

The `physicx` package

Wenjian Chern (Longaster*)

November 6, 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

```

34         { \prg_return_false: }
35     }
36
37     \bool_new:N \l__cwamcro_physics_tmpa_bool
38     \int_new:N \l__cwamcro_physics_tmpa_int
39     \int_new:N \l__cwamcro_physics_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_end_int > \l__physicx_max_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__cwamcro_physics_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__cwamcro_physics_tmpa_bool
177         \keys_set:no {##1} { \l_keys_key_str = #2 }
178       }
179     }
180   }
181   \bool_if:NTF \l__cwamcro_physics_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 \physicx_bf: \boldsymbol
191 }
192 \cs_new:Npn \physicx_use_uni_bfit_type:
193 {
194   \cs_set_eq:NN \physicx_bf: \symbfit
195 }
196 \cs_new:Npn \physicx_use_uni_bf_type:
197 {
198   \cs_set_eq:NN \physicx_bf: \symbf
199 }
200 \keys_define:nn { physicx }
201 {
202   compat .bool_set:N = \g__physicx_compat_bool ,
203   compat .default:n = true ,
204   short .bool_set:N = \g__physicx_short_bool ,
205   short .default:n = true ,
206   physics .code:n = \RequirePackage{physics} ,
207   mathtools .code:n = \RequirePackage{mathtools} ,
208   unimath .code:n = \RequirePackage{unicode-math} ,
209 }
210 %
211 \ProcessKeysPackageOptions { physicx }
212 %
213 \@ifpackageloaded{physics}
214 { \bool_set_true:N \g__physicx_compat_bool }
215 { }
216 \@ifpackageloaded{mathtools}
217 { \bool_set_true:N \g__physicx_mathtools_bool }
218 { \bool_set_false:N \g__physicx_mathtools_bool }
219 %
220 \physicx_compat:T
221 {
222   \tl_set_eq:NN \ordersymbol \c_physicx_order_tl
223   \tl_set_eq:NN \Ordersymbol \c_physicx_Order_tl
224 }
225 %
226 \@ifpackageloaded {unicode-math}
227 { \physicx_use_uni_bfit_type: }
228 { \physicx_use_amssymb_type: }

```

`\physicxset` `physicx` setup command.

```

229 \NewDocumentCommand \physicxset { s m }
230 {
231   \IfBooleanTF {#1}
232   { \keys_set:nn { physicx/#2 } }
233   { \keys_set:nn { physicx } {#2} }
234 }

```

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

1.2 Quantity things

```

\physicx_declare_legacy_quantity:nn\N
\@declarequantitycmd

```

```

235 \tl_new:N \physicx_tmp
236 \tl_new:N \l__physicx_cmd_noauto_body_tl
237 \bool_new:N \l__physicx_cmd_noauto_body_bool
238 \tl_new:N \l__physicx_cmd_auto_body_tl
239 \bool_new:N \l__physicx_cmd_auto_body_bool
240 \tl_new:N \l__physicx_cmd_arg_spec_tl
241 \int_new:N \l__physicx_cmd_arg_int
242 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
243 {
244   \tl_clear:N \l__physicx_cmd_noauto_body_tl
245   \tl_clear:N \l__physicx_cmd_auto_body_tl
246   \tl_clear:N \l__physicx_cmd_arg_spec_tl
247   \int_set:Nn \l__physicx_cmd_arg_int {#1}
248   \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
249   \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
250 }
251 % noauto, auto, cmd, body
252 \cs_new:Npn \physicx_declare_legacy_quantity:nnNn #1#2#3#4
253 {
254   \__physicx_declare_init:nnn { 3 } {#1} {#2}
255   \__physicx_declare_legacy_quantity_aux:nw #4
256   \q_recursion_tail \q_recursion_tail \q_recursion_stop
257   \__physicx_declare_legacy_quantity_aux:NcVVV
258   #3 { \cs_to_str:N #3 ~ body }
259   \l__physicx_cmd_arg_spec_tl
260   \l__physicx_cmd_noauto_body_tl
261   \l__physicx_cmd_auto_body_tl
262 }
263 % arg spec, pre, body to replace(start from #4), post
264 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
265 {
266   \int_incr:N \l__physicx_cmd_arg_int
267   \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
268     \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
269     \tl_set:Nx \l__physicx_tmp_tl
270     {
271       {
272         \exp_not:N \tl_if_novalue_p:n
273         {
274           \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:
275           \or: \or: \or:
276           \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
277           \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
278           \fi:
279         }
280       }
281     }
282   \if_bool:N \l__physicx_cmd_noauto_body_bool
283     \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
284     \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
285     {
286       {
287         % if is '.', use none
288         \str_if_eq:nnTF {#2} {.} {} {#2}

```

```

289         #3
290         \str_if_eq:nnTF {#4} {.} {} {#4}
291     }
292 }
293 \fi:
294 \if_bool:N \l__physicx_cmd_auto_body_bool
295     \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
296     \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
297     { { ##1 #2 #3 ##2 #4 } }
298 \fi:
299 \fi:
300 }
301 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
302 {
303     \quark_if_recursion_tail_stop:n {#1}
304     \quark_if_recursion_tail_stop:n {#2}
305     \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
306     \__physicx_declare_legacy_quantity_aux:nw
307 }
308 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
309 {
310     \__physicx_nauto_case:nnnn
311     { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
312     {
313         \cs_set_protected:Npn #1
314         {
315             \peek_charcode_ignore_spaces:NTF \let
316             { #2 } { #2 [ \left ] \right }
317         }
318         \DeclareDocumentCommand #2 { 0{##2} m s #3 }
319         {
320             \IfBooleanTF { ##3 }
321             { \bool_case_false:n {#4} }
322             { \bool_case_false:n {#5} }
323         }
324     }
325     {
326         \cs_set_protected:Npn #1
327         { #2 \c_empty_tl \c_empty_tl }
328         \DeclareDocumentCommand #2 { m m s #3 }
329         { \bool_case_false:n {#4} }
330     }
331 }
332 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
333 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
334 {
335     \bool_if:NTF \l__physicx_cmd_noauto_body_bool
336     {
337         \bool_if:NTF \l__physicx_cmd_auto_body_bool
338         {#1} {#2}
339     }
340     {
341         \bool_if:NTF \l__physicx_cmd_auto_body_bool
342         {#3} {#4}

```

```

343     }
344 }
345 \cs_set_protected:Npn \@declarequantitycmd
346 { \physicsx_declare_legacy_quantity:nnNn }

```

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

```

\quantity      Redefine some macros in physics package.
\evaluated
\matrixquantity
\smallmatrixquantity
347 \physicsx_declare_legacy_quantity:nnNn
348 \c_true_bool \c_true_bool \quantity
349 {
350   { !g } { { \{ } { #4 } { \} } }
351   { !o } { { [ } { #5 } { ] } }
352   { !d() } { { ( } { #6 } { ) } }
353   { !d|| } { { \vert } { #7 } { \vert } }
354   { !d<> } { { \langle } { #8 } { \rangle } }
355   { !d== } { { \Vert } { #9 } { \Vert } }
356 }
357 \physicsx_declare_legacy_quantity:nnNn
358 \c_true_bool \c_true_bool \evaluated
359 {
360   { !g } { { . } { #4 \nobreak } { \vert } }
361   { !d[ ] } { { [ ] } { #5 \nobreak } { \vert } }
362   { !d( ) } { { ( ) } { #6 \nobreak } { \vert } }
363 }
364 \physicsx_declare_legacy_quantity:nnNn
365 \c_true_bool \c_false_bool \matrixquantity
366 {
367   { !g }
368   {
369     { \IfBooleanT{#3}{\left\{ } }
370     { \begin{matrix} #4 \end{matrix} }
371     { \IfBooleanT{#3}{\right\} }
372   }
373   { !o } { { \begin{bmatrix} } { #5 } { \end{bmatrix} } }
374   { !d() }
375   {
376     { \IfBooleanTF{#3}{\left\lgroup}{\left( } }
377     { \begin{matrix} #6 \end{matrix} }
378     { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
379   }
380   { !d|| } { { \begin{vmatrix} } { #7 } { \end{vmatrix} } }
381   { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } { \right\rangle } }
382   { !d== } { { \begin{Vmatrix} } { #9 } { \end{Vmatrix} } }
383 }
384 \physicsx_declare_legacy_quantity:nnNn
385 \c_true_bool \c_false_bool \smallmatrixquantity
386 {
387   { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } { \right\} } }
388   { !o } { { \left[ ] } { \begin{smallmatrix} #5 \end{smallmatrix} } { \right[ ] } }
389   { !d() }
390   {
391     { \IfBooleanTF{#3}{\left\lgroup}{\left( } }

```



```

392     { \begin{smallmatrix} #6 \end{smallmatrix} }
393     { \IfBooleanTF{#3}{\right\rgroup}{\right}} }
394   }
395   { !d|| } { {\left\vert} { \begin{smallmatrix} #7 \end{smallmatrix} } {\right\vert} }
396   { !d<> } { {\left\langle} { \begin{smallmatrix} #8 \end{smallmatrix} } {\right\rangle} }
397   { !d== } { {\left\Vert} { \begin{smallmatrix} #9 \end{smallmatrix} } {\right\Vert} }
398 }

```

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

\physicsx_declare_legacy_paren:NnnnNn

\@declareparencmd

```

399 %% cmd, arg spec, replace(start from #6), pre, left, right, post
400 \cs_new:Npn \physicsx_declare_legacy_paren:NnnnNn #1#2#3#4#5#6#7
401 {
402   \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
403   {
404     \bool_case_true:nF
405     {
406       { \bool_if_p:n {##2} } { #4 \bigl #5 #3 \bigr #6 #7 }
407       { \bool_if_p:n {##3} } { #4 \Bigl #5 #3 \Bigr #6 #7 }
408       { \bool_if_p:n {##4} } { #4 \biggl #5 #3 \biggr #6 #7 }
409       { \bool_if_p:n {##5} } { #4 \Biggl #5 #3 \Biggr #6 #7 }
410     }
411     {
412       \IfBooleanTF {##1}
413       { #4 #5 #3 #6 #7 }
414       { #4 \!\left #5 #3 \right #6 #7 }
415     }
416   }
417 }
418 \cs_set_protected:Npn \@declareparencmd
419 { \physicsx_declare_legacy_paren:NnnnNn }

```

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

```

\qty      Redefine some macros in physics package.
\mqty     420 \physicsx_option_or:nnT { compat } { short }
\smqty    421 {
\pqty     422   \cs_set:Npn \qty { \quantity }
\bqty     423   \physicsx_declare_legacy_paren:NnnnNn \pqty { m } {#6} { } { } { }
\vqty     424   \physicsx_declare_legacy_paren:NnnnNn \bqty { m } {#6} { } { } { }
\Bqty     425   \physicsx_declare_legacy_paren:NnnnNn \vqty { m } {#6} { } {\vert \vert { }
\absolutevalue 426   \physicsx_declare_legacy_paren:NnnnNn \Bqty { m } {#6} { } {\{ \} { }
427 }
\eval     427 }
\abs      428 \physicsx_declare_legacy_paren:NnnnNn \absolutevalue
429 { m } {#6} { } {\vert \vert { }
\norm     429 { m } {#6} { } {\vert \vert { }
\order    430 \physicsx_option_or:nnT { compat } { short }
431 {
\Order    431 {
432   \cs_set:Npn \eval { \evaluated }
\oorder   432   \cs_set:Npn \abs { \absolutevalue }
433 }
\commutator 433 }
434 }
\poissonbracket 434 }
435 \physicsx_declare_legacy_paren:NnnnNn \norm
\pb       435 { m } {#6} { } {\lVert \rVert { }
436 }
\anticommutator 436 \physicsx_compat:TF
437 }
\acomm    437 }

```

```

438 {
439   \physicx_declare_legacy_paren:NnnnNNn \order
440   { m } {#6} { \c_physicx_Order_tl } ( ) { }
441   \physicx_declare_legacy_paren:NnnnNNn \oorder
442   { m } {#6} { \c_physicx_order_tl } ( ) { }
443   \cs_set:Npn \Order { \order }
444   \cs_set:Npn \OOrder { \order }
445 }
446 {
447   \physicx_declare_legacy_paren:NnnnNNn \Order
448   { m } {#6} { \c_physicx_Order_tl } ( ) { }
449   \physicx_declare_legacy_paren:NnnnNNn \order
450   { m } {#6} { \c_physicx_order_tl } ( ) { }
451   \cs_set:Npn \oorder { \order }
452   \cs_set:Npn \OOrder { \Order }
453 }
454 \physicx_declare_legacy_paren:NnnnNNn \commutator
455 { m m } { #6 , #7 } { } [ ] { }
456 \physicx_option_or:nnT { compat } { short }
457 { \cs_set:Npn \comm { \commutator } }
458 \physicx_declare_legacy_paren:NnnnNNn \poissonbracket
459 { m m } { #6 , #7 } { } \{ \} { }
460 \physicx_option_or:nnT { compat } { short }
461 {
462   \cs_set:Npn \pb { \poissonbracket }
463   \cs_set:Npn \anticommutator { \poissonbracket }
464   \cs_set:Npn \acomm { \poissonbracket }
465 }

```

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

1.3 Matrix things

1.3.1 Matrix auxillary functions

```

466 \cs_new_nopar:Npn \__physicx_matrix_calc:nn #1#2
467 {
468   \int_set:Nn \l__physicx_matrix_rows_int
469   { \int_max:nn {#1} \l__physicx_matrix_rows_int }
470   \int_set:Nn \l__physicx_matrix_cols_int
471   { \int_max:nn {#2} \l__physicx_matrix_cols_int }
472 }
473 % use matrix element
474 \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
475 {
476   \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
477   \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
478   \else:
479   \exp_not:o { \physicxempty }
480   \fi:
481 }
482 % set matrix element, check or not
483 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
484 { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
485 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3

```

```

486 {
487   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
488   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
489 }
490 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckepp:nnn #1#2#3
491 {
492   \tl_if_empty:nTF {#3}
493   { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
494   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
495 }
496 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
497 {
498   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
499   {
500     \tl_if_empty:nTF {#3}
501     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
502     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
503   }
504 }
505 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
506 \__physicx_matrix_set_r_c_ckigep:nnn
507 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
508 \__physicx_matrix_set_r_c_nock:nnn
509 % align, cr, sep symbol
510 \str_const:Nn \physicx@align { , }
511 \str_const:Nn \physicx@cr { ; }
512 \str_const:Nn \physicx@sep { , }
513 \bool_new:N \l__physicx_matrix_infinite_bool
514 \bool_new:N \l__physicx_matrix_dotrow_bool
515 \bool_new:N \l__physicx_matrix_dotcol_bool
516 \tl_new:N \l__physicx_matrix_array_tl
517 \tl_new:N \l__physicx_matrix_body_tl
518 \int_new:N \l__physicx_matrix_rows_int
519 \int_new:N \l__physicx_matrix_cols_int
520 \tl_new:N \l__physicx_matrix_main_tl
521 \clist_new:N \l__physicx_matrix_diag_clist
522 \clist_new:N \l__physicx_matrix_item_clist
523 \bool_new:N \l__physicx_matrix_diag_bool
524 \seq_new:N \l__physicx_row_list_seq
525 \seq_new:N \l__physicx_col_list_seq
526 % expand input
527 \cs_new_eq:NN \__physicx_expand:w \exp_not:o
528 %% main, row iterate, col iterate
529 \cs_new_nopar:Npn \physicx@matrilement #1#2#3 { #1 \sb { #2 #3 } }
530 \cs_new_nopar:Npn \__physicx_matrix_row_iterate:n #1 { #1 }
531 \tl_new:N \l__physicx_matrix_last_row_tl
532 \tl_new:N \l__physicx_matrix_last_col_tl
533 \cs_new_nopar:Npn \__physicx_matrix_col_iterate:n #1 { #1 }
534 \cs_new_nopar:Npn \__physicx_matrix_begin:w { }
535 \cs_new_nopar:Npn \__physicx_matrix_end:w { }
536 \cs_new_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn
537 \bool_new:N \l__physicx_matrix_expand_element_bool
538 % when element is empty use \physicxempty
539 \tl_new:N \physicxempty

```

```

540 % save 'element-except' key's value
541 \tl_new:N \physicsexcept
542 \tl_new:N \l__physicx_matrix_args_tl
543 \tl_new:N \l__physicx_matrix_after_begin_tl
544 \tl_new:N \l__physicx_matrix_after_end_tl
545 \bool_new:N \l__physicx_matrix_transpose_bool
546 \bool_new:N \l__physicx_matrix_enhanced_bool
547 \dim_new:N \l__physicx_matrix_sep_dim
548 \cs_new:Npn \__physicx_adi:nnn #1#2#3 { #1#2#3 }
549 \tl_new:N \l__physicx_matrix_beginning_tl
550 \tl_new:N \l__physicx_matrix_ending_tl

```

1.3.2 Matrix keys

```

551 \keys_define:nn { physicx }
552 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
553 \keys_define:nn { physicx/matrix }
554 {
555   array .tl_set:N = \l__physicx_matrix_array_tl ,
556   expand .choice: ,
557   expand / none .code:n =
558     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
559   expand / text-expand .code:n =
560     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
561   expand / f .code:n =
562     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
563   expand / romanual .meta:n = { expand = f } ,
564   expand / x .code:n =
565     \cs_set_eq:NN \__physicx_expand:w \use:n ,
566   expand / edef .meta:n = { expand = x } ,
567   rows .int_set:N = \l__physicx_matrix_rows_int ,
568   cols .int_set:N = \l__physicx_matrix_cols_int ,
569   auto-update .choice: ,
570   auto-update / true .code:n =
571     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,
572   auto-update / false .code:n =
573     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
574   auto-update .default:n = true ,
575   main .tl_set:N = \l__physicx_matrix_main_tl ,
576   row-list .code:n =
577     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,
578   col-list .code:n =
579     \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
580   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
581   infinite .default:n = true ,
582   !infinite .code:n =
583     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
584   element-code .cs_set:Np = \physicx@matricelement #1#2#3 ,
585   element-code* .choice: ,
586   element-code* / except-empty .code:n =
587     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
588       \physicx@matricelement
589     \cs_set:Npn \physicx@matricelement ##1##2##3
590       {
591         \tl_if_empty:nTF {##1}

```

```

592         {##1}
593         { \_physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
594     } ,
595 element-code* / except-dots .code:n =
596     \cs_set_eq:NN \_physicx_matrix_element_aux:nnn
597     \physicx@matrxiement
598     \cs_set:Npn \physicx@matrxiement ##1##2##3
599     {
600         \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
601         {##1}
602         { \_physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
603     } ,
604 element-code* / except-tl .code:n =
605     \cs_set_eq:NN \_physicx_matrix_element_aux:nnn
606     \physicx@matrxiement
607     \cs_set:Npn \physicx@matrxiement ##1##2##3
608     {
609         \tl_if_in:onTF { \physicxexcept } {##1}
610         {##1}
611         { \_physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
612     } ,
613 element-code* / except-regex .code:n =
614     \cs_set_eq:NN \_physicx_matrix_element_aux:nnn
615     \physicx@matrxiement
616     \cs_set:Npn \physicx@matrxiement ##1##2##3
617     {
618         \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
619         {##1}
620         { \_physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
621     } ,
622 element-code* / only-regex .code:n =
623     \cs_set_eq:NN \_physicx_matrix_element_aux:nnn
624     \physicx@matrxiement
625     \cs_set:Npn \physicx@matrxiement ##1##2##3
626     {
627         \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
628         { \_physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
629         {##1}
630     } ,
631 element-code* / unknown .code:n =
632     \cs_set:Npx \physicx@matrxiement { \exp_not:c {#1} },
633 element-except .tl_set:N = \physicxexcept ,
634 element-except+ .code:n =
635     \tl_put_right:Nn \physicxexcept {#1} ,
636 expand-element .bool_set:N = \l_physicx_matrix_expand_element_bool ,
637 expand-element .default:n = true ,
638 empty .tl_set:N = \physicxempty ,
639 check .choice: ,
640 check / none .code:n =
641     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
642     \_physicx_matrix_set_r_c_nock:nnn ,
643 check / empty .code:n =
644     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
645     \_physicx_matrix_set_r_c_ckepp:nnn ,

```

```

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

```

```

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

```

```

\physicx_matrix_new_type:nnn
\physicx_matrix_new_type:nn
  \setmatritxtpe

```

```

738 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
739 {
740   \keys_define:nn { physicx/matrix }
741     { type / #1 .meta:n = { begin={#2} , end={#3} } }
742 }
743 \cs_new:Npn \physicx_matrix_new_type:nn #1#2
744 {
745   \keys_define:nn { physicx/matrix }
746     { type / #1 .meta:n = {#2} }
747 }
748 \NewDocumentCommand \setmatritxtpe { s >{ \TrimSpaces } m }
749 {
750   \IfBooleanTF {#1}
751     { \physicx_matrix_new_type:nn {#2} }

```

```

752     { \physicx_matrix_new_type:nnn {#2} }
753 }

```

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

A few types.

```

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

```

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

```

781 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
782 { \clist_set:cn { physicx@ #1 data@ #2 } }
783 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
784 {
785   \clist_clear:c { l__physicx_matrix_ #1 _clist }
786   \__physicx_matrix_add_data:nn {#1} {#2}
787 }
788 \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
789 {
790   \clist_map_inline:nn {#2}
791   {
792     \clist_concat:ccc
793     { l__physicx_matrix_ #1 _clist }
794     { l__physicx_matrix_ #1 _clist }
795     { physicx@ #1 data@ #2 }
796   }
797 }

```

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

Initial settings.

```

798 \keys_set:nn { physicx/matrix }
799 {
800     type = m ,
801     saveto = ? ,
802 }

```

\qxmatrix

```

803 %% basically, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
      xmatmnm-in-the-physics-package, but changed some
804 % #1 = boolean, saveto matrix
805 % #2 = star, infinite
806 % #3 = options
807 % #4 = letter for the entries
808 % #5 = number of rows
809 % #6 = number of explicit rows, default = 3
810 % #7 = number of columns
811 % #8 = number of explicit columns, default = 3
812 \DeclareDocumentCommand \qxmatrix { t= s 0{type=p} m m 0{3} m 0{3} }
813 {
814     \group_begin:
815     \IfBooleanTF { #2 }
816     { \bool_set_true:N \l__physicx_matrix_infinite_bool }
817     { \bool_set_false:N \l__physicx_matrix_infinite_bool }
818     \int_set:Nn \l__physicx_matrix_rows_int {#6}
819     \int_set:Nn \l__physicx_matrix_cols_int {#8}
820     \IfBooleanTF {#1}
821     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
822     { \keys_set:nn { physicx/matrix } {#3} }
823     \physicx_qxmatrix:nnn {#4} {#5} {#7}
824     \__physicx_matrix_save_or_print:
825     \group_end:
826 }
827 \cs_new_protected:Nn \physicx_qxmatrix:nnn
828 {
829     \bool_if:NTF \l__physicx_matrix_expand_element_bool
830     {
831         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
832         \__physicx_matrix_appto_body_e:nnn
833     }
834     {
835         \cs_set_eq:NN \__physicx_qxmatrix_appto_body:nnn
836         \__physicx_matrix_appto_body_ne:nnn
837     }
838     % clear the variable containing the body of the matrix
839     \tl_clear:N \l__physicx_matrix_body_tl
840     % set the tentative number of explicit rows
841     \physicx_if_num:nTF { #2 }
842     {% number of rows is an integer
843         \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
844         {% if #2 <= rows, we don't want a row of dots
845             \bool_set_false:N \l__physicx_matrix_dotrow_bool
846             \int_set:Nn \l__physicx_matrix_rows_int { #2 }
847         }

```

```

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

```

```

902 {
903   % finish up the row
904   \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
905   % if we have a row of dots, fill it in
906   \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
907   \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
908     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
909   \bool_if:NT \l__physicx_matrix_dotcol_bool
910     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
911   \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
912   % fill the last row
913   %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
914   \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
915   \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
916     {
917       %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
918       \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
919       \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
920     }
921   \bool_if:NT \l__physicx_matrix_dotcol_bool
922     {
923       %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
924       \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
925       \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
926     }
927   % if the matrix is infinite, add a further column with \cdots
928   \bool_if:NT \l__physicx_matrix_infinite_bool
929     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
930 }
931 % if the matrix is infinite, add a final row
932 \bool_if:NT \l__physicx_matrix_infinite_bool
933 {
934   % finish up the row
935   \tl_put_right:Nx \l__physicx_matrix_body_tl { \[\dim_use:N \l__physicx_matrix_sep_d
936   \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
937   \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
938     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
939   \bool_if:NT \l__physicx_matrix_dotcol_bool
940     { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
941   \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
942   % update cols
943   \bool_if:NTF \l__physicx_matrix_dotcol_bool
944     { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
945     { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
946 }
947 }

```

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

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
948   \cs_new:Npn \physicx_matrix_diag_parse:n #1
949   {
950     \keyval_parse:nnn
951     \__physicx_matrix_diag_parse_aux:n

```

```

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

```

```

1006     \l__physicx_matrix_cols_int
1007 }
1008 \cs_new:Npn \__physicx_diagonalmatrix_no_diag:
1009 {
1010   \seq_map_indexed_inline:Nn \l__cwamcro_physics_tmpdiag_seq
1011   { \physicx_matrix_set_r_c:nnn {##1} {##1} {##2} }
1012   \__physicx_matrix_diag_calc:nn
1013   { \seq_count:N \l__cwamcro_physics_tmpdiag_seq }
1014   { \seq_count:N \l__cwamcro_physics_tmpdiag_seq }
1015 }
1016 \cs_new_eq:NN \__physicx_diagonalmatrix_diag_main:
1017   \__physicx_diagonalmatrix_no_diag:
1018 \cs_new:Npn \__physicx_matrix_diag_parse_aux_regu:n #1
1019 {
1020   \if_int_compare:w #1 = 0 \exp_stop_f:
1021     \__physicx_diagonalmatrix_diag_main:
1022   \else:
1023     \if_int_compare:w #1 > 0 \exp_stop_f:
1024       \seq_map_indexed_inline:Nn \l__cwamcro_physics_tmpdiag_seq
1025       {
1026         \physicx_matrix_set_r_c:nnn
1027         {##1} { \int_eval:n { ##1 + #1 } } {##2}
1028       }
1029       \__physicx_matrix_diag_calc:nn
1030       { \seq_count:N \l__cwamcro_physics_tmpdiag_seq }
1031       { \seq_count:N \l__cwamcro_physics_tmpdiag_seq + #1 }
1032     \else:
1033       \seq_map_indexed_inline:Nn \l__cwamcro_physics_tmpdiag_seq
1034       {
1035         \physicx_matrix_set_r_c:nnn
1036         { \int_eval:n { ##1 - #1 } } {##1} {##2}
1037       }
1038       \__physicx_matrix_diag_calc:nn
1039       { \seq_count:N \l__cwamcro_physics_tmpdiag_seq - #1 }
1040       { \seq_count:N \l__cwamcro_physics_tmpdiag_seq }
1041     \fi:
1042   \fi:
1043 }
1044 \cs_new:Npn \__physicx_matrix_diag_parse_aux_anti:n #1
1045 {
1046   \if_int_compare:w #1 = 0 \exp_stop_f:
1047     \__physicx_matrix_diag_calc:nn
1048     { \seq_count:N \l__cwamcro_physics_tmpdiag_seq }
1049     { \seq_count:N \l__cwamcro_physics_tmpdiag_seq }
1050   \seq_map_indexed_inline:Nn \l__cwamcro_physics_tmpdiag_seq
1051   {
1052     \physicx_matrix_set_r_c:nnn
1053     {##1}
1054     { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1055     {##2}
1056   }
1057   \else:
1058     \if_int_compare:w #1 > 0 \exp_stop_f:
1059     \__physicx_matrix_diag_calc:nn

```

```

1060         { \seq_count:N \l__cwamcro_physics_tmpdiag_seq }
1061         { \seq_count:N \l__cwamcro_physics_tmpdiag_seq + #1 }
1062     \seq_map_indexed_inline:Nn \l__cwamcro_physics_tmpdiag_seq
1063     {
1064         \physicx_matrix_set_r_c:nnn
1065         {##1}
1066         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1067         {##2}
1068     }
1069     \else:
1070         \__physicx_matrix_diag_calc:nn
1071         { \seq_count:N \l__cwamcro_physics_tmpdiag_seq - #1 }
1072         { \seq_count:N \l__cwamcro_physics_tmpdiag_seq }
1073     \seq_map_indexed_inline:Nn \l__cwamcro_physics_tmpdiag_seq
1074     {
1075         \physicx_matrix_set_r_c:nnn
1076         { \int_eval:n { ##1 - #1 } }
1077         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1078         {##2}
1079     }
1080     \fi:
1081     \fi:
1082 }
1083 \cs_new:Npn \__physicx_matrix_diag_calc:nn
1084 { \__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
1085 \cs_new:Npn \physicx_matrix_item_parse:n #1
1086 {
1087     \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1088     \keyval_parse:NNn
1089     \__physicx_matrix_item_parse_aux:n
1090     \__physicx_matrix_item_parse_aux:nn
1091     {#1}
1092 }
1093 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }
1094 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1095 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2
1096 {
1097     \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1098     \tl_set:Nx \l__physicx_tmpitem_tl
1099     { \__physicx_expand:w \l__physicx_tmpitem_tl }
1100     \physicx_parse_range:nxN \l__physicx_matrix_rows_int
1101     { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1102     \physicx_parse_range:nxN \l__physicx_matrix_cols_int
1103     { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1104     \exp_args:No \tl_if_eq:nnTF
1105     { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1106     {
1107         \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1108         {
1109             \seq_map_inline:Nn \l__physicx_tmp_colnum_seq

```

```

1110         {
1111             \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1112         }
1113     }
1114 }
1115 {
1116     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1117     {
1118         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1119         {
1120             \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }
1121             {
1122                 \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1123                 {##1} {####1} { \l__physicx_tmpitem_tl }
1124             }
1125         }
1126     }
1127 }
1128 }

```

(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 1129 \cs_new:Npn \physicx_matrix_array_parse:n #1
1130 {
1131     \tl_set:Nn \l__physicx_tmparr_tl {#1}
1132     \tl_set:Nx \l__physicx_tmparr_tl
1133     { \__physicx_expand:w \l__physicx_tmparr_tl }
1134     \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl
1135     \__physicx_matrix_autocalc:nn
1136     { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1137     { 0 }
1138     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1139     {
1140         \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1141         \__physicx_matrix_autocalc:nn
1142         { 0 }
1143         { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1144         \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1145         {
1146             \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1147         }
1148     }
1149 }
1150 \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.

```

1151 \cs_new:Npn \physicx_matrix_array_parse_main:
1152 {
1153     \int_step_inline:nn \l__physicx_matrix_rows_int
1154     {
1155         \int_step_inline:nn \l__physicx_matrix_cols_int
1156         {

```

```

1157         \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1158         {##1} {####1} \l__physicx_matrix_main_tl
1159     }
1160 }
1161 }

```

(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).

```

1162 \prg_new_conditional:Npnn \_physicx_if_can_num:n #1 { T, F, TF }
1163 {
1164     \physicx_if_num:nTF {#1}
1165     { \prg_return_true: }
1166     {
1167         \bool_case_true:nTF
1168         {
1169             { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1170             { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1171             {
1172                 \bool_lazy_and_p:nn
1173                 { \cs_if_exist_p:N \inteval }
1174                 { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1175             } { }
1176             {
1177                 \bool_lazy_and_p:nn
1178                 { \cs_if_exist_p:N \fpeval }
1179                 { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1180             } { }
1181         }
1182         { \prg_return_true: }
1183         { \prg_return_false: }
1184     }
1185 }

```

(End definition for _physicx_if_can_num:n.)

\diagonalmatrix Define \diagonalmatrix.

```

1186 \DeclareDocumentCommand \diagonalmatrix { t= t+ 0{ } m }
1187 {
1188     \group_begin:
1189     \IfBooleanTF {#1}
1190     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1191     { \keys_set:nn { physicx/matrix } { #3 } }
1192     \physicx_construct:nnn { }
1193     {
1194         \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist
1195         \tl_if_empty:nF {#4}
1196         {
1197             \_physicx_if_keyval:nTF {#4}
1198             { \physicx_matrix_diag_parse:n { true, #4 } }
1199             { \physicx_matrix_diag_parse:n { true, 0 = {#4} } }
1200         }
1201     }

```



```

1202     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1203 \bool_lazy_or:nnTF
1204   { \bool_if_p:n {#2} }
1205   { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1206   {
1207     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1208     {
1209       \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1210       \__physicx_matrix_appto_body_e:off
1211     }
1212     {
1213       \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1214       \__physicx_matrix_appto_body_ne:off
1215     }
1216     \use_i_ii:nnn
1217   }
1218   { \use_i:nn }
1219   \__physicx_matrix_transpose:N
1220   \__physicx_diagonalmatrix_generate_enhanced_body:NNN
1221   \__physicx_diagonalmatrix_generate_body:NNN
1222 \__physicx_matrix_save_or_print:
1223 \group_end:
1224 }
1225 \cs_new:Npn \__physicx_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1226 {
1227   \__physicx_matrix_generate_body:NNNN #1#2#3
1228   \__physicx_diagonalmatrix_enhanced:nnn
1229 }
1230 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1231 {
1232   \int_step_inline:nn { #1 - 1 }
1233   {
1234     \int_step_inline:nn { #2 - 1 }
1235     {
1236       \tl_put_right:Nx \l__physicx_matrix_body_tl
1237       {
1238         \exp_after:wN
1239         \physicx_matrix_use_r_c:nn
1240         #3 {{##1}} {{####1}} &
1241       }
1242     }
1243     \tl_put_right:Nx \l__physicx_matrix_body_tl
1244     {
1245       \exp_after:wN
1246       \physicx_matrix_use_r_c:nn
1247       #3 {{##1}} {{ \int_use:N #2 }} \[\dim_use:N \l__physicx_matrix_sep_dim]
1248     }
1249   }
1250   \int_step_inline:nn { #2 - 1 }
1251   {
1252     \tl_put_right:Nx \l__physicx_matrix_body_tl
1253     {
1254       \exp_after:wN
1255       \physicx_matrix_use_r_c:nn

```

```

1256         #3 {{ \int_use:N #1 }} {{##1}} &
1257     }
1258 }
1259 \tl_put_right:Nx \l__physicx_matrix_body_tl
1260 {
1261     \exp_after:wN
1262     \physicx_matrix_use_r_c:nn
1263     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1264 }
1265 }

```

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

`__physicx_declare_init:`

```

1266 \cs_new:Npn \__physicx_matrix_enhanced_init:
1267 {
1268     \seq_if_empty:NF \l__physicx_row_list_seq
1269     {
1270         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1271         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1272         { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1273     }
1274     \seq_if_empty:NF \l__physicx_col_list_seq
1275     {
1276         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1277         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1278         { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1279     }
1280 }

```

(End definition for `__physicx_declare_init:`.)

`\commamatrix` Define `\commamatrix`.

```

1281 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1282 {
1283     \group_begin:
1284     \keys_set:nn { physicx/matrix } {#3}
1285     \tl_if_empty:nF {#4}
1286     { \keys_set:nn { physicx/matrix } { array = {#4} } }
1287     \IfBooleanT {#1}
1288     { \keys_set:nn { physicx/matrix } { saveto = \physicxtmp } }
1289     \tl_set:Nx \l__physicx_matrix_array_tl
1290     { \__physicx_expand:w \l__physicx_matrix_array_tl }
1291     \bool_lazy_or:nnTF
1292     { \bool_if_p:n {#2} }
1293     { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1294     { \__physicx_commamatrix_enhanced: }
1295     {
1296         \tl_replace_all:Nox \l__physicx_matrix_array_tl
1297         { \physicx@cr } { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1298         \tl_replace_all:Non \l__physicx_matrix_array_tl
1299         { \physicx@align } { & }
1300         \tl_set_eq:NN \l__physicx_matrix_body_tl
1301         \l__physicx_matrix_array_tl
1302     }

```

```

1303     \__physicx_matrix_save_or_print:
1304     \group_end:
1305 }
1306 \cs_new_nopar:Npn \__physicx_matrix_save_or_print:
1307 {
1308     \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1309     {
1310         \exp_after:wN \tl_gset_eq:NN
1311         \l__physicx_matrix_save_tl
1312         \l__physicx_matrix_body_tl
1313     }
1314     {
1315         \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1316         \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1317         \fi:
1318         \exp_after:wN \__physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matrix_body_tl
1319         \l__physicx_matrix_body_tl
1320         \__physicx_matrix_end: \l__physicx_matrix_after_end_tl
1321     }
1322 }
1323 \cs_new:Npn \__physicx_commamatrix_enhanced:
1324 {
1325     \tl_clear:N \l__physicx_matrix_body_tl
1326     \int_zero:N \l__cwamcro_physics_tmpa_int
1327     \seq_set_split:NVV \l__cwamcro_physics_tmp_seq \physicx@cr
1328     \l__physicx_matrix_array_tl
1329     \int_set:Nn \l__physicx_matrix_rows_int
1330     { \seq_count:N \l__cwamcro_physics_tmp_seq }
1331     \__physicx_matrix_enhanced_init:
1332     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1333     {
1334         \seq_map_tokens:Nn \l__cwamcro_physics_tmp_seq
1335         {
1336             \int_incr:N \l__cwamcro_physics_tmpa_int
1337             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1338             \l__cwamcro_physics_tmpa_int \__physicx_commamatrix_enhanced_aux_e:nnn
1339         }
1340     }
1341     {
1342         \seq_map_tokens:Nn \l__cwamcro_physics_tmp_seq
1343         {
1344             \int_incr:N \l__cwamcro_physics_tmpa_int
1345             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1346             \l__cwamcro_physics_tmpa_int \__physicx_commamatrix_enhanced_aux_ne:nnn
1347         }
1348     }
1349 }
1350 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1351 {
1352     \seq_set_split:Non \l__physicx_tmp_col_seq
1353     { \physicx@align } {#3}
1354     \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1355     \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1356     { #2 {##2} {#1} {##1} }

```

```

1357 \tl_put_right:Nx \l__physicx_matrix_body_tl
1358 {
1359   \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1360   \if_int_compare:w \l__physicx_matrix_rows_int = #1
1361     \scan_stop:
1362   \else:
1363     \[\dim_use:N \l__physicx_matrix_sep_dim]
1364   \fi:
1365 }
1366 }
1367 \cs_new:Npn \__physicx_commmatrix_enhanced_aux_e:nnn #1#2#3
1368 {
1369   \seq_put_right:Nx \l__physicx_tmp_coled_seq
1370   {
1371     \text_expand:n % \text_expand:n do the magic thing, but slower
1372     {
1373       \physicx@matrixellement { #1 }
1374       { \__physicx_matrix_row_iterate:n {#2} }
1375       { \__physicx_matrix_col_iterate:n {#3} }
1376     }
1377   }
1378 }
1379 \cs_new:Npn \__physicx_commmatrix_enhanced_aux_ne:nnn #1#2#3
1380 {
1381   \seq_put_right:No \l__physicx_tmp_coled_seq
1382   {
1383     \physicx@matrixellement {#1}
1384     { \__physicx_matrix_row_iterate:n {#2} }
1385     { \__physicx_matrix_col_iterate:n {#3} }
1386   }
1387 }

```

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

\generalmatrix Define \generalmatrix.

```

1388 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1389 {
1390   \IfBooleanTF {#2}
1391   {
1392     \group_begin:
1393     \IfBooleanTF {#1}
1394     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicx_tmp } }
1395     { \keys_set:nn { physicx/matrix } {#4} }
1396     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1397     \physicx_construct:nnn
1398     {
1399       \tl_if_empty:NTF \l__physicx_matrix_main_tl
1400       {
1401         \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1402       }
1403       { \physicx_matrix_array_parse_main: }
1404     }
1405     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1406     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }

```

```

1407     \__physicx_generalmatrix:
1408     \__physicx_matrix_save_or_print:
1409     \group_end:
1410 }
1411 {
1412     \IfBooleanTF {#1}
1413     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1414     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1415     \qmatrix = * [#4]
1416 }
1417 }
1418 \cs_new:Npn \__physicx_generalmatrix:
1419 {
1420     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1421     {
1422         \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1423         \__physicx_matrix_appto_body_e:off
1424     }
1425     {
1426         \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1427         \__physicx_matrix_appto_body_ne:off
1428     }
1429     \__physicx_matrix_transpose:N
1430     \__physicx_matrix_generate_body:NNNN
1431     \__physicx_generalmatrix_generate:nnn
1432 }

```

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

`__physicx_matrix_generate_body:NNNN`

```

1433 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1434 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1435 {
1436     \__physicx_matrix_enhanced_init:
1437     \int_step_inline:nn { #1 - 1 }
1438     {
1439         \int_step_inline:nn { #2 - 1 }
1440         {
1441             \tl_set:Nx \l__physicx_tmp_tl
1442             {
1443                 \exp_after:wN
1444                 \physicx_matrix_use_r_c:nn
1445                 #3 {{##1}} {{####1}}
1446             }
1447             #4 \l__physicx_tmp_tl {{##1}} {{####1}}
1448             \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1449         }
1450     }
1451     \tl_set:Nx \l__physicx_tmp_tl
1452     {
1453         \exp_after:wN
1454         \physicx_matrix_use_r_c:nn
1455         #3 {{##1}} {{ \int_use:N #2 }}
1456     }
1457     #4 \l__physicx_tmp_tl {{##1}} { \int_use:N #2 }

```

```

1457         \tl_put_right:Nx \l__physicx_matrix_body_tl
1458         { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1459     }
1460     \int_step_inline:nn { #2 - 1 }
1461     {
1462         \tl_set:Nx \l__physicx_tmp_tl
1463         {
1464             \exp_after:wN
1465             \physicx_matrix_use_r_c:nn
1466             #3 {{ \int_use:N #1 }} {{##1}}
1467         }
1468         #4 \l__physicx_tmp_tl { \int_use:N #1 } {{##1}}
1469         \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1470     }
1471     \tl_set:Nx \l__physicx_tmp_tl
1472     {
1473         \exp_after:wN
1474         \physicx_matrix_use_r_c:nn
1475         #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1476     }
1477     #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1478 }

```

(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
1479 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1480 {
1481     \tl_put_right:Nx \l__physicx_matrix_body_tl
1482     {
1483         \text_expand:n
1484         {
1485             \physicx@matricelement {#1}
1486             { \__physicx_matrix_row_iterate:n {#2} }
1487             { \__physicx_matrix_col_iterate:n {#3} }
1488         }
1489     }
1490 }
1491 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1492 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1493 {
1494     \tl_put_right:No \l__physicx_matrix_body_tl
1495     {
1496         \physicx@matricelement {#1}
1497         { \__physicx_matrix_row_iterate:n {#2} }
1498         { \__physicx_matrix_col_iterate:n {#3} }
1499     }
1500 }
1501 \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

```

1502 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1503 {

```

```

1504 \bool_if:NTF \l__physicx_matrix_transpose_bool
1505 {
1506     #1
1507     \l__physicx_matrix_cols_int
1508     \l__physicx_matrix_rows_int
1509     \use_ii_i:nn
1510 }
1511 {
1512     #1
1513     \l__physicx_matrix_rows_int
1514     \l__physicx_matrix_cols_int
1515     \use:nn
1516 }
1517 }

```

(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.

```

1518 \cs_new:Npn \physicx_construct:nnn #1#2#3
1519 {
1520     \l__physicx_matrix_beginning_tl
1521     \__physicx_adi:nnn {#1} {#2} {#3}
1522     \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1523     {
1524         \int_incr:N \l__physicx_matrix_cols_int
1525         \__physicx_matrix_last_aux_c:
1526         \int_incr:N \l__physicx_matrix_cols_int
1527     }
1528     \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1529     {
1530         \int_incr:N \l__physicx_matrix_rows_int
1531         \__physicx_matrix_last_aux_r:
1532         \int_incr:N \l__physicx_matrix_rows_int
1533     }
1534     \bool_lazy_or:nnF
1535     { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1536     { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1537     {
1538         \physicx_matrix_set_r_c:nnn
1539         { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1540         { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1541         { \ddots }
1542     }
1543     \bool_if:NT \l__physicx_matrix_infinite_bool
1544     {
1545         \int_incr:N \l__physicx_matrix_rows_int
1546         \int_incr:N \l__physicx_matrix_cols_int
1547         \__physicx_matrix_last_aux_c:
1548         \__physicx_matrix_last_aux_r:
1549         \physicx_matrix_set_r_c:nnn
1550         { \int_use:N \l__physicx_matrix_rows_int }
1551         { \int_use:N \l__physicx_matrix_cols_int }
1552         { \ddots }

```

```

1553     }
1554     \l__physicx_matrix_ending_tl
1555   }
1556   \cs_new:Npn \__physicx_matrix_last_aux_c:
1557   {
1558     \int_step_inline:nn \l__physicx_matrix_rows_int
1559     {
1560       \physicx_matrix_set_r_c:nnn
1561       {##1} { \int_use:N \l__physicx_matrix_cols_int }
1562       { \cdots }
1563     }
1564   }
1565   \cs_new:Npn \__physicx_matrix_last_aux_r:
1566   {
1567     \int_step_inline:nn \l__physicx_matrix_cols_int
1568     {
1569       \physicx_matrix_set_r_c:nnn
1570       { \int_use:N \l__physicx_matrix_rows_int } {##1}
1571       { \vdots }
1572     }
1573   }

```

(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 1574 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
  \NewGeneralMatrix 1575 {
  \newdiagonalmatrix 1576   \NewDocumentCommand #2 { t+ m o o m m }
  \NewDiagonalMatrix 1577   {
  \newcommamatrix 1578     \IfBooleanTF {##1}
  \NewCommaMatrix 1579     {
1580       \IfNoValueTF {##3}
1581       { \newcommand ##2 { #1 + [##5] {##6} } }
1582       {
1583         \IfNoValueTF {##4}
1584         { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1585         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1586       }
1587     }
1588   {
1589     \IfNoValueTF {##3}
1590     { \newcommand ##2 { #1 [##5] {##6} } }
1591     {
1592       \IfNoValueTF {##4}
1593       { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1594       { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1595     }
1596   }
1597 }
1598 \NewDocumentCommand #3 { t+ m m m m }
1599 {
1600   \IfBooleanTF {##1}

```



```

1601         { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1602         { \NewDocumentCommand ##2 {##3} { #1    [##4] {##5} } }
1603     }
1604 }
1605 \__physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1606 \__physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1607 \NewDocumentCommand \newgeneralmatrix { t+ m o o m }
1608 {
1609     \IfBooleanTF {#1}
1610     {
1611         \IfNoValueTF {#3}
1612         { \newcommand #2 { \generalmatrix + {#5} } }
1613         {
1614             \IfNoValueTF {#4}
1615             { \newcommand #2 [#3] { \generalmatrix + {#5} } }
1616             { \newcommand #2 [#3] [#4] { \generalmatrix + {#5} } }
1617         }
1618     }
1619     {
1620         \IfNoValueTF {#3}
1621         { \newcommand #2 { \generalmatrix {#5} } }
1622         {
1623             \IfNoValueTF {#4}
1624             { \newcommand #2 [#3] { \generalmatrix {#5} } }
1625             { \newcommand #2 [#3] [#4] { \generalmatrix {#5} } }
1626         }
1627     }
1628 }
1629 \NewDocumentCommand \NewGeneralMatrix { t+ m m m }
1630 {
1631     \IfBooleanTF {#1}
1632     { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1633     { \NewDocumentCommand #2 {#3} { \generalmatrix    {#4} } }
1634 }

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

1635 </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	
\! 414
\+ 153
\- 153
\ 43, 897, 904, 911, 935, 1247, 1297, 1363, 1458
\{ 350, 369, 387, 426, 459
\} 350, 371, 387, 426, 459
	A
\A 148, 153
\abs <u>420</u>
\absolutevalue <u>420</u>
\acomm <u>420</u>

<code>\anticommutator</code>	420	<code>\clist_if_in:NnTF</code>	1120
B			
<code>\begin</code> 370, 373, 377, 380, 381, 382, 387, 388, 392, 395, 396, 397, 754, 755, 756, 757, 758, 759, 760, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779		<code>\clist_map_break:n</code>	162 , 174
<code>\Big</code>	402	<code>\clist_map_inline:nn</code> 68, 158, 170, 790	
<code>\big</code>	402	<code>\clist_new:N</code>	521 , 522
<code>\Bigg</code>	402	<code>\clist_put_right:Nn</code> ..	662 , 668 , 1111
<code>\bigg</code>	402	<code>\clist_set:Nn</code>	782
<code>\Biggl</code>	409	<code>\clist_set_eq:NN</code>	1087
<code>\biggl</code>	408	<code>\c_empty_clist</code>	1087
<code>\Biggr</code>	409	<code>\comm</code>	457
<code>\biggr</code>	408	<code>\commamatrix</code>	26 , 1281 , 1606
<code>\Bigl</code>	407	<code>\commutator</code>	420
<code>\bigl</code>	406	cs commands:	
<code>\Bigr</code>	407	<code>\cs:w</code>	31 , 32 , 691
<code>\bigr</code>	406	<code>\cs_end:</code>	31 , 32 , 476 , 691
<code>\boldsymbol</code>	190	<code>\cs_generate_variant:Nn</code>	
bool commands:		3 , 4 , 5 , 6 , 75 , 78 , 184 , 332 , 955 , 1093 , 1150 , 1491 , 1501
<code>\bool_case_false:n</code>	321 , 322 , 329	<code>\cs_if_exist_p:N</code>	1173 , 1178
<code>\bool_case_true:nTF</code>	404 , 1167	<code>\cs_new:Npn</code>	7 , 110 , 118 , 121 , 133 , 144 , 156 , 188 , 192 , 196 , 242 , 252 , 264 , 301 , 308 , 333 , 400 , 548 , 738 , 743 , 948 , 956 , 984 , 997 , 1008 , 1018 , 1044 , 1083 , 1085 , 1094 , 1095 , 1129 , 1151 , 1225 , 1230 , 1266 , 1323 , 1350 , 1367 , 1379 , 1418 , 1434 , 1479 , 1492 , 1502 , 1518 , 1556 , 1565 , 1574
<code>\bool_if:NnTF</code>	15 , 20 , 25 , 71 , 181 , 335 , 337 , 341 , 829 , 884 , 891 , 901 , 909 , 921 , 928 , 932 , 939 , 943 , 1207 , 1332 , 1420 , 1504 , 1543	<code>\cs_new_eq:NN</code>	
<code>\bool_if_p:N</code>	1205 , 1293	120 , 143 , 507 , 527 , 536 , 1016
<code>\bool_if_p:n</code>		<code>\cs_new_nopar:Npn</code> 466 , 474 , 483 , 485 , 490 , 496 , 529 , 530 , 533 , 534 , 535 , 1306	
.....	406 , 407 , 408 , 409 , 1204 , 1292	<code>\cs_new_protected:Nn</code>	827
<code>\bool_lazy_and_p:nn</code>	1172 , 1177	<code>\cs_new_protected:Npn</code> 53 , 58 , 63 , 76 , 79	
<code>\bool_lazy_or:nnTF</code>		<code>\cs_new_protected_nopar:Npn</code>	
.....	30 , 112 , 123 , 1203 , 1291 , 1534	781 , 783 , 788
<code>\bool_new:N</code> 9, 10, 11, 12, 37, 52, 237, 239 , 513 , 514 , 515 , 523 , 537 , 545 , 546		<code>\cs_set:Nn</code> .	705 , 706 , 707 , 708 , 709 , 710
<code>\bool_set:Nn</code>	248 , 249 , 1396	<code>\cs_set:Npn</code>	422 , 432 , 433 , 443 , 444 , 451 , 452 , 457 , 462 , 463 , 464 , 589 , 598 , 607 , 616 , 625
<code>\bool_set_false:N</code>		<code>\cs_set:Npx</code>	632
.....	81 , 169 , 218 , 817 , 845 , 860 , 977	<code>\cs_set_eq:NN</code> 55 , 56 , 60 , 61 , 190 , 194 , 198 , 505 , 558 , 560 , 562 , 565 , 571 , 573 , 587 , 596 , 605 , 614 , 623 , 641 , 644 , 647 , 650 , 653 , 831 , 835 , 962 , 967 , 972 , 978 , 1209 , 1213 , 1422 , 1426	
<code>\bool_set_inverse:N</code>	583 , 700	<code>\cs_set_nopar:Npn</code>	1271 , 1277
<code>\bool_set_true:N</code>		<code>\cs_set_protected:Npn</code>	
.....	115 , 126 , 136 , 176 , 214 , 217 , 816 , 849 , 853 , 864 , 868 , 971 , 1270 , 1276	313 , 326 , 345 , 418
<code>\c_false_bool</code>	365 , 385	<code>\cs_to_str:N</code>	258
<code>\c_true_bool</code>	348 , 358 , 365 , 385	cwamcro internal commands:	
<code>\Bqty</code>	420	<code>\l_cwamcro_physics_tmp_seq</code>	
<code>\bqty</code>	420	1327 , 1330 , 1334 , 1342
C			
<code>\cdots</code>	600 , 883 , 886 , 887 , 890 , 892 , 923 , 924 , 927 , 929 , 1562	<code>\l_cwamcro_physics_tmpa_bool</code> ...	
clist commands:		37 , 169 , 176 , 181
<code>\clist_clear:N</code>	785		
<code>\clist_concat:NNN</code>	792		

<code>\l_cwamcro_physics_tmpa_int</code> 38, 1326, 1336, 1338, 1344, 1346 <code>\l_cwamcro_physics_tmpb_int</code> ... 39 <code>\l_cwamcro_physics_tmpdiag_seq</code> 989, 1000, 1010, 1013, 1014, 1024, 1030, 1031, 1033, 1039, 1040, 1048, 1049, 1050, 1060, 1061, 1062, 1071, 1072, 1073 <code>\l_cwamcro_physics_tmpdiag_tl</code> 986, 987, 988, 989	
D	
<code>\ddots</code> 600, 910, 941, 1541, 1552 <code>\DeclareDocumentCommand</code> 318, 328, 402, 812, 1186, 1281, 1388 <code>\diagonalmatrix</code> 24, 1186, 1605 dim commands: <code>\dim_new:N</code> 547 <code>\dim_use:N</code> 897, 904, 911, 935, 1247, 1297, 1363, 1458	
E	
else commands: <code>\else:</code> 478, 895, 1022, 1032, 1057, 1069, 1362 <code>\end</code> .. 370, 373, 377, 380, 381, 382, 387, 388, 392, 395, 396, 397, 754, 755, 756, 757, 758, 759, 760, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779 <code>\eval</code> 420 <code>\evaluated</code> 347, 432 exp commands: <code>\exp_after:wN</code> 1238, 1245, 1254, 1261, 1308, 1310, 1318, 1443, 1452, 1464, 1473 <code>\exp_args:Nf</code> 992 <code>\exp_args:Nnno</code> 1122, 1157 <code>\exp_args:Nno</code> 160, 172 <code>\exp_args:No</code> 618, 627, 727, 1104 <code>\exp_args:NV</code> 1337, 1345 <code>\exp_not:N</code> 8, 272, 632 <code>\exp_not:n</code> 276, 277, 477, 479, 527, 558, 562 <code>\exp_stop_f:</code> 8, 267, 274, 1020, 1023, 1046, 1058	
F	
fi commands: <code>\fi:</code> 278, 293, 298, 299, 480, 898, 1041, 1042, 1080, 1081, 1317, 1364 fp commands: <code>\fp_eval:n</code> 24, 1170 <code>\fpeval</code> 24, 1178, 1179	
G	
<code>\generalmatrix</code> .. 28, 1388, 1612, 1615, 1616, 1621, 1624, 1625, 1632, 1633 group commands: <code>\group_begin:</code> .. 814, 1188, 1283, 1392 <code>\group_end:</code> 825, 1223, 1304, 1409	
I	
if commands: <code>\if_bool:N</code> 282, 294 <code>\if_case:w</code> 274 <code>\if_cs_exist:w</code> 476 <code>\if_int_compare:w</code> 267, 893, 1020, 1023, 1046, 1058, 1315, 1360 <code>\IfBooleanT</code> 369, 371, 1287 <code>\IfBooleanTF</code> 231, 320, 376, 378, 391, 393, 412, 750, 815, 820, 1189, 1390, 1393, 1412, 1413, 1414, 1578, 1600, 1609, 1631 <code>\IfNoValueTF</code> 1580, 1583, 1589, 1592, 1611, 1614, 1620, 1623 int commands: <code>\int_compare:nNnTF</code> 91, 101, 135 <code>\int_compare:nTF</code> 843, 858 <code>\int_compare_p:nNn</code> . 113, 114, 124, 125 <code>\int_eval:n</code> .. 24, 1027, 1036, 1054, 1066, 1076, 1077, 1169, 1539, 1540 <code>\int_incr:N</code> . 266, 1002, 1336, 1344, 1524, 1526, 1530, 1532, 1545, 1546 <code>\int_max:nn</code> 469, 471 <code>\int_new:N</code> 38, 39, 48, 49, 50, 51, 241, 518, 519 <code>\int_set:Nn</code> 66, 67, 90, 100, 247, 468, 470, 818, 819, 846, 861, 1329 <code>\int_set_eq:NN</code> 88, 93, 98, 103, 1005, 1316 <code>\int_step_inline:nn</code> 871, 1153, 1155, 1232, 1234, 1250, 1437, 1439, 1460, 1558, 1567 <code>\int_step_inline:nnn</code> 128, 138, 877, 915 <code>\int_use:N</code> 1247, 1256, 1263, 1454, 1456, 1466, 1468, 1475, 1477, 1550, 1551, 1561, 1570 <code>\int_zero:N</code> 999, 1326 <code>\interval</code> 24, 1173, 1174	
K	
keys commands: <code>\keys_define:nn</code> 200, 551, 553, 740, 745 <code>\keys_if_exist:nnTF</code> 160, 172 <code>\l_keys_key_str</code> 160, 163, 172, 177, 727, 730 <code>\l_keys_path_str</code> 734	

<code>\keys_set:nn</code>	3, 163, 177, 232, 233, 552, 729, 798, 821, 822, 1190, 1191, 1284, 1286, 1288, 1394, 1395
keyval commands:	
<code>\keyval_parse:NNn</code>	1088
<code>\keyval_parse:nnn</code>	950
L	
<code>\langle</code>	354, 381, 396
<code>\ldots</code>	600
<code>\left</code>	316, 369, 376, 381, 387, 388, 391, 395, 396, 397, 414
<code>\let</code>	315
<code>\lgroup</code>	376, 391
<code>\lVert</code>	436
M	
<code>\mathcal</code>	186, 187
<code>\matrixquantity</code>	347
<code>\mqty</code>	420
msg commands:	
<code>\msg_error:nnn</code>	982
<code>\msg_error:nnnn</code>	733
<code>\msg_new:nnn</code>	46
<code>\msg_new:nnnn</code>	40
N	
<code>\NewCommaMatrix</code>	1574
<code>\newcommamatrix</code>	1574
<code>\newcommand</code>	1581, 1584, 1585, 1590, 1593, 1594, 1612, 1615, 1616, 1621, 1624, 1625
<code>\NewDiagonalMatrix</code>	1574
<code>\newdiagonalmatrix</code>	1574
<code>\NewDocumentCommand</code>	229, 748, 1576, 1598, 1601, 1602, 1607, 1629, 1632, 1633
<code>\NewGeneralMatrix</code>	1574
<code>\newgeneralmatrix</code>	1574
<code>\nobreak</code>	360, 361, 362
<code>\norm</code>	420
O	
<code>\OOrder</code>	444, 452
<code>\oorder</code>	420
or commands:	
<code>\or:</code>	275, 276, 277
<code>\Order</code>	420
<code>\order</code>	420
<code>\Ordersymbol</code>	223
<code>\ordersymbol</code>	222
P	
<code>\pb</code>	420
peek commands:	
<code>\peek_charcode_ignore_spaces:N</code>	315
physicx commands:	
<code>\physicx_bf:</code>	190, 194, 198
<code>\physicx_compat:</code>	13
<code>\physicx_compat:TF</code>	220, 437
<code>\physicx_construct:nnn</code>	1192, 1397, 1518
<code>\physicx_declare_legacy_paren:NnnnNNn</code>	399, 423, 424, 425, 426, 428, 435, 439, 441, 447, 449, 454, 458
<code>\physicx_declare_legacy_quantity:nnNn</code>	235, 347, 357, 364, 384
<code>\physicx_if_num:n</code>	146
<code>\physicx_if_num:nTF</code>	727, 841, 856, 1164
<code>\physicx_if_num_sign:n</code>	151
<code>\physicx_mathtools:</code>	23
<code>\physicx_mathtools:TF</code>	761
<code>\physicx_matrix_array_parse:n</code>	1129, 1401
<code>\physicx_matrix_array_parse_-</code> main:	1151, 1403
<code>\physicx_matrix_diag_parse:n</code>	663, 948, 1194, 1198, 1199, 1405
<code>\physicx_matrix_item_parse:n</code>	669, 1085, 1202, 1406
<code>\physicx_matrix_new_type:nn</code>	738
<code>\physicx_matrix_new_type:nnn</code>	738
<code>\physicx_matrix_set_r_c:nnn</code>	507, 641, 644, 647, 650, 653, 1003, 1011, 1026, 1035, 1052, 1064, 1075, 1122, 1146, 1157, 1538, 1549, 1560, 1569
<code>\physicx_matrix_use_r_c:nn</code>	474, 1239, 1246, 1255, 1262, 1444, 1453, 1465, 1474
<code>\physicx_option_or:nn</code>	28
<code>\physicx_option_or:nnTF</code>	420, 430, 456, 460
<code>\c_physicx_Order_tl</code>	187, 223, 440, 448
<code>\c_physicx_order_tl</code>	186, 222, 442, 450
<code>\physicx_parse_range:nnN</code>	76, 78, 1100, 1102
<code>\physicx_parse_range:nnnN</code>	48
<code>\physicx_parse_range_check:</code>	48, 673
<code>\physicx_parse_range_nocheck:</code>	48, 674
<code>\physicx_qxmatrix:nnn</code>	823, 827
<code>\physicx_search_also:nn</code>	156, 167, 184, 185
<code>\physicx_search_also:nnTF</code>	719
<code>\physicx_short:</code>	18
<code>\physicx_use_amssymb_type:</code>	188, 228
<code>\physicx_use_uni_bf_type:</code>	196
<code>\physicx_use_uni_bfit_type:</code>	192, 227

physicx internal commands:

- __physicx_adi:nnn 548, 705, 706, 707, 708, 709, 710, 1521
- \l__physicx_begin_int 48, 88, 90, 91, 93, 124, 125, 129, 135, 139
- \l__physicx_cmd_arg_int 241, 247, 266, 267, 274
- \l__physicx_cmd_arg_spec_tl 240, 246, 259, 268
- \l__physicx_cmd_auto_body_bool .. 239, 249, 294, 337, 341
- \l__physicx_cmd_auto_body_tl ... 238, 245, 261, 295, 296
- \l__physicx_cmd_noauto_body_bool 237, 248, 282, 335
- \l__physicx_cmd_noauto_after_tl .. 236, 244, 260, 283, 284
- \l__physicx_col_list_seq 525, 579, 1274, 1278
- __physicx_commamatrix_enhanced: 1294, 1323
- __physicx_commamatrix_enhanced_-aux:nNn 1337, 1345, 1350
- __physicx_commamatrix_enhanced_-aux_e:nnn 1338, 1367
- __physicx_commamatrix_enhanced_-aux_ne:nnn 1346, 1379
- \g__physicx_compat_bool 11, 15, 202, 214
- __physicx_declare_init: 1266
- __physicx_declare_init:nnn 242, 254
- __physicx_declare_legacy_-quantity_aux:nnnn 264, 305
- __physicx_declare_legacy_-quantity_aux:NNnnn .. 257, 308, 332
- __physicx_declare_legacy_-quantity_aux:nw 255, 301, 306
- __physicx_diagonalmatrix_diag_-main: 972, 978, 1016, 1021
- __physicx_diagonalmatrix_-enhanced:nnn 1209, 1213, 1228
- __physicx_diagonalmatrix_-generate_body:NNN 1221, 1230
- __physicx_diagonalmatrix_-generate_enhanced_body:NNN .. 1220, 1225
- __physicx_diagonalmatrix_no_-diag: 979, 1008, 1017
- __physicx_diagonalmatrix_set_-diag: 973, 997
- \l__physicx_end_int 49, 98, 100, 101, 103, 125, 129, 135, 139
- __physicx_expand:w 527, 558, 560, 562, 565, 988, 1099, 1133, 1290
- __physicx_generalmatrix: 1407, 1418
- __physicx_generalmatrix_-generate:nnn 1422, 1426, 1431
- __physicx_if_can_num:n 1162
- __physicx_if_keyval:nTF .. 144, 1197
- \l__physicx_invalid_range_bool .. 52, 71, 81, 115, 126, 136
- \l__physicx_item_ignore_clist ... 1087, 1111, 1120
- \g__physicx_mathtools_bool 9, 25, 217, 218
- __physicx_matrix_add_data:nn ... 665, 671, 786, 788
- \l__physicx_matrix_after_begin_-tl 543, 681, 683, 1318
- \l__physicx_matrix_after_end_tl . 544, 684, 686, 1320
- __physicx_matrix_appto_body_-e:nnn 832, 1210, 1423, 1479
- __physicx_matrix_appto_body_-ne:nnn 836, 1214, 1427, 1479
- \l__physicx_matrix_args_tl 542, 679, 680, 1318
- \l__physicx_matrix_array_tl 516, 555, 1289, 1290, 1296, 1298, 1301, 1328, 1401
- __physicx_matrix_autocalc:nn ... 536, 571, 573, 1084, 1135, 1141
- __physicx_matrix_begin:w 534, 676, 1318
- \l__physicx_matrix_beginning_tl . 549, 711, 713, 1520
- \l__physicx_matrix_body_tl 517, 839, 874, 879, 880, 886, 887, 892, 897, 904, 906, 908, 910, 911, 913, 917, 918, 923, 924, 929, 935, 936, 938, 940, 941, 1236, 1243, 1252, 1259, 1300, 1312, 1319, 1325, 1357, 1448, 1457, 1469, 1481, 1494
- __physicx_matrix_calc:nn 466, 571, 963
- __physicx_matrix_col_iterate:n . 533, 657, 1277, 1375, 1385, 1487, 1498
- \l__physicx_matrix_cols_int 470, 471, 519, 568, 819, 858, 861, 877, 907, 915, 937, 944, 945, 999, 1002, 1006, 1054, 1066, 1077, 1102, 1155, 1315, 1316, 1507, 1514, 1524, 1526, 1540, 1546, 1551, 1561, 1567
- \l__physicx_matrix_diag_bool ... 523, 971, 977
- __physicx_matrix_diag_calc:nn . 962, 967, 1012, 1029, 1038, 1047, 1059, 1070, 1083

\l__physicx_matrix_diag_clist ...	818, 843, 846, 871, 893, 1005, 1100,
..... 521, 660, 662, 1194, 1405	1153, 1329, 1360, 1508, 1513, 1530,
__physicx_matrix_diag_parse_-	1532, 1539, 1545, 1550, 1558, 1570
aux:n 951, 956	__physicx_matrix_save_or_print:
__physicx_matrix_diag_parse_- 824, 1222, 1303, 1306, 1408
aux:nn 952, 984	\l__physicx_matrix_save_tl 689, 691, 1308, 1311
__physicx_matrix_diag_parse_-	\l__physicx_matrix_sep_dim 547, 687, 897,
aux_anti:n 992, 1044	904, 911, 935, 1247, 1297, 1363, 1458
__physicx_matrix_diag_parse_-	__physicx_matrix_set_data:nn ... 664, 670, 783
aux_regu:n 995, 1018	__physicx_matrix_set_r_c_-
\l__physicx_matrix_dotcol_bool ..	ckall:nnn 505, 654
..... 515,	__physicx_matrix_set_r_c_-
860, 864, 868, 884, 909, 921, 939, 943	ckep:nnn 490, 645
\l__physicx_matrix_dotrow_bool ..	__physicx_matrix_set_r_c_-
..... 514, 845, 849, 853, 901	ckig:nnn 485, 648
__physicx_matrix_element_-	__physicx_matrix_set_r_c_-
aux:nnn 587, 593,	ckigep:nnn 496, 506, 651
596, 602, 605, 611, 614, 620, 623, 628	__physicx_matrix_set_r_c_-
__physicx_matrix_end: 677, 1320	nock:nnn 483, 508, 642
__physicx_matrix_end:w 535	\l__physicx_matrix_tmparr_c_sep . 1140, 1143, 1144
\l__physicx_matrix_ending_tl ...	\l__physicx_matrix_tmparr_r_sep . 1134, 1136, 1138
..... 550, 714, 716, 1554	__physicx_matrix_transpose:N ... 1219, 1429, 1502
\l__physicx_matrix_enhanced_bool	\l__physicx_matrix_transpose_-
..... 546, 697, 700, 1205, 1293	bool 545, 692, 1504
__physicx_matrix_enhanced_init:	\l__physicx_max_int 50, 67, 98, 101, 103, 113, 124
..... 1266, 1331, 1436	\l__physicx_min_int 51, 66, 88, 91, 93, 114
\l__physicx_matrix_expand_-	__physicx_nauto_case:nnnn . 310, 333
element_bool 537,	__physicx_new_matrix_cmd:NNN . 1574
636, 829, 1207, 1270, 1276, 1332, 1420	__physicx_parse_range_aux:n 70, 79
__physicx_matrix_generate_-	__physicx_parse_range_range: ... 56, 61, 106, 143
body:NNNN 1227, 1430, 1433	__physicx_parse_range_range_-
\l__physicx_matrix_infinite_bool	check: 56, 121, 143
..... 513, 580,	__physicx_parse_range_range_-
583, 816, 817, 891, 928, 932, 1396, 1543	nocheck: 61, 133
\l__physicx_matrix_item_clist ...	__physicx_parse_range_single:n . 55, 60, 108, 120
..... 522, 666, 668, 1202, 1406	__physicx_parse_range_single_-
__physicx_matrix_item_parse_-	check:n 55, 110, 120
aux:n 1089, 1094	__physicx_parse_range_single_-
__physicx_matrix_item_parse_-	nocheck:n 60, 118
aux:nn 1090, 1095	\g__physicx_physics_bool 10
__physicx_matrix_last_aux_c: ...	__physicx_qxmatrix_appto_-
..... 1525, 1547, 1556	body:nnn 831,
__physicx_matrix_last_aux_r: ...	835, 875, 881, 888, 914, 919, 925
..... 1531, 1548, 1565	
\l__physicx_matrix_last_col_tl ..	
..... 532, 659, 1522, 1536	
\l__physicx_matrix_last_row_tl ..	
..... 531, 658, 1528, 1535	
\l__physicx_matrix_main_tl 520, 575, 1158, 1399	
__physicx_matrix_row_iterate:n . 530, 656, 1271, 1374, 1384, 1486, 1497	
\l__physicx_matrix_rows_int 468, 469, 518, 567,	

<code>\l__physicx_row_list_seq</code>	524, 577, 1268, 1272
<code>\g__physicx_short_bool</code> ...	12, 20, 204
<code>\l__physicx_tmp_col_seq</code> ..	1352, 1355
<code>\l__physicx_tmp_coled_seq</code>	1354, 1359, 1369, 1381
<code>\l__physicx_tmp_colnum_seq</code>	1103, 1109, 1118
<code>\l__physicx_tmp_rownum_seq</code>	1101, 1107, 1116
<code>\l__physicx_tmp_tl</code>	269, 283, 295, 1441, 1447, 1450, 1456, 1462, 1468, 1471, 1477
<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>	1131, 1132, 1133, 1134
<code>\l__physicx_tmpb_seq</code>	85, 86, 96
<code>\l__physicx_tmpitem_tl</code>	1097, 1098, 1099, 1105, 1123
<code>\physicxempty</code> .	479, 493, 501, 538, 539, 638
<code>\physicxexcept</code>	541, 609, 618, 627, 633, 635
<code>\PHYSICXIGNORE</code>	7, 8, 487, 498, 1105
<code>\physicxset</code>	229
<code>\physicxtmp</code>	235, 821, 1190, 1288, 1394
<code>\poissonbracket</code>	420
<code>\pqty</code>	420
prg commands:	
<code>\prg_generate_conditional_</code> variant:Nnn	185
<code>\prg_new_conditional:Npnn</code>	13, 18, 23, 28, 146, 151, 167, 1162
<code>\prg_replicate:nn</code>	907, 937
<code>\prg_return_false:</code>	16, 21, 26, 34, 149, 154, 182, 1183
<code>\prg_return_true:</code>	16, 21, 26, 33, 149, 154, 182, 1165, 1182
<code>\ProcessKeysPackageOptions</code>	211
Q	
<code>\qty</code>	420
<code>\quantity</code>	347, 422
quark commands:	
<code>\quark_if_recursion_tail_stop:n</code> .	303, 304
<code>\q_recursion_stop</code>	256
<code>\q_recursion_tail</code>	256
<code>\qmatrix</code>	803, 1415
R	
<code>\rangle</code>	354, 381, 396
regex commands:	
<code>\regex_match:nnTF</code> ..	148, 153, 618, 627
<code>\RequirePackage</code>	206, 207, 208
<code>\rgroup</code>	378, 393
<code>\right</code>	316, 371, 378, 381, 387, 388, 393, 395, 396, 397, 414
<code>\rVert</code>	436
S	
<code>\sb</code>	529, 874, 879, 886, 913, 917, 923
scan commands:	
<code>\scan_stop:</code>	894, 1361
seq commands:	
<code>\c_empty_seq</code>	65, 1354
<code>\seq_clear:N</code>	82
<code>\seq_concat:NNN</code>	72
<code>\seq_count:N</code>	1013, 1014, 1030, 1031, 1039, 1040, 1048, 1049, 1060, 1061, 1071, 1072, 1136, 1143, 1330
<code>\seq_if_empty:N</code>	1268, 1274
<code>\seq_item:Nn</code>	1272, 1278
<code>\seq_map_indexed_inline:Nn</code>	1000, 1010, 1024, 1033, 1050, 1062, 1073, 1138, 1144, 1355
<code>\seq_map_inline:Nn</code>	1107, 1109, 1116, 1118
<code>\seq_map_tokens:Nn</code>	1334, 1342
<code>\seq_new:N</code>	524, 525
<code>\seq_pop_left:NN</code>	86, 96
<code>\seq_put_right:Nn</code>	116, 119, 130, 140, 1369, 1381
<code>\seq_set_eq:NN</code>	65, 1354
<code>\seq_set_split:Nnn</code>	5, 85, 577, 579, 989, 1134, 1140, 1327, 1352
<code>\seq_use:Nn</code>	1359
<code>\setmatrixdata</code>	781
<code>\setmatrixtype</code>	738, 754, 755, 756, 757, 758, 759, 760, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779
<code>\smallmatrixquantity</code>	347
<code>\smqty</code>	420
str commands:	
<code>\str_case_e:nnTF</code>	958
<code>\str_const:Nn</code>	510, 511, 512
<code>\str_if_eq:nnTF</code>	288, 290
<code>\symbf</code>	198
<code>\symbfit</code>	194
T	
T _E X and L ^A T _E X 2 _ε commands:	
<code>\@declareparencmd</code>	399
<code>\@declarequantitycmd</code>	235
<code>\@ifpackageloaded</code>	213, 216, 226

