

The `physicx` package

Wenjian Chern (Longaster*)

November 11, 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:nmm { nmm }
5 \cs_generate_variant:Nn \seq_set_split:Nnn { Non, NVV, c, cnV, cVV }
6 \cs_generate_variant:Nn \tl_replace_all:Nnn { Non, Nox }
7 \cs_new:Npn \PHYSICXIGNORE
8 { \exp_stop_f: \exp_not:N \PHYSICXIGNORE }
9 \bool_new:N \g__physicx_mathtools_bool
10 \bool_new:N \g__physicx_physics_bool
11 \bool_new:N \g__physicx_compat_bool
12 \bool_new:N \g__physicx_short_bool
13 \bool_new:N \g__physicx_quantity_bool
14 \prg_new_conditional:Npnn \physicx_compat: { T, F, TF }
15 {
16   \bool_if:NTF \g__physicx_compat_bool
17   { \prg_return_true: } { \prg_return_false: }
18 }
19 \prg_new_conditional:Npnn \physicx_short: { T, F, TF }
20 {
21   \bool_if:NTF \g__physicx_short_bool
22   { \prg_return_true: } { \prg_return_false: }
23 }
24 \prg_new_conditional:Npnn \physicx_mathtools: { T, F, TF }
25 {
26   \bool_if:NTF \g__physicx_mathtools_bool
27   { \prg_return_true: } { \prg_return_false: }
28 }
29 \prg_new_conditional:Npnn \physicx_option_or:nn #1#2 { T, F, TF }
30 {
31   \bool_lazy_or:nnTF
32   { \cs:w g__physicx_ #1 _bool \cs_end: }
33   { \cs:w g__physicx_ #2 _bool \cs_end: }
```

*Email: longaster@163.com

```

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

```

1.1 Utils functions

```

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

```

```

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

```

```

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

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

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

187 \tl_const:Nn \c_physicx_order_tl { \mathcal{o} }
188 \tl_const:Nn \c_physicx_Order_tl { \mathcal{O} }

```

```

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

```

`\physicsxset` physicsx setup command.

```

239 \NewDocumentCommand \physicsxset { s m }
240 {

```

```

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

```

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

1.2 Quantity things

```

\physicx_declare_legacy_quantity:nnNn
  \@declarequantitycmd
245 \tl_new:N \physicx_tmp
246 \tl_new:N \l__physicx_cmd_noauto_body_tl
247 \bool_new:N \l__physicx_cmd_noauto_body_bool
248 \tl_new:N \l__physicx_cmd_auto_body_tl
249 \bool_new:N \l__physicx_cmd_auto_body_bool
250 \tl_new:N \l__physicx_cmd_arg_spec_tl
251 \int_new:N \l__physicx_cmd_arg_int
252 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
253 {
254   \tl_clear:N \l__physicx_cmd_noauto_body_tl
255   \tl_clear:N \l__physicx_cmd_auto_body_tl
256   \tl_clear:N \l__physicx_cmd_arg_spec_tl
257   \int_set:Nn \l__physicx_cmd_arg_int {#1}
258   \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
259   \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
260 }
261 % noauto, auto, cmd, body
262 \cs_new:Npn \physicx_declare_legacy_quantity:nnNn #1#2#3#4
263 {
264   \__physicx_declare_init:nnn { 3 } {#1} {#2}
265   \__physicx_declare_legacy_quantity_aux:nw #4
266   \q_recursion_tail \q_recursion_tail \q_recursion_stop
267   \__physicx_declare_legacy_quantity_aux:NcVVV
268   #3 { \cs_to_str:N #3 ~ body }
269   \l__physicx_cmd_arg_spec_tl
270   \l__physicx_cmd_noauto_body_tl
271   \l__physicx_cmd_auto_body_tl
272 }
273 % arg spec, pre, body to replace(start from #4), post
274 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
275 {
276   \int_incr:N \l__physicx_cmd_arg_int
277   \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
278     \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
279     \tl_set:Nx \l__physicx_tmp_tl
280     {
281       {
282         \exp_not:N \tl_if_novalue_p:n
283         {
284           \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:
285             \or: \or: \or:
286             \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
287             \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
288             \fi:

```

```

289     }
290   }
291 }
292 \if_bool:N \l__physicx_cmd_noauto_body_bool
293   \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
294   \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
295     {
296       {
297         % if is '.', use none
298         \str_if_eq:nnTF {#2} {.} {} {#2}
299         #3
300         \str_if_eq:nnTF {#4} {.} {} {#4}
301       }
302     }
303 \fi:
304 \if_bool:N \l__physicx_cmd_auto_body_bool
305   \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
306   \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
307     { { ##1 #2 #3 ##2 #4 } }
308 \fi:
309 \fi:
310 }
311 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
312 {
313   \quark_if_recursion_tail_stop:n {#1}
314   \quark_if_recursion_tail_stop:n {#2}
315   \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2
316   \__physicx_declare_legacy_quantity_aux:nw
317 }
318 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
319 {
320   \__physicx_nauto_case:nnnn
321   { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
322   {
323     \cs_set_protected:Npn #1
324     {
325       \peek_charcode_ignore_spaces:NTF \let
326       { #2 } { #2 [ \physicx_left: ] \physicx_right: }
327     }
328     \DeclareDocumentCommand #2 { 0{##2} m s #3 }
329     {
330       \IfBooleanTF { ##3 }
331       { \bool_case_false:n {#4} }
332       { \bool_case_false:n {#5} }
333     }
334   }
335   {
336     \cs_set_protected:Npn #1
337     { #2 \c_empty_tl \c_empty_tl }
338     \DeclareDocumentCommand #2 { m m s #3 }
339     { \bool_case_false:n {#4} }
340   }
341 }
342 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }

```

```

343 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
344 {
345   \bool_if:NTF \l__physicx_cmd_noauto_body_bool
346   {
347     \bool_if:NTF \l__physicx_cmd_auto_body_bool
348     {#1} {#2}
349   }
350   {
351     \bool_if:NTF \l__physicx_cmd_auto_body_bool
352     {#3} {#4}
353   }
354 }
355 \cs_set_protected:Npn \@declarequantitycmd
356 { \physicx_declare_legacy_quantity:nnNn }

```

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

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


```

392 { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } { \right\rangle } }
393 { !d== } { { \begin{Vmatrix} } {#9} { \end{Vmatrix} } } }
394 }
395 \physicsx_declare_legacy_quantity:nnNn
396 \c_true_bool \c_false_bool \smallmatrixquantity
397 {
398 { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } { \right\} } }
399 { !o } { { \left[ } { \begin{smallmatrix} #5 \end{smallmatrix} } { \right]} } }
400 { !d() }
401 {
402 { \IfBooleanTF{#3}{\left\lgroup}{\left(} }
403 { \begin{smallmatrix} #6 \end{smallmatrix} } }
404 { \IfBooleanTF{#3}{\right\rgroup}{\right)} } }
405 }
406 { !d|| } { { \left\vert } { \begin{smallmatrix} #7 \end{smallmatrix} } { \right\vert} } }
407 { !d<> } { { \left\langle } { \begin{smallmatrix} #8 \end{smallmatrix} } { \right\rangle } } }
408 { !d== } { { \left\Vert } { \begin{smallmatrix} #9 \end{smallmatrix} } { \right\Vert} } }
409 }
410 \fi:

```

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

```

\physicsx_declare_legacy_paren:NnnnNnN
  \@declareparenccmd
2411 %% cmd, arg spec, replace(start from #6), pre, left, right, post
2412 \cs_new:Npn \physicsx_declare_legacy_paren:NnnnNnN #1#2#3#4#5#6#7
2413 {
2414   \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }
2415   {
2416     \bool_case_true:nF
2417     {
2418       { \bool_if_p:n {##2} } { #4 \physicsx_left:N \bigl #5 #3 \physicsx_right:N \bigr
2419       { \bool_if_p:n {##3} } { #4 \physicsx_left:N \Bigl #5 #3 \physicsx_right:N \Bigr
2420       { \bool_if_p:n {##4} } { #4 \physicsx_left:N \biggl #5 #3 \physicsx_right:N \biggr
2421       { \bool_if_p:n {##5} } { #4 \physicsx_left:N \Biggl #5 #3 \physicsx_right:N \Biggr
2422     }
2423     {
2424       \IfBooleanTF {##1}
2425       { #4 #5 #3 #6 #7 }
2426       { #4 \physicsx_left: #5 #3 \physicsx_right: #6 #7 }
2427     }
2428   }
2429 }
2430 \cs_set_protected:Npn \@declareparenccmd
2431 { \physicsx_declare_legacy_paren:NnnnNnN }

```

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

```

\qty Redefine some macros in physics package.
\mqty 2432 \if_bool:N \g__physicsx_quantity_bool
\smqty 2433 \physicsx_option_or:nnT { compat } { short }
\pqty 2434 {
\bqty 2435 \cs_set:Npn \qty { \quantity }
\vqty 2436 \physicsx_declare_legacy_paren:NnnnNnN \pqty { m } {#6} { } { } { }
\Bqty 2437 \physicsx_declare_legacy_paren:NnnnNnN \bqty { m } {#6} { } { } { }

```

```

\absolutevalue
\eval
\abs
\norm
\order
\Order
\oorder
\commutator
\poissonbracket
\ph

```

```

438 \physicx_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } \vert \vert { }
439 \physicx_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } \{ \} { }
440 }
441 \physicx_declare_legacy_paren:NnnnNNn \absolutevalue
442 { m } {#6} { } \vert \vert { }
443 \physicx_option_or:nnT { compat } { short }
444 {
445 \cs_set:Npn \eval { \evaluated }
446 \cs_set:Npn \abs { \absolutevalue }
447 }
448 \physicx_declare_legacy_paren:NnnnNNn \norm
449 { m } {#6} { } \lVert \rVert { }
450 \physicx_compat:TF
451 {
452 \physicx_declare_legacy_paren:NnnnNNn \order
453 { m } {#6} { \c_physicx_Order_tl } ( ) { }
454 \physicx_declare_legacy_paren:NnnnNNn \oorder
455 { m } {#6} { \c_physicx_order_tl } ( ) { }
456 }
457 {
458 \physicx_declare_legacy_paren:NnnnNNn \Order
459 { m } {#6} { \c_physicx_Order_tl } ( ) { }
460 \physicx_declare_legacy_paren:NnnnNNn \order
461 { m } {#6} { \c_physicx_order_tl } ( ) { }
462 }
463 \physicx_declare_legacy_paren:NnnnNNn \commutator
464 { m m } { #6 , #7 } { } [ ] { }
465 \physicx_option_or:nnT { compat } { short }
466 { \cs_set:Npn \comm { \commutator } }
467 \physicx_declare_legacy_paren:NnnnNNn \poissonbracket
468 { m m } { #6 , #7 } { } \{ \} { }
469 \physicx_option_or:nnT { compat } { short }
470 {
471 \cs_set:Npn \pb { \poissonbracket }
472 \cs_set:Npn \anticommutator { \poissonbracket }
473 \cs_set:Npn \acomm { \poissonbracket }
474 }
475 \fi:
476 \physicx_declare_legacy_paren:NnnnNNn \OOrder
477 { m } {#6} { \c_physicx_Order_tl } ( ) { }
478 \physicx_declare_legacy_paren:NnnnNNn \oorder
479 { m } {#6} { \c_physicx_oorder_tl } ( ) { }

```

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

1.3 Matrix things

1.3.1 Matrix auxillary functions

```

480 \cs_new_nopar:Npn \__physicx_matrix_calc:nn #1#2
481 {
482 \int_set:Nn \l__physicx_matrix_rows_int
483 { \int_max:nn {#1} \l__physicx_matrix_rows_int }
484 \int_set:Nn \l__physicx_matrix_cols_int
485 { \int_max:nn {#2} \l__physicx_matrix_cols_int }

```

```

486 }
487 % use matrix element
488 \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
489 {
490   \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
491     \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
492   \else:
493     \exp_not:o { \physicxempty }
494   \fi:
495 }
496 % set matrix element, check or not
497 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
498 { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
499 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
500 {
501   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
502   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
503 }
504 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_cke:nnn #1#2#3
505 {
506   \tl_if_empty:nTF {#3}
507   { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
508   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
509 }
510 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
511 {
512   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
513   {
514     \tl_if_empty:nTF {#3}
515     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
516     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
517   }
518 }
519 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
520 \__physicx_matrix_set_r_c_ckigep:nnn
521 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
522 \__physicx_matrix_set_r_c_nock:nnn
523 % align, cr, sep symbol
524 \str_const:Nn \physicx@align { , }
525 \str_const:Nn \physicx@cr { ; }
526 \str_const:Nn \physicx@sep { , }
527 \bool_new:N \l__physicx_matrix_infinite_bool
528 \bool_new:N \l__physicx_matrix_dotrow_bool
529 \bool_new:N \l__physicx_matrix_dotcol_bool
530 \tl_new:N \l__physicx_matrix_array_tl
531 \tl_new:N \l__physicx_matrix_body_tl
532 \int_new:N \l__physicx_matrix_rows_int
533 \int_new:N \l__physicx_matrix_cols_int
534 \tl_new:N \l__physicx_matrix_main_tl
535 \clist_new:N \l__physicx_matrix_diag_clist
536 \clist_new:N \l__physicx_matrix_item_clist
537 \bool_new:N \l__physicx_matrix_diag_bool
538 \seq_new:N \l__physicx_row_list_seq
539 \seq_new:N \l__physicx_col_list_seq

```

```

540 % expand input
541 \cs_new_eq:NN \__physicx_expand:w \exp_not:o
542 %% main, row iterate, col iterate
543 \cs_new_nopar:Npn \physicx@matricxelement #1#2#3 { #1 \sb { #2 #3 } }
544 \cs_new_nopar:Npn \__physicx_matrix_row_iterate:n #1 { #1 }
545 \tl_new:N \l__physicx_matrix_last_row_tl
546 \tl_new:N \l__physicx_matrix_last_col_tl
547 \cs_new_nopar:Npn \__physicx_matrix_col_iterate:n #1 { #1 }
548 \cs_new_nopar:Npn \__physicx_matrix_begin:w { }
549 \cs_new_nopar:Npn \__physicx_matrix_end:w { }
550 \cs_new_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn
551 \bool_new:N \l__physicx_matrix_expand_element_bool
552 % when element is empty use \physicxempty
553 \tl_new:N \physicxempty
554 % save 'element-except' key's value
555 \tl_new:N \physicxexcept
556 \tl_new:N \l__physicx_matrix_args_tl
557 \tl_new:N \l__physicx_matrix_after_begin_tl
558 \tl_new:N \l__physicx_matrix_after_end_tl
559 \bool_new:N \l__physicx_matrix_transpose_bool
560 \bool_new:N \l__physicx_matrix_enhanced_bool
561 \dim_new:N \l__physicx_matrix_sep_dim
562 \cs_new:Npn \__physicx_adi:nnn #1#2#3 { #1#2#3 }
563 \tl_new:N \l__physicx_matrix_beginning_tl
564 \tl_new:N \l__physicx_matrix_ending_tl

```

1.3.2 Matrix keys

```

565 \keys_define:nn { physicx }
566 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
567 \keys_define:nn { physicx/matrix }
568 {
569   array .tl_set:N = \l__physicx_matrix_array_tl ,
570   expand .choice: ,
571   expand / none .code:n =
572     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
573   expand / text-expand .code:n =
574     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
575   expand / f .code:n =
576     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
577   expand / romanual .meta:n = { expand = f } ,
578   expand / x .code:n =
579     \cs_set_eq:NN \__physicx_expand:w \use:n ,
580   expand / edef .meta:n = { expand = x } ,
581   rows .int_set:N = \l__physicx_matrix_rows_int ,
582   cols .int_set:N = \l__physicx_matrix_cols_int ,
583   auto-update .choice: ,
584   auto-update / true .code:n =
585     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,
586   auto-update / false .code:n =
587     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
588   auto-update .default:n = true ,
589   main .tl_set:N = \l__physicx_matrix_main_tl ,
590   row-list .code:n =
591     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,

```

```

592 col-list .code:n =
593   \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
594   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
595   infinite .default:n = true ,
596   !infinite .code:n =
597     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
598   element-code .cs_set:Np = \physicx@matricielement #1#2#3 ,
599   element-code* .choice: ,
600   element-code* / except-empty .code:n =
601     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
602     \physicx@matricielement
603     \cs_set:Npn \physicx@matricielement ##1##2##3
604     {
605       \tl_if_empty:nTF {##1}
606       {##1}
607       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
608     } ,
609   element-code* / except-dots .code:n =
610     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
611     \physicx@matricielement
612     \cs_set:Npn \physicx@matricielement ##1##2##3
613     {
614       \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
615       {##1}
616       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
617     } ,
618   element-code* / except-tl .code:n =
619     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
620     \physicx@matricielement
621     \cs_set:Npn \physicx@matricielement ##1##2##3
622     {
623       \tl_if_in:onTF { \physicxexcept } {##1}
624       {##1}
625       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
626     } ,
627   element-code* / except-regex .code:n =
628     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
629     \physicx@matricielement
630     \cs_set:Npn \physicx@matricielement ##1##2##3
631     {
632       \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
633       {##1}
634       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
635     } ,
636   element-code* / only-regex .code:n =
637     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
638     \physicx@matricielement
639     \cs_set:Npn \physicx@matricielement ##1##2##3
640     {
641       \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
642       { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
643       {##1}
644     } ,
645   element-code* / unknown .code:n =

```

```

646     \cs_set:Npx \physicx@matricelement { \exp_not:c {#1} },
647 element-except .tl_set:N = \physicxexcept ,
648 element-except+ .code:n =
649     \tl_put_right:Nn \physicxexcept {#1} ,
650 expand-element .bool_set:N = \l__physicx_matrix_expand_element_bool ,
651 expand-element .default:n = true ,
652 empty .tl_set:N = \physicxempty ,
653 check .choice: ,
654 check / none .code:n =
655     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
656     \__physicx_matrix_set_r_c_nock:nnn ,
657 check / empty .code:n =
658     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
659     \__physicx_matrix_set_r_c_ckepp:nnn ,
660 check / ignore .code:n =
661     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
662     \__physicx_matrix_set_r_c_ckig:nnn ,
663 check / igep .code:n =
664     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
665     \__physicx_matrix_set_r_c_ckigep:nnn ,
666 check / all .code:n =
667     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
668     \__physicx_matrix_set_r_c_ckall:nnn ,
669 check .default:n = all ,
670 row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
671 col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
672 last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
673 last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
674 diag .clist_set:N = \l__physicx_matrix_diag_clist ,
675 diag+ .code:n =
676     \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
677 diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
678 diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
679 diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
680 item .clist_set:N = \l__physicx_matrix_item_clist ,
681 item+ .code:n =
682     \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
683 item-now .code:n = \physicx_matrix_item_parse:n {#1} ,
684 item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
685 item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
686 check-range .choice: ,
687 check-range / true .code:n = \physicx_parse_range_check: ,
688 check-range / false .code:n = \physicx_parse_range_noccheck: ,
689 check-range .default:n = true ,
690 begin .tl_set:N = \__physicx_matrix_begin:w ,
691 end .tl_set:N = \__physicx_matrix_end: ,
692 args .code:n =
693     \tl_set:Nn \l__physicx_matrix_args_tl { [ #1 ] } ,
694 args* .tl_set:N = \l__physicx_matrix_args_tl ,
695 after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
696 after-begin+ .code:n =
697     { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
698 after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
699 after-end+ .code:n =

```

```

700     { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
701     sepdim .dim_set:N = \l__physicx_matrix_sepdim ,
702     type .multichoice: ,
703     saveto .tl_set:N = \l__physicx_matrix_save_tl ,
704     saveto* .code:n =
705       \tl_set:Nn \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
706     transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
707     transpose .default:n = true ,
708     ' .meta:n = { transpose = true } ,
709     T .meta:n = { transpose = true } ,
710     MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
711     enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
712     enhanced .default:n = true ,
713     !enhanced .code:n =
714       \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
715     cr .tl_set:N = \physicx@cr ,
716     align .tl_set:N = \physicx@align ,
717     sep .tl_set:N = \physicx@sep ,
718     adi-order .choice: ,
719     adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
720     adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,
721     adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
722     adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
723     adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
724     adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
725     beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
726     beginning+ .code:n =
727       \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
728     ending .tl_set:N = \l__physicx_matrix_ending_tl ,
729     ending+ .code:n =
730       \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
731
732     unknown .code:n =
733       \physicx_search_also:nnF
734       {
735         physicx/matrix/type ,
736         physicx/matrix/expand ,
737         physicx/matrix/element-code* ,
738       }
739       {#1}
740       {
741         \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
742         {
743           \keys_set:nx { physicx/matrix }
744           { MaxMatrixCols = \l_keys_key_str }
745         }
746         {
747           \msg_error:nnxx { physicx } { unknown-key }
748             \l_keys_path_str { physicx }
749         }
750       } ,
751   }

```

```

\physicx_matrix_new_type:nnn
\physicx_matrix_new_type:nn
  \setmatrixtype

```

```

752 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
753 {
754   \keys_define:nn { physicx/matrix }
755   { type / #1 .meta:n = { begin={#2} , end={#3} } }
756 }
757 \cs_new:Npn \physicx_matrix_new_type:nn #1#2
758 {
759   \keys_define:nn { physicx/matrix }
760   { type / #1 .meta:n = {#2} }
761 }
762 \NewDocumentCommand \setmatrixtype { s >{ \TrimSpaces } m }
763 {
764   \IfBooleanTF {#1}
765   { \physicx_matrix_new_type:nn {#2} }
766   { \physicx_matrix_new_type:nnn {#2} }
767 }

```

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

A few types.

```

768 \setmatrixtype {m} {\begin{matrix}} {\end{matrix}}
769 \setmatrixtype {p} {\begin{pmatrix}} {\end{pmatrix}}
770 \setmatrixtype {b} {\begin{bmatrix}} {\end{bmatrix}}
771 \setmatrixtype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
772 \setmatrixtype {v} {\begin{vmatrix}} {\end{vmatrix}}
773 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
774 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
775 \physicx_mathtools:T
776 {
777   \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
778   \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
779   \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
780   \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
781   \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
782   \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
783   \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
784   \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
785   \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
786   \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
787   \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
788   \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
789   \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
790   \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
791   \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
792   \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
793   \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
794 }

```

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

```

795 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
796 { \clist_set:cn { physicx@ #1 data@ #2 } }
797 \cs_new_protected_nopar:Npn \__physicx_matrix_set_data:nn #1#2
798 {
799   \clist_clear:c { l__physicx_matrix_ #1 _clist }

```



```

800   \__physicx_matrix_add_data:nn {#1} {#2}
801   }
802   \cs_new_protected_nopar:Npn \__physicx_matrix_add_data:nn #1#2
803   {
804     \clist_map_inline:nn {#2}
805     {
806       \clist_concat:ccc
807       { l__physicx_matrix_ #1 _clist }
808       { l__physicx_matrix_ #1 _clist }
809       { physicx@ #1 data@ #2 }
810     }
811   }

```

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

Initial settings.

```

812   \keys_set:nn { physicx/matrix }
813   {
814     type = m ,
815     saveto = ? ,
816   }

```

\qxmatri

```

817   %% basicly, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
      xmatnm-in-the-physics-package, but changed some
818   % #1 = boolean, saveto matrix
819   % #2 = star, infinite
820   % #3 = options
821   % #4 = letter for the entries
822   % #5 = number of rows
823   % #6 = number of explicit rows, default = 3
824   % #7 = number of columns
825   % #8 = number of explicit columns, default = 3
826   \DeclareDocumentCommand \qxmatri { t= s 0{type=p} m m 0{3} m 0{3} }
827   {
828     \group_begin:
829     \IfBooleanTF { #2 }
830     { \bool_set_true:N \l__physicx_matrix_infinite_bool }
831     { \bool_set_false:N \l__physicx_matrix_infinite_bool }
832     \int_set:Nn \l__physicx_matrix_rows_int {#6}
833     \int_set:Nn \l__physicx_matrix_cols_int {#8}
834     \IfBooleanTF {#1}
835     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicx_tmp } }
836     { \keys_set:nn { physicx/matrix } {#3} }
837     \physicx_qxmatri:nnn {#4} {#5} {#7}
838     \__physicx_matrix_save_or_print:
839     \group_end:
840   }
841   \cs_new_protected:Nn \physicx_qxmatri:nnn
842   {
843     \bool_if:NTF \l__physicx_matrix_expand_element_bool
844     {
845       \cs_set_eq:NN \__physicx_qxmatri_appto_body:nnn
846       \__physicx_matrix_appto_body_e:nnn
847     }

```

```

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

```

```

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

```

```

956         % update cols
957         \bool_if:NTF \l__physicx_matrix_dotcol_bool
958         { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
959         { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
960     }
961 }

```

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

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
962 \cs_new:Npn \physicx_matrix_diag_parse:n #1
963 {
964     \keyval_parse:nnn
965     \__physicx_matrix_diag_parse_aux:n
966     \__physicx_matrix_diag_parse_aux:nn
967     {#1}
968 }
969 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
970 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
971 {
972     \str_case:e:nnF {#1}
973     {
974         { auto-update }
975         {
976             \cs_set_eq:NN \__physicx_matrix_diag_calc:nn
977             \__physicx_matrix_calc:nn
978         }
979         { noauto-update }
980         {
981             \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
982         }
983         { true }
984         {
985             \bool_set_true:N \l__physicx_matrix_diag_bool
986             \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
987             \__physicx_diagonalmatrix_set_diag:
988         }
989         { false }
990         {
991             \bool_set_false:N \l__physicx_matrix_diag_bool
992             \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
993             \__physicx_diagonalmatrix_no_diag:
994         }
995     }
996     { \msg_error:nnn { physicx } { diag-key } {#1} }
997 }
998 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
999 {
1000     \tl_set:Nn \l__physicx_tmpdiag_tl {#2}
1001     \tl_set:Nx \l__physicx_tmpdiag_tl
1002     { \__physicx_expand:w \l__physicx_tmpdiag_tl }
1003     \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl
1004     \tl_if_head_eq_charcode:nNTF {#1} '
1005     {

```

```

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

```

```

1060 \if_int_compare:w #1 = 0 \exp_stop_f:
1061 \__physicx_matrix_diag_calc:nn
1062 { \seq_count:N \l__physicx_tmpdiag_seq }
1063 { \seq_count:N \l__physicx_tmpdiag_seq }
1064 \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1065 {
1066 \physicx_matrix_set_r_c:nnn
1067 {##1}
1068 { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1069 {##2}
1070 }
1071 \else:
1072 \if_int_compare:w #1 > 0 \exp_stop_f:
1073 \__physicx_matrix_diag_calc:nn
1074 { \seq_count:N \l__physicx_tmpdiag_seq }
1075 { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1076 \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1077 {
1078 \physicx_matrix_set_r_c:nnn
1079 {##1}
1080 { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1081 {##2}
1082 }
1083 \else:
1084 \__physicx_matrix_diag_calc:nn
1085 { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1086 { \seq_count:N \l__physicx_tmpdiag_seq }
1087 \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1088 {
1089 \physicx_matrix_set_r_c:nnn
1090 { \int_eval:n { ##1 - #1 } }
1091 { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1092 {##2}
1093 }
1094 \fi:
1095 \fi:
1096 }
1097 \cs_new:Npn \__physicx_matrix_diag_calc:nn
1098 { \__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
1099 \cs_new:Npn \physicx_matrix_item_parse:n #1
1100 {
1101 \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1102 \keyval_parse:NNn
1103 \__physicx_matrix_item_parse_aux:n
1104 \__physicx_matrix_item_parse_aux:nn
1105 {#1}
1106 }
1107 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }
1108 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1109 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2

```

```

1110 {
1111   \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1112   \tl_set:Nx \l__physicx_tmpitem_tl
1113     { \__physicx_expand:w \l__physicx_tmpitem_tl }
1114   \physicx_parse_range:nxN \l__physicx_matrix_rows_int
1115     { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1116   \physicx_parse_range:nxN \l__physicx_matrix_cols_int
1117     { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1118   \exp_args:No \tl_if_eq:nnTF
1119     { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1120   {
1121     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1122       {
1123         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1124           {
1125             \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1126           }
1127       }
1128   }
1129   {
1130     \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1131       {
1132         \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1133           {
1134             \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }
1135             {
1136               \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1137                 {##1} {####1} { \l__physicx_tmpitem_tl }
1138             }
1139           }
1140       }
1141   }
1142 }

```

(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
1143 \cs_new:Npn \physicx_matrix_array_parse:n #1
1144 {
1145   \tl_set:Nn \l__physicx_tmparr_tl {#1}
1146   \tl_set:Nx \l__physicx_tmparr_tl
1147     { \__physicx_expand:w \l__physicx_tmparr_tl }
1148   \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl
1149   \__physicx_matrix_autocalc:nn
1150     { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1151     { 0 }
1152   \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1153     {
1154       \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1155       \__physicx_matrix_autocalc:nn
1156         { 0 }
1157         { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1158       \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1159         {

```

```

1160         \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1161     }
1162 }
1163 }
1164 \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.
1165 \cs_new:Npn \physicx_matrix_array_parse_main:
1166 {
1167     \int_step_inline:nn \l__physicx_matrix_rows_int
1168     {
1169         \int_step_inline:nn \l__physicx_matrix_cols_int
1170         {
1171             \exp_args:Nno \physicx_matrix_set_r_c:nnn
1172             {##1} {####1} \l__physicx_matrix_main_tl
1173         }
1174     }
1175 }

(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).
1176 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }
1177 {
1178     \physicx_if_num:nTF {#1}
1179     { \prg_return_true: }
1180     {
1181         \bool_case_true:nTF
1182         {
1183             { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1184             { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1185             {
1186                 \bool_lazy_and_p:nn
1187                 { \cs_if_exist_p:N \inteval }
1188                 { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1189             } { }
1190             {
1191                 \bool_lazy_and_p:nn
1192                 { \cs_if_exist_p:N \fpeval }
1193                 { \tl_if_head_eq_meaning_p:nN {#1} \fpeval }
1194             } { }
1195         }
1196         { \prg_return_true: }
1197         { \prg_return_false: }
1198     }
1199 }

(End definition for \__physicx_if_can_num:n.)

```

```

\diagonalmatrix Define \diagonalmatrix.
1200 \DeclareDocumentCommand \diagonalmatrix { t+ 0{} m }
1201 {

```



```

1202 \group_begin:
1203 \IfBooleanTF {#1}
1204 { \keys_set:nn { physix/matrix } { #3 , saveto = \physixtmp } }
1205 { \keys_set:nn { physix/matrix } { #3 } }
1206 \physix_construct:nnn { }
1207 {
1208   \physix_matrix_diag_parse:o \l__physix_matrix_diag_clist
1209   \tl_if_empty:nF {#4}
1210   {
1211     \__physix_if_keyval:nTF {#4}
1212     { \physix_matrix_diag_parse:n { true, #4 } }
1213     { \physix_matrix_diag_parse:n { true, 0 = {#4} } }
1214   }
1215 }
1216 { \physix_matrix_item_parse:o \l__physix_matrix_item_clist }
1217 \bool_lazy_or:nnTF
1218 { \bool_if_p:n {#2} }
1219 { \bool_if_p:N \l__physix_matrix_enhanced_bool }
1220 {
1221   \bool_if:NTF \l__physix_matrix_expand_element_bool
1222   {
1223     \cs_set_eq:NN \__physix_diagonalmatrix_enhanced:nnn
1224     \__physix_matrix_appto_body_e:off
1225   }
1226   {
1227     \cs_set_eq:NN \__physix_diagonalmatrix_enhanced:nnn
1228     \__physix_matrix_appto_body_ne:off
1229   }
1230   \use_i_ii:nnn
1231 }
1232 { \use_i:nn }
1233 \__physix_matrix_transpose:N
1234 \__physix_diagonalmatrix_generate_enhanced_body:NNN
1235 \__physix_diagonalmatrix_generate_body:NNN
1236 \__physix_matrix_save_or_print:
1237 \group_end:
1238 }
1239 \cs_new:Npn \__physix_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1240 {
1241   \__physix_matrix_generate_body:NNNN #1#2#3
1242   \__physix_diagonalmatrix_enhanced:nnn
1243 }
1244 \cs_new:Npn \__physix_diagonalmatrix_generate_body:NNN #1#2#3
1245 {
1246   \int_step_inline:nn { #1 - 1 }
1247   {
1248     \int_step_inline:nn { #2 - 1 }
1249     {
1250       \tl_put_right:Nx \l__physix_matrix_body_tl
1251       {
1252         \exp_after:wN
1253         \physix_matrix_use_r_c:nn
1254         #3 {{##1}} {{####1}} &
1255       }
1256     }
1257   }
1258 }

```

```

1256     }
1257     \tl_put_right:Nx \l__physicx_matrix_body_tl
1258     {
1259         \exp_after:wN
1260         \physicx_matrix_use_r_c:nn
1261         #3 {{##1}} {{ \int_use:N #2 }} \[\dim_use:N \l__physicx_matrix_sep_dim]
1262     }
1263 }
1264 \int_step_inline:nn { #2 - 1 }
1265 {
1266     \tl_put_right:Nx \l__physicx_matrix_body_tl
1267     {
1268         \exp_after:wN
1269         \physicx_matrix_use_r_c:nn
1270         #3 {{ \int_use:N #1 }} {{##1}} &
1271     }
1272 }
1273 \tl_put_right:Nx \l__physicx_matrix_body_tl
1274 {
1275     \exp_after:wN
1276     \physicx_matrix_use_r_c:nn
1277     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1278 }
1279 }

```

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

__physicx_declare_init:

```

1280 \cs_new:Npn \__physicx_matrix_enhanced_init:
1281 {
1282     \seq_if_empty:NF \l__physicx_row_list_seq
1283     {
1284         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1285         \cs_set_nopar:Npn \__physicx_matrix_row_iterate:n ##1
1286             { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1287     }
1288     \seq_if_empty:NF \l__physicx_col_list_seq
1289     {
1290         \bool_set_true:N \l__physicx_matrix_expand_element_bool
1291         \cs_set_nopar:Npn \__physicx_matrix_col_iterate:n ##1
1292             { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1293     }
1294 }

```

(End definition for __physicx_declare_init:.)

\commamatrix Define \commamatrix.

```

1295 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1296 {
1297     \group_begin:
1298     \keys_set:nn { physicx/matrix } {#3}
1299     \tl_if_empty:nF {#4}
1300     { \keys_set:nn { physicx/matrix } { array = {#4} } }
1301     \IfBooleanT {#1}
1302     { \keys_set:nn { physicx/matrix } { saveto = \physicx_tmp } }

```

```

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

```

```

1357         {
1358             \int_incr:N \l__physicx_tmpa_int
1359             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1360             \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_ne:nnn
1361         }
1362     }
1363 }
1364 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1365 {
1366     \seq_set_split:Non \l__physicx_tmp_col_seq
1367     { \physicx@align } {#3}
1368     \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1369     \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1370     { #2 {##2} {#1} {##1} }
1371     \tl_put_right:Nx \l__physicx_matrix_body_tl
1372     {
1373         \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1374         \if_int_compare:w \l__physicx_matrix_rows_int = #1
1375             \scan_stop:
1376         \else:
1377             \[\dim_use:N \l__physicx_matrix_sep_dim]
1378         \fi:
1379     }
1380 }
1381 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1382 {
1383     \seq_put_right:Nx \l__physicx_tmp_coled_seq
1384     {
1385         \text_expand:n % \text_expand:n do the magic thing, but slower
1386         {
1387             \physicx@matricelement { #1 }
1388             { \__physicx_matrix_row_iterate:n {#2} }
1389             { \__physicx_matrix_col_iterate:n {#3} }
1390         }
1391     }
1392 }
1393 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1394 {
1395     \seq_put_right:No \l__physicx_tmp_coled_seq
1396     {
1397         \physicx@matricelement {#1}
1398         { \__physicx_matrix_row_iterate:n {#2} }
1399         { \__physicx_matrix_col_iterate:n {#3} }
1400     }
1401 }

```

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

\generalmatrix Define \generalmatrix.

```

1402 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1403 {
1404     \IfBooleanTF {#2}
1405     {
1406         \group_begin:

```

```

1407 \IfBooleanTF {#1}
1408 { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicxtmp } }
1409 { \keys_set:nn { physicx/matrix } {#4} }
1410 \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1411 \physicx_construct:nnn
1412 {
1413   \tl_if_empty:NTF \l__physicx_matrix_main_tl
1414   {
1415     \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1416   }
1417   { \physicx_matrix_array_parse_main: }
1418 }
1419 { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1420 { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1421 \__physicx_generalmatrix:
1422 \__physicx_matrix_save_or_print:
1423 \group_end:
1424 }
1425 {
1426 \IfBooleanTF {#1}
1427 { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1428 { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1429 \qxmatrix = * [#4]
1430 }
1431 }
1432 \cs_new:Npn \__physicx_generalmatrix:
1433 {
1434   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1435   {
1436     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1437     \__physicx_matrix_appto_body_e:off
1438   }
1439   {
1440     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1441     \__physicx_matrix_appto_body_ne:off
1442   }
1443   \__physicx_matrix_transpose:N
1444   \__physicx_matrix_generate_body:NNNN
1445   \__physicx_generalmatrix_generate:nnn
1446 }

```

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

__physicx_matrix_generate_body:NNNN

```

1447 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1448 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1449 {
1450   \__physicx_matrix_enhanced_init:
1451   \int_step_inline:nn { #1 - 1 }
1452   {
1453     \int_step_inline:nn { #2 - 1 }
1454     {
1455       \tl_set:Nx \l__physicx_tmp_tl
1456       {

```

```

1457         \exp_after:wN
1458         \physicx_matrix_use_r_c:nn
1459         #3 {{##1}} {{####1}}
1460     }
1461     #4 \l__physicx_tmp_tl {##1} {####1}
1462     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1463 }
1464 \tl_set:Nx \l__physicx_tmp_tl
1465 {
1466     \exp_after:wN
1467     \physicx_matrix_use_r_c:nn
1468     #3 {{##1}} {{ \int_use:N #2 }}
1469 }
1470 #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1471 \tl_put_right:Nx \l__physicx_matrix_body_tl
1472 { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1473 }
1474 \int_step_inline:nn { #2 - 1 }
1475 {
1476     \tl_set:Nx \l__physicx_tmp_tl
1477     {
1478         \exp_after:wN
1479         \physicx_matrix_use_r_c:nn
1480         #3 {{ \int_use:N #1 }} {{##1}}
1481     }
1482     #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}
1483     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1484 }
1485 \tl_set:Nx \l__physicx_tmp_tl
1486 {
1487     \exp_after:wN
1488     \physicx_matrix_use_r_c:nn
1489     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1490 }
1491 #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1492 }

```

(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
1493 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1494 {
1495     \tl_put_right:Nx \l__physicx_matrix_body_tl
1496     {
1497         \text_expand:n
1498         {
1499             \physicx@matricelement {#1}
1500             { \__physicx_matrix_row_iterate:n {#2} }
1501             { \__physicx_matrix_col_iterate:n {#3} }
1502         }
1503     }
1504 }
1505 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1506 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3

```

```

1507 {
1508   \tl_put_right:No \l__physicx_matrix_body_tl
1509   {
1510     \physicx@matricelement {#1}
1511     { \__physicx_matrix_row_iterate:n {#2} }
1512     { \__physicx_matrix_col_iterate:n {#3} }
1513   }
1514 }
1515 \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

```

1516 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1517 {
1518   \bool_if:NTF \l__physicx_matrix_transpose_bool
1519   {
1520     #1
1521     \l__physicx_matrix_cols_int
1522     \l__physicx_matrix_rows_int
1523     \use_ii_i:nn
1524   }
1525   {
1526     #1
1527     \l__physicx_matrix_rows_int
1528     \l__physicx_matrix_cols_int
1529     \use:nn
1530   }
1531 }

```

(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.

```

1532 \cs_new:Npn \physicx_construct:nnn #1#2#3
1533 {
1534   \l__physicx_matrix_beginning_tl
1535   \__physicx_adi:nnn {#1} {#2} {#3}
1536   \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1537   {
1538     \int_incr:N \l__physicx_matrix_cols_int
1539     \__physicx_matrix_last_aux_c:
1540     \int_incr:N \l__physicx_matrix_cols_int
1541   }
1542   \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1543   {
1544     \int_incr:N \l__physicx_matrix_rows_int
1545     \__physicx_matrix_last_aux_r:
1546     \int_incr:N \l__physicx_matrix_rows_int
1547   }
1548   \bool_lazy_or:nnF
1549   { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1550   { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1551   {

```

```

1552     \physicx_matrix_set_r_c:nnn
1553     { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1554     { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1555     { \ddots }
1556   }
1557   \bool_if:NT \l__physicx_matrix_infinite_bool
1558   {
1559     \int_incr:N \l__physicx_matrix_rows_int
1560     \int_incr:N \l__physicx_matrix_cols_int
1561     \__physicx_matrix_last_aux_c:
1562     \__physicx_matrix_last_aux_r:
1563     \physicx_matrix_set_r_c:nnn
1564     { \int_use:N \l__physicx_matrix_rows_int }
1565     { \int_use:N \l__physicx_matrix_cols_int }
1566     { \ddots }
1567   }
1568   \l__physicx_matrix_ending_tl
1569 }
1570 \cs_new:Npn \__physicx_matrix_last_aux_c:
1571 {
1572   \int_step_inline:nn \l__physicx_matrix_rows_int
1573   {
1574     \physicx_matrix_set_r_c:nnn
1575     {##1} { \int_use:N \l__physicx_matrix_cols_int }
1576     { \cdots }
1577   }
1578 }
1579 \cs_new:Npn \__physicx_matrix_last_aux_r:
1580 {
1581   \int_step_inline:nn \l__physicx_matrix_cols_int
1582   {
1583     \physicx_matrix_set_r_c:nnn
1584     { \int_use:N \l__physicx_matrix_rows_int } {##1}
1585     { \vdots }
1586   }
1587 }

```

(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 1588 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
\NewGeneralMatrix 1589 {
\newdiagonalmatrix 1590   \NewDocumentCommand #2 { t+ m o o m m }
\NewDiagonalMatrix 1591   {
  \IfBooleanTF {##1}
  {
    \IfNoValueTF {##3}
    { \newcommand ##2 { #1 + [##5] {##6} } }
    {
      \IfNoValueTF {##4}
      { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
      { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
    }
  }

```



```

1600     }
1601   }
1602   {
1603     \IfNoValueTF {##3}
1604     { \newcommand ##2 { #1 [##5] {##6} } }
1605     {
1606       \IfNoValueTF {##4}
1607       { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1608       { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1609     }
1610   }
1611 }
1612 \NewDocumentCommand #3 { t+ m m m m }
1613 {
1614   \IfBooleanTF {##1}
1615   { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1616   { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1617 }
1618 }
1619 \__physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1620 \__physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1621 \NewDocumentCommand \newgeneralmatrix { t+ m o o m }
1622 {
1623   \IfBooleanTF {#1}
1624   {
1625     \IfNoValueTF {#3}
1626     { \newcommand #2 { \generalmatrix + {#5} } }
1627     {
1628       \IfNoValueTF {#4}
1629       { \newcommand #2 [#3] { \generalmatrix + {#5} } }
1630       { \newcommand #2 [#3] [#4] { \generalmatrix + {#5} } }
1631     }
1632   }
1633   {
1634     \IfNoValueTF {#3}
1635     { \newcommand #2 { \generalmatrix {#5} } }
1636     {
1637       \IfNoValueTF {#4}
1638       { \newcommand #2 [#3] { \generalmatrix {#5} } }
1639       { \newcommand #2 [#3] [#4] { \generalmatrix {#5} } }
1640     }
1641   }
1642 }
1643 \NewDocumentCommand \NewGeneralMatrix { t+ m m m m }
1644 {
1645   \IfBooleanTF {#1}
1646   { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1647   { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1648 }

```

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

```

1649 \endpackage

```

Index

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

Symbols	
$\backslash +$	154
$\backslash -$	154
$\backslash \backslash$	44, 911, 918, 925, 949, 1261, 1311, 1377, 1472
$\backslash \{$	361, 380, 398, 439, 468
$\backslash \}$	361, 382, 398, 439, 468
A	
$\backslash A$	149, 154
$\backslash \text{abs}$	<u>432</u>
$\backslash \text{absolutevalue}$	<u>432</u>
$\backslash \text{acomm}$	<u>432</u>
$\backslash \text{aftergroup}$	202
$\backslash \text{anticommutator}$	<u>432</u>
B	
$\backslash \text{begin}$ 381, 384, 388, 391, 392, 393, 398, 399, 403, 406, 407, 408, 768, 769, 770, 771, 772, 773, 774, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793	
$\backslash \text{bgroup}$	201, 203
$\backslash \text{Big}$	414
$\backslash \text{big}$	414
$\backslash \text{Bigg}$	414
$\backslash \text{bigg}$	414
$\backslash \text{Biggl}$	421
$\backslash \text{biggl}$	420
$\backslash \text{Biggr}$	421
$\backslash \text{biggr}$	420
$\backslash \text{Bigl}$	419
$\backslash \text{bigl}$	418
$\backslash \text{Bigr}$	419
$\backslash \text{bigr}$	418
$\backslash \text{boldsymbol}$	191
bool commands:	
$\backslash \text{bool_case_false:n}$	331, 332, 339
$\backslash \text{bool_case_true:nTF}$	416, 1181
$\backslash \text{bool_if:N}$	16, 21, 26, 72, 182, 345, 347, 351, 843, 898, 905, 915, 923, 935, 942, 946, 953, 957, 1221, 1346, 1434, 1518, 1557
$\backslash \text{bool_if_p:N}$	1219, 1307
$\backslash \text{bool_if_p:n}$	418, 419, 420, 421, 1218, 1306
$\backslash \text{bool_lazy_and_p:nn}$	1186, 1191
$\backslash \text{bool_lazy_or:nnTF}$	31, 113, 124, 1217, 1305, 1548
$\backslash \text{bool_new:N}$	9, 10, 11, 12, 13, 38, 53, 247, 249, 527, 528, 529, 537, 551, 559, 560
$\backslash \text{bool_set:Nn}$	258, 259, 1410
$\backslash \text{bool_set_false:N}$	82, 170, 228, 831, 859, 874, 991
$\backslash \text{bool_set_inverse:N}$	597, 714
$\backslash \text{bool_set_true:N}$	116, 127, 137, 177, 224, 227, 830, 863, 867, 878, 882, 985, 1284, 1290
$\backslash \text{c_false_bool}$	376, 396
$\backslash \text{c_true_bool}$	359, 369, 376, 396
$\backslash \text{Bqty}$	<u>432</u>
$\backslash \text{bqty}$	<u>432</u>
C	
$\backslash \text{cdots}$	614, 897, 900, 901, 904, 906, 937, 938, 941, 943, 1576
clist commands:	
$\backslash \text{clist_clear:N}$	799
$\backslash \text{clist_concat:NNN}$	806
$\backslash \text{clist_if_in:NnTF}$	1134
$\backslash \text{clist_map_break:n}$	163, 175
$\backslash \text{clist_map_inline:nn}$ 69, 159, 171, 804	
$\backslash \text{clist_new:N}$	535, 536
$\backslash \text{clist_put_right:Nn}$..	676, 682, 1125
$\backslash \text{clist_set:Nn}$	796
$\backslash \text{clist_set_eq:NN}$	1101
$\backslash \text{c_empty_clist}$	1101
$\backslash \text{comm}$	466
$\backslash \text{commamatrix}$	26, <u>1295</u> , 1620
$\backslash \text{commutator}$	<u>432</u>
cs commands:	
$\backslash \text{cs:w}$	32, 33, 705
$\backslash \text{cs_end:}$	32, 33, 490, 705
$\backslash \text{cs_generate_variant:Nn}$	3, 4, 5, 6, 76, 79, 185, 342, 969, 1107, 1164, 1505, 1515
$\backslash \text{cs_if_exist_p:N}$	1187, 1192
$\backslash \text{cs_new:Npn}$	7, 111, 119, 122, 134, 145, 157, 189, 193, 197, 201, 202, 203, 204, 252, 262, 274, 311, 318, 343, 412, 562, 752, 757, 962, 970, 998, 1011, 1022, 1032, 1058, 1097, 1099, 1108, 1109, 1143, 1165, 1239, 1244, 1280, 1337,

1364, 1381, 1393, 1432, 1448, 1493, 1506, 1516, 1532, 1570, 1579, 1588	\exp_args:NV 1351, 1359
\cs_new_eq:NN	\exp_not:N 8, 282, 646
. 121, 144, 521, 541, 550, 1030	\exp_not:n
\cs_new_nopar:Npn 480, 488, 497, 499, 504, 510, 543, 544, 547, 548, 549, 1320 286, 287, 491, 493, 541, 572, 576
\cs_new_protected:Nn 841	\exp_stop_f:
\cs_new_protected:Npn 54, 59, 64, 77, 80 8, 277, 284, 1034, 1037, 1060, 1072
\cs_new_protected_nopar:Npn	F
. 795, 797, 802	fi commands:
\cs_set:Nn . 719, 720, 721, 722, 723, 724	\fi: 288, 303, 308, 309, 410, 475, 494, 912, 1055, 1056, 1094, 1095, 1331, 1378
\cs_set:Npn 435, 445, 446, 466, 471, 472, 473, 603, 612, 621, 630, 639	fp commands:
\cs_set:Npx 646	\fp_eval:n 24, 1184
\cs_set_eq:NN 56, 57, 61, 62, 191, 195, 199, 519, 572, 574, 576, 579, 585, 587, 601, 610, 619, 628, 637, 655, 658, 661, 664, 667, 845, 849, 976, 981, 986, 992, 1223, 1227, 1436, 1440	\fpeval 24, 1192, 1193
\cs_set_nopar:Npn 1285, 1291	G
\cs_set_protected:Npn	\generalmatrix . . 28, 1402, 1626, 1629, 1630, 1635, 1638, 1639, 1646, 1647
. 323, 336, 355, 430	group commands:
\cs_to_str:N 268	\group_begin: . . 828, 1202, 1297, 1406
D	\group_end: 839, 1237, 1318, 1423
\ddots 614, 924, 955, 1555, 1566	I
\DeclareDocumentCommand	if commands:
. 328, 338, 414, 826, 1200, 1295, 1402	\if_bool:N 292, 304, 357, 432
\diagonalmatrix 24, 1200, 1619	\if_case:w 284
dim commands:	\if_cs_exist:w 490
\dim_new:N 561	\if_int_compare:w 277, 907, 1034, 1037, 1060, 1072, 1329, 1374
\dim_use:N 911, 918, 925, 949, 1261, 1311, 1377, 1472	\IfBooleanT 380, 382, 1301
E	\IfBooleanTF 241, 330, 387, 389, 402, 404, 424, 764, 829, 834, 1203, 1404, 1407, 1426, 1427, 1428, 1592, 1614, 1623, 1645
\egroup 202, 204	\IfNoValueTF 1594, 1597, 1603, 1606, 1625, 1628, 1634, 1637
else commands:	int commands:
\else:	\int_compare:nNnTF 92, 102, 136
. 492, 909, 1036, 1046, 1071, 1083, 1376	\int_compare:nTF 857, 872
\end . . 381, 384, 388, 391, 392, 393, 398, 399, 403, 406, 407, 408, 768, 769, 770, 771, 772, 773, 774, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793	\int_compare_p:nNn . 114, 115, 125, 126
\eval 432	\int_eval:n . . 24, 1041, 1050, 1068, 1080, 1090, 1091, 1183, 1553, 1554
\evaluated 357, 445	\int_incr:N . 276, 1016, 1350, 1358, 1538, 1540, 1544, 1546, 1559, 1560
exp commands:	\int_max:nn 483, 485
\exp_after:wN	\int_new:N
. . . 1252, 1259, 1268, 1275, 1322, 1324, 1332, 1457, 1466, 1478, 1487	. . . 39, 40, 49, 50, 51, 52, 251, 532, 533
\exp_args:Nf 1006	\int_set:Nn 67, 68, 91, 101, 257, 482, 484, 832, 833, 860, 875, 1343
\exp_args:Nnno 1136, 1171	\int_set_eq:NN
\exp_args:Nno 161, 173 89, 94, 99, 104, 1019, 1330
\exp_args:No 632, 641, 741, 1118	\int_step_inline:nn
 885, 1167, 1169, 1246, 1248, 1264, 1451, 1453, 1474, 1572, 1581
	\int_step_inline:nnn 129, 139, 891, 929

<code>\int_use:N</code>	1261,	<code>\norm</code>	432
	1270, 1277, 1468, 1470, 1480, 1482,		
	1489, 1491, 1564, 1565, 1575, 1584		
<code>\int_zero:N</code>	1013, 1340		
<code>\interval</code>	24, 1187, 1188		
K			
keys commands:			
<code>\keys_define:nn</code>	205, 565, 567, 754, 759		
<code>\keys_if_exist:nnTF</code>	161, 173		
<code>\l_keys_key_str</code>			
	161, 164, 173, 178, 741, 744		
<code>\l_keys_path_str</code>	748		
<code>\keys_set:nn</code>	3, 164, 178, 242,		
	243, 566, 743, 812, 835, 836, 1204,		
	1205, 1298, 1300, 1302, 1408, 1409		
keyval commands:			
<code>\keyval_parse:NNn</code>	1102		
<code>\keyval_parse:nnn</code>	964		
L			
<code>\langle</code>	365, 392, 407		
<code>\ldots</code>	614		
<code>\left</code>	201, 380,		
	387, 392, 398, 399, 402, 406, 407, 408		
<code>\let</code>	325		
<code>\lgroup</code>	387, 402		
<code>\lVert</code>	449		
M			
<code>\mathcal</code>	187, 188		
<code>\mathclose</code>	201, 203		
<code>\mathopen</code>	201, 203		
<code>\matrixquantity</code>	357		
<code>\mqty</code>	432		
msg commands:			
<code>\msg_error:nnn</code>	996		
<code>\msg_error:nnnn</code>	747		
<code>\msg_new:nnn</code>	47		
<code>\msg_new:nnnn</code>	41		
N			
<code>\NewCommaMatrix</code>	1588		
<code>\newcommamatrix</code>	1588		
<code>\newcommand</code>			
	1595, 1598, 1599, 1604, 1607, 1608,		
	1626, 1629, 1630, 1635, 1638, 1639		
<code>\NewDiagonalMatrix</code>	1588		
<code>\newdiagonalmatrix</code>	1588		
<code>\NewDocumentCommand</code>			
	239, 762, 1590, 1612,		
	1615, 1616, 1621, 1643, 1646, 1647		
<code>\NewGeneralMatrix</code>	1588		
<code>\newgeneralmatrix</code>	1588		
<code>\nobreak</code>	371, 372, 373		
		<code>\OOrder</code>	476
		<code>\oorder</code>	432
		or commands:	
		<code>\or:</code>	285, 286, 287
		<code>\Order</code>	432
		<code>\order</code>	432
		<code>\Ordersymbol</code>	233
		<code>\ordersymbol</code>	232
P			
<code>\pb</code>	432		
peek commands:			
		<code>\peek_charcode_ignore_spaces:N</code>	325
physicx commands:			
		<code>\physicx_bf:</code>	191, 195, 199
		<code>\physicx_compat:</code>	14
		<code>\physicx_compat:TF</code>	230, 450
		<code>\physicx_construct:nnn</code>	
			1206, 1411, 1532
		<code>\physicx_declare_legacy_paren:NnnnNNn</code>	
			411, 436, 437, 438, 439, 441, 448,
			452, 454, 458, 460, 463, 467, 476, 478
		<code>\physicx_declare_legacy_quantity:nnNn</code>	
			245, 358, 368, 375, 395
		<code>\physicx_if_num:n</code>	147
		<code>\physicx_if_num:nTF</code>	741, 855, 870, 1178
		<code>\physicx_if_num_sign:n</code>	152
		<code>\physicx_left:</code>	201, 326, 426
		<code>\physicx_left:N</code>	203, 418, 419, 420, 421
		<code>\physicx_mathtools:</code>	24
		<code>\physicx_mathtools:TF</code>	775
		<code>\physicx_matrix_array_parse:n</code> ...	
			1143, 1415
		<code>\physicx_matrix_array_parse_-</code>	
		main:	1165, 1417
		<code>\physicx_matrix_diag_parse:n</code> ...	
			677, 962, 1208, 1212, 1213, 1419
		<code>\physicx_matrix_item_parse:n</code> ...	
			683, 1099, 1216, 1420
		<code>\physicx_matrix_new_type:nn</code> ...	752
		<code>\physicx_matrix_new_type:nnn</code> ...	752
		<code>\physicx_matrix_set_r_c:nnn</code>	521,
			655, 658, 661, 664, 667, 1017, 1025,
			1040, 1049, 1066, 1078, 1089, 1136,
			1160, 1171, 1552, 1563, 1574, 1583
		<code>\physicx_matrix_use_r_c:nn</code>	
			488, 1253, 1260,
			1269, 1276, 1458, 1467, 1479, 1488
		<code>\c_physicx_oorder_tl</code>	479
		<code>\physicx_option_or:nn</code>	29
		<code>\physicx_option_or:nnTF</code>	
			433, 443, 465, 469

\c_physicx_Order_tl	188, 233, 453, 459, 477	__physicx_declare_legacy_-	
\c_physicx_order_tl	187, 232, 455, 461	quantity_aux:nw	265, 311, 316
\physicx_parse_range:nnN	77, 79, 1114, 1116	__physicx_diagonalmatrix_diag_-	
\physicx_parse_range:nnnN	49	main:	986, 992, 1030, 1035
\physicx_parse_range_check: ..	49, 687	__physicx_diagonalmatrix_-	
\physicx_parse_range_nocheck: ..	49, 688	enhanced:nnn	1223, 1227, 1242
\physicx_qxmatrix:nnn	837, 841	__physicx_diagonalmatrix_-	
\physicx_right:	202, 326, 426	generate_body:NNN	1235, 1244
\physicx_right:N	204, 418, 419, 420, 421	__physicx_diagonalmatrix_-	
\physicx_search_also:nn	157, 168, 185, 186	generate_enhanced_body:NNN ..	1234, 1239
\physicx_search_also:nnTF	733	__physicx_diagonalmatrix_no_-	
\physicx_short:	19	diag:	993, 1022, 1031
\physicx_use_amssymb_type: ..	189, 238	__physicx_diagonalmatrix_set_-	
\physicx_use_uni_bf_type:	197	diag:	987, 1011
\physicx_use_uni_bfit_type: ..	193, 237	\l__physicx_end_int	50,
physicx internal commands:		99, 101, 102, 104, 126, 130, 136, 140	
__physicx_adi:nnn	562, 719, 720, 721, 722, 723, 724, 1535	__physicx_expand:w	541, 572,
\l__physicx_begin_int	49,	574, 576, 579, 1002, 1113, 1147, 1304	
89, 91, 92, 94, 125, 126, 130, 136, 140		__physicx_generalmatrix: ..	1421, 1432
\l__physicx_cmd_arg_int	251, 257, 276, 277, 284	__physicx_generalmatrix_-	
\l__physicx_cmd_arg_spec_tl	250, 256, 269, 278	generate:nnn	1436, 1440, 1445
\l__physicx_cmd_auto_body_bool ..	249, 259, 304, 347, 351	__physicx_if_can_num:n	1176
\l__physicx_cmd_auto_body_tl ..	248, 255, 271, 305, 306	__physicx_if_keyval:nTF ..	145, 1211
\l__physicx_cmd_noauto_body_bool ..	247, 258, 292, 345	\l__physicx_invalid_range_bool ..	53, 72, 82, 116, 127, 137
\l__physicx_cmd_noauto_body_tl ..	246, 254, 270, 293, 294	\l__physicx_item_ignore_clist ..	1101, 1125, 1134
\l__physicx_col_list_seq	539, 593, 1288, 1292	\g__physicx_mathtools_bool	9, 26, 227, 228
__physicx_commamatrix_enhanced:	1308, 1337	__physicx_matrix_add_data:nn ..	679, 685, 800, 802
__physicx_commamatrix_enhanced_-	1351, 1359, 1364	\l__physicx_matrix_after_begin_-	
aux:nNn	1352, 1381	tl	557, 695, 697, 1332
__physicx_commamatrix_enhanced_-	1360, 1393	\l__physicx_matrix_after_end_tl ..	558, 698, 700, 1334
aux_e:nnn	11, 16, 207, 224	__physicx_matrix_appto_body_-	
__physicx_commamatrix_enhanced_-	1280	e:nnn	846, 1224, 1437, 1493
aux_ne:nnn	252, 264	__physicx_matrix_appto_body_-	
\g__physicx_compat_bool	274, 315	ne:nnn	850, 1228, 1441, 1493
__physicx_declare_init:	267, 318, 342	\l__physicx_matrix_args_tl	556, 693, 694, 1332
__physicx_declare_init:nnn ..		\l__physicx_matrix_array_tl	530, 569, 1303,
__physicx_declare_legacy_-		1304, 1310, 1312, 1315, 1342, 1415	
quantity_aux:nnnn		__physicx_matrix_autocalc:nn ..	550, 585, 587, 1098, 1149, 1155
__physicx_declare_legacy_-		..	548, 690, 1332
quantity_aux:NNnnn ..		__physicx_matrix_begin:w	563, 725, 727, 1534
		\l__physicx_matrix_beginning_tl ..	531, 853, 888, 893, 894, 900,
		..	901, 906, 911, 918, 920, 922, 924,

925, 927, 931, 932, 937, 938, 943,
 949, 950, 952, 954, 955, 1250, 1257,
 1266, 1273, 1314, 1326, 1333, 1339,
 1371, 1462, 1471, 1483, 1495, 1508
 _physicx_matrix_calc:nn
 480, 585, 977
 _physicx_matrix_col_iterate:n .
 547, 671, 1291, 1389, 1399, 1501, 1512
 \l_physicx_matrix_cols_int
 . 484, 485, 533, 582, 833, 872, 875,
 891, 921, 929, 951, 958, 959, 1013,
 1016, 1020, 1068, 1080, 1091, 1116,
 1169, 1329, 1330, 1521, 1528, 1538,
 1540, 1554, 1560, 1565, 1575, 1581
 \l_physicx_matrix_diag_bool ...
 537, 985, 991
 _physicx_matrix_diag_calc:nn ..
 976, 981, 1026,
 1043, 1052, 1061, 1073, 1084, 1097
 \l_physicx_matrix_diag_clist ...
 535, 674, 676, 1208, 1419
 _physicx_matrix_diag_parse_-
 aux:n 965, 970
 _physicx_matrix_diag_parse_-
 aux:nn 966, 998
 _physicx_matrix_diag_parse_-
 aux_anti:n 1006, 1058
 _physicx_matrix_diag_parse_-
 aux_regu:n 1009, 1032
 \l_physicx_matrix_dotcol_bool ..
 529,
 874, 878, 882, 898, 923, 935, 953, 957
 \l_physicx_matrix_dotrow_bool ..
 528, 859, 863, 867, 915
 _physicx_matrix_element_-
 aux:nnn 601, 607,
 610, 616, 619, 625, 628, 634, 637, 642
 _physicx_matrix_end: 691, 1334
 _physicx_matrix_end:w 549
 \l_physicx_matrix_ending_tl ...
 564, 728, 730, 1568
 \l_physicx_matrix_enhanced_bool
 560, 711, 714, 1219, 1307
 _physicx_matrix_enhanced_init:
 1280, 1345, 1450
 \l_physicx_matrix_expand_-
 element_bool 551,
 650, 843, 1221, 1284, 1290, 1346, 1434
 _physicx_matrix_generate_-
 body:NNNN 1241, 1444, 1447
 \l_physicx_matrix_infinite_bool
 527, 594,
 597, 830, 831, 905, 942, 946, 1410, 1557
 \l_physicx_matrix_item_clist ...
 536, 680, 682, 1216, 1420
 _physicx_matrix_item_parse_-
 aux:n 1103, 1108
 _physicx_matrix_item_parse_-
 aux:nn 1104, 1109
 _physicx_matrix_last_aux_c: ...
 1539, 1561, 1570
 _physicx_matrix_last_aux_r: ...
 1545, 1562, 1579
 \l_physicx_matrix_last_col_tl ..
 546, 673, 1536, 1550
 \l_physicx_matrix_last_row_tl ..
 545, 672, 1542, 1549
 \l_physicx_matrix_main_tl
 534, 589, 1172, 1413
 _physicx_matrix_row_iterate:n .
 544, 670, 1285, 1388, 1398, 1500, 1511
 \l_physicx_matrix_rows_int
 482, 483, 532, 581,
 832, 857, 860, 885, 907, 1019, 1114,
 1167, 1343, 1374, 1522, 1527, 1544,
 1546, 1553, 1559, 1564, 1572, 1584
 _physicx_matrix_save_or_print:
 838, 1236, 1317, 1320, 1422
 \l_physicx_matrix_save_tl
 703, 705, 1322, 1325
 \l_physicx_matrix_sep_dim
 561, 701, 911,
 918, 925, 949, 1261, 1311, 1377, 1472
 _physicx_matrix_set_data:nn ...
 678, 684, 797
 _physicx_matrix_set_r_c_-
 ckall:nnn 519, 668
 _physicx_matrix_set_r_c_-
 ckcp:nnn 504, 659
 _physicx_matrix_set_r_c_-
 ckig:nnn 499, 662
 _physicx_matrix_set_r_c_-
 ckigep:nnn 510, 520, 665
 _physicx_matrix_set_r_c_-
 nock:nnn 497, 522, 656
 \l_physicx_matrix_tmparr_c_sep .
 1154, 1157, 1158
 \l_physicx_matrix_tmparr_r_sep .
 1148, 1150, 1152
 _physicx_matrix_transpose:N ...
 1233, 1443, 1516
 \l_physicx_matrix_transpose_-
 bool 559, 706, 1518
 \l_physicx_max_int
 51, 68, 99, 102, 104, 114, 125
 \l_physicx_min_int
 52, 67, 89, 92, 94, 115

<code>\seq_map_indexed_inline:Nn</code>	<code>\tl_if_empty:NTF</code>
. 1014, 1024, 1038, 1047, 88, 98, 1413, 1536, 1542
1064, 1076, 1087, 1152, 1158, 1369	<code>\tl_if_empty:nTF</code>
<code>\seq_map_inline:Nn</code> 506, 514, 605, 1209, 1299
. 1121, 1123, 1130, 1132	<code>\tl_if_empty_p:N</code> 1549, 1550
<code>\seq_map_tokens:Nn</code> 1348, 1356	<code>\tl_if_eq:nnTF</code> 501, 512, 1118
<code>\seq_new:N</code> 538, 539	<code>\tl_if_head_eq_charcode:nNTF</code> 1004
<code>\seq_pop_left:NN</code> 87, 97	<code>\tl_if_head_eq_meaning_p:nN</code>
<code>\seq_put_right:Nn</code> 1183, 1184, 1188, 1193
. 117, 120, 131, 141, 1383, 1395	<code>\tl_if_in:nnTF</code> 84, 146, 614, 623
<code>\seq_set_eq:NN</code> 66, 1368	<code>\tl_if_novalue_p:n</code> 282
<code>\seq_set_split:Nnn</code> 5, 86,	<code>\tl_new:N</code> 245,
591, 593, 1003, 1148, 1154, 1341, 1366	246, 248, 250, 530, 531, 534, 545,
<code>\seq_use:Nn</code> 1373	546, 553, 555, 556, 557, 558, 563, 564
<code>\setmatrixdata</code> 795	<code>\tl_put_right:Nn</code> 278, 293, 294,
<code>\setmatrixtype</code> 752, 768, 769,	305, 306, 649, 697, 700, 727, 730,
770, 771, 772, 773, 774, 777, 778,	888, 893, 894, 900, 901, 906, 911,
779, 780, 781, 782, 783, 784, 785,	918, 920, 922, 924, 925, 927, 931,
786, 787, 788, 789, 790, 791, 792, 793	932, 937, 938, 943, 949, 950, 952,
<code>\smallmatrixquantity</code> 357	954, 955, 1250, 1257, 1266, 1273,
<code>\smqty</code> 432	1371, 1462, 1471, 1483, 1495, 1508
str commands:	<code>\tl_replace_all:Nnn</code> 6, 1310, 1312
<code>\str_case_e:nnTF</code> 972	<code>\tl_set:Nn</code> 279,
<code>\str_const:Nn</code> 524, 525, 526	498, 502, 507, 508, 515, 516, 693,
<code>\str_if_eq:nnTF</code> 298, 300	705, 1000, 1001, 1111, 1112, 1145,
<code>\symbf</code> 199	1146, 1303, 1455, 1464, 1476, 1485
<code>\symbfit</code> 195	<code>\tl_set_eq:NN</code> 232, 233, 1314
	<code>\tl_tail:n</code> 1007
	token commands:
	<code>\token_if_cs:NTF</code> 1322
	<code>\TrimSpaces</code> 762
	U
	use commands:
	<code>\use:n</code> 579
	<code>\use:nn</code> 1447, 1529
	<code>\use:nnnn</code> 4
	<code>\use_i:nn</code> 321, 1115, 1232, 1428
	<code>\use_i:nnn</code> 1428
	<code>\use_i_ii:nnn</code> 1230, 1427
	<code>\use_ii:nn</code> 321, 1117
	<code>\use_ii_i:nn</code> 1447, 1523
	<code>\use_none:nn</code> 550, 587, 981
	V
	<code>\vdots</code> 614, 920, 922, 924, 950, 952, 954, 1585
	<code>\Vert</code> 366, 408
	<code>\vert</code> 364, 371, 372, 373, 406, 438, 442
	<code>\vqty</code> 432
	Z
	<code>\Z</code> 149, 154
<code>\seq_map_indexed_inline:Nn</code>	
. 1014, 1024, 1038, 1047,	
1064, 1076, 1087, 1152, 1158, 1369	
<code>\seq_map_inline:Nn</code>	
. 1121, 1123, 1130, 1132	
<code>\seq_map_tokens:Nn</code> 1348, 1356	
<code>\seq_new:N</code> 538, 539	
<code>\seq_pop_left:NN</code> 87, 97	
<code>\seq_put_right:Nn</code>	
. 117, 120, 131, 141, 1383, 1395	
<code>\seq_set_eq:NN</code> 66, 1368	
<code>\seq_set_split:Nnn</code> 5, 86,	
591, 593, 1003, 1148, 1154, 1341, 1366	
<code>\seq_use:Nn</code> 1373	
<code>\setmatrixdata</code> 795	
<code>\setmatrixtype</code> 752, 768, 769,	
770, 771, 772, 773, 774, 777, 778,	
779, 780, 781, 782, 783, 784, 785,	
786, 787, 788, 789, 790, 791, 792, 793	
<code>\smallmatrixquantity</code> 357	
<code>\smqty</code> 432	
str commands:	
<code>\str_case_e:nnTF</code> 972	
<code>\str_const:Nn</code> 524, 525, 526	
<code>\str_if_eq:nnTF</code> 298, 300	
<code>\symbf</code> 199	
<code>\symbfit</code> 195	
T	
TeX and L ^A T _E X 2 _ε commands:	
<code>\@declareparncmd</code> 411	
<code>\@declarequantitycmd</code> 245	
<code>\@ifpackageloaded</code> 223, 226, 236	
<code>\c@MaxMatrixCols</code> 710, 1329, 1330	
<code>\physicsx@align</code>	
. 524, 716, 1154, 1313, 1367	
<code>\physicsx@cr</code> 525, 715, 1148, 1311, 1341	
<code>\physicsx@matricelement</code>	
. 543, 598, 602,	
603, 611, 612, 620, 621, 629, 630,	
638, 639, 646, 1387, 1397, 1499, 1510	
<code>\physicsx@sep</code> 526, 591, 593, 717, 1003	
tex commands:	
<code>\tex_advance:D</code> 958, 959	
text commands:	
<code>\text_expand:n</code> 574, 1385, 1497	
tl commands:	
<code>\c_empty_tl</code> 337	
<code>\tl_clear:N</code> 254, 255, 256, 853, 1339	
<code>\tl_const:Nn</code> 187, 188	
<code>\tl_gset_eq:NN</code> 1324	