

# The `physicx` package

Wenjian Chern (Longaster\*)

November 8, 2021, version v0.1

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

---

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

```

34         { \prg_return_false: }
35     }
36
37     \bool_new:N \l__physicx_tmpa_bool
38     \int_new:N \l__physicx_tmpa_int
39     \int_new:N \l__physicx_tmpb_int
40     \msg_new:nnnn { physicx } { unknown-key }
41     { The~key~'#1'~is-unknown~and~is-being-ignored. }
42     {
43         The~module~#2~does-not~have~a~key~called~#1.\\
44         Check~that~you~have~spelled~the~key~name~correctly.
45     }
46     \msg_new:nnn { physicx } { diag-key }
47     { 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:
48     \int_new:N \l__physicx_begin_int
49     \int_new:N \l__physicx_end_int
50     \int_new:N \l__physicx_max_int
51     \int_new:N \l__physicx_min_int
52     \bool_new:N \l__physicx_invalid_range_bool
53     \cs_new_protected:Npn \physicx_parse_range_check:
54     {
55         \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_check:n
56         \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_check:
57     }
58     \cs_new_protected:Npn \physicx_parse_range_nocheck:
59     {
60         \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_nocheck:n
61         \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_nocheck:
62     }
63     \cs_new_protected:Npn \physicx_parse_range:nnnN #1#2#3#4
64     {
65         \seq_set_eq:NN #4 \c_empty_seq
66         \int_set:Nn \l__physicx_min_int {#1}
67         \int_set:Nn \l__physicx_max_int {#2}
68         \clist_map_inline:nn {#3}
69         {
70             \__physicx_parse_range_aux:n {##1}
71             \bool_if:NF \l__physicx_invalid_range_bool
72             { \seq_concat:NNN #4 #4 \l__physicx_tmpa_seq }
73         }
74     }
75     \cs_generate_variant:Nn \physicx_parse_range:nnnN { nnvN, nnxN }
76     \cs_new_protected:Npn \physicx_parse_range:nnN
77     { \physicx_parse_range:nnnN { 1 } }
78     \cs_generate_variant:Nn \physicx_parse_range:nnN { nvN, nxN }
79     \cs_new_protected:Npn \__physicx_parse_range_aux:n #1
80     {
81         \bool_set_false:N \l__physicx_invalid_range_bool
82         \seq_clear:N \l__physicx_tmpa_seq
83         \tl_if_in:nnTF {#1} { - }

```

```

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

```

```

138         \int_step_inline:nnn
139         { \l__physicx_begin_int } { \l__physicx_end_int }
140         { \seq_put_right:Nn \l__physicx_tmpa_seq {##1} }
141     }
142 }
143 \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 ??.)

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

186 \tl_const:Nn \c_physicx_order_tl { \mathcal{o} }
187 \tl_const:Nn \c_physicx_Order_tl { \mathcal{O} }
188 \cs_new:Npn \physicx_use_amssymb_type:

```

```

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

```

`\physicsxset` `physicsx` setup command.

```

233 \NewDocumentCommand \physicsxset { s m }
234 {
235   \IfBooleanTF {#1}
236   { \keys_set:nn { physicsx/#2 } }
237   { \keys_set:nn { physicsx } {#2} }
238 }

```

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

## 1.2 Quantity things

\physicx\_declare\_legacy\_quantity:nnNn

\@declarequantitycmd

```

239 \tl_new:N \physicx_tmp
240 \tl_new:N \l__physicx_cmd_noauto_body_tl
241 \bool_new:N \l__physicx_cmd_noauto_body_bool
242 \tl_new:N \l__physicx_cmd_auto_body_tl
243 \bool_new:N \l__physicx_cmd_auto_body_bool
244 \tl_new:N \l__physicx_cmd_arg_spec_tl
245 \int_new:N \l__physicx_cmd_arg_int
246 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
247 {
248   \tl_clear:N \l__physicx_cmd_noauto_body_tl
249   \tl_clear:N \l__physicx_cmd_auto_body_tl
250   \tl_clear:N \l__physicx_cmd_arg_spec_tl
251   \int_set:Nn \l__physicx_cmd_arg_int {#1}
252   \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
253   \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
254 }
255 % noauto, auto, cmd, body
256 \cs_new:Npn \physicx_declare_legacy_quantity:nnNn #1#2#3#4
257 {
258   \__physicx_declare_init:nnn { 3 } {#1} {#2}
259   \__physicx_declare_legacy_quantity_aux:nw #4
260   \q_recursion_tail \q_recursion_tail \q_recursion_stop
261   \__physicx_declare_legacy_quantity_aux:NcVVV
262   #3 { \cs_to_str:N #3 ~ body }
263   \l__physicx_cmd_arg_spec_tl
264   \l__physicx_cmd_noauto_body_tl
265   \l__physicx_cmd_auto_body_tl
266 }
267 % arg spec, pre, body to replace(start from #4), post
268 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
269 {
270   \int_incr:N \l__physicx_cmd_arg_int
271   \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
272     \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
273     \tl_set:Nx \l__physicx_tmp_tl
274     {
275       {
276         \exp_not:N \tl_if_novalue_p:n
277         {
278           \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:
279           \or: \or: \or:
280           \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
281           \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
282           \fi:
283         }
284       }
285     }
286   \if_bool:N \l__physicx_cmd_noauto_body_bool
287     \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
288     \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
289     {

```

```

290         {
291             % if is '.', use none
292             \str_if_eq:nnTF {#2} {.} {} {#2}
293             #3
294             \str_if_eq:nnTF {#4} {.} {} {#4}
295         }
296     }
297     \fi:
298     \if_bool:N \l__physicx_cmd_auto_body_bool
299         \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
300         \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
301         { { ##1 #2 #3 ##2 #4 } }
302     \fi:
303     \fi:
304 }
305 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
306 {
307     \quark_if_recursion_tail_stop:n {#1}
308     \quark_if_recursion_tail_stop:n {#2}
309     \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
310     \__physicx_declare_legacy_quantity_aux:nw
311 }
312 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
313 {
314     \__physicx_nauto_case:nnnn
315     { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
316     {
317         \cs_set_protected:Npn #1
318         {
319             \peek_charcode_ignore_spaces:NTF \let
320             { #2 } { #2 [ \physicx_left: ] \physicx_right: }
321         }
322         \DeclareDocumentCommand #2 { 0{##2} m s #3 }
323         {
324             \IfBooleanTF { ##3 }
325             { \bool_case_false:n {#4} }
326             { \bool_case_false:n {#5} }
327         }
328     }
329     {
330         \cs_set_protected:Npn #1
331         { #2 \c_empty_tl \c_empty_tl }
332         \DeclareDocumentCommand #2 { m m s #3 }
333         { \bool_case_false:n {#4} }
334     }
335 }
336 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
337 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
338 {
339     \bool_if:NTF \l__physicx_cmd_noauto_body_bool
340     {
341         \bool_if:NTF \l__physicx_cmd_auto_body_bool
342         {#1} {#2}
343     }

```

```

344     {
345         \bool_if:NTF \l__physicx_cmd_auto_body_bool
346         {#3} {#4}
347     }
348 }
349 \cs_set_protected:Npn \@declarequantitycmd
350 { \physicx_declare_legacy_quantity:nnNn }

```

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

```

\quantity      Redefine some macros in physics package.
\evaluated
\matrixquantity
\smallmatrixquantity
351 \physicx_declare_legacy_quantity:nnNn
352 \c_true_bool \c_true_bool \quantity
353 {
354     { !g } { { \{ } { #4 } { \} } }
355     { !o } { { [ } { #5 } { ] } }
356     { !d() } { { ( } { #6 } { ) } }
357     { !d|| } { { \vert } { #7 } { \vert } }
358     { !d<> } { { \langle } { #8 } { \rangle } }
359     { !d== } { { \Vert } { #9 } { \Vert } }
360 }
361 \physicx_declare_legacy_quantity:nnNn
362 \c_true_bool \c_true_bool \evaluated
363 {
364     { !g } { { . } { #4 \nobreak } { \vert } }
365     { !d[] } { { [ ] } { #5 \nobreak } { \vert } }
366     { !d( ) } { { ( ) } { #6 \nobreak } { \vert } }
367 }
368 \physicx_declare_legacy_quantity:nnNn
369 \c_true_bool \c_false_bool \matrixquantity
370 {
371     { !g }
372     {
373         { \IfBooleanT{#3}{\left\{ } }
374         { \begin{matrix} #4 \end{matrix} }
375         { \IfBooleanT{#3}{\right\} }
376     }
377     { !o } { { \begin{bmatrix} } { #5 } { \end{bmatrix} } }
378     { !d() }
379     {
380         { \IfBooleanTF{#3}{\left\lgroup}{\left( } }
381         { \begin{matrix} #6 \end{matrix} }
382         { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
383     }
384     { !d|| } { { \begin{vmatrix} } { #7 } { \end{vmatrix} } }
385     { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } { \right\rangle } }
386     { !d== } { { \begin{Vmatrix} } { #9 } { \end{Vmatrix} } }
387 }
388 \physicx_declare_legacy_quantity:nnNn
389 \c_true_bool \c_false_bool \smallmatrixquantity
390 {
391     { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } { \right\} } }
392     { !o } { { \left[ ] } { \begin{smallmatrix} #5 \end{smallmatrix} } { \right]} }

```



```

393 { !d() }
394 {
395   { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
396   { \begin{smallmatrix} #6 \end{smallmatrix} } }
397   { \IfBooleanTF{#3}{\right\rgroup}{\right)} } }
398 }
399 { !d|| } { {\left\vert} } { \begin{smallmatrix} #7 \end{smallmatrix} } {\right\vert} }
400 { !d< } { {\left\langle} } { \begin{smallmatrix} #8 \end{smallmatrix} } {\right\rangle} }
401 { !d== } { {\left\Vert} } { \begin{smallmatrix} #9 \end{smallmatrix} } {\right\Vert} }
402 }

```

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

```

\physics_declare_legacy_paren:NnnnNn
\@declareparencmd
403 %% cmd, arg spec, replace(start from #6), pre, left, right, post
404 \cs_new:Npn \physics_declare_legacy_paren:NnnnNn #1#2#3#4#5#6#7
405 {
406   \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
407   {
408     \bool_case_true:nF
409     {
410       { \bool_if_p:n {##2} } { #4 \physics_left:N \bigl #5 #3 \physics_right:N \bigr
411       { \bool_if_p:n {##3} } { #4 \physics_left:N \Bigl #5 #3 \physics_right:N \Bigr
412       { \bool_if_p:n {##4} } { #4 \physics_left:N \biggl #5 #3 \physics_right:N \biggr
413       { \bool_if_p:n {##5} } { #4 \physics_left:N \Biggl #5 #3 \physics_right:N \Biggr
414     }
415     {
416       \IfBooleanTF {##1}
417       { #4 #5 #3 #6 #7 }
418       { #4 \physics_left: #5 #3 \physics_right: #6 #7 }
419     }
420   }
421 }
422 \cs_set_protected:Npn \@declareparencmd
423 { \physics_declare_legacy_paren:NnnnNn }

```

(End definition for \physics\_declare\_legacy\_paren:NnnnNn and \@declareparencmd. These functions are documented on page ??.)

```

\qty Redefine some macros in physics package.
\mqty \physics_option_or:nnT { compat } { short }
\smqty {
424   \cs_set:Npn \qty { \quantity }
425   \physics_declare_legacy_paren:NnnnNn \pqty { m } {#6} { } { } { }
426   \physics_declare_legacy_paren:NnnnNn \bqty { m } {#6} { } { } { }
427   \physics_declare_legacy_paren:NnnnNn \vqty { m } {#6} { } { } \vert \vert { }
428   \physics_declare_legacy_paren:NnnnNn \Bqty { m } {#6} { } { } \{ \} { }
429 }
\absolutevalue
430 \eval \physics_declare_legacy_paren:NnnnNn \absolutevalue
431 \abs { m } {#6} { } { } \vert \vert { }
432 \norm \physics_option_or:nnT { compat } { short }
433 {
434   \cs_set:Npn \eval { \evaluated }
435   \cs_set:Npn \abs { \absolutevalue }
436 }
\commutator
437 \poissonbracket
438 \pb
\anticommutator
\acomm

```

```

439 \physicx_declare_legacy_paren:NnnnNNn \norm
440 { m } {#6} { } \lVert \rVert { }
441 \physicx_compat:TF
442 {
443   \physicx_declare_legacy_paren:NnnnNNn \order
444   { m } {#6} { \c_physicx_Order_tl } ( ) { }
445   \physicx_declare_legacy_paren:NnnnNNn \oorder
446   { m } {#6} { \c_physicx_order_tl } ( ) { }
447   \cs_set:Npn \Order { \order }
448   \cs_set:Npn \OOrder { \order }
449 }
450 {
451   \physicx_declare_legacy_paren:NnnnNNn \Order
452   { m } {#6} { \c_physicx_Order_tl } ( ) { }
453   \physicx_declare_legacy_paren:NnnnNNn \order
454   { m } {#6} { \c_physicx_order_tl } ( ) { }
455   \cs_set:Npn \oorder { \order }
456   \cs_set:Npn \OOrder { \Order }
457 }
458 \physicx_declare_legacy_paren:NnnnNNn \commutator
459 { m m } { #6 , #7 } { } [ ] { }
460 \physicx_option_or:nnT { compat } { short }
461 { \cs_set:Npn \comm { \commutator } }
462 \physicx_declare_legacy_paren:NnnnNNn \poissonbracket
463 { m m } { #6 , #7 } { } \{ \} { }
464 \physicx_option_or:nnT { compat } { short }
465 {
466   \cs_set:Npn \pb { \poissonbracket }
467   \cs_set:Npn \anticommutator { \poissonbracket }
468   \cs_set:Npn \acomm { \poissonbracket }
469 }

```

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

## 1.3 Matrix things

### 1.3.1 Matrix auxillary functions

```

470 \cs_new_nopar:Npn \__physicx_matrix_calc:nn #1#2
471 {
472   \int_set:Nn \l__physicx_matrix_rows_int
473   { \int_max:nn {#1} \l__physicx_matrix_rows_int }
474   \int_set:Nn \l__physicx_matrix_cols_int
475   { \int_max:nn {#2} \l__physicx_matrix_cols_int }
476 }
477 % use matrix element
478 \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
479 {
480   \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
481   \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
482   \else:
483   \exp_not:o { \physicxempty }
484   \fi:
485 }
486 % set matrix element, check or not

```

```

487 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
488 { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
489 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
490 {
491   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
492   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
493 }
494 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_cke:nnn #1#2#3
495 {
496   \tl_if_empty:nTF {#3}
497   { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
498   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
499 }
500 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
501 {
502   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
503   {
504     \tl_if_empty:nTF {#3}
505     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
506     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
507   }
508 }
509 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
510 \__physicx_matrix_set_r_c_ckigep:nnn
511 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
512 \__physicx_matrix_set_r_c_nock:nnn
513 % align, cr, sep symbol
514 \str_const:Nn \physicx@align { , }
515 \str_const:Nn \physicx@cr { ; }
516 \str_const:Nn \physicx@sep { , }
517 \bool_new:N \l__physicx_matrix_infinite_bool
518 \bool_new:N \l__physicx_matrix_dotrow_bool
519 \bool_new:N \l__physicx_matrix_dotcol_bool
520 \tl_new:N \l__physicx_matrix_array_tl
521 \tl_new:N \l__physicx_matrix_body_tl
522 \int_new:N \l__physicx_matrix_rows_int
523 \int_new:N \l__physicx_matrix_cols_int
524 \tl_new:N \l__physicx_matrix_main_tl
525 \clist_new:N \l__physicx_matrix_diag_clist
526 \clist_new:N \l__physicx_matrix_item_clist
527 \bool_new:N \l__physicx_matrix_diag_bool
528 \seq_new:N \l__physicx_row_list_seq
529 \seq_new:N \l__physicx_col_list_seq
530 % expand input
531 \cs_new_eq:NN \__physicx_expand:w \exp_not:o
532 %% main, row iterate, col iterate
533 \cs_new_nopar:Npn \physicx@matricelement #1#2#3 { #1 \sb { #2 #3 } }
534 \cs_new_nopar:Npn \__physicx_matrix_row_iterate:n #1 { #1 }
535 \tl_new:N \l__physicx_matrix_last_row_tl
536 \tl_new:N \l__physicx_matrix_last_col_tl
537 \cs_new_nopar:Npn \__physicx_matrix_col_iterate:n #1 { #1 }
538 \cs_new_nopar:Npn \__physicx_matrix_begin:w { }
539 \cs_new_nopar:Npn \__physicx_matrix_end:w { }
540 \cs_new_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn

```

```

541 \bool_new:N \l__physicx_matrix_expand_element_bool
542 % when element is empty use \physicxempty
543 \tl_new:N \physicxempty
544 % save 'element-except' key's value
545 \tl_new:N \physicxexcept
546 \tl_new:N \l__physicx_matrix_args_tl
547 \tl_new:N \l__physicx_matrix_after_begin_tl
548 \tl_new:N \l__physicx_matrix_after_end_tl
549 \bool_new:N \l__physicx_matrix_transpose_bool
550 \bool_new:N \l__physicx_matrix_enhanced_bool
551 \dim_new:N \l__physicx_matrix_sep_dim
552 \cs_new:Npn \__physicx_adi:nnn #1#2#3 { #1#2#3 }
553 \tl_new:N \l__physicx_matrix_beginning_tl
554 \tl_new:N \l__physicx_matrix_ending_tl

```

### 1.3.2 Matrix keys

```

555 \keys_define:nn { physicx }
556 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
557 \keys_define:nn { physicx/matrix }
558 {
559   array .tl_set:N = \l__physicx_matrix_array_tl ,
560   expand .choice: ,
561   expand / none .code:n =
562     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
563   expand / text-expand .code:n =
564     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
565   expand / f .code:n =
566     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
567   expand / romanual .meta:n = { expand = f } ,
568   expand / x .code:n =
569     \cs_set_eq:NN \__physicx_expand:w \use:n ,
570   expand / edef .meta:n = { expand = x } ,
571   rows .int_set:N = \l__physicx_matrix_rows_int ,
572   cols .int_set:N = \l__physicx_matrix_cols_int ,
573   auto-update .choice: ,
574   auto-update / true .code:n =
575     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,
576   auto-update / false .code:n =
577     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
578   auto-update .default:n = true ,
579   main .tl_set:N = \l__physicx_matrix_main_tl ,
580   row-list .code:n =
581     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,
582   col-list .code:n =
583     \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
584   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
585   infinite .default:n = true ,
586   !infinite .code:n =
587     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
588   element-code .cs_set:Np = \physicx@matricelement #1#2#3 ,
589   element-code* .choice: ,
590   element-code* / except-empty .code:n =
591     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
592     \physicx@matricelement

```

```

593 \cs_set:Npn \physicsx@matricelement ##1##2##3
594 {
595   \tl_if_empty:nTF {##1}
596     {##1}
597     { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
598   } ,
599 element-code* / except-dots .code:n =
600 \cs_set_eq:NN \_physicsx_matrix_element_aux:nnn
601 \physicsx@matricelement
602 \cs_set:Npn \physicsx@matricelement ##1##2##3
603 {
604   \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
605     {##1}
606     { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
607   } ,
608 element-code* / except-tl .code:n =
609 \cs_set_eq:NN \_physicsx_matrix_element_aux:nnn
610 \physicsx@matricelement
611 \cs_set:Npn \physicsx@matricelement ##1##2##3
612 {
613   \tl_if_in:onTF { \physicsxexcept } {##1}
614     {##1}
615     { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
616   } ,
617 element-code* / except-regex .code:n =
618 \cs_set_eq:NN \_physicsx_matrix_element_aux:nnn
619 \physicsx@matricelement
620 \cs_set:Npn \physicsx@matricelement ##1##2##3
621 {
622   \exp_args:No \regex_match:nnTF { \physicsxexcept } {##1}
623     {##1}
624     { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
625   } ,
626 element-code* / only-regex .code:n =
627 \cs_set_eq:NN \_physicsx_matrix_element_aux:nnn
628 \physicsx@matricelement
629 \cs_set:Npn \physicsx@matricelement ##1##2##3
630 {
631   \exp_args:No \regex_match:nnTF { \physicsxexcept } {##1}
632     { \_physicsx_matrix_element_aux:nnn {##1} {##2} {##3} }
633     {##1}
634   } ,
635 element-code* / unknown .code:n =
636 \cs_set:Npx \physicsx@matricelement { \exp_not:c {#1} } ,
637 element-except .tl_set:N = \physicsxexcept ,
638 element-except+ .code:n =
639 \tl_put_right:Nn \physicsxexcept {#1} ,
640 expand-element .bool_set:N = \l_physicsx_matrix_expand_element_bool ,
641 expand-element .default:n = true ,
642 empty .tl_set:N = \physicsxempty ,
643 check .choice: ,
644 check / none .code:n =
645 \cs_set_eq:NN \physicsx_matrix_set_r_c:nnn
646 \_physicsx_matrix_set_r_c_nock:nnn ,

```

```

647 check / empty .code:n =
648     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
649     \__physicx_matrix_set_r_c_cke:nnn ,
650 check / ignore .code:n =
651     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
652     \__physicx_matrix_set_r_c_ckig:nnn ,
653 check / igep .code:n =
654     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
655     \__physicx_matrix_set_r_c_ckigep:nnn ,
656 check / all .code:n =
657     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
658     \__physicx_matrix_set_r_c_ckall:nnn ,
659 check .default:n = all ,
660 row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
661 col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
662 last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
663 last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
664 diag .clist_set:N = \l__physicx_matrix_diag_clist ,
665 diag+ .code:n =
666     \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
667 diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
668 diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
669 diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
670 item .clist_set:N = \l__physicx_matrix_item_clist ,
671 item+ .code:n =
672     \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
673 item-now .code:n = \physicx_matrix_item_parse:n {#1} ,
674 item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
675 item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
676 check-range .choice: ,
677 check-range / true .code:n = \physicx_parse_range_check: ,
678 check-range / false .code:n = \physicx_parse_range_nocheck: ,
679 check-range .default:n = true ,
680 begin .tl_set:N = \__physicx_matrix_begin:w ,
681 end .tl_set:N = \__physicx_matrix_end: ,
682 args .code:n =
683     \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,
684 args* .tl_set:N = \l__physicx_matrix_args_tl ,
685 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
686 after-begin+ .code:n =
687     { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
688 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
689 after-end+ .code:n =
690     { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
691 sepdim .dim_set:N = \l__physicx_matrix_sep_dim ,
692 type .multichoice: ,
693 saveto .tl_set:N = \l__physicx_matrix_save_tl ,
694 saveto* .code:n =
695     \tl_set:Nn \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
696 transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
697 transpose .default:n = true ,
698 ' .meta:n = { transpose = true } ,
699 T .meta:n = { transpose = true } ,
700 MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,

```

```

701     enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
702     enhanced .default:n = true ,
703     !enhanced .code:n =
704         \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
705     cr .tl_set:N = \physicx@cr ,
706     align .tl_set:N = \physicx@align ,
707     sep .tl_set:N = \physicx@sep ,
708     adi-order .choice: ,
709     adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
710     adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
711     adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
712     adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
713     adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
714     adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
715     beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
716     beginning+ .code:n =
717         \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
718     ending .tl_set:N = \l__physicx_matrix_ending_tl ,
719     ending+ .code:n =
720         \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
721
722     unknown .code:n =
723         \physicx_search_also:nnF
724         {
725             physicx/matrix/type ,
726             physicx/matrix/expand ,
727             physicx/matrix/element-code* ,
728         }
729         {#1}
730         {
731             \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
732             {
733                 \keys_set:nx { physicx/matrix }
734                 { MaxMatrixCols = \l_keys_key_str }
735             }
736             {
737                 \msg_error:nnxx { physicx } { unknown-key }
738                 \l_keys_path_str { physicx }
739             }
740         } ,
741     }

```

```

\physicx_matrix_new_type:nnn
\physicx_matrix_new_type:nn
\setmatrixtype

```

```

742 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
743 {
744     \keys_define:nn { physicx/matrix }
745     { type / #1 .meta:n = { begin={#2} , end={#3} } }
746 }
747 \cs_new:Npn \physicx_matrix_new_type:nn #1#2
748 {
749     \keys_define:nn { physicx/matrix }
750     { type / #1 .meta:n = {#2} }
751 }
752 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }

```

```

753 {
754   \IfBooleanTF {#1}
755     { \physicx_matrix_new_type:nn {#2} }
756     { \physicx_matrix_new_type:nnn {#2} }
757 }

```

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

A few types.

```

758 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
759 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
760 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
761 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
762 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
763 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
764 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
765 \physicx_mathtools:T
766 {
767   \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
768   \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
769   \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
770   \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
771   \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
772   \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
773   \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
774   \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
775   \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
776   \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
777   \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
778   \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
779   \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
780   \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
781   \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
782   \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
783   \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
784 }

```

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

```

785 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
786 { \clist_set:cn { physicx@ #1 data@ #2 } }
787 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
788 {
789   \clist_clear:c { l__physicx_matrix_ #1 _clist }
790   \__physicx_matrix_add_data:nn {#1} {#2}
791 }
792 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
793 {
794   \clist_map_inline:nn {#2}
795   {
796     \clist_concat:ccc
797       { l__physicx_matrix_ #1 _clist }
798       { l__physicx_matrix_ #1 _clist }
799       { physicx@ #1 data@ #2 }
800   }

```



```

801 }

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

Initial settings.

802 \keys_set:nn { physicx/matrix }
803 {
804     type = m ,
805     saveto = ? ,
806 }

\qxmatrix
807 %% basicly, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
      xmatnmn-in-the-physics-package, but changed some
808 % #1 = boolean, saveto matrix
809 % #2 = star, infinite
810 % #3 = options
811 % #4 = letter for the entries
812 % #5 = number of rows
813 % #6 = number of explicit rows, default = 3
814 % #7 = number of columns
815 % #8 = number of explicit columns, default = 3
816 \DeclareDocumentCommand \qxmatrix { t= s 0{type=p} m m 0{3} m 0{3} }
817 {
818     \group_begin:
819     \IfBooleanTF { #2 }
820     { \bool_set_true:N \l__physicx_matrix_infinite_bool }
821     { \bool_set_false:N \l__physicx_matrix_infinite_bool }
822     \int_set:Nn \l__physicx_matrix_rows_int {#6}
823     \int_set:Nn \l__physicx_matrix_cols_int {#8}
824     \IfBooleanTF {#1}
825     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
826     { \keys_set:nn { physicx/matrix } {#3} }
827     \physicx_qxmatrix:nnn {#4} {#5} {#7}
828     \__physicx_matrix_save_or_print:
829     \group_end:
830 }
831 \cs_new_protected:Nn \physicx_qxmatrix:nnn
832 {
833     \bool_if:NTF \l__physicx_matrix_expand_element_bool
834     {
835         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
836         \__physicx_matrix_appto_body_e:nnn
837     }
838     {
839         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
840         \__physicx_matrix_appto_body_ne:nnn
841     }
842     % clear the variable containing the body of the matrix
843     \tl_clear:N \l__physicx_matrix_body_tl
844     % set the tentative number of explicit rows
845     \physicx_if_num:nTF { #2 }
846     {% number of rows is an integer
847         \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
848         {% if #2 <= rows, we don't want a row of dots

```

```

849         \bool_set_false:N \l__physicx_matrix_dotrow_bool
850         \int_set:Nn \l__physicx_matrix_rows_int { #2 }
851     }
852     {% we want a row of dots
853         \bool_set_true:N \l__physicx_matrix_dotrow_bool
854     }
855 }
856 {% number of rows is symbolic, we want a row of dots
857     \bool_set_true:N \l__physicx_matrix_dotrow_bool
858 }
859 % set the tentative number of explicit columns
860 \physicx_if_num:nTF { #3 }
861 {
862     {% number of cols is an integer
863         \int_compare:nTF { #3 <= \l__physicx_matrix_cols_int }
864         {
865             {% if #3 <= cols, we don't want a column of dots
866                 \bool_set_false:N \l__physicx_matrix_dotcol_bool
867                 \int_set:Nn \l__physicx_matrix_cols_int { #3 }
868             }
869             {% we want a column of dots
870                 \bool_set_true:N \l__physicx_matrix_dotcol_bool
871             }
872         }
873         {% number of columns is symbolic, we want a column of dots
874             \bool_set_true:N \l__physicx_matrix_dotcol_bool
875         }
876     }
877     % loop through the rows
878     \int_step_inline:nn { \l__physicx_matrix_rows_int }
879     {
880         % add the first entry in the row
881         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
882         \__physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
883         % add the further entries in the explicit columns
884         \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
885         {
886             %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
887             \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
888             \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
889         }
890         % if we have a column of dots, add \cdots and the last entry
891         \bool_if:NT \l__physicx_matrix_dotcol_bool
892         {
893             %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{##1 #3} }
894             \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
895             \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {#3}
896         }
897         % infinite matrix, add \cdots
898         \bool_if:NT \l__physicx_matrix_infinite_bool
899         {
900             \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots }
901         }
902         \if_int_compare:w ##1 = \l__physicx_matrix_rows_int
903         \scan_stop:
904         \else:
905             % finish up the row
906             \tl_put_right:Nx \l__physicx_matrix_body_tl { \\[\dim_use:N \l__physicx_matrix_sep
907         \fi:

```

```

903     }
904     % finish up the rows
905     \bool_if:NT \l__physicx_matrix_dotrow_bool
906     {
907         % finish up the row
908         \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
909         % if we have a row of dots, fill it in
910         \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
911         \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
912         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
913         \bool_if:NT \l__physicx_matrix_dotcol_bool
914         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
915         \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
916         % fill the last row
917         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
918         \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
919         \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
920         {
921             %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
922             \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
923             \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
924         }
925         \bool_if:NT \l__physicx_matrix_dotcol_bool
926         {
927             %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
928             \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
929             \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
930         }
931         % if the matrix is infinite, add a further column with \cdots
932         \bool_if:NT \l__physicx_matrix_infinite_bool
933         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
934     }
935     % if the matrix is infinite, add a final row
936     \bool_if:NT \l__physicx_matrix_infinite_bool
937     {
938         % finish up the row
939         \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
940         \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
941         \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
942         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
943         \bool_if:NT \l__physicx_matrix_dotcol_bool
944         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
945         \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
946         % update cols
947         \bool_if:NTF \l__physicx_matrix_dotcol_bool
948         { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
949         { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
950     }
951 }

```

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

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o 952 \cs_new:Npn \physicx_matrix_diag_parse:n #1

```

```

953 {
954   \keyval_parse:nnn
955   \__physicx_matrix_diag_parse_aux:n
956   \__physicx_matrix_diag_parse_aux:nn
957   {#1}
958 }
959 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
960 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
961 {
962   \str_case_e:nnF {#1}
963   {
964     { auto-update }
965     {
966       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
967       \__physicx_matrix_calc:nn
968     }
969     { noauto-update }
970     {
971       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
972     }
973     { true }
974     {
975       \bool_set_true:N \l__physicx_matrix_diag_bool
976       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
977       \__physicx_diagonalmatrix_set_diag:
978     }
979     { false }
980     {
981       \bool_set_false:N \l__physicx_matrix_diag_bool
982       \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
983       \__physicx_diagonalmatrix_no_diag:
984     }
985   }
986   { \msg_error:nnn { physicx } { diag-key } {#1} }
987 }
988 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
989 {
990   \tl_set:Nn \l__physicx_tmpdiag_tl {#2}
991   \tl_set:Nx \l__physicx_tmpdiag_tl
992   { \__physicx_expand:w \l__physicx_tmpdiag_tl }
993   \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl
994   \tl_if_head_eq_charcode:nNTF {#1} '
995   {
996     \exp_args:Nf \__physicx_matrix_diag_parse_aux_anti:n
997     { \tl_tail:n {#1} }
998   }
999   { \__physicx_matrix_diag_parse_aux_regu:n {#1} }
1000 }
1001 \cs_new:Npn \__physicx_diagonalmatrix_set_diag:
1002 {
1003   \int_zero:N \l__physicx_matrix_cols_int
1004   \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1005   {
1006     \int_incr:N \l__physicx_matrix_cols_int

```

```

1007         \physicx_matrix_set_r_c:nnn {##1} {##1} {##2}
1008     }
1009     \int_set_eq:NN \l__physicx_matrix_rows_int
1010     \l__physicx_matrix_cols_int
1011 }
1012 \cs_new:Npn \__physicx_diagonalmatrix_no_diag:
1013 {
1014     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1015     { \physicx_matrix_set_r_c:nnn {##1} {##1} {##2} }
1016     \__physicx_matrix_diag_calc:nn
1017     { \seq_count:N \l__physicx_tmpdiag_seq }
1018     { \seq_count:N \l__physicx_tmpdiag_seq }
1019 }
1020 \cs_new_eq:NN \__physicx_diagonalmatrix_diag_main:
1021     \__physicx_diagonalmatrix_no_diag:
1022 \cs_new:Npn \__physicx_matrix_diag_parse_aux_regu:n #1
1023 {
1024     \if_int_compare:w #1 = 0 \exp_stop_f:
1025         \__physicx_diagonalmatrix_diag_main:
1026     \else:
1027         \if_int_compare:w #1 > 0 \exp_stop_f:
1028             \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1029             {
1030                 \physicx_matrix_set_r_c:nnn
1031                 {##1} { \int_eval:n { ##1 + #1 } } {##2}
1032             }
1033             \__physicx_matrix_diag_calc:nn
1034             { \seq_count:N \l__physicx_tmpdiag_seq }
1035             { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1036         \else:
1037             \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1038             {
1039                 \physicx_matrix_set_r_c:nnn
1040                 { \int_eval:n { ##1 - #1 } } {##1} {##2}
1041             }
1042             \__physicx_matrix_diag_calc:nn
1043             { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1044             { \seq_count:N \l__physicx_tmpdiag_seq }
1045         \fi:
1046     \fi:
1047 }
1048 \cs_new:Npn \__physicx_matrix_diag_parse_aux_anti:n #1
1049 {
1050     \if_int_compare:w #1 = 0 \exp_stop_f:
1051         \__physicx_matrix_diag_calc:nn
1052         { \seq_count:N \l__physicx_tmpdiag_seq }
1053         { \seq_count:N \l__physicx_tmpdiag_seq }
1054     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1055     {
1056         \physicx_matrix_set_r_c:nnn
1057         {##1}
1058         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1059         {##2}
1060     }

```

```

1061 \else:
1062 \if_int_compare:w #1 > 0 \exp_stop_f:
1063 \__physicx_matrix_diag_calc:nn
1064 { \seq_count:N \l__physicx_tmpdiag_seq }
1065 { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1066 \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1067 {
1068 \physicx_matrix_set_r_c:nnn
1069 {##1}
1070 { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1071 {##2}
1072 }
1073 \else:
1074 \__physicx_matrix_diag_calc:nn
1075 { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1076 { \seq_count:N \l__physicx_tmpdiag_seq }
1077 \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1078 {
1079 \physicx_matrix_set_r_c:nnn
1080 { \int_eval:n { ##1 - #1 } }
1081 { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1082 {##2}
1083 }
1084 \fi:
1085 \fi:
1086 }
1087 \cs_new:Npn \__physicx_matrix_diag_calc:nn
1088 { \__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
1089 \cs_new:Npn \physicx_matrix_item_parse:n #1
1090 {
1091 \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1092 \keyval_parse:NNn
1093 \__physicx_matrix_item_parse_aux:n
1094 \__physicx_matrix_item_parse_aux:nn
1095 {#1}
1096 }
1097 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }
1098 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1099 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2
1100 {
1101 \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1102 \tl_set:Nx \l__physicx_tmpitem_tl
1103 { \__physicx_expand:w \l__physicx_tmpitem_tl }
1104 \physicx_parse_range:nxN \l__physicx_matrix_rows_int
1105 { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1106 \physicx_parse_range:nxN \l__physicx_matrix_cols_int
1107 { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1108 \exp_args:No \tl_if_eq:nnTF
1109 { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1110 {

```

```

1111     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1112     {
1113         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1114         {
1115             \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1116         }
1117     }
1118 }
1119 {
1120     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1121     {
1122         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1123         {
1124             \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }
1125             {
1126                 \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1127                 {##1} {####1} { \l__physicx_tmpitem_tl }
1128             }
1129         }
1130     }
1131 }
1132 }

```

(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

```

1133 \cs_new:Npn \physicx_matrix_array_parse:n #1
1134 {
1135     \tl_set:Nn \l__physicx_tmparr_tl {#1}
1136     \tl_set:Nx \l__physicx_tmparr_tl
1137     { \__physicx_expand:w \l__physicx_tmparr_tl }
1138     \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl
1139     \__physicx_matrix_autocalc:nn
1140     { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1141     { 0 }
1142     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1143     {
1144         \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1145         \__physicx_matrix_autocalc:nn
1146         { 0 }
1147         { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1148         \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1149         {
1150             \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1151         }
1152     }
1153 }
1154 \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.

```

1155 \cs_new:Npn \physicx_matrix_array_parse_main:
1156 {
1157     \int_step_inline:nn \l__physicx_matrix_rows_int

```

```

1158     {
1159         \int_step_inline:nn \l__physicx_matrix_cols_int
1160         {
1161             \exp_args:Nno \physicx_matrix_set_r_c:nnn
1162             {##1} {####1} \l__physicx_matrix_main_tl
1163         }
1164     }
1165 }

```

(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).

```

1166 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1167 {
1168     \physicx_if_num:nTF {#1}
1169     { \prg_return_true: }
1170     {
1171         \bool_case_true:nTF
1172         {
1173             { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1174             { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1175             {
1176                 \bool_lazy_and_p:nn
1177                 { \cs_if_exist_p:N \inteval }
1178                 { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1179             } { }
1180             {
1181                 \bool_lazy_and_p:nn
1182                 { \cs_if_exist_p:N \fpeval }
1183                 { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1184             } { }
1185         }
1186         { \prg_return_true: }
1187         { \prg_return_false: }
1188     }
1189 }

```

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

\diagonalmatrix Define \diagonalmatrix.

```

1190 \DeclareDocumentCommand \diagonalmatrix { t= t+ 0{ } m }
1191 {
1192     \group_begin:
1193     \IfBooleanTF {#1}
1194     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicx_tmp } }
1195     { \keys_set:nn { physicx/matrix } { #3 } }
1196     \physicx_construct:nnn { }
1197     {
1198         \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist
1199         \tl_if_empty:nF {#4}
1200         {
1201             \__physicx_if_keyval:nTF {#4}
1202             { \physicx_matrix_diag_parse:n { true, #4 } }

```



```

1203         { \physicx_matrix_diag_parse:n { true, 0 = {#4} } }
1204     }
1205 }
1206 { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1207 \bool_lazy_or:nnTF
1208 { \bool_if_p:n {#2} }
1209 { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1210 {
1211     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1212     {
1213         \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1214         \__physicx_matrix_appto_body_e:off
1215     }
1216     {
1217         \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1218         \__physicx_matrix_appto_body_ne:off
1219     }
1220     \use_i_ii:nnn
1221 }
1222 { \use_i:nn }
1223 \__physicx_matrix_transpose:N
1224 \__physicx_diagonalmatrix_generate_enhanced_body:NNN
1225 \__physicx_diagonalmatrix_generate_body:NNN
1226 \__physicx_matrix_save_or_print:
1227 \group_end:
1228 }
1229 \cs_new:Npn \__physicx_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1230 {
1231     \__physicx_matrix_generate_body:NNNN #1#2#3
1232     \__physicx_diagonalmatrix_enhanced:nnn
1233 }
1234 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1235 {
1236     \int_step_inline:nn { #1 - 1 }
1237     {
1238         \int_step_inline:nn { #2 - 1 }
1239         {
1240             \tl_put_right:Nx \l__physicx_matrix_body_tl
1241             {
1242                 \exp_after:wN
1243                 \physicx_matrix_use_r_c:nn
1244                 #3 {{##1}} {{####1}} &
1245             }
1246         }
1247         \tl_put_right:Nx \l__physicx_matrix_body_tl
1248         {
1249             \exp_after:wN
1250             \physicx_matrix_use_r_c:nn
1251             #3 {{##1}} {{ \int_use:N #2 }} \[\dim_use:N \l__physicx_matrix_sep_dim]
1252         }
1253     }
1254     \int_step_inline:nn { #2 - 1 }
1255     {
1256         \tl_put_right:Nx \l__physicx_matrix_body_tl

```

```

1257     {
1258         \exp_after:wN
1259         \physicx_matrix_use_r_c:nn
1260         #3 {{ \int_use:N #1 }} {{##1}} &
1261     }
1262 }
1263 \tl_put_right:Nx \l__physicx_matrix_body_tl
1264 {
1265     \exp_after:wN
1266     \physicx_matrix_use_r_c:nn
1267     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1268 }
1269 }

```

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

\\_\_physicx\_declare\_init:

```

1270 \cs_new:Npn \__physicx_matrix_enhanced_init:
1271 {
1272     \seq_if_empty:NF \l__physicx_row_list_seq
1273     {
1274         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1275         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1276             { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1277     }
1278     \seq_if_empty:NF \l__physicx_col_list_seq
1279     {
1280         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1281         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1282             { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1283     }
1284 }

```

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

\commamatrix Define \commamatrix.

```

1285 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1286 {
1287     \group_begin:
1288     \keys_set:nn { physicx/matrix } {#3}
1289     \tl_if_empty:nF {#4}
1290         { \keys_set:nn { physicx/matrix } { array = {#4} } }
1291     \IfBooleanT {#1}
1292         { \keys_set:nn { physicx/matrix } { saveto = \physicxtmp } }
1293     \tl_set:Nx \l__physicx_matrix_array_tl
1294         { \__physicx_expand:w \l__physicx_matrix_array_tl }
1295     \bool_lazy_or:nnTF
1296         { \bool_if_p:n {#2} }
1297         { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1298         { \__physicx_commamatrix_enhanced: }
1299     {
1300         \tl_replace_all:Nox \l__physicx_matrix_array_tl
1301             { \physicx@cr } { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1302         \tl_replace_all:Non \l__physicx_matrix_array_tl
1303             { \physicx@align } { & }

```

```

1304         \tl_set_eq:NN \l__physicx_matrix_body_tl
1305         \l__physicx_matrix_array_tl
1306     }
1307     \__physicx_matrix_save_or_print:
1308     \group_end:
1309 }
1310 \cs_new_nopar:Npn \__physicx_matrix_save_or_print:
1311 {
1312     \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1313     {
1314         \exp_after:wN \tl_gset_eq:NN
1315         \l__physicx_matrix_save_tl
1316         \l__physicx_matrix_body_tl
1317     }
1318     {
1319         \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1320         \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1321         \fi:
1322         \exp_after:wN \__physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matrix_body_tl
1323         \l__physicx_matrix_body_tl
1324         \__physicx_matrix_end: \l__physicx_matrix_after_end_tl
1325     }
1326 }
1327 \cs_new:Npn \__physicx_commamatrix_enhanced:
1328 {
1329     \tl_clear:N \l__physicx_matrix_body_tl
1330     \int_zero:N \l__physicx_tmpa_int
1331     \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1332     \l__physicx_matrix_array_tl
1333     \int_set:Nn \l__physicx_matrix_rows_int
1334     { \seq_count:N \l__physicx_tmp_seq }
1335     \__physicx_matrix_enhanced_init:
1336     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1337     {
1338         \seq_map_tokens:Nn \l__physicx_tmp_seq
1339         {
1340             \int_incr:N \l__physicx_tmpa_int
1341             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1342             \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_e:nnn
1343         }
1344     }
1345     {
1346         \seq_map_tokens:Nn \l__physicx_tmp_seq
1347         {
1348             \int_incr:N \l__physicx_tmpa_int
1349             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1350             \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_ne:nnn
1351         }
1352     }
1353 }
1354 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1355 {
1356     \seq_set_split:Non \l__physicx_tmp_col_seq
1357     { \physicx@align } {#3}

```

```

1358 \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1359 \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1360 { #2 {##2} {#1} {##1} }
1361 \tl_put_right:Nx \l__physicx_matrix_body_tl
1362 {
1363   \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1364   \if_int_compare:w \l__physicx_matrix_rows_int = #1
1365     \scan_stop:
1366   \else:
1367     \[\dim_use:N \l__physicx_matrix_sep_dim]
1368   \fi:
1369 }
1370 }
1371 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1372 {
1373   \seq_put_right:Nx \l__physicx_tmp_coled_seq
1374   {
1375     \text_expand:n % \text_expand:n do the magic thing, but slower
1376     {
1377       \physicx@matricelement { #1 }
1378       { \__physicx_matrix_row_iterate:n {#2} }
1379       { \__physicx_matrix_col_iterate:n {#3} }
1380     }
1381   }
1382 }
1383 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1384 {
1385   \seq_put_right:No \l__physicx_tmp_coled_seq
1386   {
1387     \physicx@matricelement {#1}
1388     { \__physicx_matrix_row_iterate:n {#2} }
1389     { \__physicx_matrix_col_iterate:n {#3} }
1390   }
1391 }

```

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

\generalmatrix Define \generalmatrix.

```

1392 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1393 {
1394   \IfBooleanTF {#2}
1395   {
1396     \group_begin:
1397     \IfBooleanTF {#1}
1398     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicx_tmp } }
1399     { \keys_set:nn { physicx/matrix } {#4} }
1400     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1401     \physicx_construct:nnn
1402     {
1403       \tl_if_empty:NTF \l__physicx_matrix_main_tl
1404       {
1405         \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1406       }
1407       { \physicx_matrix_array_parse_main: }

```

```

1408     }
1409     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1410     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1411     \__physicx_generalmatrix:
1412     \__physicx_matrix_save_or_print:
1413     \group_end:
1414 }
1415 {
1416     \IfBooleanTF {#1}
1417     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1418     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1419     \qxmatrix = * [#4]
1420 }
1421 }
1422 \cs_new:Npn \__physicx_generalmatrix:
1423 {
1424     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1425     {
1426         \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1427         \__physicx_matrix_appto_body_e:off
1428     }
1429     {
1430         \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1431         \__physicx_matrix_appto_body_ne:off
1432     }
1433     \__physicx_matrix_transpose:N
1434     \__physicx_matrix_generate_body:NNNN
1435     \__physicx_generalmatrix_generate:nnn
1436 }

```

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

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

```

1437 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1438 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1439 {
1440     \__physicx_matrix_enhanced_init:
1441     \int_step_inline:nn { #1 - 1 }
1442     {
1443         \int_step_inline:nn { #2 - 1 }
1444         {
1445             \tl_set:Nx \l__physicx_tmp_tl
1446             {
1447                 \exp_after:wN
1448                 \physicx_matrix_use_r_c:nn
1449                 #3 {{##1}} {{####1}}
1450             }
1451             #4 \l__physicx_tmp_tl {{##1}} {{####1}}
1452             \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1453         }
1454     }
1455     \tl_set:Nx \l__physicx_tmp_tl
1456     {
1457         \exp_after:wN
1458         \physicx_matrix_use_r_c:nn

```

```

1458         #3 {{##1}} {{ \int_use:N #2 }}
1459     }
1460     #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1461     \tl_put_right:Nx \l__physicx_matrix_body_tl
1462     { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1463 }
1464 \int_step_inline:nn { #2 - 1 }
1465 {
1466     \tl_set:Nx \l__physicx_tmp_tl
1467     {
1468         \exp_after:wN
1469         \physicx_matrix_use_r_c:nn
1470         #3 {{ \int_use:N #1 }} {{##1}}
1471     }
1472     #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}
1473     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1474 }
1475 \tl_set:Nx \l__physicx_tmp_tl
1476 {
1477     \exp_after:wN
1478     \physicx_matrix_use_r_c:nn
1479     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1480 }
1481 #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1482 }

```

(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
1483 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1484 {
1485     \tl_put_right:Nx \l__physicx_matrix_body_tl
1486     {
1487         \text_expand:n
1488         {
1489             \physicx@matricelement {#1}
1490             { \__physicx_matrix_row_iterate:n {#2} }
1491             { \__physicx_matrix_col_iterate:n {#3} }
1492         }
1493     }
1494 }
1495 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1496 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1497 {
1498     \tl_put_right:No \l__physicx_matrix_body_tl
1499     {
1500         \physicx@matricelement {#1}
1501         { \__physicx_matrix_row_iterate:n {#2} }
1502         { \__physicx_matrix_col_iterate:n {#3} }
1503     }
1504 }
1505 \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

```

1506 \cs_new:Npn \_physicx_matrix_transpose:N #1 % generate body command
1507 {
1508   \bool_if:NTF \l__physicx_matrix_transpose_bool
1509   {
1510     #1
1511     \l__physicx_matrix_cols_int
1512     \l__physicx_matrix_rows_int
1513     \use_ii_i:nn
1514   }
1515   {
1516     #1
1517     \l__physicx_matrix_rows_int
1518     \l__physicx_matrix_cols_int
1519     \use:nn
1520   }
1521 }

```

(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.

```

1522 \cs_new:Npn \physicx_construct:nnn #1#2#3
1523 {
1524   \l__physicx_matrix_beginning_tl
1525   \_physicx_adi:nnn {#1} {#2} {#3}
1526   \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1527   {
1528     \int_incr:N \l__physicx_matrix_cols_int
1529     \_physicx_matrix_last_aux_c:
1530     \int_incr:N \l__physicx_matrix_cols_int
1531   }
1532   \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1533   {
1534     \int_incr:N \l__physicx_matrix_rows_int
1535     \_physicx_matrix_last_aux_r:
1536     \int_incr:N \l__physicx_matrix_rows_int
1537   }
1538   \bool_lazy_or:nnF
1539   { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1540   { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1541   {
1542     \physicx_matrix_set_r_c:nnn
1543     { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1544     { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1545     { \ddots }
1546   }
1547   \bool_if:NT \l__physicx_matrix_infinite_bool
1548   {
1549     \int_incr:N \l__physicx_matrix_rows_int
1550     \int_incr:N \l__physicx_matrix_cols_int
1551     \_physicx_matrix_last_aux_c:
1552     \_physicx_matrix_last_aux_r:
1553     \physicx_matrix_set_r_c:nnn

```

```

1554         { \int_use:N \l__physicx_matrix_rows_int }
1555         { \int_use:N \l__physicx_matrix_cols_int }
1556         { \ddots }
1557     }
1558     \l__physicx_matrix_ending_tl
1559 }
1560 \cs_new:Npn \__physicx_matrix_last_aux_c:
1561 {
1562     \int_step_inline:nn \l__physicx_matrix_rows_int
1563     {
1564         \physicx_matrix_set_r_c:nnn
1565         {##1} { \int_use:N \l__physicx_matrix_cols_int }
1566         { \cdots }
1567     }
1568 }
1569 \cs_new:Npn \__physicx_matrix_last_aux_r:
1570 {
1571     \int_step_inline:nn \l__physicx_matrix_cols_int
1572     {
1573         \physicx_matrix_set_r_c:nnn
1574         { \int_use:N \l__physicx_matrix_rows_int } {##1}
1575         { \vdots }
1576     }
1577 }

```

(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 1578 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
\NewGeneralMatrix 1579 {
\newdiagonalmatrix 1580 \NewDocumentCommand #2 { t+ m o o m m }
\NewDiagonalMatrix 1581 {
    \IfBooleanTF {##1}
    {
1582         \IfNoValueTF {##3}
1583         {
1584             \IfNoValueTF {##3}
1585             { \newcommand ##2 { #1 + [##5] {##6} } }
1586             {
1587                 \IfNoValueTF {##4}
1588                 { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1589                 { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1590             }
1591         }
1592     }
1593     \IfNoValueTF {##3}
1594     { \newcommand ##2 { #1 [##5] {##6} } }
1595     {
1596         \IfNoValueTF {##4}
1597         { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1598         { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1599     }
1600 }
1601 }

```



```

1602 \NewDocumentCommand #3 { t+ m m m m }
1603 {
1604   \IfBooleanTF {##1}
1605   { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1606   { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1607 }
1608 }
1609 \_physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1610 \_physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1611 \NewDocumentCommand \newgeneralmatrix { t+ m o o m }
1612 {
1613   \IfBooleanTF {#1}
1614   {
1615     \IfNoValueTF {#3}
1616     { \newcommand #2 { \generalmatrix + {#5} } }
1617     {
1618       \IfNoValueTF {#4}
1619       { \newcommand #2 [#3] { \generalmatrix + {#5} } }
1620       { \newcommand #2 [#3] [#4] { \generalmatrix + {#5} } }
1621     }
1622   }
1623   {
1624     \IfNoValueTF {#3}
1625     { \newcommand #2 { \generalmatrix {#5} } }
1626     {
1627       \IfNoValueTF {#4}
1628       { \newcommand #2 [#3] { \generalmatrix {#5} } }
1629       { \newcommand #2 [#3] [#4] { \generalmatrix {#5} } }
1630     }
1631   }
1632 }
1633 \NewDocumentCommand \NewGeneralMatrix { t+ m m m }
1634 {
1635   \IfBooleanTF {#1}
1636   { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1637   { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1638 }

```

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

```

1639 </package>

```

## Index

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

Symbols		
$\backslash +$	153	$\backslash \backslash$ ..... 43, 901, 908, 915, 939, 1251, 1301, 1367, 1462
$\backslash -$	153	$\backslash \{$ ..... 354, 373, 391, 430, 463
		$\backslash \}$ ..... 354, 375, 391, 430, 463

A		C	
<code>\A</code> .....	148, 153	<code>\cdots</code> .....	604, 887, 890, 891, 894, 896, 927, 928, 931, 933, 1566
<code>\abs</code> .....	424	clist commands:	
<code>\absolutevalue</code> .....	424	<code>\clist_clear:N</code> .....	789
<code>\acomm</code> .....	424	<code>\clist_concat:NNN</code> .....	796
<code>\aftergroup</code> .....	201	<code>\clist_if_in:NnTF</code> .....	1124
<code>\anticommutator</code> .....	424	<code>\clist_map_break:n</code> .....	162, 174
B		<code>\clist_map_inline:nn</code> .....	68, 158, 170, 794
<code>\begin</code> .....	374, 377, 381, 384, 385, 386, 391, 392, 396, 399, 400, 401, 758, 759, 760, 761, 762, 763, 764, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783	<code>\clist_new:N</code> .....	525, 526
<code>\bgroup</code> .....	200, 202	<code>\clist_put_right:Nn</code> ..	666, 672, 1115
<code>\Big</code> .....	406	<code>\clist_set:Nn</code> .....	786
<code>\big</code> .....	406	<code>\clist_set_eq:NN</code> .....	1091
<code>\Bigg</code> .....	406	<code>\c_empty_clist</code> .....	1091
<code>\bigg</code> .....	406	<code>\comm</code> .....	461
<code>\Biggl</code> .....	413	<code>\commamatrix</code> .....	26, 1285, 1610
<code>\biggl</code> .....	412	<code>\commutator</code> .....	424
<code>\Biggr</code> .....	413	cs commands:	
<code>\biggr</code> .....	412	<code>\cs:w</code> .....	31, 32, 695
<code>\Bigl</code> .....	411	<code>\cs_end:</code> .....	31, 32, 480, 695
<code>\bigl</code> .....	410	<code>\cs_generate_variant:Nn</code> .....	3, 4, 5, 6, 75, 78, 184, 336, 959, 1097, 1154, 1495, 1505
<code>\Bigr</code> .....	411	<code>\cs_if_exist_p:N</code> .....	1177, 1182
<code>\bigr</code> .....	410	<code>\cs_new:Npn</code> .....	7, 110, 118, 121, 133, 144, 156, 188, 192, 196, 200, 201, 202, 203, 246, 256, 268, 305, 312, 337, 404, 552, 742, 747, 952, 960, 988, 1001, 1012, 1022, 1048, 1087, 1089, 1098, 1099, 1133, 1155, 1229, 1234, 1270, 1327, 1354, 1371, 1383, 1422, 1438, 1483, 1496, 1506, 1522, 1560, 1569, 1578
<code>\boldsymbol</code> .....	190	<code>\cs_new_eq:NN</code> .....	120, 143, 511, 531, 540, 1020
bool commands:		<code>\cs_new_nopar:Npn</code> .....	470, 478, 487, 489, 494, 500, 533, 534, 537, 538, 539, 1310
<code>\bool_case_false:n</code> ....	325, 326, 333	<code>\cs_new_protected:Nn</code> .....	831
<code>\bool_case_true:nTF</code> .....	408, 1171	<code>\cs_new_protected:Npn</code> .....	53, 58, 63, 76, 79
<code>\bool_if:NnTF</code> .....	15, 20, 25, 71, 181, 339, 341, 345, 833, 888, 895, 905, 913, 925, 932, 936, 943, 947, 1211, 1336, 1424, 1508, 1547	<code>\cs_new_protected_nopar:Npn</code> .....	785, 787, 792
<code>\bool_if_p:N</code> .....	1209, 1297	<code>\cs_set:Nn</code> ..	709, 710, 711, 712, 713, 714
<code>\bool_if_p:n</code> .....	410, 411, 412, 413, 1208, 1296	<code>\cs_set:Npn</code> .....	426, 436, 437, 447, 448, 455, 456, 461, 466, 467, 468, 593, 602, 611, 620, 629
<code>\bool_lazy_and_p:nn</code> ....	1176, 1181	<code>\cs_set:Npx</code> .....	636
<code>\bool_lazy_or:nnTF</code> .....	30, 112, 123, 1207, 1295, 1538	<code>\cs_set_eq:NN</code> .....	55, 56, 60, 61, 190, 194, 198, 509, 562, 564, 566, 569, 575, 577, 591, 600, 609, 618, 627, 645, 648, 651, 654, 657, 835, 839, 966, 971, 976, 982, 1213, 1217, 1426, 1430
<code>\bool_new:N</code> ..	9, 10, 11, 12, 37, 52, 241, 243, 517, 518, 519, 527, 541, 549, 550	<code>\cs_set_nopar:Npn</code> .....	1275, 1281
<code>\bool_set:Nn</code> .....	252, 253, 1400	<code>\cs_set_protected:Npn</code> .....	317, 330, 349, 422
<code>\bool_set_false:N</code> .....	81, 169, 222, 821, 849, 864, 981		
<code>\bool_set_inverse:N</code> .....	587, 704		
<code>\bool_set_true:N</code> .....	115, 126, 136, 176, 218, 221, 820, 853, 857, 868, 872, 975, 1274, 1280		
<code>\c_false_bool</code> .....	369, 389		
<code>\c_true_bool</code> .....	352, 362, 369, 389		
<code>\Bqty</code> .....	424		
<code>\bqty</code> .....	424		

`\cs_to_str:N` ..... 262

## D

`\ddots` ..... 604, 914, 945, 1545, 1556  
`\DeclareDocumentCommand` .....  
    . 322, 332, 406, 816, 1190, 1285, 1392  
`\diagonalmatrix` ..... 24, 1190, 1609  
dim commands:  
    `\dim_new:N` ..... 551  
    `\dim_use:N` ..... 901,  
        908, 915, 939, 1251, 1301, 1367, 1462

## E

`\egroup` ..... 201, 203  
else commands:  
    `\else:` .....  
        482, 899, 1026, 1036, 1061, 1073, 1366  
`\end` .. 374, 377, 381, 384, 385, 386, 391,  
        392, 396, 399, 400, 401, 758, 759,  
        760, 761, 762, 763, 764, 767, 768,  
        769, 770, 771, 772, 773, 774, 775,  
        776, 777, 778, 779, 780, 781, 782, 783  
`\eval` ..... 424  
`\evaluated` ..... 351, 436  
exp commands:  
    `\exp_after:wN` .....  
        ... 1242, 1249, 1258, 1265, 1312,  
        1314, 1322, 1447, 1456, 1468, 1477  
    `\exp_args:Nf` ..... 996  
    `\exp_args:Nnno` ..... 1126, 1161  
    `\exp_args:Nno` ..... 160, 172  
    `\exp_args:No` ..... 622, 631, 731, 1108  
    `\exp_args:NV` ..... 1341, 1349  
    `\exp_not:N` ..... 8, 276, 636  
    `\exp_not:n` .....  
        ... 280, 281, 481, 483, 531, 562, 566  
    `\exp_stop_f:` .....  
        ... 8, 271, 278, 1024, 1027, 1050, 1062

## F

fi commands:  
    `\fi:` ..... 282, 297, 302, 303, 484,  
        902, 1045, 1046, 1084, 1085, 1321, 1368  
fp commands:  
    `\fp_eval:n` ..... 24, 1174  
`\fpeval` ..... 24, 1182, 1183

## G

`\generalmatrix` .. 28, 1392, 1616, 1619,  
    1620, 1625, 1628, 1629, 1636, 1637  
group commands:  
    `\group_begin:` .. 818, 1192, 1287, 1396  
    `\group_end:` .... 829, 1227, 1308, 1413

## I

if commands:  
    `\if_bool:N` ..... 286, 298  
    `\if_case:w` ..... 278  
    `\if_cs_exist:w` ..... 480  
    `\if_int_compare:w` ..... 271,  
        897, 1024, 1027, 1050, 1062, 1319, 1364  
    `\IfBooleanT` ..... 373, 375, 1291  
    `\IfBooleanTF` ..... 235,  
        324, 380, 382, 395, 397, 416, 754,  
        819, 824, 1193, 1394, 1397, 1416,  
        1417, 1418, 1582, 1604, 1613, 1635  
    `\IfNoValueTF` ..... 1584, 1587,  
        1593, 1596, 1615, 1618, 1624, 1627  
int commands:  
    `\int_compare:nNnTF` ..... 91, 101, 135  
    `\int_compare:nTF` ..... 847, 862  
    `\int_compare_p:nNn` . 113, 114, 124, 125  
    `\int_eval:n` .. 24, 1031, 1040, 1058,  
        1070, 1080, 1081, 1173, 1543, 1544  
    `\int_incr:N` . 270, 1006, 1340, 1348,  
        1528, 1530, 1534, 1536, 1549, 1550  
    `\int_max:nn` ..... 473, 475  
    `\int_new:N` .....  
        ... 38, 39, 48, 49, 50, 51, 245, 522, 523  
    `\int_set:Nn` ..... 66, 67, 90, 100,  
        251, 472, 474, 822, 823, 850, 865, 1333  
    `\int_set_eq:NN` .....  
        ... 88, 93, 98, 103, 1009, 1320  
    `\int_step_inline:nn` .....  
        ... 875, 1157, 1159, 1236, 1238,  
        1254, 1441, 1443, 1464, 1562, 1571  
    `\int_step_inline:nnn` 128, 138, 881, 919  
    `\int_use:N` ..... 1251,  
        1260, 1267, 1458, 1460, 1470, 1472,  
        1479, 1481, 1554, 1555, 1565, 1574  
    `\int_zero:N` ..... 1003, 1330  
`\interval` ..... 24, 1177, 1178

## K

keys commands:  
    `\keys_define:nn` 204, 555, 557, 744, 749  
    `\keys_if_exist:nnTF` ..... 160, 172  
    `\l_keys_key_str` .....  
        ... 160, 163, 172, 177, 731, 734  
    `\l_keys_path_str` ..... 738  
    `\keys_set:nn` .... 3, 163, 177, 236,  
        237, 556, 733, 802, 825, 826, 1194,  
        1195, 1288, 1290, 1292, 1398, 1399  
keyval commands:  
    `\keyval_parse:NNn` ..... 1092  
    `\keyval_parse:nnn` ..... 954

## L

`\langle` ..... 358, 385, 400

<code>\ldots</code>	604	<code>\physicsx_declare_legacy_paren:NnnnNNn</code>	
<code>\left</code>	200, 373, 380, 385, 391, 392, 395, 399, 400, 401		403, 427, 428, 429, 430, 432, 439, 443, 445, 451, 453, 458, 462
<code>\let</code>	319	<code>\physicsx_declare_legacy_quantity:nnNn</code>	
<code>\lggroup</code>	380, 395		239, 351, 361, 368, 388
<code>\lVert</code>	440	<code>\physicsx_if_num:n</code>	146
<b>M</b>			
<code>\mathcal</code>	186, 187	<code>\physicsx_if_num:nTF</code>	731, 845, 860, 1168
<code>\mathclose</code>	200, 202	<code>\physicsx_if_num_sign:n</code>	151
<code>\mathopen</code>	200, 202	<code>\physicsx_left:</code>	200, 320, 418
<code>\matrixquantity</code>	351	<code>\physicsx_left:N</code>	202, 410, 411, 412, 413
<code>\mqty</code>	424	<code>\physicsx_mathtools:</code>	23
msg commands:		<code>\physicsx_mathtools:TF</code>	765
<code>\msg_error:nnn</code>	986	<code>\physicsx_matrix_array_parse:n</code>	1133, 1405
<code>\msg_error:nnnn</code>	737	<code>\physicsx_matrix_array_parse_-</code>	
<code>\msg_new:nnn</code>	46	main:	1155, 1407
<code>\msg_new:nnnn</code>	40	<code>\physicsx_matrix_diag_parse:n</code>	667, 952, 1198, 1202, 1203, 1409
<b>N</b>			
<code>\NewCommaMatrix</code>	1578	<code>\physicsx_matrix_item_parse:n</code>	673, 1089, 1206, 1410
<code>\newcommamatrix</code>	1578	<code>\physicsx_matrix_new_type:nn</code>	742
<code>\newcommand</code>	1585, 1588, 1589, 1594, 1597, 1598, 1616, 1619, 1620, 1625, 1628, 1629	<code>\physicsx_matrix_new_type:nnn</code>	742
<code>\NewDiagonalMatrix</code>	1578	<code>\physicsx_matrix_set_r_c:nnn</code>	511, 645, 648, 651, 654, 657, 1007, 1015, 1030, 1039, 1056, 1068, 1079, 1126, 1150, 1161, 1542, 1553, 1564, 1573
<code>\newdiagonalmatrix</code>	1578	<code>\physicsx_matrix_use_r_c:nn</code>	478, 1243, 1250, 1259, 1266, 1448, 1457, 1469, 1478
<code>\NewDocumentCommand</code>	233, 752, 1580, 1602, 1605, 1606, 1611, 1633, 1636, 1637	<code>\physicsx_option_or:nn</code>	28
<code>\NewGeneralMatrix</code>	1578	<code>\physicsx_option_or:nnTF</code>	424, 434, 460, 464
<code>\newgeneralmatrix</code>	1578	<code>\c_physicsx_Order_tl</code>	187, 227, 444, 452
<code>\nobreak</code>	364, 365, 366	<code>\c_physicsx_order_tl</code>	186, 226, 446, 454
<code>\norm</code>	424	<code>\physicsx_parse_range:nnN</code>	76, 78, 1104, 1106
<b>O</b>			
<code>\OOrder</code>	448, 456	<code>\physicsx_parse_range:nnnN</code>	48
<code>\oorder</code>	424	<code>\physicsx_parse_range_check:</code>	48, 677
or commands:		<code>\physicsx_parse_range_nocheck:</code>	48, 678
<code>\or:</code>	279, 280, 281	<code>\physicsx_qxmatrix:nnn</code>	827, 831
<code>\Order</code>	424	<code>\physicsx_right:</code>	201, 320, 418
<code>\order</code>	424	<code>\physicsx_right:N</code>	203, 410, 411, 412, 413
<code>\Ordersymbol</code>	227	<code>\physicsx_search_also:nn</code>	156, 167, 184, 185
<code>\ordersymbol</code>	226	<code>\physicsx_search_also:nnTF</code>	723
<b>P</b>			
<code>\pb</code>	424	<code>\physicsx_short:</code>	18
peek commands:		<code>\physicsx_use_amssymb_type:</code>	188, 232
<code>\peek_charcode_ignore_spaces:N</code>	319	<code>\physicsx_use_uni_bf_type:</code>	196
physicsx commands:		<code>\physicsx_use_uni_bfit_type:</code>	192, 231
<code>\physicsx_bf:</code>	190, 194, 198	physicsx internal commands:	
<code>\physicsx_compat:</code>	13	<code>\__physicsx_adi:nnn</code>	552, 709, 710, 711, 712, 713, 714, 1525
<code>\physicsx_compat:TF</code>	224, 441	<code>\l_physicsx_begin_int</code>	48, 88, 90, 91, 93, 124, 125, 129, 135, 139
<code>\physicsx_construct:nnn</code>	1196, 1401, 1522		

\l__physicx_cmd_arg_int .....	\l__physicx_invalid_range_bool ..
..... 245, 251, 270, 271, 278	..... 52, 71, 81, 115, 126, 136
\l__physicx_cmd_arg_spec_tl ....	\l__physicx_item_ignore_clist ...
..... 244, 250, 263, 272	..... 1091, 1115, 1124
\l__physicx_cmd_auto_body_bool ..	\g__physicx_mathtools_bool .....
..... 243, 253, 298, 341, 345	..... 9, 25, 221, 222
\l__physicx_cmd_auto_body_tl ...	\__physicx_matrix_add_data:nn ...
..... 242, 249, 265, 299, 300	..... 669, 675, 790, 792
\l__physicx_cmd_noauto_body_bool	\l__physicx_matrix_after_begin_-
..... 241, 252, 286, 339	tl ..... 547, 685, 687, 1322
\l__physicx_cmd_noauto_body_tl ..	\l__physicx_matrix_after_end_tl .
..... 240, 248, 264, 287, 288	..... 548, 688, 690, 1324
\l__physicx_col_list_seq .....	\__physicx_matrix_appto_body_-
..... 529, 583, 1278, 1282	e:nnn ..... 836, 1214, 1427, 1483
\__physicx_commamatrix_enhanced:	\__physicx_matrix_appto_body_-
..... 1298, 1327	ne:nnn ..... 840, 1218, 1431, 1483
\__physicx_commamatrix_enhanced_-	\l__physicx_matrix_args_tl .....
aux:nNn ..... 1341, 1349, 1354	..... 546, 683, 684, 1322
\__physicx_commamatrix_enhanced_-	\l__physicx_matrix_array_tl ....
aux_e:nnn ..... 1342, 1371	..... 520, 559, 1293,
\__physicx_commamatrix_enhanced_-	1294, 1300, 1302, 1305, 1332, 1405
aux_ne:nnn ..... 1350, 1383	\__physicx_matrix_autocalc:nn ...
\g__physicx_compat_bool .....	..... 540, 575, 577, 1088, 1139, 1145
..... 11, 15, 206, 218	\__physicx_matrix_begin:w .....
\__physicx_declare_init: ..... 1270	..... 538, 680, 1322
\__physicx_declare_init:nnn 246, 258	\l__physicx_matrix_beginning_tl .
\__physicx_declare_legacy_-	..... 553, 715, 717, 1524
quantity_aux:nnnn ..... 268, 309	\l__physicx_matrix_body_tl .....
\__physicx_declare_legacy_-	..... 521, 843, 878, 883, 884, 890,
quantity_aux:NNnnn .. 261, 312, 336	891, 896, 901, 908, 910, 912, 914,
\__physicx_declare_legacy_-	915, 917, 921, 922, 927, 928, 933,
quantity_aux:nw .... 259, 305, 310	939, 940, 942, 944, 945, 1240, 1247,
\__physicx_diagonalmatrix_diag_-	1256, 1263, 1304, 1316, 1323, 1329,
main: ..... 976, 982, 1020, 1025	1361, 1452, 1461, 1473, 1485, 1498
\__physicx_diagonalmatrix_-	\__physicx_matrix_calc:nn .....
enhanced:nnn ..... 1213, 1217, 1232	..... 470, 575, 967
\__physicx_diagonalmatrix_-	\__physicx_matrix_col_iterate:n .
generate_body:NNN ..... 1225, 1234	537, 661, 1281, 1379, 1389, 1491, 1502
\__physicx_diagonalmatrix_-	\l__physicx_matrix_cols_int ....
generate_enhanced_body:NNN ..	. 474, 475, 523, 572, 823, 862, 865,
..... 1224, 1229	881, 911, 919, 941, 948, 949, 1003,
\__physicx_diagonalmatrix_no_-	1006, 1010, 1058, 1070, 1081, 1106,
diag: ..... 983, 1012, 1021	1159, 1319, 1320, 1511, 1518, 1528,
\__physicx_diagonalmatrix_set_-	1530, 1544, 1550, 1555, 1565, 1571
diag: ..... 977, 1001	\l__physicx_matrix_diag_bool ...
\l__physicx_end_int ..... 49,	..... 527, 975, 981
98, 100, 101, 103, 125, 129, 135, 139	\__physicx_matrix_diag_calc:nn ..
\__physicx_expand:w .... 531, 562,	..... 966, 971, 1016,
564, 566, 569, 992, 1103, 1137, 1294	1033, 1042, 1051, 1063, 1074, 1087
\__physicx_generalmatrix: 1411, 1422	\l__physicx_matrix_diag_clist ...
\__physicx_generalmatrix_-	..... 525, 664, 666, 1198, 1409
generate:nnn ..... 1426, 1430, 1435	\__physicx_matrix_diag_parse_-
\__physicx_if_can_num:n ..... 1166	aux:n ..... 955, 960
\__physicx_if_keyval:nTF .. 144, 1201	

```

\__physicx_matrix_diag_parse_-
  aux:nn ..... 956, 988
\__physicx_matrix_diag_parse_-
  aux_anti:n ..... 996, 1048
\__physicx_matrix_diag_parse_-
  aux_regu:n ..... 999, 1022
\l__physicx_matrix_dotcol_bool ..
  ..... 519,
  864, 868, 872, 888, 913, 925, 943, 947
\l__physicx_matrix_dotrow_bool ..
  ..... 518, 849, 853, 857, 905
\__physicx_matrix_element_-
  aux:nnn ..... 591, 597,
  600, 606, 609, 615, 618, 624, 627, 632
\__physicx_matrix_end: .... 681, 1324
\__physicx_matrix_end:w ..... 539
\l__physicx_matrix_ending_tl ...
  ..... 554, 718, 720, 1558
\l__physicx_matrix_enhanced_bool
  ..... 550, 701, 704, 1209, 1297
\__physicx_matrix_enhanced_init:
  ..... 1270, 1335, 1440
\l__physicx_matrix_expand_-
  element_bool ..... 541,
  640, 833, 1211, 1274, 1280, 1336, 1424
\__physicx_matrix_generate_-
  body:NNNN ..... 1231, 1434, 1437
\l__physicx_matrix_infinite_bool
  ..... 517, 584,
  587, 820, 821, 895, 932, 936, 1400, 1547
\l__physicx_matrix_item_clist ...
  ..... 526, 670, 672, 1206, 1410
\__physicx_matrix_item_parse_-
  aux:n ..... 1093, 1098
\__physicx_matrix_item_parse_-
  aux:nn ..... 1094, 1099
\__physicx_matrix_last_aux_c: ...
  ..... 1529, 1551, 1560
\__physicx_matrix_last_aux_r: ...
  ..... 1535, 1552, 1569
\l__physicx_matrix_last_col_tl ..
  ..... 536, 663, 1526, 1540
\l__physicx_matrix_last_row_tl ..
  ..... 535, 662, 1532, 1539
\l__physicx_matrix_main_tl .....
  ..... 524, 579, 1162, 1403
\__physicx_matrix_row_iterate:n .
  534, 660, 1275, 1378, 1388, 1490, 1501
\l__physicx_matrix_rows_int ....
  ..... 472, 473, 522, 571,
  822, 847, 850, 875, 897, 1009, 1104,
  1157, 1333, 1364, 1512, 1517, 1534,
  1536, 1543, 1549, 1554, 1562, 1574

\__physicx_matrix_save_or_print:
  ..... 828, 1226, 1307, 1310, 1412
\l__physicx_matrix_save_tl .....
  ..... 693, 695, 1312, 1315
\l__physicx_matrix_sep_dim .....
  ..... 551, 691, 901,
  908, 915, 939, 1251, 1301, 1367, 1462
\__physicx_matrix_set_data:nn ...
  ..... 668, 674, 787
\__physicx_matrix_set_r_c_-
  ckall:nnn ..... 509, 658
\__physicx_matrix_set_r_c_-
  ckep:nnn ..... 494, 649
\__physicx_matrix_set_r_c_-
  ckig:nnn ..... 489, 652
\__physicx_matrix_set_r_c_-
  ckigep:nnn ..... 500, 510, 655
\__physicx_matrix_set_r_c_-
  nock:nnn ..... 487, 512, 646
\l__physicx_matrix_tmparr_c_sep .
  ..... 1144, 1147, 1148
\l__physicx_matrix_tmparr_r_sep .
  ..... 1138, 1140, 1142
\__physicx_matrix_transpose:N ...
  ..... 1223, 1433, 1506
\l__physicx_matrix_transpose_-
  bool ..... 549, 696, 1508
\l__physicx_max_int .....
  ..... 50, 67, 98, 101, 103, 113, 124
\l__physicx_min_int .....
  ..... 51, 66, 88, 91, 93, 114
\__physicx_nauto_case:nnnn . 314, 337
\__physicx_new_matrix_cmd:NNN . 1578
\__physicx_parse_range_aux:n 70, 79
\__physicx_parse_range_range: ...
  ..... 56, 61, 106, 143
\__physicx_parse_range_range_-
  check: ..... 56, 121, 143
\__physicx_parse_range_range_-
  nocheck: ..... 61, 133
\__physicx_parse_range_single:n .
  ..... 55, 60, 108, 120
\__physicx_parse_range_single_-
  check:n ..... 55, 110, 120
\__physicx_parse_range_single_-
  nocheck:n ..... 60, 118
\g__physicx_physics_bool ..... 10
\__physicx_qxmatrix_appto_-
  body:nnn ..... 835,
  839, 879, 885, 892, 918, 923, 929
\l__physicx_row_list_seq .....
  ..... 528, 581, 1272, 1276
\g__physicx_short_bool ... 12, 20, 208
\l__physicx_tmp_col_seq .. 1356, 1359

```

<code>\l__physicx_tmp_coled_seq</code> .....	1358, 1363, 1373, 1385
<code>\l__physicx_tmp_colnum_seq</code> .....	1107, 1113, 1122
<code>\l__physicx_tmp_rownum_seq</code> .....	1105, 1111, 1120
<code>\l__physicx_tmp_seq</code> .....	1331, 1334, 1338, 1346
<code>\l__physicx_tmp_tl</code> .....	273, 287, 299, 1445, 1451, 1454, 1460, 1466, 1472, 1475, 1481
<code>\l__physicx_tmpa_bool</code>	37, 169, 176, 181
<code>\l__physicx_tmpa_int</code> .....	38, 1330, 1340, 1342, 1348, 1350
<code>\l__physicx_tmpa_seq</code> .....	72, 82, 116, 119, 130, 140
<code>\l__physicx_tmpa_tl</code> .....	86, 87, 90, 96, 97, 100
<code>\l__physicx_tmparr_tl</code> .....	1135, 1136, 1137, 1138
<code>\l__physicx_tmpb_int</code> .....	39
<code>\l__physicx_tmpb_seq</code> .....	85, 86, 96
<code>\l__physicx_tmpdiag_seq</code>	993, 1004, 1014, 1017, 1018, 1028, 1034, 1035, 1037, 1043, 1044, 1052, 1053, 1054, 1064, 1065, 1066, 1075, 1076, 1077
<code>\l__physicx_tmpdiag_tl</code> .....	990, 991, 992, 993
<code>\l__physicx_tmpitem_tl</code> .....	1101, 1102, 1103, 1109, 1127
<code>\physicxempty</code>	483, 497, 505, 542, 543, 642
<code>\physicxexcept</code>	545, 613, 622, 631, 637, 639
<code>\PHYSICXIGNORE</code> .....	7, 8, 491, 502, 1109
<code>\physicxset</code> .....	233
<code>\physicxtmp</code> ....	239, 825, 1194, 1292, 1398
<code>\poissonbracket</code> .....	424
<code>\pqty</code> .....	424
prg commands:	
<code>\prg_generate_conditional_</code> variant:Nnn .....	185
<code>\prg_new_conditional:Npnn</code> .....	13, 18, 23, 28, 146, 151, 167, 1166
<code>\prg_replicate:nn</code> .....	911, 941
<code>\prg_return_false:</code> .....	16, 21, 26, 34, 149, 154, 182, 1187
<code>\prg_return_true:</code> .....	16, 21, 26, 33, 149, 154, 182, 1169, 1186
<code>\ProcessKeysPackageOptions</code> .....	215
Q	
<code>\qty</code> .....	424
<code>\quantity</code> .....	351, 426
quark commands:	
<code>\quark_if_recursion_tail_stop:n</code> ..	307, 308
<code>\q_recursion_stop</code> .....	260
<code>\q_recursion_tail</code> .....	260
<code>\qxmatrix</code> .....	807, 1419
R	
<code>\rangle</code> .....	358, 385, 400
regex commands:	
<code>\regex_match:nnTF</code> ..	148, 153, 622, 631
<code>\RequirePackage</code> .....	210, 211, 212
<code>\rgroup</code> .....	382, 397
<code>\right</code> .....	201, 375, 382, 385, 391, 392, 397, 399, 400, 401
<code>\rVert</code> .....	440
S	
<code>\sb</code> .....	533, 878, 883, 890, 917, 921, 927
scan commands:	
<code>\scan_stop:</code> .....	898, 1365
seq commands:	
<code>\c_empty_seq</code> .....	65, 1358
<code>\seq_clear:N</code> .....	82
<code>\seq_concat:NNN</code> .....	72
<code>\seq_count:N</code> ....	1017, 1018, 1034, 1035, 1043, 1044, 1052, 1053, 1064, 1065, 1075, 1076, 1140, 1147, 1334
<code>\seq_if_empty:N</code> .....	1272, 1278
<code>\seq_item:Nn</code> .....	1276, 1282
<code>\seq_map_indexed_inline:Nn</code> .....	1004, 1014, 1028, 1037, 1054, 1066, 1077, 1142, 1148, 1359
<code>\seq_map_inline:Nn</code> .....	1111, 1113, 1120, 1122
<code>\seq_map_tokens:Nn</code> .....	1338, 1346
<code>\seq_new:N</code> .....	528, 529
<code>\seq_pop_left:NN</code> .....	86, 96
<code>\seq_put_right:Nn</code> .....	116, 119, 130, 140, 1373, 1385
<code>\seq_set_eq:NN</code> .....	65, 1358
<code>\seq_set_split:Nnn</code> .....	5, 85, 581, 583, 993, 1138, 1144, 1331, 1356
<code>\seq_use:Nn</code> .....	1363
<code>\setmatrixdata</code> .....	785
<code>\setmatrixtype</code> .....	742, 758, 759, 760, 761, 762, 763, 764, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783
<code>\smallmatrixquantity</code> .....	351
<code>\smqty</code> .....	424
str commands:	
<code>\str_case_e:nnTF</code> .....	962
<code>\str_const:Nn</code> .....	514, 515, 516

<code>\str_if_eq:nnTF</code> . . . . .	292, 294	<code>\tl_put_right:Nn</code> . . .	272, 287, 288, 299, 300, 639, 687, 690, 717, 720, 878, 883, 884, 890, 891, 896, 901, 908, 910, 912, 914, 915, 917, 921, 922, 927, 928, 933, 939, 940, 942, 944, 945, 1240, 1247, 1256, 1263, 1361, 1452, 1461, 1473, 1485, 1498
<code>\sympf</code> . . . . .	198	<code>\tl_replace_all:Nnn</code> . . .	6, 1300, 1302
<code>\sympfit</code> . . . . .	194	<code>\tl_set:Nn</code> . . . . .	273, 488, 492, 497, 498, 505, 506, 683, 695, 990, 991, 1101, 1102, 1135, 1136, 1293, 1445, 1454, 1466, 1475
<b>T</b>			
T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:			
<code>\@declareparencommand</code> . . . . .	403	<code>\tl_set_eq:NN</code> . . . . .	226, 227, 1304
<code>\@declarequantitycommand</code> . . . . .	239	<code>\tl_tail:n</code> . . . . .	997
<code>\@ifpackageloaded</code> . . . . .	217, 220, 230	token commands:	
<code>\c@MaxMatrixCols</code> . . . . .	700, 1319, 1320	<code>\token_if_cs:NTF</code> . . . . .	1312
<code>\physicsx@align</code> . . . . .		<code>\TrimSpaces</code> . . . . .	752
. . . . .	514, 706, 1144, 1303, 1357	<b>U</b>	
<code>\physicsx@cr</code> . . . . .	515, 705, 1138, 1301, 1331	use commands:	
<code>\physicsx@matricelement</code> . . . . .		<code>\use:n</code> . . . . .	569
. . . . .	533, 588, 592,	<code>\use:nn</code> . . . . .	1437, 1519
. . . . .	593, 601, 602, 610, 611, 619, 620,	<code>\use:nnnn</code> . . . . .	4
. . . . .	628, 629, 636, 1377, 1387, 1489, 1500	<code>\use_i:nn</code> . . . . .	315, 1105, 1222, 1418
<code>\physicsx@sep</code> . . . . .	516, 581, 583, 707, 993	<code>\use_i:nnn</code> . . . . .	1418
tex commands:		<code>\use_i_ii:nnn</code> . . . . .	1220, 1417
<code>\tex_advance:D</code> . . . . .	948, 949	<code>\use_ii:nn</code> . . . . .	315, 1107
text commands:		<code>\use_ii_i:nn</code> . . . . .	1437, 1513
<code>\text_expand:n</code> . . . . .	564, 1375, 1487	<code>\use_none:nn</code> . . . . .	540, 577, 971
tl commands:		<b>V</b>	
<code>\c_empty_tl</code> . . . . .	331	<code>\vdots</code> 604, 910, 912, 914, 940, 942, 944, 1575	
<code>\tl_clear:N</code> . . . . .	248, 249, 250, 843, 1329	<code>\Vert</code> . . . . .	359, 401
<code>\tl_const:Nn</code> . . . . .	186, 187	<code>\vert</code> . . . . .	357, 364, 365, 366, 399, 429, 433
<code>\tl_gset_eq:NN</code> . . . . .	1314	<code>\vqty</code> . . . . .	424
<code>\tl_if_empty:NTF</code> . . . . .		<b>Z</b>	
. . . . .	87, 97, 1403, 1526, 1532	<code>\Z</code> . . . . .	148, 153
<code>\tl_if_empty:nTF</code> . . . . .			
. . . . .	496, 504, 595, 1199, 1289		
<code>\tl_if_empty_p:N</code> . . . . .	1539, 1540		
<code>\tl_if_eq:nnTF</code> . . . . .	491, 502, 1108		
<code>\tl_if_head_eq_charcode:nNTF</code> . . . . .	994		
<code>\tl_if_head_eq_meaning_p:nN</code> . . . . .			
. . . . .	1173, 1174, 1178, 1183		
<code>\tl_if_in:nnTF</code> . . . . .	83, 145, 604, 613		
<code>\tl_if_novalue_p:n</code> . . . . .	276		
<code>\tl_new:N</code> . . . . .	239,		
. . . . .	240, 242, 244, 520, 521, 524, 535,		
. . . . .	536, 543, 545, 546, 547, 548, 553, 554		