

# The `physicx` package

Wenjian Chern (Longaster\*)

November 12, 2021, version v0.2

## Abstract

`physicx`

## 1 Implementation

```
1 <*package>
2 <@@=physicx>
3 \cs_generate_variant:Nn \keys_set:nn { nx , on , ox }
4 \cs_generate_variant:Nn \use:n nnn { nnno }
5 \cs_generate_variant:Nn \seq_set_split:Nnn { Non, NVV, c, cnV, cVV }
6 \cs_generate_variant:Nn \tl_replace_all:Nnn { Non, Nox }
7 \cs_new:Npn \PHYSICXIGNORE
8 { \exp_stop_f: \exp_not:N \PHYSICXIGNORE }
9 \bool_new:N \g__physicx_mathtools_bool
10 \bool_new:N \g__physicx_physics_bool
11 \bool_new:N \g__physicx_compat_bool
12 \bool_new:N \g__physicx_short_bool
13 \bool_new:N \g__physicx_reqty_bool
14 \prg_new_conditional:Npnn \physicx_compat: { T, F, TF }
15 {
16   \bool_if:NTF \g__physicx_compat_bool
17   { \prg_return_true: } { \prg_return_false: }
18 }
19 \prg_new_conditional:Npnn \physicx_short: { T, F, TF }
20 {
21   \bool_if:NTF \g__physicx_short_bool
22   { \prg_return_true: } { \prg_return_false: }
23 }
24 \prg_new_conditional:Npnn \physicx_mathtools: { T, F, TF }
25 {
26   \bool_if:NTF \g__physicx_mathtools_bool
27   { \prg_return_true: } { \prg_return_false: }
28 }
29 \prg_new_conditional:Npnn \physicx_option_or:nn #1#2 { T, F, TF }
30 {
31   \bool_lazy_or:nnTF
32   { \cs:w g__physicx_ #1 _bool \cs_end: }
33   { \cs:w g__physicx_ #2 _bool \cs_end: }
```

---

\*Email: [longaster@163.com](mailto:longaster@163.com)

```

34     { \prg_return_true: }
35     { \prg_return_false: }
36 }
37
38 \bool_new:N \l__physicx_tmpa_bool
39 \int_new:N \l__physicx_tmpa_int
40 \int_new:N \l__physicx_tmpb_int
41 \msg_new:nnnn { physicx } { unknown-key }
42 { The~key~'#1'~is~unknown~and~is~being~ignored. }
43 {
44     The~module~#2~does~not~have~a~key~called~#1.\\
45     Check~that~you~have~spelled~the~key~name~correctly.
46 }
47 \msg_new:nnn { physicx } { diag-key }
48 { The~value~'#1'~of~diag~key~is~unknown~and~is~being~ignored. }

```

## 1.1 Utils functions

```

\physicx_parse_range:nnnN Parse range, such as -3,6-8,9,10-.
\physicx_parse_range_check:
\physicx_parse_range_nocheck:
49 \int_new:N \l__physicx_begin_int
50 \int_new:N \l__physicx_end_int
51 \int_new:N \l__physicx_max_int
52 \int_new:N \l__physicx_min_int
53 \bool_new:N \l__physicx_invalid_range_bool
54 \cs_new_protected:Npn \physicx_parse_range_check:
55 {
56     \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_check:n
57     \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_check:
58 }
59 \cs_new_protected:Npn \physicx_parse_range_nocheck:
60 {
61     \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_nocheck:n
62     \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_nocheck:
63 }
64 \cs_new_protected:Npn \physicx_parse_range:nnnN #1#2#3#4
65 {
66     \seq_set_eq:NN #4 \c_empty_seq
67     \int_set:Nn \l__physicx_min_int {#1}
68     \int_set:Nn \l__physicx_max_int {#2}
69     \clist_map_inline:nn {#3}
70     {
71         \__physicx_parse_range_aux:n {##1}
72         \bool_if:NF \l__physicx_invalid_range_bool
73         { \seq_concat:NNN #4 #4 \l__physicx_tmpa_seq }
74     }
75 }
76 \cs_generate_variant:Nn \physicx_parse_range:nnnN { nnvN, nnxN }
77 \cs_new_protected:Npn \physicx_parse_range:nnN
78 { \physicx_parse_range:nnnN { 1 } }
79 \cs_generate_variant:Nn \physicx_parse_range:nnN { nvN, nxN }
80 \cs_new_protected:Npn \__physicx_parse_range_aux:n #1
81 {
82     \bool_set_false:N \l__physicx_invalid_range_bool
83     \seq_clear:N \l__physicx_tmpa_seq

```

```

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

```

```

138     {
139         \int_step_inline:nnn
140         { \l__physicx_begin_int } { \l__physicx_end_int }
141         { \seq_put_right:Nn \l__physicx_tmpa_seq {##1} }
142     }
143 }
144 \cs_new_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_check:

(End definition for \physicx_parse_range:nnnN, \physicx_parse_range_check:, and \physicx_parse_
range_nocheck:. These functions are documented on page ??.)

145 \cs_new:Npn \__physicx_if_keyval:nTF #1
146 { \tl_if_in:nnTF {#1} { = } }
147 \prg_new_conditional:Npnn \physicx_if_num:n #1 { T, F, TF }
148 {
149     \regex_match:nnTF { \A [[:digit:]]+ \Z } {#1}
150     { \prg_return_true: } { \prg_return_false: }
151 }
152 \prg_new_conditional:Npnn \physicx_if_num_sign:n #1 { T, F, TF }
153 {
154     \regex_match:nnTF { \A [\+|-]* [[:digit:]]+ \Z } {#1}
155     { \prg_return_true: } { \prg_return_false: }
156 }
157 \cs_new:Npn \physicx_search_also:nn #1#2
158 {
159     \clist_map_inline:nn {#1}
160     {
161         \exp_args:Nno \keys_if_exist:nnT {##1} { \l_keys_key_str }
162         {
163             \clist_map_break:n
164             { \keys_set:no {##1} { \l_keys_key_str = {#2} } }
165         }
166     }
167 }
168 \prg_new_conditional:Npnn \physicx_search_also:nn #1#2 { T, F, TF }
169 {
170     \bool_set_false:N \l__physicx_tmpa_bool
171     \clist_map_inline:nn {#1}
172     {
173         \exp_args:Nno \keys_if_exist:nnT {##1} { \l_keys_key_str }
174         {
175             \clist_map_break:n
176             {
177                 \bool_set_true:N \l__physicx_tmpa_bool
178                 \keys_set:no {##1} { \l_keys_key_str = {#2} }
179             }
180         }
181     }
182     \bool_if:NTF \l__physicx_tmpa_bool
183     { \prg_return_true: } { \prg_return_false: }
184 }
185 \cs_generate_variant:Nn \physicx_search_also:nn { no , oo }
186 \prg_generate_conditional_variant:Nnn \physicx_search_also:nn { no , oo } { T , F , TF }
187 \cs_new_protected:Npn \physicx_new_type:nnn #1#2#3
188 { \keys_define:nn { physicx/#1 } { type / #2 .meta:n = {#3} } }

```

```

189 \tl_const:Nn \c_physicsx_order_tl { \mathcal{o} }
190 \tl_const:Nn \c_physicsx_Order_tl { \mathcal{O} }
191 \cs_new:Npn \physicsx_use_amssymb_type:
192 {
193   \cs_set_eq:NN \physicsx_bf: \boldsymbol
194 }
195 \cs_new:Npn \physicsx_use_uni_bfit_type:
196 {
197   \cs_set_eq:NN \physicsx_bf: \symbfit
198 }
199 \cs_new:Npn \physicsx_use_uni_bf_type:
200 {
201   \cs_set_eq:NN \physicsx_bf: \symbf
202 }
203 \cs_new:Npn \physicsx_left: { \mathopen{}\mathclose\bgroup\left }
204 \cs_new:Npn \physicsx_right: { \aftergroup\egroup\right }
205 \cs_new:Npn \physicsx_left:N { \mathopen{}\mathclose\bgroup }
206 \cs_new:Npn \physicsx_right:N { \egroup }
207 \keys_define:nn { physicsx }
208 {
209   compat .bool_set:N = \g__physicsx_compat_bool ,
210   compat .default:n = true ,
211   short .bool_set:N = \g__physicsx_short_bool ,
212   short .default:n = true ,
213   physics .code:n = \RequirePackage{physics} ,
214   mathtools .code:n = \RequirePackage{mathtools} ,
215   unimath .code:n = \RequirePackage{unicode-math} ,
216   reqty .bool_set:N = \g__physicsx_reqty_bool ,
217   reqty .default:n = true ,
218   reqty .initial:n = true ,
219   noqty .meta:n = { reqty = false } ,
220 }
221 %
222 \ProcessKeysPackageOptions { physicsx }
223 %
224 \@ifpackageloaded{physics}
225 { \bool_set_true:N \g__physicsx_compat_bool }
226 { }
227 \@ifpackageloaded{mathtools}
228 { \bool_set_true:N \g__physicsx_mathtools_bool }
229 { \bool_set_false:N \g__physicsx_mathtools_bool }
230 %
231 \physicsx_compat:T
232 {
233   \tl_set_eq:NN \ordersymbol \c_physicsx_order_tl
234   \tl_set_eq:NN \Ordersymbol \c_physicsx_Order_tl
235 }
236 %
237 \@ifpackageloaded {unicode-math}
238 { \physicsx_use_uni_bfit_type: }
239 { \physicsx_use_amssymb_type: }

```

`\physicsxset` physics setup command.

```

240 \NewDocumentCommand \physicsxset { s m }

```

```

241 {
242   \IfBooleanTF {#1}
243   { \keys_set:nn { physicx/#2 } }
244   { \keys_set:nn { physicx } {#2} }
245 }

```

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

## 1.2 Quantity things

### 1.2.1 New quantity interfaces

```

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

```

```

289     }
290   } ,
291 }
292 \NewDocumentCommand \setquantitytype { >{ \TrimSpaces } m }
293 { \physicx_new_type:nnn { quantity } {#1} }
294 \setquantitytype { b } { left={[] , right={[]}} , }
295 \setquantitytype { B } { left={\{ } , right={\}} , }
296 \setquantitytype { p } { left={({ } , right={})} , }
297 \setquantitytype { v } { left=\vert , right=\vert , }
298 \setquantitytype { V } { left=\Vert , right=\Vert , }
299 \setquantitytype { a } { left=\langle , right=\rangle , }
300 \setquantitytype { m } { left=\begin{matrix} , right=\end{matrix} , noauto }
301 \setquantitytype { bm } { left=\begin{bmatrix} , right=\end{bmatrix} , noauto }
302 \setquantitytype { Bm } { left=\begin{Bmatrix} , right=\end{Bmatrix} , noauto }
303 \setquantitytype { pm } { left=\begin{pmatrix} , right=\end{pmatrix} , noauto }
304 \setquantitytype { vm } { left=\begin{vmatrix} , right=\end{vmatrix} , noauto }
305 \setquantitytype { Vm } { left=\begin{Vmatrix} , right=\end{Vmatrix} , noauto }
306 \setquantitytype { sm } { left=\begin{smallmatrix} , right=\end{smallmatrix} , noauto }
307 \physicx_mathtools:T
308 {
309   \setquantitytype { m* } { left=\begin{matrix*} , right=\end{matrix*} , noauto }
310   \setquantitytype { bm* } { left=\begin{bmatrix*} , right=\end{bmatrix*} , noauto }
311   \setquantitytype { Bm* } { left=\begin{Bmatrix*} , right=\end{Bmatrix*} , noauto }
312   \setquantitytype { pm* } { left=\begin{pmatrix*} , right=\end{pmatrix*} , noauto }
313   \setquantitytype { vm* } { left=\begin{vmatrix*} , right=\end{vmatrix*} , noauto }
314   \setquantitytype { Vm* } { left=\begin{Vmatrix*} , right=\end{Vmatrix*} , noauto }
315   \setquantitytype { sm* } { left=\begin{smallmatrix*} , right=\end{smallmatrix*} , noauto }
316   \setquantitytype { sbm } { left=\begin{bsmallmatrix} , right=\end{bsmallmatrix} , noauto }
317   \setquantitytype { sBm } { left=\begin{Bsmallmatrix} , right=\end{Bsmallmatrix} , noauto }
318   \setquantitytype { spm } { left=\begin{psmallmatrix} , right=\end{psmallmatrix} , noauto }
319   \setquantitytype { svm } { left=\begin{vsmallmatrix} , right=\end{vsmallmatrix} , noauto }
320   \setquantitytype { sVm } { left=\begin{Vsmallmatrix} , right=\end{Vsmallmatrix} , noauto }
321   \setquantitytype { sbm* } { left=\begin{bsmallmatrix*} , right=\end{bsmallmatrix*} , noauto }
322   \setquantitytype { sBm* } { left=\begin{Bsmallmatrix*} , right=\end{Bsmallmatrix*} , noauto }
323   \setquantitytype { spm* } { left=\begin{psmallmatrix*} , right=\end{psmallmatrix*} , noauto }
324   \setquantitytype { svm* } { left=\begin{vsmallmatrix*} , right=\end{vsmallmatrix*} , noauto }
325   \setquantitytype { sVm* } { left=\begin{Vsmallmatrix*} , right=\end{Vsmallmatrix*} , noauto }
326 }
327 \keys_set:nn { physicx/quantity }
328 {
329   left-size = \left ,
330   right-size = \right ,
331   type = p ,
332 }
\physicx_xquantity:nn
\newxquantity
\NewXQuantity
333 \cs_new:Npn \physicx_xquantity:nn #1#2
334 {
335   \group_begin:
336   \keys_set:nn { physicx/quantity } {#1}
337   \tl_if_empty:nF {#2} { \tl_set:Nn \l__physicx_quantity_code_tl {#2} }
338   \__physicx_xquantity_aux:oooo
339   { \l__physicx_quantity_left_tl }
340   { \l__physicx_quantity_args_tl }

```

```

341     { \l__physicx_quantity_code_tl }
342     { \l__physicx_quantity_right_tl }
343   \group_end:
344 }
345 \cs_new:Npn \__physicx_xquantity_aux:nnnn #1#2#3#4
346 {
347   \l__physicx_quantity_pre_tl
348   \bool_lazy_or:nnTF
349     { \tl_if_empty_p:N \l__physicx_quantity_left_size_tl }
350     { \tl_if_empty_p:N \l__physicx_quantity_right_size_tl }
351     { #1 #2 #3 #4 }
352   {
353     \bool_lazy_or:nnTF
354       { \token_if_eq_meaning_p:NN \l__physicx_quantity_left_size_tl \left }
355       { \token_if_eq_meaning_p:NN \l__physicx_quantity_right_size_tl \right }
356       { \physicx_left: #1 #2 #3 \physicx_right: #4 }
357     {
358       \physicx_left:N \l__physicx_quantity_left_size_tl #1 #2
359       #3
360       \physicx_right:N \l__physicx_quantity_right_size_tl #4
361     }
362   }
363   \l__physicx_quantity_post_tl
364 }
365 \NewDocumentCommand \xquantity { } { \physicx_xquantity:nn }
366 \cs_generate_variant:Nn \__physicx_xquantity_aux:nnnn { oooo }
367 \NewDocumentCommand \newxquantity { m o o m m }
368 {
369   \IfNoValueTF {#2}
370   {
371     \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
372       { \newcommand ##1 }
373   }
374   {
375     \IfNoValueTF {#3}
376     {
377       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
378         { \newcommand ##1 [#2] }
379     }
380     {
381       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
382         { \newcommand ##1 [#2] [#3] }
383     }
384   }
385   \exp_args:Nc \__physicx_new_xquantity_aux:w
386     { \cs_to_str:N #1~star }
387     { \physicx_xquantity:nn { #4 , noauto } {#5} }
388   \exp_args:Nc \__physicx_new_xquantity_aux:w
389     { \cs_to_str:N #1~unstar }
390     { \physicx_xquantity:nn { #4 } {#5} }
391   \exp_args:NNx \newcommand #1
392   {
393     \exp_not:N \@ifstar
394     \exp_not:c { \cs_to_str:N #1~star }

```



```

395         \exp_not:c { \cs_to_str:N #1~unstar }
396     }
397 }
398 \NewDocumentCommand \NewXQuantity { m m m m }
399 {
400     \NewDocumentCommand #1 { s #2 }
401     {
402         \IfBooleanTF {##1}
403         { \physics_xquantity:nn { #3 , noauto } {#4} }
404         { \physics_xquantity:nn { #3 } {#4} }
405     }
406 }
407 \NewXQuantity \qxqty { 0{ } m } { #2 } {#3}

```

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

## 1.2.2 Legacy quantity

```

\physics_declare_legacy_quantity:nnNn
  \@declarequantitycmd
408 \tl_new:N \physicx_tmp
409 \tl_new:N \l__physicx_cmd_noauto_body_tl
410 \bool_new:N \l__physicx_cmd_noauto_body_bool
411 \tl_new:N \l__physicx_cmd_auto_body_tl
412 \bool_new:N \l__physicx_cmd_auto_body_bool
413 \tl_new:N \l__physicx_cmd_arg_spec_tl
414 \int_new:N \l__physicx_cmd_arg_int
415 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
416 {
417     \tl_clear:N \l__physicx_cmd_noauto_body_tl
418     \tl_clear:N \l__physicx_cmd_auto_body_tl
419     \tl_clear:N \l__physicx_cmd_arg_spec_tl
420     \int_set:Nn \l__physicx_cmd_arg_int {#1}
421     \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
422     \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
423 }
424 % noauto, auto, cmd, body
425 \cs_new:Npn \physics_declare_legacy_quantity:nnNn #1#2#3#4
426 {
427     \__physicx_declare_init:nnn { 3 } {#1} {#2}
428     \__physicx_declare_legacy_quantity_aux:nw #4
429     \q_recursion_tail \q_recursion_tail \q_recursion_stop
430     \__physicx_declare_legacy_quantity_aux:NcVVV
431     #3 { \cs_to_str:N #3 ~ body }
432     \l__physicx_cmd_arg_spec_tl
433     \l__physicx_cmd_noauto_body_tl
434     \l__physicx_cmd_auto_body_tl
435 }
436 % arg spec, pre, body to replace(start from #4), post
437 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
438 {
439     \int_incr:N \l__physicx_cmd_arg_int
440     \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
441         \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}

```

```

442 \tl_set:Nx \l__physicx_tmp_tl
443 {
444   {
445     \exp_not:N \tl_if_novalue_p:n
446     {
447       \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:
448       \or: \or: \or:
449       \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
450       \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
451       \fi:
452     }
453   }
454 }
455 \if_bool:N \l__physicx_cmd_noauto_body_bool
456 \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
457 \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
458 {
459   {
460     % if is '.', use none
461     \str_if_eq:nnTF {#2} {.} {} {#2}
462     #3
463     \str_if_eq:nnTF {#4} {.} {} {#4}
464   }
465 }
466 \fi:
467 \if_bool:N \l__physicx_cmd_auto_body_bool
468 \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
469 \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
470 { { ##1 #2 #3 ##2 #4 } }
471 \fi:
472 \fi:
473 }
474 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
475 {
476   \quark_if_recursion_tail_stop:n {#1}
477   \quark_if_recursion_tail_stop:n {#2}
478   \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
479   \__physicx_declare_legacy_quantity_aux:nw
480 }
481 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
482 {
483   \__physicx_nauto_case:nnnn
484   { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
485   {
486     \cs_set_protected:Npn #1
487     {
488       \peek_charcode_ignore_spaces:NTF \let
489       { #2 } { #2 [ \physicx_left: ] \physicx_right: }
490     }
491     \DeclareDocumentCommand #2 { 0{##2} m s #3 }
492     {
493       \IfBooleanTF { ##3 }
494       { \bool_case_false:n {#4} }
495       { \bool_case_false:n {#5} }

```

```

496     }
497   }
498   {
499     \cs_set_protected:Npn #1
500     { #2 \c_empty_tl \c_empty_tl }
501     \DeclareDocumentCommand #2 { m m s #3 }
502     { \bool_case_false:n {#4} }
503   }
504 }
505 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
506 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
507 {
508   \bool_if:NTF \l__physicx_cmd_noauto_body_bool
509   {
510     \bool_if:NTF \l__physicx_cmd_auto_body_bool
511     {#1} {#2}
512   }
513   {
514     \bool_if:NTF \l__physicx_cmd_auto_body_bool
515     {#3} {#4}
516   }
517 }
518 \cs_set_protected:Npn \@declarequantitycmd
519 { \physicx_declare_legacy_quantity:nnNn }

```

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

|  |   |
|--|---|
| <code>\quantity</code><br><code>\evaluated</code><br><code>\matrixquantity</code><br><code>\smallmatrixquantity</code> | Redefine some macros in physics package.<br><pre> 520 \if_bool:N \g__physicx_reqty_bool 521 \physicx_declare_legacy_quantity:nnNn 522 \c_true_bool \c_true_bool \quantity 523 { 524   { !g } { { \{ } { #4 } { \} } } 525   { !o } { { [ } { #5 } { ] } } 526   { !d() } { { ( } { #6 } { ) } } 527   { !d   } { { \vert } { #7 } { \vert } } 528   { !d&lt;&gt; } { { \langle } { #8 } { \rangle } } 529   { !d== } { { \Vert } { #9 } { \Vert } } 530 } 531 \physicx_declare_legacy_quantity:nnNn 532 \c_true_bool \c_true_bool \evaluated 533 { 534   { !g } { { . } { #4 \nobreak } { \vert } } 535   { !d[ ] } { { [ } { #5 \nobreak } { \vert } } 536   { !d( ) } { { ( } { #6 \nobreak } { \vert } } 537 } 538 \physicx_declare_legacy_quantity:nnNn 539 \c_true_bool \c_false_bool \matrixquantity 540 { 541   { !g } 542   { 543     { \IfBooleanT{#3}{\left\{ } } 544     { \begin{matrix} #4 \end{matrix} } </pre> |
|--|---|

```

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

```

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

\physicsx\_declare\_legacy\_paren:NnnnNn

\@declareparenccmd

```

574 %% cmd, arg spec, replace(start from #6), pre, left, right, post
575 \cs_new:Npn \physicsx_declare_legacy_paren:NnnnNn #1#2#3#4#5#6#7
576 {
577   \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
578   {
579     \bool_case_true:nF
580     {
581       { \bool_if_p:n {##2} } { #4 \physicsx_left:N \bigl #5 #3 \physicsx_right:N \bigr
582       { \bool_if_p:n {##3} } { #4 \physicsx_left:N \Bigl #5 #3 \physicsx_right:N \Bigr
583       { \bool_if_p:n {##4} } { #4 \physicsx_left:N \biggl #5 #3 \physicsx_right:N \biggr
584       { \bool_if_p:n {##5} } { #4 \physicsx_left:N \Biggl #5 #3 \physicsx_right:N \Biggr
585     }
586     {
587       \IfBooleanTF {##1}
588       { #4 #5 #3 #6 #7 }
589       { #4 \physicsx_left: #5 #3 \physicsx_right: #6 #7 }
590     }
591   }
592 }
593 \cs_set_protected:Npn \@declareparenccmd
594 { \physicsx_declare_legacy_paren:NnnnNn }

```

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

```

\qty      Redefine some macros in physics package.
\mqty     595 \if_bool:N \g__physics_reqty_bool
\smqty    596 \physics_option_or:nnT { compat } { short }
\pqty     597 {
\bqty     598   \cs_set:Npn \qty { \quantity }
\qty     599   \physics_declare_legacy_paren:NnnnNNn \pqty { m } {#6} { } { } { } { }
\Bqty    600   \physics_declare_legacy_paren:NnnnNNn \bqty { m } {#6} { } { } [ ] { }
\absolutevalue 601   \physics_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } { } \vert \vert { }
\eval     602   \physics_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } { } \{ \} { }
\abs      603 }
\norm     604 \physics_declare_legacy_paren:NnnnNNn \absolutevalue
\order    605 { m } {#6} { } { } \vert \vert { }
\oorder   606 \physics_option_or:nnT { compat } { short }
\commutator 607 {
\poissonbracket 608   \cs_set:Npn \eval { \evaluated }
\pb       609   \cs_set:Npn \abs { \absolutevalue }
\anticommutator 610 }
\acomm    611 \physics_declare_legacy_paren:NnnnNNn \norm
\acomm    612 { m } {#6} { } { } \lVert \rVert { }
\acomm    613 \physics_compat:TF
\acomm    614 {
\acomm    615   \physics_declare_legacy_paren:NnnnNNn \order
\acomm    616   { m } {#6} { } { } \c_physicx_Order_t1 { } { } { }
\acomm    617 }
\acomm    618 {
\acomm    619   \physics_declare_legacy_paren:NnnnNNn \order
\acomm    620   { m } {#6} { } { } \c_physicx_order_t1 { } { } { }
\acomm    621 }
\acomm    622 \physics_declare_legacy_paren:NnnnNNn \commutator
\acomm    623 { m m } { #6 , #7 } { } { } [ ] { }
\acomm    624 \physics_option_or:nnT { compat } { short }
\acomm    625 { \cs_set:Npn \comm { \commutator } }
\acomm    626 \physics_declare_legacy_paren:NnnnNNn \poissonbracket
\acomm    627 { m m } { #6 , #7 } { } { } \{ \} { }
\acomm    628 \physics_option_or:nnT { compat } { short }
\acomm    629 {
\acomm    630   \cs_set:Npn \pb { \poissonbracket }
\acomm    631   \cs_set:Npn \anticommutator { \poissonbracket }
\acomm    632   \cs_set:Npn \acomm { \poissonbracket }
\acomm    633 }
\acomm    634 \fi:
\acomm    635 \physics_declare_legacy_paren:NnnnNNn \OOrder
\acomm    636 { m } {#6} { } { } \c_physicx_Order_t1 { } { } { }
\acomm    637 \physics_declare_legacy_paren:NnnnNNn \oorder
\acomm    638 { m } {#6} { } { } \c_physicx_order_t1 { } { } { }

```

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

## 1.3 Matrix things

### 1.3.1 Matrix auxillary functions

```

639 \cs_new_nopar:Npn \__physicx_matrix_calc:nn #1#2
640 {
641   \int_set:Nn \l__physicx_matrix_rows_int
642     { \int_max:nn {#1} \l__physicx_matrix_rows_int }
643   \int_set:Nn \l__physicx_matrix_cols_int
644     { \int_max:nn {#2} \l__physicx_matrix_cols_int }
645 }
646 % use matrix element
647 \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
648 {
649   \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
650   \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
651   \else:
652     \exp_not:o { \physicxempty }
653   \fi:
654 }
655 % set matrix element, check or not
656 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
657 { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
658 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
659 {
660   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
661   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
662 }
663 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_cke:nnn #1#2#3
664 {
665   \tl_if_empty:nTF {#3}
666     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
667     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
668 }
669 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
670 {
671   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
672   {
673     \tl_if_empty:nTF {#3}
674       { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
675       { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
676   }
677 }
678 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
679   \__physicx_matrix_set_r_c_ckigep:nnn
680 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
681   \__physicx_matrix_set_r_c_nock:nnn
682 % align, cr, sep symbol
683 \str_const:Nn \physicx@align { , }
684 \str_const:Nn \physicx@cr { ; }
685 \str_const:Nn \physicx@sep { , }
686 \bool_new:N \l__physicx_matrix_infinite_bool
687 \bool_new:N \l__physicx_matrix_dotrow_bool
688 \bool_new:N \l__physicx_matrix_dotcol_bool
689 \tl_new:N \l__physicx_matrix_array_tl
690 \tl_new:N \l__physicx_matrix_body_tl
691 \int_new:N \l__physicx_matrix_rows_int
692 \int_new:N \l__physicx_matrix_cols_int

```

```

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

```

### 1.3.2 Matrix keys

```

724 \keys_define:nn { physicx }
725 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
726 \keys_define:nn { physicx/matrix }
727 {
728   array .tl_set:N = \l__physicx_matrix_array_tl ,
729   expand .choice: ,
730   expand / none .code:n =
731     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
732   expand / text-expand .code:n =
733     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
734   expand / f .code:n =
735     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
736   expand / romanual .meta:n = { expand = f } ,
737   expand / x .code:n =
738     \cs_set_eq:NN \__physicx_expand:w \use:n ,
739   expand / edef .meta:n = { expand = x } ,
740   rows .int_set:N = \l__physicx_matrix_rows_int ,
741   cols .int_set:N = \l__physicx_matrix_cols_int ,
742   auto-update .choice: ,
743   auto-update / true .code:n =
744     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,

```

```

745 auto-update / false .code:n =
746   \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
747   auto-update .default:n = true ,
748   main .tl_set:N = \l__physicx_matrix_main_tl ,
749   row-list .code:n =
750     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,
751   col-list .code:n =
752     \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
753   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
754   infinite .default:n = true ,
755   !infinite .code:n =
756     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
757   element-code .cs_set:Np = \physicx@matricielement #1#2##3 ,
758   element-code* .choice: ,
759   element-code* / except-empty .code:n =
760     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
761       \physicx@matricielement
762     \cs_set:Npn \physicx@matricielement ##1##2##3
763       {
764         \tl_if_empty:nTF {##1}
765           {##1}
766           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
767       } ,
768   element-code* / except-dots .code:n =
769     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
770       \physicx@matricielement
771     \cs_set:Npn \physicx@matricielement ##1##2##3
772       {
773         \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
774           {##1}
775           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
776       } ,
777   element-code* / except-tl .code:n =
778     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
779       \physicx@matricielement
780     \cs_set:Npn \physicx@matricielement ##1##2##3
781       {
782         \tl_if_in:onTF { \physicxexcept } {##1}
783           {##1}
784           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
785       } ,
786   element-code* / except-regex .code:n =
787     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
788       \physicx@matricielement
789     \cs_set:Npn \physicx@matricielement ##1##2##3
790       {
791         \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
792           {##1}
793           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
794       } ,
795   element-code* / only-regex .code:n =
796     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
797       \physicx@matricielement
798     \cs_set:Npn \physicx@matricielement ##1##2##3

```



```

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

```

```

853 args* .tl_set:N = \l__physicx_matrix_args_tl ,
854 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
855 after-begin+ .code:n =
856   { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
857 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
858 after-end+ .code:n =
859   { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
860 sepdim .dim_set:N = \l__physicx_matrix_sep_dim ,
861 type .multichoice: ,
862 saveto .tl_set:N = \l__physicx_matrix_save_tl ,
863 saveto* .code:n =
864   \tl_set:N \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
865 transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
866 transpose .default:n = true ,
867 ' .meta:n = { transpose = true } ,
868 T .meta:n = { transpose = true } ,
869 MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
870 enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
871 enhanced .default:n = true ,
872 !enhanced .code:n =
873   \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
874 cr .tl_set:N = \physicx@cr ,
875 align .tl_set:N = \physicx@align ,
876 sep .tl_set:N = \physicx@sep ,
877 adi-order .choice: ,
878 adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
879 adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
880 adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
881 adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
882 adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
883 adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
884 beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
885 beginning+ .code:n =
886   \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
887 ending .tl_set:N = \l__physicx_matrix_ending_tl ,
888 ending+ .code:n =
889   \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
890
891 settype .code:n = \setmatrixtype #1 ,
892
893 unknown .code:n =
894   \physicx_search_also:nnF
895   {
896     physicx/matrix/type ,
897     physicx/matrix/expand ,
898     physicx/matrix/element-code* ,
899   }
900   {#1}
901   {
902     \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
903     {
904       \keys_set:nx { physicx/matrix }
905       { MaxMatrixCols = \l_keys_key_str }
906     }
907   }

```

```

907         {
908             \msg_error:nnxx { physicx } { unknown-key }
909             \l_keys_path_str { physicx/matrix }
910         }
911     } ,
912 }

```

\physicx\_matrix\_new\_type:nnn

```

913 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
914 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
915 \cs_new:Npn \physicx_matrix_new_type:nn
916 { \physicx_new_type:nnn { matrix } }
917 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
918 {
919     \IfBooleanTF {#1}
920     { \physicx_matrix_new_type:nn {#2} }
921     { \physicx_matrix_new_type:nnn {#2} }
922 }

```

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

A few types.

```

923 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
924 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
925 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
926 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
927 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
928 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
929 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
930 \physicx_mathtools:T
931 {
932     \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
933     \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
934     \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
935     \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
936     \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
937     \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
938     \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
939     \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
940     \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
941     \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
942     \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
943     \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
944     \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
945     \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
946     \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
947     \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
948     \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
949 }

```

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

```

950 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
951 { \clist_set:cn { physicx@ #1 data@ #2 } }
952 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2

```

```

953 {
954   \clist_clear:c { l__physicx_matrix_ #1 _clist }
955   \__physicx_matrix_add_data:nn {#1} {#2}
956 }
957 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
958 {
959   \clist_map_inline:nn {#2}
960   {
961     \clist_concat:ccc
962     { l__physicx_matrix_ #1 _clist }
963     { l__physicx_matrix_ #1 _clist }
964     { physicx@ #1 data@ #2 }
965   }
966 }

```

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

Initial settings.

```

967 \keys_set:nn { physicx/matrix }
968 {
969   type = m ,
970   saveto = ? ,
971 }

```

\qxmatrix

```

972 %% basicly, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
973   xmatmmn-in-the-physics-package, but changed some
974 % #1 = boolean, saveto matrix
975 % #2 = star, infinite
976 % #3 = options
977 % #4 = letter for the entries
978 % #5 = number of rows
979 % #6 = number of explicit rows, default = 3
980 % #7 = number of columns
981 % #8 = number of explicit columns, default = 3
982 \DeclareDocumentCommand \qxmatrix { t= s 0{type=p} m m 0{3} m 0{3} }
983 {
984   \group_begin:
985   \IfBooleanTF { #2 }
986   { \bool_set_true:N \l__physicx_matrix_infinite_bool }
987   { \bool_set_false:N \l__physicx_matrix_infinite_bool }
988   \int_set:Nn \l__physicx_matrix_rows_int {#6}
989   \int_set:Nn \l__physicx_matrix_cols_int {#8}
990   \IfBooleanTF {#1}
991   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
992   { \keys_set:nn { physicx/matrix } {#3} }
993   \physicx_qxmatrix:nnn {#4} {#5} {#7}
994   \__physicx_matrix_save_or_print:
995   \group_end:
996 }
997 \cs_new_protected:Nn \physicx_qxmatrix:nnn
998 {
999   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1000   {

```

```

1001     \_physicx_matrix_appto_body_e:nnn
1002   }
1003   {
1004     \cs_set_eq:NN \_physicx_qxmatrix_appto_body:nnn
1005     \_physicx_matrix_appto_body_ne:nnn
1006   }
1007   % clear the variable containing the body of the matrix
1008   \tl_clear:N \l__physicx_matrix_body_tl
1009   % set the tentative number of explicit rows
1010   \physicx_if_num:nTF { #2 }
1011   {% number of rows is an integer
1012     \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
1013     {% if #2 <= rows, we don't want a row of dots
1014       \bool_set_false:N \l__physicx_matrix_dotrow_bool
1015       \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1016     }
1017     {% we want a row of dots
1018       \bool_set_true:N \l__physicx_matrix_dotrow_bool
1019     }
1020   }
1021   {% number of rows is symbolic, we want a row of dots
1022     \bool_set_true:N \l__physicx_matrix_dotrow_bool
1023   }
1024   % set the tentative number of explicit columns
1025   \physicx_if_num:nTF { #3 }
1026   {% number of cols is an integer
1027     \int_compare:nTF { #3 <= \l__physicx_matrix_cols_int }
1028     {% if #3 <= cols, we don't want a column of dots
1029       \bool_set_false:N \l__physicx_matrix_dotcol_bool
1030       \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1031     }
1032     {% we want a column of dots
1033       \bool_set_true:N \l__physicx_matrix_dotcol_bool
1034     }
1035   }
1036   {% number of columns is symbolic, we want a column of dots
1037     \bool_set_true:N \l__physicx_matrix_dotcol_bool
1038   }
1039   % loop through the rows
1040   \int_step_inline:nn { \l__physicx_matrix_rows_int }
1041   {
1042     % add the first entry in the row
1043     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1044     \_physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
1045     % add the further entries in the explicit columns
1046     \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1047     {
1048       %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
1049       \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1050       \_physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
1051     }
1052     % if we have a column of dots, add \cdots and the last entry
1053     \bool_if:NT \l__physicx_matrix_dotcol_bool
1054     {

```

```

1055         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{##1 #3} }
1056         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1057         \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {#3}
1058     }
1059     % infinite matrix, add \cdots
1060     \bool_if:NT \l__physicx_matrix_infinite_bool
1061     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1062     \if_int_compare:w ##1 = \l__physicx_matrix_rows_int
1063     \scan_stop:
1064     \else:
1065         % finish up the row
1066         \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1067     \fi:
1068 }
1069 % finish up the rows
1070 \bool_if:NT \l__physicx_matrix_dotrow_bool
1071 {
1072     % finish up the row
1073     \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1074     % if we have a row of dots, fill it in
1075     \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1076     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1077     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1078     \bool_if:NT \l__physicx_matrix_dotcol_bool
1079     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
1080     \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1081     % fill the last row
1082     %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
1083     \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
1084     \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1085     {
1086         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
1087         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1088         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
1089     }
1090     \bool_if:NT \l__physicx_matrix_dotcol_bool
1091     {
1092         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
1093         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1094         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
1095     }
1096     % if the matrix is infinite, add a further column with \cdots
1097     \bool_if:NT \l__physicx_matrix_infinite_bool
1098     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1099 }
1100 % if the matrix is infinite, add a final row
1101 \bool_if:NT \l__physicx_matrix_infinite_bool
1102 {
1103     % finish up the row
1104     \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
1105     \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1106     \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1107     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1108     \bool_if:NT \l__physicx_matrix_dotcol_bool

```

```

1109         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1110     \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1111     % update cols
1112     \bool_if:NTF \l__physicx_matrix_dotcol_bool
1113     { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1114     { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1115 }
1116 }

```

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

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
1117 \cs_new:Npn \physicx_matrix_diag_parse:n #1
1118 {
1119     \keyval_parse:nnn
1120     \__physicx_matrix_diag_parse_aux:n
1121     \__physicx_matrix_diag_parse_aux:nn
1122     {#1}
1123 }
1124 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
1125 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
1126 {
1127     \str_case_e:nnF {#1}
1128     {
1129         { auto-update }
1130         {
1131             \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
1132             \__physicx_matrix_calc:nn
1133         }
1134         { noauto-update }
1135         {
1136             \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
1137         }
1138         { true }
1139         {
1140             \bool_set_true:N \l__physicx_matrix_diag_bool
1141             \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1142             \__physicx_diagonalmatrix_set_diag:
1143         }
1144         { false }
1145         {
1146             \bool_set_false:N \l__physicx_matrix_diag_bool
1147             \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1148             \__physicx_diagonalmatrix_no_diag:
1149         }
1150     }
1151     { \msg_error:nnn { physicx } { diag-key } {#1} }
1152 }
1153 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
1154 {
1155     \tl_set:Nn \l__physicx_tmpdiag_tl {#2}
1156     \tl_set:Nx \l__physicx_tmpdiag_tl
1157     { \__physicx_expand:w \l__physicx_tmpdiag_tl }
1158     \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl

```

```

1159     \tl_if_head_eq_charcode:nNTF {#1} '
1160     {
1161         \exp_args:Nf \__physicx_matrix_diag_parse_aux_anti:n
1162         { \tl_tail:n {#1} }
1163     }
1164     { \__physicx_matrix_diag_parse_aux_regu:n {#1} }
1165 }
1166 \cs_new:Npn \__physicx_diagonalmatrix_set_diag:
1167 {
1168     \int_zero:N \l__physicx_matrix_cols_int
1169     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1170     {
1171         \int_incr:N \l__physicx_matrix_cols_int
1172         \physicx_matrix_set_r_c:nnn {##1} {##1} {##2}
1173     }
1174     \int_set_eq:NN \l__physicx_matrix_rows_int
1175     \l__physicx_matrix_cols_int
1176 }
1177 \cs_new:Npn \__physicx_diagonalmatrix_no_diag:
1178 {
1179     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1180     { \physicx_matrix_set_r_c:nnn {##1} {##1} {##2} }
1181     \__physicx_matrix_diag_calc:nn
1182     { \seq_count:N \l__physicx_tmpdiag_seq }
1183     { \seq_count:N \l__physicx_tmpdiag_seq }
1184 }
1185 \cs_new_eq:NN \__physicx_diagonalmatrix_diag_main:
1186 \__physicx_diagonalmatrix_no_diag:
1187 \cs_new:Npn \__physicx_matrix_diag_parse_aux_regu:n #1
1188 {
1189     \if_int_compare:w #1 = 0 \exp_stop_f:
1190         \__physicx_diagonalmatrix_diag_main:
1191     \else:
1192         \if_int_compare:w #1 > 0 \exp_stop_f:
1193             \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1194             {
1195                 \physicx_matrix_set_r_c:nnn
1196                 {##1} { \int_eval:n { ##1 + #1 } } {##2}
1197             }
1198             \__physicx_matrix_diag_calc:nn
1199             { \seq_count:N \l__physicx_tmpdiag_seq }
1200             { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1201         \else:
1202             \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1203             {
1204                 \physicx_matrix_set_r_c:nnn
1205                 { \int_eval:n { ##1 - #1 } } {##1} {##2}
1206             }
1207             \__physicx_matrix_diag_calc:nn
1208             { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1209             { \seq_count:N \l__physicx_tmpdiag_seq }
1210         \fi:
1211     \fi:
1212 }

```



```

1213 \cs_new:Npn \__physicx_matrix_diag_parse_aux_anti:n #1
1214 {
1215   \if_int_compare:w #1 = 0 \exp_stop_f:
1216     \__physicx_matrix_diag_calc:nn
1217     { \seq_count:N \l__physicx_tmpdiag_seq }
1218     { \seq_count:N \l__physicx_tmpdiag_seq }
1219     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1220     {
1221       \physicx_matrix_set_r_c:nnn
1222       {##1}
1223       { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1224       {##2}
1225     }
1226   \else:
1227     \if_int_compare:w #1 > 0 \exp_stop_f:
1228       \__physicx_matrix_diag_calc:nn
1229       { \seq_count:N \l__physicx_tmpdiag_seq }
1230       { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1231       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1232       {
1233         \physicx_matrix_set_r_c:nnn
1234         {##1}
1235         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1236         {##2}
1237       }
1238     \else:
1239       \__physicx_matrix_diag_calc:nn
1240       { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1241       { \seq_count:N \l__physicx_tmpdiag_seq }
1242       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1243       {
1244         \physicx_matrix_set_r_c:nnn
1245         { \int_eval:n { ##1 - #1 } }
1246         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1247         {##2}
1248       }
1249     \fi:
1250   \fi:
1251 }
1252 \cs_new:Npn \__physicx_matrix_diag_calc:nn
1253 { \__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
1254 \cs_new:Npn \physicx_matrix_item_parse:n #1
1255 {
1256   \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1257   \keyval_parse:NNn
1258   \__physicx_matrix_item_parse_aux:n
1259   \__physicx_matrix_item_parse_aux:nn
1260   {#1}
1261 }
1262 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }

```

```

1263 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1264 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2
1265 {
1266   \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1267   \tl_set:Nx \l__physicx_tmpitem_tl
1268     { \__physicx_expand:w \l__physicx_tmpitem_tl }
1269   \physicx_parse_range:nxN \l__physicx_matrix_rows_int
1270     { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1271   \physicx_parse_range:nxN \l__physicx_matrix_cols_int
1272     { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1273   \exp_args:No \tl_if_eq:nnTF
1274     { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1275   {
1276     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1277       {
1278         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1279           {
1280             \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1281           }
1282       }
1283   }
1284   {
1285     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1286       {
1287         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1288           {
1289             \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }
1290             {
1291               \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1292                 {##1} {####1} { \l__physicx_tmpitem_tl }
1293             }
1294           }
1295       }
1296   }
1297 }

```

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

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

```

\physicx_matrix_array_parse:o 1298 \cs_new:Npn \physicx_matrix_array_parse:n #1
1299 {
1300   \tl_set:Nn \l__physicx_tmparr_tl {#1}
1301   \tl_set:Nx \l__physicx_tmparr_tl
1302     { \__physicx_expand:w \l__physicx_tmparr_tl }
1303   \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl
1304   \__physicx_matrix_autocalc:nn
1305     { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1306     { 0 }
1307   \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1308     {
1309       \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1310       \__physicx_matrix_autocalc:nn
1311         { 0 }
1312       { \seq_count:N \l__physicx_matrix_tmparr_c_sep }

```

```

1313     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1314     {
1315         \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1316     }
1317 }
1318 }
1319 \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.

```

1320 \cs_new:Npn \physicx_matrix_array_parse_main:
1321 {
1322     \int_step_inline:nn \l__physicx_matrix_rows_int
1323     {
1324         \int_step_inline:nn \l__physicx_matrix_cols_int
1325         {
1326             \exp_args:Nno \physicx_matrix_set_r_c:nnn
1327             {##1} {####1} \l__physicx_matrix_main_tl
1328         }
1329     }
1330 }

```

(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).

```

1331 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1332 {
1333     \physicx_if_num:nTF {#1}
1334     { \prg_return_true: }
1335     {
1336         \bool_case_true:nTF
1337         {
1338             { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1339             { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1340             {
1341                 \bool_lazy_and_p:nn
1342                 { \cs_if_exist_p:N \inteval }
1343                 { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1344             } { }
1345             {
1346                 \bool_lazy_and_p:nn
1347                 { \cs_if_exist_p:N \fpeval }
1348                 { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1349             } { }
1350         }
1351         { \prg_return_true: }
1352         { \prg_return_false: }
1353     }
1354 }

```

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

```

\diagonalmatrix Define \diagonalmatrix.
1355 \DeclareDocumentCommand \diagonalmatrix { t= t+ 0{} m }
1356 {
1357   \group_begin:
1358   \IfBooleanTF {#1}
1359   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1360   { \keys_set:nn { physicx/matrix } { #3 } }
1361   \physicx_construct:nnn { }
1362   {
1363     \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist
1364     \tl_if_empty:nF {#4}
1365     {
1366       \__physicx_if_keyval:nTF {#4}
1367       { \physicx_matrix_diag_parse:n { true, #4 } }
1368       { \physicx_matrix_diag_parse:n { true, 0 = {#4} } }
1369     }
1370   }
1371   { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1372   \bool_lazy_or:nnTF
1373   { \bool_if_p:n {#2} }
1374   { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1375   {
1376     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1377     {
1378       \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1379       \__physicx_matrix_appto_body_e:off
1380     }
1381     {
1382       \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1383       \__physicx_matrix_appto_body_ne:off
1384     }
1385     \use_i_ii:nnn
1386   }
1387   { \use_i:nn }
1388   \__physicx_matrix_transpose:N
1389   \__physicx_diagonalmatrix_generate_enhanced_body:NNN
1390   \__physicx_diagonalmatrix_generate_body:NNN
1391   \__physicx_matrix_save_or_print:
1392   \group_end:
1393 }
1394 \cs_new:Npn \__physicx_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1395 {
1396   \__physicx_matrix_generate_body:NNNN #1#2#3
1397   \__physicx_diagonalmatrix_enhanced:nnn
1398 }
1399 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1400 {
1401   \int_step_inline:nn { #1 - 1 }
1402   {
1403     \int_step_inline:nn { #2 - 1 }
1404     {
1405       \tl_put_right:Nx \l__physicx_matrix_body_tl
1406       {
1407         \exp_after:wN

```

```

1408         \physicx_matrix_use_r_c:nn
1409         #3 {{##1}} {{####1}} &
1410     }
1411 }
1412 \tl_put_right:Nx \l__physicx_matrix_body_tl
1413 {
1414     \exp_after:wN
1415     \physicx_matrix_use_r_c:nn
1416     #3 {{##1}} {{ \int_use:N #2 }} \[\dim_use:N \l__physicx_matrix_sep_dim]
1417 }
1418 }
1419 \int_step_inline:nn { #2 - 1 }
1420 {
1421     \tl_put_right:Nx \l__physicx_matrix_body_tl
1422     {
1423         \exp_after:wN
1424         \physicx_matrix_use_r_c:nn
1425         #3 {{ \int_use:N #1 }} {{##1}} &
1426     }
1427 }
1428 \tl_put_right:Nx \l__physicx_matrix_body_tl
1429 {
1430     \exp_after:wN
1431     \physicx_matrix_use_r_c:nn
1432     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1433 }
1434 }

```

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

\\_\_physicx\_declare\_init:

```

1435 \cs_new:Npn \__physicx_matrix_enhanced_init:
1436 {
1437     \seq_if_empty:NF \l__physicx_row_list_seq
1438     {
1439         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1440         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1441         { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1442     }
1443     \seq_if_empty:NF \l__physicx_col_list_seq
1444     {
1445         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1446         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1447         { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1448     }
1449 }

```

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

\commamatrix Define \commamatrix.

```

1450 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1451 {
1452     \group_begin:
1453     \keys_set:nn { physicx/matrix } {#3}
1454     \tl_if_empty:nF {#4}

```

```

1455     { \keys_set:nn { physicx/matrix } { array = {#4} } }
1456 \IfBooleanT {#1}
1457     { \keys_set:nn { physicx/matrix } { saveto = \physicx_tmp } }
1458 \tl_set:Nx \l__physicx_matrix_array_tl
1459     { \__physicx_expand:w \l__physicx_matrix_array_tl }
1460 \bool_lazy_or:nnTF
1461     { \bool_if_p:n {#2} }
1462     { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1463     { \__physicx_commamatrix_enhanced: }
1464     {
1465         \tl_replace_all:Nox \l__physicx_matrix_array_tl
1466         { \physicx@cr } { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1467         \tl_replace_all:Non \l__physicx_matrix_array_tl
1468         { \physicx@align } { & }
1469         \tl_set_eq:NN \l__physicx_matrix_body_tl
1470         \l__physicx_matrix_array_tl
1471     }
1472 \__physicx_matrix_save_or_print:
1473 \group_end:
1474 }
1475 \cs_new_nopar:Npn \__physicx_matrix_save_or_print:
1476 {
1477     \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1478     {
1479         \exp_after:wN \tl_gset_eq:NN
1480         \l__physicx_matrix_save_tl
1481         \l__physicx_matrix_body_tl
1482     }
1483     {
1484         \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1485         \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1486         \fi:
1487         \exp_after:wN \__physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matrix_body_tl
1488         \l__physicx_matrix_body_tl
1489         \__physicx_matrix_end: \l__physicx_matrix_after_end_tl
1490     }
1491 }
1492 \cs_new:Npn \__physicx_commamatrix_enhanced:
1493 {
1494     \tl_clear:N \l__physicx_matrix_body_tl
1495     \int_zero:N \l__physicx_tmpa_int
1496     \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1497     \l__physicx_matrix_array_tl
1498     \int_set:Nn \l__physicx_matrix_rows_int
1499     { \seq_count:N \l__physicx_tmp_seq }
1500     \__physicx_matrix_enhanced_init:
1501     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1502     {
1503         \seq_map_tokens:Nn \l__physicx_tmp_seq
1504         {
1505             \int_incr:N \l__physicx_tmpa_int
1506             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1507             \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_e:nnn
1508         }
1509     }
1510 }

```

```

1509     }
1510     {
1511         \seq_map_tokens:Nn \l__physicx_tmp_seq
1512         {
1513             \int_incr:N \l__physicx_tmpa_int
1514             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1515             \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_ne:nnn
1516         }
1517     }
1518 }
1519 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1520 {
1521     \seq_set_split:Non \l__physicx_tmp_col_seq
1522     { \physicx@align } {#3}
1523     \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1524     \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1525     { #2 {##2} {#1} {##1} }
1526     \tl_put_right:Nx \l__physicx_matrix_body_tl
1527     {
1528         \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1529         \if_int_compare:w \l__physicx_matrix_rows_int = #1
1530             \scan_stop:
1531         \else:
1532             \[\dim_use:N \l__physicx_matrix_sep_dim]
1533         \fi:
1534     }
1535 }
1536 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1537 {
1538     \seq_put_right:Nx \l__physicx_tmp_coled_seq
1539     {
1540         \text_expand:n % \text_expand:n do the magic thing, but slower
1541         {
1542             \physicx@matrixellement { #1 }
1543             { \__physicx_matrix_row_iterate:n {#2} }
1544             { \__physicx_matrix_col_iterate:n {#3} }
1545         }
1546     }
1547 }
1548 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1549 {
1550     \seq_put_right:No \l__physicx_tmp_coled_seq
1551     {
1552         \physicx@matrixellement {#1}
1553         { \__physicx_matrix_row_iterate:n {#2} }
1554         { \__physicx_matrix_col_iterate:n {#3} }
1555     }
1556 }

```

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

`\generalmatrix` Define `\generalmatrix`.

```

1557 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1558 {

```

```

1559 \IfBooleanTF {#2}
1560 {
1561   \group_begin:
1562   \IfBooleanTF {#1}
1563     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicxtmp } }
1564     { \keys_set:nn { physicx/matrix } {#4} }
1565   \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1566   \physicx_construct:nnn
1567   {
1568     \tl_if_empty:NTF \l__physicx_matrix_main_tl
1569     {
1570       \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1571     }
1572     { \physicx_matrix_array_parse_main: }
1573   }
1574   { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1575   { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1576   \__physicx_generalmatrix:
1577   \__physicx_matrix_save_or_print:
1578   \group_end:
1579 }
1580 {
1581   \IfBooleanTF {#1}
1582     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1583     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1584   \qxmatrix = * [#4]
1585 }
1586 }
1587 \cs_new:Npn \__physicx_generalmatrix:
1588 {
1589   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1590   {
1591     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1592     \__physicx_matrix_appto_body_e:off
1593   }
1594   {
1595     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1596     \__physicx_matrix_appto_body_ne:off
1597   }
1598   \__physicx_matrix_transpose:N
1599   \__physicx_matrix_generate_body:NNNN
1600   \__physicx_generalmatrix_generate:nnn
1601 }

```

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

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

```

1602 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1603 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1604 {
1605   \__physicx_matrix_enhanced_init:
1606   \int_step_inline:nn { #1 - 1 }
1607   {
1608     \int_step_inline:nn { #2 - 1 }

```



```

1609     {
1610         \tl_set:Nx \l__physicx_tmp_tl
1611         {
1612             \exp_after:wN
1613             \physicx_matrix_use_r_c:nn
1614             #3 {{##1}} {{####1}}
1615         }
1616         #4 \l__physicx_tmp_tl {##1} {####1}
1617         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1618     }
1619     \tl_set:Nx \l__physicx_tmp_tl
1620     {
1621         \exp_after:wN
1622         \physicx_matrix_use_r_c:nn
1623         #3 {{##1}} {{ \int_use:N #2 }}
1624     }
1625     #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1626     \tl_put_right:Nx \l__physicx_matrix_body_tl
1627     { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1628 }
1629 \int_step_inline:nn { #2 - 1 }
1630 {
1631     \tl_set:Nx \l__physicx_tmp_tl
1632     {
1633         \exp_after:wN
1634         \physicx_matrix_use_r_c:nn
1635         #3 {{ \int_use:N #1 }} {{##1}}
1636     }
1637     #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}
1638     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1639 }
1640 \tl_set:Nx \l__physicx_tmp_tl
1641 {
1642     \exp_after:wN
1643     \physicx_matrix_use_r_c:nn
1644     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1645 }
1646 #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1647 }

```

(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
1648 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1649 {
1650     \tl_put_right:Nx \l__physicx_matrix_body_tl
1651     {
1652         \text_expand:n
1653         {
1654             \physicx@matricelement {#1}
1655             { \__physicx_matrix_row_iterate:n {#2} }
1656             { \__physicx_matrix_col_iterate:n {#3} }
1657         }
1658     }

```

```

1659 }
1660 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1661 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1662 {
1663   \tl_put_right:No \l__physicx_matrix_body_tl
1664   {
1665     \physicx@matricelement {#1}
1666     { \__physicx_matrix_row_iterate:n {#2} }
1667     { \__physicx_matrix_col_iterate:n {#3} }
1668   }
1669 }
1670 \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

```

1671 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1672 {
1673   \bool_if:NTF \l__physicx_matrix_transpose_bool
1674   {
1675     #1
1676     \l__physicx_matrix_cols_int
1677     \l__physicx_matrix_rows_int
1678     \use_ii_i:nn
1679   }
1680   {
1681     #1
1682     \l__physicx_matrix_rows_int
1683     \l__physicx_matrix_cols_int
1684     \use:nn
1685   }
1686 }

```

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

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

```

1687 \cs_new:Npn \physicx_construct:nnn #1#2#3
1688 {
1689   \l__physicx_matrix_beginning_tl
1690   \__physicx_adi:nnn {#1} {#2} {#3}
1691   \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1692   {
1693     \int_incr:N \l__physicx_matrix_cols_int
1694     \__physicx_matrix_last_aux_c:
1695     \int_incr:N \l__physicx_matrix_cols_int
1696   }
1697   \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1698   {
1699     \int_incr:N \l__physicx_matrix_rows_int
1700     \__physicx_matrix_last_aux_r:
1701     \int_incr:N \l__physicx_matrix_rows_int
1702   }
1703   \bool_lazy_or:nnF

```

```

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

```

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

### 1.3.3 Define new matrix command

```

\__physicx_new_matrix_cmd:NNN
  \newgeneralmatrix 1743 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
  \NewGeneralMatrix 1744 {
  \newdiagonalmatrix 1745   \NewDocumentCommand #2 { t+ m o o m m }
  \NewDiagonalMatrix 1746   {
  \newcommamatrix 1747     \IfBooleanTF {##1}
  \NewCommaMatrix 1748     {
1749       \IfNoValueTF {##3}
1750       { \newcommand ##2 { #1 + [##5] {##6} } }
1751       {

```

```

1752         \IfNoValueTF {##4}
1753         { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1754         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1755     }
1756 }
1757 {
1758     \IfNoValueTF {##3}
1759     { \newcommand ##2 { #1 [##5] {##6} } }
1760     {
1761         \IfNoValueTF {##4}
1762         { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1763         { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1764     }
1765 }
1766 }
1767 \NewDocumentCommand #3 { t+ m m m m }
1768 {
1769     \IfBooleanTF {##1}
1770     { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1771     { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1772 }
1773 }
1774 \_physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1775 \_physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1776 \NewDocumentCommand \newgeneralmatrix { t+ m o o m }
1777 {
1778     \IfBooleanTF {#1}
1779     {
1780         \IfNoValueTF {#3}
1781         { \newcommand #2 { \generalmatrix + {#5} } }
1782         {
1783             \IfNoValueTF {#4}
1784             { \newcommand #2 [3] { \generalmatrix + {#5} } }
1785             { \newcommand #2 [3] [4] { \generalmatrix + {#5} } }
1786         }
1787     }
1788     {
1789         \IfNoValueTF {#3}
1790         { \newcommand #2 { \generalmatrix {#5} } }
1791         {
1792             \IfNoValueTF {#4}
1793             { \newcommand #2 [3] { \generalmatrix {#5} } }
1794             { \newcommand #2 [3] [4] { \generalmatrix {#5} } }
1795         }
1796     }
1797 }
1798 \NewDocumentCommand \NewGeneralMatrix { t+ m m m m }
1799 {
1800     \IfBooleanTF {#1}
1801     { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1802     { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1803 }

```

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

# Index

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

| Symbols   |  |
|---|--|
| $\backslash +$ .....                            | 154  |
| $\backslash -$ .....                            | 154  |
| $\backslash \backslash$ .....                   | 44, 1066, 1073,<br>1080, 1104, 1416, 1466, 1532, 1627  |
| $\backslash \{$ .....                           | 295, 524, 543, 561, 602, 627   |
| $\backslash \}$ .....                           | 295, 524, 545, 561, 602, 627   |
| A   |  |
| $\backslash A$ .....                            | 149, 154   |
| $\backslash \text{abs}$ .....                   | <u>595</u>   |
| $\backslash \text{absolutevalue}$ .....         | <u>595</u>   |
| $\backslash \text{acomm}$ .....                 | <u>595</u>   |
| $\backslash \text{aftergroup}$ .....            | 204  |
| $\backslash \text{anticommutator}$ .....        | <u>595</u>   |
| B   |  |
| $\backslash \text{begin}$ .....                 | 300, 301, 302,<br>303, 304, 305, 306, 309, 310, 311,<br>312, 313, 314, 315, 316, 317, 318,<br>319, 320, 321, 322, 323, 324, 325,<br>544, 547, 551, 554, 555, 556, 561,<br>562, 566, 569, 570, 571, 923, 924,<br>925, 926, 927, 928, 929, 932, 933,<br>934, 935, 936, 937, 938, 939, 940,<br>941, 942, 943, 944, 945, 946, 947, 948 |
| $\backslash \text{bgroup}$ .....                | 203, 205   |
| $\backslash \text{Big}$ .....                   | 577  |
| $\backslash \text{big}$ .....                   | 577  |
| $\backslash \text{Bigg}$ .....                  | 577  |
| $\backslash \text{bigg}$ .....                  | 577  |
| $\backslash \text{Biggl}$ .....                 | 584  |
| $\backslash \text{biggl}$ .....                 | 583  |
| $\backslash \text{Biggr}$ .....                 | 584  |
| $\backslash \text{biggr}$ .....                 | 583  |
| $\backslash \text{Bigl}$ .....                  | 582  |
| $\backslash \text{bigl}$ .....                  | 581  |
| $\backslash \text{Bigr}$ .....                  | 582  |
| $\backslash \text{bigr}$ .....                  | 581  |
| $\backslash \text{boldsymbol}$ .....            | 193  |
| bool commands:                                  |  |
| $\backslash \text{bool\_case\_false:n}$ ....    | 494, 495, 502  |
| $\backslash \text{bool\_case\_true:nTF}$ .....  | 579, 1336  |
| $\backslash \text{bool\_if:NtF}$ .....          | 16, 21, 26, 72,<br>182, 508, 510, 514, 998, 1053, 1060,<br>1070, 1078, 1090, 1097, 1101, 1108,<br>1112, 1376, 1501, 1589, 1673, 1712   |
| $\backslash \text{bool\_if\_p:N}$ .....         | 1374, 1462   |
| $\backslash \text{bool\_if\_p:n}$ .....         | 581, 582, 583, 584, 1373, 1461   |
| $\backslash \text{bool\_lazy\_and\_p:nn}$ ..... | 1341, 1346   |
| $\backslash \text{bool\_lazy\_or:nnTF}$ .....   | 31,<br>113, 124, 348, 353, 1372, 1460, 1703  |
| $\backslash \text{bool\_new:N}$ .....           | 9, 10, 11, 12, 13, 38, 53, 410,<br>412, 686, 687, 688, 696, 710, 718, 719  |
| $\backslash \text{bool\_set:Nn}$ .....          | 421, 422, 1565   |
| $\backslash \text{bool\_set\_false:N}$ .....    | 82, 170, 229, 986, 1014, 1029, 1146  |
| $\backslash \text{bool\_set\_inverse:N}$ .....  | 756, 873   |
| $\backslash \text{bool\_set\_true:N}$ .....     | 116,<br>127, 137, 177, 225, 228, 985, 1018,<br>1022, 1033, 1037, 1140, 1439, 1445  |
| $\backslash \text{c\_false\_bool}$ .....        | 539, 559   |
| $\backslash \text{c\_true\_bool}$ .....         | 522, 532, 539, 559   |
| $\backslash \text{Bqty}$ .....                  | <u>595</u>   |
| $\backslash \text{bqty}$ .....                  | <u>595</u>   |
| C   |  |
| $\backslash \text{cdots}$ ....                  | 773, 1052, 1055, 1056, 1059,<br>1061, 1092, 1093, 1096, 1098, 1731   |
| clist commands:                                 |  |
| $\backslash \text{clist\_clear:N}$ .....        | 954  |
| $\backslash \text{clist\_concat:NNN}$ .....     | 961  |
| $\backslash \text{clist\_if\_in:NnTF}$ .....    | 1289   |
| $\backslash \text{clist\_map\_break:n}$ .....   | 163, 175   |
| $\backslash \text{clist\_map\_inline:nn}$ ..... | 69, 159, 171, 959  |
| $\backslash \text{clist\_new:N}$ .....          | 694, 695   |
| $\backslash \text{clist\_put\_right:Nn}$ ..     | 835, 841, 1280   |
| $\backslash \text{clist\_set:Nn}$ .....         | 951  |
| $\backslash \text{clist\_set\_eq:NN}$ .....     | 1256   |
| $\backslash \text{c\_empty\_clist}$ .....       | 1256   |
| $\backslash \text{comm}$ .....                  | 625  |
| $\backslash \text{commamatrix}$ .....           | 29, 1450, 1775   |
| $\backslash \text{commutator}$ .....            | <u>595</u>   |

|                             |  |                   |  |
|-----------------------------|--|-------------------|--|
| cs commands:                |  | else commands:    |  |
| \cs:w                       | 32, 33, 864  | \else:            | 651, 1064, 1191, 1201, 1226, 1238, 1531  |
| \cs_end:                    | 32, 33, 649, 864   | \end              | 300, 301, 302, 303, 304, 305, 306, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 544, 547, 551, 554, 555, 556, 561, 562, 566, 569, 570, 571, 923, 924, 925, 926, 927, 928, 929, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948 |
| \cs_generate_variant:Nn     | 3, 4, 5, 6, 76, 79, 185, 366, 505, 1124, 1262, 1319, 1660, 1670  | \eval             | 595  |
| \cs_if_exist:NTF            | 272  | \evaluated        | 520, 608   |
| \cs_if_exist_p:N            | 1342, 1347   | exp commands:     |  |
| \cs_new:Npn                 | 7, 111, 119, 122, 134, 145, 157, 191, 195, 199, 203, 204, 205, 206, 333, 345, 415, 425, 437, 474, 481, 506, 575, 721, 913, 915, 1117, 1125, 1153, 1166, 1177, 1187, 1213, 1252, 1254, 1263, 1264, 1298, 1320, 1394, 1399, 1435, 1492, 1519, 1536, 1548, 1587, 1603, 1648, 1661, 1671, 1687, 1725, 1734, 1743 | \exp_after:wN     | 1407, 1414, 1423, 1430, 1477, 1479, 1487, 1612, 1621, 1633, 1642   |
| \cs_new_eq:NN               | 121, 144, 680, 700, 709, 1185  | \exp_args:Nc      | 385, 388   |
| \cs_new_nopar:Npn           | 639, 647, 656, 658, 663, 669, 702, 703, 706, 707, 708, 1475  | \exp_args:Nf      | 1161   |
| \cs_new_protected:Nn        | 996  | \exp_args:Nnno    | 1291, 1326   |
| \cs_new_protected:Npn       | 54, 59, 64, 77, 80, 187  | \exp_args:Nno     | 161, 173   |
| \cs_new_protected_nopar:Npn | 950, 952, 957  | \exp_args:NNx     | 391  |
| \cs_set:Nn                  | 878, 879, 880, 881, 882, 883   | \exp_args:No      | 791, 800, 902, 1273  |
| \cs_set:Npn                 | 371, 377, 381, 598, 608, 609, 625, 630, 631, 632, 762, 771, 780, 789, 798  | \exp_args:NW      | 1506, 1514   |
| \cs_set:Npx                 | 805  | \exp_not:N        | 8, 275, 393, 394, 395, 445, 805  |
| \cs_set_eq:NN               | 56, 57, 61, 62, 193, 197, 201, 678, 731, 733, 735, 738, 744, 746, 760, 769, 778, 787, 796, 814, 817, 820, 823, 826, 1000, 1004, 1131, 1136, 1141, 1147, 1378, 1382, 1591, 1595   | \exp_not:n        | 449, 450, 650, 652, 700, 731, 735  |
| \cs_set_nopar:Npn           | 1440, 1446   | \exp_stop_f:      | 8, 440, 447, 1189, 1192, 1215, 1227  |
| \cs_set_protected:Npn       | 486, 499, 518, 593   | <b>F</b>          |  |
| \cs_to_str:N                | 386, 389, 394, 395, 431  | fi commands:      |  |
| <b>D</b>                    |  | \fi:              | 451, 466, 471, 472, 573, 634, 653, 1067, 1210, 1211, 1249, 1250, 1486, 1533  |
| \ddots                      | 773, 1079, 1110, 1710, 1721  | fp commands:      |  |
| \DeclareDocumentCommand     | 491, 501, 577, 981, 1355, 1450, 1557   | \fp_eval:n        | 27, 1339   |
| \diagonalmatrix             | 28, 1355, 1774   | \fpeval           | 27, 1347, 1348   |
| dim commands:               |  | <b>G</b>          |  |
| \dim_new:N                  | 720  | \generalmatrix    | 31, 1557, 1781, 1784, 1785, 1790, 1793, 1794, 1801, 1802   |
| \dim_use:N                  | 1066, 1073, 1080, 1104, 1416, 1466, 1532, 1627   | group commands:   |  |
| <b>E</b>                    |  | \group_begin:     | 335, 983, 1357, 1452, 1561   |
| \egroup                     | 204, 206   | \group_end:       | 343, 994, 1392, 1473, 1578   |
|                             |  | <b>I</b>          |  |
|                             |  | if commands:      |  |
|                             |  | \if_bool:N        | 455, 467, 520, 595   |
|                             |  | \if_case:w        | 447  |
|                             |  | \if_cs_exist:w    | 649  |
|                             |  | \if_int_compare:w | 440, 1062, 1189, 1192, 1215, 1227, 1484, 1529  |



|   |   |   |
|---|---|---|
| \physicx_declare_legacy_paren:NnnnNNn   | \physicx_xquantity:nn                   | 333                                     |
| ..... 574, 599, 600, 601, 602,          | physicx internal commands:              |   |
| 604, 611, 615, 619, 622, 626, 635, 637  | \__physicx_adi:nnn                      | 721, 878, 879, 880, 881, 882, 883, 1690 |
| \physicx_declare_legacy_quantity:nnNn   | \l__physicx_begin_int                   | 49,                                     |
| ..... 408, 521, 531, 538, 558           | 89, 91, 92, 94, 125, 126, 130, 136, 140 |   |
| \physicx_if_num:n                       | \l__physicx_cmd_arg_int                 | 414, 420, 439, 440, 447                 |
| ..... 147                               | \l__physicx_cmd_arg_spec_tl             | 413, 419, 432, 441                      |
| \physicx_if_num:nTF                     | \l__physicx_cmd_auto_body_bool          | 412, 422, 467, 510, 514                 |
| ..... 902, 1010, 1025, 1333             | \l__physicx_cmd_auto_body_tl            | 411, 418, 434, 468, 469                 |
| \physicx_if_num_sign:n                  | \l__physicx_cmd_noauto_body_bool        | 410, 421, 455, 508                      |
| ..... 152                               | \l__physicx_cmd_noauto_body_tl          | 409, 417, 433, 456, 457                 |
| \physicx_left: .....                    | \l__physicx_col_list_seq                | 698, 752, 1443, 1447                    |
| 203, 356, 489, 589                      | \__physicx_commamatrix_enhanced:        | 1463, 1492                              |
| \physicx_left:N                         | \__physicx_commamatrix_enhanced_-       | aux:nNn                                 |
| ..... 205, 358, 581, 582, 583, 584      | aux_e:nnn                               | 1506, 1514, 1519                        |
| \physicx_mathtools: .....               | \__physicx_commamatrix_enhanced_-       | aux_ne:nnn                              |
| 24                                      | aux_ne:nnn                              | 1515, 1548                              |
| \physicx_mathtools:TF                   | \g__physicx_compat_bool                 | 11, 16, 209, 225                        |
| ..... 307, 930                          | \__physicx_declare_init: .....          | 1435                                    |
| \physicx_matrix_array_parse:n           | \__physicx_declare_init:nnn             | 415, 427                                |
| ..... 1298, 1570                        | \__physicx_declare_legacy_-             | quantity_aux:nnnn                       |
| \physicx_matrix_array_parse_-           | quantity_aux:NNnnn                      | 437, 478                                |
| main: .....                             | \__physicx_declare_legacy_-             | quantity_aux:nw                         |
| 1320, 1572                              | quantity_aux:nw                         | 428, 474, 479                           |
| \physicx_matrix_diag_parse:n            | \__physicx_diagonalmatrix_diag_-        | main: .....                             |
| ..... 836, 1117, 1363, 1367, 1368, 1574 | main: .....                             | 1141, 1147, 1185, 1190                  |
| \physicx_matrix_item_parse:n            | \__physicx_diagonalmatrix_-             | enhanced:nnn                            |
| ..... 842, 1254, 1371, 1575             | enhanced:nnn                            | 1378, 1382, 1397                        |
| \physicx_matrix_new_type:nn             | \__physicx_diagonalmatrix_-             | generate_body:NNN                       |
| ..... 913                               | generate_body:NNN                       | 1390, 1399                              |
| \physicx_matrix_new_type:nnn            | \__physicx_diagonalmatrix_-             | generate_enhanced_body:NNN              |
| ..... 913                               | generate_enhanced_body:NNN              | 1389, 1394                              |
| \physicx_matrix_set_r_c:nnn             | \__physicx_diagonalmatrix_no_-          | diag: .....                             |
| 680,                                    | diag: .....                             | 1148, 1177, 1186                        |
| 814, 817, 820, 823, 826, 1172, 1180,    | \__physicx_diagonalmatrix_set_-         | diag: .....                             |
| 1195, 1204, 1221, 1233, 1244, 1291,     | diag: .....                             | 1142, 1166                              |
| 1315, 1326, 1707, 1718, 1729, 1738      | \l__physicx_end_int                     | 50,                                     |
| \physicx_matrix_use_r_c:nn              | 99, 101, 102, 104, 126, 130, 136, 140   |   |
| ..... 647, 1408, 1415,                  |   |   |
| 1424, 1431, 1613, 1622, 1634, 1643      |   |   |
| \physicx_new_type:nnn                   |   |   |
| ..... 187, 293, 914, 916                |   |   |
| \physicx_option_or:nn                   |   |   |
| ..... 29                                |   |   |
| \physicx_option_or:nnTF                 |   |   |
| ..... 596, 606, 624, 628                |   |   |
| \c_physicx_Order_tl                     |   |   |
| 190, 234, 616, 636                      |   |   |
| \c_physicx_order_tl                     |   |   |
| 189, 233, 620, 638                      |   |   |
| \physicx_parse_range:nnN                |   |   |
| ..... 77, 79, 1269, 1271                |   |   |
| \physicx_parse_range:nnnN               |   |   |
| ..... 49                                |   |   |
| \physicx_parse_range_check: .           |   |   |
| 49, 846                                 |   |   |
| \physicx_parse_range_nocheck: .         |   |   |
| 49, 847                                 |   |   |
| \physicx_qxmatrix:nnn                   |   |   |
| ..... 992, 996                          |   |   |
| \physicx_right: .....                   |   |   |
| 204, 356, 489, 589                      |   |   |
| \physicx_right:N                        |   |   |
| ..... 206, 360, 581, 582, 583, 584      |   |   |
| \physicx_search_also:nn                 |   |   |
| ..... 157, 168, 185, 186                |   |   |
| \physicx_search_also:nnTF               |   |   |
| ..... 281, 894                          |   |   |
| \physicx_short: .....                   |   |   |
| 19                                      |   |   |
| \physicx_use_amsymb_type: .             |   |   |
| 191, 239                                |   |   |
| \physicx_use_uni_bf_type: .....         |   |   |
| 199                                     |   |   |
| \physicx_use_uni_bfit_type: .....       |   |   |
| 195, 238                                |   |   |



```

\__physicx_expand:w .... 700, 731,
733, 735, 738, 1157, 1268, 1302, 1459
\__physicx_generalmatrix: 1576, 1587
\__physicx_generalmatrix_-
generate:nnn .... 1591, 1595, 1600
\__physicx_if_can_num:n ..... 1331
\__physicx_if_keyval:nTF .. 145, 1366
\l__physicx_invalid_range_bool ..
..... 53, 72, 82, 116, 127, 137
\l__physicx_item_ignore_clist ...
..... 1256, 1280, 1289
\g__physicx_mathtools_bool .....
..... 9, 26, 228, 229
\__physicx_matrix_add_data:nn ...
..... 838, 844, 955, 957
\l__physicx_matrix_after_begin_-
tl ..... 716, 854, 856, 1487
\l__physicx_matrix_after_end_tl .
..... 717, 857, 859, 1489
\__physicx_matrix_appto_body_-
e:nnn ..... 1001, 1379, 1592, 1648
\__physicx_matrix_appto_body_-
ne:nnn ..... 1005, 1383, 1596, 1648
\l__physicx_matrix_args_tl .....
..... 715, 852, 853, 1487
\l__physicx_matrix_array_tl ....
..... 689, 728, 1458,
1459, 1465, 1467, 1470, 1497, 1570
\__physicx_matrix_autocalc:nn ...
..... 709, 744, 746, 1253, 1304, 1310
\__physicx_matrix_begin:w .....
..... 707, 849, 1487
\l__physicx_matrix_beginning_tl .
..... 722, 884, 886, 1689
\l__physicx_matrix_body_tl .....
..... 690, 1008, 1043,
1048, 1049, 1055, 1056, 1061, 1066,
1073, 1075, 1077, 1079, 1080, 1082,
1086, 1087, 1092, 1093, 1098, 1104,
1105, 1107, 1109, 1110, 1405, 1412,
1421, 1428, 1469, 1481, 1488, 1494,
1526, 1617, 1626, 1638, 1650, 1663
\__physicx_matrix_calc:nn .....
..... 639, 744, 1132
\__physicx_matrix_col_iterate:n .
706, 830, 1446, 1544, 1554, 1656, 1667
\l__physicx_matrix_cols_int ....
..... 643, 644,
692, 741, 988, 1027, 1030, 1046,
1076, 1084, 1106, 1113, 1114, 1168,
1171, 1175, 1223, 1235, 1246, 1271,
1324, 1484, 1485, 1676, 1683, 1693,
1695, 1709, 1715, 1720, 1730, 1736
\l__physicx_matrix_diag_bool ...
..... 696, 1140, 1146
\__physicx_matrix_diag_calc:nn ..
..... 1131, 1136, 1181,
1198, 1207, 1216, 1228, 1239, 1252
\l__physicx_matrix_diag_clist ...
..... 694, 833, 835, 1363, 1574
\__physicx_matrix_diag_parse_-
aux:n ..... 1120, 1125
\__physicx_matrix_diag_parse_-
aux:nn ..... 1121, 1153
\__physicx_matrix_diag_parse_-
aux_anti:n ..... 1161, 1213
\__physicx_matrix_diag_parse_-
aux_regu:n ..... 1164, 1187
\l__physicx_matrix_dotcol_bool ..
..... 688, 1029, 1033,
1037, 1053, 1078, 1090, 1108, 1112
\l__physicx_matrix_dotorow_bool ..
..... 687, 1014, 1018, 1022, 1070
\__physicx_matrix_element_-
aux:nnn ..... 760, 766,
769, 775, 778, 784, 787, 793, 796, 801
\__physicx_matrix_end: .... 850, 1489
\__physicx_matrix_end:w ..... 708
\l__physicx_matrix_ending_tl ...
..... 723, 887, 889, 1723
\l__physicx_matrix_enhanced_bool
..... 719, 870, 873, 1374, 1462
\__physicx_matrix_enhanced_init:
..... 1435, 1500, 1605
\l__physicx_matrix_expand_-
element_bool ..... 710,
809, 998, 1376, 1439, 1445, 1501, 1589
\__physicx_matrix_generate_-
body:NNNN ..... 1396, 1599, 1602
\l__physicx_matrix_infinite_bool
..... 686, 753, 756,
985, 986, 1060, 1097, 1101, 1565, 1712
\l__physicx_matrix_item_clist ...
..... 695, 839, 841, 1371, 1575
\__physicx_matrix_item_parse_-
aux:n ..... 1258, 1263
\__physicx_matrix_item_parse_-
aux:nn ..... 1259, 1264
\__physicx_matrix_last_aux_c: ...
..... 1694, 1716, 1725
\__physicx_matrix_last_aux_r: ...
..... 1700, 1717, 1734
\l__physicx_matrix_last_col_tl ..
..... 705, 832, 1691, 1705
\l__physicx_matrix_last_row_tl ..
..... 704, 831, 1697, 1704

```

|  |  |
|--|--|
| \l__physicx_matrix_main_tl . . . . .             | \__physicx_parse_range_single_-              |
| . . . . . 693, 748, 1327, 1568                   | check:n . . . . . 56, 111, 121               |
| \__physicx_matrix_row_iterate:n .                | \__physicx_parse_range_single_-              |
| . . . . . 703, 829, 1440, 1543, 1553, 1655, 1666 | nocheck:n . . . . . 61, 119                  |
| \l__physicx_matrix_rows_int . . . . .            | \g__physicx_physics_bool . . . . . 10        |
| . . . . . 641, 642, 691, 740, 987,               | \l__physicx_quantity_args_tl . . . . .       |
| 1012, 1015, 1040, 1062, 1174, 1269,              | . . . . . 260, 261, 340                      |
| 1322, 1498, 1529, 1677, 1682, 1699,              | \l__physicx_quantity_code_tl . . . . .       |
| 1701, 1708, 1714, 1719, 1727, 1739               | . . . . . 262, 337, 341                      |
| \__physicx_matrix_save_or_print:                 | \l__physicx_quantity_left_size_-             |
| . . . . . 993, 1391, 1472, 1475, 1577            | tl . . . . . 254, 349, 354, 358              |
| \l__physicx_matrix_save_tl . . . . .             | \l__physicx_quantity_left_tl 252, 339        |
| . . . . . 862, 864, 1477, 1480                   | \l__physicx_quantity_post_tl 251, 363        |
| \l__physicx_matrix_sep_dim . . . . .             | \l__physicx_quantity_pre_tl 250, 347         |
| . . . . . 720, 860, 1066, 1073,                  | \l__physicx_quantity_right_size_-            |
| 1080, 1104, 1416, 1466, 1532, 1627               | tl . . . . . 255, 350, 355, 360              |
| \__physicx_matrix_set_data:nn . . . . .          | \l__physicx_quantity_right_tl . . . . .      |
| . . . . . 837, 843, 952                          | . . . . . 253, 342                           |
| \__physicx_matrix_set_r_c_-                      | \__physicx_qxmatrix_appto_-                  |
| ckall:nnn . . . . . 678, 827                     | body:nnn . . . . . 1000, 1004,               |
| \__physicx_matrix_set_r_c_-                      | 1044, 1050, 1057, 1083, 1088, 1094           |
| ckep:nnn . . . . . 663, 818                      | \g__physicx_reqty_bool . . . . .             |
| \__physicx_matrix_set_r_c_-                      | . . . . . 13, 216, 520, 595                  |
| ckig:nnn . . . . . 658, 821                      | \l__physicx_row_list_seq . . . . .           |
| \__physicx_matrix_set_r_c_-                      | . . . . . 697, 750, 1437, 1441               |
| ckigep:nnn . . . . . 669, 679, 824               | \g__physicx_short_bool . . . . . 12, 21, 211 |
| \__physicx_matrix_set_r_c_-                      | \l__physicx_tmp_col_seq . . . . . 1521, 1524 |
| nock:nnn . . . . . 656, 681, 815                 | \l__physicx_tmp_coled_seq . . . . .          |
| \l__physicx_matrix_tmparr_c_sep .                | . . . . . 1523, 1528, 1538, 1550             |
| . . . . . 1309, 1312, 1313                       | \l__physicx_tmp_colnum_seq . . . . .         |
| \l__physicx_matrix_tmparr_r_sep .                | . . . . . 1272, 1278, 1287                   |
| . . . . . 1303, 1305, 1307                       | \l__physicx_tmp_rownum_seq . . . . .         |
| \__physicx_matrix_transpose:N . . . . .          | . . . . . 1270, 1276, 1285                   |
| . . . . . 1388, 1598, 1671                       | \l__physicx_tmp_seq . . . . .                |
| \l__physicx_matrix_transpose_-                   | . . . . . 1496, 1499, 1503, 1511             |
| bool . . . . . 718, 865, 1673                    | \l__physicx_tmp_tl . . . . .                 |
| \l__physicx_max_int . . . . .                    | . . . . . 442, 456, 468, 1610, 1616,         |
| . . . . . 51, 68, 99, 102, 104, 114, 125         | 1619, 1625, 1631, 1637, 1640, 1646           |
| \l__physicx_min_int . . . . .                    | \l__physicx_tmpa_bool 38, 170, 177, 182      |
| . . . . . 52, 67, 89, 92, 94, 115                | \l__physicx_tmpa_int . . . . .               |
| \__physicx_nauto_case:nnnn . 483, 506            | . . . . . 39, 1495, 1505, 1507, 1513, 1515   |
| \__physicx_new_matrix_cmd:NNN . 1743             | \l__physicx_tmpa_seq . . . . .               |
| \__physicx_new_xquantity_aux:w . .               | . . . . . 73, 83, 117, 120, 131, 141         |
| . . . . . 371, 377, 381, 385, 388                | \l__physicx_tmpa_tl . . . . .                |
| \__physicx_parse_range_aux:n 71, 80              | . . . . . 87, 88, 91, 97, 98, 101, 268, 269  |
| \__physicx_parse_range_range: . . . . .          | \l__physicx_tmparr_tl . . . . .              |
| . . . . . 57, 62, 107, 144                       | . . . . . 1300, 1301, 1302, 1303             |
| \__physicx_parse_range_range_-                   | \l__physicx_tmpb_int . . . . . 40            |
| check: . . . . . 57, 122, 144                    | \l__physicx_tmpb_seq . . . . . 86, 87, 97    |
| \__physicx_parse_range_range_-                   | \l__physicx_tmpdiag_seq 1158, 1169,          |
| nocheck: . . . . . 62, 134                       | 1179, 1182, 1183, 1193, 1199, 1200,          |
| \__physicx_parse_range_single:n .                | 1202, 1208, 1209, 1217, 1218, 1219,          |
| . . . . . 56, 61, 109, 121                       | 1229, 1230, 1231, 1240, 1241, 1242           |

|  |  |
|--|--|
| \l__physicx_tmpdiag_tl .....                                     | 1155, 1156, 1157, 1158   |
| \l__physicx_tmpitem_tl .....                                     | 1266, 1267, 1268, 1274, 1292   |
| \__physicx_xquantity_aux:nnnn .....                              | 338, 345, 366  |
| \physicxempty ..   | 652, 666, 674, 711, 712, 811   |
| \physicxexcept ..  | 714, 782, 791, 800, 806, 808   |
| \PHYSICXIGNORE .....   | 7, 8, 660, 671, 1274   |
| \physicxset .....  | 240  |
| \physicxtmp ....   | 408, 990, 1359, 1457, 1563   |
| \poissonbracket .....  | 595  |
| \pqty .....  | 595  |
| prg commands:  |  |
| \prg_generate_conditional_-variant:Nnn .....                     | 186  |
| \prg_new_conditional:Npnn .....                                  | 14, 19, 24, 29, 147, 152, 168, 1331  |
| \prg_replicate:nn .....  | 1076, 1106   |
| \prg_return_false: .....   | 17, 22, 27, 35, 150, 155, 183, 1352  |
| \prg_return_true: .....  | 17, 22, 27, 34, 150, 155, 183, 1334, 1351  |
| \ProcessKeysPackageOptions .....                                 | 222  |
| <b>Q</b>   |  |
| \qty .....   | 595  |
| \quantity .....  | 520, 598   |
| quark commands:  |  |
| \quark_if_recursion_tail_stop:n .....                            | 476, 477   |
| \q_recursion_stop .....  | 429  |
| \q_recursion_tail .....  | 429  |
| \qxmatrix .....  | 972, 1584  |
| \qxqty .....   | 407  |
| <b>R</b>   |  |
| \rangle .....  | 299, 528, 555, 570   |
| regex commands:  |  |
| \regex_match:nnTF ..   | 149, 154, 791, 800   |
| \RequirePackage .....  | 213, 214, 215  |
| \rgroup .....  | 552, 567   |
| \right .....   | 204, 330, 355, 545, 552, 555, 561, 562, 567, 569, 570, 571   |
| \rVert .....   | 612  |
| <b>S</b>   |  |
| \sb ..   | 702, 1043, 1048, 1055, 1082, 1086, 1092  |
| scan commands:   |  |
| \scan_stop: .....  | 1063, 1530   |
| seq commands:  |  |
| \c_empty_seq .....   | 66, 1523   |
| \seq_clear:N .....   | 83   |
| \seq_concat:NNN .....  | 73   |
| \seq_count:N ....  | 1182, 1183, 1199, 1200, 1208, 1209, 1217, 1218, 1229, 1230, 1240, 1241, 1305, 1312, 1499   |
| \seq_if_empty:NTF .....  | 1437, 1443   |
| \seq_item:Nn .....   | 1441, 1447   |
| \seq_map_indexed_inline:Nn .....                                 | 1169, 1179, 1193, 1202, 1219, 1231, 1242, 1307, 1313, 1524   |
| \seq_map_inline:Nn .....   | 1276, 1278, 1285, 1287   |
| \seq_map_tokens:Nn .....   | 1503, 1511   |
| \seq_new:N .....   | 697, 698   |
| \seq_pop_left:NN .....   | 87, 97   |
| \seq_put_right:Nn .....  | 117, 120, 131, 141, 1538, 1550   |
| \seq_set_eq:NN .....   | 66, 1523   |
| \seq_set_split:Nnn .....   | 5, 86, 750, 752, 1158, 1303, 1309, 1496, 1521  |
| \seq_use:Nn .....  | 1528   |
| \setmatrixdata .....   | 950  |
| \setmatrixtype ....  | 891, 913, 923, 924, 925, 926, 927, 928, 929, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948                               |
| \setquantitytype ....  | 265, 292, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325 |
| \smallmatrixquantity .....                                       | 520  |
| \smqty .....   | 595  |
| str commands:  |  |
| \c_backslash_str .....   | 269  |
| \str_case_e:nnTF .....   | 1127   |
| \str_const:Nn .....  | 683, 684, 685  |
| \str_if_eq:nnTF .....  | 461, 463   |
| \symbf .....   | 201  |
| \symbfite .....  | 197  |
| <b>T</b>   |  |
| TeX and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands: |  |
| \@declareparencmd .....  | 574  |
| \@declarequantitycmd .....                                       | 408  |
| \@ifpackageloaded .....  | 224, 227, 237  |
| \@ifstar .....   | 393  |
| \c@MaxMatrixCols .....   | 869, 1484, 1485  |
| \physicx@align .....   | 683, 875, 1309, 1468, 1522   |
| \physicx@cr ..   | 684, 874, 1303, 1466, 1496   |
| \physicx@matricelement .....                                     | 702, 757, 761, 762, 770, 771, 779, 780, 788, 789, 797, 798, 805, 1542, 1552, 1654, 1665  |
| \physicx@sep ..  | 685, 750, 752, 876, 1158   |

|  |  |
|--|--|
| tex commands:                          | 864, 1155, 1156, 1266, 1267, 1300,<br>1301, 1458, 1610, 1619, 1631, 1640   |
| \tex_advance:D . . . . .               | 1113, 1114   |
| text commands:                         | \tl_set_eq:NN 233, 234, 254, 255, 1469   |
| \text_expand:n . . . . .               | 733, 1540, 1652  |
| tl commands:                           | \tl_tail:N . . . . . 272, 275  |
| \c_empty_tl . . . . .                  | 257, 500   |
| \tl_clear:N . . . . .                  | 417, 418, 419, 1008, 1494  |
| \tl_const:Nn . . . . .                 | 189, 190   |
| \tl_gset_eq:NN . . . . .               | 1479   |
| \tl_head:N . . . . .                   | 268  |
| \tl_if_empty:NNTF . . . . .            | 88, 98, 1568, 1691, 1697   |
| \tl_if_empty:nTF . . . . .             | 337, 665, 673, 764, 1364, 1454   |
| \tl_if_empty_p:N . . . . .             | 349, 350, 1704, 1705   |
| \tl_if_eq:nnTF . . . . .               | 660, 671, 1273   |
| \tl_if_head_eq_charcode:NNTF . . . . . | 1159   |
| \tl_if_head_eq_meaning_p:nN . . . . .  | 1338, 1339, 1343, 1348   |
| \tl_if_in:nnTF . . . . .               | 84, 146, 773, 782  |
| \tl_if_novalue_p:n . . . . .           | 445  |
| \tl_new:N . . . . .                    | 408,<br>409, 411, 413, 689, 690, 693, 704,<br>705, 712, 714, 715, 716, 717, 722, 723   |
| \tl_put_right:Nn . . . . .             | 441, 456, 457, 468, 469, 808, 856,<br>859, 886, 889, 1043, 1048, 1049,<br>1055, 1056, 1061, 1066, 1073, 1075,<br>1077, 1079, 1080, 1082, 1086, 1087,<br>1092, 1093, 1098, 1104, 1105, 1107,<br>1109, 1110, 1405, 1412, 1421, 1428,<br>1526, 1617, 1626, 1638, 1650, 1663 |
| \tl_replace_all:Nnn . . . . .          | 6, 1465, 1467  |
| \tl_set:Nn . . . . .                   | 260, 268, 337, 442,<br>657, 661, 666, 667, 674, 675, 852,  |
| \tl_tail:n . . . . .                   | 1162   |
| token commands:                        | \token_if_cs:NNTF . . . . . 1477   |
| \token_if_eq_charcode:NNTF . . . . .   | 269  |
| \token_if_eq_meaning_p:NN . . . . .    | 354, 355   |
| \TrimSpaces . . . . .                  | 292, 917   |
| U                                      |  |
| use commands:                          | \use:n . . . . . 270, 278, 738   |
| \use:nn . . . . .                      | 1602, 1684   |
| \use:nnnn . . . . .                    | 4  |
| \use_i:nn . . . . .                    | 484, 1270, 1387, 1583  |
| \use_i:nnn . . . . .                   | 1583   |
| \use_i_ii:nnn . . . . .                | 1385, 1582   |
| \use_ii:nn . . . . .                   | 270, 484, 1272   |
| \use_ii_i:nn . . . . .                 | 1602, 1678   |
| \use_none:n . . . . .                  | 276  |
| \use_none:nn . . . . .                 | 709, 746, 1136   |
| V                                      |  |
| \vdots . . . . .                       | 773, 1075,<br>1077, 1079, 1105, 1107, 1109, 1740   |
| \Vert . . . . .                        | 298, 529, 571  |
| \vert . . . . .                        | 297, 527, 534, 535, 536, 569, 601, 605   |
| \vqty . . . . .                        | 595  |
| X                                      |  |
| \xquantity . . . . .                   | 365  |
| Z                                      |  |
| \Z . . . . .                           | 149, 154   |