix_ #1 _clist }
1073     { physicx@ #1 data@ #2 }
1074   }
1075 }

```

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

Initial settings.

```

1076 \keys_set:nn { physicx/matrix }
1077 {
1078   type = m ,
1079   saveto = ? ,
1080 }

```

\qxmatri

```

1081 %% basicly, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
1082   xmatmmn-in-the-physics-package, but changed some
1083 % #1 = boolean, saveto matrix
1084 % #2 = star, infinite
1085 % #3 = options
1086 % #4 = letter for the entries
1087 % #5 = number of rows
1088 % #6 = number of explicit rows, default = 3
1089 % #7 = number of columns
1090 % #8 = number of explicit columns, default = 3
1091 \DeclareDocumentCommand \qxmatri { t= s 0{type=p} m m 0{3} m 0{3} }
1092 {
1093   \group_begin:
1094   \IfBooleanTF { #2 }
1095   { \bool_set_true:N \l__physicx_matrix_infinite_bool }
1096   { \bool_set_false:N \l__physicx_matrix_infinite_bool }
1097   \int_set:Nn \l__physicx_matrix_rows_int {#6}
1098   \int_set:Nn \l__physicx_matrix_cols_int {#8}
1099   \IfBooleanTF {#1}
1100   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1101   { \keys_set:nn { physicx/matrix } {#3} }
1102   \physicx_qxmatri:nnn {#4} {#5} {#7}
1103   \__physicx_matrix_save_or_print:
1104   \group_end:
1105 }
1106 \cs_new_protected:Nn \physicx_qxmatri:nnn
1107 {

```

```

1107 \bool_if:NTF \l__physicx_matrix_expand_element_bool
1108 {
1109     \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1110     \__physicx_matrix_appto_body_e:nnn
1111 }
1112 {
1113     \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
1114     \__physicx_matrix_appto_body_ne:nnn
1115 }
1116 % clear the variable containing the body of the matrix
1117 \tl_clear:N \l__physicx_matrix_body_tl
1118 % set the tentative number of explicit rows
1119 \physicx_if_num:NTF { #2 }
1120 {
1121     % number of rows is an integer
1122     \int_compare:NTF { #2 <= \l__physicx_matrix_rows_int }
1123     {
1124         % if #2 <= rows, we don't want a row of dots
1125         \bool_set_false:N \l__physicx_matrix_dotrow_bool
1126         \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1127     }
1128     {
1129         % we want a row of dots
1130         \bool_set_true:N \l__physicx_matrix_dotrow_bool
1131     }
1132 }
1133 % set the tentative number of explicit columns
1134 \physicx_if_num:NTF { #3 }
1135 {
1136     % number of cols is an integer
1137     \int_compare:NTF { #3 <= \l__physicx_matrix_cols_int }
1138     {
1139         % if #3 <= cols, we don't want a column of dots
1140         \bool_set_false:N \l__physicx_matrix_dotcol_bool
1141         \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1142     }
1143     {
1144         % we want a column of dots
1145         \bool_set_true:N \l__physicx_matrix_dotcol_bool
1146     }
1147 }
1148 % number of columns is symbolic, we want a column of dots
1149 \bool_set_true:N \l__physicx_matrix_dotcol_bool
1150 }
1151 % loop through the rows
1152 \int_step_inline:nn { \l__physicx_matrix_rows_int }
1153 {
1154     % add the first entry in the row
1155     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1156     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
1157     % add the further entries in the explicit columns
1158     \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1159     {
1160         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
1161         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1162         \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
1163     }
1164 }

```

```

1161 % if we have a column of dots, add \cdots and the last entry
1162 \bool_if:NT \l__physicx_matrix_dotcol_bool
1163 {
1164     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{##1 #3} }
1165     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1166     \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {#3}
1167 }
1168 % infinite matrix, add \cdots
1169 \bool_if:NT \l__physicx_matrix_infinite_bool
1170 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1171 \if_int_compare:w ##1 = \l__physicx_matrix_rows_int
1172 \scan_stop:
1173 \else:
1174 % finish up the row
1175 \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1176 \fi:
1177 }
1178 % finish up the rows
1179 \bool_if:NT \l__physicx_matrix_dotrow_bool
1180 {
1181 % finish up the row
1182 \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1183 % if we have a row of dots, fill it in
1184 \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1185 \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1186 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1187 \bool_if:NT \l__physicx_matrix_dotcol_bool
1188 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
1189 \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1190 % fill the last row
1191 %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
1192 \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
1193 \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1194 {
1195     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
1196     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1197     \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
1198 }
1199 \bool_if:NT \l__physicx_matrix_dotcol_bool
1200 {
1201     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
1202     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1203     \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
1204 }
1205 % if the matrix is infinite, add a further column with \cdots
1206 \bool_if:NT \l__physicx_matrix_infinite_bool
1207 { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1208 }
1209 % if the matrix is infinite, add a final row
1210 \bool_if:NT \l__physicx_matrix_infinite_bool
1211 {
1212 % finish up the row
1213 \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1214 \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }

```



```

1215     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1216     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1217     \bool_if:NT \l__physicx_matrix_dotcol_bool
1218     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1219     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1220     % update cols
1221     \bool_if:NTF \l__physicx_matrix_dotcol_bool
1222     { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1223     { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1224   }
1225 }

```

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

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
1226 \cs_new:Npn \physicx_matrix_diag_parse:n #1
1227 {
1228   \keyval_parse:nnn
1229   \__physicx_matrix_diag_parse_aux:n
1230   \__physicx_matrix_diag_parse_aux:nn
1231   {#1}
1232 }
1233 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
1234 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
1235 {
1236   \str_case:e:nnF {#1}
1237   {
1238     { auto-update }
1239     {
1240       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
1241       \__physicx_matrix_calc:nn
1242     }
1243     { noauto-update }
1244     {
1245       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
1246     }
1247     { true }
1248     {
1249       \bool_set_true:N \l__physicx_matrix_diag_bool
1250       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1251       \__physicx_diagonalmatrix_set_diag:
1252     }
1253     { false }
1254     {
1255       \bool_set_false:N \l__physicx_matrix_diag_bool
1256       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1257       \__physicx_diagonalmatrix_no_diag:
1258     }
1259   }
1260   { \msg_error:nnn { physicx } { diag-key } {#1} }
1261 }
1262 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
1263 {
1264   \tl_set:Nn \l__physicx_tmpdiag_tl {#2}

```

```

1265 \tl_set:Nx \l__physicx_tmpdiag_tl
1266 { \__physicx_expand:w \l__physicx_tmpdiag_tl }
1267 \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl
1268 \tl_if_head_eq_charcode:nNTF {#1} '
1269 {
1270   \exp_args:Nf \__physicx_matrix_diag_parse_aux_anti:n
1271   { \tl_tail:n {#1} }
1272 }
1273 { \__physicx_matrix_diag_parse_aux_regu:n {#1} }
1274 }
1275 \cs_new:Npn \__physicx_diagonalmatrix_set_diag:
1276 {
1277   \int_zero:N \l__physicx_matrix_cols_int
1278   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1279   {
1280     \int_incr:N \l__physicx_matrix_cols_int
1281     \physicx_matrix_set_r_c:nnn {##1} {##1} {##2}
1282   }
1283   \int_set_eq:NN \l__physicx_matrix_rows_int
1284   \l__physicx_matrix_cols_int
1285 }
1286 \cs_new:Npn \__physicx_diagonalmatrix_no_diag:
1287 {
1288   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1289   { \physicx_matrix_set_r_c:nnn {##1} {##1} {##2} }
1290   \__physicx_matrix_diag_calc:nn
1291   { \seq_count:N \l__physicx_tmpdiag_seq }
1292   { \seq_count:N \l__physicx_tmpdiag_seq }
1293 }
1294 \cs_new_eq:NN \__physicx_diagonalmatrix_diag_main:
1295 \__physicx_diagonalmatrix_no_diag:
1296 \cs_new:Npn \__physicx_matrix_diag_parse_aux_regu:n #1
1297 {
1298   \if_int_compare:w #1 = 0 \exp_stop_f:
1299   \__physicx_diagonalmatrix_diag_main:
1300   \else:
1301     \if_int_compare:w #1 > 0 \exp_stop_f:
1302     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1303     {
1304       \physicx_matrix_set_r_c:nnn
1305       {##1} { \int_eval:n { ##1 + #1 } } {##2}
1306     }
1307     \__physicx_matrix_diag_calc:nn
1308     { \seq_count:N \l__physicx_tmpdiag_seq }
1309     { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1310   \else:
1311     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1312     {
1313       \physicx_matrix_set_r_c:nnn
1314       { \int_eval:n { ##1 - #1 } } {##1} {##2}
1315     }
1316     \__physicx_matrix_diag_calc:nn
1317     { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1318     { \seq_count:N \l__physicx_tmpdiag_seq }

```

```

1319     \fi:
1320     \fi:
1321   }
1322   \cs_new:Npn \__physicx_matrix_diag_parse_aux_anti:n #1
1323   {
1324     \if_int_compare:w #1 = 0 \exp_stop_f:
1325       \__physicx_matrix_diag_calc:nn
1326       { \seq_count:N \l__physicx_tmpdiag_seq }
1327       { \seq_count:N \l__physicx_tmpdiag_seq }
1328       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1329       {
1330         \physicx_matrix_set_r_c:nnn
1331         {##1}
1332         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1333         {##2}
1334       }
1335     \else:
1336       \if_int_compare:w #1 > 0 \exp_stop_f:
1337         \__physicx_matrix_diag_calc:nn
1338         { \seq_count:N \l__physicx_tmpdiag_seq }
1339         { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1340         \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1341         {
1342           \physicx_matrix_set_r_c:nnn
1343           {##1}
1344           { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1345           {##2}
1346         }
1347       \else:
1348         \__physicx_matrix_diag_calc:nn
1349         { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1350         { \seq_count:N \l__physicx_tmpdiag_seq }
1351         \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1352         {
1353           \physicx_matrix_set_r_c:nnn
1354           { \int_eval:n { ##1 - #1 } }
1355           { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1356           {##2}
1357         }
1358     \fi:
1359     \fi:
1360   }
1361   \cs_new:Npn \__physicx_matrix_diag_calc:nn
1362   { \__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
1363 \cs_new:Npn \physicx_matrix_item_parse:n #1
1364 {
1365   \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1366   \keyval_parse:NNn
1367   \__physicx_matrix_item_parse_aux:n
1368   \__physicx_matrix_item_parse_aux:nn

```

```

1369     {#1}
1370   }
1371   \cs_generate_variant:Nn \physicsx_matrix_item_parse:n { o }
1372   \cs_new:Npn \__physicsx_matrix_item_parse_aux:n #1 { }
1373   \cs_new:Npn \__physicsx_matrix_item_parse_aux:nn #1#2
1374   {
1375     \tl_set:Nn \l__physicsx_tmpitem_tl {#2}
1376     \tl_set:Nx \l__physicsx_tmpitem_tl
1377       { \__physicsx_expand:w \l__physicsx_tmpitem_tl }
1378     \physicsx_parse_range:neN \l__physicsx_matrix_rows_int
1379     { \use_i:nn #1 } \l__physicsx_tmp_rownum_seq
1380     \physicsx_parse_range:neN \l__physicsx_matrix_cols_int
1381     { \use_ii:nn #1 } \l__physicsx_tmp_colnum_seq
1382     \exp_args:No \tl_if_eq:nnTF
1383     { \l__physicsx_tmpitem_tl } { \PHYSICXIGNORE }
1384     {
1385       \seq_map_inline:Nn \l__physicsx_tmp_rownum_seq
1386       {
1387         \seq_map_inline:Nn \l__physicsx_tmp_colnum_seq
1388         {
1389           \clist_put_right:Nn \l__physicsx_item_ignore_clist { [##1][####1] }
1390         }
1391       }
1392     }
1393     {
1394       \seq_map_inline:Nn \l__physicsx_tmp_rownum_seq
1395       {
1396         \seq_map_inline:Nn \l__physicsx_tmp_colnum_seq
1397         {
1398           \clist_if_in:NnF \l__physicsx_item_ignore_clist { [##1][####1] }
1399           {
1400             \exp_args:Nnno \physicsx_matrix_set_r_c:nnn
1401             {##1} {####1} { \l__physicsx_tmpitem_tl }
1402           }
1403         }
1404       }
1405     }
1406   }

```

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

\physicsx\_matrix\_array\_parse:n Parse ‘array...’ keys.

```

\physicsx_matrix_array_parse:o
1407   \cs_new:Npn \physicsx_matrix_array_parse:n #1
1408   {
1409     \tl_set:Nn \l__physicsx_tmparr_tl {#1}
1410     \tl_set:Nx \l__physicsx_tmparr_tl
1411       { \__physicsx_expand:w \l__physicsx_tmparr_tl }
1412     \seq_set_split:NVV \l__physicsx_matrix_tmparr_r_sep \physicsx@cr \l__physicsx_tmparr_tl
1413     \__physicsx_matrix_autocalc:nn
1414     { \seq_count:N \l__physicsx_matrix_tmparr_r_sep }
1415     { 0 }
1416     \seq_map_indexed_inline:Nn \l__physicsx_matrix_tmparr_r_sep
1417     {
1418       \seq_set_split:Non \l__physicsx_matrix_tmparr_c_sep { \physicsx@align } {##2}

```

```

1419     \__physicx_matrix_autocalc:nn
1420     { 0 }
1421     { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1422     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1423     {
1424         \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1425     }
1426 }
1427 }
1428 \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.

```

1429 \cs_new:Npn \physicx_matrix_array_parse_main:
1430 {
1431     \int_step_inline:nn \l__physicx_matrix_rows_int
1432     {
1433         \int_step_inline:nn \l__physicx_matrix_cols_int
1434         {
1435             \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1436             {##1} {####1} \l__physicx_matrix_main_tl
1437         }
1438     }
1439 }

```

(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).

```

1440 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1441 {
1442     \physicx_if_num:nTF {#1}
1443     { \prg_return_true: }
1444     {
1445         \bool_case_true:nTF
1446         {
1447             { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1448             { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1449             {
1450                 \bool_lazy_and_p:nn
1451                 { \cs_if_exist_p:N \inteval }
1452                 { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1453             } { }
1454             {
1455                 \bool_lazy_and_p:nn
1456                 { \cs_if_exist_p:N \fpeval }
1457                 { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1458             } { }
1459         }
1460         { \prg_return_true: }
1461         { \prg_return_false: }
1462     }
1463 }

```

(End definition for `\_physicx_if_can_num:n`.)

`\diagonalmatrix` Define `\diagonalmatrix`.

```

1464 \DeclareDocumentCommand \diagonalmatrix { t= t+ 0{} m }
1465 {
1466   \group_begin:
1467   \IfBooleanTF {#1}
1468     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1469     { \keys_set:nn { physicx/matrix } { #3 } }
1470   \physicx_construct:nnn { }
1471   {
1472     \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist
1473     \tl_if_empty:nF {#4}
1474     {
1475       \__physicx_if_keyval:nTF {#4}
1476       { \physicx_matrix_diag_parse:n { true, #4 } }
1477       { \physicx_matrix_diag_parse:n { true, 0 = {#4} } }
1478     }
1479   }
1480   { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1481   \bool_lazy_or:nnTF
1482     { \bool_if_p:n {#2} }
1483     { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1484     {
1485       \bool_if:NTF \l__physicx_matrix_expand_element_bool
1486       {
1487         \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1488         \__physicx_matrix_appto_body_e:off
1489       }
1490       {
1491         \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1492         \__physicx_matrix_appto_body_ne:off
1493       }
1494       \use_i_ii:nnn
1495     }
1496     { \use_i:nn }
1497     \__physicx_matrix_transpose:N
1498     \__physicx_diagonalmatrix_generate_enhanced_body:NNN
1499     \__physicx_diagonalmatrix_generate_body:NNN
1500     \__physicx_matrix_save_or_print:
1501     \group_end:
1502   }
1503   \cs_new:Npn \__physicx_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1504   {
1505     \__physicx_matrix_generate_body:NNNN #1#2#3
1506     \__physicx_diagonalmatrix_enhanced:nnn
1507   }
1508   \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1509   {
1510     \int_step_inline:nn { #1 - 1 }
1511     {
1512       \int_step_inline:nn { #2 - 1 }
1513       {
1514         \tl_put_right:Nx \l__physicx_matrix_body_tl

```

```

1515         {
1516             \exp_after:wN
1517             \physicx_matrix_use_r_c:nn
1518             #3 {{##1}} {{####1}} &
1519         }
1520     }
1521     \tl_put_right:Nx \l__physicx_matrix_body_tl
1522     {
1523         \exp_after:wN
1524         \physicx_matrix_use_r_c:nn
1525         #3 {{##1}} {{ \int_use:N #2 }} \__physicx_matrix_sep:
1526     }
1527 }
1528 \int_step_inline:nn { #2 - 1 }
1529 {
1530     \tl_put_right:Nx \l__physicx_matrix_body_tl
1531     {
1532         \exp_after:wN
1533         \physicx_matrix_use_r_c:nn
1534         #3 {{ \int_use:N #1 }} {{##1}} &
1535     }
1536 }
1537 \tl_put_right:Nx \l__physicx_matrix_body_tl
1538 {
1539     \exp_after:wN
1540     \physicx_matrix_use_r_c:nn
1541     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1542 }
1543 }

```

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

\\_\_physicx\_declare\_init:

```

1544 \cs_new:Npn \__physicx_matrix_enhanced_init:
1545 {
1546     \seq_if_empty:NF \l__physicx_row_list_seq
1547     {
1548         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1549         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1550             { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1551     }
1552     \seq_if_empty:NF \l__physicx_col_list_seq
1553     {
1554         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1555         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1556             { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1557     }
1558 }

```

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

\commamatrix Define \commamatrix.

```

1559 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1560 {
1561     \group_begin:

```

```

1562 \keys_set:nn { physicx/matrix } {#3}
1563 \tl_if_empty:nF {#4}
1564 { \keys_set:nn { physicx/matrix } { array = {#4} } }
1565 \IfBooleanT {#1}
1566 { \keys_set:nn { physicx/matrix } { saveto = \physicxtmp } }
1567 \tl_set:Nx \l__physicx_matrix_array_tl
1568 { \__physicx_expand:w \l__physicx_matrix_array_tl }
1569 \bool_lazy_or:nnTF
1570 { \bool_if_p:n {#2} }
1571 { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1572 { \__physicx_commmatrix_enhanced: }
1573 {
1574   \tl_replace_all:Nox \l__physicx_matrix_array_tl
1575   { \physicx@cr } { \__physicx_matrix_sep: }
1576   \tl_replace_all:Non \l__physicx_matrix_array_tl
1577   { \physicx@align } { & }
1578   \tl_set_eq:NN \l__physicx_matrix_body_tl
1579   \l__physicx_matrix_array_tl
1580 }
1581 \__physicx_matrix_save_or_print:
1582 \group_end:
1583 }
1584 \cs_new_nopar:Npn \__physicx_matrix_save_or_print:
1585 {
1586   \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1587   {
1588     \exp_after:wN \tl_gset_eq:NN
1589     \l__physicx_matrix_save_tl
1590     \l__physicx_matrix_body_tl
1591   }
1592   {
1593     \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1594     \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1595     \fi:
1596     \exp_after:wN \__physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matri
1597     \l__physicx_matrix_body_tl
1598     \__physicx_matrix_end: \l__physicx_matrix_after_end_tl
1599   }
1600 }
1601 \cs_new:Npn \__physicx_commmatrix_enhanced:
1602 {
1603   \tl_clear:N \l__physicx_matrix_body_tl
1604   \int_zero:N \l__physicx_tmpa_int
1605   \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1606   \l__physicx_matrix_array_tl
1607   \int_set:Nn \l__physicx_matrix_rows_int
1608   { \seq_count:N \l__physicx_tmp_seq }
1609   \__physicx_matrix_enhanced_init:
1610   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1611   {
1612     \seq_map_tokens:Nn \l__physicx_tmp_seq
1613     {
1614       \int_incr:N \l__physicx_tmpa_int
1615       \exp_args:NV \__physicx_commmatrix_enhanced_aux:nNn

```



```

1616         \l__physicx_tmpa_int \l__physicx_commamatrix_enhanced_aux_e:nnn
1617     }
1618 }
1619 {
1620     \seq_map_tokens:Nn \l__physicx_tmp_seq
1621     {
1622         \int_incr:N \l__physicx_tmpa_int
1623         \exp_args:NV \l__physicx_commamatrix_enhanced_aux:nNn
1624         \l__physicx_tmpa_int \l__physicx_commamatrix_enhanced_aux_ne:nnn
1625     }
1626 }
1627 }
1628 \cs_new:Npn \l__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1629 {
1630     \seq_set_split:Non \l__physicx_tmp_col_seq
1631     { \physicx@align } {#3}
1632     \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1633     \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1634     { #2 {##2} {#1} {##1} }
1635     \tl_put_right:Nx \l__physicx_matrix_body_tl
1636     {
1637         \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1638         \if_int_compare:w \l__physicx_matrix_rows_int = #1
1639             \scan_stop:
1640         \else:
1641             \l__physicx_matrix_sep:
1642         \fi:
1643     }
1644 }
1645 \cs_new:Npn \l__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1646 {
1647     \seq_put_right:Nx \l__physicx_tmp_coled_seq
1648     {
1649         \text_expand:n % \text_expand:n do the magic thing, but slower
1650         {
1651             \physicx@matricelement { #1 }
1652             { \l__physicx_matrix_row_iterate:n {#2} }
1653             { \l__physicx_matrix_col_iterate:n {#3} }
1654         }
1655     }
1656 }
1657 \cs_new:Npn \l__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1658 {
1659     \seq_put_right:No \l__physicx_tmp_coled_seq
1660     {
1661         \physicx@matricelement {#1}
1662         { \l__physicx_matrix_row_iterate:n {#2} }
1663         { \l__physicx_matrix_col_iterate:n {#3} }
1664     }
1665 }

```

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

\generalmatrix Define \generalmatrix.

```

1666 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1667 {
1668   \IfBooleanTF {#2}
1669   {
1670     \group_begin:
1671     \IfBooleanTF {#1}
1672     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicxtmp } }
1673     { \keys_set:nn { physicx/matrix } {#4} }
1674     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1675     \physicx_construct:nnn
1676     {
1677       \tl_if_empty:NTF \l__physicx_matrix_main_tl
1678       {
1679         \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1680       }
1681       { \physicx_matrix_array_parse_main: }
1682     }
1683     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1684     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1685     \__physicx_generalmatrix:
1686     \__physicx_matrix_save_or_print:
1687     \group_end:
1688   }
1689   {
1690     \IfBooleanTF {#1}
1691     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1692     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1693     \qxmatrix = * [#4]
1694   }
1695 }
1696 \cs_new:Npn \__physicx_generalmatrix:
1697 {
1698   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1699   {
1700     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1701     \__physicx_matrix_appto_body_e:off
1702   }
1703   {
1704     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1705     \__physicx_matrix_appto_body_ne:off
1706   }
1707   \__physicx_matrix_transpose:N
1708   \__physicx_matrix_generate_body:NNNN
1709   \__physicx_generalmatrix_generate:nnn
1710 }

```

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

\\_\_physicx\_matrix\_generate\_body:NNNN

```

1711 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1712 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1713 {
1714   \__physicx_matrix_enhanced_init:
1715   \int_step_inline:nn { #1 - 1 }

```

```

1716 {
1717   \int_step_inline:nn { #2 - 1 }
1718   {
1719     \tl_set:Nx \l__physicx_tmp_tl
1720     {
1721       \exp_after:wN
1722       \physicx_matrix_use_r_c:nn
1723       #3 {{##1}} {{####1}}
1724     }
1725     #4 \l__physicx_tmp_tl {##1} {####1}
1726     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1727   }
1728   \tl_set:Nx \l__physicx_tmp_tl
1729   {
1730     \exp_after:wN
1731     \physicx_matrix_use_r_c:nn
1732     #3 {{##1}} {{ \int_use:N #2 }}
1733   }
1734   #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1735   \tl_put_right:Nx \l__physicx_matrix_body_tl
1736   { \__physicx_matrix_sep: }
1737 }
1738 \int_step_inline:nn { #2 - 1 }
1739 {
1740   \tl_set:Nx \l__physicx_tmp_tl
1741   {
1742     \exp_after:wN
1743     \physicx_matrix_use_r_c:nn
1744     #3 {{ \int_use:N #1 }} {{##1}}
1745   }
1746   #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}
1747   \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1748 }
1749 \tl_set:Nx \l__physicx_tmp_tl
1750 {
1751   \exp_after:wN
1752   \physicx_matrix_use_r_c:nn
1753   #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1754 }
1755 #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1756 }

```

(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
1757 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1758 {
1759   \tl_put_right:Nx \l__physicx_matrix_body_tl
1760   {
1761     \text_expand:n
1762     {
1763       \physicx@matricelement {#1}
1764       { \__physicx_matrix_row_iterate:n {#2} }
1765       { \__physicx_matrix_col_iterate:n {#3} }

```

```

1766     }
1767   }
1768 }
1769 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1770 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1771 {
1772   \tl_put_right:No \l__physicx_matrix_body_tl
1773   {
1774     \physicx@matricelement {#1}
1775     { \__physicx_matrix_row_iterate:n {#2} }
1776     { \__physicx_matrix_col_iterate:n {#3} }
1777   }
1778 }
1779 \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

```

1780 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1781 {
1782   \bool_if:NTF \l__physicx_matrix_transpose_bool
1783   {
1784     #1
1785     \l__physicx_matrix_cols_int
1786     \l__physicx_matrix_rows_int
1787     \use_ii_i:nn
1788   }
1789   {
1790     #1
1791     \l__physicx_matrix_rows_int
1792     \l__physicx_matrix_cols_int
1793     \use:nn
1794   }
1795 }

```

(End definition for \\_\_physicx\_matrix\_transpose:N.)

\\_\_physicx\_matrix\_sep:

```

1796 \cs_new:Npn \__physicx_matrix_sep:
1797 {
1798   \dim_compare:nNnTF \l__physicx_matrix_sep_dim = \c_zero_dim
1799   { \ \ } { \ \ [ \dim_use:N \l__physicx_matrix_sep_dim ] }
1800 }

```

(End definition for \\_\_physicx\_matrix\_sep:.)

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

```

1801 \cs_new:Npn \physicx_construct:nnn #1#2#3
1802 {
1803   \l__physicx_matrix_beginning_tl
1804   \__physicx_adi:nnn {#1} {#2} {#3}
1805   \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1806   {
1807     \int_incr:N \l__physicx_matrix_cols_int

```

```

1808     \__physicx_matrix_last_aux_c:
1809     \int_incr:N \l__physicx_matrix_cols_int
1810   }
1811   \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1812   {
1813     \int_incr:N \l__physicx_matrix_rows_int
1814     \__physicx_matrix_last_aux_r:
1815     \int_incr:N \l__physicx_matrix_rows_int
1816   }
1817   \bool_lazy_or:nnF
1818   { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1819   { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1820   {
1821     \physicx_matrix_set_r_c:nnn
1822     { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1823     { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1824     { \ddots }
1825   }
1826   \bool_if:NT \l__physicx_matrix_infinite_bool
1827   {
1828     \int_incr:N \l__physicx_matrix_rows_int
1829     \int_incr:N \l__physicx_matrix_cols_int
1830     \__physicx_matrix_last_aux_c:
1831     \__physicx_matrix_last_aux_r:
1832     \physicx_matrix_set_r_c:nnn
1833     { \int_use:N \l__physicx_matrix_rows_int }
1834     { \int_use:N \l__physicx_matrix_cols_int }
1835     { \ddots }
1836   }
1837   \l__physicx_matrix_ending_tl
1838 }
1839 \cs_new:Npn \__physicx_matrix_last_aux_c:
1840 {
1841   \int_step_inline:nn \l__physicx_matrix_rows_int
1842   {
1843     \physicx_matrix_set_r_c:nnn
1844     {##1} { \int_use:N \l__physicx_matrix_cols_int }
1845     { \cdots }
1846   }
1847 }
1848 \cs_new:Npn \__physicx_matrix_last_aux_r:
1849 {
1850   \int_step_inline:nn \l__physicx_matrix_cols_int
1851   {
1852     \physicx_matrix_set_r_c:nnn
1853     { \int_use:N \l__physicx_matrix_rows_int } {##1}
1854     { \vdots }
1855   }
1856 }

```

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

### 1.3.3 Define new matrix command

```

1857 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
1858 {
1859   \NewDocumentCommand #2 { t+ m o o m m }
1860   {
1861     \IfBooleanTF {##1}
1862     {
1863       \IfNoValueTF {##3}
1864       { \newcommand ##2 { #1 + [##5] {##6} } }
1865       {
1866         \IfNoValueTF {##4}
1867         { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1868         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1869       }
1870     }
1871     {
1872       \IfNoValueTF {##3}
1873       { \newcommand ##2 { #1 [##5] {##6} } }
1874       {
1875         \IfNoValueTF {##4}
1876         { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1877         { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1878       }
1879     }
1880   }
1881   \NewDocumentCommand #3 { t+ m m m m }
1882   {
1883     \IfBooleanTF {##1}
1884     { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1885     { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1886   }
1887 }
1888 \__physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1889 \__physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1890 \NewDocumentCommand \newgeneralmatrix { t+ m o o m m }
1891 {
1892   \IfBooleanTF {#1}
1893   {
1894     \IfNoValueTF {#3}
1895     { \newcommand #2 { \generalmatrix + {#5} } }
1896     {
1897       \IfNoValueTF {#4}
1898       { \newcommand #2 [#3] { \generalmatrix + {#5} } }
1899       { \newcommand #2 [#3] [#4] { \generalmatrix + {#5} } }
1900     }
1901   }
1902   {
1903     \IfNoValueTF {#3}
1904     { \newcommand #2 { \generalmatrix {#5} } }
1905     {
1906       \IfNoValueTF {#4}
1907       { \newcommand #2 [#3] { \generalmatrix {#5} } }

```

```

1908         { \newcommand #2 [#3] [#4] { \generalmatrix {#5} } }
1909     }
1910 }
1911 }
1912 \NewDocumentCommand \NewGeneralMatrix { t+ m m m }
1913 {
1914     \IfBooleanTF {#1}
1915     { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1916     { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1917 }

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

1918 \end{package}

```

## Index

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

| Symbols             |       |   |
|---------------------|-------|---|
| \+                  | ..... | 182   |
| \-                  | ..... | 182   |
| \\                  | ..... | 57, 1799  |
| \{                  | ..... | 394, 624, 643, 661, 702, <u>727</u>   |
| \}                  | ..... | 394, 624, 645, 661, 702, <u>727</u>   |
| A                   |       |   |
| \A                  | ..... | 177, 182  |
| \abs                | ..... | <u>695</u>  |
| \absolutevalue      | ..... | <u>695</u>  |
| \acomm              | ..... | <u>695</u>  |
| \aftergroup         | ..... | 232   |
| \anticommutator     | ..... | <u>695</u>  |
| \AtBeginDocument    | ..... | 279, 294, 308   |
| B                   |       |   |
| \begin              | ..... | 399,<br>400, 401, 402, 403, 404, 405, 408,<br>409, 410, 411, 412, 413, 414, 415,<br>416, 417, 418, 419, 420, 421, 422,<br>423, 424, 644, 647, 651, 654, 655,<br>656, 661, 662, 666, 669, 670, 671,<br>1032, 1033, 1034, 1035, 1036, 1037,<br>1038, 1041, 1042, 1043, 1044, 1045,<br>1046, 1047, 1048, 1049, 1050, 1051,<br>1052, 1053, 1054, 1055, 1056, 1057 |
| \bgroup             | ..... | 231, 233  |
| \Big                | ..... | 677   |
| \big                | ..... | 677   |
| \Bigg               | ..... | 677   |
| \bigg               | ..... | 677   |
| \Biggl              | ..... | 684   |
| \biggl              | ..... | 683   |
| \Biggr              | ..... | 684   |
| \biggr              | ..... | 683   |
| \Bigl               | ..... | 682   |
| \bigl               | ..... | 681   |
| \Bigr               | ..... | 682   |
| \bigr               | ..... | 681   |
| \boldsymbol         | ..... | 221   |
| bool commands:      |       |   |
| \bool_case_false:n  | ..... | 594, 595, 602   |
| \bool_case_true:nTF | ..... | 679, 1445   |
| \bool_if:N          | ..... | 16, 21, 26, 85, 210, 295, 302,<br>608, 610, 614, 1107, 1162, 1169,<br>1179, 1187, 1199, 1206, 1210, 1217,<br>1221, 1485, 1610, 1698, 1782, 1826   |
| \bool_if_p:N        | ..... | 1483, 1571  |
| \bool_if_p:n        | ..... | 681, 682, 683, 684, 1482, 1570  |
| \bool_lazy_and:p:nn | ..... | 1450, 1455  |
| \bool_lazy_or:nnTF  | ..... | 31,<br>137, 152, 447, 452, 1481, 1569, 1817   |
| \bool_new:N         | ..... | 9, 10, 11, 12, 13, 51, 66, 510,<br>512, 786, 787, 788, 796, 810, 818, 819   |
| \bool_set:Nn        | ..... | 521, 522, 1674  |
| \bool_set_false:N   | ..... | 95, 198, 266, 1095, 1123, 1138, 1255  |
| \bool_set_inverse:N | ..... | 856, 982  |

|   |                                  |   |   |
|---|----------------------------------|---|---|
| $\backslash$ bool_set_true:N            | 140,                             | $\backslash$ cs_new_protected:Npn       | 67,                                     |
| 155, 165, 205, 262, 265, 1094, 1127,    |                                  | 72, 77, 90, 93, 215, 310, 314, 320, 326 |   |
| 1131, 1142, 1146, 1249, 1548, 1554      |                                  | $\backslash$ cs_new_protected_nopar:Npn | 1059, 1061, 1066                        |
| $\backslash$ c_false_bool               | 639, 659                         | $\backslash$ cs_set:Nn                  | 987, 988, 989, 990, 991, 992            |
| $\backslash$ c_true_bool                | 622, 632, 639, 659               | $\backslash$ cs_set:Npn                 | 114, 116, 124, 126, 278, 470,           |
| $\backslash$ Bqty                       | 695                              | 476, 480, 698, 708, 709, 725, 730,      |   |
| $\backslash$ bqty                       | 695                              | 731, 732, 862, 871, 880, 889, 898, 907  |   |
| <b>C</b>                                |                                  | $\backslash$ cs_set:Npx                 | 914                                     |
| $\backslash$ cdots                      | 882, 1161, 1164, 1165, 1168,     | $\backslash$ cs_set_eq:NN               | 69, 70, 74, 75, 221, 225,               |
| 1170, 1201, 1202, 1205, 1207, 1845      |                                  | 229, 290, 291, 295, 305, 778, 831,      |   |
| clist commands:                         |                                  | 833, 835, 838, 844, 846, 860, 869,      |   |
| $\backslash$ clist_clear:N              | 1063                             | 878, 887, 896, 905, 923, 926, 929,      |   |
| $\backslash$ clist_concat:NNN           | 1070                             | 932, 935, 1109, 1113, 1240, 1245,       |   |
| $\backslash$ clist_if_empty:nTF         | 237                              | 1250, 1256, 1487, 1491, 1700, 1704      |   |
| $\backslash$ clist_if_in:NnTF           | 1398                             | $\backslash$ cs_set_nopar:Npn           | 1549, 1555                              |
| $\backslash$ clist_map_break:n          | 191, 203                         | $\backslash$ cs_set_protected:Npn       | 586, 599, 618, 693                      |
| $\backslash$ clist_map_inline:nn        | 82, 187, 199, 1068               | $\backslash$ cs_to_str:N                | 312, 314, 485, 488, 493, 494, 531       |
| $\backslash$ clist_new:N                | 50, 794, 795                     | <b>D</b>                                |   |
| $\backslash$ clist_put_right:Nn         | 944, 950, 1389                   | $\backslash$ ddots                      | 882, 1188, 1219, 1824, 1835             |
| $\backslash$ clist_set:Nn               | 1060                             | $\backslash$ DeclareDocumentCommand     | 280,                                    |
| $\backslash$ clist_set_eq:NN            | 1365                             | 282, 284, 287, 288, 289, 296, 297,      |   |
| $\backslash$ c_empty_clist              | 1365                             | 591, 601, 677, 1090, 1464, 1559, 1666   |   |
| $\backslash$ comm                       | 725                              | $\backslash$ diagonalmatrix             | 30, 1464, 1888                          |
| $\backslash$ commamatrix                | 31, 1559, 1889                   | $\backslash$ diffd                      | 295                                     |
| $\backslash$ commutator                 | 695                              | dim commands:                           |   |
| $\backslash$ crossproduct               | 288                              | $\backslash$ dim_compare:nNnTF          | 1798                                    |
| cs commands:                            |                                  | $\backslash$ dim_new:N                  | 820                                     |
| $\backslash$ cs:w                       | 32, 33, 973                      | $\backslash$ dim_use:N                  | 1799                                    |
| $\backslash$ cs_end:                    | 32, 33, 749, 973                 | $\backslash$ c_zero_dim                 | 1798                                    |
| $\backslash$ cs_generate_variant:Nn     | 3, 4, 5, 6, 89, 92, 213,         | $\backslash$ div                        | 290, 291                                |
| 465, 605, 1233, 1371, 1428, 1769, 1779  |                                  | $\backslash$ divergence                 | 291                                     |
| $\backslash$ cs_gset_eq:NN              | 40, 41, 42                       | $\backslash$ divisionsymbol             | 290                                     |
| $\backslash$ cs_if_exist:NnTF           | 371                              | $\backslash$ dotproduct                 | 287                                     |
| $\backslash$ cs_if_exist_p:N            | 1451, 1456                       | <b>E</b>                                |   |
| $\backslash$ cs_if_exist_use:NnTF       | 37                               | $\backslash$ egroup                     | 232, 234                                |
| $\backslash$ cs_if_free:NnTF            | 312                              | else commands:                          |   |
| $\backslash$ cs_new:Npn                 | 7, 110, 135, 143, 146, 162, 173, | $\backslash$ else:                      | 751, 1173, 1300, 1310, 1335, 1347, 1640 |
| 185, 219, 223, 227, 231, 232, 233,      |                                  | $\backslash$ end                        | 399,                                    |
| 234, 235, 432, 444, 515, 525, 537,      |                                  | 400, 401, 402, 403, 404, 405, 408,      |   |
| 574, 581, 606, 675, 821, 1022, 1024,    |                                  | 409, 410, 411, 412, 413, 414, 415,      |   |
| 1226, 1234, 1262, 1275, 1286, 1296,     |                                  | 416, 417, 418, 419, 420, 421, 422,      |   |
| 1322, 1361, 1363, 1372, 1373, 1407,     |                                  | 423, 424, 644, 647, 651, 654, 655,      |   |
| 1429, 1503, 1508, 1544, 1601, 1628,     |                                  | 656, 661, 662, 666, 669, 670, 671,      |   |
| 1645, 1657, 1696, 1712, 1757, 1770,     |                                  | 1032, 1033, 1034, 1035, 1036, 1037,     |   |
| 1780, 1796, 1801, 1839, 1848, 1857      |                                  | 1038, 1041, 1042, 1043, 1044, 1045,     |   |
| $\backslash$ cs_new_eq:NN               | 145, 172, 780, 800, 809, 1294    | 1046, 1047, 1048, 1049, 1050, 1051,     |   |
| $\backslash$ cs_new_nopar:Npn           | 739, 747, 756, 758,              | 1052, 1053, 1054, 1055, 1056, 1057      |   |
| 763, 769, 802, 803, 806, 807, 808, 1584 |                                  |   |   |
| $\backslash$ cs_new_protected:Nn        | 1105                             |   |   |



|  |   |  |
|--|---|--|
| \eval                                      | 695   | 1093, 1098, 1467, 1668, 1671, 1690,  |
| \evaluated                                 | 620, 708  | 1691, 1692, 1861, 1883, 1892, 1914   |
| exp commands:                              |   |  |
| \exp_after:wN                              |   | \IfNoValueTF . . . . . 296, 297, 468, 474, 1863, 1866,<br>1516, 1523, 1532, 1539, 1586,<br>1588, 1596, 1721, 1730, 1742, 1751                    |
| \exp_args:Nc                               | 324, 484, 487   | \Im . . . . . 297  |
| \exp_args:Nf                               | 1270  | \imaginary . . . . . 297   |
| \exp_args:NNc                              | 323   | int commands:  |
| \exp_args:Nno                              | 1400, 1435  | \int_compare:nNnTF . . . . . 148, 150, 164   |
| \exp_args:NNx                              | 328   | \int_compare:nTF . . . . . 1121, 1136  |
| \exp_args:Nno                              | 189, 201  | \int_compare_p:nNn . . . . . 138, 139, 153, 154  |
| \exp_args:NNx                              | 490   | \int_eval:n . . . . . 29, 1305, 1314, 1332,<br>1344, 1354, 1355, 1447, 1822, 1823  |
| \exp_args:No                               | 900, 909, 1011, 1382  | \int_incr:N . . . . . 539, 1280, 1614, 1622,<br>1807, 1809, 1813, 1815, 1828, 1829   |
| \exp_args:NV                               | 1615, 1623  | \int_max:nn . . . . . 742, 744   |
| \exp_end:                                  | 8   | \int_new:N . . . . .<br>. . . . . 52, 53, 62, 63, 64, 65, 514, 791, 792  |
| \exp_not:N                                 | 8, 374, 492, 493, 494, 545, 914   | \int_set:Nn . . . . . 80, 81, 103, 106, 520,<br>741, 743, 1096, 1097, 1124, 1139, 1607   |
| \exp_not:n                                 |   | \int_set_eq:NN . . . . .<br>. . . . . 102, 105, 149, 151, 1283, 1594   |
| \exp_stop_f:                               |   | \int_step_inline:nn . . . . .<br>. . . . . 1149, 1431, 1433, 1510, 1512,<br>1528, 1715, 1717, 1738, 1841, 1850                                   |
| . . . . . 540, 547, 1298, 1301, 1324, 1336 |   | \int_step_inline:nnn . . . . .<br>. . . . . 157, 167, 1155, 1193   |
| F  |   | \int_use:N . . . . . 1525,<br>1534, 1541, 1732, 1734, 1744, 1746,<br>1753, 1755, 1833, 1834, 1844, 1853  |
| \fbraces                                   | 296, 297  | \int_zero:N . . . . . 1277, 1604   |
| fi commands:                               |   | \interval . . . . . 29, 1451, 1452   |
| \fi:                                       | 551,<br>566, 571, 572, 673, 734, 753, 1176,<br>1319, 1320, 1358, 1359, 1595, 1642 | K  |
| fp commands:                               |   | keys commands:   |
| \fp_eval:n                                 | 29, 1448  | \keys_define:nn . . . . .<br>. . . . . 216, 240, 345, 347, 824, 826  |
| \fpeval                                    | 29, 1456, 1457  | \keys_if_exist:nnTF . . . . . 189, 201   |
| G  |   | \l_keys_key_str . . . . . 189,<br>192, 201, 206, 367, 371, 374, 1011, 1014   |
| \generalmatrix                             | 33, 1666, 1895, 1898,<br>1899, 1904, 1907, 1908, 1915, 1916                       | \l_keys_path_str . . . . . 387, 1018   |
| group commands:                            |   | \keys_set:nn . . . . . 3, 192,<br>206, 334, 335, 346, 373, 426, 435,<br>825, 1013, 1076, 1099, 1100, 1468,<br>1469, 1562, 1564, 1566, 1672, 1673 |
| \group_begin:                              |   | keyval commands:   |
| . . . . . 434, 1092, 1466, 1561, 1670      |   | \keyval_parse:Nn . . . . . 1366  |
| \group_end:                                | 442, 1103, 1501, 1582, 1687   | \keyval_parse:nnn . . . . . 316, 1228  |
| H  |   | L  |
| \hat                                       | 285   | \langle . . . . . 398, 628, 655, 670   |
| hook commands:                             |   | \lbrace . . . . . 296, 297   |
| \hook_gput_code:nnn                        | 37, 304, 306  | \ldots . . . . . 882   |
| I  |   |  |
| if commands:                               |   |  |
| \if_bool:N                                 | 555, 567, 620, 695  |  |
| \if_case:w                                 | 547   |  |
| \if_cs_exist:w                             | 749   |  |
| \if_int_compare:w                          | 540, 1171,<br>1298, 1301, 1324, 1336, 1593, 1638                                  |  |
| \IfBooleanT                                | 643, 645, 1565  |  |
| \IfBooleanTF                               | 281, 283, 285, 333, 501,<br>593, 650, 652, 665, 667, 687, 1028,                   |  |

`\left` ..... 231, 428, 453, 643,  
           650, 655, 661, 662, 665, 669, 670, 671  
`\let` ..... 296, 297, 588  
`\letdif` ..... 307  
`\lgrou` ..... 650, 665  
`\lVert` ..... 712

## M

`\mathbf` ..... 281, 283, 285  
`\mathcal` ..... 217, 218  
`\mathclose` ..... 231, 233  
`\mathopen` ..... 231, 233  
`\matrixquantity` ..... 620  
`\mqty` ..... 695

msg commands:

`\msg_error:nnn` ..... 1260  
`\msg_error:nnnn` ..... 386, 1017  
`\msg_new:nnn` ..... 60  
`\msg_new:nnnn` ..... 54

## N

`\nabla` ..... 278, 305  
`\NewCommaMatrix` ..... 1857  
`\newcommamatrix` ..... 1857  
`\newcommand` ..... 471, 477, 481, 490,  
           1864, 1867, 1868, 1873, 1876, 1877,  
           1895, 1898, 1899, 1904, 1907, 1908  
`\NewDiagonalMatrix` ..... 1857  
`\newdiagonalmatrix` ..... 1857  
`\newdif` ..... 307  
`\NewDocumentCommand` ... 331, 391, 464,  
           466, 497, 499, 1026, 1859, 1881,  
           1884, 1885, 1890, 1912, 1915, 1916  
`\NewGeneralMatrix` ..... 1857  
`\newgeneralmatrix` ..... 1857  
`\NewXQuantity` ..... 432  
`\newxquantity` ..... 432  
`\nobreak` ..... 634, 635, 636  
`\norm` ..... 695

## O

`\OOrder` ..... 735  
`\oorder` ..... 695  
`\operatorname` ..... 296, 297  
 or commands:  
   `\or:` ..... 548, 549, 550  
`\order` ..... 695  
`\Ordersymbol` ..... 271  
`\ordersymbol` ..... 270

## P

`\PassOptionsToPackage` ..... 237  
`\pb` ..... 695  
 peek commands:  
   `\peek_charcode_ignore_spaces:NTF` 588

physics commands:

`\physics_bf:` 221, 225, 229, 281, 283, 285  
`\physics_compat:` ..... 14  
`\physics_compat:TF` .... 268, 293, 713  
`\physics_construct:nnn` .....  
           1470, 1675, 1801, 1801  
`\physics_declare_legacy_paren:NnnnNnN`  
   . 674, 675, 694, 699, 700, 701, 702,  
   704, 711, 715, 719, 722, 726, 735, 737  
`\physics_declare_legacy_quantity:nnNn`  
   .... 508, 525, 619, 621, 631, 638, 658  
`\physics_if_num:n` ..... 175  
`\physics_if_num:nTF` .....  
           1011, 1119, 1134, 1442  
`\physics_if_num_sign:n` ..... 180  
`\physics_left:` .... 231, 455, 589, 689  
`\physics_left:N` .....  
           233, 457, 681, 682, 683, 684  
`\physics_mathtools:` ..... 24  
`\physics_mathtools:TF` .... 406, 1039  
`\physics_matrix_array_parse:n` ...  
           1407, 1407, 1428, 1679  
`\physics_matrix_array_parse_-`  
   main: ..... 1429, 1429, 1681  
`\physics_matrix_diag_parse:n` ...  
           945, 1226,  
           1226, 1233, 1472, 1476, 1477, 1683  
`\physics_matrix_item_parse:n` ...  
   ... 951, 1363, 1363, 1371, 1480, 1684  
`\physics_matrix_new_type:nn` ....  
           1022, 1024, 1029  
`\physics_matrix_new_type:nnn` ...  
           1022, 1022, 1030  
`\physics_matrix_set_r_c:nnn` 780,  
   923, 926, 929, 932, 935, 1281, 1289,  
   1304, 1313, 1330, 1342, 1353, 1400,  
   1424, 1435, 1821, 1832, 1843, 1852  
`\physics_matrix_use_r_c:nn` .....  
           747, 1517, 1524,  
           1533, 1540, 1722, 1731, 1743, 1752  
`\physics_new_type:nnn` .....  
           215, 392, 1023, 1025  
`\physics_option_or:nn` ..... 29  
`\physics_option_or:nnTF` .....  
           696, 706, 724, 728  
`\c_physics_Order_tl` 218, 271, 716, 736  
`\c_physics_order_tl` 217, 270, 720, 738  
`\physics_parse_range:nnN` .....  
           90, 92, 1378, 1380  
`\physics_parse_range:nnnN` .....  
           62, 77, 89, 91  
`\physics_parse_range_check:` ....  
           62, 67, 955

|                                      |   |                                      |  |
|--------------------------------------|---|--------------------------------------|--|
| \physicx_parse_range_nocheck: ...    | 62, 72, 956                             | \__physicx_declare_legacy_-          | quantity_aux:NNnnn .. 530, 581, 605    |
| \physicx_qxmatrix:nnn ...            | 1101, 1105                              | \__physicx_declare_legacy_-          | quantity_aux:nw .... 528, 574, 579     |
| \physicx_right: ...                  | 232, 455, 589, 689                      | \__physicx_diagonalmatrix_diag_-     | main: .... 1250, 1256, 1294, 1299      |
| \physicx_right:N .....               | 234, 459, 681, 682, 683, 684            | \__physicx_diagonalmatrix_-          | enhanced:nnn .... 1487, 1491, 1506     |
| \physicx_search_also:nn .....        | 185, 196, 213, 214                      | \__physicx_diagonalmatrix_-          | generate_body:NNN .... 1499, 1508      |
| \physicx_search_also:nnTF .          | 380, 1003                               | \__physicx_diagonalmatrix_-          | generate_enhanced_body:NNN ..          |
| \physicx_set_parse_range_-           |   |                                      | ..... 1498, 1503                       |
| delimiter:n .....                    | 110, 134                                | \__physicx_diagonalmatrix_no_-       | diag: .... 1257, 1286, 1295            |
| \physicx_short: .....                | 19                                      | \__physicx_diagonalmatrix_set_-      | diag: .... 1251, 1275                  |
| \physicx_unimath: .....              | 44                                      | \__physicx_do_nothing: .....         | 115                                    |
| \physicx_unimath:TF ..               | 40, 41, 42, 277                         | \l__physicx_end_range_int ...        | 63,                                    |
| \physicx_use_amsymb_type: .          | 219, 276                                |                                      | 105, 106, 150, 151, 154, 158, 164, 168 |
| \physicx_use_uni_bf_type: ....       | 227                                     | \__physicx_expand:w ....             | 800, 831,                              |
| \physicx_use_uni_bfit_type: 223, 275 |   |                                      | 833, 835, 838, 1266, 1377, 1411, 1568  |
| \physicx_xquantity:nn .....          | 432, 432, 464, 486, 489, 502, 503       | \__physicx_fixdif:Nn .....           | 316, 326                               |
| physicx internal commands:           |   | \__physicx_fixdif:Nnn .              | 316, 320, 328                          |
| \__physicx_adi:nnn .....             | 821, 987, 988, 989, 990, 991, 992, 1804 | \g__physicx_fixdif_bool              | 256, 295, 302                          |
| \l__physicx_begin_range_int ....     |   | \__physicx_fixdif_list:N ...         | 307, 310                               |
|                                      | 62, 102,                                | \__physicx_generalmatrix: 1685, 1696 |  |
|                                      | 103, 148, 149, 153, 154, 158, 164, 168  | \__physicx_generalmatrix_-           | generate:nnn .... 1700, 1704, 1709     |
| \l__physicx_cmd_arg_int .....        | 514, 520, 539, 540, 547                 | \__physicx_if_can_num:n ..           | 1440, 1440                             |
| \l__physicx_cmd_arg_spec_tl ....     | 513, 519, 532, 541                      | \__physicx_if_keyval:nTF ..          | 173, 1475                              |
| \l__physicx_cmd_auto_body_bool ..    | 512, 522, 567, 610, 614                 | \l__physicx_invalid_range_bool ..    | 66, 85, 95, 140, 155, 165              |
| \l__physicx_cmd_auto_body_tl ...     | 511, 518, 534, 568, 569                 | \l__physicx_item_ignore_clist ...    | 1365, 1389, 1398                       |
| \l__physicx_cmd_noauto_body_bool     | 510, 521, 555, 608                      | \__physicx_loadpackage_options:nnn   | 235, 246, 248, 250                     |
| \l__physicx_cmd_noauto_body_tl ..    | 509, 517, 533, 556, 557                 | \g__physicx_mathtools_bool .....     | 9, 26, 265, 266                        |
| \l__physicx_col_list_seq .....       | 798, 852, 1552, 1556                    | \__physicx_matrix_add_data:nn ...    | 947, 953, 1064, 1066                   |
| \__physicx_commamatrix_enhanced:     | 1572, 1601                              | \l__physicx_matrix_after_begin_-     | tl .....                               |
| \__physicx_commamatrix_enhanced_-    | aux:nNn .....                           |                                      | 816, 963, 965, 1596                    |
| \__physicx_commamatrix_enhanced_-    | aux_e:nnn .....                         | \l__physicx_matrix_after_end_tl .    | 817, 966, 968, 1598                    |
| \__physicx_commamatrix_enhanced_-    | aux_ne:nnn .....                        | \__physicx_matrix_appto_body_-       | e:nnn .....                            |
| \g__physicx_compat_bool .....        | 11, 16, 242, 262                        |                                      | 1110, 1488, 1701, 1757, 1757, 1769     |
| \__physicx_declare_init: ....        | 1544                                    | \__physicx_matrix_appto_body_-       | ne:nnn .....                           |
| \__physicx_declare_init:nnn          | 515, 527                                |                                      | 1114, 1492, 1705, 1757, 1770, 1779     |
| \__physicx_declare_legacy_-          |   | \l__physicx_matrix_args_tl .....     | 815, 961, 962, 1596                    |
| quantity_aux:nnnn .....              | 537, 578                                |                                      |  |

|  |   |
|--|---|
| \l__physicx_matrix_array_tl . . . .      | \l__physicx_matrix_ending_tl . . .      |
| 789, 828, 1567,                          | 823, 996, 998, 1837                     |
| 1568, 1574, 1576, 1579, 1606, 1679       | \l__physicx_matrix_enhanced_bool        |
| \__physicx_matrix_autocalc:nn . . .      | 819, 979, 982, 1483, 1571               |
| 809, 844, 846, 1362, 1413, 1419          | \__physicx_matrix_enhanced_init:        |
| \__physicx_matrix_begin:w . . . . .      | 1544, 1609, 1714                        |
| 807, 958, 1596                           | \l__physicx_matrix_expand_-             |
| \l__physicx_matrix_beginning_tl .        | element_bool . . . . . 810,             |
| 822, 993, 995, 1803                      | 918, 1107, 1485, 1548, 1554, 1610, 1698 |
| \l__physicx_matrix_body_tl . . . . .     | \__physicx_matrix_generate_-            |
| 790, 1117, 1152,                         | body:NNNN . . 1505, 1708, 1711, 1712    |
| 1157, 1158, 1164, 1165, 1170, 1175,      | \l__physicx_matrix_infinite_bool        |
| 1182, 1184, 1186, 1188, 1189, 1191,      | 786, 853, 856, 1094,                    |
| 1195, 1196, 1201, 1202, 1207, 1213,      | 1095, 1169, 1206, 1210, 1674, 1826      |
| 1214, 1216, 1218, 1219, 1514, 1521,      | \l__physicx_matrix_item_clist . . .     |
| 1530, 1537, 1578, 1590, 1597, 1603,      | 795, 948, 950, 1480, 1684               |
| 1635, 1726, 1735, 1747, 1759, 1772       | \__physicx_matrix_item_parse_-          |
| \__physicx_matrix_calc:nn . . . . .      | aux:n . . . . . 1367, 1372              |
| 739, 844, 1241                           | \__physicx_matrix_item_parse_-          |
| \__physicx_matrix_col_iterate:n .        | aux:nn . . . . . 1368, 1373             |
| 806, 939, 1555, 1653, 1663, 1765, 1776   | \__physicx_matrix_last_aux_c: . . .     |
| \l__physicx_matrix_cols_int . . . .      | 1808, 1830, 1839                        |
| 743, 744,                                | \__physicx_matrix_last_aux_r: . . .     |
| 792, 841, 1097, 1136, 1139, 1155,        | 1814, 1831, 1848                        |
| 1185, 1193, 1215, 1222, 1223, 1277,      | \l__physicx_matrix_last_col_tl . .      |
| 1280, 1284, 1332, 1344, 1355, 1380,      | 805, 941, 1805, 1819                    |
| 1433, 1593, 1594, 1785, 1792, 1807,      | \l__physicx_matrix_last_row_tl . .      |
| 1809, 1823, 1829, 1834, 1844, 1850       | 804, 940, 1811, 1818                    |
| \l__physicx_matrix_diag_bool . . .       | \l__physicx_matrix_main_tl . . . . .    |
| 796, 1249, 1255                          | 793, 848, 1436, 1677                    |
| \__physicx_matrix_diag_calc:nn . .       | \__physicx_matrix_row_iterate:n .       |
| 1240, 1245, 1290,                        | 803, 938, 1549, 1652, 1662, 1764, 1775  |
| 1307, 1316, 1325, 1337, 1348, 1361       | \l__physicx_matrix_rows_int . . . .     |
| \l__physicx_matrix_diag_clist . . .      | 741, 742, 791, 840, 1096,               |
| 794, 942, 944, 1472, 1683                | 1121, 1124, 1149, 1171, 1283, 1378,     |
| \__physicx_matrix_diag_parse_-           | 1431, 1607, 1638, 1786, 1791, 1813,     |
| aux:n . . . . . 1229, 1234               | 1815, 1822, 1828, 1833, 1841, 1853      |
| \__physicx_matrix_diag_parse_-           | \__physicx_matrix_save_or_print:        |
| aux:nn . . . . . 1230, 1262              | 1102, 1500, 1581, 1584, 1686            |
| \__physicx_matrix_diag_parse_-           | \l__physicx_matrix_save_tl . . . . .    |
| aux_anti:n . . . . . 1270, 1322          | 971, 973, 1586, 1589                    |
| \__physicx_matrix_diag_parse_-           | \__physicx_matrix_sep: . . . . .        |
| aux_regu:n . . . . . 1273, 1296          | 1175, 1182, 1189, 1213,                 |
| \l__physicx_matrix_dotcol_bool . .       | 1525, 1575, 1641, 1736, 1796, 1796      |
| 788, 1138, 1142,                         | \l__physicx_matrix_sep_dim . . . . .    |
| 1146, 1162, 1187, 1199, 1217, 1221       | 820, 969, 1798, 1799                    |
| \l__physicx_matrix_dotrow_bool . .       | \__physicx_matrix_set_data:nn . . .     |
| 787, 1123, 1127, 1131, 1179              | 946, 952, 1061                          |
| \__physicx_matrix_element_-              | \__physicx_matrix_set_r_c_-             |
| aux:nnn . . . . 860, 866, 869, 875,      | ckall:nnn . . . . . 778, 936            |
| 878, 884, 887, 893, 896, 902, 905, 910   | \__physicx_matrix_set_r_c_-             |
| \__physicx_matrix_end: . . . . 959, 1598 | ckep:nnn . . . . . 763, 927             |
| \__physicx_matrix_end:w . . . . . 808    | \__physicx_matrix_set_r_c_-             |
|  | ckig:nnn . . . . . 758, 930             |

|   |   |
|---|---|
| \__physicx_matrix_set_r_c_-                                     | \l__physicx_quantity_right_tl ...                           |
| ckigep:nnn ..... 769, 779, 933                                  | ..... 344, 352, 441   |
| \__physicx_matrix_set_r_c_-                                     | \__physicx_qxmatrix_appto_-                                 |
| nock:nnn ..... 756, 781, 924                                    | body:nnn ..... 1109, 1113,                                  |
| \l__physicx_matrix_tmparr_c_sep .                               | 1153, 1159, 1166, 1192, 1197, 1203                          |
| ..... 1418, 1421, 1422  | \g__physicx_reqty_bool ..... 13, 252, 620, 695              |
| \l__physicx_matrix_tmparr_r_sep .                               | \l__physicx_row_list_seq ..... 797, 850, 1546, 1550         |
| ..... 1412, 1414, 1416  | \g__physicx_short_bool ... 12, 21, 244                      |
| \__physicx_matrix_transpose:N ...                               | \l__physicx_tmp_col_seq .. 1630, 1633                       |
| ..... 1497, 1707, 1780, 1780                                    | \l__physicx_tmp_coled_seq ..... 1632, 1637, 1647, 1659      |
| \l__physicx_matrix_transpose_-                                  | \l__physicx_tmp_colnum_seq ..... 1381, 1387, 1396           |
| bool ..... 818, 974, 1782                                       | \l__physicx_tmp_rownum_seq ..... 1379, 1385, 1394           |
| \l__physicx_max_range_int ..... 64, 81, 105, 138, 150, 151, 153 | \l__physicx_tmp_seq ..... 1605, 1608, 1612, 1620            |
| \l__physicx_min_range_int ..... 65, 80, 102, 139, 148, 149      | \l__physicx_tmp_tl ..... 542, 556, 568, 1719, 1725,         |
| \__physicx_nabla: ..... 305, 308                                | 1728, 1734, 1740, 1746, 1749, 1755                          |
| \__physicx_nauto_case:nnnn . 583, 606                           | \l__physicx_tmpa_bool 51, 198, 205, 210                     |
| \__physicx_new_matrix_cmd:NNN ...                               | \l__physicx_tmpa_clist ..... 50                             |
| ..... 1857, 1857, 1888, 1889                                    | \l__physicx_tmpa_int ..... 52, 1604, 1614, 1616, 1622, 1624 |
| \__physicx_new_xquantity_aux:w ..                               | \l__physicx_tmpa_seq ..... 86, 96, 141, 144, 159, 169       |
| ..... 470, 476, 480, 484, 487                                   | \l__physicx_tmpa_tl ..... 101, 103, 118, 119, 128, 367, 368 |
| \__physicx_parse_range_action:nnn                               | \l__physicx_tmparr_tl ..... 1409, 1410, 1411, 1412          |
| ..... 97, 114, 124  | \l__physicx_tmpb_int ..... 53                               |
| \__physicx_parse_range_aux:n 84, 93                             | \l__physicx_tmpb_tl ... 104, 106, 129                       |
| \__physicx_parse_range_aux:w ...                                | \l__physicx_tmpdiag_seq 1267, 1278,                         |
| ..... 115, 116, 125, 126  | 1288, 1291, 1292, 1302, 1308, 1309,                         |
| \__physicx_parse_range: ...                                     | 1311, 1317, 1318, 1326, 1327, 1328,                         |
| ..... 70, 75, 107, 172  | 1338, 1339, 1340, 1349, 1350, 1351                          |
| \__physicx_parse_range_range_-                                  | \l__physicx_tmpdiag_tl ..... 1264, 1265, 1266, 1267         |
| check: ..... 70, 146, 172                                       | \l__physicx_tmpitem_tl ..... 1375, 1376, 1377, 1383, 1401   |
| \__physicx_parse_range_range_-                                  | \__physicx_vnabla: .... 278, 289, 308                       |
| nocheck: ..... 75, 162  | \__physicx_xquantity_aux:nnnn ...                           |
| \__physicx_parse_range_single:n .                               | ..... 437, 444, 465   |
| ..... 69, 74, 99, 145   | \physicxempty . 752, 766, 774, 811, 812, 920                |
| \__physicx_parse_range_single_-                                 | \physicxexcept 814, 891, 900, 909, 915, 917                 |
| check:n ..... 69, 135, 145                                      | \PHYSICXIGNORE ..... 7, 8, 760, 771, 1383                   |
| \__physicx_parse_range_single_-                                 | \physicxset ..... 331                                       |
| nocheck:n ..... 74, 143   | \physicxtmp ... 508, 1099, 1468, 1566, 1672                 |
| \g__physicx_physics_bool ..... 10                               | \poissonbracket ..... 695                                   |
| \l__physicx_quantity_args_tl ...                                | \pqty ..... 695   |
| ..... 337, 359, 360, 439  |   |
| \l__physicx_quantity_code_tl ...                                |   |
| ..... 338, 361, 436, 440  |   |
| \l__physicx_quantity_left_size_-                                |   |
| tl ..... 339, 353, 448, 453, 457                                |   |
| \l__physicx_quantity_left_tl ...                                |   |
| ..... 340, 351, 438   |   |
| \l__physicx_quantity_post_tl ...                                |   |
| ..... 341, 350, 462   |   |
| \l__physicx_quantity_pre_tl ....                                |   |
| ..... 342, 349, 446   |   |
| \l__physicx_quantity_right_size_-                               |   |
| tl ..... 343, 354, 449, 454, 459                                |   |

|   |   |   |  |
|---|---|---|--|
| prg commands:                             |   | \seq_map_indexed_inline:Nn . . . . .  |  |
| \prg_generate_conditional_-               |   | . . . . .   | 1278, 1288, 1302, 1311,                |
| variant:Nnn . . . . .                     | 214                                     | . . . . .   | 1328, 1340, 1351, 1416, 1422, 1633     |
| \prg_new_conditional:Npnn . . . . .       |   | \seq_map_inline:Nn . . . . .  |  |
| . . . 14, 19, 24, 29, 175, 180, 196, 1440 |   | . . . . .   | 1385, 1387, 1394, 1396                 |
| \prg_replicate:nn . . . . .               | 1185, 1215                              | \seq_map_tokens:Nn . . . . .  | 1612, 1620                             |
| \prg_return_false: . . . . .              | 17,                                     | \seq_new:N . . . . .  | 797, 798                               |
| 22, 27, 35, 47, 178, 183, 211, 1461       |   | \seq_put_right:Nn . . . . .   |  |
| \prg_return_true: . . . . .               | 17,                                     | . . . . .   | 141, 144, 159, 169, 1647, 1659         |
| 22, 27, 34, 47, 178, 183, 211, 1443, 1460 |   | \seq_set_eq:NN . . . . .  | 79, 1632                               |
| \prg_set_conditional:Npnn . . . . .       | 44                                      | \seq_set_split:Nnn . . . . .  | 5,                                     |
| \ProcessKeysPackageOptions . . . . .      | 259                                     | . . . . .   | 850, 852, 1267, 1412, 1418, 1605, 1630 |
| <b>Q</b>                                  |   | \seq_use:Nn . . . . .   | 1637                                   |
| \qty . . . . .                            | 695                                     | \setmathfont . . . . .  | 286                                    |
| \quantity . . . . .                       | 620, 698                                | \setmatrixdata . . . . .  | 1059                                   |
| quark commands:                           |   | \setmatrixtype . . . . .  | 1000, 1022,                            |
| \q_nil . . . . .                          | 115                                     | . . . . .   | 1032, 1033, 1034, 1035, 1036, 1037,    |
| \q_physicsx_special . . . . .             | 115, 116, 125, 126                      | . . . . .   | 1038, 1041, 1042, 1043, 1044, 1045,    |
| \quark_if_nil:nTF . . . . .               | 120                                     | . . . . .   | 1046, 1047, 1048, 1049, 1050, 1051,    |
| \quark_if_recursion_tail_stop:n . . . . . | 576, 577                                | . . . . .   | 1052, 1053, 1054, 1055, 1056, 1057     |
| \q_recursion_stop . . . . .               | 529                                     | \setquantitytype . . . . .  | 364, 391, 393,                         |
| \q_recursion_tail . . . . .               | 529                                     | . . . . .   | 394, 395, 396, 397, 398, 399, 400,     |
| \qxmatrix . . . . .                       | 1081, 1693                              | . . . . .   | 401, 402, 403, 404, 405, 408, 409,     |
| \qxqty . . . . .                          | 506                                     | . . . . .   | 410, 411, 412, 413, 414, 415, 416,     |
| <b>R</b>                                  |   | . . . . .   | 417, 418, 419, 420, 421, 422, 423, 424 |
| \rangle . . . . .                         | 398, 628, 655, 670                      | \smallmatrixquantity . . . . .  | 620                                    |
| \rbrace . . . . .                         | 296, 297                                | \smqty . . . . .  | 695                                    |
| \Re . . . . .                             | 296                                     | str commands:   |  |
| \real . . . . .                           | 296                                     | \c_backslash_str . . . . .  | 368                                    |
| regex commands:                           |   | \str_case_e:nnTF . . . . .  | 1236                                   |
| \regex_match:nnTF . . . . .               | 177, 182, 900, 909                      | \str_const:Nn . . . . .   | 783, 784, 785                          |
| \renewdif . . . . .                       | 307, 308                                | \str_if_eq:nnTF . . . . .   | 561, 563                               |
| \RequirePackage . . . . .                 | 238                                     | \sympf . . . . .  | 229, 278, 308                          |
| \rgroup . . . . .                         | 652, 667                                | \sympfit . . . . .  | 225                                    |
| \right . . . . .                          | 232, 429, 454, 645,                     | <b>T</b>  |  |
| 652, 655, 661, 662, 667, 669, 670, 671    |   | T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands: |  |
| \rVert . . . . .                          | 712                                     | \@declareparencmd . . . . .   | 674                                    |
| <b>S</b>                                  |   | \@declarequantitycmd . . . . .  | 508                                    |
| \sb . . . . .                             | 802, 1152, 1157, 1164, 1191, 1195, 1201 | \@dif . . . . .   | 295                                    |
| scan commands:                            |   | \@ifpackageloaded . . . . .   | 261, 264, 274                          |
| \scan_stop: . . . . .                     | 1172, 1639                              | \@ifstar . . . . .  | 492                                    |
| seq commands:                             |   | \@pkgextension . . . . .  | 46                                     |
| \c_empty_seq . . . . .                    | 79, 1632                                | \c@MaxMatrixCols . . . . .  | 978, 1593, 1594                        |
| \seq_clear:N . . . . .                    | 96                                      | \physicsx@align . . . . .   |  |
| \seq_concat:NNN . . . . .                 | 86                                      | . . . . .   | 783, 984, 1418, 1577, 1631             |
| \seq_count:N . . . . .                    | 1291, 1292, 1308,                       | \physicsx@cr . . . . .  | 784, 983, 1412, 1575, 1605             |
| 1309, 1317, 1318, 1326, 1327, 1338,       |   | \physicsx@matricelement . . . . .   |  |
| 1339, 1349, 1350, 1414, 1421, 1608        |   | . . . . .   | 802, 857, 861, 862, 870,               |
| \seq_if_empty:NTF . . . . .               | 1546, 1552                              | . . . . .   | 871, 879, 880, 888, 889, 897, 898,     |
| \seq_item:Nn . . . . .                    | 1550, 1556                              | . . . . .   | 906, 907, 914, 1651, 1661, 1763, 1774  |
|   |   | \physicsx@sep . . . . .   | 785, 850, 852, 985, 1267               |
|   |   | tex commands:   |  |
|   |   | \tex_advance:D . . . . .  | 1222, 1223                             |

|                              |  |  |
|------------------------------|--|--|
| text commands:               |  |  |
| \text_expand:n               | 833, 1649, 1761  |  |
| tl commands:                 |  |  |
| \c_empty_tl                  | 356, 600   |  |
| \tl_clear:N                  | 517, 518, 519, 1117, 1603  |  |
| \tl_const:Nn                 | 217, 218   |  |
| \tl_gset_eq:NN               | 1588   |  |
| \tl_head:N                   | 367  |  |
| \tl_if_blank:nTF             | 130, 873   |  |
| \tl_if_empty:nTF             | 101, 104, 1677, 1805, 1811   |  |
| \tl_if_empty:nTF             | 112, 436, 765, 773, 864, 1473, 1563  |  |
| \tl_if_empty_p:N             | 448, 449, 1818, 1819   |  |
| \tl_if_eq:nnTF               | 760, 771, 1382   |  |
| \tl_if_exist:nTF             | 46   |  |
| \tl_if_head_eq_charcode:nNTF | 1268   |  |
| \tl_if_head_eq_meaning:nNTF  | 322, 329   |  |
| \tl_if_head_eq_meaning_p:nN  | 1447, 1448, 1452, 1457   |  |
| \tl_if_in:nnTF               | 174, 882, 891  |  |
| \tl_if_novalue_p:n           | 545  |  |
| \tl_map_function:nN          | 307  |  |
| \tl_new:N                    | 337, 338, 339, 340, 341, 342, 343, 344, 508, 509, 511, 513, 789, 790, 793, 804, 805, 812, 814, 815, 816, 817, 822, 823   |  |
| \tl_put_right:Nn             | 541, 556, 557, 568, 569, 917, 965, 968, 995, 998, 1152, 1157, 1158, 1164, 1165, 1170, 1175, 1182, 1184, 1186, 1188, 1189, 1191, 1195, 1196, 1201, 1202, 1207, 1213, 1214, 1216, 1218, 1219, 1514, 1521, 1530, 1537, 1635, 1726, 1735, 1747, 1759, 1772 |  |
| \tl_replace_all:Nnn          | 6, 1574, 1576  |  |
| \tl_set:Nn                   | 118, 119, 128, 129, 359, 367, 436, 542, 757, 761, 766, 767, 774, 775, 961, 973, 1264, 1265, 1375, 1376, 1409, 1410, 1567, 1719, 1728, 1740, 1749   |  |
| \tl_set_eq:NN                | 270, 271, 353, 354, 1578   |  |
| \tl_tail:N                   | 371, 374   |  |
| \tl_tail:n                   | 323, 329, 1271   |  |
| \tl_trim_spaces:n            | 118, 119, 128, 129   |  |
| token commands:              |  |  |
| \token_if_cs:nTF             | 1586   |  |
| \token_if_eq_meaning:nNTF    | 368  |  |
| \token_if_eq_meaning_p:NN    | 453, 454   |  |
| \TrimSpaces                  | 391, 1026  |  |
| \txqty                       | 507  |  |
| U                            |  |  |
| use commands:                |  |  |
| \use:n                       | 41, 369, 377, 838  |  |
| \use:nn                      | 1711, 1793   |  |
| \use:nnnn                    | 4  |  |
| \use_i:nn                    | 40, 584, 1379, 1496, 1692  |  |
| \use_i:nnn                   | 1692   |  |
| \use_i_ii:nnn                | 1494, 1691   |  |
| \use_ii:nn                   | 369, 584, 1381   |  |
| \use_ii_i:nn                 | 1711, 1787   |  |
| \use_none:n                  | 42, 375  |  |
| \use_none:nn                 | 809, 846, 1245   |  |
| \use_none:nnn                | 37   |  |
| V                            |  |  |
| \vdots                       | 882, 1184, 1186, 1188, 1214, 1216, 1218, 1854  |  |
| \vec                         | 283  |  |
| \vectimes                    | 288  |  |
| \vectorarrow                 | 282  |  |
| \vectorbold                  | 280  |  |
| \vectorunit                  | 284  |  |
| \Vert                        | 397, 629, 671  |  |
| \vert                        | 396, 627, 634, 635, 636, 669, 701, 705   |  |
| \vnabla                      | 289  |  |
| \vqty                        | 695  |  |
| \vysmbkcircle                | 287  |  |
| X                            |  |  |
| \xquantity                   | 464  |  |
| Z                            |  |  |
| \Z                           | 177, 182   |  |