

# The `physicx` package

Wenjian Chern (Longaster\*)

September 13, 2022, version v0.3.4

## Abstract

`physicx`

## 1 Implementation

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

---

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

```

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

```

## 1.1 Utils functions

```

\physicx_parse_range:nnnN Parse range, such as -3,6-8,9,10-.
\physicx_parse_range_check:
\physicx_parse_range_nocheck:
62 \int_new:N \l__physicx_begin_range_int
63 \int_new:N \l__physicx_end_range_int
64 \int_new:N \l__physicx_max_range_int
65 \int_new:N \l__physicx_min_range_int
66 \bool_new:N \l__physicx_invalid_range_bool % range
67 \cs_new_protected:Npn \physicx_parse_range_check:
68 {
69     \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_check:n
70     \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_check:
71 }
72 \cs_new_protected:Npn \physicx_parse_range_nocheck:
73 {
74     \cs_set_eq:NN \__physicx_parse_range_single:n \__physicx_parse_range_single_nocheck:n
75     \cs_set_eq:NN \__physicx_parse_range_range: \__physicx_parse_range_range_nocheck:
76 }
77 \cs_new_protected:Npn \physicx_parse_range:nnnN #1#2#3#4
78 {
79     \seq_set_eq:NN #4 \c_empty_seq
80     \int_set:Nn \l__physicx_min_range_int {#1}
81     \int_set:Nn \l__physicx_max_range_int {#2}
82     \clist_map_inline:nn {#3}
83     {

```

```

84     \__physicx_parse_range_aux:n {##1}
85     \bool_if:NF \l__physicx_invalid_range_bool
86     { \seq_concat:NNN #4 #4 \l__physicx_tmpa_seq }
87   }
88 }
89 \cs_generate_variant:Nn \physicx_parse_range:nnnN { nnvN, nneN }
90 \cs_new_protected:Npn \physicx_parse_range:nnN
91   { \physicx_parse_range:nnnN { 1 } }
92 \cs_generate_variant:Nn \physicx_parse_range:nnN { nvN, neN }
93 \cs_new_protected:Npn \__physicx_parse_range_aux:n #1
94   {
95     \bool_set_false:N \l__physicx_invalid_range_bool
96     \seq_clear:N \l__physicx_tmpa_seq
97     \__physicx_parse_range_action:nnn
98       {#1}
99     { \__physicx_parse_range_single:n {#1} }
100    {
101      \tl_if_empty:NTF \l__physicx_tmpa_tl
102        { \int_set_eq:NN \l__physicx_begin_range_int \l__physicx_min_range_int }
103        { \int_set:Nn \l__physicx_begin_range_int { \l__physicx_tmpa_tl } }
104      \tl_if_empty:NTF \l__physicx_tmpb_tl
105        { \int_set_eq:NN \l__physicx_end_range_int \l__physicx_max_range_int }
106        { \int_set:Nn \l__physicx_end_range_int { \l__physicx_tmpb_tl } }
107      \__physicx_parse_range_range:
108    }
109  }
110 \cs_new:Npn \physicx_set_parse_range_delimiter:n #1
111   {
112     \tl_if_empty:NTF {#1}
113     {
114       \cs_set:Npn \__physicx_parse_range_action:nnn ##1
115         { \__physicx_parse_range_aux:w ##1 \__physicx_do_nothing: \q_nil \q_physicx_special }
116       \cs_set:Npn \__physicx_parse_range_aux:w ##1##2 ##3 \q_physicx_special
117         {
118           \tl_set:Nx \l__physicx_tmpa_tl { \tl_trim_spaces:n {##1} }
119           \tl_set:Nx \l__physicx_tmpb_tl { \tl_trim_spaces:n {##2} }
120           \quark_if_nil:NTF {##3}
121         }
122     }
123     {
124       \cs_set:Npn \__physicx_parse_range_action:nnn ##1
125         { \__physicx_parse_range_aux:w ##1 #1 #1 \q_physicx_special }
126       \cs_set:Npn \__physicx_parse_range_aux:w ##1 #1 ##2 #1 ##3 \q_physicx_special
127         {
128           \tl_set:Nx \l__physicx_tmpa_tl { \tl_trim_spaces:n {##1} }
129           \tl_set:Nx \l__physicx_tmpb_tl { \tl_trim_spaces:n {##2} }
130           \tl_if_blank:NTF {##3}
131         }
132     }
133   }
134 \physicx_set_parse_range_delimiter:n { - }
135 \cs_new:Npn \__physicx_parse_range_single_check:n #1
136   {
137     \bool_lazy_or:nnTF

```

```

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

```

(End definition for \physicx\_parse\_range:nnnN, \physicx\_parse\_range\_check:, and \physicx\_parse\_range\_nocheck:. These functions are documented on page ??.)

```

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

```

```

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

```

```

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

```

```

296     {\IfBooleanTF{#1} { \physics_bf:{\hat{#2}} } { \hat{\mathbf{#2}} } }
297     \setmathfont[range={"2219}]{STIX~Two~Math}
298     \DeclareDocumentCommand \dotproduct { } { \vysmbllkcircle }
299     \DeclareDocumentCommand \crossproduct { } { \vectimes }
300     \DeclareDocumentCommand \vnabla { } { \_physics_vnabla: }
301 }
302 \ifpackageloaded {physics} {
303   \AtBeginDocument{
304     \cs_set_eq:NN \divisionsymbol \div
305     \cs_set_eq:NN \div \divergence
306     \bool_if:NT \g__physics_fixdif_bool { \cs_set_eq:NN \diffd \@dif }
307     \let\real\Re \DeclareDocumentCommand\Re{g}{\IfNoValueTF{#1}{\operatorname{Re}}{\fbrace
308     \let\imaginary\Im \DeclareDocumentCommand\Im{g}{\IfNoValueTF{#1}{\operatorname{Im}}{\fbrace
309   }
310 } { }
311 }
312 \bool_if:NT \g__physics_original_bool
313 {
314   \AtBeginDocument{
315     \ifpackageloaded{physics}
316     {
317       \cs_set_eq:NN \Re \real
318       \cs_set_eq:NN \Im \imaginary
319       \cs_set_eq:NN \div \divisionsymbol
320     }
321     {}
322   }
323 }
324 %
325 \bool_if:NT \g__physics_fixdif_bool
326 {
327   \hook_gput_code:nnn { package/fixdif/before } { physics }
328   { \cs_set_eq:NN \_physics_nabla: \nabla }
329   \hook_gput_code:nnn { package/fixdif/after } { physics }
330   { \tl_map_function:nN { \letdif \newdif \renewdif } \_physics_fixdif_list:N }
331   \AtBeginDocument { \renewdif \_physics_vnabla: { \sympbf \_physics_nabla: } }
332 }
333 \cs_new_protected:Npn \_physics_fixdif_list:N #1
334 {
335   \cs_if_free:cT { \cs_to_str:N #1 list }
336   {
337     \cs_new_protected:cpn { \cs_to_str:N #1 list }
338     {
339       \keyval_parse:nnn { \_physics_fixdif:Nn #1 } { \_physics_fixdif:Nnn #1 }
340     }
341   }
342 }
343 \cs_new_protected:Npn \_physics_fixdif:Nnn #1#2#3
344 {
345   \tl_if_head_eq_meaning:nNTF {#2} *
346   { \exp_args:Nnc #1 * { \tl_tail:n {#2} } {#3} }
347   { \exp_args:Nc #1 {#2} {#3} }
348 }
349 \cs_new_protected:Npn \_physics_fixdif:Nn #1#2

```

```

350 {
351   \exp_args:NNnx \__physicx_fixdif:Nnn #1 {#2}
352   { \tl_if_head_eq_meaning:NNTF {#2} * { \tl_tail:n {#2} } {#2} }
353 }

```

`\physicxset` `physicx` setup command.

```

354 \NewDocumentCommand \physicxset { s m }
355 {
356   \IfBooleanTF {#1}
357   { \keys_set:nn { physicx/#2 } }
358   { \keys_set:nn { physicx } {#2} }
359 }

```

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

## 1.2 Quantity things

### 1.2.1 New quantity interfaces

```

360 \tl_new:N \l__physicx_quantity_args_tl
361 \tl_new:N \l__physicx_quantity_code_tl
362 \tl_new:N \l__physicx_quantity_left_size_tl
363 \tl_new:N \l__physicx_quantity_left_tl
364 \tl_new:N \l__physicx_quantity_post_tl
365 \tl_new:N \l__physicx_quantity_pre_tl
366 \tl_new:N \l__physicx_quantity_right_size_tl
367 \tl_new:N \l__physicx_quantity_right_tl
368 \keys_define:nn { physicx }
369 { quantity .code:n = \keys_set:nn { physicx/quantity } {#1} }
370 \keys_define:nn { physicx/quantity }
371 {
372   pre .tl_set:N = \l__physicx_quantity_pre_tl ,
373   post .tl_set:N = \l__physicx_quantity_post_tl ,
374   left .tl_set:N = \l__physicx_quantity_left_tl ,
375   right .tl_set:N = \l__physicx_quantity_right_tl ,
376   left-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_left_size_tl #1 } ,
377   right-size .code:n = { \tl_set_eq:NN \l__physicx_quantity_right_size_tl #1 } ,
378   size .meta:n = { left-size = {#1} , right-size = {#1} } ,
379   noauto .meta:n = { left-size = \c_empty_tl , right-size = \c_empty_tl } ,
380   noauto .value_required:n = false ,
381   args .code:n =
382     \tl_set:Nn \l__physicx_quantity_args_tl { [#1] } ,
383   args* .tl_set:N = \l__physicx_quantity_args_tl ,
384   code .tl_set:N = \l__physicx_quantity_code_tl ,
385   type .multichoice: ,
386
387   settype .code:n = \setquantitytype #1 ,
388
389   unknown .code:n =
390     \tl_set:Nx \l__physicx_tmpa_tl { \tl_head:N \l_keys_key_str }
391     \token_if_eq_meaning:NNTF \l__physicx_tmpa_tl \c_backslash_str
392     { \use:n } { \use_ii:nn }
393   {
394     \cs_if_exist:cTF { \tl_tail:N \l_keys_key_str }

```



```

395         {
396             \keys_set:nx { physicx/quantity }
397             { size = \exp_not:c { \tl_tail:N \l_keys_key_str } }
398             \use_none:n
399         }
400         { \use:n }
401     }
402     {
403         \physicx_search_also:nnF
404         {
405             physicx/quantity/type ,
406         }
407         {#1}
408         {
409             \msg_error:nxxx { physicx } { unknown-key }
410             \l_keys_path_str { physicx/quantity }
411         }
412     } ,
413 }
414 \NewDocumentCommand \setquantitytype { >{ \TrimSpaces } m }
415 { \physicx_new_type:nnn { quantity } {#1} }
416 \setquantitytype { b } { left={[] , right={[] } , }
417 \setquantitytype { B } { left={\{ } , right={\} } , }
418 \setquantitytype { p } { left={ ( } , right={ ) } , }
419 \setquantitytype { v } { left=\vert , right=\vert , }
420 \setquantitytype { V } { left=\Vert , right=\Vert , }
421 \setquantitytype { a } { left=\langle , right=\rangle , }
422 \setquantitytype { m } { left=\begin{matrix} , right=\end{matrix} , noauto }
423 \setquantitytype { bm } { left=\begin{bmatrix} , right=\end{bmatrix} , noauto }
424 \setquantitytype { Bm } { left=\begin{Bmatrix} , right=\end{Bmatrix} , noauto }
425 \setquantitytype { pm } { left=\begin{pmatrix} , right=\end{pmatrix} , noauto }
426 \setquantitytype { vm } { left=\begin{vmatrix} , right=\end{vmatrix} , noauto }
427 \setquantitytype { Vm } { left=\begin{Vmatrix} , right=\end{Vmatrix} , noauto }
428 \setquantitytype { sm } { left=\begin{smallmatrix} , right=\end{smallmatrix} , noauto }
429 \physicx_mathtools:T
430 {
431     \setquantitytype { m* } { left=\begin{matrix*} , right=\end{matrix*} , noauto }
432     \setquantitytype { bm* } { left=\begin{bmatrix*} , right=\end{bmatrix*} , noauto }
433     \setquantitytype { Bm* } { left=\begin{Bmatrix*} , right=\end{Bmatrix*} , noauto }
434     \setquantitytype { pm* } { left=\begin{pmatrix*} , right=\end{pmatrix*} , noauto }
435     \setquantitytype { vm* } { left=\begin{vmatrix*} , right=\end{vmatrix*} , noauto }
436     \setquantitytype { Vm* } { left=\begin{Vmatrix*} , right=\end{Vmatrix*} , noauto }
437     \setquantitytype { sm* } { left=\begin{smallmatrix*} , right=\end{smallmatrix*} , noauto }
438     \setquantitytype { sbm } { left=\begin{bsmallmatrix} , right=\end{bsmallmatrix} , noauto }
439     \setquantitytype { sBm } { left=\begin{Bsmallmatrix} , right=\end{Bsmallmatrix} , noauto }
440     \setquantitytype { spm } { left=\begin{psmallmatrix} , right=\end{psmallmatrix} , noauto }
441     \setquantitytype { svm } { left=\begin{vsmallmatrix} , right=\end{vsmallmatrix} , noauto }
442     \setquantitytype { sVm } { left=\begin{Vsmallmatrix} , right=\end{Vsmallmatrix} , noauto }
443     \setquantitytype { sbm* } { left=\begin{bsmallmatrix*} , right=\end{bsmallmatrix*} , noauto }
444     \setquantitytype { sBm* } { left=\begin{Bsmallmatrix*} , right=\end{Bsmallmatrix*} , noauto }
445     \setquantitytype { spm* } { left=\begin{psmallmatrix*} , right=\end{psmallmatrix*} , noauto }
446     \setquantitytype { svm* } { left=\begin{vsmallmatrix*} , right=\end{vsmallmatrix*} , noauto }
447     \setquantitytype { sVm* } { left=\begin{Vsmallmatrix*} , right=\end{Vsmallmatrix*} , noauto }
448 }

```

```

449 \keys_set:nn { physicx/quantity }
450 {
451   left-size = \left ,
452   right-size = \right ,
453   type = p ,
454 }
\physicx_xquantity:nn
\newxquantity
\NewXQuantity
455 \cs_new:Npn \physicx_xquantity:nn #1#2
456 {
457   \group_begin:
458   \keys_set:nn { physicx/quantity } {#1}
459   \tl_if_empty:nF {#2} { \tl_set:Nn \l__physicx_quantity_code_tl {#2} }
460   \__physicx_xquantity_aux:oooo
461   { \l__physicx_quantity_left_tl }
462   { \l__physicx_quantity_args_tl }
463   { \l__physicx_quantity_code_tl }
464   { \l__physicx_quantity_right_tl }
465   \group_end:
466 }
467 \cs_new:Npn \__physicx_xquantity_aux:nnnn #1#2#3#4
468 {
469   \l__physicx_quantity_pre_tl
470   \bool_lazy_or:nnTF
471   { \tl_if_empty_p:N \l__physicx_quantity_left_size_tl }
472   { \tl_if_empty_p:N \l__physicx_quantity_right_size_tl }
473   { #1 #2 #3 #4 }
474   {
475     \bool_lazy_or:nnTF
476     { \token_if_eq_meaning_p:NN \l__physicx_quantity_left_size_tl \left }
477     { \token_if_eq_meaning_p:NN \l__physicx_quantity_right_size_tl \right }
478     { \physicx_left: #1 #2 #3 \physicx_right: #4 }
479     {
480       \physicx_left:N \l__physicx_quantity_left_size_tl #1 #2
481       #3
482       \physicx_right:N \l__physicx_quantity_right_size_tl #4
483     }
484   }
485   \l__physicx_quantity_post_tl
486 }
487 \NewDocumentCommand \xquantity { } { \physicx_xquantity:nn }
488 \cs_generate_variant:Nn \__physicx_xquantity_aux:nnnn { oooo }
489 \NewDocumentCommand \newxquantity { m o o m m }
490 {
491   \IfNoValueTF {#2}
492   {
493     \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
494     { \newcommand ##1 }
495   }
496   {
497     \IfNoValueTF {#3}
498     {
499       \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
500       { \newcommand ##1 [#2] }

```

```

501     }
502     {
503         \cs_set:Npn \__physicx_new_xquantity_aux:w ##1
504             { \newcommand ##1 [#2] [#3] }
505     }
506 }
507 \exp_args:Nc \__physicx_new_xquantity_aux:w
508 { \cs_to_str:N #1~star }
509 { \physicx_xquantity:nn { #4 , noauto } {#5} }
510 \exp_args:Nc \__physicx_new_xquantity_aux:w
511 { \cs_to_str:N #1~unstar }
512 { \physicx_xquantity:nn { #4 } {#5} }
513 \exp_args:NNx \newcommand #1
514 {
515     \exp_not:N \@ifstar
516     \exp_not:c { \cs_to_str:N #1~star }
517     \exp_not:c { \cs_to_str:N #1~unstar }
518 }
519 }
520 \NewDocumentCommand \NewXQuantity { m m m m }
521 {
522     \NewDocumentCommand #1 { s #2 }
523     {
524         \IfBooleanTF {##1}
525         { \physicx_xquantity:nn { #3 , noauto } {#4} }
526         { \physicx_xquantity:nn { #3 } {#4} }
527     }
528 }
529 \NewXQuantity \qxqty { 0{} m } { #2 } {#3}
530 \NewXQuantity \txqty { 0{p} 0{} m } { type={#2}, #3 } {#4}

```

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

## 1.2.2 Legacy quantity

```

\physicx_declare_legacy_quantity:nnNn
  \@declarequantitycmd
531 \tl_new:N \physicx_tmp
532 \tl_new:N \l__physicx_cmd_noauto_body_tl
533 \bool_new:N \l__physicx_cmd_noauto_body_bool
534 \tl_new:N \l__physicx_cmd_auto_body_tl
535 \bool_new:N \l__physicx_cmd_auto_body_bool
536 \tl_new:N \l__physicx_cmd_arg_spec_tl
537 \int_new:N \l__physicx_cmd_arg_int
538 \cs_new:Npn \__physicx_declare_init:nnn #1#2#3
539 {
540     \tl_clear:N \l__physicx_cmd_noauto_body_tl
541     \tl_clear:N \l__physicx_cmd_auto_body_tl
542     \tl_clear:N \l__physicx_cmd_arg_spec_tl
543     \int_set:Nn \l__physicx_cmd_arg_int {#1}
544     \bool_set:Nn \l__physicx_cmd_noauto_body_bool {#2}
545     \bool_set:Nn \l__physicx_cmd_auto_body_bool {#3}
546 }
547 % noauto, auto, cmd, body

```

```

548 \cs_new:Npn \physicx_declare_legacy_quantity:nnNn #1#2#3#4
549 {
550   \__physicx_declare_init:nnn { 3 } {#1} {#2}
551   \__physicx_declare_legacy_quantity_aux:nw #4
552   \q_recursion_tail \q_recursion_tail \q_recursion_stop
553   \__physicx_declare_legacy_quantity_aux:NcVVV
554   #3 { \cs_to_str:N #3 ~ body }
555   \l__physicx_cmd_arg_spec_tl
556   \l__physicx_cmd_noauto_body_tl
557   \l__physicx_cmd_auto_body_tl
558 }
559 % arg spec, pre, body to replace(start from #4), post
560 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nnnn #1#2#3#4
561 {
562   \int_incr:N \l__physicx_cmd_arg_int
563   \if_int_compare:w \l__physicx_cmd_arg_int < 10 \exp_stop_f:
564     \tl_put_right:Nn \l__physicx_cmd_arg_spec_tl {#1}
565     \tl_set:Nx \l__physicx_tmp_tl
566     {
567       {
568         \exp_not:N \tl_if_novalue_p:n
569         {
570           \if_case:w \l__physicx_cmd_arg_int \exp_stop_f:
571           \or: \or: \or:
572           \or: \exp_not:n {##4} \or: \exp_not:n {##5} \or: \exp_not:n {##6}
573           \or: \exp_not:n {##7} \or: \exp_not:n {##8} \or: \exp_not:n {##9}
574           \fi:
575         }
576       }
577     }
578     \if_bool:N \l__physicx_cmd_noauto_body_bool
579       \tl_put_right:No \l__physicx_cmd_noauto_body_tl { \l__physicx_tmp_tl }
580       \tl_put_right:Nn \l__physicx_cmd_noauto_body_tl
581       {
582         {
583           % if is '.', use none
584           \str_if_eq:nnTF {#2} {.} {} {#2}
585           #3
586           \str_if_eq:nnTF {#4} {.} {} {#4}
587         }
588       }
589       \fi:
590       \if_bool:N \l__physicx_cmd_auto_body_bool
591         \tl_put_right:No \l__physicx_cmd_auto_body_tl { \l__physicx_tmp_tl }
592         \tl_put_right:Nn \l__physicx_cmd_auto_body_tl
593         { { ##1 #2 #3 ##2 #4 } }
594       \fi:
595     \fi:
596   }
597   \cs_new:Npn \__physicx_declare_legacy_quantity_aux:nw #1#2
598   {
599     \quark_if_recursion_tail_stop:n {#1}
600     \quark_if_recursion_tail_stop:n {#2}
601     \__physicx_declare_legacy_quantity_aux:nnnn {#1} #2

```

```

602   \__physicx_declare_legacy_quantity_aux:nw
603 }
604 \cs_new:Npn \__physicx_declare_legacy_quantity_aux:NNnnn #1#2#3#4#5
605 {
606   \__physicx_nauto_case:nnnn
607   { \use_i:nn } { \use_ii:nn } { \use_i:nn } { \use_i:nn }
608   {
609     \cs_set_protected:Npn #1
610     {
611       \peek_charcode_ignore_spaces:NTF \let
612       { #2 } { #2 [ \physicx_left: ] \physicx_right: }
613     }
614     \DeclareDocumentCommand #2 { 0{##2} m s #3 }
615     {
616       \IfBooleanTF { ##3 }
617       { \bool_case_false:n {#4} }
618       { \bool_case_false:n {#5} }
619     }
620   }
621   {
622     \cs_set_protected:Npn #1
623     { #2 \c_empty_tl \c_empty_tl }
624     \DeclareDocumentCommand #2 { m m s #3 }
625     { \bool_case_false:n {#4} }
626   }
627 }
628 \cs_generate_variant:Nn \__physicx_declare_legacy_quantity_aux:NNnnn { NcVVV }
629 \cs_new:Npn \__physicx_nauto_case:nnnn #1#2#3#4
630 {
631   \bool_if:NTF \l__physicx_cmd_noauto_body_bool
632   {
633     \bool_if:NTF \l__physicx_cmd_auto_body_bool
634     {#1} {#2}
635   }
636   {
637     \bool_if:NTF \l__physicx_cmd_auto_body_bool
638     {#3} {#4}
639   }
640 }
641 \cs_set_protected:Npn \@declarequantitycmd
642 { \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> 643 \if_bool:N \g__physicx_reqty_bool 644 \physicx_declare_legacy_quantity:nnNn 645 \c_true_bool \c_true_bool \quantity 646 { 647   { !g } { { \{ } { #4 } { \} } } 648   { !o } { { [ } { #5 } { ] } } 649   { !d() } { { ( } { #6 } { ) } } 650   { !d   } { { \vert } { #7 } { \vert } } </pre>
--	--

```

651 { !d<> } { { \langle } { #8 } { \rangle } }
652 { !d== } { { \Vert } { #9 } { \Vert } }
653 }
654 \physicsx_declare_legacy_quantity:nnNn
655 \c_true_bool \c_true_bool \evaluated
656 {
657 { !g } { { . } { #4 \nobreak } { \vert } }
658 { !d[ ] } { { [ ] } { #5 \nobreak } { \vert } }
659 { !d( ) } { { ( ) } { #6 \nobreak } { \vert } }
660 }
661 \physicsx_declare_legacy_quantity:nnNn
662 \c_true_bool \c_false_bool \matrixquantity
663 {
664 { !g }
665 {
666 { \IfBooleanT{#3}{\left\{ } }
667 { \begin{matrix} #4 \end{matrix} }
668 { \IfBooleanT{#3}{\right\} }
669 }
670 { !o } { { \begin{bmatrix} } { #5 } { \end{bmatrix} } }
671 { !d() }
672 {
673 { \IfBooleanTF{#3}{\left\lgroup}{\left( } }
674 { \begin{matrix} #6 \end{matrix} }
675 { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
676 }
677 { !d|| } { { \begin{vmatrix} } { #7 } { \end{vmatrix} } }
678 { !d<> } { { \left\langle } { \begin{matrix} #8 \end{matrix} } { \right\rangle } }
679 { !d== } { { \begin{Vmatrix} } { #9 } { \end{Vmatrix} } }
680 }
681 \physicsx_declare_legacy_quantity:nnNn
682 \c_true_bool \c_false_bool \smallmatrixquantity
683 {
684 { !g } { { \left\{ } { \begin{smallmatrix} #4 \end{smallmatrix} } { \right\} } }
685 { !o } { { \left[ ] } { \begin{smallmatrix} #5 \end{smallmatrix} } { \right]} }
686 { !d() }
687 {
688 { \IfBooleanTF{#3}{\left\lgroup}{\left( } }
689 { \begin{smallmatrix} #6 \end{smallmatrix} }
690 { \IfBooleanTF{#3}{\right\rgroup}{\right)} }
691 }
692 { !d|| } { { \left\vert } { \begin{smallmatrix} #7 \end{smallmatrix} } { \right\vert } }
693 { !d<> } { { \left\langle } { \begin{smallmatrix} #8 \end{smallmatrix} } { \right\rangle } }
694 { !d== } { { \left\Vert } { \begin{smallmatrix} #9 \end{smallmatrix} } { \right\Vert } }
695 }
696 \fi:

```

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

```

\physicsx_declare_legacy_paren:NnnnNn
  \@declareparencmd
697 %% cmd, arg spec, replace(start from #6), pre, left, right, post
698 \cs_new:Npn \physicsx_declare_legacy_paren:NnnnNn #1#2#3#4#5#6#7
699 {
700   \DeclareDocumentCommand #1 { s t\big t\Big t\bigg t\Bigg #2 }

```

```

701 {
702   \bool_case_true:nF
703   {
704     { \bool_if_p:n {##2} } { #4 \physicx_left:N \bigl #5 #3 \physicx_right:N \bigr
705     { \bool_if_p:n {##3} } { #4 \physicx_left:N \Bigl #5 #3 \physicx_right:N \Bigr
706     { \bool_if_p:n {##4} } { #4 \physicx_left:N \biggl #5 #3 \physicx_right:N \biggr
707     { \bool_if_p:n {##5} } { #4 \physicx_left:N \Biggl #5 #3 \physicx_right:N \Biggr
708   }
709   {
710     \IfBooleanTF {##1}
711     { #4 #5 #3 #6 #7 }
712     { #4 \physicx_left: #5 #3 \physicx_right: #6 #7 }
713   }
714 }
715 }
716 \cs_set_protected:Npn \@declareparen cmd
717 { \physicx_declare_legacy_paren:NnnnNNn }

```

(End definition for `\physicx_declare_legacy_paren:NnnnNNn` and `\@declareparen cmd`. These functions are documented on page ??.)

```

\qty Redefine some macros in physics package.
\mqty 718 \if_bool:N \g__physicx_reqty_bool
\smqty 719 \physicx_option_or:nnT { compat } { short }
\pqty 720 {
\qty 721   \cs_set:Npn \qty { \quantity }
\pqty 722   \physicx_declare_legacy_paren:NnnnNNn \pqty { m } {#6} { } { } { } { }
\vqty 723   \physicx_declare_legacy_paren:NnnnNNn \bqty { m } {#6} { } { } [ ] { }
\Bqty 724   \physicx_declare_legacy_paren:NnnnNNn \vqty { m } {#6} { } { } \vert \vert { }
\absolutevalue 725   \physicx_declare_legacy_paren:NnnnNNn \Bqty { m } {#6} { } { } \{ \} { }
\eval 726 }
\abs 727 \physicx_declare_legacy_paren:NnnnNNn \absolutevalue
\abs 728 { m } {#6} { } { } \vert \vert { }
\norm 729 \physicx_option_or:nnT { compat } { short }
\order 730 {
\oorder 731   \cs_set:Npn \eval { \evaluated }
\commutator 732   \cs_set:Npn \abs { \absolutevalue }
\poissonbracket 733 }
\pb 734 \physicx_declare_legacy_paren:NnnnNNn \norm
\anticommutator 735 { m } {#6} { } { } \lVert \rVert { }
\acomm 736 \physicx_compat:TF
737 {
738   \physicx_declare_legacy_paren:NnnnNNn \order
739   { m } {#6} { } { \c_physicx_Order_tl } { } { }
740 }
741 {
742   \physicx_declare_legacy_paren:NnnnNNn \order
743   { m } {#6} { } { \c_physicx_order_tl } { } { }
744 }
745 \physicx_declare_legacy_paren:NnnnNNn \commutator
746 { m m } { #6 , #7 } { } { } [ ] { }
747 \physicx_option_or:nnT { compat } { short }
748 { \cs_set:Npn \comm { \commutator } }
749 \physicx_declare_legacy_paren:NnnnNNn \poissonbracket

```

```

750 { m m } { #6 , #7 } { } \{ \} { }
751 \physicx_option_or:nnT { compat } { short }
752 {
753   \cs_set:Npn \pb { \poissonbracket }
754   \cs_set:Npn \anticommutator { \poissonbracket }
755   \cs_set:Npn \acomm { \poissonbracket }
756 }
757 \fi:
758 \physicx_declare_legacy_paren:NnnnNn \OOrder
759 { m } {#6} { \c_physicx_Order_tl } ( ) { }
760 \physicx_declare_legacy_paren:NnnnNn \oorder
761 { m } {#6} { \c_physicx_order_tl } ( ) { }

```

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

## 1.3 Matrix things

### 1.3.1 Matrix auxillary functions

```

762 \cs_new_nopar:Npn \__physicx_matrix_calc:nn #1#2
763 {
764   \int_set:Nn \l__physicx_matrix_rows_int
765     { \int_max:nn {#1} \l__physicx_matrix_rows_int }
766   \int_set:Nn \l__physicx_matrix_cols_int
767     { \int_max:nn {#2} \l__physicx_matrix_cols_int }
768 }
769 % use matrix element
770 \cs_new_nopar:Npn \physicx_matrix_use_r_c:nn #1#2
771 {
772   \if_cs_exist:w l__physicx_matrix_r@#1_c@#2_tl \cs_end:
773   \exp_not:v { l__physicx_matrix_r@#1_c@#2_tl }
774   \else:
775     \exp_not:o { \physicxempty }
776   \fi:
777 }
778 % set matrix element, check or not
779 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_nock:nnn #1#2
780 { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } }
781 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckig:nnn #1#2#3
782 {
783   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
784   { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
785 }
786 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckepe:nnn #1#2#3
787 {
788   \tl_if_empty:nTF {#3}
789     { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }
790     { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
791 }
792 \cs_new_nopar:Npn \__physicx_matrix_set_r_c_ckigep:nnn #1#2#3
793 {
794   \tl_if_eq:nnF {#3} { \PHYSICXIGNORE }
795   {
796     \tl_if_empty:nTF {#3}
797       { \tl_set:co { l__physicx_matrix_r@#1_c@#2_tl } { \physicxempty } }

```



```

798         { \tl_set:cn { l__physicx_matrix_r@#1_c@#2_tl } {#3} }
799     }
800 }
801 \cs_set_eq:NN \__physicx_matrix_set_r_c_ckall:nnn
802 \__physicx_matrix_set_r_c_ckigep:nnn
803 \cs_new_eq:NN \physicx_matrix_set_r_c:nnn
804 \__physicx_matrix_set_r_c_nock:nnn
805 % align, cr, sep symbol
806 \str_const:Nn \physicx@align { , }
807 \str_const:Nn \physicx@cr { ; }
808 \str_const:Nn \physicx@sep { , }
809 \bool_new:N \l__physicx_matrix_infinite_bool
810 \bool_new:N \l__physicx_matrix_dotrow_bool
811 \bool_new:N \l__physicx_matrix_dotcol_bool
812 \tl_new:N \l__physicx_matrix_array_tl
813 \tl_new:N \l__physicx_matrix_body_tl
814 \int_new:N \l__physicx_matrix_rows_int
815 \int_new:N \l__physicx_matrix_cols_int
816 \tl_new:N \l__physicx_matrix_main_tl
817 \clist_new:N \l__physicx_matrix_diag_clist
818 \clist_new:N \l__physicx_matrix_item_clist
819 \bool_new:N \l__physicx_matrix_diag_bool
820 \seq_new:N \l__physicx_row_list_seq
821 \seq_new:N \l__physicx_col_list_seq
822 % expand input
823 \cs_new_eq:NN \__physicx_expand:w \exp_not:o
824 %% main, row iterate, col iterate
825 \cs_new_nopar:Npn \physicx@matricelement #1#2#3 { #1 \sb { #2 #3 } }
826 \cs_new_nopar:Npn \__physicx_matrix_row_iterate:n #1 { #1 }
827 \tl_new:N \l__physicx_matrix_last_row_tl
828 \tl_new:N \l__physicx_matrix_last_col_tl
829 \cs_new_nopar:Npn \__physicx_matrix_col_iterate:n #1 { #1 }
830 \cs_new_nopar:Npn \__physicx_matrix_begin:w { }
831 \cs_new_nopar:Npn \__physicx_matrix_end:w { }
832 \cs_new_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn
833 \bool_new:N \l__physicx_matrix_expand_element_bool
834 % when element is empty use \physicxempty
835 \tl_new:N \physicxempty
836 % save 'element-except' key's value
837 \tl_new:N \physicxexcept
838 \tl_new:N \l__physicx_matrix_args_tl
839 \tl_new:N \l__physicx_matrix_after_begin_tl
840 \tl_new:N \l__physicx_matrix_after_end_tl
841 \bool_new:N \l__physicx_matrix_transpose_bool
842 \bool_new:N \l__physicx_matrix_enhanced_bool
843 \dim_new:N \l__physicx_matrix_sep_dim
844 \cs_new:Npn \__physicx_adi:nnn #1#2#3 { #1#2#3 }
845 \tl_new:N \l__physicx_matrix_beginning_tl
846 \tl_new:N \l__physicx_matrix_ending_tl

```

### 1.3.2 Matrix keys

```

847 \keys_define:nn { physicx }
848 { matrix .code:n = \keys_set:nn { physicx/matrix } {#1} }
849 \keys_define:nn { physicx/matrix }

```

```

850 {
851   array .tl_set:N = \l__physicx_matrix_array_tl ,
852   expand .choice: ,
853   expand / none .code:n =
854     \cs_set_eq:NN \__physicx_expand:w \exp_not:o ,
855   expand / text-expand .code:n =
856     \cs_set_eq:NN \__physicx_expand:w \text_expand:n ,
857   expand / f .code:n =
858     \cs_set_eq:NN \__physicx_expand:w \exp_not:f ,
859   expand / romanual .meta:n = { expand = f } ,
860   expand / x .code:n =
861     \cs_set_eq:NN \__physicx_expand:w \use:n ,
862   expand / edef .meta:n = { expand = x } ,
863   rows .int_set:N = \l__physicx_matrix_rows_int ,
864   cols .int_set:N = \l__physicx_matrix_cols_int ,
865   auto-update .choice: ,
866   auto-update / true .code:n =
867     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \__physicx_matrix_calc:nn ,
868   auto-update / false .code:n =
869     \cs_set_eq:NN \__physicx_matrix_autocalc:nn \use_none:nn ,
870   auto-update .default:n = true ,
871   main .tl_set:N = \l__physicx_matrix_main_tl ,
872   row-list .code:n =
873     \seq_set_split:Non \l__physicx_row_list_seq { \physicx@sep } {#1} ,
874   col-list .code:n =
875     \seq_set_split:Non \l__physicx_col_list_seq { \physicx@sep } {#1} ,
876   infinite .bool_set:N = \l__physicx_matrix_infinite_bool ,
877   infinite .default:n = true ,
878   !infinite .code:n =
879     \bool_set_inverse:N \l__physicx_matrix_infinite_bool ,
880   element-code .cs_set:Np = \physicx@matricelement #1#2#3 ,
881   element-code* .choice: ,
882   element-code* / except-empty .code:n =
883     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
884       \physicx@matricelement
885     \cs_set:Npn \physicx@matricelement ##1##2##3
886       {
887         \tl_if_empty:nTF {##1}
888           {##1}
889           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
890       } ,
891   element-code* / except-blank .code:n =
892     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
893       \physicx@matricelement
894     \cs_set:Npn \physicx@matricelement ##1##2##3
895       {
896         \tl_if_blank:nTF {##1}
897           {##1}
898           { \__physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
899       } ,
900   element-code* / except-dots .code:n =
901     \cs_set_eq:NN \__physicx_matrix_element_aux:nnn
902       \physicx@matricelement
903     \cs_set:Npn \physicx@matricelement ##1##2##3

```

```

904     {
905         \tl_if_in:nnTF { \cdots\vdots\ldots\ddots } {##1}
906         {##1}
907         { \_physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
908     } ,
909 element-code* / except-tl .code:n =
910     \cs_set_eq:NN \_physicx_matrix_element_aux:nnn
911     \physicx@matricelement
912     \cs_set:Npn \physicx@matricelement ##1##2##3
913     {
914         \tl_if_in:onTF { \physicxexcept } {##1}
915         {##1}
916         { \_physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
917     } ,
918 element-code* / except-regex .code:n =
919     \cs_set_eq:NN \_physicx_matrix_element_aux:nnn
920     \physicx@matricelement
921     \cs_set:Npn \physicx@matricelement ##1##2##3
922     {
923         \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
924         {##1}
925         { \_physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
926     } ,
927 element-code* / only-regex .code:n =
928     \cs_set_eq:NN \_physicx_matrix_element_aux:nnn
929     \physicx@matricelement
930     \cs_set:Npn \physicx@matricelement ##1##2##3
931     {
932         \exp_args:No \regex_match:nnTF { \physicxexcept } {##1}
933         { \_physicx_matrix_element_aux:nnn {##1} {##2} {##3} }
934         {##1}
935     } ,
936 element-code* / unknown .code:n =
937     \cs_set:Npx \physicx@matricelement { \exp_not:c {#1} } ,
938 element-except .tl_set:N = \physicxexcept ,
939 element-except+ .code:n =
940     \tl_put_right:Nn \physicxexcept {#1} ,
941 expand-element .bool_set:N = \l__physicx_matrix_expand_element_bool ,
942 expand-element .default:n = true ,
943 empty .tl_set:N = \physicxempty ,
944 check .choice: ,
945 check / none .code:n =
946     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
947     \_physicx_matrix_set_r_c_nock:nnn ,
948 check / empty .code:n =
949     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
950     \_physicx_matrix_set_r_c_ckepp:nnn ,
951 check / ignore .code:n =
952     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
953     \_physicx_matrix_set_r_c_ckig:nnn ,
954 check / igep .code:n =
955     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
956     \_physicx_matrix_set_r_c_ckigep:nnn ,
957 check / all .code:n =

```

```

958     \cs_set_eq:NN \physicx_matrix_set_r_c:nnn
959     \__physicx_matrix_set_r_c_ckall:nnn ,
960     check .default:n = all ,
961     row-iterate .cs_set:Np = \__physicx_matrix_row_iterate:n #1 ,
962     col-iterate .cs_set:Np = \__physicx_matrix_col_iterate:n #1 ,
963     last-row .tl_set:N = \l__physicx_matrix_last_row_tl ,
964     last-col .tl_set:N = \l__physicx_matrix_last_col_tl ,
965     diag .clist_set:N = \l__physicx_matrix_diag_clist ,
966     diag+ .code:n =
967         \clist_put_right:Nn \l__physicx_matrix_diag_clist {#1} ,
968     diag-now .code:n = \physicx_matrix_diag_parse:n {#1} ,
969     diag-data .code:n = \__physicx_matrix_set_data:nn { diag } {#1} ,
970     diag-data+ .code:n = \__physicx_matrix_add_data:nn { diag } {#1} ,
971     item .clist_set:N = \l__physicx_matrix_item_clist ,
972     item+ .code:n =
973         \clist_put_right:Nn \l__physicx_matrix_item_clist {#1} ,
974     item-now .code:n = \physicx_matrix_item_parse:n {#1} ,
975     item-data .code:n = \__physicx_matrix_set_data:nn { item } {#1} ,
976     item-data+ .code:n = \__physicx_matrix_add_data:nn { item } {#1} ,
977     check-range .choice: ,
978     check-range / true .code:n = \physicx_parse_range_check: ,
979     check-range / false .code:n = \physicx_parse_range_nocheck: ,
980     check-range .default:n = true ,
981     begin .tl_set:N = \__physicx_matrix_begin:w ,
982     end .tl_set:N = \__physicx_matrix_end: ,
983     args .code:n =
984         \tl_set:Nn \l__physicx_matrix_args_tl { [#1] } ,
985     args* .tl_set:N = \l__physicx_matrix_args_tl ,
986     after-begin .tl_set:N = \l__physicx_matrix_after_begin_tl ,
987     after-begin+ .code:n =
988         { \tl_put_right:Nn \l__physicx_matrix_after_begin_tl {#1} } ,
989     after-end .tl_set:N = \l__physicx_matrix_after_end_tl ,
990     after-end+ .code:n =
991         { \tl_put_right:Nn \l__physicx_matrix_after_end_tl {#1} } ,
992     sepdim .dim_set:N = \l__physicx_matrix_sepdim ,
993     type .multichoice: ,
994     saveto .tl_set:N = \l__physicx_matrix_save_tl ,
995     saveto* .code:n =
996         \tl_set:Nn \l__physicx_matrix_save_tl { \cs:w #1 \cs_end: } ,
997     transpose .bool_set:N = \l__physicx_matrix_transpose_bool ,
998     transpose .default:n = true ,
999     ' .meta:n = { transpose = true } ,
1000     T .meta:n = { transpose = true } ,
1001     MaxMatrixCols .int_set:N = \c@MaxMatrixCols ,
1002     enhanced .bool_set:N = \l__physicx_matrix_enhanced_bool ,
1003     enhanced .default:n = true ,
1004     !enhanced .code:n =
1005         \bool_set_inverse:N \l__physicx_matrix_enhanced_bool ,
1006     cr .tl_set:N = \physicx@cr ,
1007     align .tl_set:N = \physicx@align ,
1008     sep .tl_set:N = \physicx@sep ,
1009     adi-order .choice: ,
1010     adi-order / adi .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##2##3} ,
1011     adi-order / dia .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##3##1} ,

```

```

1012 adi-order / iad .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##1##2} ,
1013 adi-order / aid .code:n = \cs_set:Nn \__physicx_adi:nnn {##1##3##2} ,
1014 adi-order / ida .code:n = \cs_set:Nn \__physicx_adi:nnn {##3##2##1} ,
1015 adi-order / dai .code:n = \cs_set:Nn \__physicx_adi:nnn {##2##1##3} ,
1016 beginning .tl_set:N = \l__physicx_matrix_beginning_tl ,
1017 beginning+ .code:n =
1018   \tl_put_right:Nn \l__physicx_matrix_beginning_tl {#1} ,
1019 ending .tl_set:N = \l__physicx_matrix_ending_tl ,
1020 ending+ .code:n =
1021   \tl_put_right:Nn \l__physicx_matrix_ending_tl {#1} ,
1022
1023 settype .code:n = \setmatritxttype #1 ,
1024
1025 unknown .code:n =
1026   \physicx_search_also:nnF
1027   {
1028     physicx/matrix/type ,
1029     physicx/matrix/expand ,
1030     physicx/matrix/element-code* ,
1031   }
1032   {#1}
1033   {
1034     \exp_args:No \physicx_if_num:nTF { \l_keys_key_str }
1035     {
1036       \keys_set:nx { physicx/matrix }
1037       { MaxMatrixCols = \l_keys_key_str }
1038     }
1039     {
1040       \msg_error:nnxx { physicx } { unknown-key }
1041       \l_keys_path_str { physicx/matrix }
1042     }
1043   } ,
1044 }

```

```

\physicx_matrix_new_type:nnn
\physicx_matrix_new_type:nn
\setmatritxttype
1045 \cs_new:Npn \physicx_matrix_new_type:nnn #1#2#3
1046 { \physicx_new_type:nnn { matrix } {#1} { begin={#2} , end={#3} } }
1047 \cs_new:Npn \physicx_matrix_new_type:nn
1048 { \physicx_new_type:nnn { matrix } }
1049 \NewDocumentCommand \setmatritxttype { s >{ \TrimSpaces } m }
1050 {
1051   \IfBooleanTF {#1}
1052   { \physicx_matrix_new_type:nn {#2} }
1053   { \physicx_matrix_new_type:nnn {#2} }
1054 }

```

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

A few types.

```

1055 \setmatritxttype {m} {\begin{matrix}} {\end{matrix}}
1056 \setmatritxttype {p} {\begin{pmatrix}} {\end{pmatrix}}
1057 \setmatritxttype {b} {\begin{bmatrix}} {\end{bmatrix}}
1058 \setmatritxttype {B} {\begin{Bmatrix}} {\end{Bmatrix}}
1059 \setmatritxttype {v} {\begin{vmatrix}} {\end{vmatrix}}

```

```

1060 \setmatrixtype {V} {\begin{Vmatrix}} {\end{Vmatrix}}
1061 \setmatrixtype {sm} {\begin{smallmatrix}} {\end{smallmatrix}}
1062 \physics_mathtools:T
1063 {
1064   \setmatrixtype {m*} {\begin{matrix*}} {\end{matrix*}}
1065   \setmatrixtype {p*} {\begin{pmatrix*}} {\end{pmatrix*}}
1066   \setmatrixtype {b*} {\begin{bmatrix*}} {\end{bmatrix*}}
1067   \setmatrixtype {B*} {\begin{Bmatrix*}} {\end{Bmatrix*}}
1068   \setmatrixtype {v*} {\begin{vmatrix*}} {\end{vmatrix*}}
1069   \setmatrixtype {V*} {\begin{Vmatrix*}} {\end{Vmatrix*}}
1070   \setmatrixtype {sm*} {\begin{smallmatrix*}} {\end{smallmatrix*}}
1071   \setmatrixtype {sp} {\begin{psmallmatrix}} {\end{psmallmatrix}}
1072   \setmatrixtype {sb} {\begin{bsmallmatrix}} {\end{bsmallmatrix}}
1073   \setmatrixtype {sB} {\begin{Bsmallmatrix}} {\end{Bsmallmatrix}}
1074   \setmatrixtype {sv} {\begin{vsmallmatrix}} {\end{vsmallmatrix}}
1075   \setmatrixtype {sV} {\begin{Vsmallmatrix}} {\end{Vsmallmatrix}}
1076   \setmatrixtype {sp*} {\begin{psmallmatrix*}} {\end{psmallmatrix*}}
1077   \setmatrixtype {sb*} {\begin{bsmallmatrix*}} {\end{bsmallmatrix*}}
1078   \setmatrixtype {sB*} {\begin{Bsmallmatrix*}} {\end{Bsmallmatrix*}}
1079   \setmatrixtype {sv*} {\begin{vsmallmatrix*}} {\end{vsmallmatrix*}}
1080   \setmatrixtype {sV*} {\begin{Vsmallmatrix*}} {\end{Vsmallmatrix*}}
1081 }

```

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

```

1082 \cs_new_protected_nopar:Npn \setmatrixdata #1#2
1083 { \clist_set:cn { physics@ #1 data@ #2 } }
1084 \cs_new_protected_nopar:Npn \__physics_matrix_set_data:nn #1#2
1085 {
1086   \clist_clear:c { l__physics_matrix_ #1 _clist }
1087   \__physics_matrix_add_data:nn {#1} {#2}
1088 }
1089 \cs_new_protected_nopar:Npn \__physics_matrix_add_data:nn #1#2
1090 {
1091   \clist_map_inline:nn {#2}
1092   {
1093     \clist_concat:ccc
1094     { l__physics_matrix_ #1 _clist }
1095     { l__physics_matrix_ #1 _clist }
1096     { physics@ #1 data@ #2 }
1097   }
1098 }

```

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

Initial settings.

```

1099 \keys_set:nn { physics/matrix }
1100 {
1101   type = m ,
1102   saveto = ? ,
1103 }

```

`\qxmatrix`

```

1104 %% basically, https://tex.stackexchange.com/questions/486154/is-there-a-way-to-define-
1105 xmatnm-in-the-physics-package, but changed some
1106 % #1 = boolean, saveto matrix

```

```

1106 % #2 = star, infinite
1107 % #3 = options
1108 % #4 = letter for the entries
1109 % #5 = number of rows
1110 % #6 = number of explicit rows, default = 3
1111 % #7 = number of columns
1112 % #8 = number of explicit columns, default = 3
1113 \DeclareDocumentCommand \qmatrix { t= s 0{type=p} m m 0{3} m 0{3} }
1114 {
1115   \group_begin:
1116   \IfBooleanTF { #2 }
1117     { \bool_set_true:N \l__physicx_matrix_infinite_bool }
1118     { \bool_set_false:N \l__physicx_matrix_infinite_bool }
1119   \int_set:Nn \l__physicx_matrix_rows_int {#6}
1120   \int_set:Nn \l__physicx_matrix_cols_int {#8}
1121   \IfBooleanTF {#1}
1122     { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicx_tmp } }
1123     { \keys_set:nn { physicx/matrix } {#3} }
1124   \physicx_qmatrix:nnn {#4} {#5} {#7}
1125   \__physicx_matrix_save_or_print:
1126   \group_end:
1127 }
1128 \cs_new_protected:Nn \physicx_qmatrix:nnn
1129 {
1130   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1131     {
1132       \cs_set_eq:NN \__physicx_qmatrix_appto_body:nnn
1133       \__physicx_matrix_appto_body_e:nnn
1134     }
1135     {
1136       \cs_set_eq:NN \__physicx_qmatrix_appto_body:nnn
1137       \__physicx_matrix_appto_body_ne:nnn
1138     }
1139   % clear the variable containing the body of the matrix
1140   \tl_clear:N \l__physicx_matrix_body_tl
1141   % set the tentative number of explicit rows
1142   \physicx_if_num:nTF { #2 }
1143     {% number of rows is an integer
1144       \int_compare:nTF { #2 <= \l__physicx_matrix_rows_int }
1145       {% if #2 <= rows, we don't want a row of dots
1146         \bool_set_false:N \l__physicx_matrix_dotrow_bool
1147         \int_set:Nn \l__physicx_matrix_rows_int { #2 }
1148       }
1149       {% we want a row of dots
1150         \bool_set_true:N \l__physicx_matrix_dotrow_bool
1151       }
1152     }
1153     {% number of rows is symbolic, we want a row of dots
1154       \bool_set_true:N \l__physicx_matrix_dotrow_bool
1155     }
1156   % set the tentative number of explicit columns
1157   \physicx_if_num:nTF { #3 }
1158     {% number of cols is an integer
1159       \int_compare:nTF { #3 <= \l__physicx_matrix_cols_int }

```

```

1160         {% if #3 <= cols, we don't want a column of dots
1161             \bool_set_false:N \l__physicx_matrix_dotcol_bool
1162             \int_set:Nn \l__physicx_matrix_cols_int { #3 }
1163         }
1164         {% we want a column of dots
1165             \bool_set_true:N \l__physicx_matrix_dotcol_bool
1166         }
1167     }
1168     {% number of columns is symbolic, we want a column of dots
1169         \bool_set_true:N \l__physicx_matrix_dotcol_bool
1170     }
1171     % loop through the rows
1172     \int_step_inline:nn { \l__physicx_matrix_rows_int }
1173     {
1174         % add the first entry in the row
1175         %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{##1 1} }
1176         \__physicx_qxmatrix_appto_body:nnn {#1} {##1} { 1 }
1177         % add the further entries in the explicit columns
1178         \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1179         {
1180             %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{##1 ####1} }
1181             \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1182             \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {####1}
1183         }
1184         % if we have a column of dots, add \cdots and the last entry
1185         \bool_if:NT \l__physicx_matrix_dotcol_bool
1186         {
1187             %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{##1 #3} }
1188             \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1189             \__physicx_qxmatrix_appto_body:nnn {#1} {##1} {#3}
1190         }
1191         % infinite matrix, add \cdots
1192         \bool_if:NT \l__physicx_matrix_infinite_bool
1193         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1194         \if_int_compare:w ##1 = \l__physicx_matrix_rows_int
1195             \scan_stop:
1196         \else:
1197             % finish up the row
1198             \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1199         \fi:
1200     }
1201     % finish up the rows
1202     \bool_if:NT \l__physicx_matrix_dotrow_bool
1203     {
1204         % finish up the row
1205         \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1206         % if we have a row of dots, fill it in
1207         \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1208         \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1209         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1210         \bool_if:NT \l__physicx_matrix_dotcol_bool
1211         { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots & \vdots } }
1212         \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1213         % fill the last row

```



```

1214      %%\tl_put_right:Nn \l__physicx_matrix_body_tl { #1\sb{#2 1} }
1215      \__physicx_qxmatrix_appto_body:nnn {#1} {#2} { 1 }
1216      \int_step_inline:nnn { 2 } { \l__physicx_matrix_cols_int }
1217      {
1218        %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & #1\sb{#2 ##1} }
1219        \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1220        \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {##1}
1221      }
1222      \bool_if:NT \l__physicx_matrix_dotcol_bool
1223      {
1224        %%\tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & #1\sb{#2 #3} }
1225        \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots & }
1226        \__physicx_qxmatrix_appto_body:nnn {#1} {#2} {#3}
1227      }
1228      % if the matrix is infinite, add a further column with \cdots
1229      \bool_if:NT \l__physicx_matrix_infinite_bool
1230      { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \cdots } }
1231    }
1232    % if the matrix is infinite, add a final row
1233    \bool_if:NT \l__physicx_matrix_infinite_bool
1234    {
1235      % finish up the row
1236      \tl_put_right:Nx \l__physicx_matrix_body_tl { \__physicx_matrix_sep: }
1237      \tl_put_right:Nn \l__physicx_matrix_body_tl { \vdots }
1238      \prg_replicate:nn { \l__physicx_matrix_cols_int - 1 }
1239      { \tl_put_right:Nn \l__physicx_matrix_body_tl { & \vdots } }
1240      \bool_if:NT \l__physicx_matrix_dotcol_bool
1241      { \tl_put_right:Nn \l__physicx_matrix_body_tl { & & \vdots } }
1242      \tl_put_right:Nn \l__physicx_matrix_body_tl { & \ddots }
1243      % update cols
1244      \bool_if:NTF \l__physicx_matrix_dotcol_bool
1245      { \tex_advance:D \l__physicx_matrix_cols_int by 3 }
1246      { \tex_advance:D \l__physicx_matrix_cols_int by 2 }
1247    }
1248  }

```

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

```

\physicx_matrix_diag_parse:n Parse 'diag...' keys.
\physicx_matrix_diag_parse:o
1249 \cs_new:Npn \physicx_matrix_diag_parse:n #1
1250 {
1251   \keyval_parse:nnn
1252   \__physicx_matrix_diag_parse_aux:n
1253   \__physicx_matrix_diag_parse_aux:nn
1254   {#1}
1255 }
1256 \cs_generate_variant:Nn \physicx_matrix_diag_parse:n { o }
1257 \cs_new:Npn \__physicx_matrix_diag_parse_aux:n #1
1258 {
1259   \str_case_e:nnF {#1}
1260   {
1261     { auto-update }
1262     {
1263       \cs_set_eq:NN \__physicx_matrix_diag_calc:nn

```

```

1264         \__physicx_matrix_calc:nn
1265     }
1266     { noauto-update }
1267     {
1268         \cs_set_eq:NN \__physicx_matrix_diag_calc:nn \use_none:nn
1269     }
1270     { true }
1271     {
1272         \bool_set_true:N \l__physicx_matrix_diag_bool
1273         \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1274             \__physicx_diagonalmatrix_set_diag:
1275     }
1276     { false }
1277     {
1278         \bool_set_false:N \l__physicx_matrix_diag_bool
1279         \cs_set_eq:NN \__physicx_diagonalmatrix_diag_main:
1280             \__physicx_diagonalmatrix_no_diag:
1281     }
1282     }
1283     { \msg_error:nnn { physicx } { diag-key } {#1} }
1284 }
1285 \cs_new:Npn \__physicx_matrix_diag_parse_aux:nn #1#2
1286 {
1287     \tl_set:Nn \l__physicx_tmpdiag_tl {#2}
1288     \tl_set:Nx \l__physicx_tmpdiag_tl
1289         { \__physicx_expand:w \l__physicx_tmpdiag_tl }
1290     \seq_set_split:NVV \l__physicx_tmpdiag_seq \physicx@sep \l__physicx_tmpdiag_tl
1291     \tl_if_head_eq_charcode:nNTF {#1} '
1292     {
1293         \exp_args:Nf \__physicx_matrix_diag_parse_aux_anti:n
1294             { \tl_tail:n {#1} }
1295     }
1296     { \__physicx_matrix_diag_parse_aux_regu:n {#1} }
1297 }
1298 \cs_new:Npn \__physicx_diagonalmatrix_set_diag:
1299 {
1300     \int_zero:N \l__physicx_matrix_cols_int
1301     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1302     {
1303         \int_incr:N \l__physicx_matrix_cols_int
1304         \physicx_matrix_set_r_c:nnn {##1} {##1} {##2}
1305     }
1306     \int_set_eq:NN \l__physicx_matrix_rows_int
1307         \l__physicx_matrix_cols_int
1308 }
1309 \cs_new:Npn \__physicx_diagonalmatrix_no_diag:
1310 {
1311     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1312     { \physicx_matrix_set_r_c:nnn {##1} {##1} {##2} }
1313     \__physicx_matrix_diag_calc:nn
1314     { \seq_count:N \l__physicx_tmpdiag_seq }
1315     { \seq_count:N \l__physicx_tmpdiag_seq }
1316 }
1317 \cs_new_eq:NN \__physicx_diagonalmatrix_diag_main:

```

```

1318 \__physicx_diagonalmatrix_no_diag:
1319 \cs_new:Npn \__physicx_matrix_diag_parse_aux_regu:n #1
1320 {
1321   \if_int_compare:w #1 = 0 \exp_stop_f:
1322     \__physicx_diagonalmatrix_diag_main:
1323   \else:
1324     \if_int_compare:w #1 > 0 \exp_stop_f:
1325       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1326       {
1327         \physicx_matrix_set_r_c:nnn
1328         {##1} { \int_eval:n { ##1 + #1 } } {##2}
1329       }
1330     \__physicx_matrix_diag_calc:nn
1331     { \seq_count:N \l__physicx_tmpdiag_seq }
1332     { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1333   \else:
1334     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1335     {
1336       \physicx_matrix_set_r_c:nnn
1337       { \int_eval:n { ##1 - #1 } } {##1} {##2}
1338     }
1339     \__physicx_matrix_diag_calc:nn
1340     { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1341     { \seq_count:N \l__physicx_tmpdiag_seq }
1342   \fi:
1343 \fi:
1344 }
1345 \cs_new:Npn \__physicx_matrix_diag_parse_aux_anti:n #1
1346 {
1347   \if_int_compare:w #1 = 0 \exp_stop_f:
1348     \__physicx_matrix_diag_calc:nn
1349     { \seq_count:N \l__physicx_tmpdiag_seq }
1350     { \seq_count:N \l__physicx_tmpdiag_seq }
1351     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1352     {
1353       \physicx_matrix_set_r_c:nnn
1354       {##1}
1355       { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1356       {##2}
1357     }
1358   \else:
1359     \if_int_compare:w #1 > 0 \exp_stop_f:
1360       \__physicx_matrix_diag_calc:nn
1361       { \seq_count:N \l__physicx_tmpdiag_seq }
1362       { \seq_count:N \l__physicx_tmpdiag_seq + #1 }
1363       \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1364       {
1365         \physicx_matrix_set_r_c:nnn
1366         {##1}
1367         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 - #1 + 1 } }
1368         {##2}
1369       }
1370     \else:
1371       \__physicx_matrix_diag_calc:nn

```

```

1372         { \seq_count:N \l__physicx_tmpdiag_seq - #1 }
1373         { \seq_count:N \l__physicx_tmpdiag_seq }
1374     \seq_map_indexed_inline:Nn \l__physicx_tmpdiag_seq
1375     {
1376         \physicx_matrix_set_r_c:nnn
1377         { \int_eval:n { ##1 - #1 } }
1378         { \int_eval:n { \l__physicx_matrix_cols_int - ##1 + 1 } }
1379         {##2}
1380     }
1381     \fi:
1382     \fi:
1383 }
1384 \cs_new:Npn \__physicx_matrix_diag_calc:nn
1385 { \__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
1386 \cs_new:Npn \physicx_matrix_item_parse:n #1
1387 {
1388     \clist_set_eq:NN \l__physicx_item_ignore_clist \c_empty_clist
1389     \keyval_parse:NNn
1390     \__physicx_matrix_item_parse_aux:n
1391     \__physicx_matrix_item_parse_aux:nn
1392     {#1}
1393 }
1394 \cs_generate_variant:Nn \physicx_matrix_item_parse:n { o }
1395 \cs_new:Npn \__physicx_matrix_item_parse_aux:n #1 { }
1396 \cs_new:Npn \__physicx_matrix_item_parse_aux:nn #1#2
1397 {
1398     \tl_set:Nn \l__physicx_tmpitem_tl {#2}
1399     \tl_set:Nx \l__physicx_tmpitem_tl
1400     { \__physicx_expand:w \l__physicx_tmpitem_tl }
1401     \physicx_parse_range:neN \l__physicx_matrix_rows_int
1402     { \use_i:nn #1 } \l__physicx_tmp_rownum_seq
1403     \physicx_parse_range:neN \l__physicx_matrix_cols_int
1404     { \use_ii:nn #1 } \l__physicx_tmp_colnum_seq
1405     \exp_args:No \tl_if_eq:nnTF
1406     { \l__physicx_tmpitem_tl } { \PHYSICXIGNORE }
1407     {
1408         \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1409         {
1410             \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1411             {
1412                 \clist_put_right:Nn \l__physicx_item_ignore_clist { [##1][####1] }
1413             }
1414         }
1415     }
1416     {
1417         \seq_map_inline:Nn \l__physicx_tmp_rownum_seq
1418         {
1419             \seq_map_inline:Nn \l__physicx_tmp_colnum_seq
1420             {
1421                 \clist_if_in:NnF \l__physicx_item_ignore_clist { [##1][####1] }

```

```

1422         {
1423             \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1424             {##1} {####1} { \l__physicx_tmpitem_tl }
1425         }
1426     }
1427 }
1428 }
1429 }

```

(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
1430 \cs_new:Npn \physicx_matrix_array_parse:n #1
1431 {
1432     \tl_set:Nn \l__physicx_tmparr_tl {#1}
1433     \tl_set:Nx \l__physicx_tmparr_tl
1434     { \__physicx_expand:w \l__physicx_tmparr_tl }
1435     \seq_set_split:NVV \l__physicx_matrix_tmparr_r_sep \physicx@cr \l__physicx_tmparr_tl
1436     \__physicx_matrix_autocalc:nn
1437     { \seq_count:N \l__physicx_matrix_tmparr_r_sep }
1438     { 0 }
1439     \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_r_sep
1440     {
1441         \seq_set_split:Non \l__physicx_matrix_tmparr_c_sep { \physicx@align } {##2}
1442         \__physicx_matrix_autocalc:nn
1443         { 0 }
1444         { \seq_count:N \l__physicx_matrix_tmparr_c_sep }
1445         \seq_map_indexed_inline:Nn \l__physicx_matrix_tmparr_c_sep
1446         {
1447             \physicx_matrix_set_r_c:nnn {##1} {####1} {####2}
1448         }
1449     }
1450 }
1451 \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.

```

1452 \cs_new:Npn \physicx_matrix_array_parse_main:
1453 {
1454     \int_step_inline:nn \l__physicx_matrix_rows_int
1455     {
1456         \int_step_inline:nn \l__physicx_matrix_cols_int
1457         {
1458             \exp_args:Nnno \physicx_matrix_set_r_c:nnn
1459             {##1} {####1} \l__physicx_matrix_main_tl
1460         }
1461     }
1462 }

```

(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).

```

1463 \prg_new_conditional:Npnn \__physicx_if_can_num:n #1 { T, F, TF }

```

```

1464 {
1465   \physicx_if_num:nTF {#1}
1466   { \prg_return_true: }
1467   {
1468     \bool_case_true:nTF
1469     {
1470       { \tl_if_head_eq_meaning_p:nN {#1} \int_eval:n } { }
1471       { \tl_if_head_eq_meaning_p:nN {#1} \fp_eval:n } { }
1472       {
1473         \bool_lazy_and_p:nn
1474         { \cs_if_exist_p:N \inteval }
1475         { \tl_if_head_eq_meaning_p:nN {#1} \inteval }
1476       } { }
1477       {
1478         \bool_lazy_and_p:nn
1479         { \cs_if_exist_p:N \fp eval }
1480         { \tl_if_head_eq_meaning_p:nN {#1} \fp eval }
1481       } { }
1482     }
1483     { \prg_return_true: }
1484     { \prg_return_false: }
1485   }
1486 }

```

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

\diagonalmatrix Define \diagonalmatrix.

```

1487 \DeclareDocumentCommand \diagonalmatrix { t= t+ 0{ } m }
1488 {
1489   \group_begin:
1490   \IfBooleanTF {#1}
1491   { \keys_set:nn { physicx/matrix } { #3 , saveto = \physicxtmp } }
1492   { \keys_set:nn { physicx/matrix } { #3 } }
1493   \physicx_construct:nnn { }
1494   {
1495     \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist
1496     \tl_if_empty:nF {#4}
1497     {
1498       \__physicx_if_keyval:nTF {#4}
1499       { \physicx_matrix_diag_parse:n { true, #4 } }
1500       { \physicx_matrix_diag_parse:n { true, 0 = {#4} } }
1501     }
1502   }
1503   { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1504   \bool_lazy_or:nnTF
1505   { \bool_if_p:n {#2} }
1506   { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1507   {
1508     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1509     {
1510       \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1511       \__physicx_matrix_appto_body_e:off
1512     }
1513   }

```

```

1514         \cs_set_eq:NN \__physicx_diagonalmatrix_enhanced:nnn
1515         \__physicx_matrix_appto_body_ne:off
1516     }
1517     \use_i_ii:nnn
1518 }
1519 { \use_i:nn }
1520 \__physicx_matrix_transpose:N
1521 \__physicx_diagonalmatrix_generate_enhanced_body:NNN
1522 \__physicx_diagonalmatrix_generate_body:NNN
1523 \__physicx_matrix_save_or_print:
1524 \group_end:
1525 }
1526 \cs_new:Npn \__physicx_diagonalmatrix_generate_enhanced_body:NNN #1#2#3
1527 {
1528     \__physicx_matrix_generate_body:NNNN #1#2#3
1529     \__physicx_diagonalmatrix_enhanced:nnn
1530 }
1531 \cs_new:Npn \__physicx_diagonalmatrix_generate_body:NNN #1#2#3
1532 {
1533     \int_step_inline:nn { #1 - 1 }
1534     {
1535         \int_step_inline:nn { #2 - 1 }
1536         {
1537             \tl_put_right:Nx \l__physicx_matrix_body_tl
1538             {
1539                 \exp_after:wN
1540                 \physicx_matrix_use_r_c:nn
1541                 #3 {{##1}} {{####1}} &
1542             }
1543         }
1544         \tl_put_right:Nx \l__physicx_matrix_body_tl
1545         {
1546             \exp_after:wN
1547             \physicx_matrix_use_r_c:nn
1548             #3 {{##1}} {{ \int_use:N #2 }} \__physicx_matrix_sep:
1549         }
1550     }
1551     \int_step_inline:nn { #2 - 1 }
1552     {
1553         \tl_put_right:Nx \l__physicx_matrix_body_tl
1554         {
1555             \exp_after:wN
1556             \physicx_matrix_use_r_c:nn
1557             #3 {{ \int_use:N #1 }} {{##1}} &
1558         }
1559     }
1560     \tl_put_right:Nx \l__physicx_matrix_body_tl
1561     {
1562         \exp_after:wN
1563         \physicx_matrix_use_r_c:nn
1564         #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1565     }
1566 }

```

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

\\_physicx\_declare\_init:

```

1567 \cs_new:Npn \_physicx_matrix_enhanced_init:
1568 {
1569   \seq_if_empty:NF \l__physicx_row_list_seq
1570   {
1571     \bool_set_true:N \l__physicx_matrix_expand_element_bool
1572     \cs_set_nopar:Npn \_physicx_matrix_row_iterate:n ##1
1573       { \seq_item:Nn \l__physicx_row_list_seq {##1} }
1574   }
1575   \seq_if_empty:NF \l__physicx_col_list_seq
1576   {
1577     \bool_set_true:N \l__physicx_matrix_expand_element_bool
1578     \cs_set_nopar:Npn \_physicx_matrix_col_iterate:n ##1
1579       { \seq_item:Nn \l__physicx_col_list_seq {##1} }
1580   }
1581 }

```

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

\commamatrix Define \commamatrix.

```

1582 \DeclareDocumentCommand \commamatrix { t= t+ 0{ } m }
1583 {
1584   \group_begin:
1585   \keys_set:nn { physicx/matrix } {#3}
1586   \tl_if_empty:NF {#4}
1587     { \keys_set:nn { physicx/matrix } { array = {#4} } }
1588   \IfBooleanT {#1}
1589     { \keys_set:nn { physicx/matrix } { saveto = \physicx_tmp } }
1590   \tl_set:Nx \l__physicx_matrix_array_tl
1591     { \_physicx_expand:w \l__physicx_matrix_array_tl }
1592   \bool_lazy_or:nnTF
1593     { \bool_if_p:n {#2} }
1594     { \bool_if_p:N \l__physicx_matrix_enhanced_bool }
1595     { \_physicx_commamatrix_enhanced: }
1596   {
1597     \tl_replace_all:Nox \l__physicx_matrix_array_tl
1598       { \physicx@cr } { \_physicx_matrix_sep: }
1599     \tl_replace_all:Non \l__physicx_matrix_array_tl
1600       { \physicx@align } { & }
1601     \tl_set_eq:NN \l__physicx_matrix_body_tl
1602       \l__physicx_matrix_array_tl
1603   }
1604   \_physicx_matrix_save_or_print:
1605   \group_end:
1606 }
1607 \cs_new_nopar:Npn \_physicx_matrix_save_or_print:
1608 {
1609   \exp_after:wN \token_if_cs:NTF \l__physicx_matrix_save_tl
1610   {
1611     \exp_after:wN \tl_gset_eq:NN
1612       \l__physicx_matrix_save_tl
1613       \l__physicx_matrix_body_tl
1614   }
1615   {

```



```

1616         \if_int_compare:w \c@MaxMatrixCols < \l__physicx_matrix_cols_int
1617         \int_set_eq:NN \c@MaxMatrixCols \l__physicx_matrix_cols_int
1618         \fi:
1619         \exp_after:wN \__physicx_matrix_begin:w \l__physicx_matrix_args_tl \l__physicx_matrix_body_tl
1620         \l__physicx_matrix_body_tl
1621         \__physicx_matrix_end: \l__physicx_matrix_after_end_tl
1622     }
1623 }
1624 \cs_new:Npn \__physicx_commamatrix_enhanced:
1625 {
1626     \tl_clear:N \l__physicx_matrix_body_tl
1627     \int_zero:N \l__physicx_tmpa_int
1628     \seq_set_split:NVV \l__physicx_tmp_seq \physicx@cr
1629     \l__physicx_matrix_array_tl
1630     \int_set:Nn \l__physicx_matrix_rows_int
1631     { \seq_count:N \l__physicx_tmp_seq }
1632     \__physicx_matrix_enhanced_init:
1633     \bool_if:NTF \l__physicx_matrix_expand_element_bool
1634     {
1635         \seq_map_tokens:Nn \l__physicx_tmp_seq
1636         {
1637             \int_incr:N \l__physicx_tmpa_int
1638             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1639             \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_e:nnn
1640         }
1641     }
1642     {
1643         \seq_map_tokens:Nn \l__physicx_tmp_seq
1644         {
1645             \int_incr:N \l__physicx_tmpa_int
1646             \exp_args:NV \__physicx_commamatrix_enhanced_aux:nNn
1647             \l__physicx_tmpa_int \__physicx_commamatrix_enhanced_aux_ne:nnn
1648         }
1649     }
1650 }
1651 \cs_new:Npn \__physicx_commamatrix_enhanced_aux:nNn #1#2#3
1652 {
1653     \seq_set_split:Non \l__physicx_tmp_col_seq
1654     { \physicx@align } {#3}
1655     \seq_set_eq:NN \l__physicx_tmp_coled_seq \c_empty_seq
1656     \seq_map_indexed_inline:Nn \l__physicx_tmp_col_seq
1657     { #2 {##2} {#1} {##1} }
1658     \tl_put_right:Nx \l__physicx_matrix_body_tl
1659     {
1660         \seq_use:Nn \l__physicx_tmp_coled_seq { & }
1661         \if_int_compare:w \l__physicx_matrix_rows_int = #1
1662         \scan_stop:
1663         \else:
1664             \__physicx_matrix_sep:
1665         \fi:
1666     }
1667 }
1668 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_e:nnn #1#2#3
1669 {

```

```

1670 \seq_put_right:Nx \l__physicx_tmp_coled_seq
1671 {
1672   \text_expand:n % \text_expand:n do the magic thing, but slower
1673   {
1674     \physicx@matricelement { #1 }
1675     { \__physicx_matrix_row_iterate:n {#2} }
1676     { \__physicx_matrix_col_iterate:n {#3} }
1677   }
1678 }
1679 }
1680 \cs_new:Npn \__physicx_commamatrix_enhanced_aux_ne:nnn #1#2#3
1681 {
1682   \seq_put_right:No \l__physicx_tmp_coled_seq
1683   {
1684     \physicx@matricelement {#1}
1685     { \__physicx_matrix_row_iterate:n {#2} }
1686     { \__physicx_matrix_col_iterate:n {#3} }
1687   }
1688 }

```

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

\generalmatrix Define \generalmatrix.

```

1689 \DeclareDocumentCommand \generalmatrix { t= t+ s m }
1690 {
1691   \IfBooleanTF {#2}
1692   {
1693     \group_begin:
1694     \IfBooleanTF {#1}
1695     { \keys_set:nn { physicx/matrix } { #4 , saveto = \physicx_tmp } }
1696     { \keys_set:nn { physicx/matrix } {#4} }
1697     \bool_set:Nn \l__physicx_matrix_infinite_bool {#3}
1698     \physicx_construct:nnn
1699     {
1700       \tl_if_empty:NTF \l__physicx_matrix_main_tl
1701       {
1702         \physicx_matrix_array_parse:o \l__physicx_matrix_array_tl
1703       }
1704       { \physicx_matrix_array_parse_main: }
1705     }
1706     { \physicx_matrix_diag_parse:o \l__physicx_matrix_diag_clist }
1707     { \physicx_matrix_item_parse:o \l__physicx_matrix_item_clist }
1708     \__physicx_generalmatrix:
1709     \__physicx_matrix_save_or_print:
1710     \group_end:
1711   }
1712   {
1713     \IfBooleanTF {#1}
1714     { \IfBooleanTF {#3} { } { \use_i_ii:nnn } }
1715     { \IfBooleanTF {#3} { \use_i:nn } { \use_i:nnn } }
1716     \qxmatrix = * [#4]
1717   }
1718 }
1719 \cs_new:Npn \__physicx_generalmatrix:

```

```

1720 {
1721   \bool_if:NTF \l__physicx_matrix_expand_element_bool
1722   {
1723     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1724     \__physicx_matrix_appto_body_e:off
1725   }
1726   {
1727     \cs_set_eq:NN \__physicx_generalmatrix_generate:nnn
1728     \__physicx_matrix_appto_body_ne:off
1729   }
1730   \__physicx_matrix_transpose:N
1731   \__physicx_matrix_generate_body:NNNN
1732   \__physicx_generalmatrix_generate:nnn
1733 }

```

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

`\__physicx_matrix_generate_body:NNNN`

```

1734 % row, col, \use:nn or \use_ii_i:nn, appto body cmd
1735 \cs_new:Npn \__physicx_matrix_generate_body:NNNN #1#2#3#4
1736 {
1737   \__physicx_matrix_enhanced_init:
1738   \int_step_inline:nn { #1 - 1 }
1739   {
1740     \int_step_inline:nn { #2 - 1 }
1741     {
1742       \tl_set:Nx \l__physicx_tmp_tl
1743       {
1744         \exp_after:wN
1745         \physicx_matrix_use_r_c:nn
1746         #3 {{##1}} {{####1}}
1747       }
1748       #4 \l__physicx_tmp_tl {##1} {####1}
1749       \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1750     }
1751     \tl_set:Nx \l__physicx_tmp_tl
1752     {
1753       \exp_after:wN
1754       \physicx_matrix_use_r_c:nn
1755       #3 {{##1}} {{ \int_use:N #2 }}
1756     }
1757     #4 \l__physicx_tmp_tl {##1} { \int_use:N #2 }
1758     \tl_put_right:Nx \l__physicx_matrix_body_tl
1759     { \__physicx_matrix_sep: }
1760   }
1761   \int_step_inline:nn { #2 - 1 }
1762   {
1763     \tl_set:Nx \l__physicx_tmp_tl
1764     {
1765       \exp_after:wN
1766       \physicx_matrix_use_r_c:nn
1767       #3 {{ \int_use:N #1 }} {{##1}}
1768     }
1769     #4 \l__physicx_tmp_tl { \int_use:N #1 } {##1}

```

```

1770     \tl_put_right:Nn \l__physicx_matrix_body_tl { & }
1771   }
1772   \tl_set:Nx \l__physicx_tmp_tl
1773   {
1774     \exp_after:wN
1775     \physicx_matrix_use_r_c:nn
1776     #3 {{ \int_use:N #1 }} {{ \int_use:N #2 }}
1777   }
1778   #4 \l__physicx_tmp_tl { \int_use:N #1 } { \int_use:N #2 }
1779 }

```

(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
1780 \cs_new:Npn \__physicx_matrix_appto_body_e:nnn #1#2#3
1781 {
1782   \tl_put_right:Nx \l__physicx_matrix_body_tl
1783   {
1784     \text_expand:n
1785     {
1786       \physicx@matricelement {#1}
1787       { \__physicx_matrix_row_iterate:n {#2} }
1788       { \__physicx_matrix_col_iterate:n {#3} }
1789     }
1790   }
1791 }
1792 \cs_generate_variant:Nn \__physicx_matrix_appto_body_e:nnn { off, xff }
1793 \cs_new:Npn \__physicx_matrix_appto_body_ne:nnn #1#2#3
1794 {
1795   \tl_put_right:No \l__physicx_matrix_body_tl
1796   {
1797     \physicx@matricelement {#1}
1798     { \__physicx_matrix_row_iterate:n {#2} }
1799     { \__physicx_matrix_col_iterate:n {#3} }
1800   }
1801 }
1802 \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

```

1803 \cs_new:Npn \__physicx_matrix_transpose:N #1 % generate body command
1804 {
1805   \bool_if:NTF \l__physicx_matrix_transpose_bool
1806   {
1807     #1
1808     \l__physicx_matrix_cols_int
1809     \l__physicx_matrix_rows_int
1810     \use_ii_i:nn
1811   }
1812   {
1813     #1
1814     \l__physicx_matrix_rows_int
1815     \l__physicx_matrix_cols_int
1816     \use:nn

```

```

1817     }
1818 }

(End definition for \_physicx_matrix_transpose:N.)

```

\\_physicx\_matrix\_sep:

```

1819 \cs_new:Npn \_physicx_matrix_sep:
1820 {
1821   \dim_compare:nNnTF \l__physicx_matrix_sep_dim = \c_zero_dim
1822     { \ } { \[\dim_use:N \l__physicx_matrix_sep_dim] }
1823 }

(End definition for \_physicx_matrix_sep:.)

```

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

```

1824 \cs_new:Npn \physicx_construct:nnn #1#2#3
1825 {
1826   \l__physicx_matrix_beginning_tl
1827   \_physicx_adi:nnn {#1} {#2} {#3}
1828   \tl_if_empty:NF \l__physicx_matrix_last_col_tl
1829     {
1830       \int_incr:N \l__physicx_matrix_cols_int
1831       \_physicx_matrix_last_aux_c:
1832       \int_incr:N \l__physicx_matrix_cols_int
1833     }
1834   \tl_if_empty:NF \l__physicx_matrix_last_row_tl
1835     {
1836       \int_incr:N \l__physicx_matrix_rows_int
1837       \_physicx_matrix_last_aux_r:
1838       \int_incr:N \l__physicx_matrix_rows_int
1839     }
1840   \bool_lazy_or:nnF
1841     { \tl_if_empty_p:N \l__physicx_matrix_last_row_tl }
1842     { \tl_if_empty_p:N \l__physicx_matrix_last_col_tl }
1843     {
1844       \physicx_matrix_set_r_c:nnn
1845         { \int_eval:n { \l__physicx_matrix_rows_int - 1 } }
1846         { \int_eval:n { \l__physicx_matrix_cols_int - 1 } }
1847         { \ddots }
1848     }
1849   \bool_if:NT \l__physicx_matrix_infinite_bool
1850     {
1851       \int_incr:N \l__physicx_matrix_rows_int
1852       \int_incr:N \l__physicx_matrix_cols_int
1853       \_physicx_matrix_last_aux_c:
1854       \_physicx_matrix_last_aux_r:
1855       \physicx_matrix_set_r_c:nnn
1856         { \int_use:N \l__physicx_matrix_rows_int }
1857         { \int_use:N \l__physicx_matrix_cols_int }
1858         { \ddots }
1859     }
1860   \l__physicx_matrix_ending_tl
1861 }

```

```

1862 \cs_new:Npn \__physicx_matrix_last_aux_c:
1863 {
1864   \int_step_inline:nn \l__physicx_matrix_rows_int
1865   {
1866     \physicx_matrix_set_r_c:nnn
1867     {##1} { \int_use:N \l__physicx_matrix_cols_int }
1868     { \cdots }
1869   }
1870 }
1871 \cs_new:Npn \__physicx_matrix_last_aux_r:
1872 {
1873   \int_step_inline:nn \l__physicx_matrix_cols_int
1874   {
1875     \physicx_matrix_set_r_c:nnn
1876     { \int_use:N \l__physicx_matrix_rows_int } {##1}
1877     { \vdots }
1878   }
1879 }

```

(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 1880 \cs_new:Npn \__physicx_new_matrix_cmd:NNN #1#2#3
\NewGeneralMatrix 1881 {
\newdiagonalmatrix 1882   \NewDocumentCommand #2 { t+ m o o m m }
\NewDiagonalMatrix 1883   {
\newcommamatrix 1884     \IfBooleanTF {##1}
\NewCommaMatrix 1885     {
1886       \IfNoValueTF {##3}
1887       { \newcommand ##2 { #1 + [##5] {##6} } }
1888       {
1889         \IfNoValueTF {##4}
1890         { \newcommand ##2 [##3] { #1 + [##5] {##6} } }
1891         { \newcommand ##2 [##3] [##4] { #1 + [##5] {##6} } }
1892       }
1893     }
1894     {
1895       \IfNoValueTF {##3}
1896       { \newcommand ##2 { #1 [##5] {##6} } }
1897       {
1898         \IfNoValueTF {##4}
1899         { \newcommand ##2 [##3] { #1 [##5] {##6} } }
1900         { \newcommand ##2 [##3] [##4] { #1 [##5] {##6} } }
1901       }
1902     }
1903   }
1904   \NewDocumentCommand #3 { t+ m m m m }
1905   {
1906     \IfBooleanTF {##1}
1907     { \NewDocumentCommand ##2 {##3} { #1 + [##4] {##5} } }
1908     { \NewDocumentCommand ##2 {##3} { #1 [##4] {##5} } }
1909   }

```

```

1910 }
1911 \_physicx_new_matrix_cmd:NNN \diagonalmatrix \newdiagonalmatrix \NewDiagonalMatrix
1912 \_physicx_new_matrix_cmd:NNN \commamatrix \newcommamatrix \NewCommaMatrix
1913 \NewDocumentCommand \newgeneralmatrix { t+ m o o m }
1914 {
1915   \IfBooleanTF {#1}
1916   {
1917     \IfNoValueTF {#3}
1918     { \newcommand #2 { \generalmatrix + {#5} } }
1919     {
1920       \IfNoValueTF {#4}
1921       { \newcommand #2 [#3] { \generalmatrix + {#5} } }
1922       { \newcommand #2 [#3] [#4] { \generalmatrix + {#5} } }
1923     }
1924   }
1925   {
1926     \IfNoValueTF {#3}
1927     { \newcommand #2 { \generalmatrix {#5} } }
1928     {
1929       \IfNoValueTF {#4}
1930       { \newcommand #2 [#3] { \generalmatrix {#5} } }
1931       { \newcommand #2 [#3] [#4] { \generalmatrix {#5} } }
1932     }
1933   }
1934 }
1935 \NewDocumentCommand \NewGeneralMatrix { t+ m m m }
1936 {
1937   \IfBooleanTF {#1}
1938   { \NewDocumentCommand #2 {#3} { \generalmatrix + {#4} } }
1939   { \NewDocumentCommand #2 {#3} { \generalmatrix {#4} } }
1940 }

```

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

```

1941 \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			
$\backslash +$	182	<code>\absolutevalue</code>	<a href="#">718</a>
$\backslash -$	182	<code>\acomm</code>	<a href="#">718</a>
$\backslash \backslash$	57, 1822	<code>\aftergroup</code>	232
$\backslash \{$	417, 647, 666, 684, 725, 750	<code>\anticommutator</code>	<a href="#">718</a>
$\backslash \}$	417, 647, 668, 684, 725, 750	<code>\AtBeginDocument</code>	267, 290, 303, 314, 331
A		B	
$\backslash A$	177, 182	<code>\begin</code>	422,
$\backslash \text{abs}$	<a href="#">718</a>		423, 424, 425, 426, 427, 428, 431,
			432, 433, 434, 435, 436, 437, 438,

439, 440, 441, 442, 443, 444, 445,  
 446, 447, 667, 670, 674, 677, 678,  
 679, 684, 685, 689, 692, 693, 694,  
 1055, 1056, 1057, 1058, 1059, 1060,  
 1061, 1064, 1065, 1066, 1067, 1068,  
 1069, 1070, 1071, 1072, 1073, 1074,  
 1075, 1076, 1077, 1078, 1079, 1080  
 \bgroup ..... 231, 233  
 \Big ..... 700  
 \big ..... 700  
 \Bigg ..... 700  
 \bigg ..... 700  
 \Biggl ..... 707  
 \biggl ..... 706  
 \Biggr ..... 707  
 \biggr ..... 706  
 \Bigl ..... 705  
 \bigl ..... 704  
 \Bigr ..... 705  
 \bigr ..... 704  
 \boldsymbol ..... 221  
 bool commands:  
 \bool\_case\_false:n .... 617, 618, 625  
 \bool\_case\_true:nTF ..... 702, 1468  
 \bool\_if:NTF ..... 16,  
 21, 26, 85, 210, 265, 306, 312, 325,  
 631, 633, 637, 1130, 1185, 1192,  
 1202, 1210, 1222, 1229, 1233, 1240,  
 1244, 1508, 1633, 1721, 1805, 1849  
 \bool\_if\_p:N ..... 1506, 1594  
 \bool\_if\_p:n .....  
 ..... 704, 705, 706, 707, 1505, 1593  
 \bool\_lazy\_and\_p:nn .... 1473, 1478  
 \bool\_lazy\_or:nnTF ..... 31,  
 137, 152, 470, 475, 1504, 1592, 1840  
 \bool\_new:N .....  
 .... 9, 10, 11, 12, 13, 51, 66, 533,  
 535, 809, 810, 811, 819, 833, 841, 842  
 \bool\_set:Nn ..... 544, 545, 1697  
 \bool\_set\_false:N .....  
 . 95, 198, 277, 1118, 1146, 1161, 1278  
 \bool\_set\_inverse:N ..... 879, 1005  
 \bool\_set\_true:N ..... 140,  
 155, 165, 205, 263, 276, 1117, 1150,  
 1154, 1165, 1169, 1272, 1571, 1577  
 \c\_false\_bool ..... 662, 682  
 \c\_true\_bool ..... 645, 655, 662, 682  
 \Bqty ..... 718  
 \bqty ..... 718  
  
 C  
 \cdots .... 905, 1184, 1187, 1188, 1191,  
 1193, 1224, 1225, 1228, 1230, 1868

clist commands:  
 \clist\_clear:N ..... 1086  
 \clist\_concat:NNN ..... 1093  
 \clist\_if\_empty:nTF ..... 237  
 \clist\_if\_in:NnTF ..... 1421  
 \clist\_map\_break:n ..... 191, 203  
 \clist\_map\_inline:nn 82, 187, 199, 1091  
 \clist\_new:N ..... 50, 817, 818  
 \clist\_put\_right:Nn .. 967, 973, 1412  
 \clist\_set:Nn ..... 1083  
 \clist\_set\_eq:NN ..... 1388  
 \c\_empty\_clist ..... 1388  
 \comm ..... 748  
 \commamatrix ..... 32, 1582, 1912  
 \commutator ..... 718  
 \crossproduct ..... 299  
 cs commands:  
 \cs:w ..... 32, 33, 996  
 \cs\_end: ..... 32, 33, 772, 996  
 \cs\_generate\_variant:Nn .....  
 ..... 3, 4, 5, 6, 89, 92, 213,  
 488, 628, 1256, 1394, 1451, 1792, 1802  
 \cs\_gset\_eq:NN ..... 40, 41, 42  
 \cs\_if\_exist:NTF ..... 394  
 \cs\_if\_exist\_p:N ..... 1474, 1479  
 \cs\_if\_exist\_use:NTF ..... 37  
 \cs\_if\_free:NTF ..... 335  
 \cs\_new:Npn .....  
 .. 7, 110, 135, 143, 146, 162, 173,  
 185, 219, 223, 227, 231, 232, 233,  
 234, 235, 455, 467, 538, 548, 560,  
 597, 604, 629, 698, 844, 1045, 1047,  
 1249, 1257, 1285, 1298, 1309, 1319,  
 1345, 1384, 1386, 1395, 1396, 1430,  
 1452, 1526, 1531, 1567, 1624, 1651,  
 1668, 1680, 1719, 1735, 1780, 1793,  
 1803, 1819, 1824, 1862, 1871, 1880  
 \cs\_new\_eq:NN .....  
 ..... 145, 172, 803, 823, 832, 1317  
 \cs\_new\_nopar:Npn 762, 770, 779, 781,  
 786, 792, 825, 826, 829, 830, 831, 1607  
 \cs\_new\_protected:Nn ..... 1128  
 \cs\_new\_protected:Npn ..... 67,  
 72, 77, 90, 93, 215, 333, 337, 343, 349  
 \cs\_new\_protected\_nopar:Npn .....  
 ..... 1082, 1084, 1089  
 \cs\_set:Nn .....  
 .. 1010, 1011, 1012, 1013, 1014, 1015  
 \cs\_set:Npn .....  
 .... 114, 116, 124, 126, 289, 493,  
 499, 503, 721, 731, 732, 748, 753,  
 754, 755, 885, 894, 903, 912, 921, 930  
 \cs\_set:Npx ..... 937



<code>\cs_set_eq:NN</code> . . . . .	69, 70, 74, 75, 221, 225, 229, 269, 270, 271, 304, 305, 306, 317, 318, 319, 328, 801, 854, 856, 858, 861, 867, 869, 883, 892, 901, 910, 919, 928, 946, 949, 952, 955, 958, 1132, 1136, 1263, 1268, 1273, 1279, 1510, 1514, 1723, 1727
<code>\cs_set_nopar:Npn</code> . . . . .	1572, 1578
<code>\cs_set_protected:Npn</code> . . . . .	609, 622, 641, 716
<code>\cs_to_str:N</code> . . . . .	335, 337, 508, 511, 516, 517, 554
<b>D</b>	
<code>\ddots</code> . . . . .	905, 1211, 1242, 1847, 1858
<code>\DeclareDocumentCommand</code> . . . . .	291, 293, 295, 298, 299, 300, 307, 308, 614, 624, 700, 1113, 1487, 1582, 1689
<code>\diagonalmatrix</code> . . . . .	30, 1487, 1911
<code>\diffd</code> . . . . .	306
dim commands:	
<code>\dim_compare:nNnTF</code> . . . . .	1821
<code>\dim_new:N</code> . . . . .	843
<code>\dim_use:N</code> . . . . .	1822
<code>\c_zero_dim</code> . . . . .	1821
<code>\div</code> . . . . .	269, 304, 305, 319
<code>\divergence</code> . . . . .	305
<code>\divisionsymbol</code> . . . . .	269, 304, 319
<code>\dotproduct</code> . . . . .	298
<b>E</b>	
<code>\egroup</code> . . . . .	232, 234
else commands:	
<code>\else:</code> . . . . .	774, 1196, 1323, 1333, 1358, 1370, 1663
<code>\end</code> . . . . .	422, 423, 424, 425, 426, 427, 428, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 667, 670, 674, 677, 678, 679, 684, 685, 689, 692, 693, 694, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1080
<code>\eval</code> . . . . .	718
<code>\evaluated</code> . . . . .	643, 731
exp commands:	
<code>\exp_after:wN</code> . . . . .	1539, 1546, 1555, 1562, 1609, 1611, 1619, 1744, 1753, 1765, 1774
<code>\exp_args:Nc</code> . . . . .	347, 507, 510
<code>\exp_args:Nf</code> . . . . .	1293
<code>\exp_args:NNc</code> . . . . .	346
<code>\exp_args:Nnno</code> . . . . .	1423, 1458
<code>\exp_args:NNnx</code> . . . . .	351
<code>\exp_args:Nno</code> . . . . .	189, 201
<code>\exp_args:NNx</code> . . . . .	513
<code>\exp_args:No</code> . . . . .	923, 932, 1034, 1405
<code>\exp_args:NV</code> . . . . .	1638, 1646
<code>\exp_end:</code> . . . . .	8
<code>\exp_not:N</code> 8, 397, 515, 516, 517, 568, 937	
<code>\exp_not:n</code> . . . . .	572, 573, 773, 775, 823, 854, 858
<code>\exp_stop_f:</code> . . . . .	563, 570, 1321, 1324, 1347, 1359
<b>F</b>	
<code>\fbraces</code> . . . . .	307, 308
fi commands:	
<code>\fi:</code> . . . . .	574, 589, 594, 595, 696, 757, 776, 1199, 1342, 1343, 1381, 1382, 1618, 1665
fp commands:	
<code>\fp_eval:n</code> . . . . .	29, 1471
<code>\fpeval</code> . . . . .	29, 1479, 1480
<b>G</b>	
<code>\generalmatrix</code> . . . . .	34, 1689, 1918, 1921, 1922, 1927, 1930, 1931, 1938, 1939
group commands:	
<code>\group_begin:</code> . . . . .	457, 1115, 1489, 1584, 1693
<code>\group_end:</code> . . . . .	465, 1126, 1524, 1605, 1710
<b>H</b>	
<code>\hat</code> . . . . .	296
hook commands:	
<code>\hook_gput_code:nnn</code> . . . . .	37, 327, 329
<b>I</b>	
if commands:	
<code>\if_bool:N</code> . . . . .	578, 590, 643, 718
<code>\if_case:w</code> . . . . .	570
<code>\if_cs_exist:w</code> . . . . .	772
<code>\if_int_compare:w</code> . . . . .	563, 1194, 1321, 1324, 1347, 1359, 1616, 1661
<code>\IfBooleanT</code> . . . . .	666, 668, 1588
<code>\IfBooleanTF</code> . . . . .	292, 294, 296, 356, 524, 616, 673, 675, 688, 690, 710, 1051, 1116, 1121, 1490, 1691, 1694, 1713, 1714, 1715, 1884, 1906, 1915, 1937
<code>\IfNoValueTF</code> . . . . .	307, 308, 491, 497, 1886, 1889, 1895, 1898, 1917, 1920, 1926, 1929
<code>\Im</code> . . . . .	271, 308, 318
<code>\imaginary</code> . . . . .	271, 308, 318
int commands:	
<code>\int_compare:nNnTF</code> . . . . .	148, 150, 164



<code>\physicx_declare_legacy_quantity:nnNn</code>	<code>\physicx_set_parse_range_-</code>
.... 531, 548, 642, 644, 654, 661, 681	delimiter:n ..... 110, 134
<code>\physicx_if_num:n</code> ..... 175	<code>\physicx_short:</code> ..... 19
<code>\physicx_if_num:nTF</code> .....	<code>\physicx_unimath:</code> ..... 44
..... 1034, 1142, 1157, 1465	<code>\physicx_unimath:TF</code> .. 40, 41, 42, 288
<code>\physicx_if_num_sign:n</code> ..... 180	<code>\physicx_use_amssymb_type:</code> . 219, 287
<code>\physicx_left:</code> .... 231, 478, 612, 712	<code>\physicx_use_uni_bf_type:</code> ..... 227
<code>\physicx_left:N</code> .....	<code>\physicx_use_uni_bfit_type:</code> 223, 286
..... 233, 480, 704, 705, 706, 707	<code>\physicx_xquantity:nn</code> .....
<code>\physicx_mathtools:</code> ..... 24	.... 455, 455, 487, 509, 512, 525, 526
<code>\physicx_mathtools:TF</code> .... 429, 1062	physicx internal commands:
<code>\physicx_matrix_array_parse:n</code> ...	<code>\__physicx_adi:nnn</code> .... 844, 1010,
..... 1430, 1430, 1451, 1702	1011, 1012, 1013, 1014, 1015, 1827
<code>\physicx_matrix_array_parse_-</code>	<code>\l__physicx_begin_range_int</code> ....
main: ..... 1452, 1452, 1704	..... 62, 102,
<code>\physicx_matrix_diag_parse:n</code> ...	103, 148, 149, 153, 154, 158, 164, 168
..... 968, 1249,	<code>\l__physicx_cmd_arg_int</code> .....
1249, 1256, 1495, 1499, 1500, 1706	..... 537, 543, 562, 563, 570
<code>\physicx_matrix_item_parse:n</code> ...	<code>\l__physicx_cmd_arg_spec_tl</code> ....
... 974, 1386, 1386, 1394, 1503, 1707	..... 536, 542, 555, 564
<code>\physicx_matrix_new_type:nn</code> ....	<code>\l__physicx_cmd_auto_body_bool</code> ..
..... 1045, 1047, 1052	..... 535, 545, 590, 633, 637
<code>\physicx_matrix_new_type:nnn</code> ...	<code>\l__physicx_cmd_auto_body_tl</code> ...
..... 1045, 1045, 1053	..... 534, 541, 557, 591, 592
<code>\physicx_matrix_set_r_c:nnn</code> 803,	<code>\l__physicx_cmd_noauto_body_bool</code>
946, 949, 952, 955, 958, 1304, 1312,	..... 533, 544, 578, 631
1327, 1336, 1353, 1365, 1376, 1423,	<code>\l__physicx_cmd_noauto_body_tl</code> ..
1447, 1458, 1844, 1855, 1866, 1875	..... 532, 540, 556, 579, 580
<code>\physicx_matrix_use_r_c:nn</code> .....	<code>\l__physicx_col_list_seq</code> .....
..... 770, 1540, 1547,	..... 821, 875, 1575, 1579
1556, 1563, 1745, 1754, 1766, 1775	<code>\__physicx_commamatrix_enhanced:</code>
<code>\physicx_new_type:nnn</code> .....	..... 1595, 1624
..... 215, 415, 1046, 1048	<code>\__physicx_commamatrix_enhanced_-</code>
<code>\physicx_option_or:nn</code> ..... 29	aux:nNn ..... 1638, 1646, 1651
<code>\physicx_option_or:nnTF</code> .....	<code>\__physicx_commamatrix_enhanced_-</code>
..... 719, 729, 747, 751	aux_e:nnn ..... 1639, 1668
<code>\c_physicx_Order_tl</code> 218, 282, 739, 759	<code>\__physicx_commamatrix_enhanced_-</code>
<code>\c_physicx_order_tl</code> 217, 281, 743, 761	aux_ne:nnn ..... 1647, 1680
<code>\physicx_parse_range:nnN</code> .....	<code>\g__physicx_compat_bool</code> .....
..... 90, 92, 1401, 1403	..... 11, 16, 242, 263, 265
<code>\physicx_parse_range:nnnN</code> .....	<code>\__physicx_declare_init:</code> ..... 1567
..... 62, 77, 89, 91	<code>\__physicx_declare_init:nnn</code> 538, 550
<code>\physicx_parse_range_check:</code> ....	<code>\__physicx_declare_legacy_-</code>
..... 62, 67, 978	quantity_aux:nnnn ..... 560, 601
<code>\physicx_parse_range_nocheck:</code> ...	<code>\__physicx_declare_legacy_-</code>
..... 62, 72, 979	quantity_aux:NNnnn .. 553, 604, 628
<code>\physicx_qxmatrix:nnn</code> ... 1124, 1128	<code>\__physicx_declare_legacy_-</code>
<code>\physicx_right:</code> ... 232, 478, 612, 712	quantity_aux:nw .... 551, 597, 602
<code>\physicx_right:N</code> .....	<code>\__physicx_diagonalmatrix_diag_-</code>
..... 234, 482, 704, 705, 706, 707	main: ..... 1273, 1279, 1317, 1322
<code>\physicx_search_also:nn</code> .....	<code>\__physicx_diagonalmatrix_-</code>
..... 185, 196, 213, 214	enhanced:nnn .... 1510, 1514, 1529
<code>\physicx_search_also:nnTF</code> . 403, 1026	<code>\__physicx_diagonalmatrix_-</code>
	generate_body:NNN .... 1522, 1531

```

\__physicx_diagonalmatrix_-
  generate_enhanced_body:NNN ...
    ..... 1521, 1526
\__physicx_diagonalmatrix_no_-
  diag: ..... 1280, 1309, 1318
\__physicx_diagonalmatrix_set_-
  diag: ..... 1274, 1298
\__physicx_do_nothing: ..... 115
\l__physicx_end_range_int ... 63,
  105, 106, 150, 151, 154, 158, 164, 168
\__physicx_expand:w ... 823, 854,
  856, 858, 861, 1289, 1400, 1434, 1591
\__physicx_fixdif:Nn ..... 339, 349
\__physicx_fixdif:Nnn . 339, 343, 351
\g__physicx_fixdif_bool 256, 306, 325
\__physicx_fixdif_list:N ... 330, 333
\__physicx_generalmatrix: 1708, 1719
\__physicx_generalmatrix_-
  generate:nnn .... 1723, 1727, 1732
\__physicx_if_can_num:n .. 1463, 1463
\__physicx_if_keyval:nTF .. 173, 1498
\l__physicx_invalid_range_bool ..
  ..... 66, 85, 95, 140, 155, 165
\l__physicx_item_ignore_clist ...
  ..... 1388, 1412, 1421
\__physicx_loadpackage_options:nnn
  ..... 235, 246, 248, 250
\g__physicx_mathtools_bool .....
  ..... 9, 26, 276, 277
\__physicx_matrix_add_data:nn ...
  ..... 970, 976, 1087, 1089
\l__physicx_matrix_after_begin_-
  tl ..... 839, 986, 988, 1619
\l__physicx_matrix_after_end_tl .
  ..... 840, 989, 991, 1621
\__physicx_matrix_appto_body_-
  e:nnn .....
  .. 1133, 1511, 1724, 1780, 1780, 1792
\__physicx_matrix_appto_body_-
  ne:nnn .....
  .. 1137, 1515, 1728, 1780, 1793, 1802
\l__physicx_matrix_args_tl .....
  ..... 838, 984, 985, 1619
\l__physicx_matrix_array_tl ....
  ..... 812, 851, 1590,
  1591, 1597, 1599, 1602, 1629, 1702
\__physicx_matrix_autocalc:nn ...
  ..... 832, 867, 869, 1385, 1436, 1442
\__physicx_matrix_begin:w .....
  ..... 830, 981, 1619
\l__physicx_matrix_beginning_tl .
  ..... 845, 1016, 1018, 1826
\l__physicx_matrix_body_tl .....
  ..... 813, 1140, 1175,
  1180, 1181, 1187, 1188, 1193, 1198,
  1205, 1207, 1209, 1211, 1212, 1214,
  1218, 1219, 1224, 1225, 1230, 1236,
  1237, 1239, 1241, 1242, 1537, 1544,
  1553, 1560, 1601, 1613, 1620, 1626,
  1658, 1749, 1758, 1770, 1782, 1795
\__physicx_matrix_calc:nn .....
  ..... 762, 867, 1264
\__physicx_matrix_col_iterate:n .
  829, 962, 1578, 1676, 1686, 1788, 1799
\l__physicx_matrix_cols_int ....
  ..... 766, 767,
  815, 864, 1120, 1159, 1162, 1178,
  1208, 1216, 1238, 1245, 1246, 1300,
  1303, 1307, 1355, 1367, 1378, 1403,
  1456, 1616, 1617, 1808, 1815, 1830,
  1832, 1846, 1852, 1857, 1867, 1873
\l__physicx_matrix_diag_bool ...
  ..... 819, 1272, 1278
\__physicx_matrix_diag_calc:nn ..
  ..... 1263, 1268, 1313,
  1330, 1339, 1348, 1360, 1371, 1384
\l__physicx_matrix_diag_clist ...
  ..... 817, 965, 967, 1495, 1706
\__physicx_matrix_diag_parse_-
  aux:n ..... 1252, 1257
\__physicx_matrix_diag_parse_-
  aux:nn ..... 1253, 1285
\__physicx_matrix_diag_parse_-
  aux_anti:n ..... 1293, 1345
\__physicx_matrix_diag_parse_-
  aux_regu:n ..... 1296, 1319
\l__physicx_matrix_dotcol_bool ..
  ..... 811, 1161, 1165,
  1169, 1185, 1210, 1222, 1240, 1244
\l__physicx_matrix_dotrow_bool ..
  ..... 810, 1146, 1150, 1154, 1202
\__physicx_matrix_element_-
  aux:nnn .... 883, 889, 892, 898,
  901, 907, 910, 916, 919, 925, 928, 933
\__physicx_matrix_end: .... 982, 1621
\__physicx_matrix_end:w ..... 831
\l__physicx_matrix_ending_tl ...
  ..... 846, 1019, 1021, 1860
\l__physicx_matrix_enhanced_bool
  ..... 842, 1002, 1005, 1506, 1594
\__physicx_matrix_enhanced_init:
  ..... 1567, 1632, 1737
\l__physicx_matrix_expand_-
  element_bool ..... 833,
  941, 1130, 1508, 1571, 1577, 1633, 1721
\__physicx_matrix_generate_-
  body:NNNN .. 1528, 1731, 1734, 1735

```

\l__physicx_matrix_infinite_bool	\l__physicx_matrix_transpose_-
..... 809, 876, 879, 1117,	bool ..... 841, 997, 1805
1118, 1192, 1229, 1233, 1697, 1849	\l__physicx_max_range_int .....
\l__physicx_matrix_item_clist ...	..... 64, 81, 105, 138, 150, 151, 153
..... 818, 971, 973, 1503, 1707	\l__physicx_min_range_int .....
\__physicx_matrix_item_parse_-	..... 65, 80, 102, 139, 148, 149
aux:n ..... 1390, 1395	\__physicx_nabla: ..... 328, 331
\__physicx_matrix_item_parse_-	\__physicx_nauto_case:nnnn . 606, 629
aux:nn ..... 1391, 1396	\__physicx_new_matrix_cmd:NNN ...
\__physicx_matrix_last_aux_c: ...	..... 1880, 1880, 1911, 1912
..... 1831, 1853, 1862	\__physicx_new_xquantity_aux:w ..
\__physicx_matrix_last_aux_r: ...	..... 493, 499, 503, 507, 510
..... 1837, 1854, 1871	\g__physicx_original_bool .. 257, 312
\l__physicx_matrix_last_col_tl ..	\__physicx_parse_range_action:nnn
..... 828, 964, 1828, 1842	..... 97, 114, 124
\l__physicx_matrix_last_row_tl ..	\__physicx_parse_range_aux:n 84, 93
..... 827, 963, 1834, 1841	\__physicx_parse_range_aux:w ...
\l__physicx_matrix_main_tl .....	..... 115, 116, 125, 126
..... 816, 871, 1459, 1700	\__physicx_parse_range_range: ...
\__physicx_matrix_row_iterate:n .	..... 70, 75, 107, 172
826, 961, 1572, 1675, 1685, 1787, 1798	\__physicx_parse_range_range_-
\l__physicx_matrix_rows_int ....	check: ..... 70, 146, 172
..... 764, 765, 814, 863, 1119,	\__physicx_parse_range_range_-
1144, 1147, 1172, 1194, 1306, 1401,	nocheck: ..... 75, 162
1454, 1630, 1661, 1809, 1814, 1836,	\__physicx_parse_range_single:n .
1838, 1845, 1851, 1856, 1864, 1876	..... 69, 74, 99, 145
\__physicx_matrix_save_or_print:	\__physicx_parse_range_single_-
..... 1125, 1523, 1604, 1607, 1709	check:n ..... 69, 135, 145
\l__physicx_matrix_save_tl .....	\__physicx_parse_range_single_-
..... 994, 996, 1609, 1612	nocheck:n ..... 74, 143
\__physicx_matrix_sep: .....	\g__physicx_physics_bool ..... 10
..... 1198, 1205, 1212, 1236,	\l__physicx_quantity_args_tl ...
1548, 1598, 1664, 1759, 1819, 1819	..... 360, 382, 383, 462
\l__physicx_matrix_sep_dim .....	\l__physicx_quantity_code_tl ...
..... 843, 992, 1821, 1822	..... 361, 384, 459, 463
\__physicx_matrix_set_data:nn ...	\l__physicx_quantity_left_size_-
..... 969, 975, 1084	tl ..... 362, 376, 471, 476, 480
\__physicx_matrix_set_r_c_-	\l__physicx_quantity_left_tl ...
ckall:nnn ..... 801, 959	..... 363, 374, 461
\__physicx_matrix_set_r_c_-	\l__physicx_quantity_post_tl ...
ckep:nnn ..... 786, 950	..... 364, 373, 485
\__physicx_matrix_set_r_c_-	\l__physicx_quantity_pre_tl ....
ckig:nnn ..... 781, 953	..... 365, 372, 469
\__physicx_matrix_set_r_c_-	\l__physicx_quantity_right_size_-
ckigep:nnn ..... 792, 802, 956	tl ..... 366, 377, 472, 477, 482
\__physicx_matrix_set_r_c_-	\l__physicx_quantity_right_tl ...
nock:nnn ..... 779, 804, 947	..... 367, 375, 464
\l__physicx_matrix_tmparr_c_sep .	\__physicx_qxmatrix_appto_-
..... 1441, 1444, 1445	body:nnn ..... 1132, 1136,
\l__physicx_matrix_tmparr_r_sep .	1176, 1182, 1189, 1215, 1220, 1226
..... 1435, 1437, 1439	\g__physicx_reqty_bool .....
\__physicx_matrix_transpose:N ...	..... 13, 252, 643, 718
..... 1520, 1730, 1803, 1803	\l__physicx_row_list_seq .....
	..... 820, 873, 1569, 1573

\g__physicx_short_bool . . .	12, 21, 244	\ProcessKeysPackageOptions . . . . .	260
\l__physicx_tmp_col_seq . .	1653, 1656		
\l__physicx_tmp_coled_seq . . . . .		<b>Q</b>	
..... 1655, 1660, 1670, 1682		\qty . . . . .	718
\l__physicx_tmp_colnum_seq . . . . .		\quantity . . . . .	643, 721
..... 1404, 1410, 1419		quark commands:	
\l__physicx_tmp_rownum_seq . . . . .		\q_nil . . . . .	115
..... 1402, 1408, 1417		\q_physicx_special . . . . .	115, 116, 125, 126
\l__physicx_tmp_seq . . . . .		\quark_if_nil:nTF . . . . .	120
..... 1628, 1631, 1635, 1643		\quark_if_recursion_tail_stop:n . . . . .	599, 600
\l__physicx_tmp_tl . . . . .		\q_recursion_stop . . . . .	552
..... 565, 579, 591, 1742, 1748,		\q_recursion_tail . . . . .	552
1751, 1757, 1763, 1769, 1772, 1778		\qxmtrix . . . . .	1104, 1716
\l__physicx_tmpa_bool 51, 198, 205, 210		\qxqty . . . . .	529
\l__physicx_tmpa_clist . . . . .	50		
\l__physicx_tmpa_int . . . . .		<b>R</b>	
.... 52, 1627, 1637, 1639, 1645, 1647		\rangle . . . . .	421, 651, 678, 693
\l__physicx_tmpa_seq . . . . .		\rbrace . . . . .	307, 308
..... 86, 96, 141, 144, 159, 169		\Re . . . . .	270, 307, 317
\l__physicx_tmpa_tl . . . . .		\real . . . . .	270, 307, 317
.... 101, 103, 118, 119, 128, 390, 391		regex commands:	
\l__physicx_tmparr_tl . . . . .		\regex_match:nnTF . . . . .	177, 182, 923, 932
..... 1432, 1433, 1434, 1435		\renewdif . . . . .	330, 331
\l__physicx_tmpb_int . . . . .	53	\RequirePackage . . . . .	238
\l__physicx_tmpb_tl . . . . .	104, 106, 129	\rgroup . . . . .	675, 690
\l__physicx_tmpdiag_seq 1290, 1301,		\right . . . . .	232, 452, 477, 668,
1311, 1314, 1315, 1325, 1331, 1332,		675, 678, 684, 685, 690, 692, 693, 694	
1334, 1340, 1341, 1349, 1350, 1351,		\rVert . . . . .	735
1361, 1362, 1363, 1372, 1373, 1374			
\l__physicx_tmpdiag_tl . . . . .		<b>S</b>	
..... 1287, 1288, 1289, 1290		\sb 825, 1175, 1180, 1187, 1214, 1218, 1224	
\l__physicx_tmpitem_tl . . . . .		scan commands:	
..... 1398, 1399, 1400, 1406, 1424		\scan_stop: . . . . .	1195, 1662
\__physicx_vnabla: . . . . .	289, 300, 331	seq commands:	
\__physicx_xquantity_aux:nnnn . . . . .		\c_empty_seq . . . . .	79, 1655
..... 460, 467, 488		\seq_clear:N . . . . .	96
\physicxempty . 775, 789, 797, 834, 835, 943		\seq_concat:NNN . . . . .	86
\physicxexcept 837, 914, 923, 932, 938, 940		\seq_count:N . . . . .	1314, 1315, 1331,
\PHYSICXIGNORE . . . . .	7, 8, 783, 794, 1406	1332, 1340, 1341, 1349, 1350, 1361,	
\physicxset . . . . .	354	1362, 1372, 1373, 1437, 1444, 1631	
\physicxtmp . . . . .	531, 1122, 1491, 1589, 1695	\seq_if_empty:NTF . . . . .	1569, 1575
\poissonbracket . . . . .	718	\seq_item:Nn . . . . .	1573, 1579
\pqty . . . . .	718	\seq_map_indexed_inline:Nn . . . . .	
prg commands:		..... 1301, 1311, 1325, 1334,	
\prg_generate_conditional_-		1351, 1363, 1374, 1439, 1445, 1656	
variant:Nnn . . . . .	214	\seq_map_inline:Nn . . . . .	
\prg_new_conditional:Npnn . . . . .		..... 1408, 1410, 1417, 1419	
. . . 14, 19, 24, 29, 175, 180, 196, 1463		\seq_map_tokens:Nn . . . . .	1635, 1643
\prg_replicate:nn . . . . .	1208, 1238	\seq_new:N . . . . .	820, 821
\prg_return_false: . . . . .	17,	\seq_put_right:Nn . . . . .	
22, 27, 35, 47, 178, 183, 211, 1484		..... 141, 144, 159, 169, 1670, 1682	
\prg_return_true: . . . . .	17,	\seq_set_eq:NN . . . . .	79, 1655
22, 27, 34, 47, 178, 183, 211, 1466, 1483		\seq_set_split:Nnn . . . . .	5,
\prg_set_conditional:Npnn . . . . .	44	873, 875, 1290, 1435, 1441, 1628, 1653	

<code>\seq_use:Nn</code> .....	1660	<code>\tl_if_empty:nTF</code> .....	112, 459, 788, 796, 887, 1496, 1586
<code>\setmathfont</code> .....	297	<code>\tl_if_empty_p:N</code> ..	471, 472, 1841, 1842
<code>\setmatrixdata</code> .....	1082	<code>\tl_if_eq:nnTF</code> .....	783, 794, 1405
<code>\setmatrixtype</code> .....	1023, 1045, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1078, 1079, 1080	<code>\tl_if_exist:NTF</code> .....	46
<code>\setquantitytype</code> .....	387, 414, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447	<code>\tl_if_head_eq_charcode:nNTF</code> ..	1291
<code>\smallmatrixquantity</code> .....	643	<code>\tl_if_head_eq_meaning:nNTF</code> ..	345, 352
<code>\smqty</code> .....	718	<code>\tl_if_head_eq_meaning_p:nN</code> ....	1470, 1471, 1475, 1480
str commands:		<code>\tl_if_in:nnTF</code> .....	174, 905, 914
<code>\c_backslash_str</code> .....	391	<code>\tl_if_noalue_p:n</code> .....	568
<code>\str_case_e:nnTF</code> .....	1259	<code>\tl_map_function:nN</code> .....	330
<code>\str_const:Nn</code> .....	806, 807, 808	<code>\tl_new:N</code> .....	360, 361, 362, 363, 364, 365, 366, 367, 531, 532, 534, 536, 812, 813, 816, 827, 828, 835, 837, 838, 839, 840, 845, 846
<code>\str_if_eq:nnTF</code> .....	584, 586	<code>\tl_put_right:Nn</code> .....	564, 579, 580, 591, 592, 940, 988, 991, 1018, 1021, 1175, 1180, 1181, 1187, 1188, 1193, 1198, 1205, 1207, 1209, 1211, 1212, 1214, 1218, 1219, 1224, 1225, 1230, 1236, 1237, 1239, 1241, 1242, 1537, 1544, 1553, 1560, 1658, 1749, 1758, 1770, 1782, 1795
<code>\symbf</code> .....	229, 289, 331	<code>\tl_replace_all:Nnn</code> ...	6, 1597, 1599
<code>\symbfit</code> .....	225	<code>\tl_set:Nn</code> .....	118, 119, 128, 129, 382, 390, 459, 565, 780, 784, 789, 790, 797, 798, 984, 996, 1287, 1288, 1398, 1399, 1432, 1433, 1590, 1742, 1751, 1763, 1772
		<code>\tl_set_eq:NN</code> ..	281, 282, 376, 377, 1601
		<code>\tl_tail:N</code> .....	394, 397
		<code>\tl_tail:n</code> .....	346, 352, 1294
		<code>\tl_trim_spaces:n</code> ..	118, 119, 128, 129
TeX and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> commands:		token commands:	
<code>\@declareparen</code> cmd .....	697	<code>\token_if_cs:NTF</code> .....	1609
<code>\@declarequantity</code> cmd .....	531	<code>\token_if_eq_meaning:NNTF</code> .....	391
<code>\@dif</code> .....	306	<code>\token_if_eq_meaning_p:NN</code> ..	476, 477
<code>\@ifpackageloaded</code> .....	262, 275, 285, 302, 315	<code>\TrimSpaces</code> .....	414, 1049
<code>\@ifstar</code> .....	515	<code>\txqty</code> .....	530
<code>\@pkgextension</code> .....	46		
<code>\c@MaxMatrixCols</code> ....	1001, 1616, 1617		
<code>\physicsx@align</code> .....	806, 1007, 1441, 1600, 1654		
<code>\physicsx@cr</code> .....	807, 1006, 1435, 1598, 1628		
<code>\physicsx@matrixe</code> lement .....	825, 880, 884, 885, 893, 894, 902, 903, 911, 912, 920, 921, 929, 930, 937, 1674, 1684, 1786, 1797		
<code>\physicsx@sep</code> .....	808, 873, 875, 1008, 1290		
tex commands:			
<code>\tex_advance:D</code> .....	1245, 1246		
text commands:			
<code>\text_expand:n</code> .....	856, 1672, 1784		
tl commands:			
<code>\c_empty_tl</code> .....	379, 623		
<code>\tl_clear:N</code> ..	540, 541, 542, 1140, 1626		
<code>\tl_const:Nn</code> .....	217, 218		
<code>\tl_gset_eq:NN</code> .....	1611		
<code>\tl_head:N</code> .....	390		
<code>\tl_if_blank:nTF</code> .....	130, 896		
<code>\tl_if_empty:NTF</code> .....	101, 104, 1700, 1828, 1834		

	<b>V</b>		<b>\vert</b> . 419, 650, 657, 658, 659, 692, 724, 728
<b>\vdots</b> .....	905, 1207,	<b>\vnabla</b> .....	300
	1209, 1211, 1237, 1239, 1241, 1877	<b>\vqty</b> .....	<u>718</u>
<b>\vec</b> .....	294	<b>\vysmbkcircle</b> .....	298
<b>\vectimes</b> .....	299		
<b>\vectorarrow</b> .....	293	<b>X</b>	
<b>\vectorbold</b> .....	291	<b>\xquantity</b> .....	487
<b>\vectorunit</b> .....	295		
<b>\Vert</b> .....	420, 652, 694	<b>Z</b>	
		<b>\Z</b> .....	177, 182